# Acknowledgement

The internship opportunity I had with BCAS campus was a great chance for learning and professional development. Therefore I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having chance to meet so many wonderful people and professionals who led me through this internship period.

I express my deepest thanks to the assessor Mr. Mohamed Nizzad, who works as an IT professional and take lectures on computer programming related subjects and others supportive subjects. He helped me taking part in useful decision & giving necessary advices and guidance and arrange all facilities for this assignment. I choose this moment to acknowledge his contribution gratefully. I acknowledge that this assignment was done with help of internet resources.

Sincerely,

S.SHALOMSHAN

Date:25.03.2021

# Introduction

The primary purpose of this assignment is to give a deep brief explanation about the web design & development concept. This assignment points out some most important essential fundamentals about web design & development. It also focuses on web design & development paradigms and the other required tools for web design & development more.

All of the requirements of this assignment have been done in a perfect manner. The task one is mainly focused on the basic idea of web design & development such as HTML, CSS, JavaScript and php web design & development.

Next I have to be done as a 20 minutes’ presentation. The department of computing at BCAS Kalmunai Campus has decided to design and develop a web application for the benefit of students in order to make their academic progress accessible via a web application.

The more details about each task has been explained in their own introduction section. This main introduction is only giving the basement idea of the assignment.

Contents

[Acknowledgement 1](#_Toc65521999)

[Introduction 2](#_Toc65522000)

[Part:-1 6](#_Toc65522001)

[Introduction 6](#_Toc65522002)

[What Is an Internet Protocol (IP) Address? 7](#_Toc65522003)

[What Is a Domain Name? 7](#_Toc65522004)

[Morphology of a Domain Name 7](#_Toc65522005)

[So What Happens Next? 8](#_Toc65522006)

[Identify the purpose and types of DNS including explanations on how domain name organized and managed 8](#_Toc65522007)

[Resolver 8](#_Toc65522008)

[Iterative 9](#_Toc65522009)

[Root server 9](#_Toc65522010)

[TLD (Top Level Domain) name server 9](#_Toc65522011)

[Authoritative name servers 9](#_Toc65522012)

[URLs 10](#_Toc65522013)

[Part:-2 11](#_Toc65522014)

[Introduction 11](#_Toc65522015)

[Communication protocol 12](#_Toc65522016)

[Basic requirements 12](#_Toc65522017)

[Data formats for data exchange 12](#_Toc65522018)

[Address formats for data exchange 13](#_Toc65522019)

[Address mapping 13](#_Toc65522020)

[Routing 13](#_Toc65522021)

[Detection of transmission errors 13](#_Toc65522022)

[Acknowledgement 13](#_Toc65522023)

[Loss of information-timeouts and retries 13](#_Toc65522024)

[Direction of information flow 14](#_Toc65522025)

[Sequence control 14](#_Toc65522026)

[Flow control 14](#_Toc65522027)

[Queueing 14](#_Toc65522028)

[Protocol design 15](#_Toc65522029)

[Design patterns for application protocols 15](#_Toc65522030)

[Formal specification 15](#_Toc65522031)

[Protocol development 16](#_Toc65522032)

[What is server hardware ? 16](#_Toc65522033)

[What is an operating system ? 16](#_Toc65522034)

[What does an operating system do ? 17](#_Toc65522035)

[What is a web server ? 18](#_Toc65522036)

[Hosting files 19](#_Toc65522037)

[Communicating through HTTP 20](#_Toc65522038)

[Static vs. dynamic content 21](#_Toc65522039)

[Task:-3 22](#_Toc65522040)

[Introduction 22](#_Toc65522041)

[How front-end and back-end relate to presentation and application layer 23](#_Toc65522042)

[Front-End Technologies 24](#_Toc65522043)

[HTML 24](#_Toc65522044)

[CSS 25](#_Toc65522045)

[JavaScript 26](#_Toc65522046)

[Back-End Technologies 27](#_Toc65522047)

[Server-side language 28](#_Toc65522048)

[Server-side frameworks 29](#_Toc65522049)

[Part:-4 30](#_Toc65522050)

[Introduction 30](#_Toc65522051)

[Website Creation Tools 31](#_Toc65522052)

[Unique Customization of Website 31](#_Toc65522053)

[Better SEO 31](#_Toc65522054)

[Extant Support by Professionals 31](#_Toc65522055)

[Better Page Loading Speed 32](#_Toc65522056)

[Ease in E-commerce Tools 32](#_Toc65522057)

[Own Your Website 32](#_Toc65522058)

[Limited Terms and Conditions 32](#_Toc65522059)

[What is UI ? 33](#_Toc65522060)

[A brief history of the user interface 33](#_Toc65522061)

[What is UX ? 34](#_Toc65522062)

[What’s the difference between UI & UX ? 35](#_Toc65522063)

[Part:-5 36](#_Toc65522064)

[Introduction 36](#_Toc65522065)

[Site map 37](#_Toc65522066)

[Wireframe 37](#_Toc65522067)

[A full set of client and user requirements 40](#_Toc65522068)

[Part:-6 41](#_Toc65522069)

[Entire function of the system 41](#_Toc65522070)

[Essential Function 41](#_Toc65522071)

[Technology used in the system 41](#_Toc65522072)

[MySQL database 43](#_Toc65522073)

[Important code snippets 44](#_Toc65522074)

[Add result 44](#_Toc65522075)

[Add students 45](#_Toc65522076)

[Add subject combination 45](#_Toc65522077)

[Change password 46](#_Toc65522078)

[Create class 46](#_Toc65522079)

[Create subject 47](#_Toc65522080)

[Dashboard 47](#_Toc65522081)

[Edit classes 48](#_Toc65522082)

[Edit result 48](#_Toc65522083)

[Edit students 49](#_Toc65522084)

[Edit subjects 49](#_Toc65522085)

[Index 50](#_Toc65522086)

[Logout 50](#_Toc65522087)

[Result 51](#_Toc65522088)

[Part:-7 52](#_Toc65522089)

[Introduction 52](#_Toc65522090)

[Functionality testing 53](#_Toc65522091)

[Usability testing 53](#_Toc65522092)

[Usability testing includes the following 53](#_Toc65522093)

[Other user information for user help 53](#_Toc65522094)

[Interface testing 54](#_Toc65522095)

[Compatibility testing 54](#_Toc65522096)

[Conclusion 55](#_Toc65522097)

[Reference 56](#_Toc65522098)

# Part:-1

# Introduction

Typing in a domain like magic. you quickly punch in a few letters on your keyboard and suddenly you’re sucked into the matrix of the internet. but browsing the web isn’t like solving the mystery on stranger things. it’s straightforward technology this report will explain how domain work and how to domain names are organized and managed and types of DNS.

# What Is an Internet Protocol (IP) Address?

Before we talk domains, it’s important to understand what an IP address is.

Each computer has a unique IP address, which consists of a set of numbers separated by periods (for example, 94.172.0.912). This set of numbers is a language that computers use to communicate with each other over a network.

IP addresses allow any number of internet-connected computers to be distinguished from other computers. It’s just like calling someone on the telephone. Everyone in the world has a unique telephone number, and you have to dial someone’s exact number to reach them.

# What Is a Domain Name?

So what is a domain name? Let’s continue with the phone analogy. You’ve probably got a long list of contacts saved in your mobile phone. Each contact has a unique phone number. When you want to call someone, you usually click on the contact’s name rather than typing in the full phone number.

Domain names are just like a contact in your phone. Rather than typing a complicated set of numbers (the IP address) into your browser, you type in a domain name. That domain name is human-friendly and much easier to remember than an IP address. All domain names are connected to a unique IP address.

## Morphology of a Domain Name

Now that we understand what a domain name is, let’s break it down a bit more. There are three different parts of a domain name that assign meaning to the domain.

* **Domain name:** Bluehost is the domain name in bluehost.com.
* **Top-level domain:** This is the suffix at the end of the URL. Examples include .com, .org, or .blog.
* **Subdomain:** This is a prefix that further classifies a domain, such as subdomain.bluehost.com.

To start a website, you need a domain name, but don’t worry. It’s easy! Bluehost provides you with a free domain name when you purchase hosting. As soon as you select your hosting package, Bluehost will instruct you to choose your domain name. You can select your domain name and your top-level domain.

As long as each part of the domain name is unique, you can pick any domain name you want, and register it with a domain name registry.

## So What Happens Next?

When you register your domain, you are registering with a DNS (Domain Name System). A DNS is a database that connects IP addresses with their corresponding domain names. It’s like your phone’s contact list for all domain names.

When you go to your browser and type in a domain name, it will connect with the DNS. The DNS searches through all of the registered IP addresses and connects that domain name with the IP address.

The end result? The server returns a web page back to your browser in less than a second. It’s sort a like magic.

# Identify the purpose and types of DNS including explanations on how domain name organized and managed

The Domain Name System (DNS) is a directory of names that matches with numbers, the numbers are the IP addresses which computers use to communicate with each other. The DNS is a protocol which uses the TCP/IP protocol set. It helps convert URLs into IP addresses that computers use to identify each other on a network, it is a system that matches names with numbers like a phonebook does. With a DNS you do not need to have the IP address of everyone, you connect to a Domain name server, which holds a large database of domain names and translates them to IP addresses 1. So, when you type in a website like www.bbc.co.uk for example, your internet service provider will request the DNS linked with the domain name and then translates it into an IP address that a computer understands then will direct you to the correct website. If you have previously visited the website, the computer will first check the cache to see if it has already visited the website before, if not it will do a DNS query to find the website.

Main DNS server types

## Resolver

A resolver is the first stop in a DNS query, this acts as the middle link between the client and a DNS name server, once a DNS query has been sent from the client a recursive resolver will bring back cached data if it is found in the cache, or it will send a request to a root name server, then another request to a TLD (Top Level Domain) name server and then a last request to an authoritative name server. If the recursive resolver does not have the website that is requested in its cache, then it will go through the process of getting the IP address of the website wanted and return it. It will then store this IP address in its memory for a period time. The period of time that it is stored in the memory is chosen by the owner of the domain by using a setting called time to live2, in which how long the website stays in the cache’s memory for.

## Iterative

The iterative DNS query will query a DNS server for information on a website, the user will enter a website they are trying to find and then the resolver may provide an answer from the cached memory or will ask another DNS server if they have the answer by returning the address of that DNS server, if it cannot return a definitive answer it will ask the next server. The query will start at the root server and then go down the levels according to its referrals that it receives.

## Root server

The root name server for the DNS records is a server that has a big role in the translating of a domain name into an IP address for a computer, they answer requests made in the root zone of the DNS. The servers can answer queries records which are cached in the root zone. They can also divert other requests made onto the TLD server.

## TLD (Top Level Domain) name server

The TLD name server is referring to the last part of a domain name, the part after the dot. TLDs made up into two categories, a country TLD and industries TLD. For example .FR for France and .GOV for governments websites. The TLD is the highest level of domain names in the root zone of the DNS for the internet. For the other domains in lower levels, it is the last section of the domain name 1. The managing of the TLD name servers is take care of by the IANA (Internet Assigned Numbers Authority) which is a part of the organization ICANN.

## Authoritative name servers

An authoritative name server provides answers to an original DNS query. It does not provide answers of websites from the cached memory 3. The Authoritative name server provides answers to the recursive DNS name servers with IP mapping of the website (or other server) searched for. The authoritative name server holds the records for domain names, these can be either A, CNAME, MX, NS, SOA or TXT records. An example could be if a DNS server in a network has stored a record for a website for example www.example.co.uk, then it would be that this DNS server is the authoritative server for the example.co.uk domain.

## URLs

A URL is the address of the website that you type into the address bar in a browser. A URL is unique and cannot be shared between different websites, it is just like a fingerprint with a person each person’s fingerprint is unique to them and there is no other the same. A URL is organized by the structure of different parts, this example shows the different parts of a URL and how they are organized – https://moodle.nptcgroup.ac.uk. The HTTPS is the protocol that is used to send data over the browser and the website you want to connect to.

The host is “Moodle” which is the name of the web servers where it can be accessed. Then the .ac is the second level domain, this is showing that it is an academic second level domain and is used by universities and colleges in the UK. Then the .uk is the top level domain, this is the highest level of domain and shows that it is in the country of the UK by its top level domain. This is the end of the URL and is normally either a company (.com), country (.fr) or a government (.gov) Domain name management includes keeping the DNS up to date, the management aspect of it is controlled using a portal by a third-party company.

Using this portal, the personal or business website can control DNS records which allow access the website, your blog, email and other online assets. Lots of companies online offer services that will control your domain names for you, these will keep your domain name from expiring by offering an auto renewal service in with their service. Once your domain name is up and running you can then use the portal to manage your domain and all aspects of it easily from one place, you can there change the DNS records quickly and easily for different domains.

# Part:-2

# Introduction

This section describes the basic protocol of communication protocol and the basic requirement of communication protocol and data formats for data exchange, address formats for data exchange, address mapping, rooting, detection of transmission errors, protocol design, design patterns for application protocol, formal specification, Explains protocol development and server hardware and what an operating system is. Also explains what a web server is hosting files, communication, HTTP, static vs dynamic content

# Communication protocol

A communication protocol is system of result that allow two or more entities of a communication system to transmit to information via any kind of variation of a physical quantity. The protocol defines the rules syntax semantics and synchronization of communication and possible error recovery methods. Protocols may be implemented by hardware software or a combination of both.

Communicating systems use well-defined formats for exchanging various messages. Each message has an exact meaning intended to elicit a response from a range of possible responses pre-determined for that particular situation. The specified behavior is typically independent of how it is to be implemented. Communication protocols have to be agreed upon by the parties involved. To reach an agreement, a protocol may be developed into a technical standard. A programming language describes the same for computations, so there is a close analogy between protocols and programming languages: protocols are to communication what programming languages are to computations. An alternate formulation states that protocols are to communication what algorithms are to computation.

Multiple protocols often describe different aspects of a single communication. A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack.

Internet communication protocols are published by the Internet Engineering Task Force (IETF). The IEEE (Institute of Electrical and Electronics Engineers) handles wired and wireless networking and the International Organization for Standardization (ISO) handles other types. The ITU-T handles telecommunication protocols and formats for the public switched telephone network (PSTN). As the PSTN and Internet converge, the standards are also being driven towards convergence.

# Basic requirements

Getting the data across a network is only part of the problem for a protocol. The data received has to be evaluated in the context of the progress of the conversation, so a protocol must include rules describing the context. These kind of rules are said to express the syntax of the communication. Other rules determine whether the data is meaningful for the context in which the exchange takes place. These kind of rules are said to express the semantics of the communication.

Messages are sent and received on communicating systems to establish communication. Protocols should therefore specify rules governing the transmission. In general, much of the following should be addressed.

## Data formats for data exchange

Digital message bit strings are exchanged. The bit strings are divided in fields and each field carries information relevant to the protocol. Conceptually the bit string is divided into two parts called the header and the payload. The actual message is carried in the payload. The header area contains the fields with relevance to the operation of the protocol. Bit strings longer than the maximum transmission unit (MTU) are divided in pieces of appropriate size.

## Address formats for data exchange

Addresses are used to identify both the sender and the intended receiver. The addresses are carried in the header area of the bit strings, allowing the receivers to determine whether the bit strings are of interest and should be processed or should be ignored. A connection between a sender and a receiver can be identified using an address pair (sender address, receiver address). Usually, some address values have special meanings. An all-1s address could be taken to mean an addressing of all stations on the network, so sending to this address would result in a broadcast on the local network. The rules describing the meanings of the address value are collectively called an addressing scheme.

## Address mapping

Sometimes protocols need to map addresses of one scheme on addresses of another scheme. For instance to translate a logical IP address specified by the application to an Ethernet MAC address. This is referred to as address mapping

## Routing

When systems are not directly connected, intermediary systems along the route to the intended receiver need to forward messages on behalf of the sender. On the Internet, the networks are connected using routers. The interconnection of networks through routers is called internetworking.

## Detection of transmission errors

Error detection is necessary on networks where data corruption is possible. In a common approach, a CRC of the data area is added to the end of packets, making it possible for the receiver to detect differences caused by corruption. The receiver rejects the packets on CRC differences and arranges somehow for retransmission.

## Acknowledgement

Acknowledgement of correct reception of packets is required for connection-oriented communication. Acknowledgments are sent from receivers back to their respective senders.

## Loss of information-timeouts and retries

Packets may be lost on the network or be delayed in transit. To cope with this, under some protocols, a sender may expect an acknowledgment of correct reception from the receiver within a certain amount of time. Thus, on timeouts, the sender may need to retransmit the information. In case of a permanently broken link, the retransmission has no effect so the number of retransmissions is limited. Exceeding the retry limit is considered an error.

## Direction of information flow

Direction needs to be addressed if transmissions can only occur in one direction at a time as on half-duplex links or from one sender at a time as on a shared medium. This is known as media access control. Arrangements have to be made to accommodate the case of collision or contention where two parties respectively simultaneously transmit or wish to transmit.

## Sequence control

If long bit strings are divided into pieces and then sent on the network individually, the pieces may get lost or delayed or, on some types of networks, take different routes to their destination. As a result, pieces may arrive out of sequence. Retransmissions can result in duplicate pieces. By marking the pieces with sequence information at the sender, the receiver can determine what was lost or duplicated, ask for necessary retransmissions and reassemble the original message.

