Full Stack Web Development Virtual Internship

An Internship report submitted to

Jawaharlal Nehru Technological University Anantapur, Anantapuramu

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Submitted by

Sane Navya

21121A3360
IV B. Tech I Semester

Under the esteemed supervision of

Ms. K. Jaya Deepthi
Assistant Professor, Dept of CSE, SVEC

Department of Computer Science and Engineering

SREE
VIDYANIKETHAN
Engineering College (Autonomous)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)
(Affiliated to JNTUA, Anantapuramu and approved by AICTE, New Delhi) Accredited by NAAC with A Grade
Sree Sainath Nagar, Tirupati, Chittoor Dist. -517 102, A.P, INDIA
2024 - 2025.

SREE VIDYANIKETHAN ENGINEERING COLLEGE

(AUTONOMOUS)

Sree Sainath Nagar, A. Rangampet

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

*This is to certify that the internship report entitled **"AICTE EduSkills Internship"** is the bonafide work done by **Sane Navya** (Roll No:21121A3360) in the Department of **Computer Science and Engineering**, and submitted to Jawaharlal Nehru Technological University Anantapur, Anantapuramu in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science** during the academic year 2024-2025*

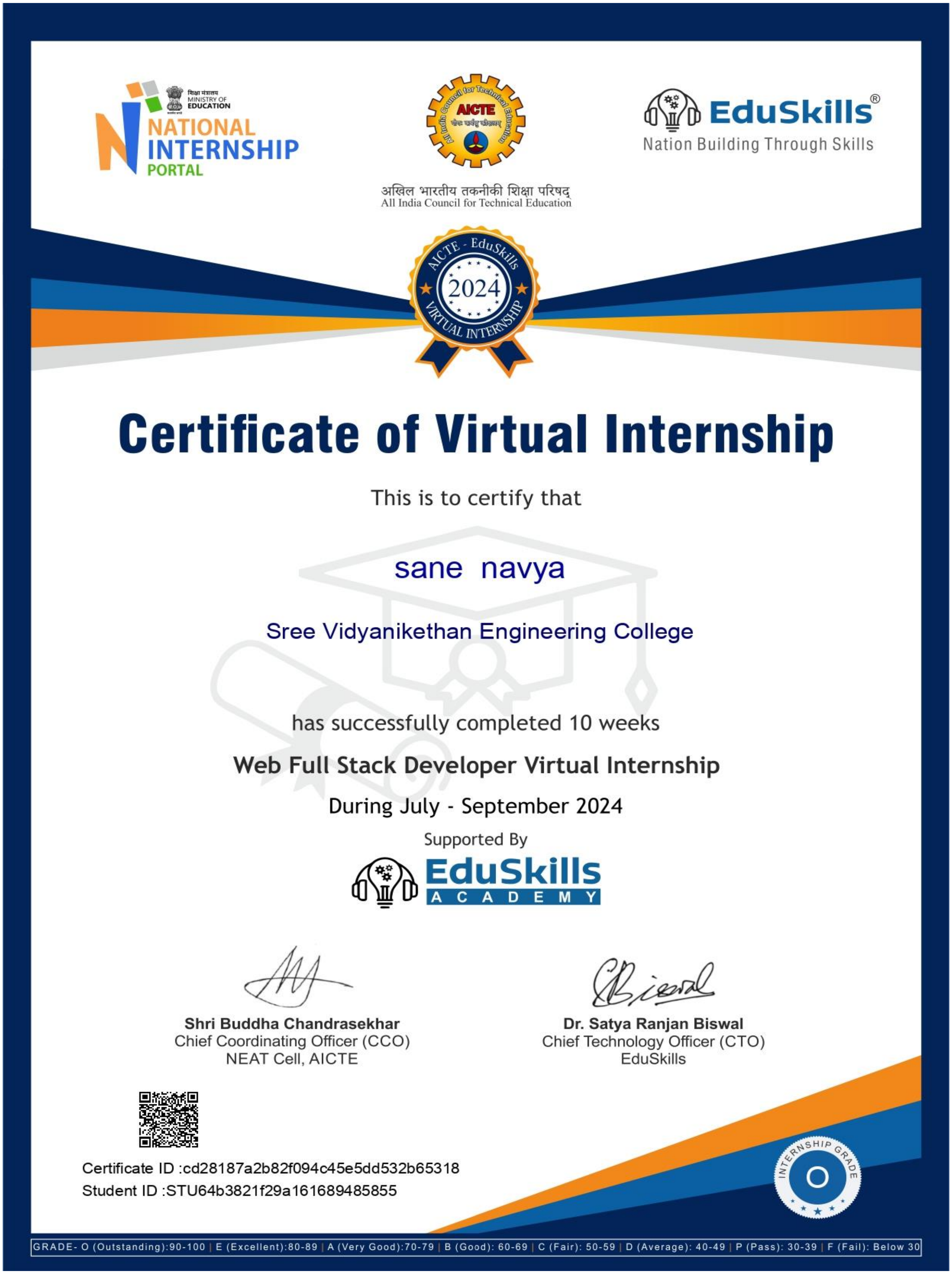
Head:

Dr. B. Narendra Kumar Rao
Professor & Head
Dept. of CSE

`

INTERNAL EXAMINER	**EXTERNAL EXAMINER**
K. Jaya Deepthi	**M. Chiranjeevi**

COMPLETION CERTIFICATE FROM COMPANY



शिक्षा मंत्रालय MINISTRY OF EDUCATION

NATIONAL INTERNSHIP PORTAL

अखिल भारतीय तकनीकी शिक्षा परिषद्
All India Council for Technical Education

EduSkills®
Nation Building Through Skills

AICTE - EduSkills 2024 VIRTUAL INTERNSHIP

Certificate of Virtual Internship

This is to certify that

sane navya

Sree Vidyanikethan Engineering College

has successfully completed 10 weeks

Web Full Stack Developer Virtual Internship

During July - September 2024

Supported By

EduSkills ACADEMY

Shri Buddha Chandrasekhar
Chief Coordinating Officer (CCO)
NEAT Cell, AICTE

Dr. Satya Ranjan Biswal
Chief Technology Officer (CTO)
EduSkills

Certificate ID :cd28187a2b82f094c45e5dd532b65318
Student ID :STU64b3821f29a161689485855

INTERNSHIP GRADE O

GRADE- O (Outstanding):90-100 | E (Excellent):80-89 | A (Very Good):70-79 | B (Good): 60-69 | C (Fair): 50-59 | D (Average): 40-49 | P (Pass): 30-39 | F (Fail): Below 30

ABSTRACT

The Full Stack Web Development internship with AICTE-Eduskills provided an intensive and hands-on learning experience, designed to equip participants with essential skills in both front-end and back-end development. Throughout the program, I engaged with a wide range of web technologies, including HTML, CSS, JavaScript, React, Node.js, and databases such as MongoDB and MySQL. This exposure enabled me to build comprehensive, scalable, and user-centric web applications, while gaining proficiency in developing responsive interfaces and implementing efficient server-side functionality.

In addition to technical skills, the internship emphasized collaboration, version control, and agile development practices. Working in teams under the guidance of experienced mentors, I learned how to structure, test, and debug code effectively while maintaining industry standards. This holistic training not only enhanced my technical capabilities but also developed my problem-solving and project management skills, preparing me for real-world scenarios in web development.

In conclusion, the internship provided a robust foundation in full stack development, bridging the gap between academic knowledge and practical application. This experience has prepared me to tackle complex web projects, collaborate effectively in team settings, and continue advancing my skills as a web developer.

.

Keywords: Full Stack Development, Web Development, Front-End Technologies, Back-End Technologies, HTML

ACKNOWLEDGEMENT

We are extremely thankful to our beloved Chairman and founder **Dr. M. Mohan Babu** who took keen interest to provide us the oppurtunity for carrying out the project work.

We are highly indebted to **Dr. B.M.Satish**, Principal of Sree Vidyanikethan Engineering College for his valuable support in all academic matters.

We are very much obliged to **Dr. B. Narendra Kumar Rao,** Professor & Head, Department of CSE, for providing us the guidance and encouragement in completion of this work.

I would like to express my special thanks of gratitude to the *AICTE EduSkills*, Odisha who gave me the golden opportunity to do this wonderful internship, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Sane Navya
21121A3360

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

Abbreviations

- HTML: Hypertext Markup Language
- CSS: Cascading Style Sheets
- API: Application Programming Interface
- Git: Version Control System
- IDE: Integrated Development Environment
- CI/CD: Continuous Integration/Continuous Deployment
- SQL: Structured Query Language
- PDF: Portable Document Format
- PNG: Portable Network Graphics
- JPEG: Joint Photographic Experts Group
- JS : Java script
- Json : java script object notation

CHAPTER 1
INTRODUCTION

1.1Company Profile

EduSkills is a non-profit organization dedicated to building an Industry 4.0-ready digital workforce in India. The organization bridges the gap between academia and industry by providing students and faculties with access to world-class curriculum, tools, and platforms. By enabling skilling interventions in collaboration with various stakeholders, EduSkills ensures that students are well-equipped with the technical skills necessary to meet the demands of the modern workforce.

EduSkills collaborates with students, faculties, educational institutions, and Central/State Governments, creating a synergistic approach to upskilling. Through its efforts, EduSkills aims to revolutionize teaching methodologies and promote an ICT (Information and Communication Technology)-based education system, ultimately transforming India's education landscape.

1.2Vision

EduSkills' vision is to bridge the gap between academia and industry, ensuring that students and faculties across India have access to a world-class curriculum. The organization seeks to disrupt traditional teaching methodologies by leveraging cutting-edge technology and ICT tools, making the education system more relevant and effective for today's digital economy. The broader goal is to support the vision of a 'Skilled India,' empowering students to be industry-ready and contributing to the nation's economic growth.

1.3History

EduSkills was founded with the mission of transforming the education ecosystem in India by focusing on three critical areas: Education, Employment, and Entrepreneurship. Over the years, the organization has grown by fostering partnerships with educational institutions, governments, and industry leaders, ensuring a 360-degree approach to skill development. EduSkills has actively worked to integrate modern teaching practices, ensuring that the students and faculties are equipped with the tools they need to succeed in a rapidly changing job market. The organization's holistic approach continues to make a social and business impact, helping build a robust, skilled workforce for the future.

1.4 Job Description

As a Full Stack Web Development Intern at EduSkills, my role encompassed both frontend and backend development, along with API creation and integration of AI capabilities. On the backend, I used frameworks like Node.js to build scalable RESTful APIs, manage data storage, and ensure data security. For the frontend, I employed technologies such as HTML, CSS, and JavaScript frameworks to develop interactive, responsive interfaces.

My responsibilities included designing and implementing APIs to connect the frontend with backend databases, initially using SQL databases like MySQL to handle relational data more efficiently. The project also involved integrating the Gemini AI API to build chatbot functionality, where I created endpoints for seamless user interaction and response handling. I utilized tools like Postman for API testing and Git for version control to maintain code reliability and adhere to industry best practices. This experience allowed me to work with a range of technologies, providing a well-rounded understanding of full-stack development.

