INDUSTRY INTERNSHIP REPORT

ON

"WEB DEVELOPMENT"

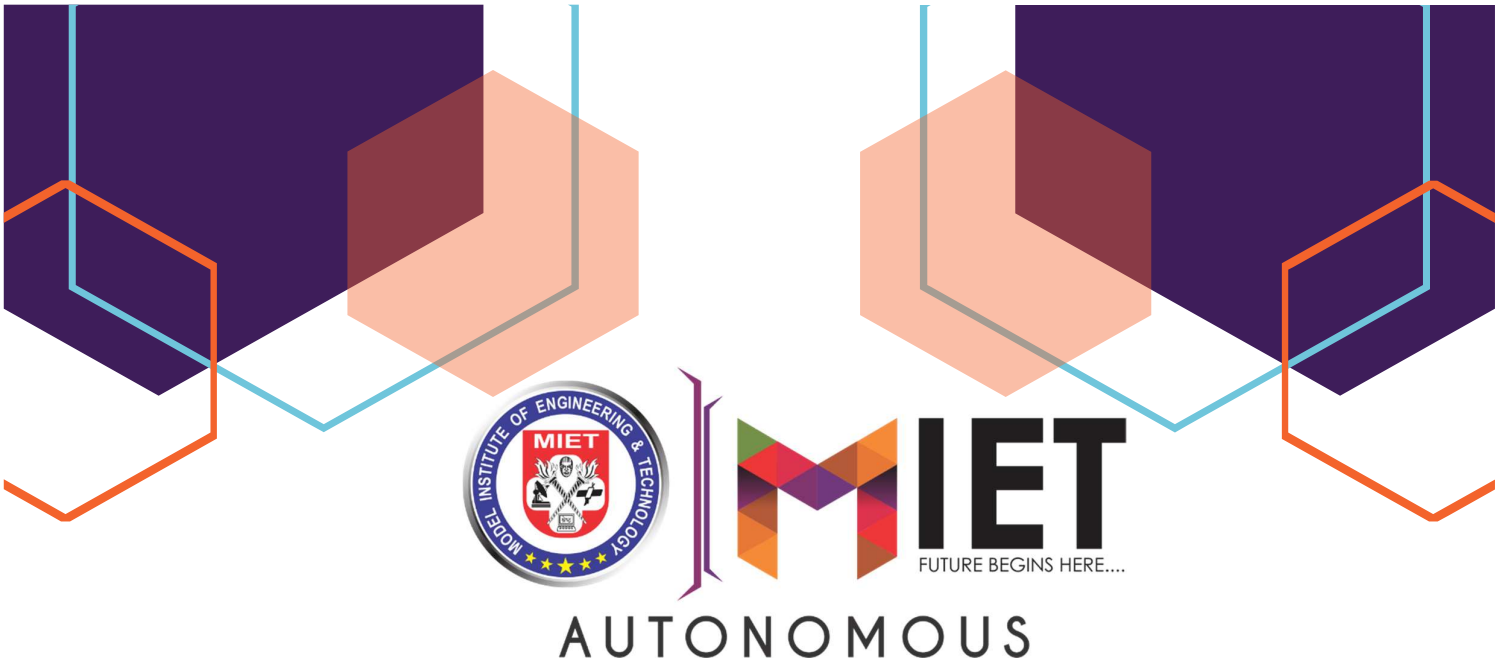
AT

NUCLEON
IIT Jammu, India

AN INDUSTRY INTERNSHIP REPORT SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING
In
Computer Science and Engineering

SUBMITTED BY
SACHIT SHARMA
2021A1R027



SUBMITTED TO

Department of Computer Science Engineering
Model Institute of Engineering and Technology (Autonomous)
Jammu, India
2023

CANDIDATES' DECLARATION

I, **Sachit Sharma, 2021A1R027,** hereby declare that the work which is being presented in the Industry Internship Report entitled, "**Web Development**" in partial fulfillment of requirement for the award of degree of B.E. (CSE) and submitted in the DepJammu isame, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me at "Nucleon IIT Jammu, India" under the supervision and mentorship of **Parmveer Nandal** Co-Founder, Nucleon, IIT Jammu. The matter presented in this report has not been submitted in this or any other University / Institute for the award of B.E. Degree.

Signature of the Student *Dated*: 9th Oct,2023

SACHIT SHARMA
2021A1R027

INTERNSHIP CERTIFICATE

Nucleon
(IIT Alumni Initiative)
x
Centre for Essential Skills
(IIT Jammu)

Certificate ID
NU/SS23/WD/0115

CERTIFICATE
OF COMPLETION

This is presented to

Sachit Sharma

for completing the "**Web Development Internship Program**"
from **June 15, 2023 - July 26, 2023**.

During the program, the student has shown great dedication and diligence towards the work.

We wish her/him the best for future endeavours.

Dr Lakhvinder Singh
Co-ordinator
CES IIT Jammu

Parmveer Nandal
Co-Founder & CEO
Nucleon

Arjun Singh Chaudhary
Co-Founder & CMO
Nucleon

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Kot Bhalwal, Jammu, India

(NAAC "A" Grade Accredited)

Ref. No.: 2021a1r027 **Date:9th October,2023**

CERTIFICATE

Certified that this Industry Internship Report entitled **"FAQ PAGE"** is the bonafide work of "**Sachit Sharma, 2021A1R027, of 5th Semester, Computer Science & Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu",** who carried out the Industry Internship at "Nucleon, IIT Jammu" work under my mentorship during June,2023- July,2023.

Ms. Shafalika Vijayal
Assistant
Professor
Department of CSE, MIET

ACKNOWLEDGEMENTS

This Summer internship opportunity was a great chance for learning and professional development. I am grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

It is my pleasant duty to pay my heartfelt gratitude to Mr. Parmveer Nandal, Founder, Nucleon, IIT Jammu who has guided me through the course of this internship. I must record my deep sense of gratitude to Prof. (Dr.) Ankur Gupta (Director, MIET) and Prof. (Dr.) Ashok Kumar (Dean Academics) for their guidance, constant inspiration and encouragement, and for their keen involvement throughout the course of present work.

Gratitude and thanks although mean a very small thing to convey my thanks to my parents who have always given me a parental source of love, motivation and strength right from the journey of my life. Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the teachers who in spite of being extraordinarily busy with their duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I perceive this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future. I express my sincere gratitude to Nucleon, IIT Jammu and Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity.

SACHIT SHARMA
2021A1R027

SELF EVALUATION

I am a 3rd-year undergraduate student majoring in Computer Science and Engineering at Model Institute of Engineering and Technology, Jammu. During my internship with Nucleon as a Web developer Intern, I gained significant insights and skills.