## Flow control

Flow control is needed when the sender transmits faster than the receiver or intermediate network equipment can process the transmissions. Flow control can be implemented by messaging from receiver to sender.

## Queueing

Communicating processes or state machines employ queues (or "buffers"), usually FIFO queues, to deal with the messages in the order sent, and may sometimes have multiple queues with different prioritization.

# Protocol design

Systems engineering principles have been applied to create a set of common network protocol design principles. The design of complex protocols often involves decomposition into simpler, cooperating protocols. Such a set of cooperating protocols is sometimes called a protocol family or a protocol suite, within a conceptual framework.

Communicating systems operate concurrently. An important aspect of concurrent programming is the synchronization of software for receiving and transmitting messages of communication in proper sequencing. Concurrent programming has traditionally been a topic in operating systems theory texts. Formal verification seems indispensable because concurrent programs are notorious for the hidden and sophisticated bugs they contain. A mathematical approach to the study of concurrency and communication is referred to as communicating sequential processes (CSP). Concurrency can also be modeled using finite state machines, such as Mealy and Moore machines. Mealy and Moore machines are in use as design tools in digital electronics systems encountered in the form of hardware used in telecommunication or electronic devices in general.

The literature presents numerous analogies between computer communication and programming. In analogy, a transfer mechanism of a protocol is comparable to a central processing unit (CPU). The framework introduces rules that allow the programmer to design cooperating protocols independently of one another.

# Design patterns for application protocols

Commonly reoccurring problems in the design and implementation of communication protocols can be addressed by patterns from several different pattern languages: Pattern Language for Application-level Communication Protocols Service Design Patterns, Patterns of Enterprise Application Architecture, Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing. The first of these pattern languages focuses on the design of protocols and not their implementations. The others address issues in either both areas or just the latter.

# Formal specification

Formal methods of describing communication syntax are Abstract Syntax Notation One (an ISO standard) and Augmented Backus-Naur form (an IETF standard).

Finite state machine models and communicating finite-state machines are used to formally describe the possible interactions of the protocol.

# Protocol development

For communication to occur, protocols have to be selected. The rules can be expressed by algorithms and data structures. Hardware and operating system independence is enhanced by expressing the algorithms in a portable programming language. Source independence of the specification provides wider interoperability.

Protocol standards are commonly created by obtaining the approval or support of a standards organization, which initiates the standardization process. This activity is referred to as protocol development. The members of the standards organization agree to adhere to the work result on a voluntary basis. Often the members are in control of large market-shares relevant to the protocol and in many cases, standards are enforced by law or the government because they are thought to serve an important public interest, so getting approval can be very important for the protocol.

# What is server hardware ?

server could be any computer if it serve any services that mines it's server, but server definitions will be different in every Level it's depend on your requirement, if services need to access with 2 or 3 computer or services execution on server is less not needed much RAM and CPU , it's simple for big task you need more resources, more man , for small task we can do our-self Accordingly we categorize server hardware, below mention server given according to our need and it's in sequence, most powerful server is mainframes server which is used by Bank, where the execution is very high "Rack Servers", "Blade Servers" , "Tower Servers" , "Mainframes".

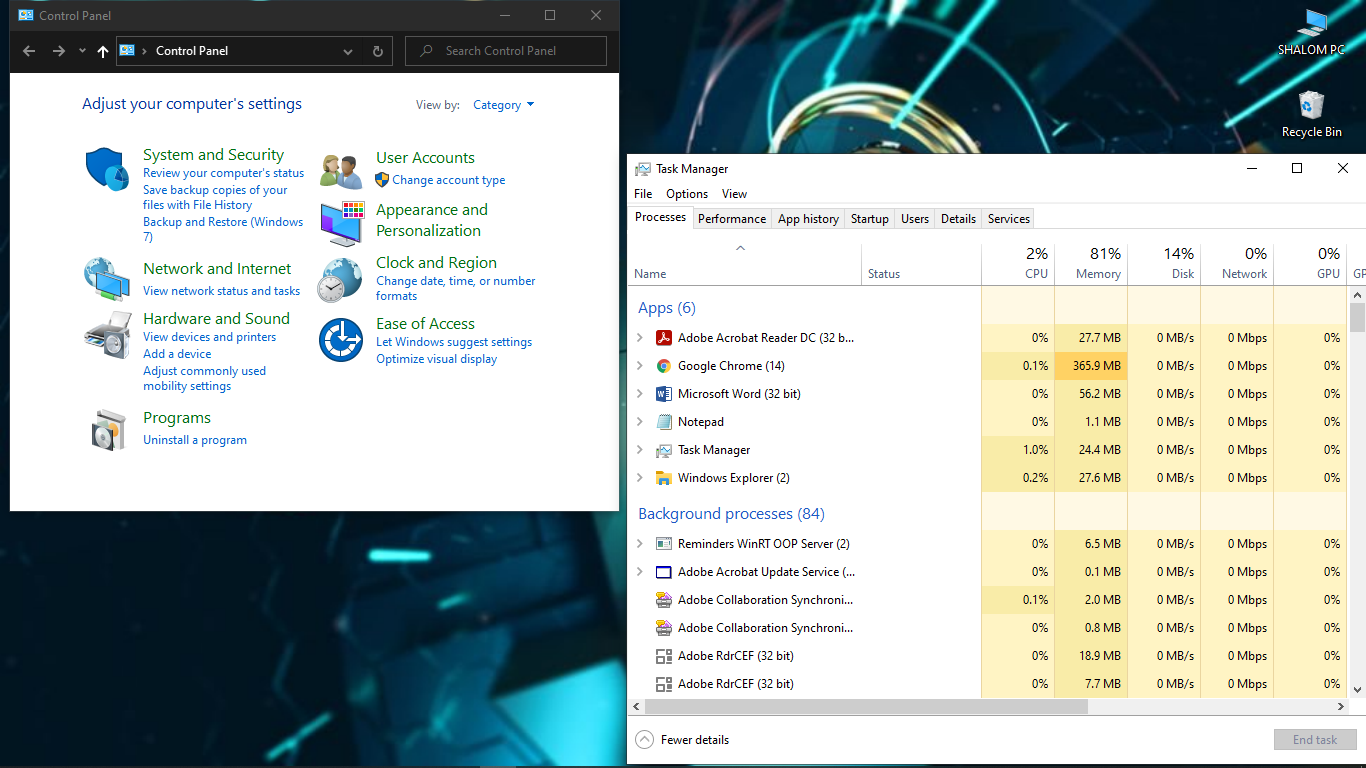
# What is an operating system ?

An operating system is the primary software that manages all the hardware and other software on a computer. The operating system, also known as an “OS,” interfaces with the computer’s hardware and provides services that applications can use.

## What does an operating system do ?

An operating system is the core set of software on a device that keeps everything together. Operating systems communicate with the device’s hardware. They handle everything from your keyboard and mice to the Wi-Fi radio, storage devices, and display. In other words, an operating system handles input and output devices. Operating systems use device drivers written by hardware creators to communicate with their devices.

Operating systems also include a lot of software—things like common system services, libraries, and application programming interfaces (APIs) that developers can use to write programs that run on the operating system.

The operating system sits in between the applications you run and the hardware, using the hardware drivers as the interface between the two. For example, when an application wants to print something, it hands that task off to the operating system. The operating system sends the instructions to the printer, using the printer’s drivers to send the correct signals. The application that’s printing doesn’t have to care about what printer you have or understand how it works. The OS handles the details.

The OS also handles multi-tasking, allocating hardware resources among multiple running programs. The operating system controls which processes run, and it allocates them between different CPUs if you have a computer with multiple CPUs or cores, letting multiple processes run in parallel. It also manages the system’s internal memory, allocating memory between running applications.

The operating system is the one big piece of software running the show, and it’s in charge of everything else. For example, the operating system also controls the files and other resources these programs can access.

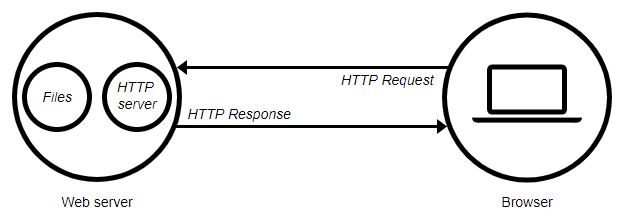
Most software applications are written for operating systems, which lets the operating system do a lot of the heavy lifting. For example, when you run Minecraft, you run it on an operating system. Minecraft doesn’t have to know exactly how each different hardware component works. Minecraft uses a variety of operating system functions, and the operating system translates those into low-level hardware instructions. This saves the developers of Minecraft—and every other program that runs on an operating system—a lot of trouble.

# What is a web server ?

The term web server can refer to hardware or software, or both of them working together. On the hardware side, a web server is a computer that stores web server software and a website's component files. (for example, HTML documents, images, CSS stylesheets, and JavaScript files) A web server connects to the Internet and supports physical data interchange with other devices connected to the web.

On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an HTTP server. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct (hardware) web server, the (software) HTTP server accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. (If the server doesn't find the requested document, it returns a 404 response instead.)



To publish a website, you need either a static or a dynamic web server. A static web server, or stack, consists of a computer (hardware) with an HTTP server (software). We call it "static" because the server sends its hosted files as-is to your browser.

A dynamic web server consists of a static web server plus extra software, most commonly an application server and a database. We call it "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server.

For example, to produce the final webpages you see in the browser, the application server might fill an HTML template with content from a database. Sites like MDN or Wikipedia have thousands of webpages. Typically, these kinds of sites are composed of only a few HTML templates and a giant database, rather than thousands of static HTML documents. This setup makes it easier to maintain and deliver the content.

## Hosting files

First, a web server has to store the website's files, namely all HTML documents and their related assets, including images, CSS stylesheets, JavaScript files, fonts, and video.

Technically, you could host all those files on your own computer, but it's far more convenient to store files all on a dedicated web server because:

* A dedicated web server is typically more available. (up and running)
* Excusing downtime and systems troubles, a dedicated web server is always connected to the Internet.
* A dedicated web server can have the same IP address all the time. This is known as a dedicated IP address. (not all ISPs provide a fixed IP address for home lines)
* A dedicated web server is typically maintained by a third-party.

For all these reasons, finding a good hosting provider is a key part of building your website. Examine the various services companies offer. Choose one that fits your needs and budget. (Services range from free to thousands of dollars per month.) You can find more details in this report.

Once you have web hosting service, you must upload your files to your web server.

## Communicating through HTTP

Second, a web server provides support for HTTP (Hypertext Transfer Protocol). As its name implies, HTTP specifies how to transfer hypertext (linked web documents) between two computers.

A Protocol is a set of rules for communication between two computers. HTTP is a textual, stateless protocol.

**Textual**

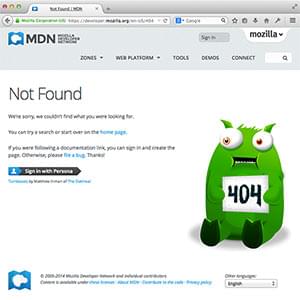
All commands are plain-text and human-readable.

**Stateless**

Neither the server nor the client remember previous communications. For example, relying on HTTP alone, a server can't remember a password you typed or remember your progress on an incomplete transaction. You need an application server for tasks like that. (We'll cover that sort of technology in other articles.)

HTTP provides clear rules for how a client and server communicate. We'll cover HTTP itself in a technical article later. For now, just be aware of these things:

* Only clients can make HTTP requests, and then only to servers. Servers can only respond to a client's HTTP request.
* When requesting a file via HTTP, clients must provide the file's URL.
* The web server must answer every HTTP request, at least with an error message.

On a web server, the HTTP server is responsible for processing and answering incoming requests.

1. Upon receiving a request, an HTTP server first checks if the requested URL matches an existing file.
2. If so, the web server sends the file content back to the browser. If not, an application server builds the necessary file.
3. If neither process is possible, the web server returns an error message to the browser, most commonly 404 Not Found. (The 404 error is so common that some web designers devote considerable time and effort to designing 404 error pages.)

## Static vs. dynamic content

Roughly speaking, a server can serve either static or dynamic content. Remember that the term static means "served as-is". Static websites are the easiest to set up, so we suggest you make your first site a static site.

The term dynamic means that the server processes the content or even generates it on the fly from a database. This approach provides more flexibility, but the technical stack is more complex, making it dramatically more challenging to build a website.

There are so many application server technologies that it's difficult to suggest a particular one. Some application servers cater to specific website categories like blogs, wikis, or ecommerce; others are more generic. If you're building a dynamic website, take the time to choose technology that fits your needs. Unless you want to learn web server programming (which is an exciting area in itself!), you don't need to create your own application server.

# Task:-3

# Introduction

‘Web developer‘ is one of those jobs that most people don’t actually understand at all. One reason for that is because the term is such a broad one. ‘Web development’ can entail anything from web design or web content development to network security configuration or even client/server-side scripting. Confused yet? Basically, web developers are the people who build websites, in every sense of the word. But the job can be loosely broken down into two categories; front end web developers and back end web developers. What’s the difference? We’ll tell you. But first…

# How front-end and back-end relate to presentation and application layer

Front-End and Back-Ends refers to the separation of concerns between the presentation layer, application layer and database layer as well as in front-tend there is application and in back-end there are two layers i.e. application and database layer. The three different layers of front-end and back-end are described below.

* **Presentation Layer:** Presentation layer is known as front-end layer that consists of user interface and user interface is graphical that can be accessible through web browser and web application which display content and information useful to an end user. Similarly, this application layer is built by using web technologies such as HTML, CSS and JavaScript and can be also built by using frameworks and communicate with other layers by using API calls.
* **Application layer:** Application layer contains the functional business logic that drives the core capabilities of an application which is mostly written in Java, .NET, C #, Python, C++, etc.

Presentation layer is mostly deployed to the desktop, tablets and phones either through web browser or through web-based application utilizing a web server as well as application layer is mostly hosted in cloud or in dedicated workstation which depend upon the complexity and processing power required by the application. There are the different benefits of using three-layer architecture such as speed of development, scalability, performance and availability as well as it also helps to improve development efficiency by allowing team to focus on their core competencies.

# Front-End Technologies

Front- End technology is a part of the website that involves the creation of a website so that a user can directly see and interact with it, as well as the structure, design, behavior and animation of everything that we see on the screen when we open websites, web applications or mobile apps. In addition, there are different technologies of front-end which are described below and I will also describe some of the frameworks like HTML, CSS, JavaScript.

## HTML

HTML (or Hypertext Markup Language) is a computer language designed to create websites that later can be explored by anybody who accesses the Internet. HTML is normally employed to structure a web document. It defines such elements as headlines or paragraphs and enables embedding images, video, and other media.

**How HTML works.** HTML is composed of a series of short codes called tags, normalized into a text-file by a site creator. The text is then stored as an HTML file and observed through a browser. The browser scans the file and interprets the text into a visible form, and in the best case, renders the page as the designer had planned.

* Hyper Text is the way by which we travel across the web by clicking hyperlinks – specific texts taking you to other pages. Hyper means it’s non-linear, which allows for moving to any other place, as there is no predefined order to do so.
* Markup determines the qualities that HTML tags apply to the text inside them. Tags mark it as a particular type of text.
* As a Language, it holds code words and syntax like any other language.

For example:

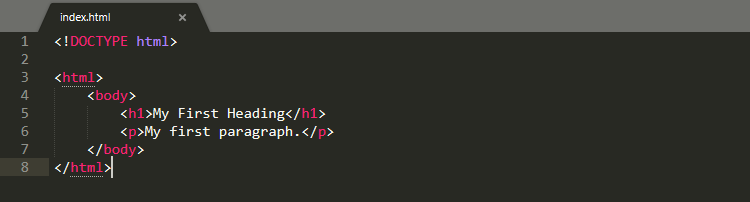


Figure 1 The example of HTML code

## CSS

CSS (or Cascading Style Sheets) is a style sheet language. It’s applied to define how HTML elements are supposed to be presented on a webpage in terms of design, layout, and variations for diverse devices with different screen sizes. CSS masters the layout of numerous different web pages at a time.

**How it works.** CSS interacts with HTML elements, the components of a web page.

* To communicate with HTML, CSS uses **selectors**. A selector is the part of CSS code defining which HTML piece the CSS styling will impact.
* A **declaration** contains properties and values that are employed by the selector.
* **Properties** define font size, color, and margins. Values are the settings for these properties.

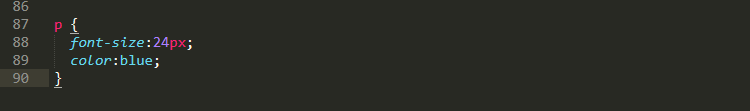


Figure 2 The example of CSS code

P (for paragraph) is selector, {  font-size:24px;  color:blue;  } is a declaration, font-size: and color: are properties, while 24px; and blue; are values.

CSS is written in plain text over a text editor or word processor on a computer. If you were to examine how the CSS code is implemented to HTML content, there are three ways to do so:

* External style sheets are stored as .css files and can be applied to define the appearance of a whole website via a single file, instead of putting extra instances of CSS code to every HTML element which has to be modified. To use an external style sheet, .html files need to contain a header section that connects to the external style sheet.
* Internal style sheets are CSS directions put straight into the header of a particular .html page.
* Inline styles are snippets of CSS recorded into HTML code itself.