1.5Objectives

The main objectives of this internship in Full Stack Web Development were as follows:

1.Hands-On Full Stack Development Experience:
To gain practical exposure to full stack web development by working on both frontend and backend applications, focusing on creating interactive user interfaces and seamless server interactions.
2.API Development and Optimization:
To design, implement, and optimize RESTful APIs using Node.js and Express, ensuring efficient data handling and the security of API endpoints for reliable communication between frontend and backend in a production environment.
3.Database Management and Migration:
To enhance understanding of database management by working with document-based databases like MongoDB and relational databases like MySQL, focusing on data scalability and the migration of data structures to support production-level applications.
4.AI Integration for Chatbot Functionality:
To integrate third-party AI solutions, specifically the Gemini AI API, into a web application to enhance interactivity, creating endpoints that enable real-time chatbot responses and a dynamic user experience.
5.Adoption of Industry-Standard Tools and Practices:
To gain proficiency with essential development tools, such as Git for version control and Postman for API testing, and to adopt secure coding practices and agile methodologies to meet industry standards for efficient and reliable development.

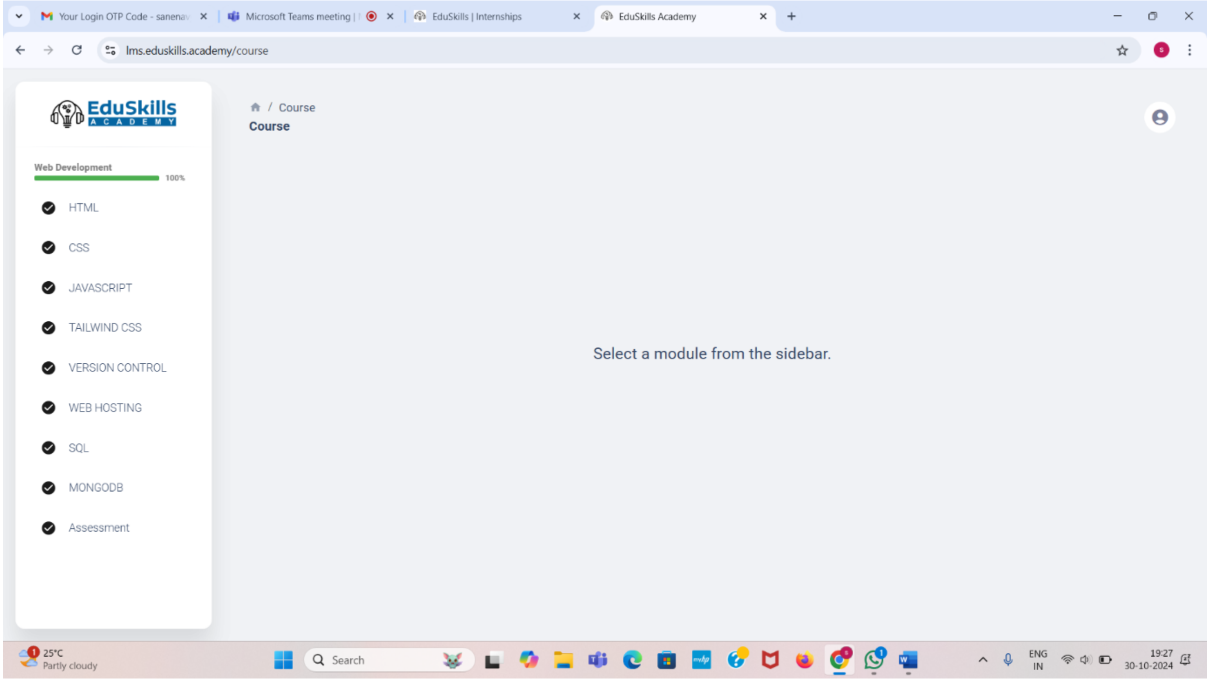
Each of these objectives contributed to a comprehensive learning experience, equipping me with technical skills in both frontend and backend development and soft skills crucial for a career in software development. The internship provided a solid foundation in full stack development, API design, and AI integration, preparing me to tackle complex projects in the future.

During the internship, I utilized a comprehensive set of tools and technologies to gain hands-on experience in full stack web development, streamlining the process of building, testing, and deploying applications. On the backend, I worked extensively with Node.js and Express.js, which simplified server-side development and enabled efficient API creation. MongoDB was used as a NoSQL database for flexible data management, while MySQL provided a relational database option to ensure data persistence and scalability as needed. For the frontend, I applied HTML, CSS, and Bootstrap to create responsive and visually appealing layouts, while JavaScript and frameworks like React (or Angular) were used to build dynamic, interactive components that enhanced user engagement. Git, paired with GitHub, was essential for managing source control, tracking code versions, and supporting collaboration, ensuring a structured and efficient development process. Additionally, tools like Postman facilitated API testing, enabling thorough validation of endpoints and streamlining communication between frontend and backend.

Collectively, these tools and technologies provided a complete platform for building robust web applications, covering everything from user interface design to data handling, server management, and version control.

List of Tools and Technologies:

1. HTML
2. CSS
3. Javascript
4. Tailwind CSS
5. Version Control
6. Web Hosting
7. SQL
8. MongoDB



Select a module from the sidebar.

Fig 1.5: Internship course overview

1.6. WORKING OF WEBSITE:

A website is a collection of webpages located under a single domain name. These webpages contain and display information about the business or organization and are usually made up of elements such as photos, videos and text. Users need a device with a web browser laptops, smartphones or tablets and an Internet connection to access a website.

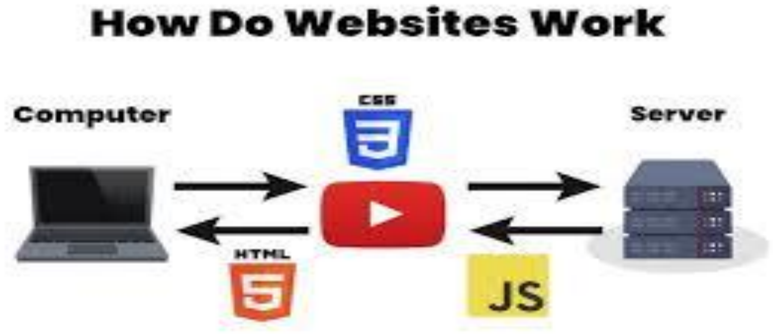


Fig 1.6.1: Working of website

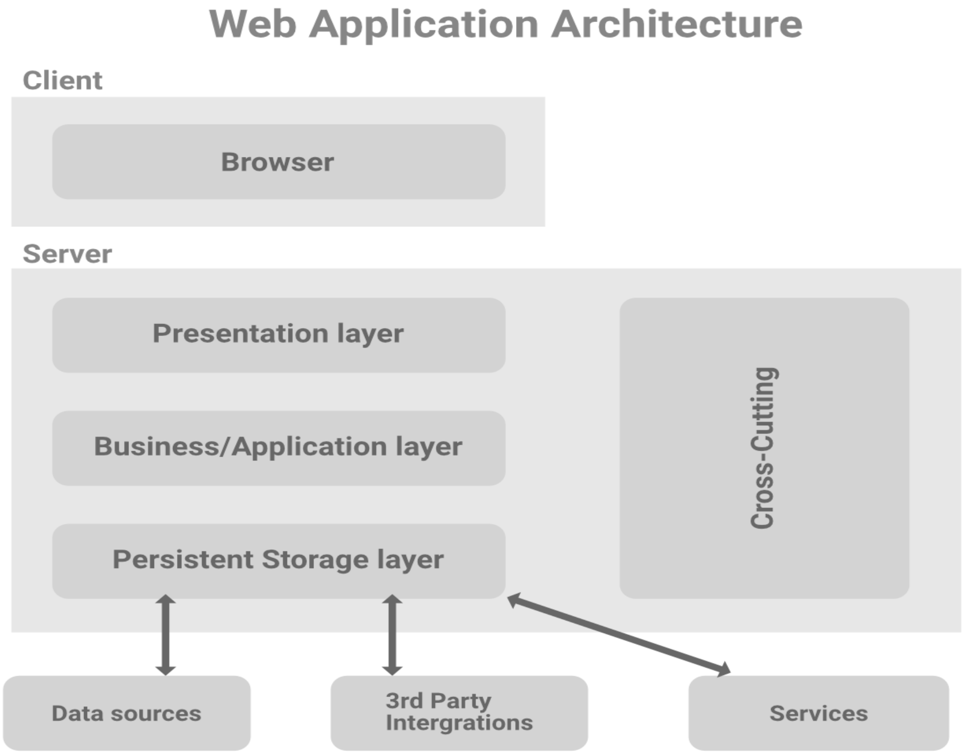


Fig 1.6.2: Web Application Archtecture

CHAPTER 2
SUMMARY OF EXPERINCE

2.1 HTML

HTML (HyperText Markup Language) is the standard markup language for creating web pages. It structures content using tags, which define elements like headings, paragraphs, images, and links. HTML is essential for building the backbone of any website, enabling browsers to render content in an organized and visually accessible format.
HTML5 is the latest version of HTML, introducing new features such as semantic tags (e.g., <header>, <footer>, <section>, <article>), multimedia support (audio and video), and enhanced form controls. HTML5 also improves compatibility across devices, including desktops and mobile, making web applications more dynamic and interactive.

