CHAPTER 1

Front End

1.1 HTML

HTML is the language in which most websites are written. HTML is used to create web pages and make them functional. HTML was first created by **Tim Berners-Lee, Robert Cailliau**, and others starting in 1989. It stands for **Hyper Text Markup Language**. Hypertext means that the document contains links that allow the reader to jump to other places in the document or to another document altogether. The latest version is known as **HTML5**. A Markup Language is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: **tags and attributes.**

Figure 1.1 Intoduction to HTML

Different types of **Code-Edit**or can be used to write our HTML Code. One of the basic editor is Notepad.
However for easily Customization, Plugins, better preview, Language supports. There are many other choices available for developers:

1. Sublime Text 3
2. Notepad ++
3. Komodo Edit
4. Visual Studio code

1.2 CSS

CSS stands for **Cascading Style Sheets** and it is the language used to style the visual presentation of web pages. CSS is the language that tells web browsers how to render the different parts of a web page.

CSS syntax includes selectors, properties, values, rulesets, and statements by which we design the appearance of a **Webpage**. That includes colors, font styling, positioning of element, animations, transitions, timing-functions and many more. identifiers should be added to elements on the webpage. These identifiers, often called **Classes** and **IDs** in the context of CSS, make it easier to identify the items that should be affected by the CSS rules.

Types of CSS.

- Inline styles
- Internal stylesheets
- External stylesheets

1.3 JavaScript

JavaScript is perhaps the most important web development language. Every website uses some amount of **JavaScript** for "**front-end**" functionality. Moreover, the evolution of the language itself and the standardization of browser technology has given rise to in-browser web apps powered almost entirely on JavaScript. **Web applications** are usually coded in browser-supported language such as JavaScript and HTML as these languages rely on the browser to render the program executable. Some applications are dynamic, requiring server-side processing. Others are completely static with no processing required at the server. Additionally, JavaScript is no longer just a front-end language.

JavaScript makes it possible to build rich UI components such as image *sliders, pop-ups, site navigation mega menus, form validations, tabs, accordions, and much more*. JavaScript includes various **Libraries** and **frameworks**. JavaScript frameworks are collections of JavaScript code libraries that provide developers with pre-written code to use for programming features and tasks.

JavaScript front-end frameworks includes:

- React
- React Native
- AngularJS
- Vue.
- jQuery
- Next.js
- Express
- Bootstrap

Chapter 2

Back End

2.1 History of Back End.

JavaScript was invented by **Brendan Eich** in 1995. It was developed for **Netscape 2,** and became the **ECMA-262** standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser.

Internet Explorer (IE4) was the first browser to support **ECMA-262** Edition 1 (ES1).

Table 2.1 Releases of Javascript

Year	**ECMA**	**Browser**
1995		JavaScript was invented by Brendan Eich
1996		Netscape 2 was released with JavaScript 1.0
1997		JavaScript became an ECMA standard (ECMA-262)
1997	ES1	ECMAScript 1 was released
1997	ES1	IE 4 was the first browser to support ES1
1998	ES2	ECMAScript 2 was released
1998		Netscape 42 was released with JavaScript 1.3
1999	ES2	IE 5 was the first browser to support ES2
1999	ES3	ECMAScript 3 was released
2000	ES3	IE 5.5 was the first browser to support ES3
2000		Netscape 62 was released with JavaScript 1.5
2000		Firefox 1 was released with JavaScript 1.5
2008	ES4	ECMAScript 4 was abandoned
2009	ES5	ECMAScript 5 was released
2011	ES5	IE 9 was the first browser to support ES5 *
2011	ES5	Firefox 4 was released with JavaScript 1.8.5
2012	ES5	Full support for ES5 in Safari 6
2012	ES5	Full support for ES5 in IE 10
2012	ES5	Full support for ES5 in Chrome 23
2013	ES5	Full support for ES5 in Firefox 21

2013	ES5	Full support for ES5 in Opera 15
2014	ES5	Full support for ES5 in all browsers
2015	ES6	ECMAScript 6 was released
2016	ES6	Full support for ES6 in Chrome 51
2016	ES6	Full support for ES6 in Opera 38
2016	ES6	Full support for ES6 in Edge 14
2016	ES6	Full support for ES6 in Safari 10
2015	ES6	Full support for ES6 in Firefox 52
2018	ES6	Full support for ES6 in all browsers.