**CSS frameworks.**A CSS framework is a set of default CSS and HTML files. It extends a front-end developer’s capabilities for website design. In addition to assisting while building a responsive design, CSS frameworks also present distinct and symmetric layouts, saving developers from writing code from ground zero at every occasion. They are usually considered a good choice to fit diverse platforms and screen sizes. With common user interface components, grid systems, layouts, and many other features, CSS frameworks considerably accelerate development workflow. Many frameworks exist in the CSS Universe:

* Full featured (Bootstrap, Foundation, Semantic UI, and more),
* Aimed at Material Design: (Materialize and Material Design Lite), and
* Lightweight (Pure).

## JavaScript

JavaScript (JS) is one of the most popular scripting languages. It’s mostly renowned for providing a full stack of technologies for both front-end and back-end development. As we are talking about the first one, it’s applied to make web pages dynamic.

**How JavaScript works.** JS boosts a website’s overall interactivity. It enables modeling animated UI components such as image sliders, pop-ups, extensive site navigation menus, and more. JavaScript provides a website with extended functionality that isn’t otherwise achievable with HTML and CSS alone. JavaScript allows web pages to reply to users’ actions and dynamically refresh themselves. Thanks to JavaScript, this process doesn’t demand a page reload to alter its representation.

Frameworks are templates to create a website or a web application. They provide a structure (such as a skeleton or scaffolding) on which to arrange the whole project. While the framework sets the page templates, they build the structure with particular allocated areas to embed a framework code in.

So, JavaScript frameworks are complete sets of tools to form and arrange a website or web application.

# Back-End Technologies

In general terms, a website or dynamic web application is a number of files containing structure, design, content, and functionality. The technologies and codes that enliven a site, the area of web development which makes the site run invisibly to users is called backend. It consists of the server, the database, and the server-side applications. Back-end is the far side of the web page or screen functionality; to set the analogy, it could be imagined as the brain or the engine of a website or app.

For example, when users navigate their web browser to https://tubikstudio.com/, they actually turn to the server where Tubik Studio website files are located and send a request to see them. It should be also noted that loading a website, users aren’t looking at that website directly on the server. The server sends the files to the web browser, which then looks at them locally, on user’s computer. In fact, backend sets the basis on which front-end development is able to come into play. Backend developer programs core computational logic of the system including software, website and information.

## Server-side language

**PHP.** It is an HTML-embedded scripting language. The main aim of the language is to allow creating dynamically generated pages quickly. PHP is easy to install and deploy, it is staying competitive with lots of modern frameworks, and is the foundation for a number of content-management systems (CMS). PHP powers famous web sites and platforms such as WordPress, Wikipedia, Facebook etc.

**Python.** It is an interpreted, object-oriented, high-level programming language with dynamic semantics. This programming language is fast, making it effective for getting things to market quickly. Python is simple; its syntax is easy-to-learn, readable and therefore reduces the cost of program maintenance. It is the oldest of the scripting languages and often referred to as powerful and effective. Python powers famous sites such as YouTube, Google, The Washington Post etc.

**Java.** It is a programming language designed for use in the distributed environment of the Web. Java is similar to C++ language, but it is simpler to use than the latter and enforces an object-oriented programming model. It can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network. Java powers famous sites such as Twitter, Verizon, Salesforce.

**Ruby.** It is dynamic, open source programming language with a focus on simplicity and productivity. Great for building complicated logic on the database side of a website, Ruby connects the back-end and database functionality that PHP and SQL can offer. It is characterized as the language of easy maintenance and high-traffic demands. Ruby powers famous sites such as Hulu, the original version of Twitter, Living Social, Basecamp etc. Moreover, the framework Ruby on Rails for this programming language has extensive libraries which support a higher level of flexibility for a developer in the process of creating code for particular aims.

**C#.** It is type-safe, object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. C# can be used to create Windows client applications, XML Web services, distribute components, client-server applications, database applications, etc. It also can be used to build iOS and Android mobile apps with the help of a cross-platform technology like Xamarin.

**С++.** It is an enhanced C language typically used for object-oriented programming. C++ is effective for complex applications also built on the .NET Framework.

**Erlang.** It is a general-purpose programming language. Erlang has built-in support which increases its level of competitiveness and distribution. It is used in several large telecommunication systems from Ericsson.

## Server-side frameworks

**ASP.NET.** It is an open source web framework for building modern web apps and services with .NET. It creates websites based on HTML, CSS, and JavaScript that are simple, fast, and can scale to millions of users.

**Ruby on Rails.** It is an open source Ruby language framework for developing database-backed web applications. Ruby on Rails is also full-stack framework which includes everything needed to create a database-driven web application, using the Model-View-Controller pattern.

**Django.** It is a high-level Python language web framework that encourages rapid development and clean, pragmatic design. Django is also free and open source.

**Node.js.** It is a JavaScript runtime built on Chrome V8 JavaScript engine. Here are some examples of Node.js frameworks:

* Hapi.js
* Socket.io
* Express.js
* Mojito
* Meteor
* Derby
* Mean.js
* Sails.js
* Koa.js
* Total.js

# Part:-4

# Introduction

At the most basic level, the user interface (UI) is the series of screens, pages, and visual elements—like buttons and icons—that enable a person to interact with a product or service.

User experience (UX), on the other hand, is the internal experience that a person has as they interact with every aspect of a company’s products and services.

It’s common for folks to use these terms interchangeably, or sometimes incorrectly. If you’ve ever wondered, “What is UI, what is UX, and what’s the difference between them?” in today’s post we’ll dig a bit deeper into UI and UX to get a better understanding of the differences between them.

# Website Creation Tools

There are several reasons behind the popularity of website creation tools. These tools allow you to develop a website, even if you do not know how to code. These are generally offered by website hosting companies as proprietary tools. These tools include an editor, several themes and easy drag and drop various elements like images, text, headers, etc. to build a site.

You do not need any special skill to build a website using such website creation tools. While such ready-made templates are easy and quick to use, and offer built-in features for a fair amount of customization, these sites become heavy and are often slow to load.

This in turn can affect the SEO performance of the site. Also, most of these tools are flash-based, making them difficult to read by Google. Also, these tools are DIY and hence do not offer support to maintain your website.

While these tools are cost and time effective, they fall short in certain aspects. Here we have compiled a list of some advantages that custom built websites offer when compared to online website creation tools.

## Unique Customization of Website

Custom websites are designed keeping in mind your objectives, needs and goals. They are developed offline using codes and are consequently made online. While customizing the website, aspects such as user interface and user experience are looked into thoroughly to increase effectiveness.

Also, your business can be highlighted distinctively and uniquely from others. In contrast, online website creation tools are limited in terms of templates and layout. Using them may result in a similar looking website to that of your competitors.

## Better SEO

The primary purpose to take business online is to boost its visibility to people. For this purpose, SEO (Search Engine Optimization) helps your page rank higher on Google or other search engines. Being ranked (or listed) higher in the search results leads to increase in traffic on your website.

To make your website optimized, website designers use planning, keyword research, sitemap creation and a variety of other tools. On the contrary, online website builders have limited SEO tools which may deprive you off these benefits.

## Extant Support by Professionals

While using website creation you may come across terminologies like DNS(Domain Name System), TLS(Transport Layer Security) etc. These can be confusing at times.

Hiring a website designer or a custom PHP development company can relieve you of this confusion as all these technicalities are looked after by them. This also ensures time saving.

## Better Page Loading Speed

Once a visitor is on your website, you need to ensure that he stays on it. For this, page speed plays an important role. Also, Google considers your page speed while ranking.

In addition to providing relevant content through best SEO strategies, custom websites offer better Page speed. Website designers help in decreasing loading time through coding

They address specifics like JavaScript and CSS (Cascading Style Sheets), Browser caching, CDNs (Content Delivery Networks), Content compression etc. for providing best loading speed. Building through online tools does not let you address these specifics in detail. Using builders may result in reduced loading speed on your website.

## Ease in E-commerce Tools

Most of the small businesses require tools such as CRM(Customer Relationship Manager), contact management tools, appointment schedulers etc. While online builders allow these e-commerce features, they rely on third party applications for the purpose.

Without knowledge, scrutinizing these applications can be overwhelming. Also, installing these third party applications requires them to be updated regularly. A custom ecommerce website design can save you from such strenuous efforts.

## Own Your Website

Creating your website through online creation tools does not give you its ownership. Instead, the ownership remains with the online building platform since you use their domain name. Whereas in a custom website, you are required to purchase a domain name and creating designs from scratch which gives you the ownership of your website.

For example, suppose you want to create a website named XYZ. Creating it through Wix will give you a name like “XYZ.Wix.com” whereas if you purchase your own domain name, you can own your website called XYZ.com. Also, you can upload your website easily on a new server in case you decide to switch your web hosting company.

## Limited Terms and Conditions

Using online creators requires you to comply with their specific terms and conditions. Some of them are-

* Requiring you to submit your information.
* Reserving the right to display advertisements.
* Termination of website on specific grounds etc.

While these conditions differ from company to company, a custom website does not require such compliances. All you need to ensure is that your website does not violate general guidelines as any inappropriate content may result in the customer reporting your website as inappropriate on Google or other search platforms.

# What is UI ?

Simply put, user interface (UI) is anything a user may interact with to use a digital product or service. This includes everything from screens and touchscreens, keyboards, sounds, and even lights. To understand the evolution of UI, however, it’s helpful to learn a bit more about its history and how it has evolved into best practices and a profession.

## A brief history of the user interface

Back in the 1970’s, if you wanted to use a computer, you had to use the command line interface. The graphical interfaces used today didn’t yet exist commercially. For a computer to work, users needed to communicate via programming language, requiring seemingly infinite lines of code to complete a simple task.

By the 1980’s the first graphical user interface (GUI) was developed by computer scientists at Xerox PARC. With this groundbreaking innovation, users could now interact with their personal computers by visually submitting commands through icons, buttons, menus, and checkboxes.

Figure 3 Old computer

his shift in technology meant that anyone could use a computer, no coding required, and the personal computer revolution began.

By 1984 Apple Computer released the Macintosh personal computer which included a point and click mouse. The Macintosh was the first commercially successful home computer to use this type of interface.

The accessibility and prevalence of personal—and office—computers meant that interfaces needed to be designed with users in mind. If users couldn’t interact with their computers, they wouldn’t sell. As a result, the UI designer was born.

As with any growing technology, the UI designer’s role has evolved as systems, preferences, expectations, and accessibility has demanded more and more from devices. Now UI designers work not just on computer interfaces, but mobile phones, augmented and virtual reality, and even “invisible” or screen less interfaces (also referred to as zero UI) like voice, gesture, and light.

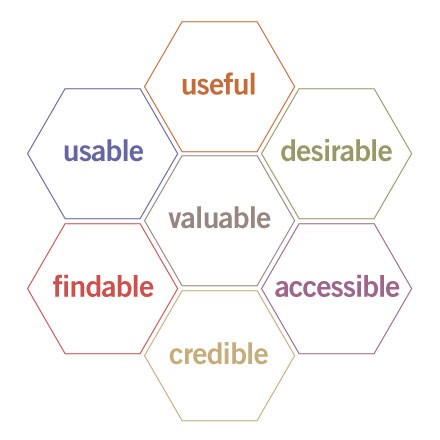
Today’s UI designer has nearly limitless opportunities to work on websites, mobile apps, wearable technology, and smart home devices, just to name a few. As long as computers continue to be a part of daily life, there will be the need to make the interfaces that enable users of all ages, backgrounds, and technical experience can effectively use.

# What is UX ?

User experience, or UX, evolved as a result of the improvements to UI. Once there was something for users to interact with, their experience, whether positive, negative, or neutral, changed how users felt about those interactions.

Cognitive scientist Don Norman is credited with coining the term, “user experience” back in the early 1990’s when he worked at Apple and defines it as follows,

Figure 4 User Experience Rocks

That’s a broad definition that could encompass every possible interaction a person could have with a product or service—not just a digital experience. Some UX professionals have opted for calling the field customer experience, and others have gone a step further to simply refer to the field as experience design.

No matter what it’s called, Norman’s original definition of UX is at the core of every thought experience design—it’s all-encompassing and always centered around the human being it's interacting with.

To understand what makes an experience a good one, Peter Moreville developed a great visual to highlight what goes into effective UX design.

Figure 5 Peter Moreville

This ‘usability honeycomb’ has become the foundation for best practices for UX professionals to help guide their efforts across multiple touchpoints with the user, including:

* How they would discover your company’s product
* The sequence of actions they take as they interact with the interface
* The thoughts and feelings that arise as they try to accomplish their task
* The impressions they take away from the interaction as a whole

UX designers are responsible for ensuring that the company delivers a product or service that meets the needs of the customer and allows them to seamlessly achieve their desired outcome.

UX designers work closely with UI designers, UX researchers, marketers, and product teams to understand their users through research and experimentation. They use the insights gained to continually iterate and improve experiences, based on both quantitative and qualitative user research.

# What’s the difference between UI & UX ?

At the most basic level, UI is made up of all the elements that enable someone to interact with a product or service. UX, on the other hand, is what the individual interacting with that product or service takes away from the entire experience.

Don Norman and Jakob Nielsen summed it up nicely when they said:

It’s important to distinguish the total user experience from the user interface (UI), even though the UI is obviously an extremely important part of the design. As an example, consider a website with movie reviews. Even if the UI for finding a film is perfect, the UX will be poor for a user who wants information about a small independent release if the underlying database only contains movies from the major studios.

Take Google, for example. Its famously spartan interface highlights how a great experience doesn’t require bells and whistles. By focusing on the user, Google knows that when they come to the site, they’re after one thing: information. And they want it quickly.

The fact that ‘google’ is a widely accepted verb shows how well the company delivers on that experience—and expectation. Just about anything a person has ever wanted to know can be accessed in the blink of an eye and few other search engines survive today.

Now imagine that every time you searched on Google, it took 15 seconds to get a result—you’d no longer be able to instantly get an answer to your question. Even if the interface stayed the same, your experience with Google would be dramatically different.

# Part:-5

# Introduction

Wire framing is a practice used by UX designers which allows them to define and plan the information hierarchy of their design for a website, app, or product. This process focuses on how the designer or client wants the user to process information on a site, based on the user research already performed by the UX design team.

When designing for the screen you need to know where all the information is going to go in plain black and white diagrams before building anything with code. Wire framing is also a great way of getting to know how a user interacts with your interface, through the positioning of buttons and menus on the diagrams.

Without the distractions of colors, typeface choices or text, wire framing lets you plan the layout and interaction of your interface. A commonly-used argument for wire framing is that if a user doesn’t know where to go on a plain hand-drawn diagram of your site page, then it is irrelevant what colors or fancy text eventually get used. A button or call to action needs to be clear to the user even it’s not brightly colored and flashing.

# Site map

A site map is a model of a website's content designed to help both users and search engines navigate the site. A site map can be a hierarchical list of pages (with links) organized by topic, an organization chart, or an XML document that provides instructions to search engine crawl bots.

Site map may also be spelled sitemap.

# Wireframe

Wire framing is a way to design a website service at the structural level. A wireframe is commonly used to lay out content and functionality on a page which takes into account user needs and user journeys. Wireframes are used early in the development process to establish the basic structure of a page before visual design and content is added.

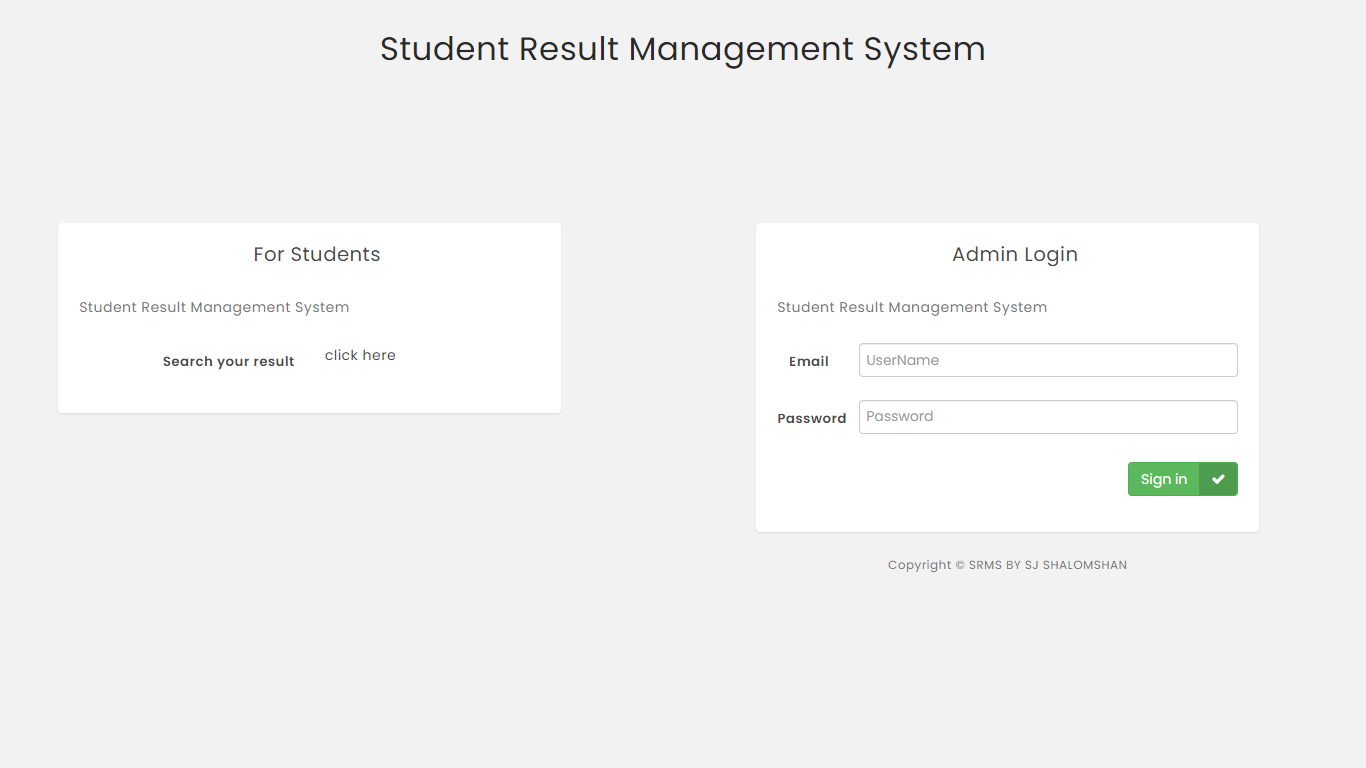
On the below, this is wireframe I made for my website, here wireframe is basically to know about our website:

Figure 6 Sign in page

The main purpose of this section is to enable a student to view the points he / she has earned using his / her Role ID and his / her class.