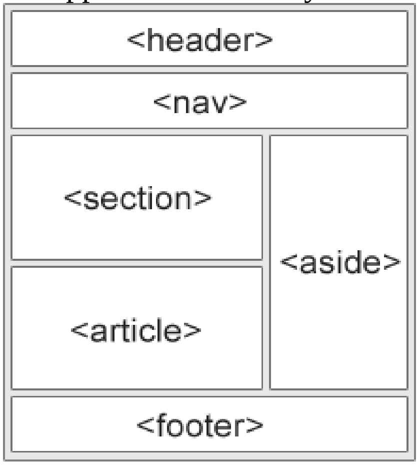


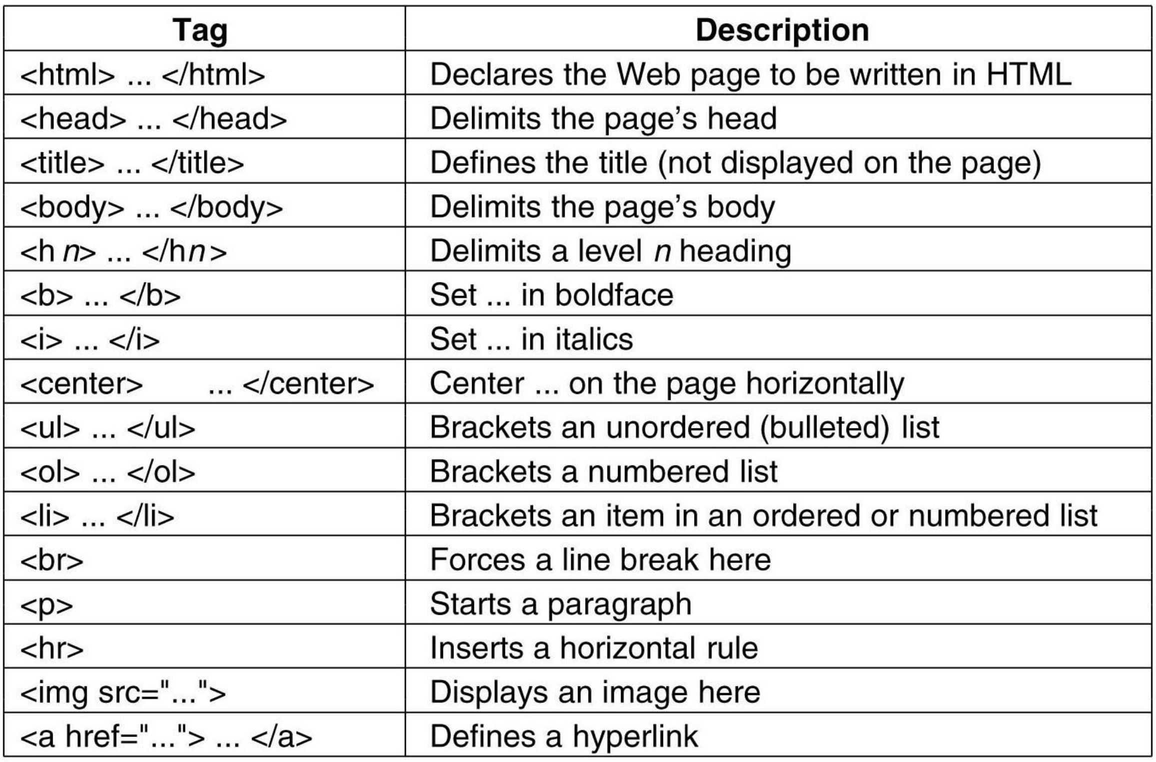
Fig 2.1: HTML Layout

- <header> - Defines a header for a document or a section
- <nav> - Defines a set of navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent, self-contained content
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details that the user can open and close on demand
- <summary> - Defines a heading for the <details> element

HTML (Hyper Text Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements".



Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li> ... </li>	Brackets an item in an ordered or numbered list
<br>	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

Table 2.1: HTML commands

2.1.1. Features of HTML:

1) It is a very easy and simple language. It can be easily understood and modified.

2) It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.

3) It is a markup language, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is platform-independent because it can be displayed on any platform like Windows, Linux etc.

6) It facilitates the programmer to add Graphics, Videos, and Sound to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lowercase or

upper-case.

2.1.2. HTML Page Structure:

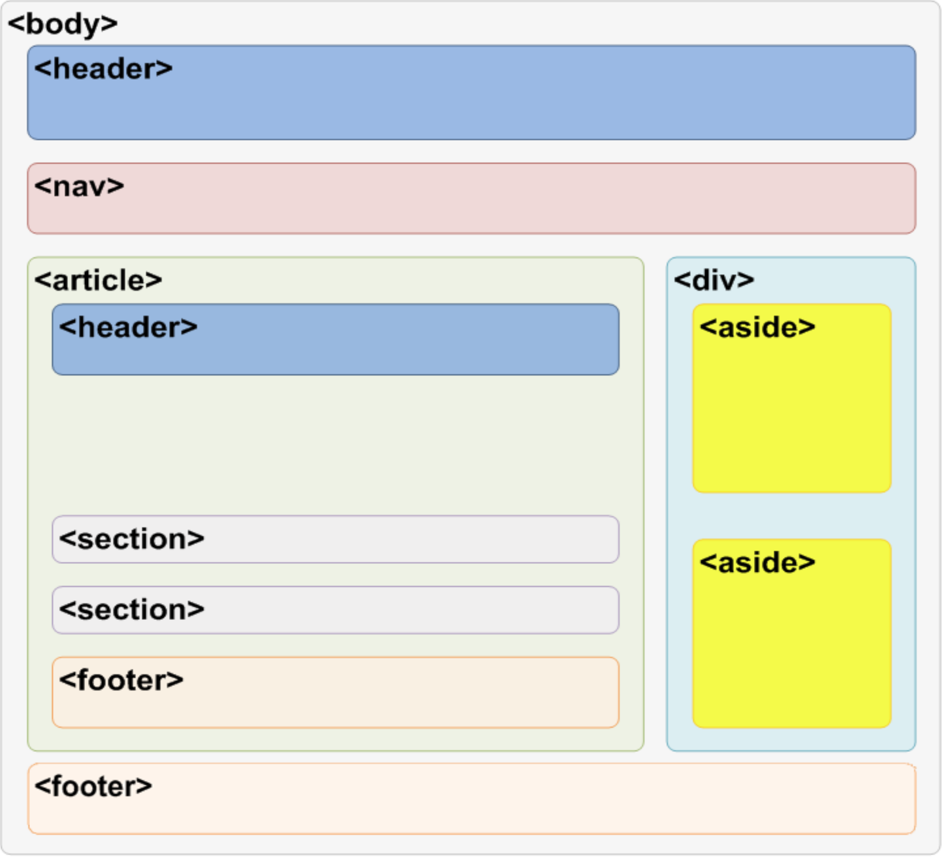


Fig 2.1.2: HTML page structure

- The <**!DOCTYPE html**> declaration defines that this document is an HTML5 document
- The <**html**> element is the root element of an HTML page
- The <**head**> element contains meta information about the HTML page
- The <**title**> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <**body**> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <**h1**> element defines a large heading
- The <**p**> element defines a paragraph

2.2 CSS

CSS (Cascading Style Sheets) is a language used to style and format the layout of web pages. While HTML provides the structure and content, CSS allows developers to control the appearance by applying styles to HTML elements, such as colors, fonts, spacing, and positioning. CSS enables the separation of content and design, making it easier to manage and update the look of a website.

CSS3 is the latest version of CSS and includes several new features, such as animations, transitions, flexbox, and grid layouts, which provide powerful tools for responsive and dynamic design. CSS3 also introduces new selectors and properties that enhance control over the visual styling of elements.

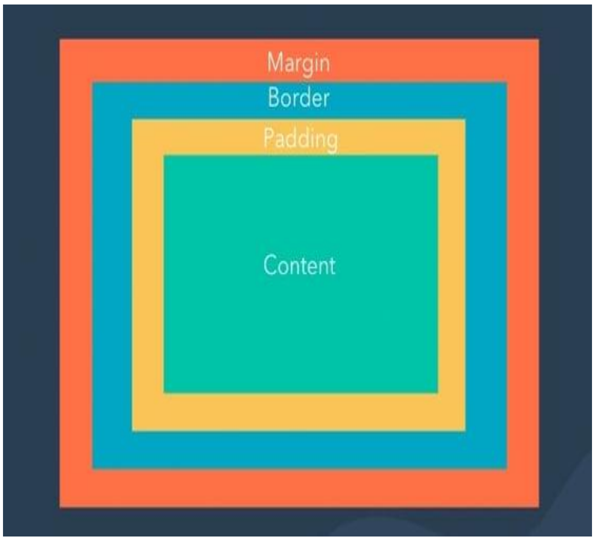


Figure 2.2: CSS Box Model

The CSS Box Model is a fundamental concept in web design and development, describing how elements are structured and spaced on a web page. Every HTML element is considered as a rectangular box, and the box model defines how the content, padding, borders, and margins are calculated to determine the total size and layout of an element.

accent-color	Specifies an accent color for user-interface controls
align-content	Specifies the alignment between the lines inside a flexible container when the items do not use all available space
align-items	Specifies the alignment for items inside a flexible container
align-self	Specifies the alignment for selected items inside a flexible container
all	Resets all properties (except unicode-bidi and direction)
animation	A shorthand property for all the *animation-** properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forward, backward or in alternate cycles
animation-duration	Specifies how long an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played

animation-name	Specifies a name for the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of an animation
aspect-ratio	Specifies preferred aspect ratio of an element

Table 2.2:CSS Properties

The box model consists of four main parts, from the innermost layer to the outermost:

1. **Content:** This is the actual content of the element, such as text, images, or other media. The content area's size can be adjusted using properties like width and height.
2. **Padding:** The space between the content and the element's border. Padding adds space within the box, making the content area appear larger without increasing the border or affecting surrounding elements. Padding can be applied individually (e.g., padding-top, padding-right) or all at once (padding shorthand).
3. **Border:** The border surrounds the padding and content. It can be styled with thickness, color, and style (e.g., solid, dashed). Properties like border-width, border-style, and border-color define the look of the border.
4. **Margin:** The space outside the border that separates the element from other elements. Margins create spacing between elements on a page and can be adjusted individually (e.g., margin-top, margin-bottom) or as a shorthand (margin).

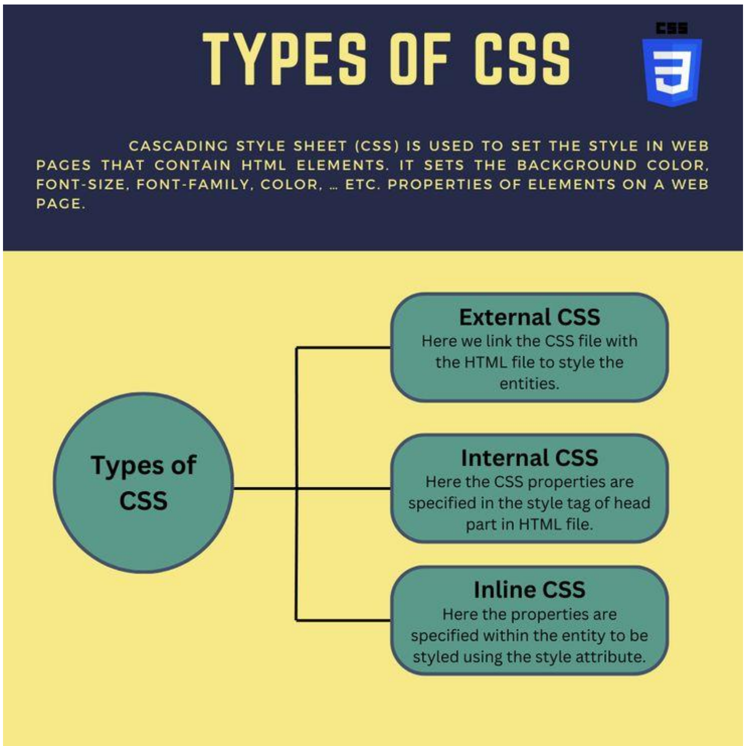
Cascading Style Sheets are fondly referred to as CSS. It is used to make web pages presentable. The reason for using this is to simplify the process of making web pages presentable. It allows you to apply styles on web pages. More importantly, it enables you to do this independently of the HTML that makes up each web page.

