Module:1 : INTRODUCTION TO INTERNET

World Wide Web

•Initiated at CERN (the European Organization for Nuclear Research)

–By Tim Berners-Lee (1989)

•GUIs

–Berners-Lee (1990)

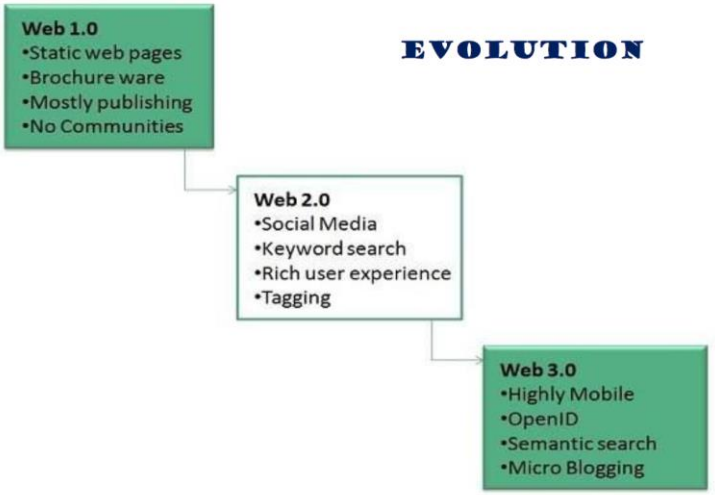
•Mosaic(1993)

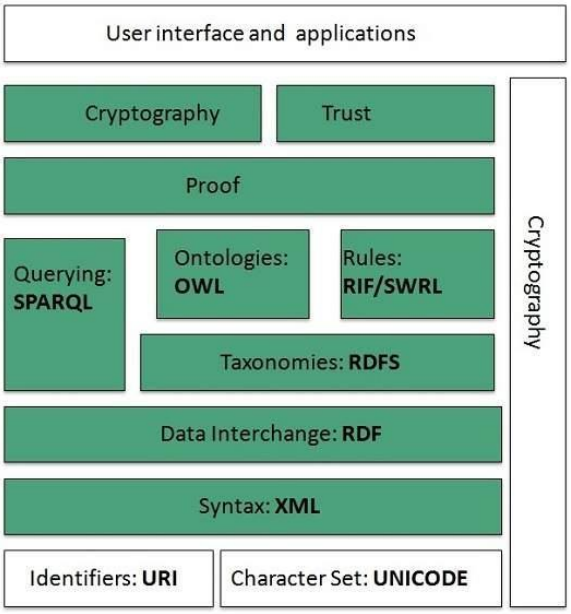
–a hypertext GUI for the X-window system

–HTML: markup language for rendering hypertext

–HTTP: hypertext transport protocol for sending HTML and other data over the Internet

–CERN HTTPD: server of hypertext documents





Ahyper link points toa whole documentor toa specific element within a document.

•Hypertext is text with hyper links

Dynamic updates of web pages

Binding different event handlers to different dynamic elements of a Webpage

Using Scripting techniques to refresh only the dynamic parts of the Webpage instead of whole page

WWW prefix

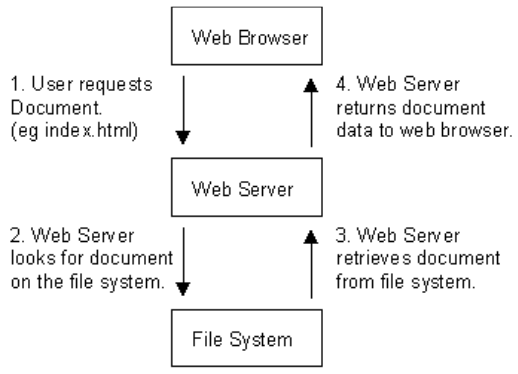
The hostname for a webserver is often www, in the same way that it may be ftp for an FTPserver, and news or nntp for a USENET news server.

The scheme specifiers(http://orhttps://)in URIs refer to the the communication protocol to be used for therequestandresponse.

•The HTTP protocol is fundamental to the operation of theWorldWideWeb;
•The added encryption layer in HTTPS is essential when confidential information such as passwords or banking information are to beex changed over the public Internet

Web Servers

•The primary function of a webserver is to deliver webpages on the request to clients using the Hypertext Transfer Protocol (HTTP).
•DeliveryofHTMLdocumentsandanyadditional contentthatmaybeincludedbyadocument,such asimages,stylesheetsandscripts.
•Auseragent,commonlyawebbrowserorwebcrawler,initiatescommunication by making a reques tfor aspecific resource using HTTP

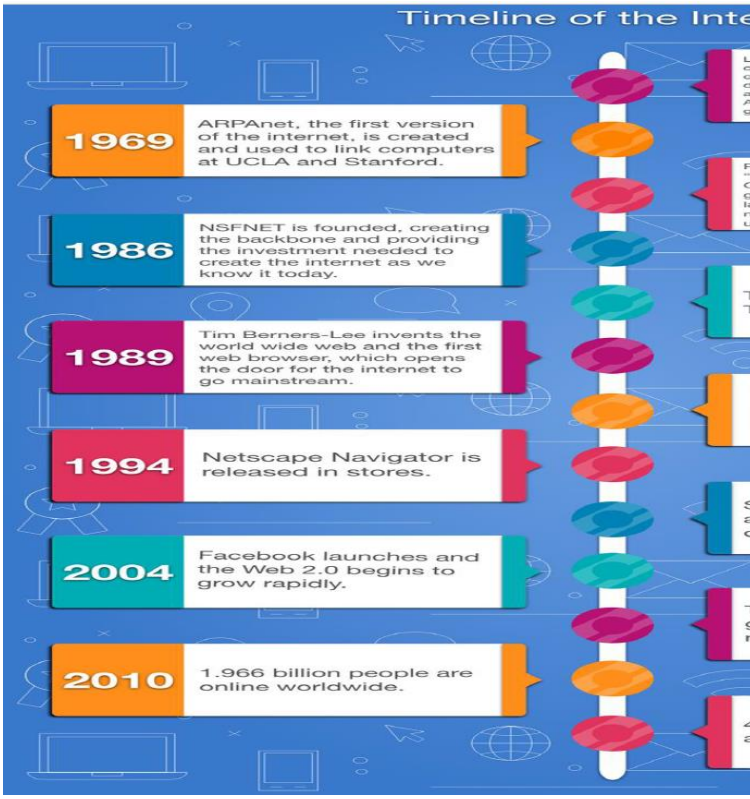


Through HTML and URIs the Web was vulnerable to attacks like cross-site scripting(XSS)
•Todaybyoneestimate,70%of all websites are open to XSS attacks on their users.
Didn't attempt to maintain links, just a common way to name things
–Uniform Resource Locators (URL)
http://www.vit.ac.in/index.html
Service
Hostname
File Path
HyperText Transfer Protocol

What Is the Internet?

A network of networks,joining many government,university and private computers together and providing an infra structure for the use of E-mail,bulletinboards,filearchives,hypertextdocuments,databases and other computational resources.

Estimated year of Origin	1969, though opening of the network to commercial interests began only in 1988	1993
Name of the first version	ARPANET	NSFnet
Comprises	Network of Computers, copper wires, fibre-optic cables & wireless networks	Files, folders documents st in various computers
Governed by	Internet Protocol	Hyper Text Transfer Proto
Dependency	This is the base, independent of the World Wide Web	It depends on Internet to wo
Nature	Hardware	Software



Who owns the internet today?