In addition, the owner of this web page, ie the author can use his username and password to go to the main part of this web side.

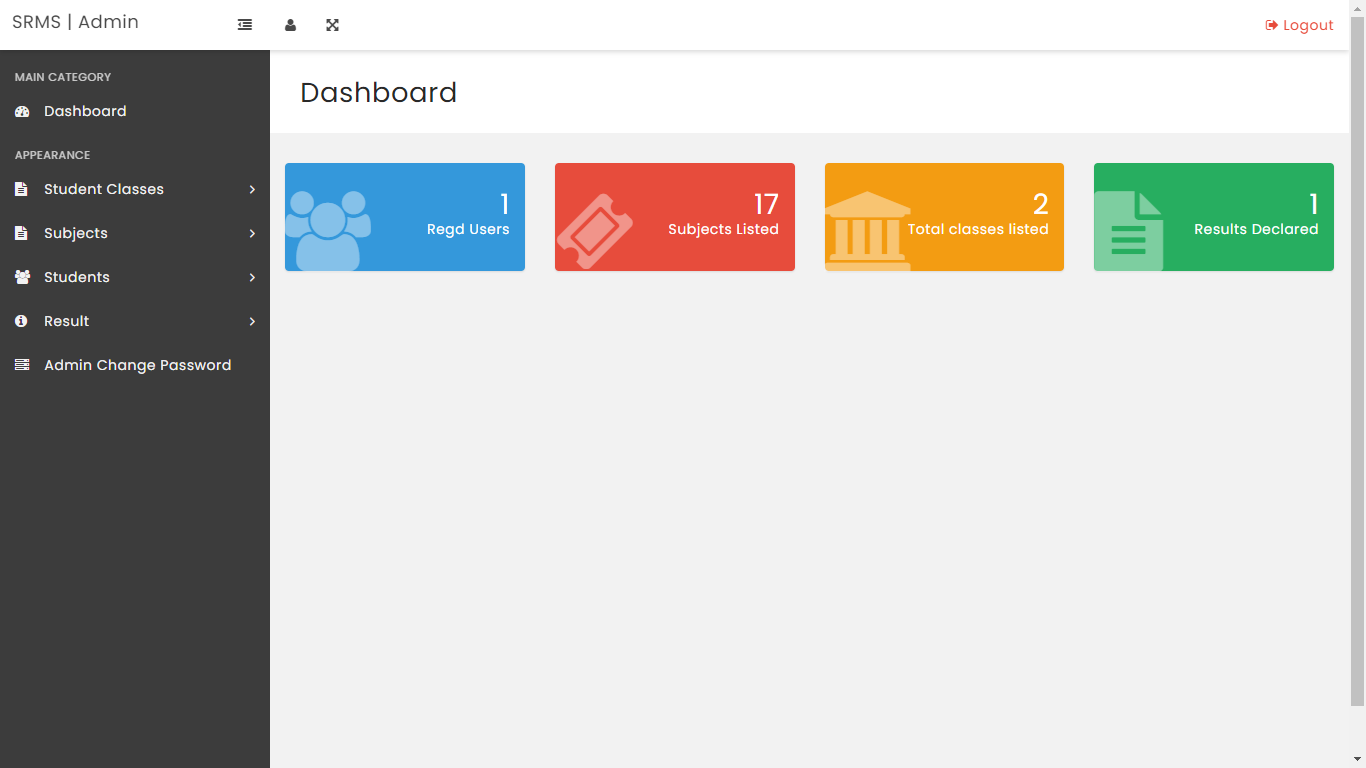


Figure 7 The main page

This section is the main part of this web side which can perform activities such as entering a student's name and details, entering subjects, entering class grades and entering a student's points.

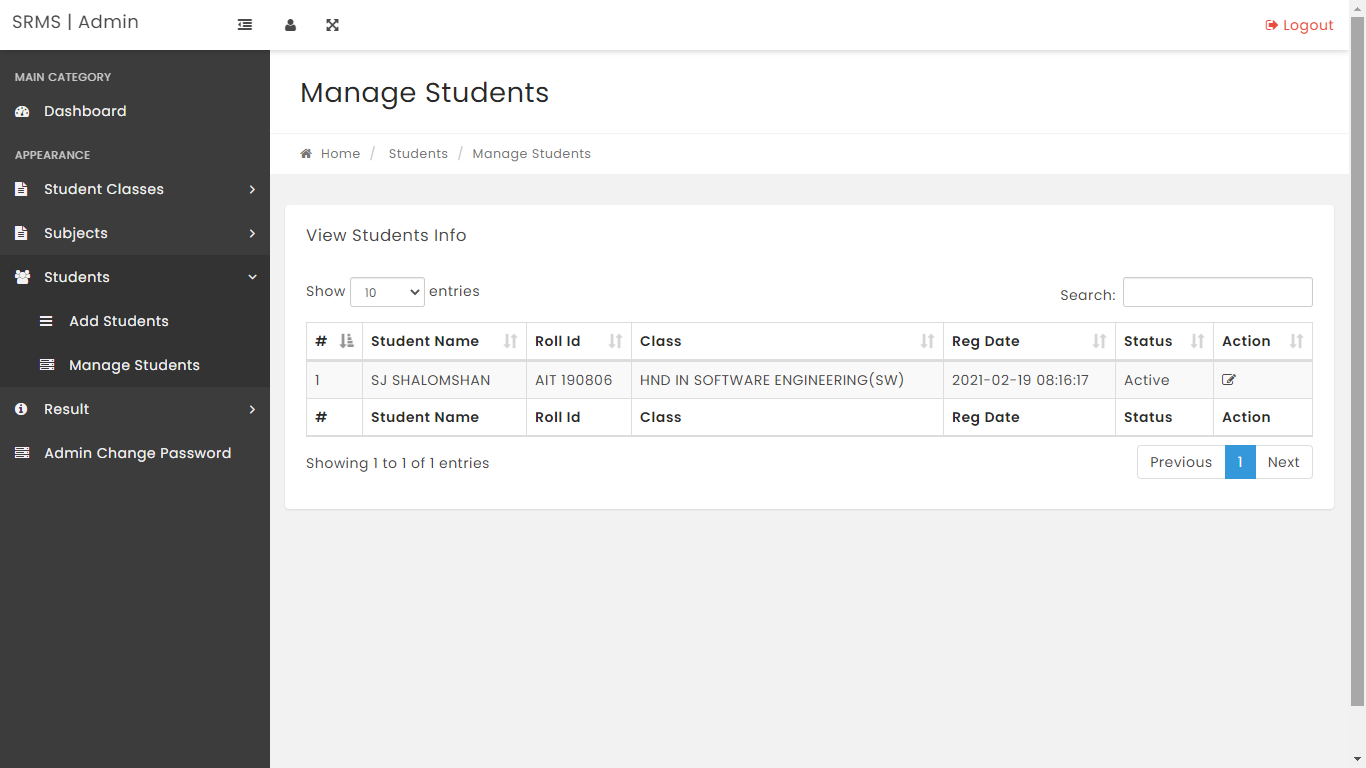


Figure 8 Manage students

The purpose of this section is to manage the student who has already entered and to enroll the new student.

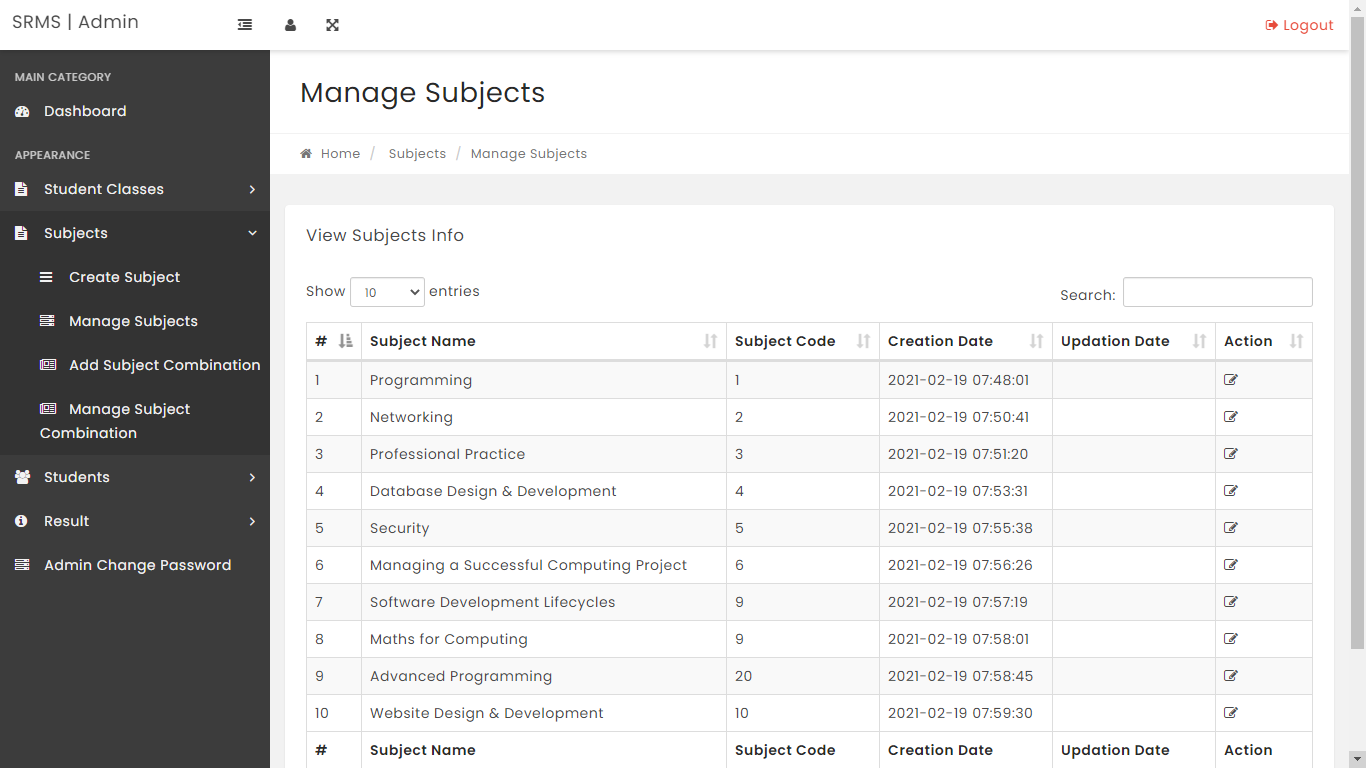


Figure 9 Manage subjects

This is to create subject, manage subject, add subject combination and manage subject.

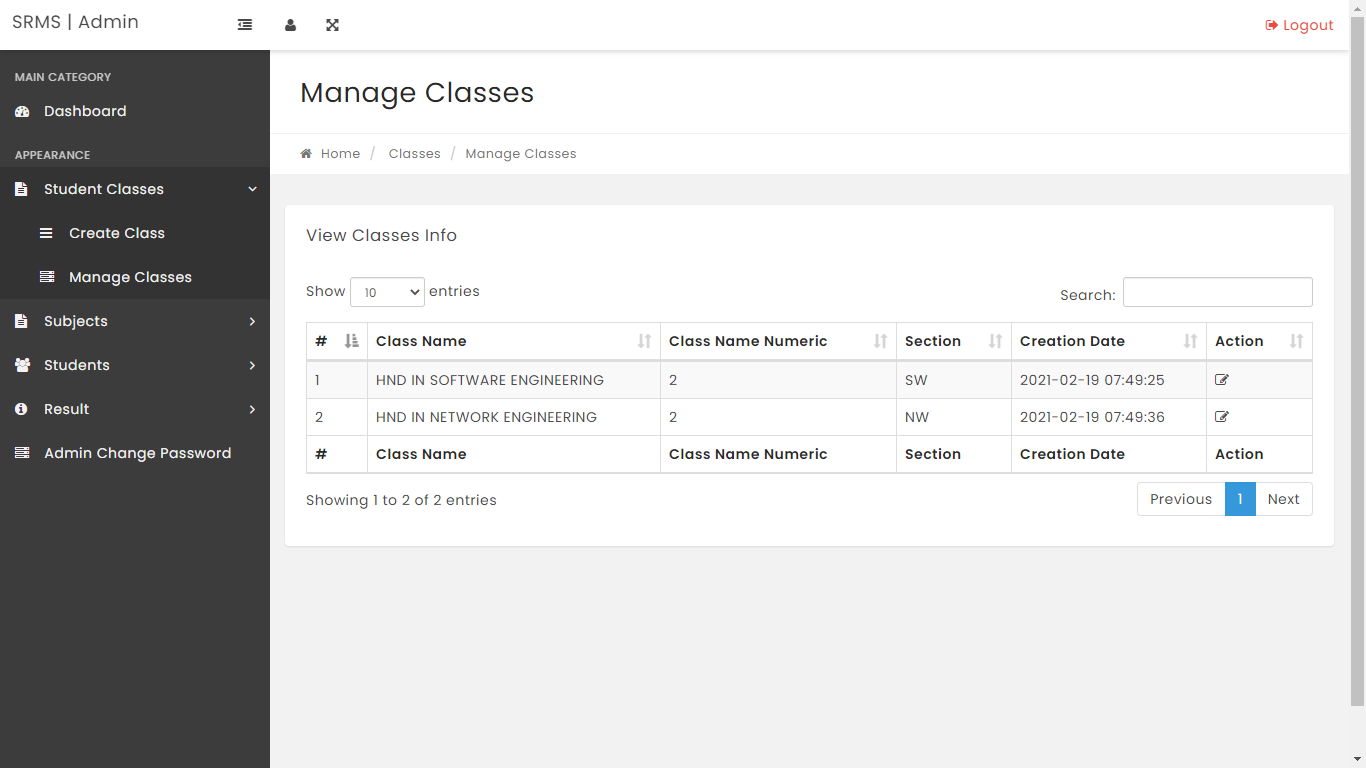


Figure 10 Manage classes

This section is about entering a student's class and managing the student's class profile.

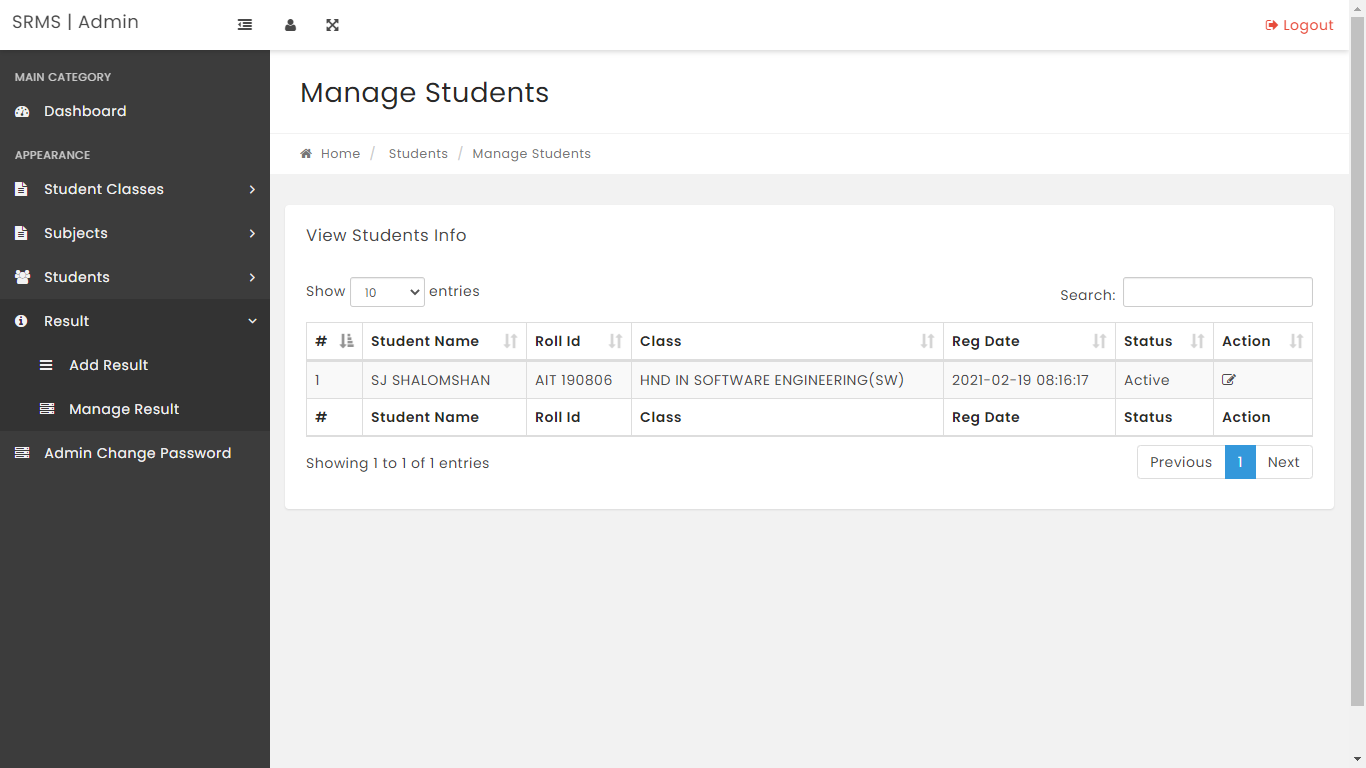


Figure 11 Result Management

This section is about entering and managing a student's marks.