There have been a number of subsequent version releases since then. Today JavaScript has crossed more than **25 years** and it has more than **1,444 ,231 libraries**.

2.2 What is Back End

The backend is the portion of the website you don't see. Backend not only deals with the appearance of the webpage but also responsible for efficiency, speed, scalability, Data storing and security, Data management, and ensuring everything on the **client-side** actually works. The backend communicates with the frontend, sending and receiving information to be displayed as a webpage.

Figure.2.1 *Introduction to Back End*.

JavaScript is basically perhaps the most important web development language. JavaScript is no longer just a front-end language. Thanks to the **Node.js** project, it is now possible to easily use JavaScript to build applications **server-sid**e as well. And with the rise of technologies such as **MongoDB** that use **JSON** as a data serialization language, it is now possible to use JavaScript for an entire web application, from data storage to server-side processing to front-end UI.

Whatever other programming languages you use or are interested in learning, it is increasingly clear that JavaScript is the ***sine qua non*****(essential)** of modern Web development.

Most popular languages for Backend:

- JavaScript
- C#
- Golang
- Java
- PHP
- Python
- Ruby
- SQL

2.3 Backend Frameworks

A backend or server-side framework is the library of tools and modules that **helps in building the architecture of a website**. It significantly impacts a **web app's performance** and can decide your **project's success**. Backend frameworks mainly help make tasks convenient and easy for developers. A Framework's main objective is to provide high performance, fast development, and flexibility for your web development projects.

Most Popular Backend frameworks are

- **Django** for Python Developers
- **ExpressJS** for JavaScript Developers
- **Laravel** for PHP Programmers
- **Ruby** on Rails for Ruby Programmers
- **ASP**.**NET** core for .NET developers
- **CakePHP** Framework for PHP Developers
- **Flask** Framework for Python Developers