PROJECT OVERVIEW:

"I played a key role in our college web development project, which aimed to create a functional and user-friendly website for designing a FAQ Page."

TECHNICAL SKILLS:

"Throughout the project, I applied my knowledge of HTML, CSS, JavaScript which helped me to design and develop the website."

PROJECT MANAGEMENT:

"I was responsible for coordinating tasks and ensuring the project remained on track by setting clear objectives and milestones. This experience improved my project management skills."

TEAM COLLABORATION:

"Working in a collaborative team environment, I communicated effectively with my team members, sharing ideas, insights, and ensuring everyone was on the same page."

PROBLEM SOLVING:

"I encountered various challenges during the project and successfully resolved them by using my problem-solving skills."

QUALITY ASSRUANCE:

"I conducted rigorous testing and quality assurance, identifying and fixing bugs and inconsistencies to ensure a seamless user experience."

Sachit Sharma
2021A1R027

ABSTRACT

In the 'FAQ Page' web development project, our team aimed to create an intuitive and user-friendly Frequently Asked Questions (FAQ) page for [mention the purpose or context of the FAQ]. This project involved the application of essential web development technologies, including HTML, CSS, JavaScript, and user interface design principles. The primary objectives were to provide users with easy access to comprehensive information, improve the website's user experience, and enhance engagement. Through collaborative teamwork, meticulous design, and extensive testing, we successfully developed an interactive FAQ page that met our project goals and exceeded user expectations. This report outlines the project's key components, challenges faced, solutions implemented, and the valuable experience gained in the process.

The "FAQ Page" web development project successfully delivered an intuitive, accessible, and responsive FAQ page that addressed user needs effectively. The project not only achieved its objectives but also provided valuable experience in web development, user experience design, and collaborative teamwork.

Contents

Candidates' Declaration i

Internship Certificate ii

Certificate iii

Acknowledgement iv

Self Evaluation v

Abstract vi

Contents vii

List of Figures viii

Abbreviations Used ix

Chapter 1 HTML, CSS & JAVASCRIPT 12-27

1.1 HTML 12-18

1.1.1. Introduction 12-13

1.1.2. Syntax 13

1.1.3. Tags in HTML 14-16

1.1.4. Features in HTML 16-18

1.2 CSS 18-22

1.2.1. Introduction 18

1.2.2. Syntax 19-20

1.2.3. Features in CSS 21-22

1.3 JavaScript 23-28

1.3.1. Introduction 23

1.3.2. Syntax 24-26

1.3.3. Features in JavaScript 26-27

Chapter 2 API (Application Programming Interface) 28-32

2.1 What is an API? 28

2.2 How do APIs work? 29

2.3 API Components 30-32

Chapter 3 PROJECT DESCRIPTION 33-46

3.1 Project Description 33-35

3.2 Technology Used 35-36
3.3 Code Implementation 36-41
3.4 Output 41-44
3.5 Project Features 44-46

Chapter 4 CONCLUSIONS 47-48

REFERENCES 49

APPENDIX 50

LIST OF FIGURES

Figure No.	Caption	Page No.
1.1	HTML Syntax.	13
1.2	CSS Syntax.	19
1.3	Java Script Syntax.	24
2.1	API.	28
2.2	API Working.	29
3.1	Code of HTML SS1.	37
3.2	Code of HTML SS2.	38
3.3	Code of HTML SS3.	39
3.4	SS of CSS.	40
3.5	SS of JAVA Script.	41
3.6	Output SS1.	42
3.7	Output SS2.	43
3.8	Output SS3.	43
3.9	Output SS4.	44

ABBREVIATIONS USED

1. SV - Sorting Visualizer

2. HTML - Hypertext Markup Language

3. CSS - Cascading Style Sheets

4. JS – JavaScript

5. Git - Version Control System

6. GitHub - Web-based Git Repository Hosting

7. VS Code - Visual Studio Code

Chapter 1: HTML, CSS & JAVASCRIPT

1.1. HTML

1.1.1. Introduction:

HTML, also known as Hypertext Markup Language, plays a fundamental role in shaping the World Wide Web. It provides the groundwork for crafting web pages, organizing content, and dictating its appearance when viewed on web browsers.

HTML documents consist of elements enclosed within angle brackets, typically arranged in pairs with opening and closing tags. These elements establish a hierarchical structure that encompasses crucial components, including the declaration of the document type (usually <!DOCTYPE html>), the root element (<html>) that encapsulates the entire page, the <head> section for metadata such as the document title and character encoding, and the <body> section housing the primary content visible to users.

Within the <body> section, a diverse set of elements is available to structure and format content. Headings, ranging from <h1> to <h6>, define different levels of headings, while paragraphs are indicated by <p>. Text emphasis and strong emphasis can be achieved using and tags, respectively. Links are created with the <a> element, and images are embedded using the <img> tag. Lists come in two varieties: ordered lists (<ol>) for numbered items and unordered lists (<ul>) for bullet-pointed items, with each item designated by the <li> tag. HTML supports forms, constructed using the <form> element, which allows users to input data through a range of form controls such as text fields, radio buttons, checkboxes, and buttons.

One of HTML's fundamental functions is its capacity to establish hyperlinks, facilitating the connection of web pages through anchor tags (<a>). Additionally, comments can be inserted within HTML code using <!-- comment text -->, serving as notes and explanations for developers collaborating on the codebase. HTML forms the cornerstone of web development, and developers often combine it with CSS

(Cascading Style Sheets) for styling and layout control, as well asJavaScript for introducing interactivity and dynamic behavior to web pages.

Together, these technologies empower the creation of diverse and captivating online experiences.