# A full set of client and user requirements

* Student registration
  + Student management
* Subject add listed
  + Manage subject
* Add Subject Combination
  + Manage subject combination
* Add classes
  + Manage classes
* Result declared
  + Manage results

# Part:-6

# Entire function of the system

In this section, the entire function of the Student Result Management System is going to explain from scratch. For the explanation purpose, the screenshots of the web side and some important code snippets will be used to demonstrate the function of the system.

This section will give a clear statement of the modules of the Student Result Management System with the perfect explanation. This Student Result Management System document has some more details information about the web side.

## Essential Function

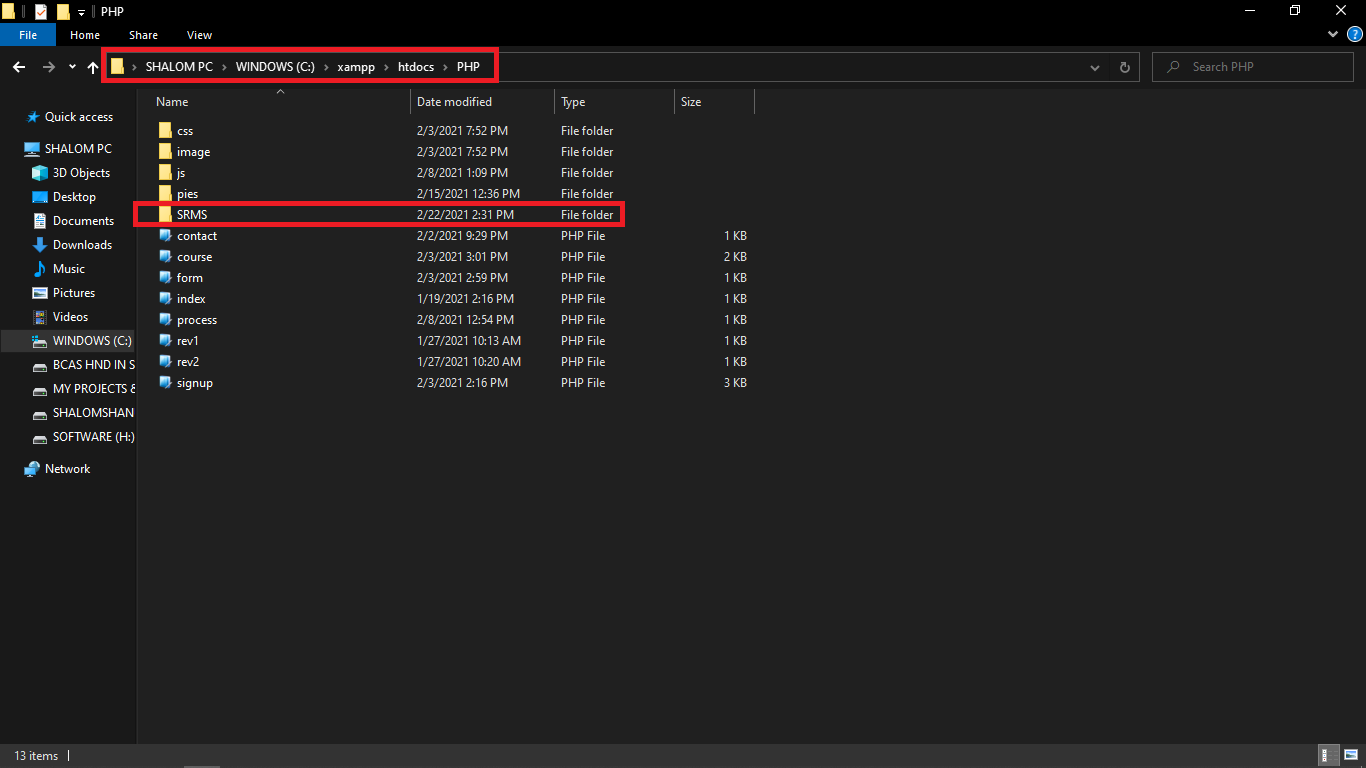
* Students management
* Subject management
* Classes management
* Result management

These are main functions that occurs inside the student result management system. beyond this there are several other types of functions are occurred inside the system. This structure of this application is so different. Because this application uses MySQL as the main component of the database. MySQL is used to store data about the student information or other classes details student result management system configurations.

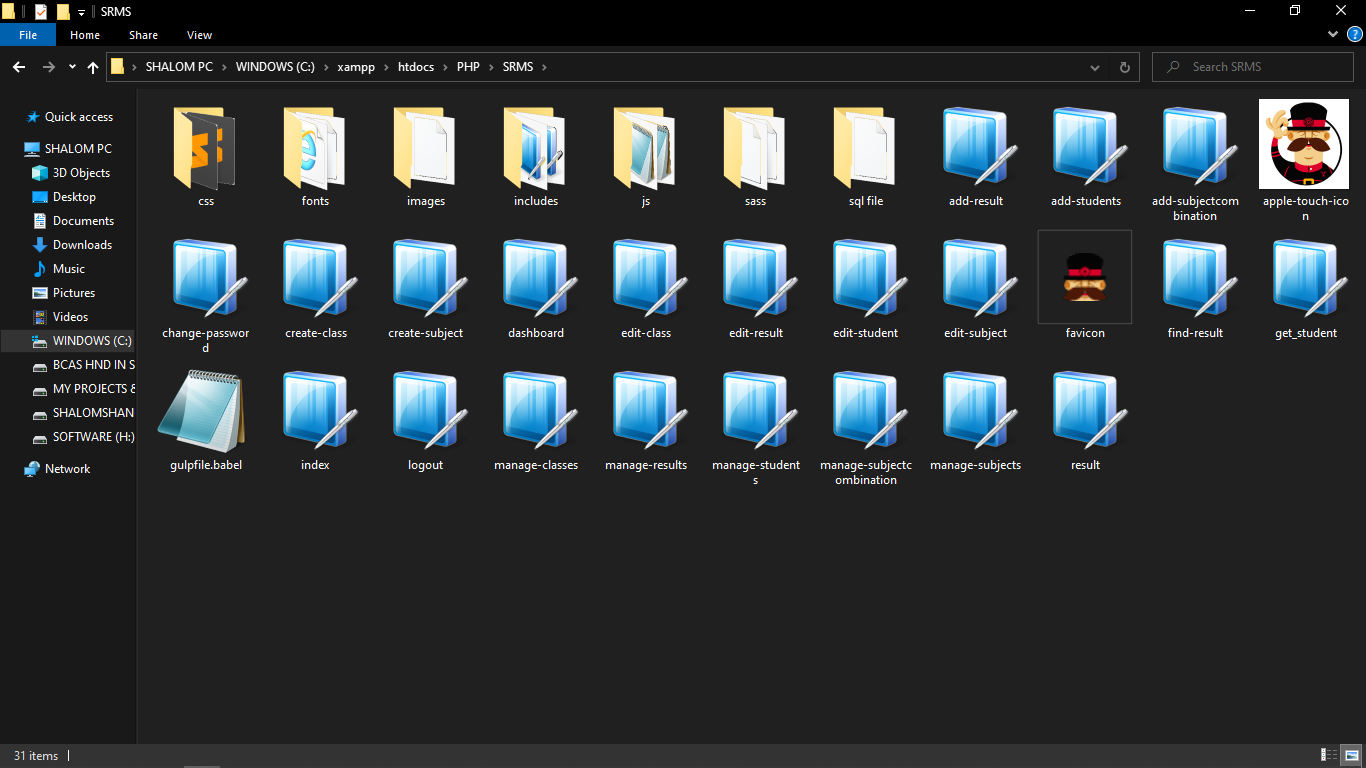
## Technology used in the system

* HTML
* CSS
* JavaScript
* PHP
* Bootstrap

This Student Result Management System. It is the first form that loads to the entire Student Result Management System. When the first time it loads, it creates a folder called SRMS in the C:\xampp\htdocs\PHP\SRMS directives in order to maintain the system configurations.



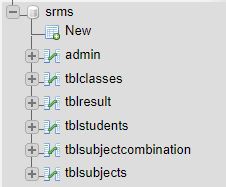
This is the SRMS folder. That will be created at the first time when we are launching the web page. Inside the folder, there will be some php and text files will be created in order to maintain the system settings and configurations. This is the main component of the web page.



Inside the CRMS folder, these are some main files that will be created at the first times when we are opening the Student Result Management System. There are 21 php files created.

## MySQL database

The Students information are stored in the MySQL database. But the system settings and configurations are stored in php files inside the SRMS folder. The database has some tables in order to maintain and save the data of the Student Result Management System.



**Admin:** Inside the admin table stored the user account and passwords

**Table Classes:** Inside the table classes stored the total class list

**Table Result:** Inside the table result stored students result

**Table Students:** Inside the table students stored total count students

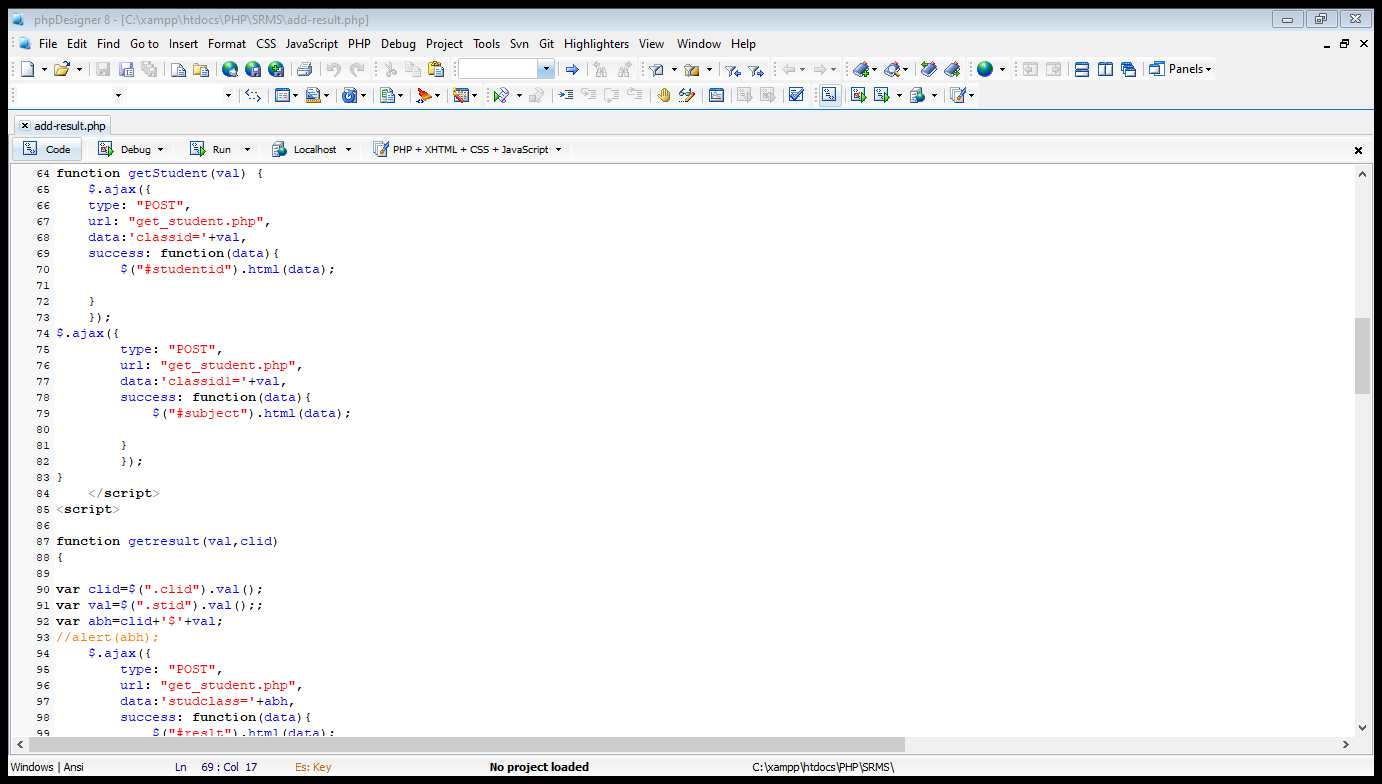
**Table Subject Combination:** Inside the table subject combination stored the subject combination

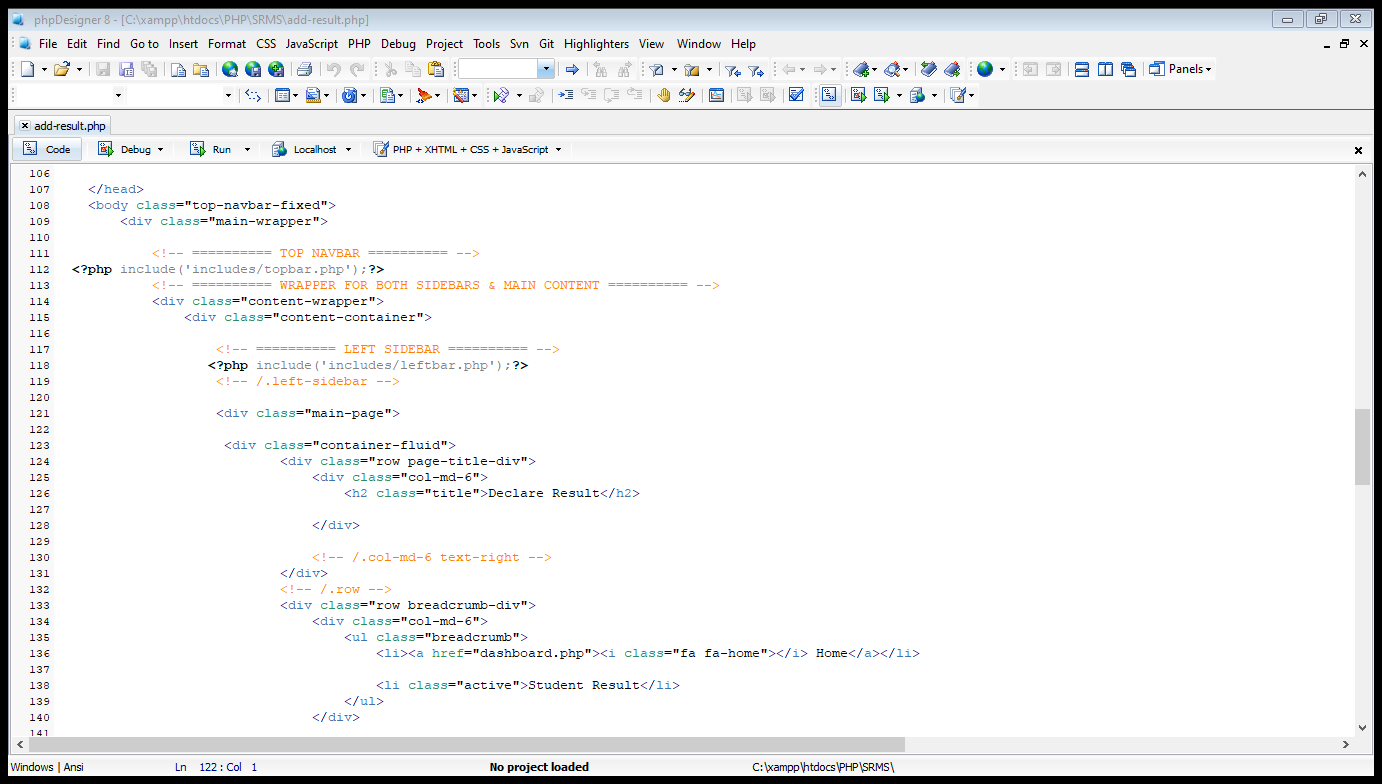
**Table Subjects:** Inside the table subject stored the total subjects