–NSFNET was officially privatized in 1995.
–Management of internet is managed by the internet Network information center
–(InterNIC)which subcontracts various tasks to such companies as AT&Tand Network Solutions
–Today no body owns the internet

Protocols

The Internet protocol suite is the set of communications protocols used for the Internet and similar networks,and generally the most popular protocolstack for wide area networks.

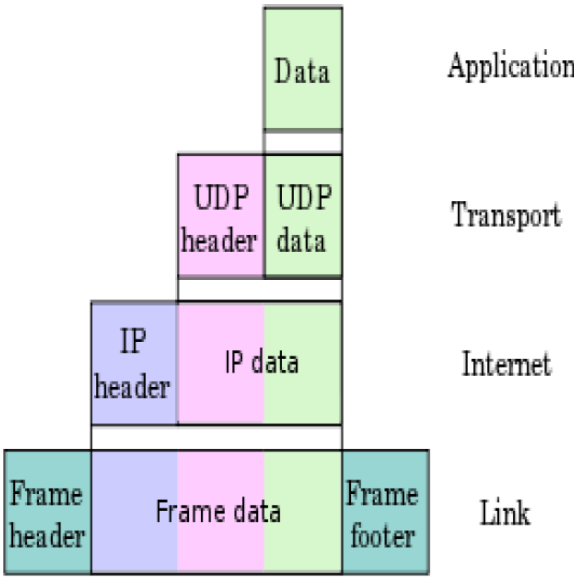
It is commonly known asTCP/IP,because of its most important protocols:Transmission Control Protocol(TCP)andInternetProtocol(IP),whichwerethefirstnetworkingprotocolsdefinedinthisstandard.

The link layer (commonlyEthernet) contains communication technologies for a local network.

The internet layer(IP) connects local networks,thus establishing internet working.

The transport layer(TCP) handles host-to-host communication.

The applicationlayer(forexampleHTTP) contains all protocols for specific data communications servicesonaprocess-to-processlevel.



Routing

Routing is the process of selecting paths in a network along which to send network traffic.

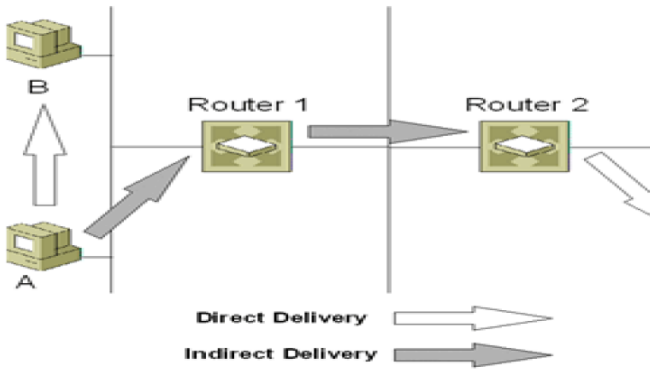
•Routing is performed for many kinds of networks

•Computers and routers use routing tables to direct IP packets among locally connected machines.

•Tables can be constructed manually or automatically via DHCP for an individual computer or a routing protocol for routers themselves.

•In single-homed situations ,a default route usually points "up" toward an ISP providing transit.

Routing



Direct delivery occurs when the IP node (either the sending ho router) forwards a packet to the final destination on a directly a The IP node encapsulates the IP datagram.

Indirect delivery occurs when the IP node (either the sending l router) forwards a packet to an intermediate node

Hardware
Servers
Communications Media
Storage Area Networks (SANs)
Connecting to the Internet
Digital Subscriber Lines (DSL)
Broadband
Integrated Services Digital Network (ISDN)

Software

Application Service Providers (ASPs)
Databases

- **Server**

– A host on the Internet that manages network resources and fulfills requests from clients

– Web servers, e-mail servers, database servers and file servers
– A single server may provide multiple services

A Web server stores Web pages and delivers the pages to clients upon request

Protocols for delivering information over the Web
– Hypertext transfer protocol (HTTP)
– File transfer protocol (FTP)
– Post office protocol (POP)
– Simple mail transfer protocol (SMTP)

• **Communications medium**
– The hardware that connects computers and other digital equipment
• **Bandwidth**
– Indicates how much data can be transferred through the medium in a fixed amount of time
– Usually measured in bits per second (bps)
• **Copper wire / fiber-optic cable**
– The primary communications mediums

The strength of a signal transmitted over a communications medium is reduced as the signal travels farther and farther
• **Repeater**
• Can be used to alleviate this problem by amplifying and retransmitting the signal across segments of copper wire or fiber-optic cable
• **Transmission costs**
• Fiber-optic cable is more expensive than copper wire
• Installation of fiber-optic cable is more complicated than installation of copper wire
• Fiber-optic cable requires fewer repeaters

Storage Area Network (SAN)
– Provides high-capacity, reliable data storage and delivery on a network
– Allows network administrators to collect data in logical groups on data servers distributed throughout the network
– SAN devices store large volumes of data and may also provide backup and recovery services

Mirroring technology
– A SAN device stores redundant copies of data, so that if one copy is lost or damaged, a mirrored copy can be used

Fiber-channel technology
– A high-speed communications medium based on fiber-optic technology that provides transfer rates of 100 Mbps

Internet Service Provider (ISP)

Modem

User connects to an ISP using the modem, which then connects the user to the Internet

Takes digital signals from the computer and turns them into analog signals

• **Digital Subscriber Lines (DSL)**
– Offers high-bandwidth Internet access over existing copper telephone lines
– Splits your phone line into three information-carrying channels
• **IAD (integrated access device)**
– Provides network connections for high-speed Internet access, as well as connections for multiple voice telephone lines
• **ADSL (asymmetric DSL)**
– The connection speed for sending data to the Internet (upstream) is slower than the connection speed for receiving data from the Internet (downstream)
• **SDSL (symmetric DSL)**
– Transfers data at the same speed both upstream and downstream
• **VoDSL (voice over DSL)**
– Provides voice telephone services and high-speed data access over a single standard telephone line

Broadband
– A category of high-bandwidth Internet service provided mainly by cable television and telephone companies to home users
– Can handle voice, data and video information

– Enables videoconferencing, real-time voice and streaming-media applications
– Always connected, eliminating the need to dial into an ISP

Cable modem
– Translates digital signals for transmission over the same cables that bring cable television to homes and businesses
– Connections is shared among many users

Application Service Providers
• Provide customized business software applications over the Internet
• Maintains and updates the application as necessary
• Companies can eliminate the costs associated with developing and maintaining business applications
• Virtual private networks (VPNs)
– Allow customers to connect to their applications securely over the Internet
– VPNs use the point-to-point tunneling protocol (PPTP) to create a secure channel of communication between the customers.
Examples:
• Google Spreadsheets• Google Docs• Free Online Logo Makers