1.1.2. Syntax:

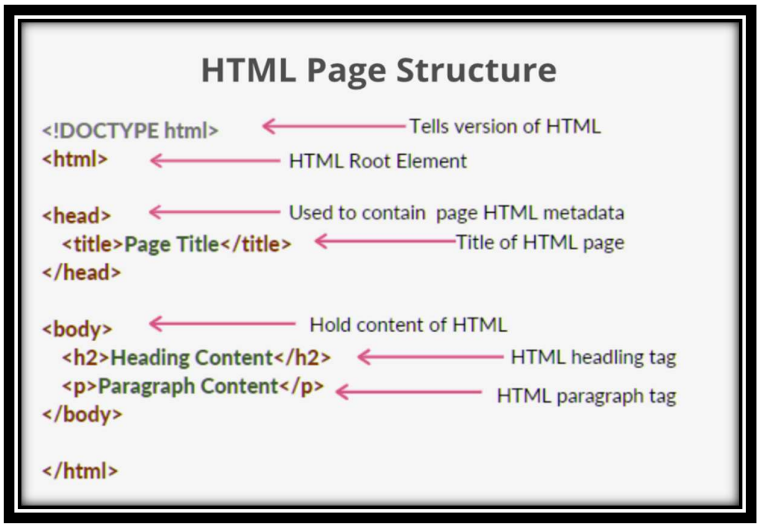


Fig1.1HTML Syntax

```
<!DOCTYPE html>
<html>
<head>
<title>Discover the Web</title>
</head>
<body>
<h1>Welcome to our online journey!</h1>
<p>Here's a brief glimpse into the world of HTML.</p>
<a href="https://www.html.com">Explore html.com</a>
</body>
</html>
```

The provided HTML code represents a foundational web page. It starts with the declaration <!DOCTYPE html>, which signals to the web browser that this document adheres to HTML5, the latest HTML version. Inside the <html> element, we establish a <head> section where we define metadata for the web page. Within the <head>

section, there's a <title> element that sets the web page's title. This title is displayed in the browser's title bar or tab. The core content of the web page resides within the <body> element. Within this section, three primary elements are utilized:

- An <h1> element, presenting a prominent heading that reads "Welcome to my website!" Typically, this is employed for the main title or heading of the page.
- A <p> element encapsulating the text "This serves as a basic HTML introduction." This constitutes a paragraph of text.
- An <a> element featuring the text "Explore html.com" and an href attribute pointing to "https://www.html.com." This creates a hyperlink, enabling users to click and visit the specified URL, which, in this case, directs them to "https://www.html.com." This code establishes a foundational structure for a web page, encompassing a title, a heading, a paragraph of text, and a hyperlink. When rendered within a web browser, it will exhibit a webpage presenting these elements. Users can click the link to navigate to the Example.com website.

This code serves as an initial introduction to fundamental HTML concepts, which are vital for web development. As this code gets rendered in a web browser, it will produce a webpage featuring a heading, a text paragraph, and a link to example.com. While HTML is the initial step in web development, creating interactive and visually appealing web pages necessitates additional skills in CSS (Cascading Style Sheets) for styling and JavaScript for incorporating interactivity.

1.1.3. Tags in HTML:

HTML (HyperText Markup Language) offers a versatile set of tags to structure and format content within web pages. Below, we provide an overview of several frequently used HTML tags and their respective functions:

1. Document Structure Tags:

- <html>: Serves as the root element that encompasses all other page elements.
- <head>: Contains document metadata like the title and character encoding.
- <title>: Establishes the webpage's title displayed in the browser's tab or title bar.
- <meta>: Offers additional document details such as character encoding or authorship.

- <link>: Specifies relationships between the current document and external resources like stylesheets.
- <style>: Holds CSS rules for document styling.
- <script>: Embeds or links to JavaScript code for interactivity.

2. Text Formatting and Structure Tags:

- Headings from <h1> (highest importance) to <h6> (lowest).
- <p>: Used for text paragraphs.
- <a>: Creates hyperlinks.
- : Represents emphasized text (typically italic).
- : Indicates strongly emphasized text (usually bold).
-
: Inserts a line break.
- <hr>: Generates a horizontal rule or thematic divider.

3. Lists:

- : Forms an unordered (bulleted) list.
- : Constructs an ordered (numbered) list.
- : Represents a list item.

4. Tables:

- <table>: Defines a table.
- <tr>: Denotes a table row.
- <th>: Identifies a table header cell.
- <td>: Marks a table data cell.
- <caption>: Provides a title or caption for a table.

5. Forms:

- <form>: Establishes a form for user input.
- <input>: Creates various input fields for forms (e.g., text, password, radio, checkbox).
- <textarea>: Allows multiline text input.
- <select>: Generates a dropdown list.

- <button>: Presents a clickable button.
- <label>: Offers labels for form elements.
- <fieldset>: Groups related form elements.
- <legend>: Supplies a title for a fieldset.

6. Media:

- <img>: Embeds images.
- <audio>: Embeds audio content.
- <video>: Embeds video content.

7. Semantic Tags (HTML5):

- <header>: Represents introductory content or a set of navigational links.
- <footer>: Represents a footer for its nearest ancestor section or article element.
- <section>: Represents a thematic grouping of content.
- <article>: Represents a self-contained composition, like a news article.
- <nav>: Represents a section of navigation links.
- <aside>: Represents content tangentially related to the surrounding content.
- <main>: Represents the document's primary content. These are key HTML tags commonly employed in web development. HTML continues to evolve, introducing new tags to enhance semantics and web accessibility.

Employing these tags effectively, along with adhering to best practices, is essential for crafting well-structured and accessible web content.

1.1.4. Features in HTML:

1. Definition and Purpose:

HTML, or HyperText Markup Language, is the foundational standard for creating web pages. It defines a web document's structure and content.

2. Markup Language:

Unlike programming languages, HTML is a markup language that relies on tags to

structure and format content on web pages.

3. Universal Compatibility:

HTML enjoys broad support from major web browsers, ensuring compatibility across various platforms.

4. Versatile Content Handling:

HTML can handle a wide range of content types, including text, images, audio, video, and more, making it suitable for multimedia-rich web pages.

5. Structural Elements:

HTML provides a set of elements to define a web page's structure, including headings (<h1>, <h2>), paragraphs (<p>), lists (<ul>, <ol>), and tables (<table>).

6. Hyperlink Creation:

The <a> (anchor) element in HTML enables the creation of hyperlinks, facilitating navigation to other web pages, files, or locations within the same page.

7. Interactive Forms:

HTML offers form-related elements (<form>, <input>, <textarea>, etc.) for building interactive web forms to collect user input.

8. Semantic Elements (HTML5):

HTML5 introduces semantic elements like <header>, <footer>, <nav>, and <article>, enhancing content structure for improved accessibility and SEO.

9. Attribute Usage:

HTML elements can include attributes that provide additional information or settings. For example, the href attribute in an <a> tag specifies a hyperlink's destination.

10. Client-Side Rendering:

HTML is processed by web browsers on the client side, enabling interactive and dynamic web content.

11. SEO-Friendly:

Proper use of semantic elements and attributes in HTML can enhance a website's search engine optimization (SEO) by improving content indexability.

12. Responsive Design:

HTML, combined with CSS and JavaScript, allows for the creation of responsive web designs that adapt to various screen sizes and devices.

13. Accessibility:

HTML supports accessibility features, including alt text for images, semantic elements, and ARIA (Accessible Rich Internet Applications) attributes, ensuring web content is accessible to users with disabilities.