# Important code snippets

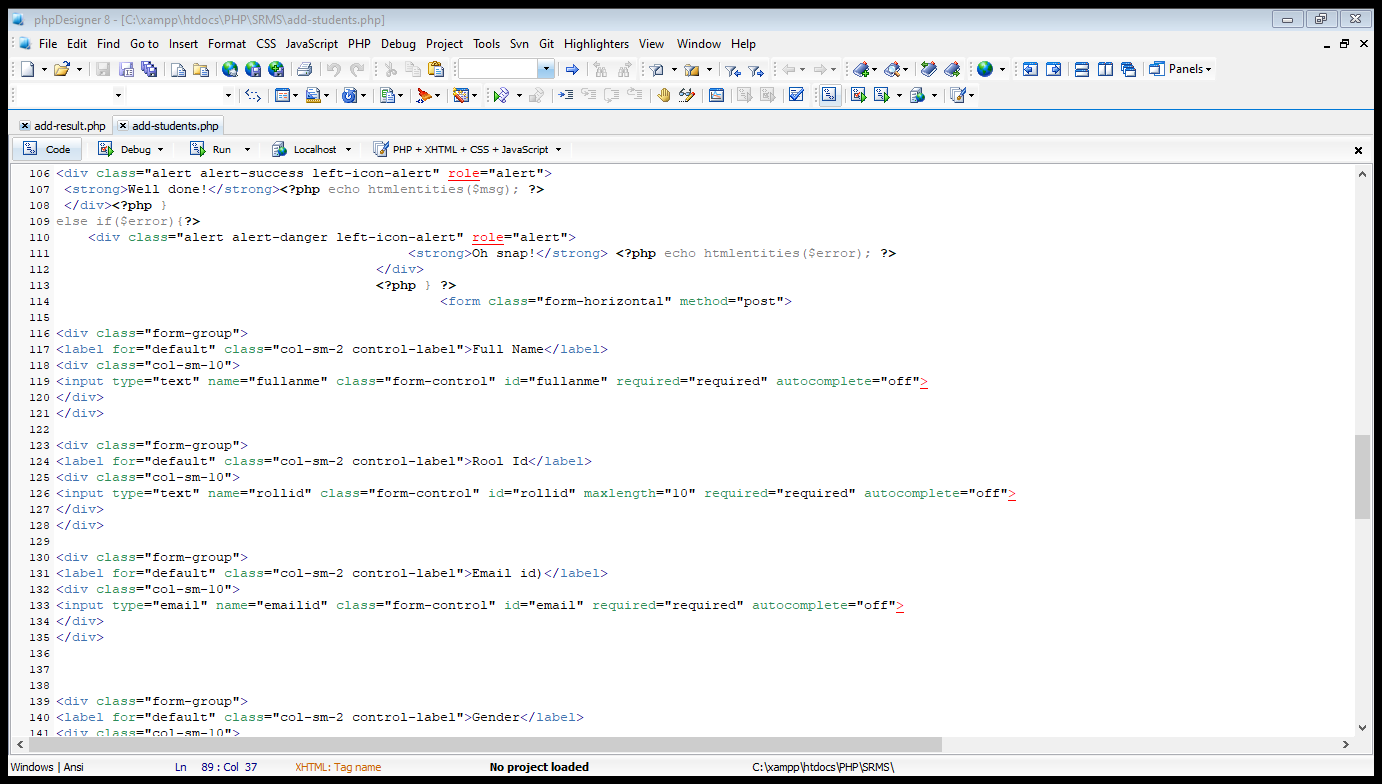
The Student Result Management System has been built 21 php file of codes. The system was developed with more lines of php, css, JS, HTML codes. But in this section, we will only look the most important code snippets of the web page.