Web Server

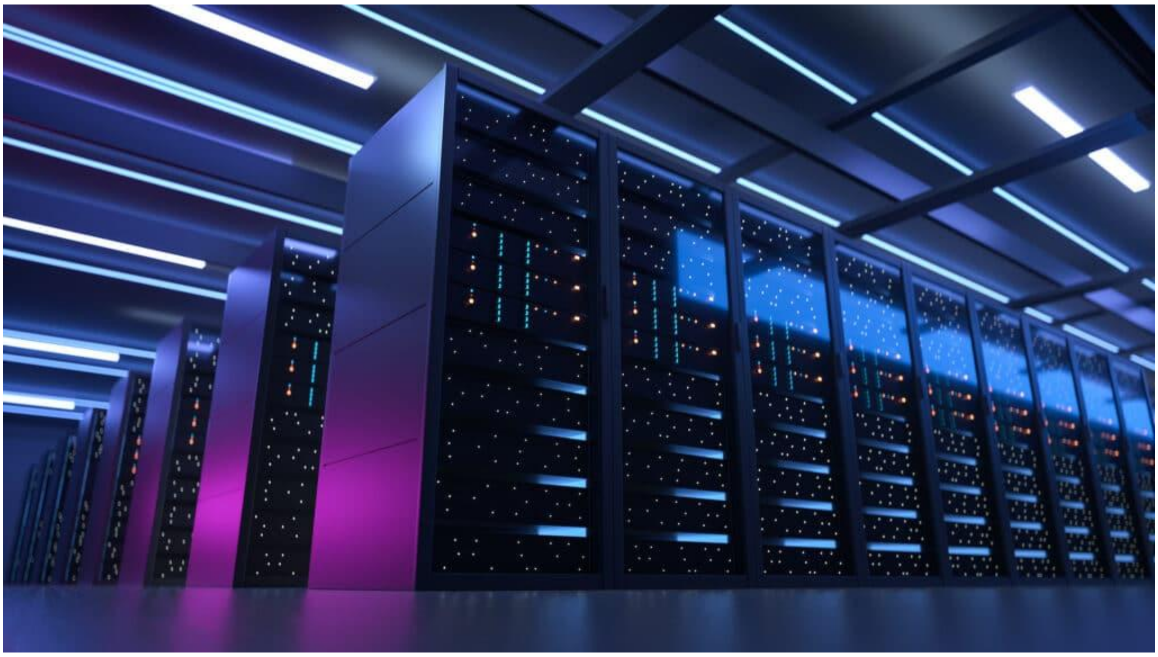

Figure.3.1 *Introduction to Web Servers.*

3.1 What is a Web Server ?

A webpage is a document made by combining the files of **HTML, CSS , JavaScript** or any other language that we use. It can be viewed on any **web browser**. All of the files were contained within the **web server**, which can be accessed by entering the **URL** for that web page, and once it is loaded, it appears on the user's web browser. Each webpage is linked with a unique URL; hence two pages cannot have the same URL. A webpage may contain ***text, links for other pages, graphics, videos, etc***. Moreover, it is mainly used to provide information to the user in text, images, etc.

A webpage may be a part of a ***website****; it means a website contains different web pages.* It can be understood as an example of a book. So, a **website** is like a complete book, and a webpage is like a page of that book. *The WWW or Internet contains millions of web pages*, and daily, a lot is being added. **Tim Berners-Lee developed the first webpage.**

3.2 How Web Servers Works.

A web server is software and hardware that uses **HTTP (Hypertext Transfer Protocol)** and other protocols to respond to client requests made over the **World Wide Web**. The main job of a web server is to display website content through storing, processing and delivering webpages to users. Web server hardware is connected to the internet and allows data to be exchanged with other connected devices.

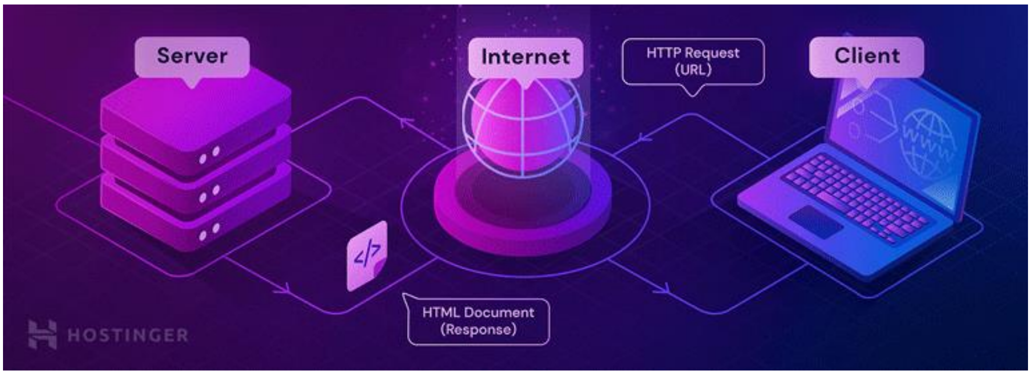

Figure- 3.2 How Web Server Works

When a web browser, like Google Chrome or Firefox, makes a request of a file(by URL) that's hosted on a web server, the browser will request the file by **HTTP**. When the request is received by the web server, the HTTP server will accept the request, find the content and send it back to the browser through HTTP. If the requested web page is not found, web server will the send an HTTP response: **Error 404 Not found**. This process includes a series of steps.

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.

3.3 List of web servers available today are:

- **Apache HTTP Server**. Developed by Apache Software Foundation, it is a free and open-source web server for Windows, Mac OS X, Unix, Linux, Solaris and other operating systems; it needs the Apache license.
- **Microsoft Internet Information Services (IIS).** Developed by Microsoft for Microsoft platforms; it is not open sourced, but widely used.
- **Nginx.** A popular open-source web server for administrators because of its light resource utilization and scalability. It can handle many concurrent sessions due to its event-driven architecture. Nginx also can be used as a proxy server and load balancer.
- **Lighttpd.** A free web server that comes with the FreeBSD operating system. It is seen as fast and secure, while consuming less CPU power.