14. Open Standard:

Maintained by the World Wide Web Consortium (W3C), HTML is an open standard, guaranteeing consistency and compatibility across platforms and browsers. Understanding HTML and its features is fundamental for developing effective and accessible web content.

1.2. CSS

1.2.1. Introduction:

Cascading Style Sheets, or CSS for short, holds a central position in the realm of web design and development. It serves as a foundational technology, integral to the control and customization of web page appearances when viewed in web browsers. CSS functions in tandem with HTML (Hypertext Markup Language), which defines the structure and content of web documents. At its core, CSS aims to accomplish the crucial task of decoupling a web page's content from its visual presentation.

This separation empowers web developers and designers to maintain consistency, exercise creative control, and ensure responsiveness across various devices and screen sizes. By specifying elements like fonts, colors, layouts, and spacing, CSS provides the means to construct engaging and user-friendly interfaces.

Within this report, we will embark on an exploration of CSS's essential principles, including selectors, properties, and values, and investigate how these components are applied to HTML elements. Furthermore, we will delve into the multifaceted advantages that CSS offers, encompassing enhanced design flexibility, improved accessibility, and its role in elevating not only the functionality but also the visual appeal of websites. Lastly, we will underscore the contemporary significance of CSS in the dynamic field of web development and its profound impact on shaping the digital landscape.

1.2.2. Syntax:

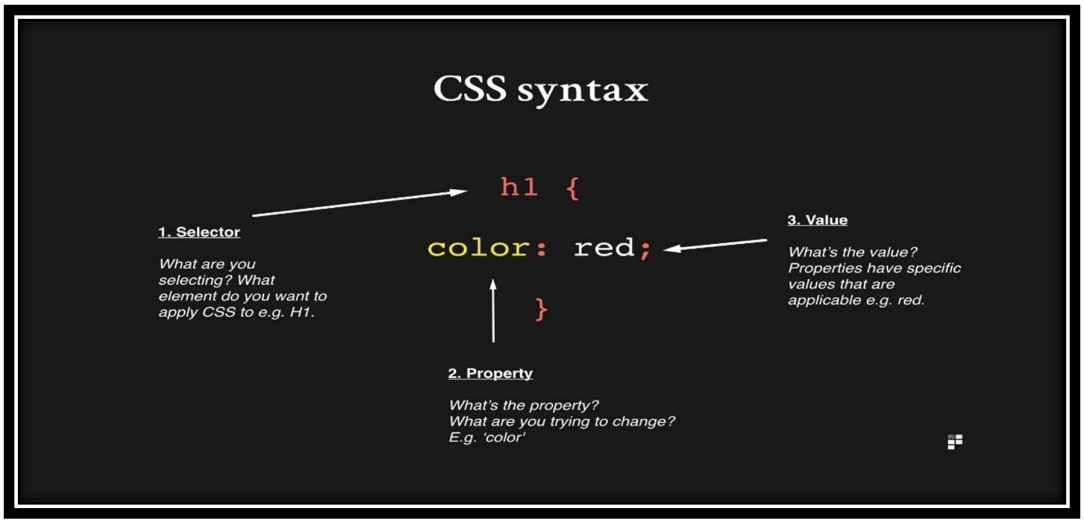


Fig 1.2 CSS Syntax.

```
/* Style the HTML <p> element */
p {
font-size: 16px; /* Font size set to 16 pixels */
color: #333; /* Text color is a dark gray (#333333) */
margin-bottom: 20px; /* Add 20 pixels of space below the <p> element */
}
/* Style the HTML <h1> element */
h1 {
font-size: 24px; /* Font size set to 24 pixels */
color: #0078d4; /* Text color is a shade of blue (#0078d4) */
margin-top: 40px; /* Add 40 pixels of space above the <h1> element */
}
/* Apply styling to an HTML class named "highlight" */
.highlight {
background-color: yellow; /* Background color set to yellow */
font-weight: bold; /* Make the text bold */
}
```

Explanation:

1. CSS Comments:

Comments are denoted by /* */. In CSS, they serve as non-executable lines that offer crucial documentation to explain the purpose or functionality of the code. They are essential for enhancing code readability and collaboration among developers.

2. Styling the HTML <p> Element:

The code begins by targeting HTML <p> elements for styling.

- font-size: 16px; sets the font size to 16 pixels, determining the size of text within <p> elements.
- color: #333; defines the text color, choosing a dark gray color with the hexadecimal value #333.
- margin-bottom: 20px; adds a 20-pixel margin below each <p> element, controlling the spacing between paragraphs.

3. Styling the HTML <h1> Element:

The code proceeds to style HTML <h1> elements.

- font-size: 24px; establishes a font size of 24 pixels for <h1> text.
- color: #0078d4; sets the text color to a shade of blue with the hexadecimal value #0078d4.
- margin-top: 40px; introduces a 40-pixel margin above each <h1> element, determining the space above the headings.

4. Applying Styling to an HTML Class Named "highlight":

The code concludes by defining styles for elements with the class "highlight."

- background-color: yellow; specifies a yellow background color for elements with the "highlight" class, creating a visually distinctive element.
- font-weight: bold; makes the text within these elements bold, enhancing its prominence and readability. This CSS code, enriched with comments, clarifies its purpose and provides detailed explanations for styling decisions.

Comments serve as valuable documentation tools, promoting code understanding, collaboration, and ease of maintenance in web development projects.

1.2.3 Features in CSS

In Cascading Style Sheets (CSS), features refer to various properties and capabilities

that allow you to control the styling and layout of web documents. CSS provides a wide range of features to design and format web content. Here are some essential CSS features:

1. Selectors:

CSS allows you to select HTML elements based on their tag names, classes, IDs, attributes, and more using selectors. This enables you to target specific elements for styling.

2. Properties:

CSS properties define how HTML elements should be styled. Properties control aspects such as color, font size, margin, padding, border, and more. elements.

3. Values:

Each CSS property accepts one or more values, specifying the specific styling to be applied. Values can be numeric, textual, or even other CSS functions.

4. Color:

CSS provides various color representations, including color names, hexadecimal codes (#RRGGBB), RGB and RGBA values, and HSL and HSLA values, allowing you to set the color of text and backgrounds.

5. Typography:

CSS allows you to control fonts, font sizes, font styles (italic, bold), text alignment, line height, and letter spacing.

6. Box Model:

CSS uses the box model to control the layout of elements, which includes properties like width, height, margin, padding, and border. The box model is crucial for controlling spacing and positioning.

7. Flexbox:

CSS Flexbox is a layout model that provides an efficient way to arrange elements within a container, making it easier to create complex layouts with responsive designs.