2.2.1. Features of CSS

1). Font, color, background style, element alignments, border, and size tags had to be duplicated on each web page before CSS. This was a lengthy procedure.
2). CSS makes website maintenance much easier. If we need to make a global change to the file, we can simply alter the style, which will update all of the elements on the web page.
3). We can use CSS with older language versions because it is compatible with them. CSS makes it possible to optimize material for several devices
4). A website that's dull and plain will not attract the user, so adding some style would surely make your website presentable to the user.
5). A website with a simple yet beautiful UI would help the users to go through the website easily. It is used to make the user interface better.
6). Since CSS is a basic requirement while learning Web Development, therefor there are abundant career opportunities for it. As a freelancer, you can land up to many projects.

2.2.2. Types of CSS:

There are three types of CSS which are given below:



TYPES OF CSS

CASCADING STYLE SHEET (CSS) IS USED TO SET THE STYLE IN WEB PAGES THAT CONTAIN HTML ELEMENTS. IT SETS THE BACKGROUND COLOR, FONT-SIZE, FONT-FAMILY, COLOR, … ETC. PROPERTIES OF ELEMENTS ON A WEB PAGE.

Types of CSS

External CSS
Here we link the CSS file with the HTML file to style the entities.

Internal CSS
Here the CSS properties are specified in the style tag of head part in HTML file.

Inline CSS
Here the properties are specified within the entity to be styled using the style attribute.

Fig 2.2.2: Types of CSS

Inline CSS: Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.
Internal or Embedded CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e., the CSS is embedded within the HTML file.
External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc.). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

2.2.3. Properties of CSS:

Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.
1) As Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.
2) Internal or Embedded stands second in the priority list and overrides the styles in the external style sheet.
3) External style sheets have the least priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

2.3 JavaScript

JavaScript was invented by **Brendan Eich** in 1995 while he was working at Netscape Communications Corporation. It is a high-level, dynamic, untyped, and interpreted programming language that is primarily used to create interactive and dynamic content on websites. JavaScript enables developers to implement complex features on web pages, such as animated graphics, form validations, dynamic content updates, and interactive elements like sliders and tabs. As a core technology of the World Wide Web, alongside HTML and CSS, JavaScript allows for the development of rich user interfaces and provides a means for client-side scripting, significantly enhancing the overall user experience.

The main uses of JavaScript in web development include enabling interactive features, enhancing user engagement, and facilitating asynchronous communication with servers through technologies like AJAX. JavaScript's versatility allows it to be used for both front-end and back-end development, especially with the advent of environments like Node.js. Additionally, its extensive ecosystem of frameworks and libraries, such as React, Angular, and Vue.js, further simplifies and accelerates web application development. With its widespread support and active community, JavaScript remains one of the most popular programming languages in the world, making it an essential skill for modern web developers.

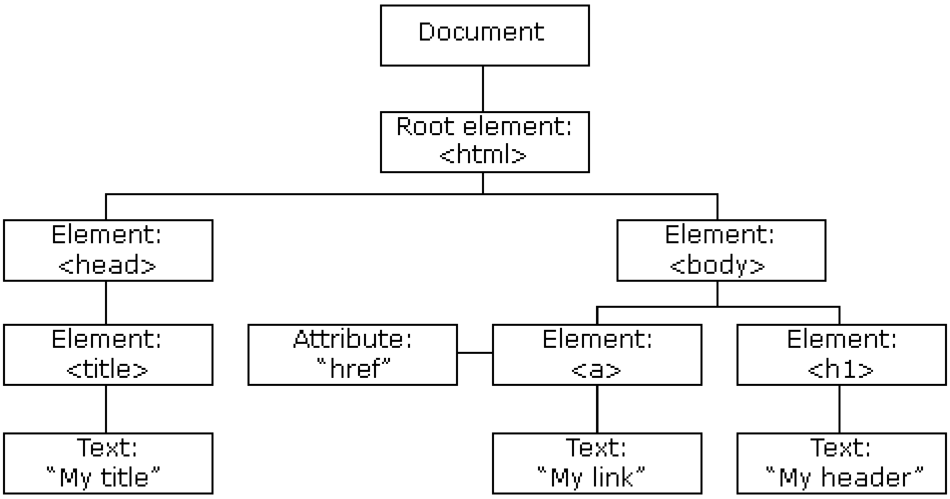


Figure 2.3: Document Object Model

The **Document Object Model (DOM)** is a programming interface that allows scripts to dynamically access and update the content, structure, and style of a document, typically a web page. It represents the document as a hierarchical tree structure, where each node corresponds to a part of the document, such as elements, attributes, and text. This tree structure enables developers to traverse, manipulate, and modify the document easily using programming languages, primarily JavaScript. The DOM facilitates dynamic interaction by allowing real-time modifications of web content, enabling features like interactive forms and animations. It also supports event handling, allowing developers to respond to user actions such as clicks or keyboard input by attaching event listeners to specific elements. Various methods and properties, such as getElementById(), appendChild(), and setAttribute(), provide powerful tools for accessing and manipulating elements. Overall, the DOM serves as an essential bridge between web documents and programming languages, enabling the creation of engaging and responsive web applications.

JavaScript (JS) is the world’s most popular lightweight, interpreted compiled programming language. It is also known as a scripting language for web pages. It can be used for Client-side as well as Server-side developments. JavaScript can be added to your HTML file in two ways:

Internal JavaScript: We can add JS code directly to our HTML file by writing the code inside the <script> tag. The <script> tag can be either be placed inside the <head> or the <body> tag according to the requirement.
External JavaScript File: We can create a file with .js extension and paste the JS code inside it. After creating the file, add this file in
<script src = \" file_name.js"> tag inside <head> tag of the HTML file.

2.3.1. Features of JavaScript

1) All popular web browsers support JavaScript as they provide built-in execution environments.
2) JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3) JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4) JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5) It is a light-weighted and interpreted language.
6) It is a case-sensitive language.

Method	Description
getData();	Get all tags as a object
getValue(number)	Get a specific tag by index or all tags value @param integer indexNumber - Null for all tags
setValue(string);	Set a new value for the javascript tagging @param string newValue - Values separate by comma
reset();	Clear all tags
isValid();	Validate tags

Table 2.3: Javascript Methods

2.4. TailwindCSS

Tailwind CSS is a utility-first CSS framework for designing websites by using its utility-first pre-defined classes. It is a low-level CSS framework that is easy to learn and maintain in your projects. Tailwind CSS has many built-in features and classes that can be directly used on HTML documents.

2.4.1. Features of Tailwind CSS:

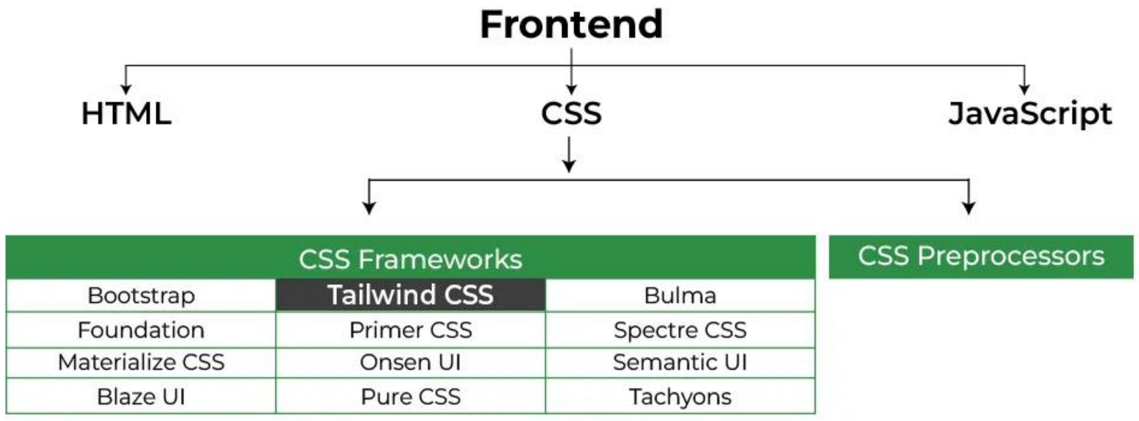
Tailwind CSS is a CSS framework that offers many advantages to creating a responsive and SEO-friendly website in this fast-paced development phase. It offers several advantages including.

- **Utility-First Paradigm**: Instead of writing custom CSS you can use pre-defined classes to decorate your HTML elements.
- **Responsive Design:** Tailwind CSS utility classes can be used based on screen size and breakpoints, so your website can be responsive.

- **Consistency and Maintainability:** Its unified design ensures that all of your pages can follow a consistent frontend design with easy maintainability.

- **Fast Pace Development:** Using pre-defined classes will always boost your development pace compared to using custom CSS.

- **Design Flexibility:** It has the largest pre-defined classes with the opportunity to create your design to make your design skills more flexible.



CSS Frameworks		
Bootstrap	Tailwind CSS	Bulma
Foundation	Primer CSS	Spectre CSS
Materialize CSS	Onsen UI	Semantic UI
Blaze UI	Pure CSS	Tachyons

Fig2.4.1: TailwindCSS

2.4.2.How to use Tailwind CSS:

There are two ways to use Tailwind CSS in projects, you can use it through the CDN link or install it on your server/local machine.

Tailwind CSS Intallation:

- Install Tailwind CSS using yarn

- Install Tailwind CSS using npm

Tailwind CSS CDN Link:

Using script Tag:

Just include a <script> tag in the <head> section of your HTML. This gives access to Tailwind's utility classes without extra files on your server.

<script src="https://cdn.tailwindcss.com"></script>

Using link Tag:

Just include a <link> tag in the <head> section of your HTML. This gives access to Tailwind's utility classes without extra files on your server.

<link href= "https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">

2.4.3.Key Advantages of Tailwind CSS:

1. **No need for complex class names**: You don’t have to worry about naming conventions for CSS classes and IDs.

2. **Minimized CSS code**: Tailwind reduces the need for large custom CSS files, keeping your codebase smaller and more manageable.
3. **Easy customization**: Tailwind allows easy customization of designs without writing additional CSS, helping you create reusable components.
4. **Built-in responsiveness**: Tailwind's classes are optimized for responsiveness, allowing developers to create mobile-friendly layouts effortlessly.
5. **Scoped styles**: Tailwind's utility classes help in making local changes to specific elements without affecting the entire stylesheet, unlike traditional global CSS.

Class	**Properties**
border-spacing-0	border-spacing: 0px 0px;
border-spacing-x-0	border-spacing: 0px var(--tw-border-spacing-y);
border-spacing-y-0	border-spacing: var(--tw-border-spacing-x) 0px;
border-spacing-px	border-spacing: 1px 1px;
border-spacing-x-px	border-spacing: 1px var(--tw-border-spacing-y);
border-spacing-y-px	border-spacing: var(--tw-border-spacing-x) 1px;
border-spacing-0.5	border-spacing: 0.125rem 0.125rem;
border-spacing-x-0.5	border-spacing: 0.125rem var(--tw-border-spacing-y);

Table 2.4: TailwindCSS border spacing properties

Why Choose Tailwind Over Other CSS Frameworks?
Tailwind CSS stands out from traditional frameworks like Bootstrap or Foundation because of its **utility-first methodology**, which offers:

- **Granular control over styling**: Tailwind provides control at the atomic level, allowing you to customize each aspect of your design.
- **Faster prototyping**: By using utility classes, developers can iterate faster without worrying about conflicting styles or overriding pre-built components.
- **Lightweight code**: Tailwind generates smaller CSS files by purging unused styles, improving website performance with faster load times.
- **Simplified responsive design**: Tailwind's utility classes make responsive design effortless without the need for custom media queries.
- **Extensive documentation**: Tailwind provides clear documentation and an intuitive syntax that speeds up the development process.