## Add result

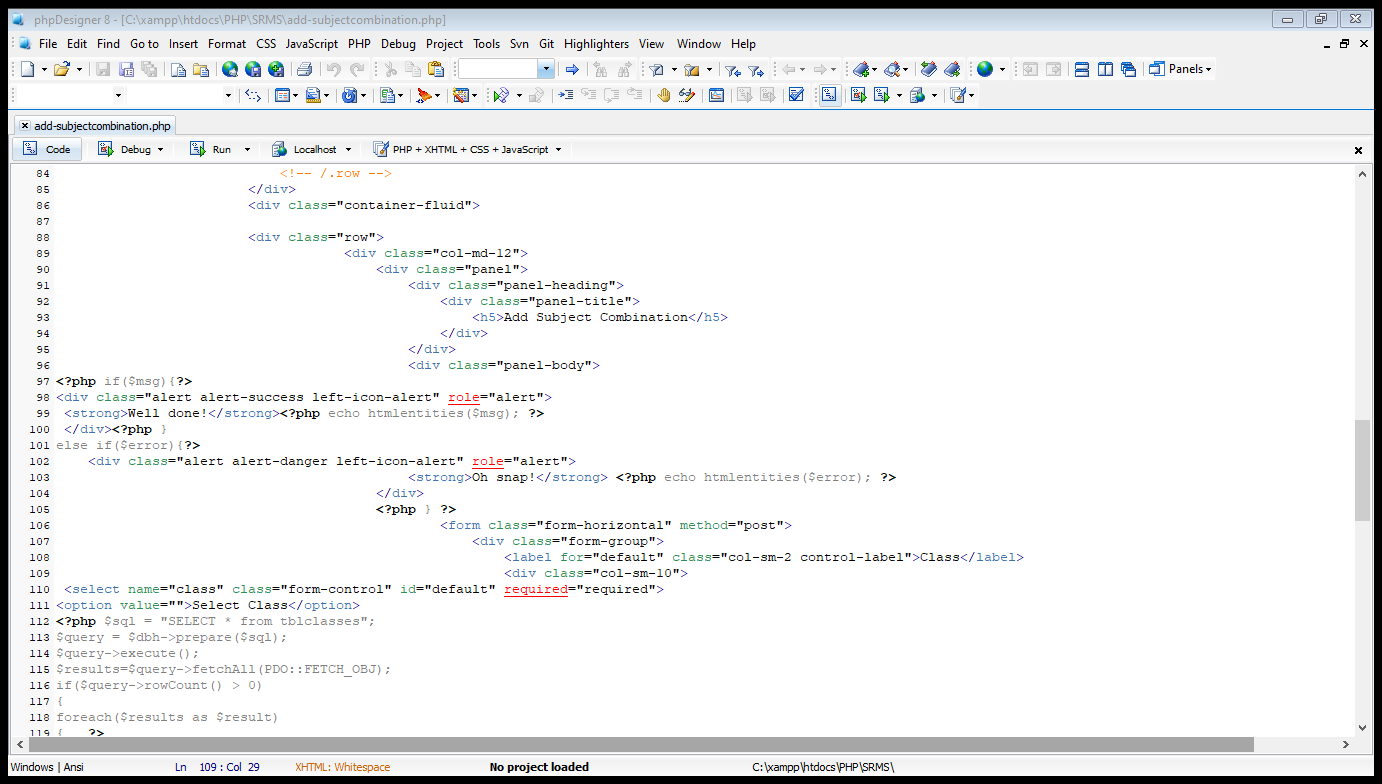




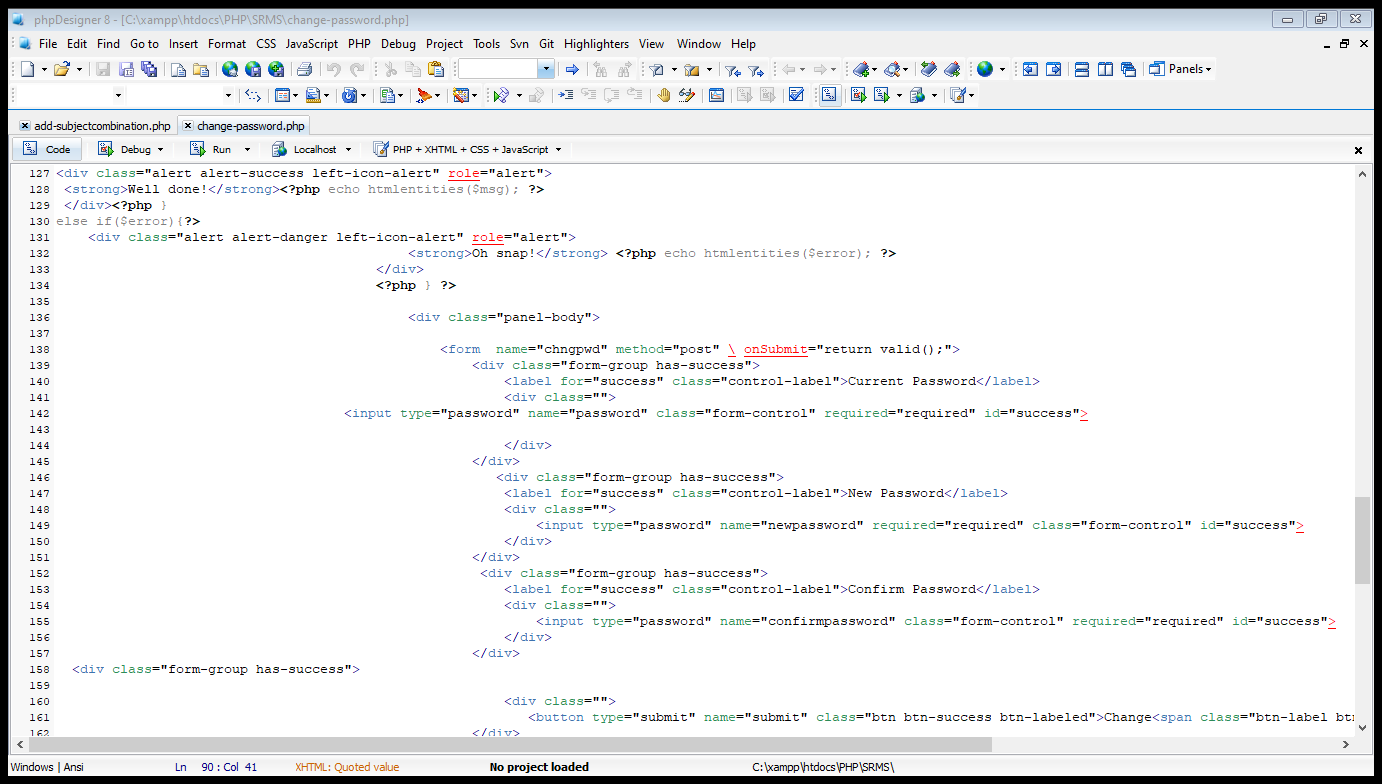
## Add students



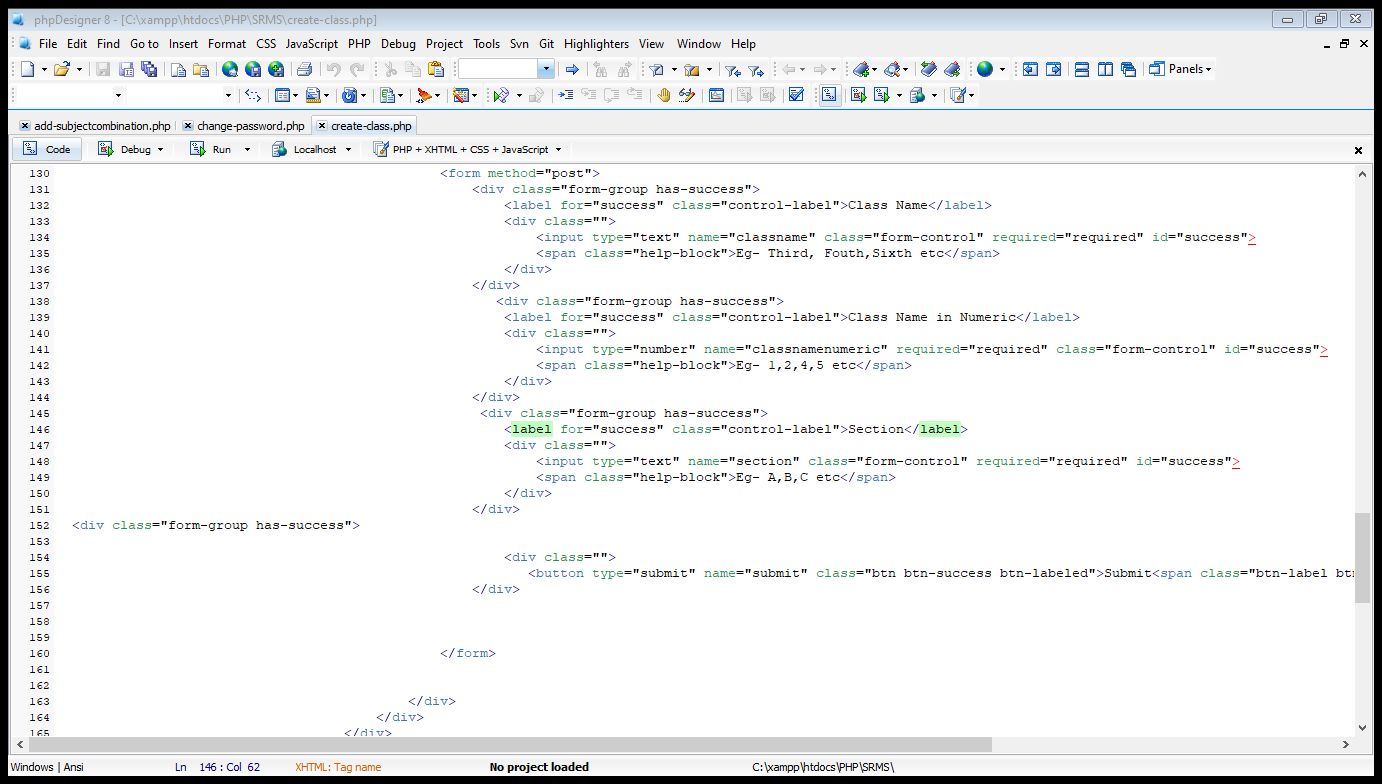
## Add subject combination



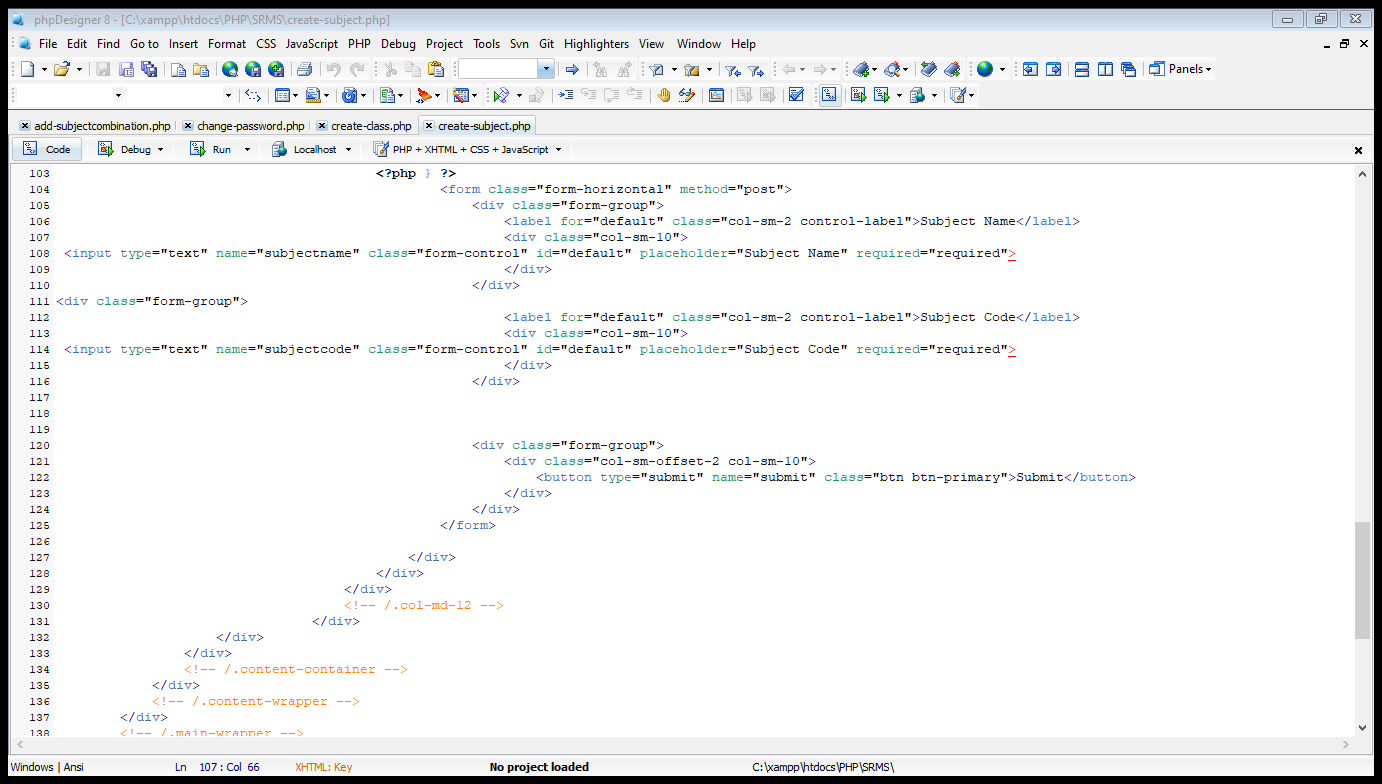
## Change password



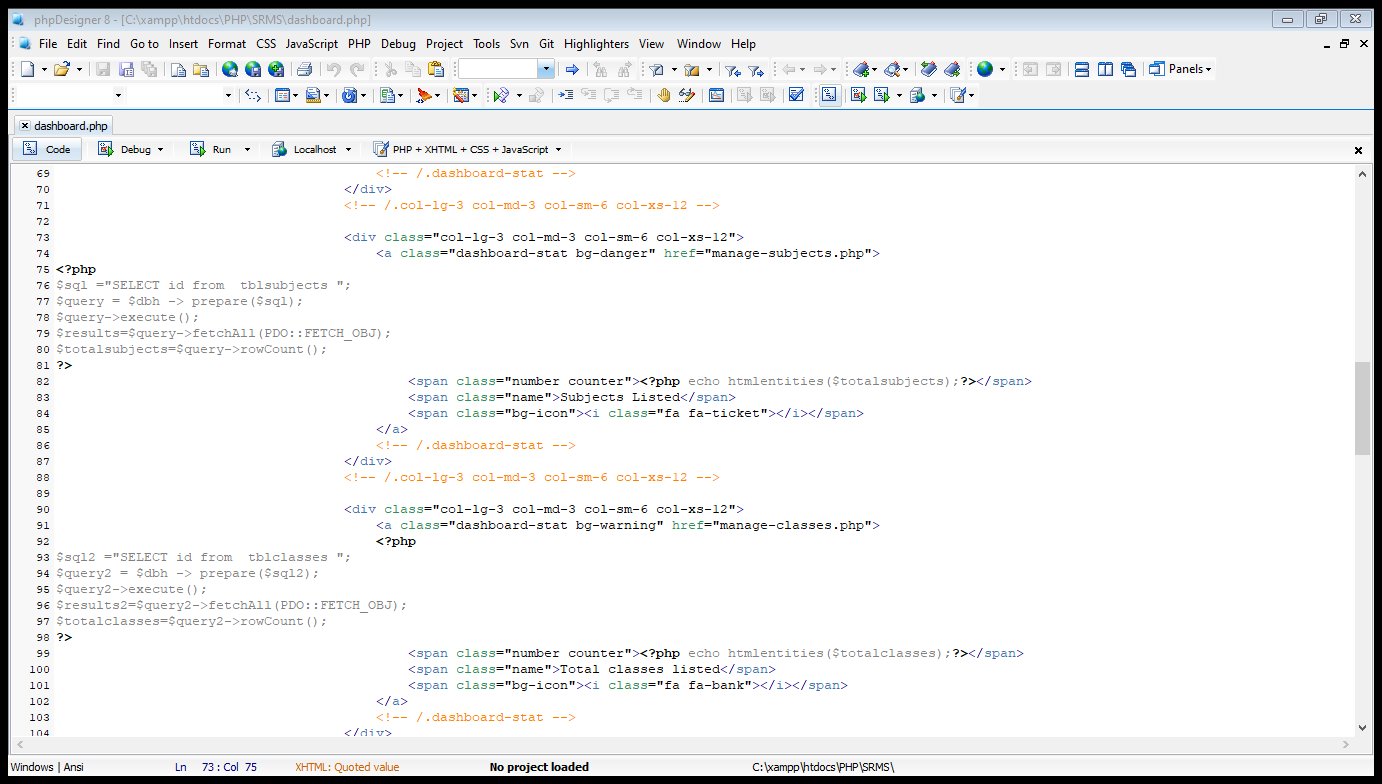
## Create class



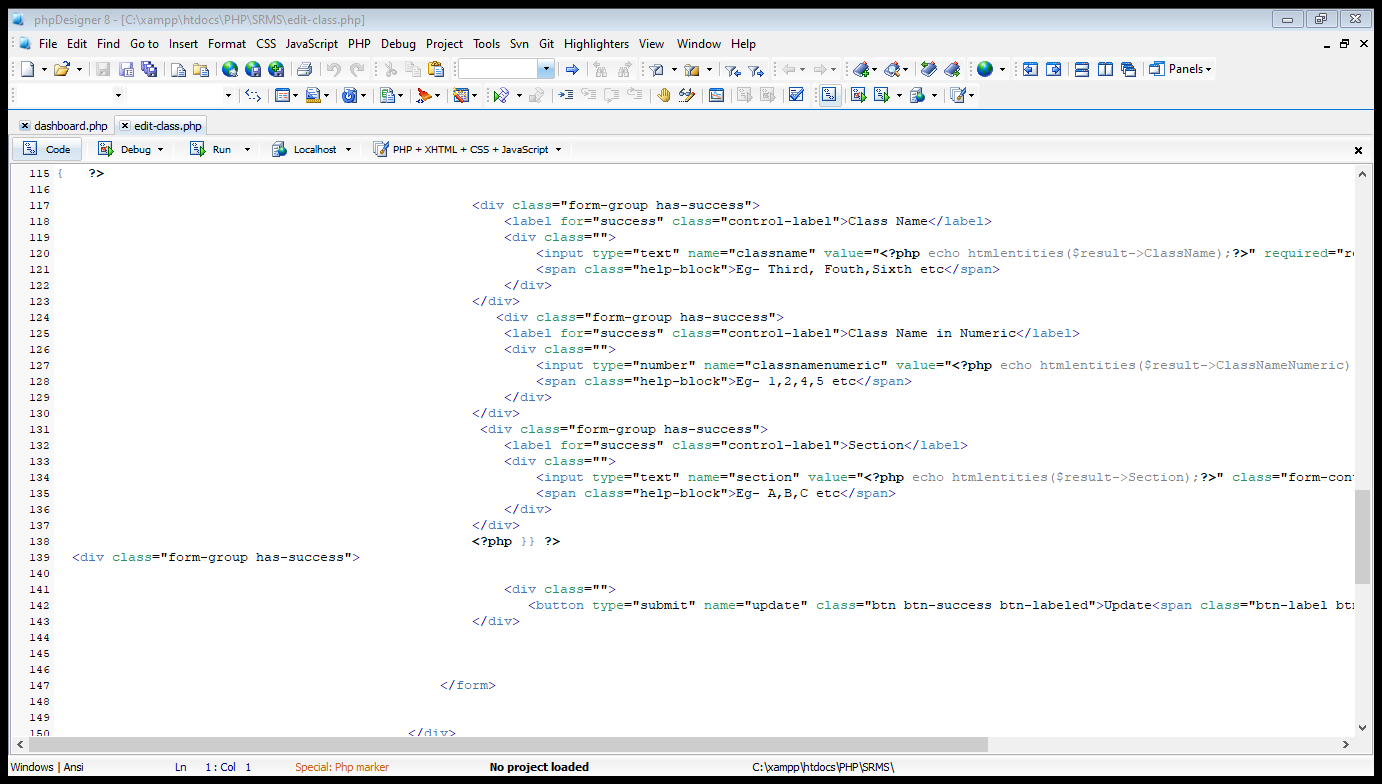
## Create subject



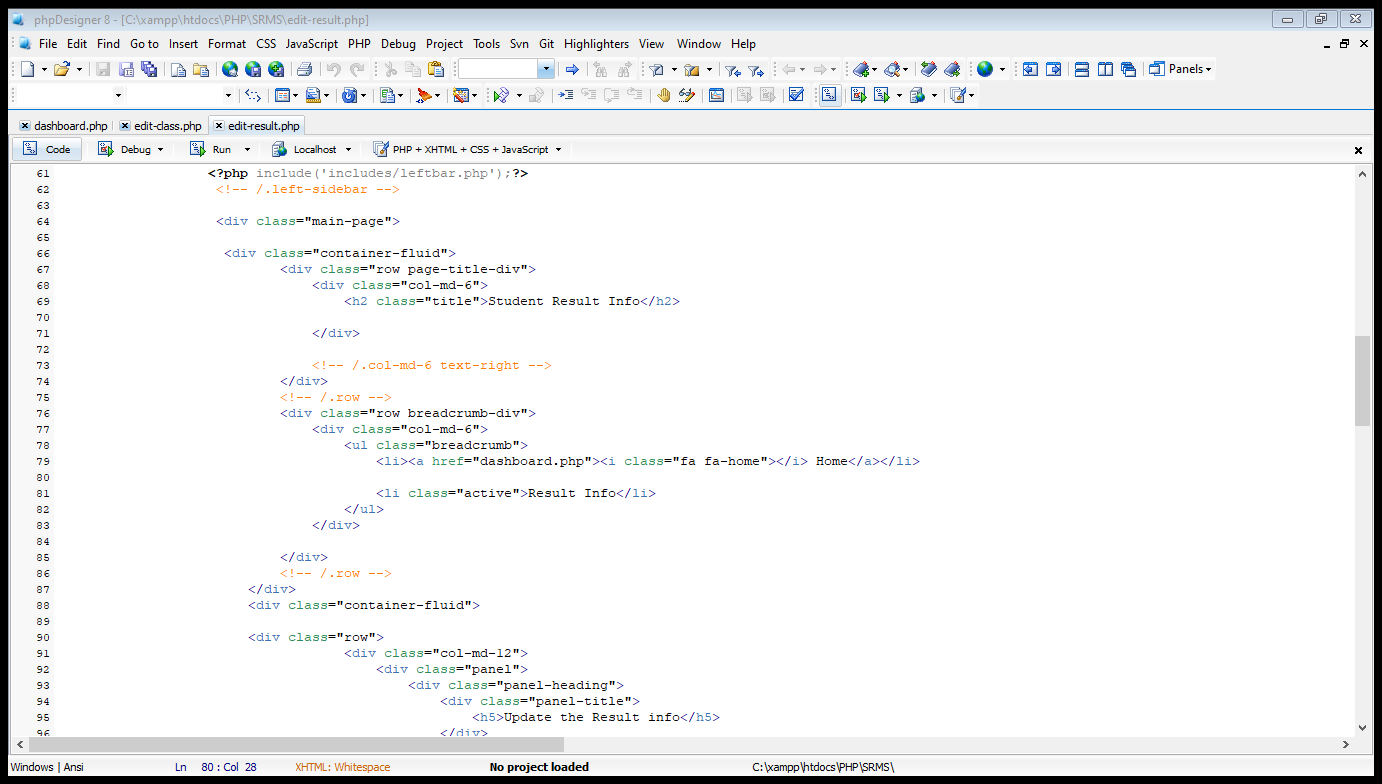
## Dashboard



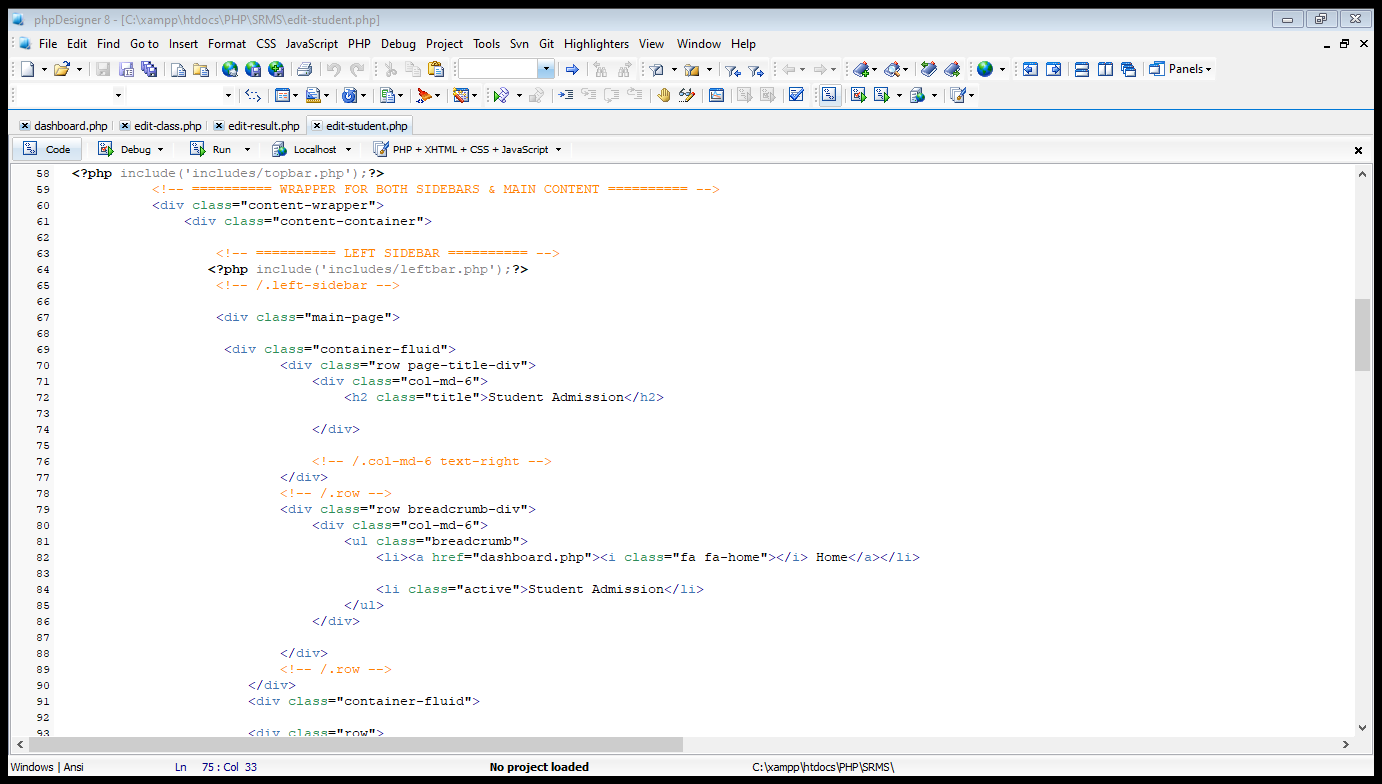
## Edit classes



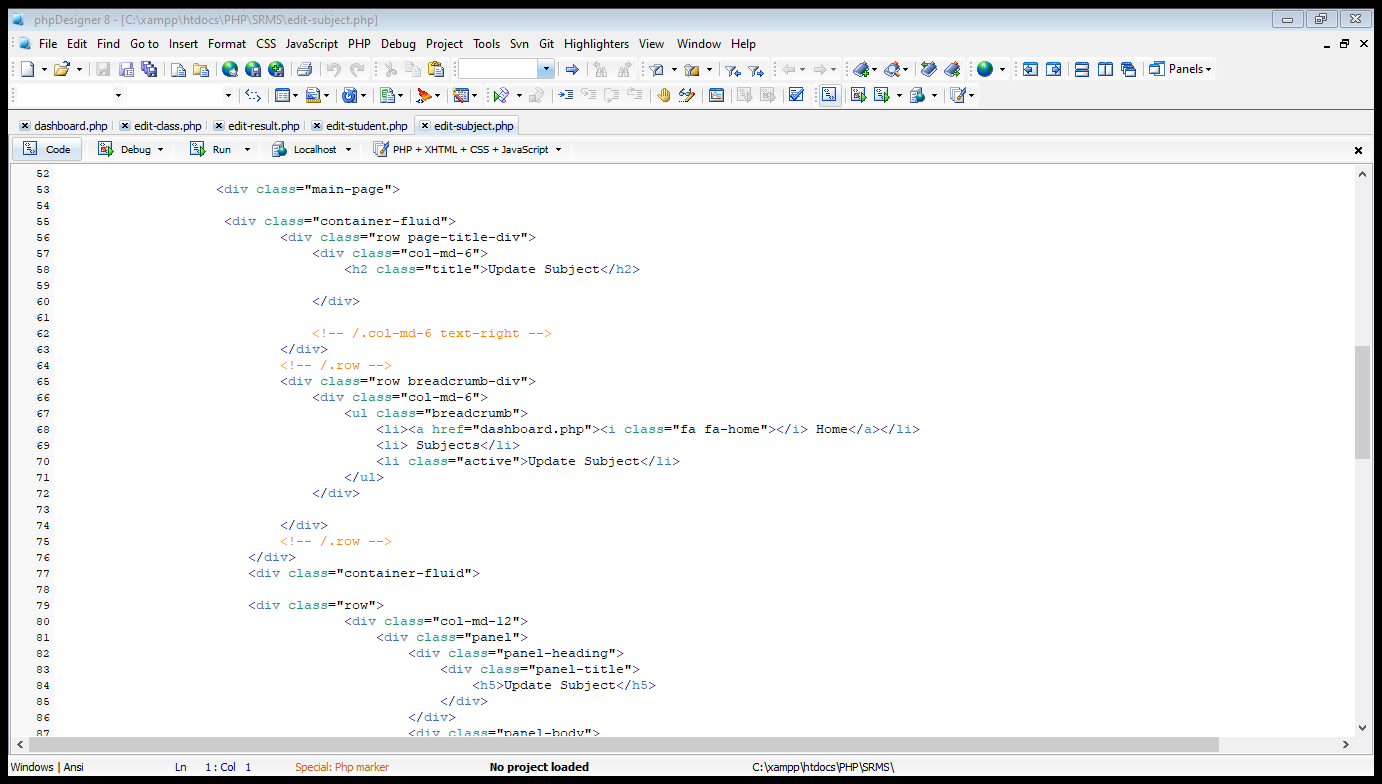
## Edit result



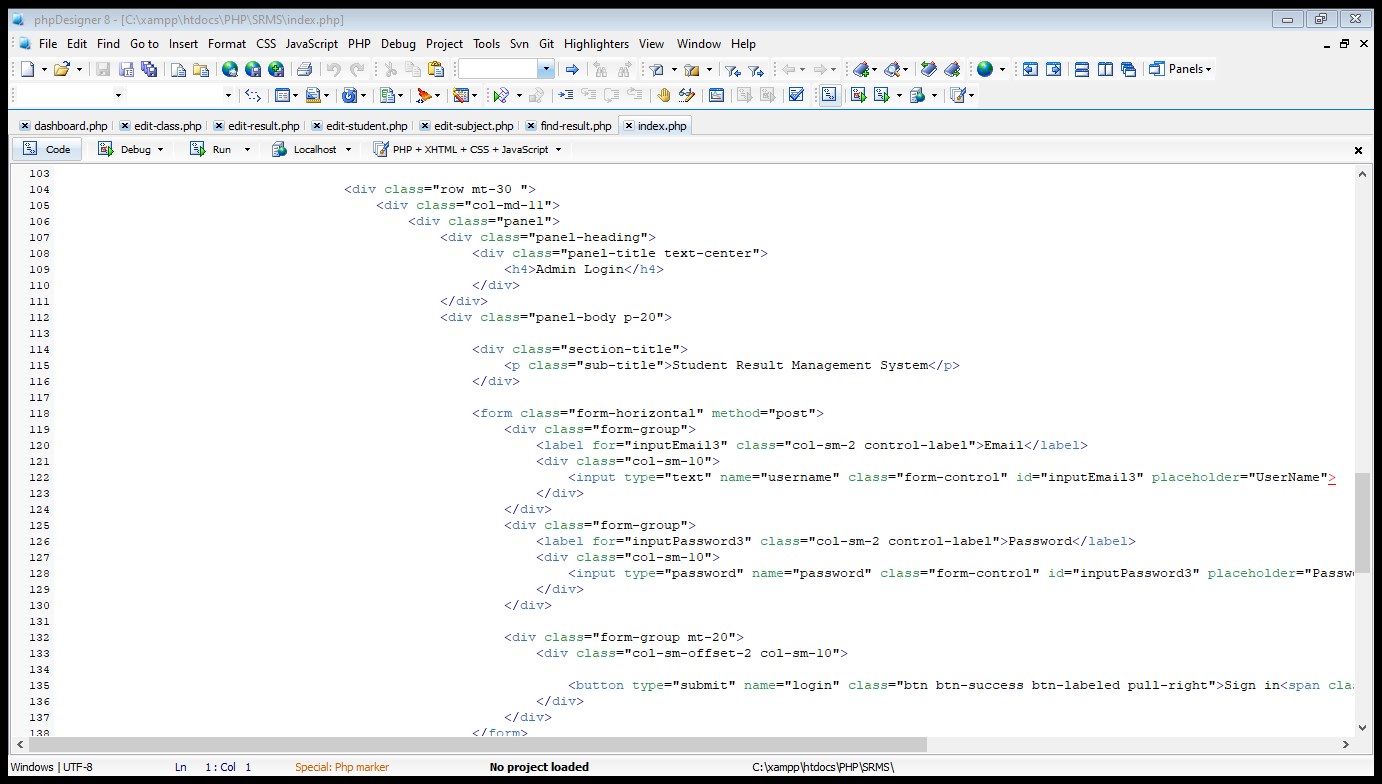
## Edit students



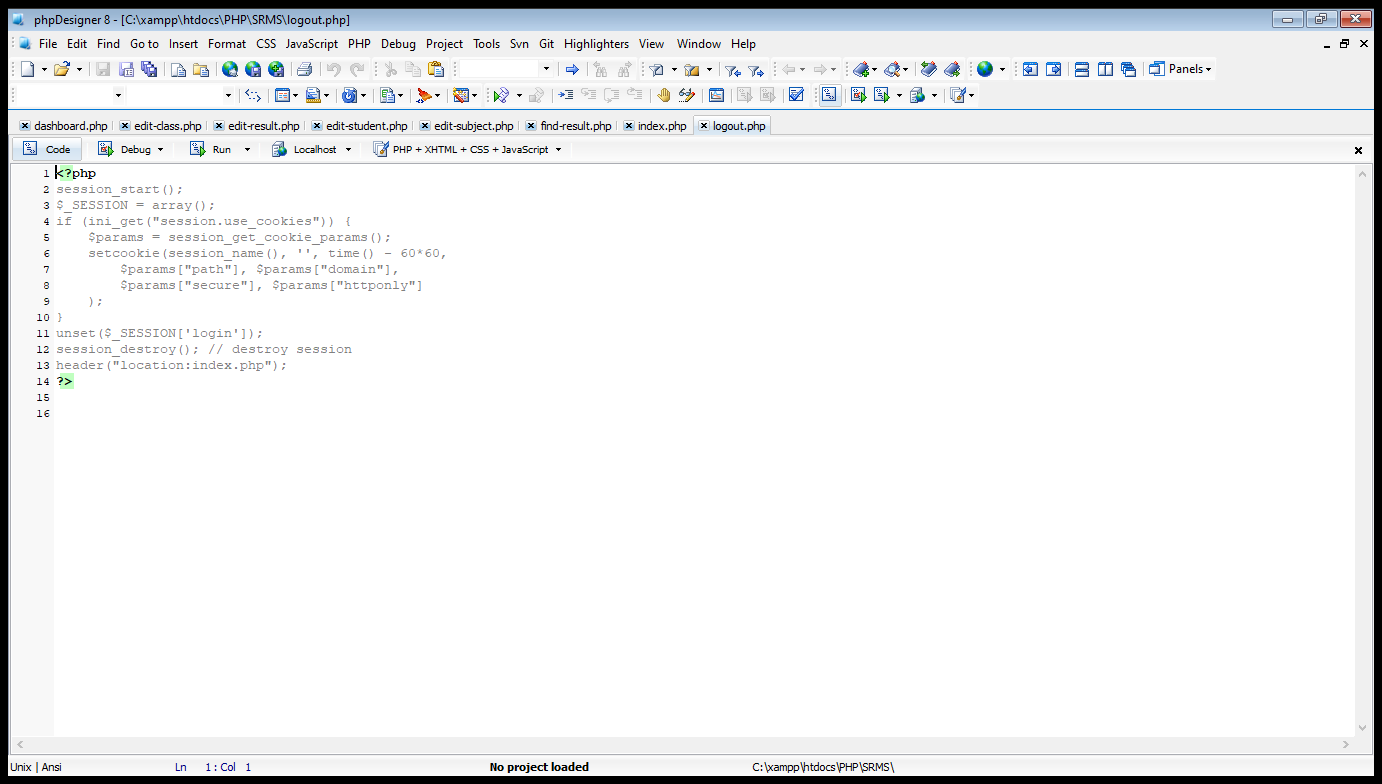
## Edit subjects



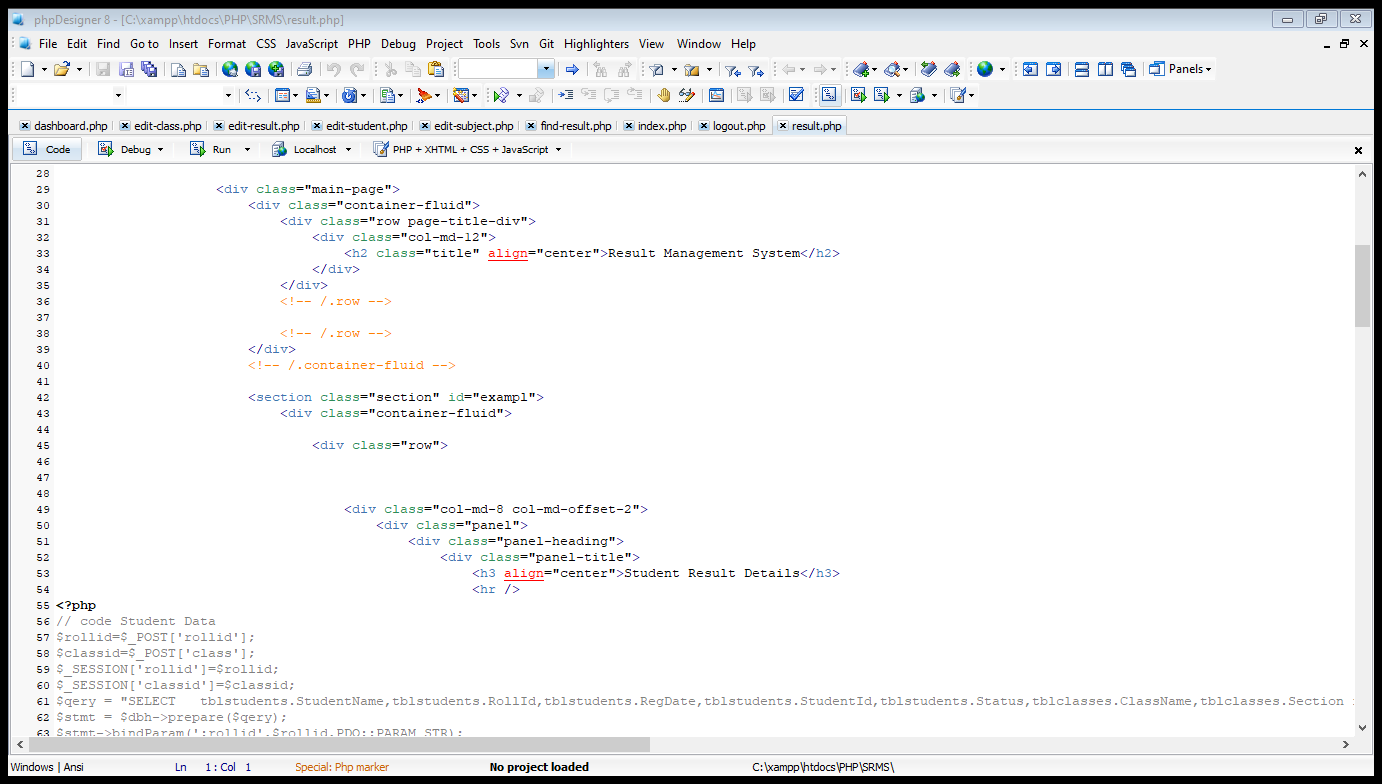
## Index



## Logout



## Result



# Part:-7

# Introduction

We have all been in a situation where a certain bug entirely ruined our app user experience. But why do things like that happen? The answer is quite simple: the lack of web application testing. The testing phase is an incredibly important part of app development yet some developers underestimate it. The problem is that the chance of a bug appearing increases with every life of code and the costs of bug fixing rise with the time. However, with a proper website testing, none of that has to happen. So if the application testing runs smoothly, the app is ready to be released. It’s as simple as that. If the development is done with Agile principles applied, it’s important to “mix” the testing phase with the development phase properly. In this report gives one good approach to doing that.

# Functionality testing

Check all the links:

Links from links to domain names, internal links, jump links, links used to administrators or other users are clearly checked and browsed thoroughly.

Other orphaned pages and broken links are not available.

Validate your HTML/CSS:

Syntax errors in html do not and have income data from different search engines.

**Links:**

* Internal links are stable and do not stick to each other
* External links are stable
* The link is broken, no broken link occurred

**Forms:**

* Field validation: The validation is certified to ensure the information
* Error message for wrong input: When entering the wrong information is an error immediately or not received
* Optional and Mandatory fields: In cases where it is required to enter the correct information properly and not be missing, it will immediately notify

# Usability testing

Test for navigation:

The way users surf websites is easy and convenient because the easy control of buttons on each box allows users to read more or to other pages to display information, each button will have its properties on each box.

## Usability testing includes the following

* Simple website easy to use so users can search for result / marks easily.
* The supply is clear enough for users to be assured of the best use.
* It is also suitable and easy to use.

## Other user information for user help

* Sitemaps must be present with the links in the site for the appropriate view.
* Optional items are clearly verified so website searches help them find it easily and conveniently.

# Interface testing

The main interface testing are:

* Web server and application server interface
* Application server and database server interface

All interactions between the server are explicitly executed and errors are processed as soon as possible and in sequence.

If database or web server returns an error message for any query by application server then application server should catch and display these error messages appropriately to the users.

# Compatibility testing

Browser compatibility:

Web application really depend on the browser, different browsers have different configurations and settings that your website must be compatible with. Just like website has tested application on different browsers such as chrome, opera, internet explorer, Firefox and safari different version.

**OS compatibility:**

It can correspond to operating systems such as Windows, Unix, and MAC.

**Mobile browsing:**

The website that has been tested running on mobile browsers is still smooth without errors.

**Printing options:**

When making the font printing page, it is not too big and not too small to read, the page alignment with the page graphics and beautiful graphics is not too eye-catching but not simple, the alignment between the pages is even.

# Conclusion

In conclusion, this assignment has more detailed information about the web design & development concept. As a student I have gained a lot of knowledge about web design & development and other essentials things about web design & development concept.