8. Grid Layout:

CSS Grid Layout is another layout model that offers a grid-based system for designing web layouts. It allows for precise control over column and row placement.

9. Positioning:

CSS offers properties like position, top, right, bottom, and left for precise control over element positioning. Elements can be positioned absolutely, relatively, or fixed within their containing elements.

10. Transitions and Animations:

CSS provides features to create smooth transitions and animations. You can control the timing, duration, and easing functions to achieve various animation effects.

11. Media Queries:

CSS supports media queries, which allow you to apply different styles based on the device's characteristics, such as screen size, orientation, and resolution. This is crucial for creating responsive designs.

12. Pseudo-Classes and Pseudo-Elements:

CSS pseudo-classes like :hover, :active, and :focus let you apply styles to elements in specific states or conditions. Pseudo-elements like ::before and ::after create virtual elements for additional styling.

13. Transforms and Transitions:

CSS provides properties like transform and transition to manipulate and animate elements, allowing for effects like scaling, rotating, and translating.

14. Custom Properties (CSS Variables):

CSS custom properties, often referred to as CSS variables, allow you to define reusable values in your stylesheets, improving maintainability and flexibility.

15. Gradients:

CSS gradients enable the creation of smooth color transitions within elements, allowing for various gradient styles and directions.

16. Filters:

CSS filters, such as blur, brightness, contrast, and sepia, provide visual effects that can be applied to images and elements. These are just some of the many features that CSS offers for styling and layout control on the web. CSS is a powerful language that enables you to create visually appealing and responsive web designs.

1.3. JAVASCRIPT

1.3.1. Introduction:

JavaScript, often referred to as "the language of the web," is a pivotal programming

language that has profoundly influenced the landscape of modern web development. It was conceived by Brendan Eich and introduced in 1995. JavaScript stands as a linchpin in the digital realm, playing a central role in endowing websites and web applications with interactivity and dynamism.

Its uniqueness lies in its capacity to execute directly within web browsers, empowering developers to craft engaging user experiences. At its core, JavaScript is versatile, transcending its traditional role as a client-side scripting language. While it undoubtedly powers interactive web pages, its applications extend far beyond this domain. JavaScript finds a home in server-side scripting, thanks to platforms like Node.js, enabling developers to create scalable and efficient web servers.

It is also a driving force in the realm of mobile app development, with frameworks like React Native facilitating the creation of cross-platform applications. Additionally, JavaScript bridges the gap between web and desktop applications through technologies like Electron. JavaScript's ability to manipulate the Document Object Model (DOM) is another hallmark feature.

This capability allows developers to dynamically alter a web page's structure and content, leading to responsive and interactive user interfaces. Furthermore, JavaScript excels in asynchronous programming, enabling the execution of tasks without blocking the main thread, resulting in smoother and more responsive web applications. In essence, JavaScript is a versatile and powerful language that underpins the dynamic, interactive, and user-centric nature of the web, transcending boundaries to impact diverse domains within the world of technology.

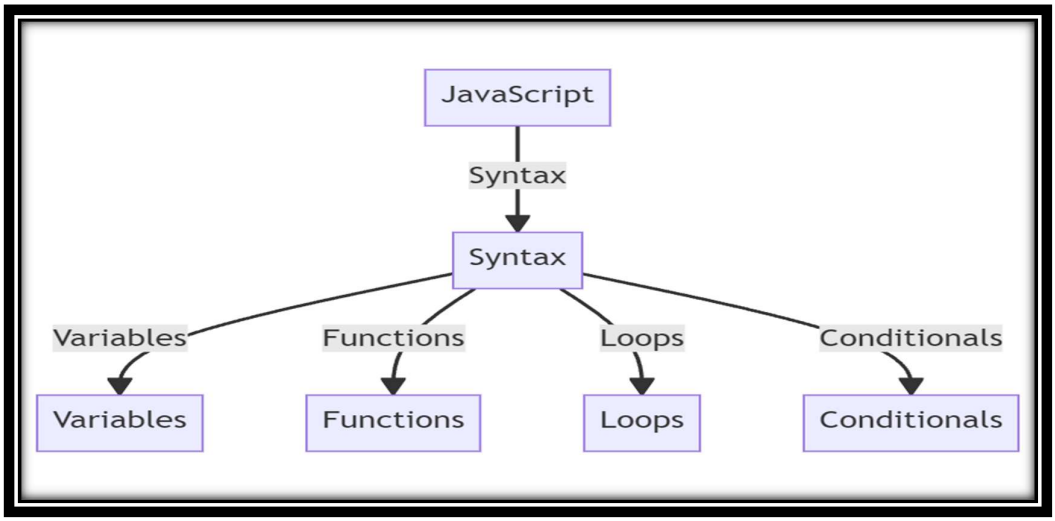


Fig 1.3 Java Script Syntax.

1.3.2. Syntax

```
// Comments:
// Single-line comment
/* Multi-line comment */
// Variables:
var name = "Rishita"; // Declare and assign a variable (var)
let age = 20; // Declare block-scoped variable (let)
const PI = 3.14159; // Declare a constant (const)
// Data Types:
let number = 42; // Number
let text = "Hello"; // String
let isTrue = true; // Boolean
let person = { // Object
name: "Raja", age: 20
};
let fruits = ["apple", "banana", "cherry"]; // Array
// Operators:
let result = 10 + 5; // Addition
let isEqual = 10 === 5; // Equality check (false)
let isAnd = true && false; // Logical AND (false)
// Functions:
function greet(name) {
return "Hello, " + name + "!";
}
// Conditional Statements:
if (age >= 18) {
console.log("You are an adult.");
} else {
console.log("You are a minor.");
}
// Loops:
for (let i = 0; i < 3; i++)
{ console.log("Iteration " + i);
```

```
}
// Objects and Properties:
let person =
{ name: "John", age:
25
};
console.log(person.name); // Access object property using dot notation
console.log(person["age"]); // Access object property using bracket notation
// Events and Event Handlers:
document.getElementById("myButton").addEventListener("click", function()
{ alert("Button clicked!");
});
```

Explanation:

● Comments:

Comments are used for adding explanatory notes to your code. Single-line comments start with //, and multi-line comments are enclosed in
/* */.

● Variables:

Variables are used to store data. var, let, and const are used for variable declaration. var has a function scope, let has a block scope, and const is used for constants.

● Data Types:

JavaScript supports various data types, including numbers, strings, booleans, objects, and arrays.