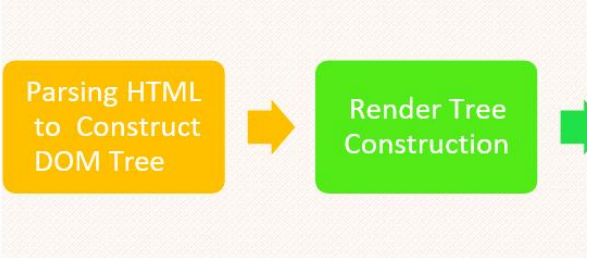
Web Browsers

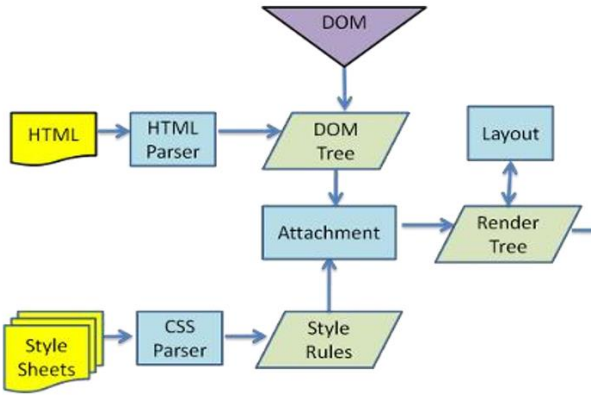
A software application that retrieves and displays information from a server including web pages, text, images, videos, and other contents.
Why do different browsers respond differently to websites, and why is there more than one to begin with?
How do browsers work and where did the need for cross-browser testing come from?
By understanding the history and backend of some major browsers

The main function of a browser is
• To present the web resource, by requesting it from the server and displaying it in the browser window.
• The display the resource usually an HTML document, PDF, image, or some other type of content.
• The find the location of the resource specified by the user using a URI (Uniform Resource Identifier).
• To interpret and display HTML files specified in the HTML and CSS specifications
• Specifications are maintained by the W3C

Browser High Level Structure
• The user interface: Includes the address bar, back/forward button, bookmarking menu, etc.
• The browser engine: Marshals actions between the UI and the rendering engine.
• The rendering engine : Responsible for displaying requested content. For example if the requested content is HTML, the rendering engine parses HTML and CSS, and displays the parsed content on the screen.
• Networking: for network calls such as HTTP requests, using different implementations for different platform behind a platform-independent interface.
• UI backend: used for drawing basic widgets like combo boxes and windows. This backend exposes a generic interface that is not platform specific. Underneath it uses operating system user interface methods.
• JavaScript interpreter. Used to parse and execute JavaScript code.
• Data storage. This is a persistence layer. The browser may need to save all sorts of data locally, such as cookies. Browsers also support storage mechanisms such as localStorage, IndexedDB, WebSQL and FileSystem.

• Different browsers use different rendering engines:
 Internet Explorer - Trident
 Firefox - Gecko
 Safari - WebKit
 Chrome and Opera – Blink (a fork of WebKit)
• WebKit is an open source rendering engine which started as an engine for the Linux platform and was modified by Apple to support Mac and Windows.





Web Server

- A Web server is a program that generates and transmits responses to client requests for Web resources.
- Handling a client request consists of several key steps:
- Parsing the request message
- Checking that the request is authorized
- Associating the URL in the request with a file name
- Constructing the response message
- Transmitting the response message to the requesting client

- The server can generate the response message in a variety of ways:

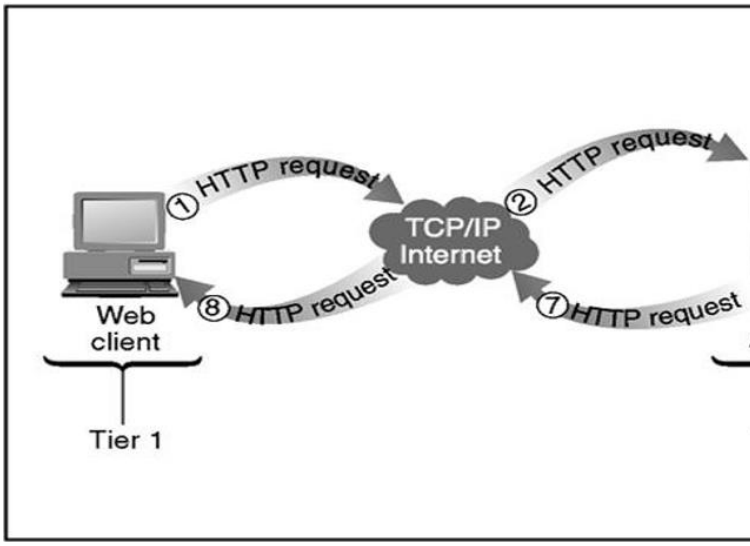
1. The server simply retrieves the file associated with the URL and returns the contents to the client.
2. The server may invoke a script that communicates with other servers or a back-end database to construct the response message.

Two-Tier

- Has only one client and one server
- Request message
- Message that a Web client sends to request a file or files from a Web server
- Typical request message
- Request line
- Optional request headers
- Optional entity body

Multi Tier

- Three-tier architecture
- Extends two-tier architecture to allow additional processing
- N-tier architectures
- Higher-order architectures
- Third tier includes software applications that supply information to Web server



authentication and authorization.

Dynamically Generated Responses

- Dynamically generated responses are created in a variety of ways:

Server-side include(SSI)

- A server-side include instructs the Web server to customize a static resource based on directives in an HTML-like file.

Server script

- A server script is a separate program that generates the request resource.
- The program may run as
- Part of the server
- A separate process

Some techniques for allocating system resources among competing client requests are :

- Event-driven server architecture

An event-driven server

Has a single process that alternates between servicing different requests

Allows the server to serialize operations that modify the same data

Performs non-blocking system calls

Not used in Most high-end Web servers

- **Process-driven server architecture**

A process-driven server

Allocates each request to a separate process

»One master process listens for new connection

»The master process creates, or forks, a separate process for each new connection

Terminates the process after parsing the client request and transmitting the response

»To prevent memory leak

Introduces overhead for switching from one process to another

- **Hybrid server architecture**

The strengths of the event-driven and process-driven models are combined

- Each process would become an event-driven server that alternates between a small collection of requests
- A single process has multiple independent threads
- Main process instructs a separate helper process to perform time-consuming operations

Web Server Hardware Architectures

•Server farms
–Large collections of servers
•Centralized architecture
–Uses a few very large and fast computers
•Distributed/decentralized architecture
–Uses large number of less powerful computers
–Divides the workload among them

•Benchmarking
–Testing used to compare the performance of hardware and software
•Throughput
–Number of HTTP requests that hardware and software combination can process in a unit of time
•Response time
–Time required by server to process one request

Types of hosting services

- Self-hosting
- Shared hosting
- Dedicated hosting
- Collocated hosting
- Mailing Lists

SQL Injection

•Attacker sends invalid data to the web application with the intention to make it do something that the application was not designed/programmed to do.
•SQL query consuming untrusted data.
•Examples String query = “SELECT * FROM accounts WHERE custID = ‘” + request.getParameter(“id”) + “’”;
•This query can be exploited by calling up the web page executing it with the following URL:
http://example.com/app/accountView?id=’ or ‘1’=’1 causing the return of all the rows stored on the database table.

INJECTION PREVENTION

•Preventing SQL injections requires keeping data separate from commands and queries

–Separation of data from the web application logic.

–Implement settings and/or restrictions to limit data exposure in case of successful injection attacks.