2.5. Version Control

Web development is a constantly evolving field, and developers must keep up with the latest version tools and techniques. Version control is one of the most important aspects of web development, and we'll go over what it is and why it's so important in this blog.

Version control is a technique that tracks changes to code over time, allowing developers to collaborate on a project and revert to previous versions as needed. Version control is an essential tool for any web developer, whether they are working on a small website or a large-scale web application.

In this blog, we'll go over the fundamentals of version control and how to use it effectively in your web development projects. We'll also discuss the various version control systems available and how to select the best one for your requirements. This guide will provide valuable insights to help you stay ahead of the game, whether you're a beginner or a web development expert.

Version control is a software development technique that lets multiple developers work on the same set of source code files. Each time they save a file, the version control system automatically saves a copy of the previous version, known as a "checkpoint."

This way, if one developer accidentally deletes or overwrites another developer's work, or if someone accidentally erases all of the files, the most recent copy can be restored.

Version control systems also let you see who changed what and when; roll back to previous versions; compare different versions (like when you're updating an existing website); and collaborate with other developers on projects that have grown too large for one person to handle alone. Version control also helps in case something goes wrong with your website - for example, if it stops working because of server issues or if there is malware on it. If you have a source control system in place, you can restore the site to its last known good state and then use that as your starting point for fixing things. This is especially important if you build websites for clients - they are going to need an easy way to get their website back up and running if something goes wrong.

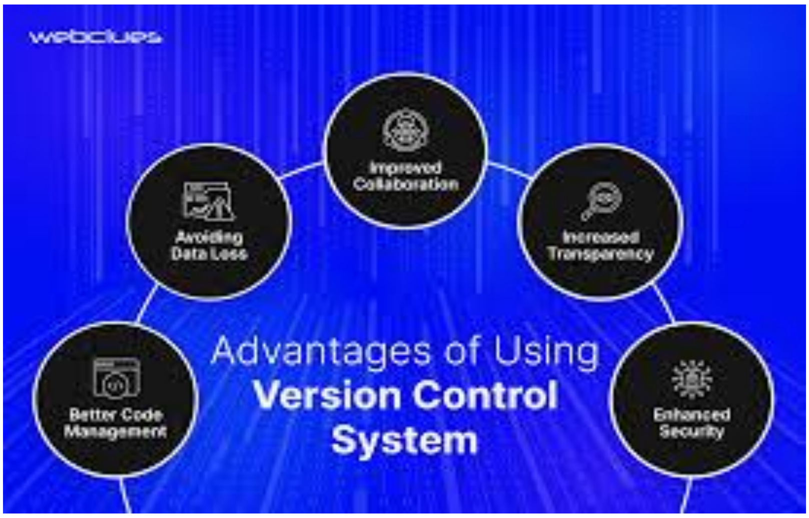


Fig 2.5: Advantages of using Version Control System

2.5.1. Importance of version control in web development

Software developers and web developers are always in search of effective tools to keep their work organized. Version control is one of the most useful tools that software developers use, especially if they are involved in large software projects. It's an important tool that helps software developers manage their projects. It is a system that keeps track of the changes made to files and allows users to revert back to previous versions if they need to.

Assume you're working on a website with a team of developers. Everyone is working on a different feature. By accident, one developer edited another developer's work and pushed it to production. The website is currently down. You're attempting to identify the problem and eventually locate it. However, you cannot go back in time and prevent this from happening. What if you could at least go back to the previous version of the source code? This is where version control comes in handy.

With version control, the code base can be managed and tracked easily. This means that any changes made will be recorded and can be traced back to see who made them and when they were made. It also helps to avoid conflicts between team members as each person's changes are kept separate from each other's changes.

Version control systems are also useful for people who need a quick way to find the most recent version of a file. If you've changed a document several times over the course of a week and want to see what it looked like at the beginning, it can be difficult to find out which version is the most current. A version control system makes this process much easier by storing multiple versions of files in one place so they're easy to access and compare.

The importance of having version control cannot be overstated as it allows developers to work independently on the same project without fear of messing up each other's work or stepping on each other's toes.

2.5.2. The Basics of Version Control

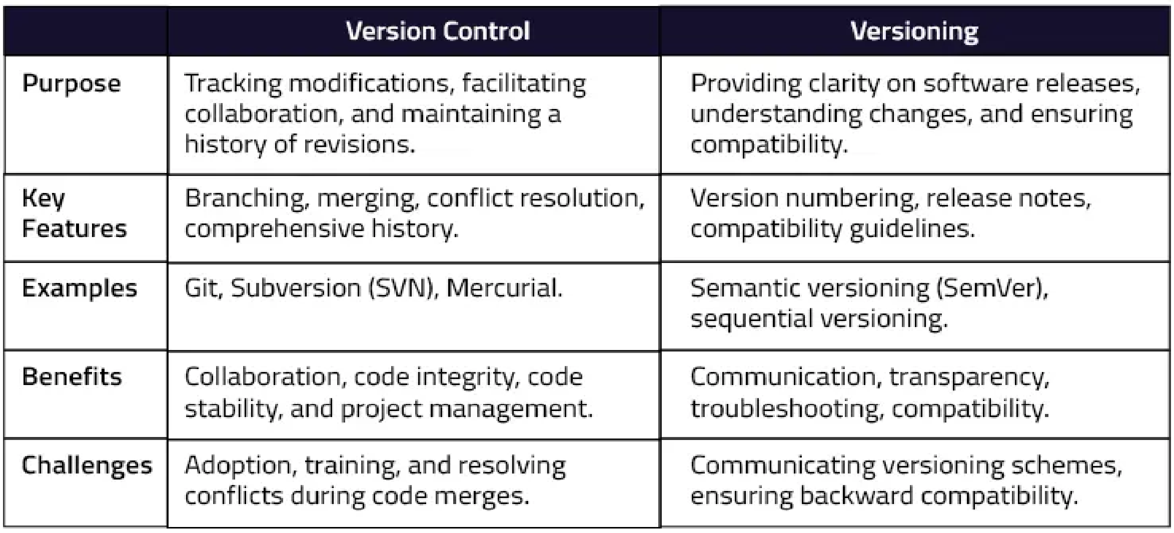
Version control is a system that tracks changes to files and allows a user to revert to older versions of the file.

It is used to maintain historical versions of files, documents, and websites. This allows users to track changes to a document or file over time. Version control can also be used to coordinate work on multiple versions of a project by multiple developers.

Version Control Systems (VCS) is designed specifically for software development teams to manage changes to codebases in a collaborative environment. There are many different types of VCS systems available, but they all have some basic concepts in common.

Version control systems are especially useful for collaborative software projects because they allow multiple users to work on the same files at the same time without overwriting each others' changes. Many source control systems allow users to merge conflicting changes into one final document so they don't have to worry about losing any of their changes when other people edit the same document at different times.

Version control systems are often used in software development, but they can also be useful for non-technical tasks. For example, if you're writing a book or blog post, it's helpful to have an easy way to track changes over time. You might even use one while working on a large project with other people—for example, if you're collaborating on a shared document or spreadsheet with others and need a way to keep track of what everyone is doing.



	Version Control	Versioning
Purpose	Tracking modifications, facilitating collaboration, and maintaining a history of revisions.	Providing clarity on software releases, understanding changes, and ensuring compatibility.
Key Features	Branching, merging, conflict resolution, comprehensive history.	Version numbering, release notes, compatibility guidelines.
Examples	Git, Subversion (SVN), Mercurial.	Semantic versioning (SemVer), sequential versioning.
Benefits	Collaboration, code integrity, code stability, and project management.	Communication, transparency, troubleshooting, compatibility.
Challenges	Adoption, training, and resolving conflicts during code merges.	Communicating versioning schemes, ensuring backward compatibility.

Table 2.5: Version Control vs Versioning

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. It's most often used in software development, but it has applications beyond that. The information stored in a version control system can be used to restore older versions of files and documents, compare versions, and collaborate with others on projects.

Version control systems are typically used by developers who are working on large projects and need a way to track their work. A version control system provides an easy way to see what changes have been made over time, so you can do code review on the history of your project and see how it evolved from day one.

Here's how it works: when you make changes in your project files directory, those changes are stored in a repository — usually with a version control system (VCS). When you want to share your work with someone else, you can send them the codebase from that repository. When they make their own changes, those changes will be stored separately from yours, so that both of your versions can be compared later on if necessary.

When you go back to an older version of your codebase, it's often called "checking out" that version from the repository. This creates a new version of the codebase on your machine that has all its files in the same state they were when you checked them out — meaning they may not be up-to-date with any changes made since then.

2.5.3. Different types of version control systems

There are two types of version control systems, each with its own advantages and disadvantages:

1. Centralized Version Control Systems (CVCS)

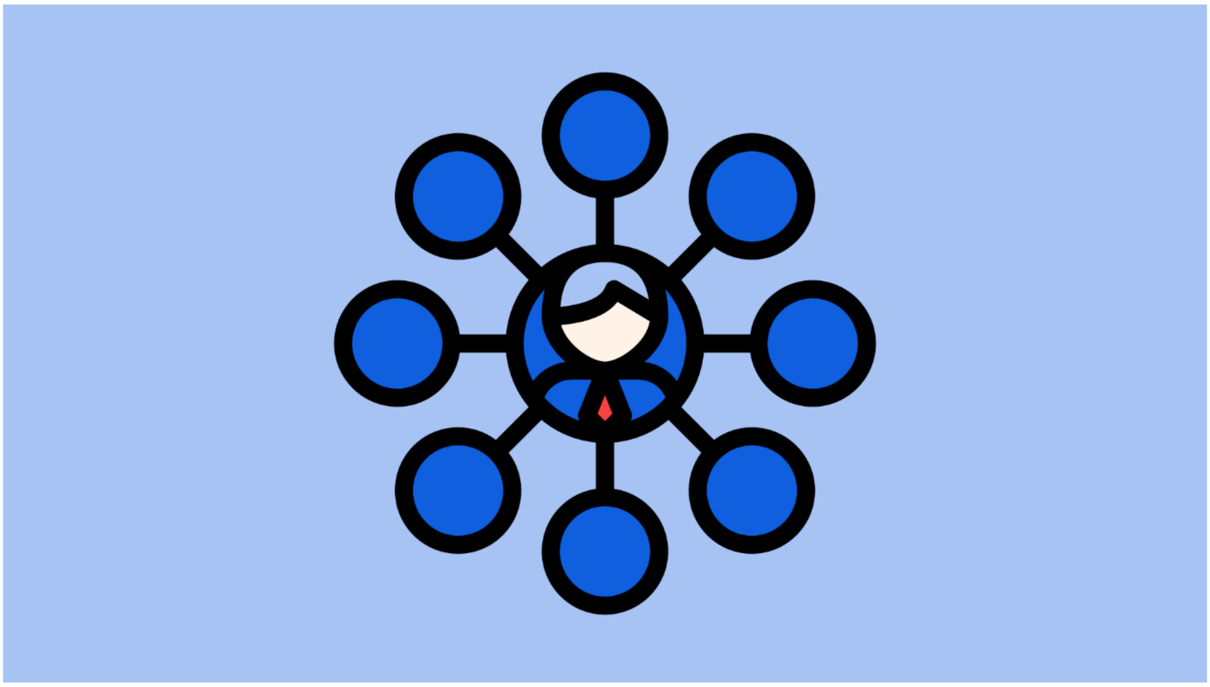


Fig 2.5.3.1: Centralized Version Control Systems (CVCS)

These systems keep all the files in a central repository, where only one person can work on them at a time. With centralized source control, only one person can edit a file at a time; others must wait for it to be checked out before they can work on it.