- **Operators:** Operators perform operations on variables and values. Common operators include arithmetic, comparison, and logical operators.
- **Functions:** Functions are reusable blocks of code that can take parameters and return values.
- **Conditional Statements:** Conditional statements (e.g., if and else) allow you to execute different code blocks based on conditions.
- **Loops:** Loops (e.g., for) are used to execute a block of code repeatedly.

- **Objects and Properties:** Objects store data as key-value pairs. You can access object properties using dot notation (object.property) or bracket notation (object["property"]).
- **Events and Event Handlers:** The code demonstrates event handling. In this case, it listens for a button click event and displays an alert when the button is clicked.

This JavaScript syntax overview covers fundamental elements that are commonly used in JavaScript programming, making it a foundational reference for writing JavaScript code.

1.3.3. Features in JavaScript:

JavaScript is a versatile and powerful programming language with a wide range of features that make it suitable for various applications. Here are some key features of JavaScript:

1. Cross-Platform Compatibility:

JavaScript is a client-side scripting language that runs in web browsers, making it compatible with multiple platforms and operating systems.

2. Ease of Learning:

JavaScript has a relatively simple and intuitive syntax, making it accessible to both beginner and experienced developers.

3. Interactivity:

JavaScript allows you to add interactivity to websites and web applications by responding to user actions such as clicks, input, and mouse movements.

4. Asynchronous Programming:

JavaScript supports asynchronous programming through callbacks, promises, and async/await, making it suitable for handling tasks like fetching data from servers without blocking the main thread.

5. Dynamic Typing:

JavaScript is dynamically typed, meaning you don't need to declare variable types explicitly. Variables can change types during runtime.

6. First-Class Functions:

JavaScript treats functions as first-class citizens, allowing you to pass functions as

arguments to other functions, return functions from functions, and assign them to variables.

7. Closures:

JavaScript supports closures, which allow functions to maintain access to variables from their containing scope even after the outer function has finished executing.

8.Prototype-based Object-Oriented Programming:

JavaScript uses a prototype-based inheritance model, where objects can inherit properties and methods from other objects, facilitating object-oriented programming.

9. Extensive Standard Library:

JavaScript comes with a rich standard library that includes functions for manipulating strings, arrays, dates, and handling asynchronous operations.

10. DOM Manipulation:

JavaScript can manipulate the Document Object Model (DOM), enabling dynamic updates and modifications to web page content, structure, and styles.

11. Libraries and Frameworks:

A vast ecosystem of libraries and frameworks, such as React, Angular, and Vue.js, enhances JavaScript's capabilities and simplifies complex web development tasks.

12. JSON (JavaScript Object Notation):

JavaScript supports JSON, a lightweight data interchange format, making it easy to exchange data between a server and a web application.

13. Security:

JavaScript has built-in security features like the Same-Origin Policy (SOP) to prevent cross-site scripting (XSS) attacks, ensuring the safety of web applications.

14. Browser Compatibility:

JavaScript is supported by all major web browsers, making it a reliable choice for web development.

15. Community and Support:

JavaScript has a vast and active developer community, providing resources, documentation, and continuous updates to the language.

16. Server-Side Development:

With technologies like Node.js, JavaScript can be used for server-side development, allowing developers to build entire web applications using a single language.

Chapter2: API(Application Programming Interface)

2.1.1 API

2.1 What is an API?

An API is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.

The application programming interface must be clearly distinguished from a user interface. The user interface accepts data from users, forwards it to the application for processing, and returns the results to the user. The API does not interact with the user, but processes the data received from one program module and transmits the results back to the other module. Here's how it happens.

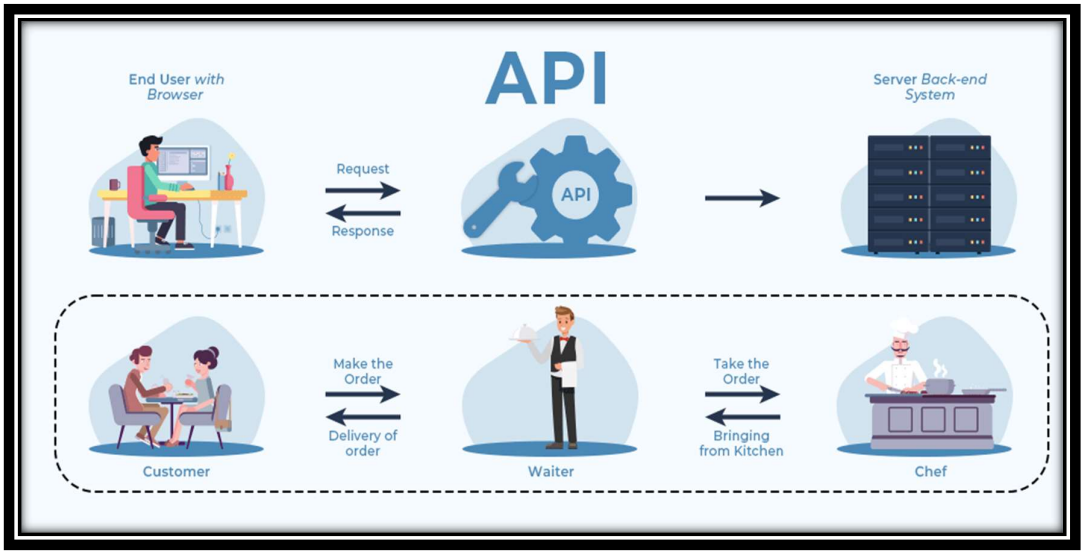


Fig 2.1API.

2.2 How do APIs work?

The working principle of an API is commonly expressed through the request-response communication between a client and a server. The client is any front-end application that a user interacts with. The server is in charge of backend logic and database operations. In this scenario, an API works as a middle layer between the client and the server, making it possible to send data requests and responses.