Broken Authentication

•Broken authentication is process to steal a user's login data, or forge session data, such as cookies, to gain unauthorized access to websites.

•How do you prevent broken authentication vulnerabilities?

–Implement multi-factor authentication to prevent automated, credential stuffing, brute force, and stolen credential reuse attacks.

–Use a server-side, secure, built-in session manager that generates a new random session ID with high entropy after login

–Align password length, complexity and rotation policies with NIST

Sensitive Data Exposure

Two types of data:

•Stored data – data at rest (Credentials, Credit card numbers, Social Security Numbers, Medical information etc)

•Transmitted data – data that is transmitted internally between servers, or to web browsers.

•SSL is the acronym for Secure Sockets Layer. The standard security technology for establishing an encrypted link between a web server and a browser.

XML External Entities (XXE)

•XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.

•Vulnerable XML processors, Vulnerable code, Vulnerable dependencies, Vulnerable integrations

•How to prevent XML external entity attacks?

Use less complex data formats ,such as JSON, and avoid serialization of sensitive data.

Virtual patching

API security gateways

WebApplicationFirewalls(WAFs) to detect, monitor, and block XXE attacks

Broken Access Control

•Access unauthorized functionality and/or data

•View sensitive files

•Change access rights

Broken Access Control Prevention

•To avoid broken access control is to developand configure software with a security-first philosophy.

•Unique application business limit requirements should be enforced by domain models

Cross Site Scripting (XSS)

•XSS attacks consist of injecting malicious client-side scripts into a website and using the website as a propagation method.

•XSS is present in about two-thirds of all applications.

How to Prevent XSS Vulnerabilities

•Using frameworks that automatically escape XSS by design, such as Ruby on Rails, React JS

•Enabling a content security policy (CSP) is a defense-in-depth mitigating control against XSS.

Insecure Deserialization

One of the attack vectors presented regarding this security risk was a super cookie containing serialized information about the logged- in user.

The roles of the user was specified in this cookie.

How to Prevent Insecure Deserializations

•The best way to protect the web application from this type of risk is not to accept serialized objects from untrusted sources.

•Implementing integrity checks such as digital signatures on any serialized objects to prevent hostile object creation or data tampering

	Client-side Scripting	Server-side scripting
Facing	Frontend – Runs on the user's computer.	Backend – Runs on the ser
Purpose	Collection of user input, interfacing with the server.	Processes the user transactions.
Processes	Mostly deals with visual and user input.	Mostly deals with trans complex computations.
Code Transparency	Scripts are downloaded onto the client computer, which can be accessed by the users. Processes can be easily tampered with.	Scripts are not open Processes are transpare invisible to the users.
Security	Restricted to a sandbox, but generally less secure as users can see and mess with the scripts.	A lot more secure as user the source code, and t cannot interrupt the proces

Client-side Scripting : M

Client-side Uses

- Makes interactive web pages
- Make stuff work dynamically
- Interact with temporary storage
- Works as an interface between user and server
- Sends requests to the server
- Retrieval of data from Server
- Interact with local storage
- Provides remote access for client-server program

Client-side Languag

- JavaScript
- VBScript
- HTML (Structure)
- CSS (Designing)
- AJAX
- jQuery

Server-side Program

Server-side Uses

- It processes the user input
- Displays the requested pages
- Structure of web applications
- Interaction with servers/storages
- Interaction with databases
- Querying the database
- Encoding of data into HTML
- Operations over databases like delete, update.

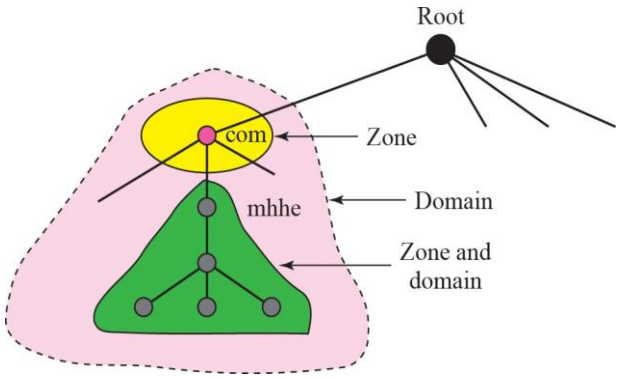
Server

- PHP
- ASP
- C++
- Java
- Pyth
- Ruby

• Aname space that maps each address to unique name can be organized in two ways:
Flat
Hierarchical

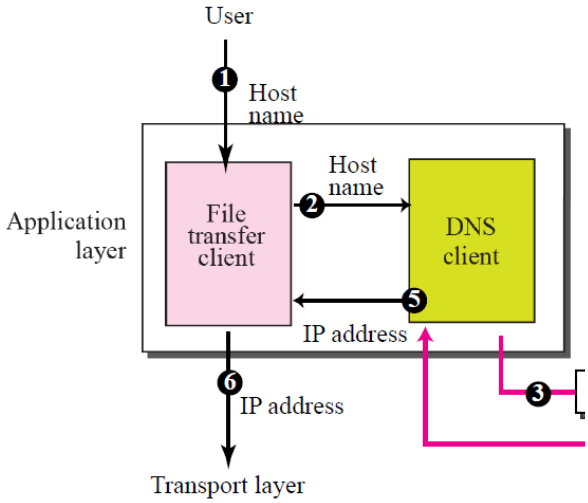
A **fully qualified domain name (FQDN),** sometimes also referred as an absolute domain name, is a domain name that specifies its exact location in the tree hierarchy of the Domain Name System (DNS).

A **partially qualified domain name (PQDN)** is a domain name that does not include all the levels between the host and the root node
A Partially Qualified Domain Name (PQDN) is used to specify a portion of a domain name, normally the host portion of it.
Example vtop. Usually the computers will add the DNS suffix along with Partially Qualified Domain Name (PQDN) before sending a DNS query for name resolution.



PURPOSE C

Zones and domains
DNS zone refers to a certain portion or administrative space within the global Domain Name System (DNS).
Each DNS zone represents a boundary of authority subject to management by certain entities.
The total of all DNS zones, which are organized in hierarchical tree-like order of cascading lower-level domains, form the DNS namespace.

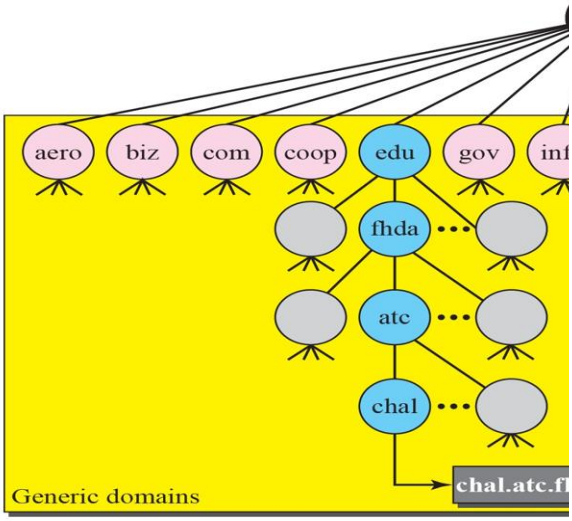


- Uniquely identifies
-
- Names instead of numeric addresses.