In this assignment, I have done a lot of things which are very useful to improve my knowledge. Specially I have a done a 20 minutes online YouTube presentation that focuses on the basic web design & development essentials such as Students Result Management System.

I have gained a bunch of knowledge about Web Design & development. I have created Students Result Management System. Then, I had to write a website for that.

I have improved a lot knowledge about web design & development Identify the purpose and types of DNS, including explanations on how domain names are organized and managed, purpose and relationships between communication protocols, server hardware, operating systems and web server software with regards to designing, publishing and accessing a website, relationships between front-end and back-end website technologies and explain how these relate to presentation and application layers, differences between online website creation tools and custom built sites with regards to design flexibility, performance, functionality, User Experience (UX) and User Interface (UI), design document for a branded, multipage website supported with medium fidelity wireframes and a full set of client and user requirements, design document with appropriate principles, standards and guidelines to produce a branded, multipage website supported with realistic content and suitable Test Plan identifying key performance areas and use it to review the functionality and performance of my website. It was very useful for me. I have also written a same php file.

I have learned so many things about the web design & development and other essential things. I have improved a lot of skills in testing. This assignment taught me a lot of things.

Finally, I express my deepest thanks to the assessor Mr. Mohamed Nizzad. Who gave this awesome opportunity to improve my web design & development knowledge.

# Reference

https://dyn.com/blog/dns-why-its-important-how-it-works/> ( accessed Aug 9, 2018 )

https://cloudacademy.com/blog/how-dns-works/>( accessed March 26, 2019 )3.

https://support.opendns.com/hc/en-us/articles/227986607-Common-DNS-Request-Types>(accessed Jan 16, 2020 )4.

https://en.wikipedia.org/wiki/Communication\_protocol>(accessed 1 February 2020 )5.

https://www.quora.com/What-is-server-hardware> (accessed June 31, 2019 )6.

https://developer.mozilla.org/vi/docs/Learn/Common\_questions/What\_is\_a\_web\_server>(accessed March 23, 2019 )

https://www.howtogeek.com/361572/what-is-an-operating-system/>(accessed AUGUST 8, 2018)

https://learn.onemonth.com/frontend-vs-backend-developers/> ( accessed June 27, 2019 )

https://ecenter.custhelp.com/ci/fattach/get/1125545/1457560254/redirect/1/filename/PM\_Domains\_%20Domain%20Basics\_\_0A2B.pdf#:~:text=is%20a%20domain%3F-,Domains%20are%20managed%20by%20using%20a%20worldwide%20system%20of%20domain,addresses%20used%20by%20networking%20equipment.

http://www.differencebetween.net/technology/difference-between-frontend-and-backend/#:~:text=Summary%20of%20Frontend%20verses%20Backend&text=While%20the%20former%20is%20everything,server%2Dside%20of%20the%20application.&text=Frontend%20is%20basically%20what%20users%20see%20whereas%20backend%20is%20how%20everything%20works.

techopedia.com/definition/5393/site-map

https://www.experienceux.co.uk/faqs/what-is-wireframing/

https://www.computerhope.com/jargon/h/homepage.htm

https://www.softwaretestinghelp.com/web-application-testing/

https://nibbler.silktide.com

https://www.simplilearn.com/how-to-test-your-website-article

https://www.softwaretestinghelp.com/how-to-write-test-plan-document-software-testing-training-day3/

https://www.treefrog.ca/how-to-qa-your-website

https://www.webilize.com/Blog/Details/49/quality-assurance-checklist-for-website-design-anddevelopment

https://kruschecompany.com/quality-assurance-in-projects/

https://en.wikipedia.org/wiki/Communication\_protocol

https://www.coursehero.com/u/file/42900843/task-2docx/#/question

https://www.coursehero.com/file/p52e11ns/How-front-end-and-back-end-relate-to-presentation-and-application-layer-Front/

https://www.altexsoft.com/blog/front-end-development-technologies-concepts/

https://blog.tubikstudio.com/back-end-development-foundation-of-digital-product/

https://www.usertesting.com/blog/ui-vs-ux

https://blog.raincastle.com/understanding-ux-ui-design-vs-web-design

https://en.wikipedia.org/wiki/Website\_builder#:~:text=Website%20builders%20are%20tools%20that,to%20build%20their%20private%20site.

https://www.icecubedigital.com/blog/advantages-of-custom-built-sites-vs-online-website-builder-tools/