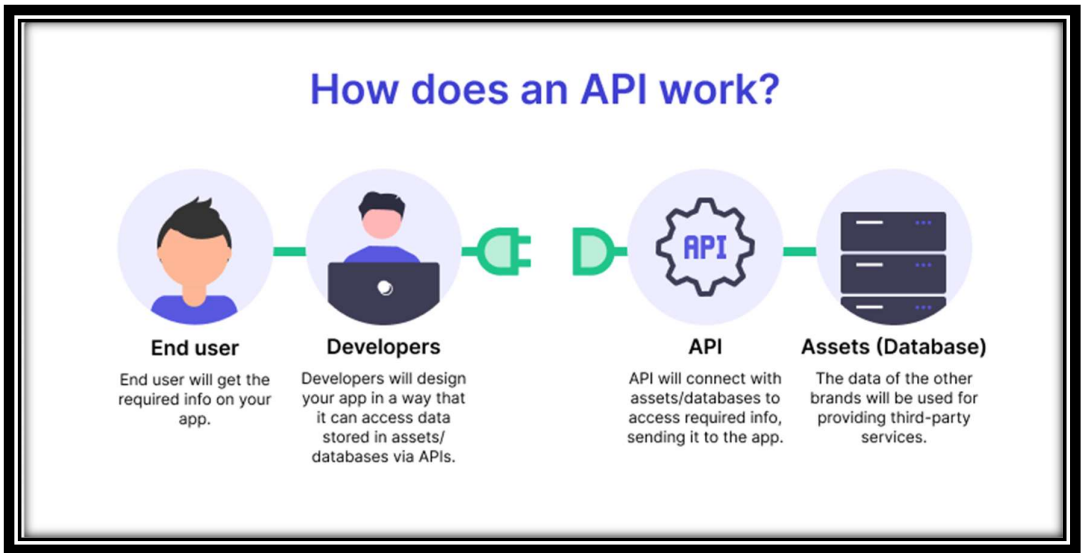


Fig2.2 API Working.

2.3 API components

Application programming interfaces consist of two components: *technical specification*, describing the data exchange options between solutions with the specification done in the form of a request for processing and data delivery protocols; and *software interface*, written to the specification that represents it The software that needs to access information (i.e., X hotel room rates for certain dates) or functionality (i.e., a route from point A to point B on a map based on user's location) from another software calls its API while specifying the requirements of how data/functionality must be provided. The other software returns data/functionality requested by the former application. And the interface by which these two applications communicate is what the API specifies.

API function calls:

Each API contains and is implemented by **function calls** – language statements that request software to perform particular actions and services. Function calls are composed of verbs (i.e., BEGIN, GET, DELETE, etc.) and nouns (i.e., Data, Access, etc.) that enable a machine to understand what to do next, say, start or finish a session, get amenities for a single room type, or restore or retrieve objects from a server. Function calls and other parameters are described in the API documentation — a manual for developers that includes all necessary information on how to work with

the API and use the services it provides. We will talk more about the documentation in one of the following sections.

As for now, we'll focus on what types of APIs exist, their use cases, and who each type is suitable for.

Types of APIs:

There are different types of APIs that can be categorized based on the ways they are available for use and according to their initial design purposes. APIs by availability aka release policies

In terms of release policies, APIs can be private, partner, and public.

Private APIs.:

These application software interfaces are designed to improve organizational solutions and services. In-house developers or contractors may use these APIs to integrate a company's IT systems or applications as well as build new systems or customer-facing apps leveraging existing systems.

Even if apps are publicly accessible, the interface itself remains available only for those working directly with the API publisher. The private strategy allows a company to fully control API usage.

Partner APIs.:

This type of API is openly promoted but shared with business partners who have signed an agreement with the publisher. The common use case for partner APIs is software integration between two parties. A company that provides partners with access to data or capability benefits from extra revenue streams. At the same time, it can monitor how the exposed digital assets are used, ensure whether third-party solutions using their APIs provide decent user experience, and maintain corporate identity in their apps.

Public APIs.:

Also known as developer-facing or external, these APIs are available for any third-

party developers. A public API program allows for increasing brand awareness and receiving an additional source of income when properly executed.

There are *two types of public APIs* – open (free of charge) and commercial ones.

Open public APIs, as the Open API Definition suggests, are those with all features public and available for use without restrictive terms and conditions. For instance, it's possible to build an application that utilizes the API without explicit approval from the API supplier or mandatory licensing fees. The definition also states that the API description and any related documentation must be openly available. On top of that, these APIs can be freely used to create and test applications.

Commercial API users pay subscription fees or use APIs on a pay-as-you-go basis. A popular approach among publishers is to offer free trials, so users can evaluate APIs before purchasing subscriptions. Learn more about how businesses benefit from opening their APIs for public use in our detailed article on the API economy.

Database APIs.:

Database APIs enable communication between an application and a database management system. Developers work with databases by writing queries to access data, change tables, etc. The Drupal 7 Database API, for example, allows users to write unified queries for different databases, both proprietary and open source (Oracle, MongoDB, PostgreSQL, MySQL, CouchDB, and MSSQL).

Another example is ORDS database API, which is embedded in Oracle REST Data Services.

Operating systems APIs.

This group of APIs defines how applications use the resources and services of operating systems. Every OS has its set of APIs, for instance, Windows API or Linux API (kernel user-space API and kernel internal API).

Apple provides API references for macOS and iOS in its developer documentation. APIs for building applications for Apple's macOS desktop operating system are included in the Cocoa set of developer tools. Those building apps for the iOS mobile operating system use Cocoa Touch – a modified version of Cocoa.

Remote APIs.:

Remote APIs define standards of interaction for applications running on different machines. In other words, one software product accesses resources located outside the device that requests them, which explains the name. Since two remotely located applications are connected over a communications network, particularly the internet, most remote APIs are written based on web standards. Java Database Connectivity API and Java Remote Method Invocation API are two examples of remote application programming interfaces.

Web APIs.:

This API class is the most common. Web APIs provide machine-readable data and functionality transfer between web-based systems which represent client-server architecture. These APIs mainly deliver requests from web applications and responses from servers using Hypertext Transfer Protocol (HTTP).

Developers can use web APIs to extend the functionality of their apps or sites. For instance, the Pinterest API comes with tools for adding users' Pinterest data like boards or Pins to a website. Google Maps API enables the addition of a map with an organization's location.

Chapter 3: PROJECT DESCRIPTION

3.1 Project Description:

Certainly, here's a detailed project description for an FAQ page web development project:

Project Title: FAQ Page Web Development Project

Project Overview:

In today's digital landscape, websites serve as crucial platforms for organizations to connect with their audiences and disseminate information. An integral component of web development is the creation and implementation of a "FAQ Page," which stands for Frequently Asked Questions. This project focuses on designing and developing an efficient, user-friendly FAQ page for [Your Organization/Website Name]. The FAQ page will serve as a valuable resource for users, providing quick access to answers for common queries and enhancing their overall experience on the website.

Project Goals:

1. **Information Accessibility**:

The primary goal of this project is to ensure that essential information is easily accessible to users, allowing them to find answers to frequently asked questions without the need for extensive searches or inquiries.

2. **User-Centered Design**:

The FAQ page will be designed with a strong focus on user experience, ensuring a seamless and intuitive interface for visitors.

3. **Mobile Responsiveness**:

In a world where users access websites through various devices, the FAQ page must be responsive and mobile-friendly to provide a consistent experience across platforms.