•In the Internet, the domain name space (tree) is divided into three different sections:

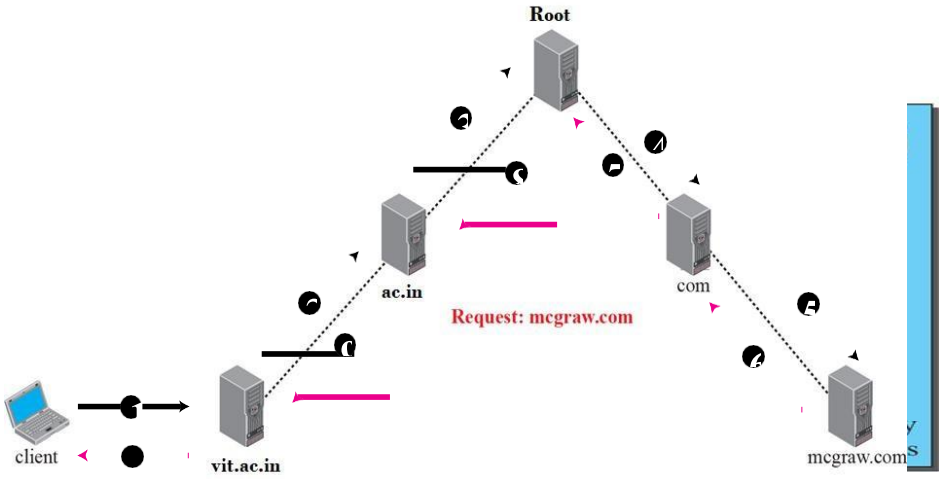
Generic domains

GENERI



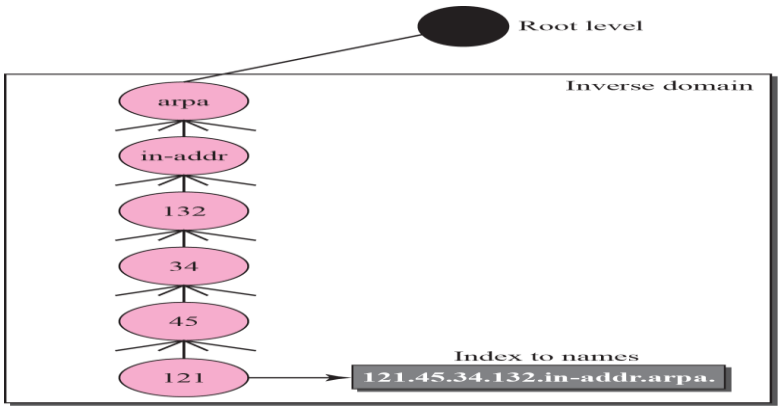
Label	*Description*
aero	Airlines and aerospace companies
biz	Businesses or firms (similar to "com")
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other non-profit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

Country domains



Iterative Resolution

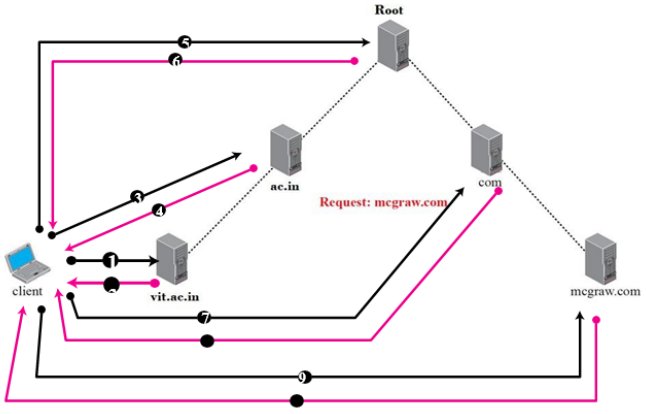
Inverse domain



DNS RESOLUTION

- Mapping a name to an address or an address to a name is called name-address resolution.

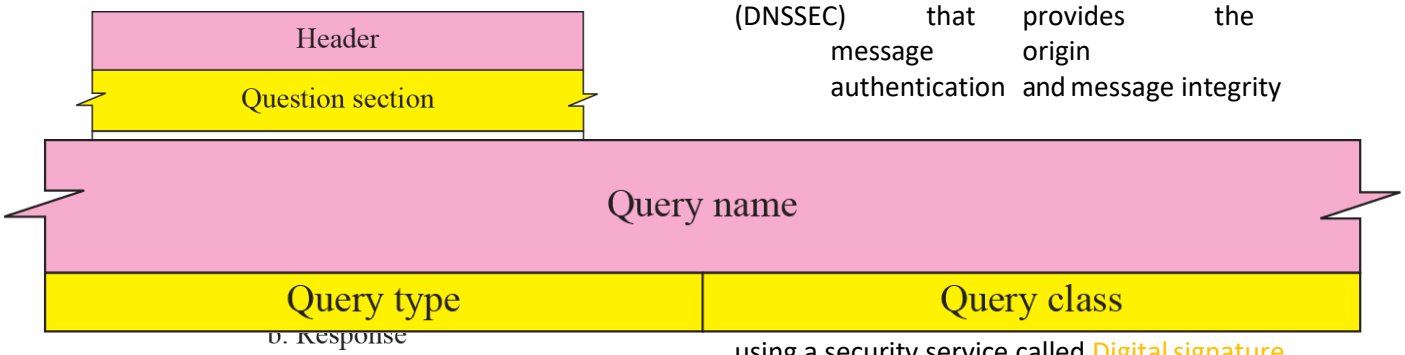
Resolver
> Mapping Names to Addresses
> Mapping Addresses to Names



DNS has two types of messages:
Query
Response.
Both types have the same format.

Recursive Resolution

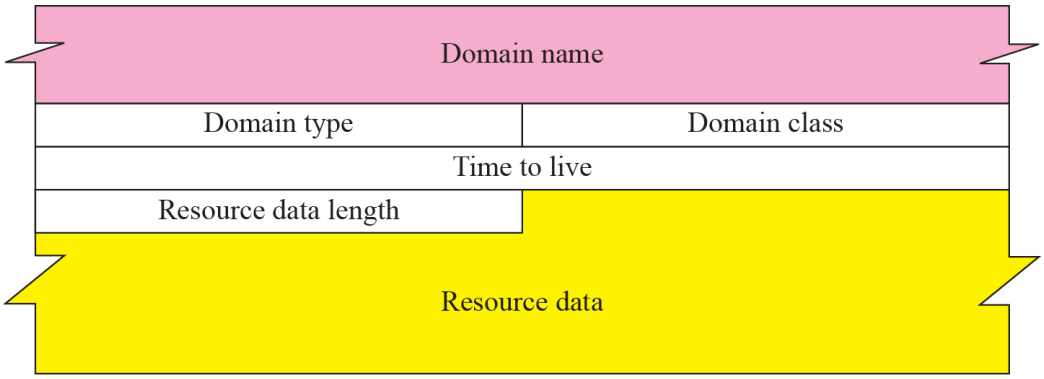
The query message consists of a header and question records.
The response message consists of a header, question records, answer records, authoritative records, and additional records.



b. Response

Question Record

Resource Record



In DNS, when there is a change, such as
- adding a new host
- removing a host
- changing an IP address

The change must be made to the DNS master file.
The DNSZmaster file must be updated dynamically.
The Dynamic Domain Name System (DDNS) therefore was devised to respond to this need.