2. Distributed Version Control Systems (DVCS)

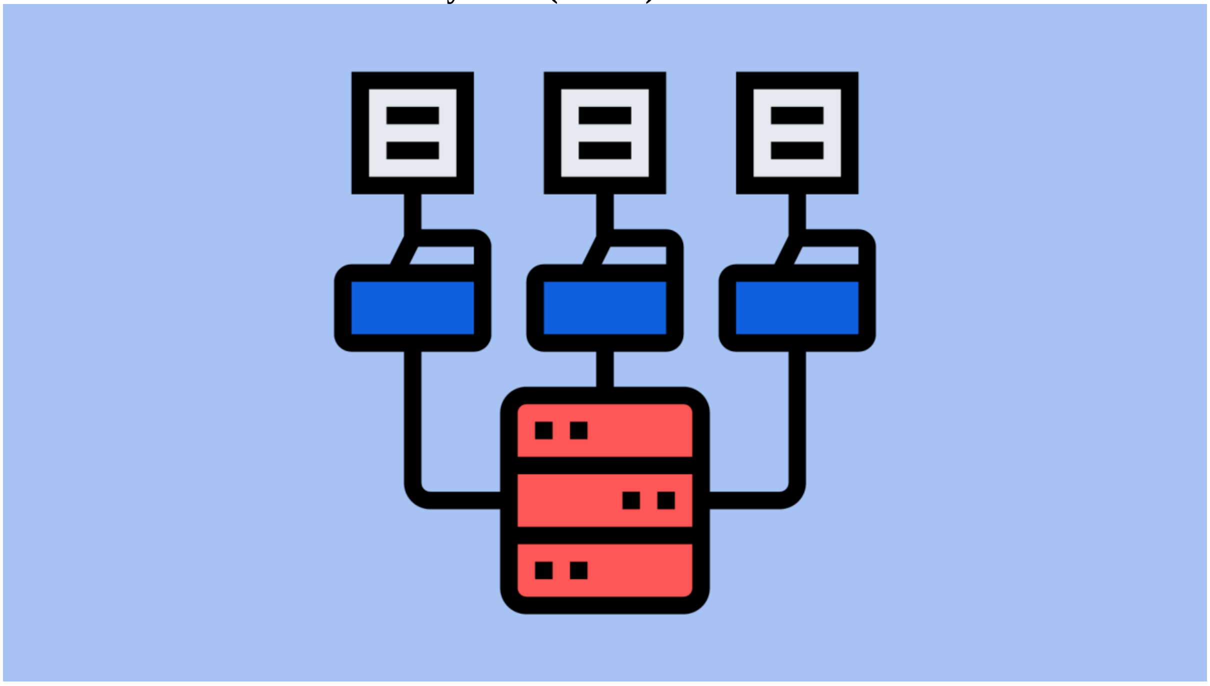


Fig 2.5.3.2: Distributed Version Control Systems (DVCS)

Distributed Version Control Systems (DVCS) are a type of version control system (VCS) that are designed to allow several developers to work on the same codebase at the same time. This means that multiple people with multiple computers can be working on the same project in parallel, and when they push their changes to the server, it will merge them together.
The most popular DVCS is Git, which was created by Linus Torvalds in 2005.

2.5.4. Benefits of Version Control Systems

Improved collaboration

Version control systems can help to improve collaboration and communication within a software team. Through version control, it's easy to see which changes have been made by whom and when. This can help to avoid instances where multiple people make different changes to the same file, causing confusion over who should be responsible for which change.

Ease of managing changes

Version control systems make it easy to manage the changes that have been made to a project. This can be useful when you need to go back and see what has changed since a certain point in time, or if you want to revert back all of your changes at once. Version control systems can help to manage changes by providing a central database where all files and their associated revisions are stored. They also enable you to view the changes that have been made over time, which means it's easy to see how a document has evolved from its original format into its current state.

Ability to revert to previous versions

If you make a change that you later regret, it’s easy to revert back to an earlier version of the file. This is especially useful for documents that have been edited over time by many different people, as it prevents any one individual from inadvertently changing the formatting or content of a document.

This feature is also useful for situations where you want to try out a new format or layout for your document, but want the ability to go back if it doesn’t work out. Version history helps prevent users from making changes that cannot be undone because they are unaware of how their actions might affect other people working on the same document.

Enhanced security and backup

Version history is a useful feature for anyone who wants to protect their documents from being inadvertently changed. It also helps make sure that important files don’t get lost in case of a computer crash or other disaster. You can use version history to create a backup of your files and then restore them if there is a problem.

With version history, you can see the date and time that each version was saved. You can also restore a previous version of a file or folder if you need to do so. If you want to make sure that your files are secure, then this tool will help.

2.5.5. Reasons to Use Version Control

Better organization of code

As a developer, you’ll find that it makes your code much easier to manage. You can go back in time and see what changes have been made over the course of development, and quickly revert changes if necessary. Version control is essential in any large project and can help you avoid rewriting existing code. It also makes it easy for your team members to work together on the same project without stepping on each other’s toes.

Increased productivity

Version control allows you to focus on coding without having to worry about making sure your changes don’t conflict with someone else’s. This frees up your brain power and makes it easier for you to get things done quickly. You can also get feedback from your team members more easily and make decisions together based on their input.

Improved collaboration and teamwork

Version control helps you work with your team members more easily. You can discuss issues and resolve them quickly without having to worry about stepping on each other’s toes. This makes it easier for your development team to collaborate effectively and move projects forward quickly.

Things to Look for in Version Control Software

Version control software is a must-have for any development team. It can improve collaboration and productivity, but it’s not always easy to choose the right one. If you are looking to add version control to your development workflow, here are some things to look for:

User-friendly interface

Version control software is a tool for developers, so it needs to be easy for them to use. If it’s hard to configure or difficult to navigate, programmers won’t use it. The version control software can be intimidating at first. But if you find one that has a user-friendly interface, you’ll be able to get up and running quickly. It will also make it easier for your team members to adopt the software as well.

Integration with other tools

Many version control platforms integrate with other tools, such as issue trackers and wikis. This makes it easier for developers to use the software in their workflows. If you want to add version control to your development workflow, look for a tool that supports these integrations. If it’s hard to integrate version control with other software tools in your development workflow, you’ll have a harder time getting the most out of it. For example, you might need to use your version control software alongside code editors like Visual Studio and Sublime Text in order to view changes and roll back versions of your codebase.

Good support and documentation

Good documentation is also important. If you’re not sure how to use a tool, it can be frustrating if there’s no documentation available for you to refer to. You should be able to find clear instructions on how to install and configure your version control software, as well as guidance on how it works with other software tools in your development workflow.

A good example of this is GitHub’s documentation page, which provides a variety of guides and tutorials that cover the basics of using Git as well as more advanced features. In addition to providing documentation, it also offers support via email if you run into any problems with your version control software.

Flexibility

The version control software can be used in many different ways, and there are many different types of repositories. This means that you should be able to find a version control system that works best for your needs. For example, Git is great if you want to host your code on GitHub so that others can collaborate with you online.

Affordable pricing

Version control software is often free to use, but some systems have a paid version that offers additional features. For example, GitLab offers both a free and paid version of its product. The paid version allows you to host your code on the GitLab server rather than using GitHub or Bitbucket, which can be helpful if you want more control over your project.

Conclusion

Finally, mastering version control in web development is critical for ensuring that your projects are well-managed and organized. The advantages of using version control software are numerous, including easy collaboration, backup, and recovery of previous file versions, tracking changes, and facilitating project management.

Version control is a key component in modern software development, especially with the emergence of cloud technologies and distributed software teams. It provides an efficient workflow, reduces the risk of errors, and improves code quality by facilitating code reviews and testing.

Choosing the best version control software for your team can be a daunting task, but it's important to consider factors like ease of use, security, collaboration features, and integrations with other tools. Popular version control tools include Git, Subversion, and Mercurial, each with its own strengths and weaknesses.

In conclusion, mastering version control in web development is a crucial step toward becoming a more efficient, organized, and productive developer. It's not only about managing code changes but also about streamlining the entire development process. With the right tools and a good understanding of the principles of version control, you can take your web development projects to the next level.

2.6. Web Hosting

In the world of web development, bringing full-stack applications to life is a very thrilling experience. However, once you are done with code the next critical step is to find a reliable and cost-effective hosting solution. Fortunately, in the vast field of cloud services, there are plenty of platforms offering hosting options, allowing us to deploy and showcase our application to the digital world.

Web hosting is a service that allows individuals and organizations to make their website accessible on the internet. Essentially, it involves storing website files on a server, a powerful computer that's always online. This server is provided by a hosting company, which maintains, secures, and manages it. When users type a website's domain name into a browser, the browser connects to the web server where the site is hosted, allowing them to view the website.