- **Sun Java System Web Server.** A free web server from Sun Microsystems that can run on Windows, Linux and Unix. It is well-equipped to handle medium to large websites

3.4 Cloud Servers

A cloud server is powerful **Physical** or **Virtual Infrastructure** that performs application- and information-processing storage. Cloud servers are created using virtualization software to divide a physical (bare metal) server into multiple virtual servers. Organizations use an infrastructure-as-a-service (IaaS) model to process workloads and store information**.** They can access virtual server functions remotely through an online interface

Figure 3.3 Cloud Servers and On Premise Servers.

Table- 3.1 Difference between Cloud and On Premise Server

Cloud Server	On Premise (Physical Server)
Cloud server is deliver and hosted over the Internet.	On Premise software is managed and hosted locally on a device
No installations or increased costs for additional equipment of infrastructure.	Large initial cash required for equipment and replacement of outdated equipment.
The vendor handles updates and product changes and notifies you of changes.	Usually limited to the solution features the product came with at the time of purchase.
Accessible for Broad range of devices through mobile and web applications.	Access generally limited to the device(s) the software was originally installed on.

Chapter-4

Web Security Considerations

4.1 Intro

Web Security is very important nowadays. Websites are always prone to security threats/risks. Web Security deals with the security of data over the internet/network or web or while it is being transferred to the internet. For e.g. when you are transferring data between client and server and you have to protect that data that security of data is your web security.

Hacking a Website may result in the theft of Important Customer Data, it may be the credit card information or the login details of a customer, while somebody hacks your website they can either steal the important information of the customers or they can even propagate the illegal content to users through your website so, therefore, security considerations are needed in the context of web security.

4.2 Security Threats:

A **Threat** is nothing but a possible event that can damage and harm an information system. can potentially harm Computer systems & organizations. Whenever an Individual or an Organization creates a website, they are vulnerable to security attacks. Security attacks are mainly aimed at stealing altering or destroying a piece of personal and confidential information, stealing the hard drive space, and illegally accessing passwords.

Web security threats are constantly emerging and evolving, but many threats consistently appear at the top of the list of web security threats. These include:

- Cross-site scripting (XSS)
- SQL Injection
- Phishing
- Ransomware
- Code Injection
- Viruses and worms
- Spyware
- Data Breaches

Figure-4.1 Data Security.

4.1 Steps to prevent Threats

- Threat prevention strategies.
  - Reduce the attack surface
  - Prevent known threats
  - Prevent unknown threats
- Zero-trust approach
  - Identify and classify sensitive data
  - Map the data flow
  - Architect the network
  - Continuous monitoring
- Assume breach approach
  - Red-team exercise
  - Continuous monitoring

Chapter – 5

Conclusion

In today's Web development is abruptly rising and a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. This includes colour contrast, text organization, font selection, style of a page, page size, graphics used, and consistency. In order to create a well-designed page for a specific audience. We need to know about the new Technologies that are continuously coming in the market. For making a good website one must have a team or have knowledge of HTML, CSS, JavaScript, Frameworks of Front-end and Back-end. Along with Knowledge of databases to work on public data.

Along with the Appearance and functionality the speed, optimization, cross-platform compatibility & security of website and databases is also a very crucial part we didn’t ignore it.

The rise of **WebApps** gives a new turn to web development. It provides new opportunities for the developers and convenience to the users.

References

1. https://html.com/css/#What_is_CSS
2. https://www.geeksforgeeks.org/how-security-system-should-evolve-to-handle-cyber-security-threats-and-vulnerabilities/?ref=rp
3. https://developer.mozilla.org/enUS/docs/Learn/Common_questions/What_is_a_web_server
4. https://en.wikipedia.org/wiki/Web_development