SECURITY OF DNS

DNS is one of the most important systemsin the Internet infrastructure;

Applications such as Web access or e-mail are heavily dependent on the proper operation of DNS.
DNS can be attacked in several Ways.
To protect DNS, IETF has devised a technology named DNS Security (DNSSEC) that provides the message origin authentication and message integrity using a security service called Digital signature

ICANN - Internet Corporation for Assigned Names and Numbers

Module 2 .
WEB DESIGNING

	GET
BACK button/Reload	Harmless
Bookmarked	Can be bookmarked
Cached	Can be cached
Encoding type	application/x-www-form-urlencoded
History	Parameters remain in browser history
Restrictions on data length	Yes, when sending data, the GET method adds t data to the URL; and the length of a URL is limit (maximum URL length is 2048 characters)
Restrictions on data type	Only ASCII characters allowed
Security	GET is less secure compared to POST because d sent is part of the URL Never use GET when sending passwords or oth sensitive information!
Visibility	Data is visible to everyone in the URL

HTML Forms : Enctype

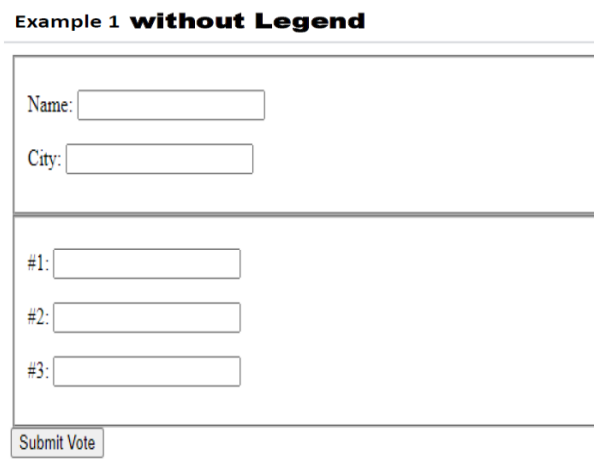
application/x-www-form-urlencodedencoding
•Thenameandvalueofeachdataitemisencoded usingthesameschemethatisusedtoencodeURLs .
•SpecialcharactersarereplacedwiththeirHTML entitycounterpart.
•Thenameofthedataitemandthevaluearesepar atedbytheequalssign(=)anddata/valuetuplesar eseparatedbytheampersandcharacter(&).
multipart/form-data
•Themultipart/form-dataencodingtendstobeusedonlyforuploadingf iles.
text/plain
•Themainstreambrowsersencodedataindiffer entwaysforthisencoding.
•GoogleChromeencodesdatainthesamewayas fortheapplication/x-www-form-urlencodedscheme,whereasFirefoxencodesth edataasfollows:
fave=xml
name=example.com
Eachdataitemisplacedonaline,andspecialchara ctersarenotencoded.

_blank The response is displayed in a new window or tab
_self The response is displayed in the same frame (this is default)
_parent The response is displayed in the parent frame
_top The response is displayed in the full body of the window

framename The response is displayed in a named iframe

Auto On –Off
on - Default. The browser will automatically complete values based on values that the user has entered before
off - The user must enter a value into each field for every use. The browser does not automatically complete entries

```
<fieldset>
<legend>Enter Your Details</legend>
<p>
<label for="name">Name: <input id="name"
name="name" /></label>
</p>
<p>
<label for="name">City: <input id="city"
name="city" /></label>
</p>
</fieldset>
<fieldset>
<legend>Vote For Your Three
FavoriteFruits</legend>
<p>
<label for="fave1">#1: <input id="fave1"
name="fave1" /></label>
</p>
<p>
<label for="fave2">#2: <input id="fave2"
name="fave2" /></label>
</p>
<p>
<label for="fave3">#3: <input id="fave3"
name="fave3" /></label>
</p>
</fieldset>
```



<buttontype="button">
DoNOTpressthisbutton
</button>

form action="" method="post">
<button type="submit">Submit</button>
<button
type="button"><imgsrc="http://www.exam
ple.com/style/download.png" alt="submit"
/></button>

dirname-Controlstextdirectionality.
list-
Specifiestheidofadatalistelementthatprovides valuesforthiselement.
maxlength-
Specifiesthemaximumnumberofcharacterstha ttheusercanenterintothetextbox.
pattern-
Specifiesaregularexpressionpatternforthepurp osesofinputvalidation.
placeholder-
Specifiesahinttotheuserastothekindofinputtha tyouexpect.
readonly-
Ifpresent,thisattributemakesthetextboxread-only.
required-
Specifiesthattheusermustenteravalueforthepu rposesofinputvalidation.
size-
Specifiesthewidthoftheelementinthenumbero fcharactersthatarevisibleinthetextbox.
value-Specifies the initial value for the text box.

Restrict Data Entr

HTML5 introduces new values for the input element's type attribute and

- **checkbox -** Restricts the input to a true/false check box.
- **color -** Restricts the input to a color.
- **date** - Restricts the input to a date.
- **datetime** - Restricts the input to a global date and time with time zone.
- **datetime-local** - Restricts the input to a global date and time without time zone.
- **email** - Restricts the input to a properly formatted e-mail address.
- **month** - Restricts the input to a year and month.
- **number** or floati
- **radiobut** fixed set
- **range** - R range.
- **tel** - Res formatte
- **time** - R
- **week** - R week.
- **url** - Res URL.

Iframes Attributes

An iframe with a name attribute value of myframe is created. This creates a browsing context called myframe.
• Then this browsing context is used in the target attribute of other elements-specifically, a, form, button, input, and base.
• a element to create a pair of hyperlinks which will load the URLs specified in their href attributes into the iframe.
• The width and height attributes specify the size in pixels. The src attribute specifies a URL that should be loaded and displayed in the iframe initially.
• The srcdoc attribute allows you to define an HTML document to display inline.
• The seamless attribute sets the browser to display the iframe as if they were an integral part of the main HTML document.
• Otherwise a border is applied by default and a scrollbar is present if the content is larger than the width and height attributes.

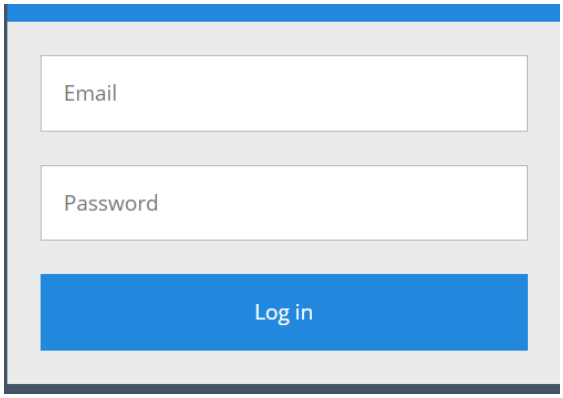
<iframe src="example.htm" height="200"
width="300" title="Iframe
Example"></iframe>

<iframe src=" example.htm "
style="height:200px;width:300px;"
title="Iframe Example"></iframe>

To remove the border, add the style attribute and use the CSS border property:

Syntax:
<iframe src="example.htm" style="border: none;" title="Iframe Example"></iframe>
<object data="../your.mp3" width="200px" height="50px"></object>
<object data="../your.ogg" width="200px" height="50px"></object>
<video width="320" height="240" autoplay>
<source src="movie.mp4" type="video/mp4">
Your browser does not support the video tag.
</video>
<Image>
• The image is inserted using the <img> tag. The only difference from other images is that you must add a usemap attribute
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