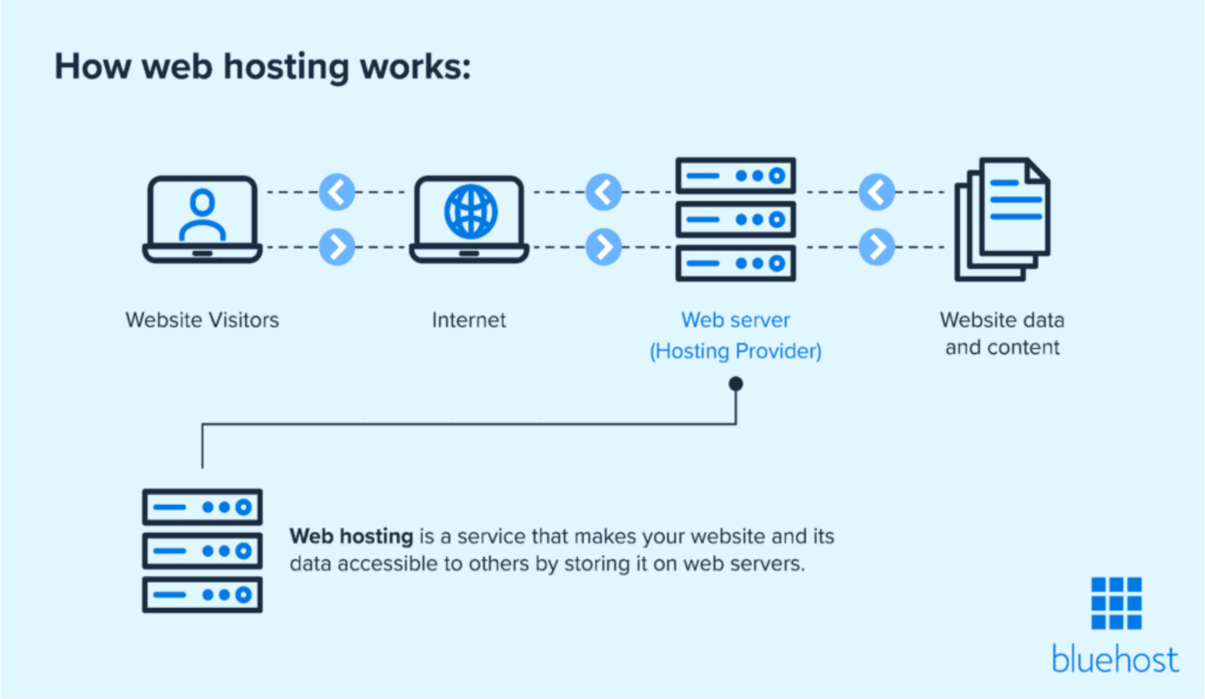

How web hosting works:
Website Visitors
Internet
Web server (Hosting Provider)
Website data and content
Web hosting is a service that makes your website and its data accessible to others by storing it on web servers.
bluehost

Fig 2.6: Web hosting

How Web Hosting Works

To publish a website, the files (HTML, CSS, JavaScript, images, videos, etc.) must be uploaded to a web server. These servers are connected to high-speed networks to ensure that websites load quickly and are accessible worldwide. Hosting providers manage the server's software, connectivity, security, and more, ensuring the website's availability 24/7. Users typically interact with these servers via control panels, which provide tools for managing files, databases, emails, security settings, and more.

Types of Web Hosting

Web hosting options vary based on storage, control, and scalability. Here's a breakdown of the most common types:

1. **Shared Hosting**
    - Description: In shared hosting, multiple websites share a single server's resources, making it the most affordable option.
    - Pros: Cost-effective, easy to set up, and good for beginners and small websites with low to moderate traffic.
    - Cons: Limited resources, potentially slower speeds due to shared resources, and less control over server settings.
2. **Virtual Private Server (VPS) Hosting**

   - Description: VPS hosting divides a physical server into virtual sections, giving each user a private environment with dedicated resources.
   - Pros: Offers more control and resources than shared hosting. Good for medium-sized websites with moderate to high traffic.
   - Cons: More expensive than shared hosting and requires some technical knowledge to manage.

3. **Dedicated Hosting**
   - Description: A single user rents an entire server dedicated to their website, providing maximum resources and control.
   - Pros: High performance, full control over server configurations, and suitable for large websites with high traffic and resource needs.
   - Cons: Expensive and typically requires technical expertise to manage.

4. **Cloud Hosting**
   - Description: Cloud hosting distributes website resources across multiple servers, creating a more flexible and scalable environment.
   - Pros: Highly scalable, pay-as-you-go pricing, and high availability due to redundancy across multiple servers.
   - Cons: Pricing can vary based on usage, and control over individual servers may be limited.

5. **Managed Hosting**
   - Description: Managed hosting provides users with additional support from the hosting provider, handling server setup, management, and maintenance.
   - Pros: Simplified management, regular updates, backups, and strong security measures.
   - Cons: Higher cost, and control over certain server aspects may be limited.

Essential Features of a Good Web Host

A good hosting provider should offer several key features to ensure your website performs well and remains secure. Here’s what to look for:

1. **Uptime and Reliability**
   - High uptime ensures that your website is accessible to users without frequent downtime. Most hosts aim for 99.9% uptime. Reliable uptime is critical for business credibility and user experience.

2. **Speed and Performance**

    - Speed affects user experience and SEO rankings. Fast hosting includes features like SSD storage, caching, and Content Delivery Networks (CDNs) to ensure rapid content delivery.

3. **Scalability**
    - As your website grows, you’ll need a hosting provider that can accommodate increased traffic, storage, and bandwidth. Cloud hosting is particularly popular for its scalability.
4. **Security Features**
    - Security features like SSL certificates, firewalls, malware scanning, and automated backups are essential to protect user data and website integrity.
5. **Control Panel Access (e.g., cPanel, Plesk)**
    - A user-friendly control panel allows for easy management of files, databases, email accounts, and security settings. Control panels streamline the hosting experience, especially for non-technical users.
6. **Customer Support**
    - 24/7 support can be crucial in case of technical issues. Look for providers that offer live chat, phone, and email support, along with an extensive knowledge base.
7. **Email Hosting**
    - Many hosts offer email hosting, allowing you to set up professional email addresses with your domain (e.g., yourname@yourdomain.com). Look for spam filtering and storage management.
8. **Backup Options**
    - Regular backups help prevent data loss. Some providers offer daily, weekly, or on-demand backup options, as well as restore points in case you need to revert to an earlier version of your site.

Domain Names and Web Hosting

A domain name is your website’s address on the internet. Most hosting providers offer domain registration services or allow you to connect an existing domain. Your domain name connects users to the server where your website files are hosted. Having a domain name related to your brand or business improves user recognition and trust.

Website Security in Hosting

Website security is crucial for protecting user data and your site’s reputation. Here are some core security measures often provided by hosts:

1. **SSL Certificates**

- An SSL (Secure Sockets Layer) certificate encrypts data between the user’s browser and your server, essential for protecting sensitive information like login credentials and credit card details.

2. **Firewall and DDoS Protection**
   - Firewalls block malicious traffic, while DDoS (Distributed Denial of Service) protection defends against traffic overload attacks that could crash your site.
3. **Regular Malware Scanning**
   - Malware scanners detect and help remove harmful software from your site, protecting it from hacking and data theft.
4. **Automated Backups**
   - Regular backups protect your site’s data by creating copies that can be restored in case of an incident, such as a security breach or accidental deletion.

Web Hosting and SEO

A website’s hosting can impact SEO (Search Engine Optimization) in several ways:

1. **Site Speed**
   - Faster-loading websites generally rank higher in search engines. Hosting that includes SSD storage and caching can help improve site speed.
2. **Uptime and Reliability**
   - Downtime negatively affects SEO rankings. Search engines aim to deliver reliable results, so a site that frequently goes offline is less likely to rank well.
3. **Location of the Server**
   - Hosting servers closer to your primary audience can improve load times, potentially benefiting SEO.

Popular Web Hosting Providers

Some popular web hosting providers include:

1. **Bluehost**: Known for its affordability and WordPress integration, with a beginner-friendly setup.
2. **HostGator**: Offers scalable hosting plans with reliable uptime and great customer support.
3. **SiteGround**: Strong in performance, security, and customer support, particularly for WordPress users.
4. **DigitalOcean**: A favorite among developers for cloud hosting with highly customizable servers.
5. **AWS (Amazon Web Services)**: Offers advanced cloud hosting solutions with powerful tools for scalability, often favored by large websites and applications.

How to Choose a Web Hosting Plan

Selecting the right hosting plan depends on several factors:

1. **Website Size and Traffic**
   - Small sites with low traffic can use shared hosting, while larger sites may need VPS or dedicated hosting.
2. **Technical Requirements**
   - Sites with advanced applications or databases might require more robust hosting options (VPS or dedicated).
3. **Budget**
   - Shared hosting is the most cost-effective but may not meet the needs of a high-traffic site. As budgets grow, VPS, cloud, or dedicated options offer better performance.
4. **Scalability**
   - Choose a host that allows for upgrades as your website grows.
5. **Ease of Use**
   - Beginners may prefer hosts with user-friendly control panels like cPanel, while experienced users may prefer more customizable options.

Web hosting is foundational to having an online presence. Choosing the right web host involves considering your website's needs, growth potential, and budget. With the right features—such as reliability, speed, security, and customer support—a good web hosting provider will ensure your website remains accessible, secure, and performs optimally for users.

2.7. SQL

MySQL is a widely used open-source relational database management system (RDBMS) that utilizes Structured Query Language (SQL) for accessing and managing data. Developed by Oracle Corporation, MySQL is known for its speed, reliability, and ease of use, making it one of the most popular database systems for web applications and software development. It supports a variety of platforms, including Windows, Linux, and macOS, and is commonly used in conjunction with programming languages like PHP, Java, and Python to create dynamic and interactive websites.

Features of MySQL:

MySQL offers a rich set of features that contribute to its popularity and effectiveness as a database management system:

1. **Open Source:** Being open-source, MySQL allows users to download, use, and modify the software freely, promoting a strong community of developers and users who contribute to its ongoing improvement.
2. **High Performance**: MySQL is optimized for high performance and can handle large volumes of data and numerous concurrent users efficiently. It includes features such as indexing, caching, and query optimization to enhance speed.

3. **Scalability:** MySQL is designed to scale easily from small applications to large enterprise solutions. It can handle databases of virtually any size, making it suitable for applications ranging from small personal projects to large-scale enterprise applications.
4. **Data Security:** MySQL provides robust security features, including user authentication, SSL support for encrypted connections, and fine-grained access control, ensuring that data is protected from unauthorized access.
5. **Data Integrity**: With support for ACID (Atomicity, Consistency, Isolation, Durability) properties, MySQL ensures data integrity even in the face of system failures or crashes. This makes it a reliable choice for applications that require consistent and accurate data.
6. **Support for Various Storage Engines**: MySQL supports multiple storage engines, allowing developers to choose the best one for their application's needs. For example, InnoDB is known for its transactional support and foreign key capabilities, while MyISAM offers fast read operations.
7. **Comprehensive Documentation and Community Support**: MySQL is backed by extensive documentation, tutorials, and a large user community, making it easier for developers to find help, resources, and best practices for database design and management.
8. **Cross-Platform Support**: MySQL can be installed on various operating systems, including Windows, Linux, and macOS, enabling developers to work in diverse environments.