1) Login Form using Html/ CSS



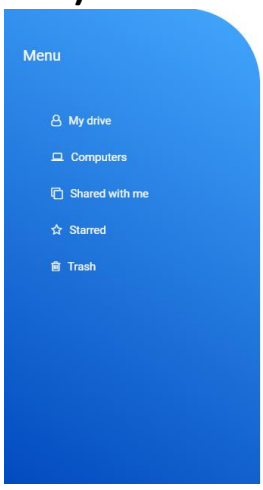

```
<div class="login">
  <div class="login-triangle"></div>

  <h2 class="login-header">Log in</h2>

  <form class="login-container">
    <p><input type="email" placeholder="Email"></p>
    <p><input type="password" placeholder="Password"></p>
    <p><input type="submit" value="Log in"></p>
  </form>
</div>
```

CSS

```
/* 'Open Sans' font from Google Fonts */
@import url(https://fonts.googleapis.com/css?family=Open+Sans:400,700);

body {
  background: #456;
  font-family: 'Open Sans', sans-serif;
}

.login {
  width: 400px;
  margin: 16px auto;
  font-size: 16px;
}

/* Reset top and bottom margins from certain elements */
.login-header,
.login p {
  margin-top: 0;
  margin-bottom: 0;
}

/* The triangle form is achieved by a CSS hack */
.login-triangle {
  width: 0;
  margin-right: auto;
  margin-left: auto;
  border: 12px solid transparent;
  border-bottom-color: #28d;
}
.login-header {
  background: #28d;
  padding: 20px;
  font-size: 1.4em;
  font-weight: normal;
  text-align: center;
  text-transform: uppercase;
  color: #fff;
}
.login-container {
  background: #ebebeb;
  padding: 12px;
}
/* Every row inside .login-container is defined with p tags */
.login p {
  padding: 12px;
}
.login input {
  box-sizing: border-box;
  display: block;
  width: 100%;
  border-width: 1px;
  border-style: solid;
  padding: 16px;
  outline: 0;
```

```
 font-family: inherit;
 font-size: 0.95em;
}

.login input[type="email"],
.login input[type="password"] {
 background: #fff;
 border-color: #bbb;
 color: #555;
}

/* Text fields' focus effect */
.login input[type="email"]:focus,
.login input[type="password"]:focus {
 border-color: #888;
}
.login input[type="submit"] {
 background: #28d;
 border-color: transparent;
 color: #fff;
 cursor: pointer;
}
.login input[type="submit"]:hover {
 background: #17c;
}
/* Buttons' focus effect */
.login input[type="submit"]:focus {
 border-color: #05a;
}
```

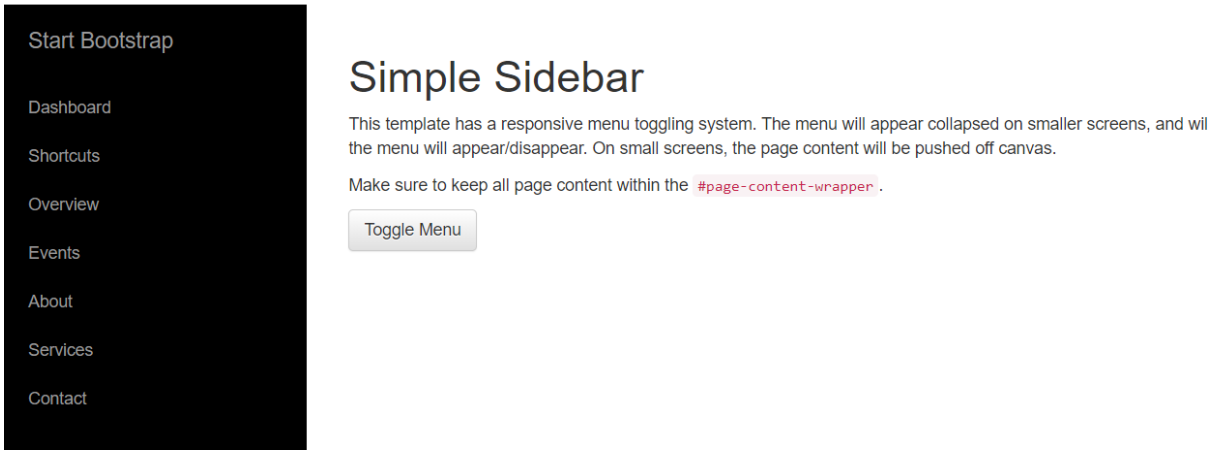
2) Side Bar




```
<aside>
 <p> Menu </p>
 <a href="javascript:void(0)">
  <i class="fa fa-user-o" aria-hidden="true"></i>
  My drive
 </a>
 <a href="javascript:void(0)">
  <i class="fa fa-laptop" aria-hidden="true"></i>
  Computers
 </a>
</aside>


aside {
 color: #fff;
 width: 250px;
 padding-left: 20px;
 height: 100vh;
 background-image: linear-gradient(30deg ,
#0048bd, #44a7fd);
 border-top-right-radius: 80px;
}

aside a {
 font-size: 12px;
 color: #fff;
 display: block;
 padding: 12px;
 padding-left: 30px;
 text-decoration: none;
 -webkit-tap-highlight-color:transparent;
}

aside a:hover {
 color: #3f5efb;
 background: #fff;
 outline: none;
 position: relative;
 background-color: #fff;
 border-top-left-radius: 20px;
 border-bottom-left-radius: 20px;
}

aside a i {
 margin-right: 5px;
}

aside a:hover::after {
 content: "";
 position: absolute;
 background-color: transparent;
 bottom: 100%;
 right: 0;
 height: 35px;
 width: 35px;
 border-bottom-right-radius: 18px;
 box-shadow: 0 20px 0 0 #fff;
}

aside a:hover::before {
 content: "";
 position: absolute;
 background-color: transparent;
 top: 38px;
 right: 0;
 height: 35px;
 width: 35px;
```

```
  border-top-right-radius: 18px;
  box-shadow: 0 -20px 0 0 #fff;
}

aside p {
  margin: 0;
  padding: 40px 0;
}
```

```
body {
  font-family: 'Roboto';
  width: 100%;
  height: 100vh;
  margin: 0;
}
```




```
<!DOCTYPE html>
<html lang="en">
<head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport"
content="width=device-width, initial-
scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Simple Sidebar - Start Bootstrap
Template</title>

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap
/3.3.4/css/bootstrap.min.css">

    <!-- Optional theme -->
    <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap
/3.3.4/css/bootstrap-theme.min.css">

    <!-- jQuery -->
    <script
src="//ajax.googleapis.com/ajax/libs/jquery/2
.1.3/jquery.min.js"></script>

    <!-- Latest compiled and minified JavaScript
-->
    <script
src="//maxcdn.bootstrapcdn.com/bootstrap/
3.3.4/js/bootstrap.min.js"></script>

    <!-- Custom CSS -->
    <link href="sidebar.css" rel="stylesheet">

    <!-- HTML5 Shim and Respond.js IE8
support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if
you view the page via file:// -->
    <!--[if lt IE 9]>
        <script
src="https://oss.maxcdn.com/libs/html5shiv/
3.7.0/html5shiv.js"></script>
        <script
src="https://oss.maxcdn.com/libs/respond.js/
1.4.2/respond.min.js"></script>
    <![endif]-->

    <style media="screen">
```

```
    /*body { padding-top: 70px; }*/
    #connectLogo {
      height: 60px;
      padding: 15px 0 5px 0;
    }
    #logo {
      height: 60px;
      padding: 5px 0 5px 20px;
    }

    .share-link {
      line-height: 60px;
      padding: 0 1em;
      font-size: 2em;
    }
   </style>

</head>

<body>

   <nav class="navbar navbar-default
NOnavbar-fixed-top">
     <div class="container-fluid">
      <div class="navbar-header">

        <button type="button" class="navbar-
toggle collapsed menu-toggle">
          <span class="sr-only">Toggle
navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>



        <a class="NOnavbar-brand" href="#">
          <img id="logo"
src="https://upload.wikimedia.org/wikipedia/
en/a/a4/Flag_of_the_United_States.svg"
alt="Talk Fusion">
        </a>

      </div>

      <div class="collapse navbar-collapse"
id="bs-example-navbar-collapse-1">

        <span class="glyphicon glyphicon-share
hidden-xs pull-right share-link" aria-
hidden="true"></span>

        <!-- Collect the nav links, forms, and
other content for toggling -->
<!--          <img class="center-block"
id="connectLogo"
src="https://upload.wikimedia.org/wikipedia/
en/a/a4/Flag_of_the_United_States.svg"
alt=""> -->
        <!-- <div id="btnShare" style="display:
none;"><img src="share.svg" width="22"
alt="share"></div> -->
      </div>


<!--
      <button type="button" class="share-link
hidden-xs pull-right">
        <span class="sr-only">Share link</span>
        <span class="glyphicon glyphicon-
link"></span>
      </button> -->

    </div>
   </nav>

   <div id="wrapper">

     <!-- Sidebar -->
     <div id="sidebar-wrapper">
       <ul class="sidebar-nav">
         <li class="sidebar-brand">
           <a href="#">
             Start Bootstrap
           </a>
         </li>
         <li>
           <a href="#">Dashboard</a>
         </li>
         <li>
           <a href="#">Shortcuts</a>
         </li>
         <li>
           <a href="#">Overview</a>
         </li>
         <li>
           <a href="#">Events</a>
         </li>
         <li>
           <a href="#">About</a>
         </li>
         <li>
```

```
            <a href="#">Services</a>
          </li>
          <li>
            <a href="#">Contact</a>
          </li>
        </ul>
      </div>
      <!-- /#sidebar-wrapper -->

      <!-- Page Content -->
      <div id="page-content-wrapper">
        <div class="container-fluid">
          <div class="row">
            <div class="col-lg-12">
              <h1>Simple Sidebar</h1>
              <p>This template has a
responsive menu toggling system. The menu
will appear collapsed on smaller screens, and
will appear non-collapsed on larger screens.
When toggled using the button below, the
menu will appear/disappear. On small
screens, the page content will be pushed off
canvas.</p>
              <p>Make sure to keep all page
content within the <code>#page-content-
wrapper</code>.</p>


#wrapper {
  padding-left: 0;
  -webkit-transition: all 0.5s ease;
  -moz-transition: all 0.5s ease;
  -o-transition: all 0.5s ease;
  transition: all 0.5s ease;
}

#wrapper.toggled {
  padding-left: 250px;
}

#sidebar-wrapper {
  z-index: 1000;
  position: fixed;
  left: 250px;
  width: 0;
  height: 100%;
  margin-left: -250px;
  overflow-y: auto;
  background: #000;
  -webkit-transition: all 0.5s ease;
  -moz-transition: all 0.5s ease;
```

```
              <a href="#menu-toggle"
class="btn btn-default menu-toggle">Toggle
Menu</a>
            </div>
          </div>
        </div>
      </div>
      <!-- /#page-content-wrapper -->

    </div>
    <!-- /#wrapper -->

    <!-- Menu Toggle Script -->
    <script>
    $(".menu-toggle").click(function(e) {
      e.preventDefault();
      $("#wrapper").toggleClass("toggled");
    });
    </script>

</body>

</html>
```

```
    -o-transition: all 0.5s ease;
    transition: all 0.5s ease;
}

#wrapper.toggled #sidebar-wrapper {
    width: 250px;
}

#page-content-wrapper {
    width: 100%;
    position: absolute;
    padding: 15px;
}

#wrapper.toggled #page-content-wrapper {
    position: absolute;
    margin-right: -250px;
}

/* Sidebar Styles */

.sidebar-nav {
    position: absolute;
    top: 0;
    width: 250px;
    margin: 0;
    padding: 0;
    list-style: none;
}

.sidebar-nav li {
    text-indent: 20px;
    line-height: 40px;
}

.sidebar-nav li a {
    display: block;
    text-decoration: none;
    color: #999999;
}

.sidebar-nav li a:hover {
    text-decoration: none;
    color: #fff;
    background: rgba(255,255,255,0.2);
}

.sidebar-nav li a:active,
.sidebar-nav li a:focus {
    text-decoration: none;
}
```

```
.sidebar-nav > .sidebar-brand {
    height: 65px;
    font-size: 18px;
    line-height: 60px;
}

.sidebar-nav > .sidebar-brand a {
    color: #999999;
}

.sidebar-nav > .sidebar-brand a:hover {
    color: #fff;
    background: none;
}

@media(min-width:768px) {
    #wrapper {
        padding-left: 250px;
    }

    #wrapper.toggled {
        padding-left: 0;
    }

    #sidebar-wrapper {
        width: 250px;
    }

    #wrapper.toggled #sidebar-wrapper {
        width: 0;
    }

    #page-content-wrapper {
        padding: 20px;
        position: relative;
    }

    #wrapper.toggled #page-content-wrapper {
        position: relative;
        margin-right: 0;
    }
}
```


```
<html>
<head>
<title>CAT 1 form</title>
<style>
#d2
{
  padding: 15px;
  background: #379cd1;
            margin:-5px;
  border-radius:15px;
  width:300px;
}
#d1
{
margin:auto;
width:320px;
  border: 10px solid black;
  background-clip: border-box;
}
progress
{
width:100%;
}
label
{
display:inline-block;
margin-top:10px;
font-weight:bold}
</style>
</head>
<body>
<form action="#">
<div id="d1">
<div id="d2">
<h1>Progress farm</h1>
<progress value="65" max="100">
</progress><br><br>
<label>Name on Card</label><br>
<input type="text" style="width:100%;"><br>
<label>Street Address</label>
<input type="text" style="width:100%;"><br>
<input type="text"
style="width:100%;display:inline-
block;margin-top:5px;"><br>
<label>zip</label><br>
<input type="text" style="width:50%;"><br>
<label>Credit Card Number</label><br>
<input type="text" style="width:100%;"><br>
<label>Security Code on Card
(CVV)</label><br>
<input type="text" style="width:30%;"><br>
<input type="submit" value="Subscribe"
style="display:inline-
block;padding:15px;width:100%;background-
color:black;color:white;border-
radius:5px;margin-top:5px;">
</div>
</div>
</form>
</body>
</html>
```