Command	**Description**	**Syntax**
CREATE	Create database or its objects (table, index, function, views, store procedure, and triggers)	**CREATE** TABLE table_name (column1 data_type, column2 data_type, ...);
DROP	Delete objects from the database	**DROP** TABLE table_name;
ALTER	Alter the structure of the database	**ALTER** TABLE table_name ADD COLUMN column_name data_type;
TRUNCATE	Remove all records from a table, including all spaces allocated for the records are removed	**TRUNCATE** TABLE table_name;
COMMENT	Add comments to the	**COMMENT** 'comment_text' ON

Command	**Description**	**Syntax**
	data dictionary	TABLE table_name;
RENAME	Rename an object existing in the database	**RENAME** TABLE old_table_name TO new_table_name;

Table 2.7.1:SQL Commands syntax

MySQL Use Cases

- **Web Applications**: Many content management systems (CMS) and e-commerce platforms use MySQL to manage user data, product information, and transactions.
- **Data Warehousing**: MySQL can serve as a data warehouse, allowing organizations to store and analyze large volumes of data.
- **Business Applications**: Many enterprise resource planning (ERP) and customer relationship management (CRM) systems utilize MySQL for their backend database needs.

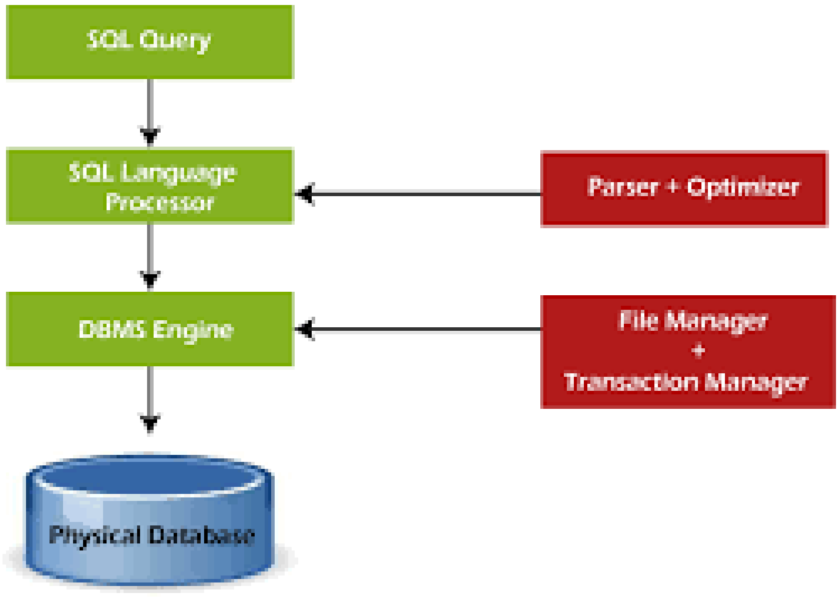


Fig 2.7: SQL working

Here are some commands that we will be learning about, and later practicing as well.

Command	Description
CREATE	Creates a new table, a view of a table, or another object in the database.
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other objects in the database.
SELECT	Retrieves certain records from one or more tables.
INSERT	Creates a record.
UPDATE	Modifies a record.
DELETE	Deletes a record.
GRANT	Gives a privilege to users.

REVOKE	Takes back privileges granted from users.

Table 2.7.2: SQL commands

If you notice, the commands 'create, alter & drop' define the complete structure of data. They come under the 'Data Definition Language' set of commands. Similarly, 'select, insert, update, delete' fall under Data Manipulation Language & 'grant, revoke' come under Data Control Language set of commands.

Let's write some SQL commands and understand what the outcomes would be like:

SELECT WHERE

```
SELECT * FROM Customers
WHERE Country='Mexico';
```

MySQL is a powerful relational database management system (RDBMS) that, when integrated with Spring Boot, enables developers to create scalable and high-performance applications efficiently. This combination allows for seamless data management and rapid application development, focusing on core business logic.

2.8. MONGODB

MongoDB is a document database. It stores data in a type of JSON format called BSON.

A record in MongoDB is a document, which is a data structure composed of key value pairs similar to the structure of JSON objects.

Records in a MongoDB database are called documents, and the field values may include numbers, strings, booleans, arrays, or even nested documents.

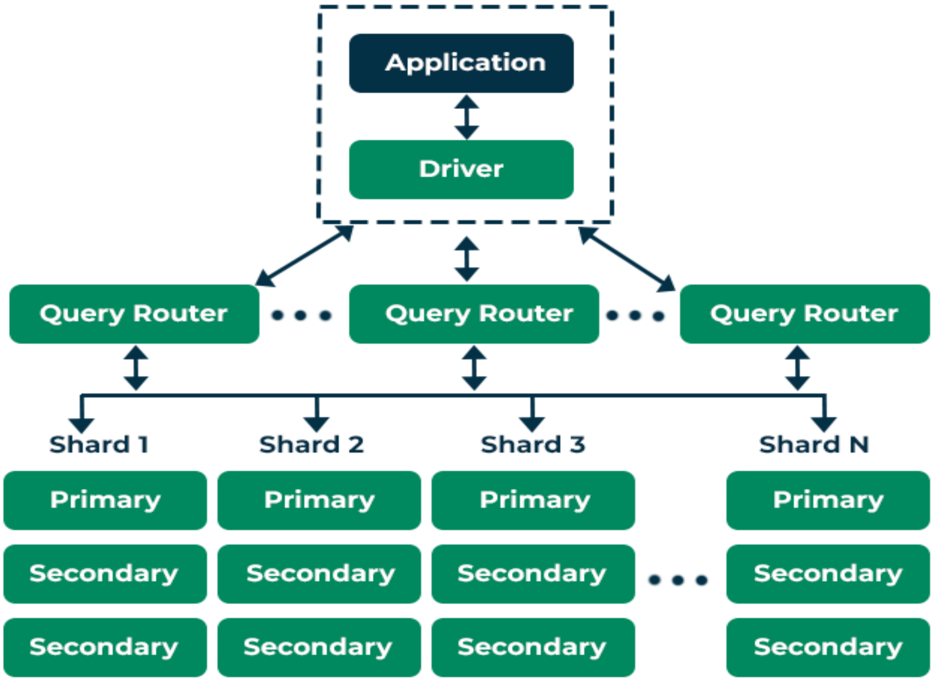


FIG 2.8: MongoDB architecture

MongoDB Query API:

The MongoDB Query API is the way you will interact with your data.

The MongoDB Query API can be used in two ways:

- CRUD Operations

- Aggregation Pipelines

MongoDB Query API Uses:
You can use the MongoDB Query API to perform:

- Adhoc queries with mongosh, Compass, VS Code, or a MongoDB driver for the programming language you use.
- Data transformations using aggregation pipelines.
- Document join support to combine data from different collections.
- Graph and geospatial queries.
- Full-text search.
- Indexing to improve MongoDB query performance.
- Time series analysis.

Advantages of MongoDB over RDBMS:

- Schema less: MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Tuning.
- Ease of scale-out: MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.
- Document Oriented Storage: Data is stored in the form of JSON style documents.
- Index on any attribute
- Replication and high availability
- Auto-sharding
- Rich queries
- Fast in-place updates
- Professional support by MongoDB

MongoDB Basic Commands:

We have a list of standard MongoDb commands to interact with the database, These commands are CREATE, READ, INSERT, UPDATE, DELETE, DROP and AGGREGATE can be classified into following groups based on their nature –

Command	Description

CREATE	Creates a new table in the database and other objects in the database.
INSERT	Inserts collection name in existing database.
DROP	Deletes an entire table or specified objects in the database.
UPDATE	Updates the document into a collection.

Table 2.8: MongoDB commands

Data Model Design

MongoDB provides two types of data models: — Embedded data model and Normalized data model. Based on the requirement, you can use either of the models while preparing your document.

Embedded Data Model

In this model, you can have (embed) all the related data in a single document, it is also known as de-normalized data model.

For example, assume we are getting the details of employees in three different documents namely, Personal_details, Contact and, Address, you can embed all the three documents in a

Normalized Data Model

In this model, you can refer the sub documents in the original document, using references.

For example, you can re-write the above document in the normalized model as:

Employee:

```
{
        _id: <ObjectId101>,
        Emp_ID: "10025AE336"
}
```

Considerations while designing Schema in MongoDB

- Design your schema according to user requirements.
- Combine objects into one document if you will use them together. Otherwise separate them (but make sure there should not be need of joins).
- Duplicate the data (but limited) because disk space is cheap as compare to compute time.
- Do joins while write, not on read.
- Optimize your schema for most frequent use cases.
- Do complex aggregation in the schema.

CHAPTER 3
REFLECTION ON LEARNING

Reflecting on my learning journey during the Full Stack Web Development internship with AICTE-Eduskills has been immensely valuable in solidifying the skills and knowledge gained. Over the course of the internship, I had the opportunity to work hands-on with a variety of web development tools, frameworks, and best practices that allowed me to grow both technically and professionally.

Technical Skills and Knowledge

Throughout the program, I strengthened my foundation in both front-end and back-end development. Working with front-end technologies such as HTML, CSS, and JavaScript helped me understand the intricacies of building responsive, user-friendly interfaces. I became proficient in libraries like React and learned how to create reusable components, manage state, and optimize performance, which made me appreciate the value of modular design and the importance of a seamless user experience.

On the back end, I dived deep into server-side programming with Node.js and Express. This experience taught me about building RESTful APIs, handling HTTP requests, and working with databases. I also became familiar with MongoDB and SQL databases, which helped me understand the pros and cons of each for different types of applications. Deploying applications was another valuable skill I gained, as I learned to set up and manage a development environment, troubleshoot issues, and ensure reliable deployments.

One of the most significant takeaways from this internship was the emphasis on problem-solving and debugging. I learned to approach bugs methodically, develop strategies to isolate problems, and apply best practices to write cleaner, more maintainable code. Moreover, working in teams improved my communication skills, taught me the importance of version control, and emphasized the need for clear documentation.

CONCLUSION

The Full Stack Web Development internship with AICTE-Eduskills has been an enriching and career-defining experience. Through structured training and hands-on projects, I was able to develop a deep understanding of end-to-end web development, from user interface design to server-side programming and database management. Working with tools like HTML, CSS, JavaScript, React, Node.js, Express, and databases like MongoDB and MySQL, I gained not only technical expertise but also practical problem-solving skills crucial for today’s fast-paced tech environment.

This internship emphasized the importance of both front-end and back-end integration, allowing me to see how different components of a web application come together. The front-end work helped me hone my skills in creating engaging, responsive, and visually appealing interfaces, while back-end tasks exposed me to concepts of server architecture, API development, data management, and security. This holistic approach has given me confidence in building scalable and efficient web applications and deepened my understanding of how users interact with technology.

One of the standout aspects of this experience was the collaborative environment fostered by AICTE-Eduskills. Working alongside talented peers and receiving mentorship from experienced developers taught me the value of effective teamwork, clear communication, and agile methodologies. Through project reviews, code assessments, and continuous feedback, I learned how to refine my work, follow industry standards, and adopt best practices in coding and version control. These skills are invaluable for a future in tech, where collaborative projects are standard, and iterative improvement is key.

This internship also offered me insights into professional development and industry expectations. I learned to manage timelines, prioritize tasks, and approach problems systematically, which are essential skills for any developer. The experience has given me a robust foundation not only in coding but also in time management, adaptability, and perseverance.