4. **Content Accuracy**:

Maintaining the accuracy and timeliness of the FAQ content is paramount to the effectiveness of the page.

5. **Accessibility**:

Compliance with web accessibility standards, such as the Web Content Accessibility Guidelines (WCAG), is a critical aspect of this project to ensure that the FAQ page is usable by all.

6. **SEO Optimization**:

The FAQ page will be optimized for search engines, making it more discoverable by users and improving its visibility in search engine results.

Project Scope:

This project encompasses the design, development, and deployment of an informative FAQ page as a core feature of the [Your Organization/Website Name] website. The FAQ page will include an organized structure, interactive elements, and a user-friendly interface to facilitate easy navigation and quick access to answers.

Limitations:

While this project aims to create a robust FAQ page, certain limitations need to be acknowledged:

1. **Content Maintenance**:

Although strategies will be implemented for maintaining up-to-date content, challenges may arise as new questions emerge and updates are required.

2. **Browser Compatibility**:

The FAQ page will be optimized for modern web browsers, and compatibility with older or less common browsers may not be perfect.

3. **Hardware Constraints**:

The responsiveness of the FAQ page depends on the capabilities of the user's device and internet connection.

**Deliverables: **

The primary deliverable of this project is a fully functional and user-friendly FAQ page integrated into the [Your Organization/Website Name] website.

**Project Timeline: **

The project will be carried out over [Specify the Project Duration], commencing on [Start Date] and concluding on [End Date]. A detailed project timeline will be developed during the planning phase.

3.2 Technology Used:

1. HTML (Hypertext Markup Language):

HTML is the fundamental language for structuring the content of webpages. It provides the basic layout and organization of the FAQ page.

2. CSS (Cascading Style Sheets):

CSS is used to style and format the FAQ page, making it visually appealing and responsive to different screen sizes.

3. JavaScript:

JavaScript is essential for adding interactivity and dynamic behavior to the FAQ page. It can be used for creating collapsible sections (accordions), search functionality, and more.

4. Front-End Frameworks:

Consider using front-end frameworks like Bootstrap, Foundation, or Bulma, which provide pre-designed components and responsive grids to speed up development.

5. Visual Studio Code:

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

6. Version Control System (e.g., Git):

Using version control is crucial for tracking changes in your code, collaborating with team members, and ensuring code reliability.

7. Responsive Design Tools:

To make your FAQ page mobile-friendly, you can use media queries in CSS to create responsive layouts.

8. Search Engine Optimization (SEO) Tools:

SEO is crucial for making your page discoverable by search engines. Tools like Google Analytics, Google Search Console, and SEO plugins can help optimize your content.

3.3 Code Implementation:

HTML CODE 1:

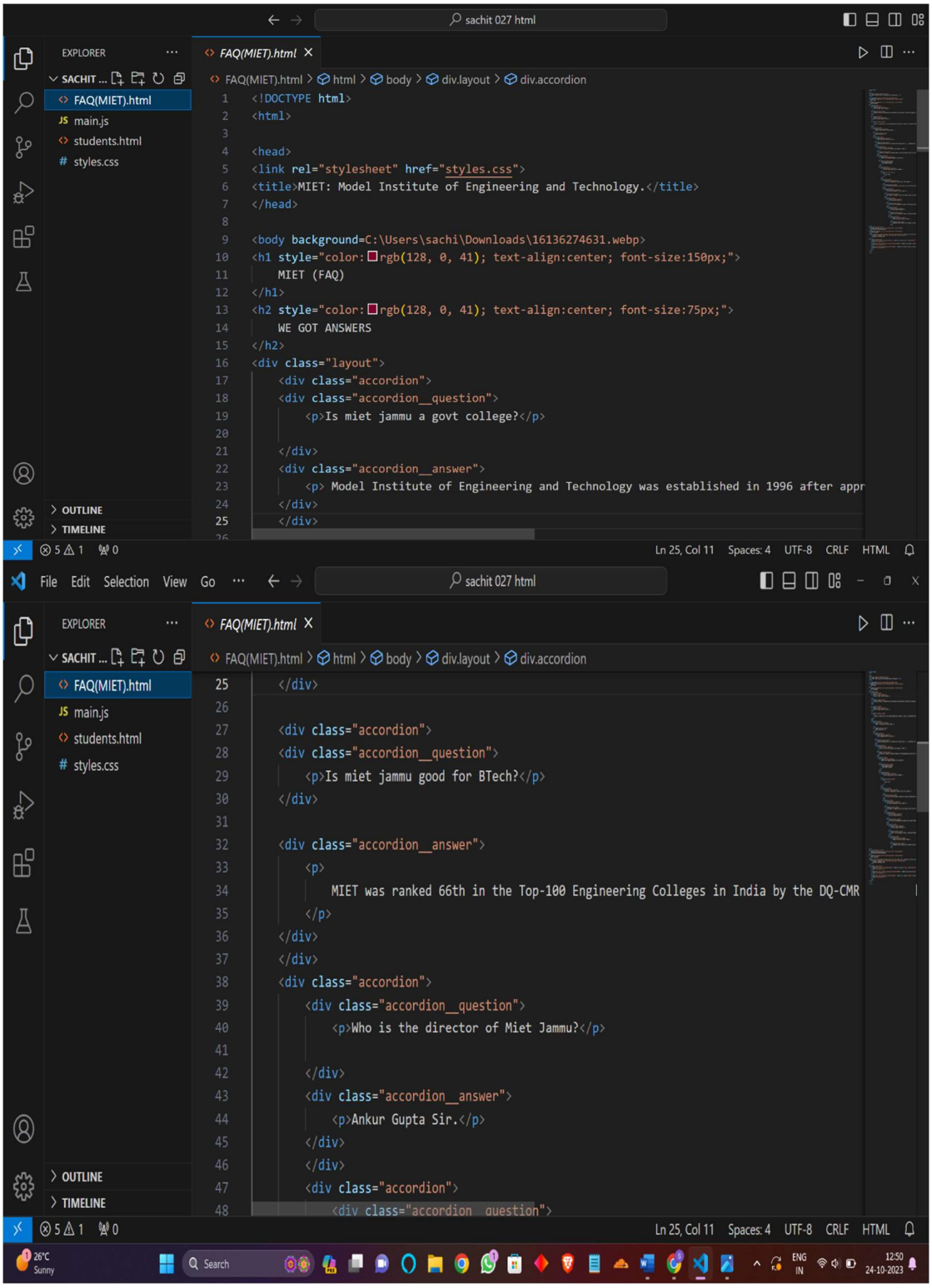


Fig3.1 Code of HTML SS1

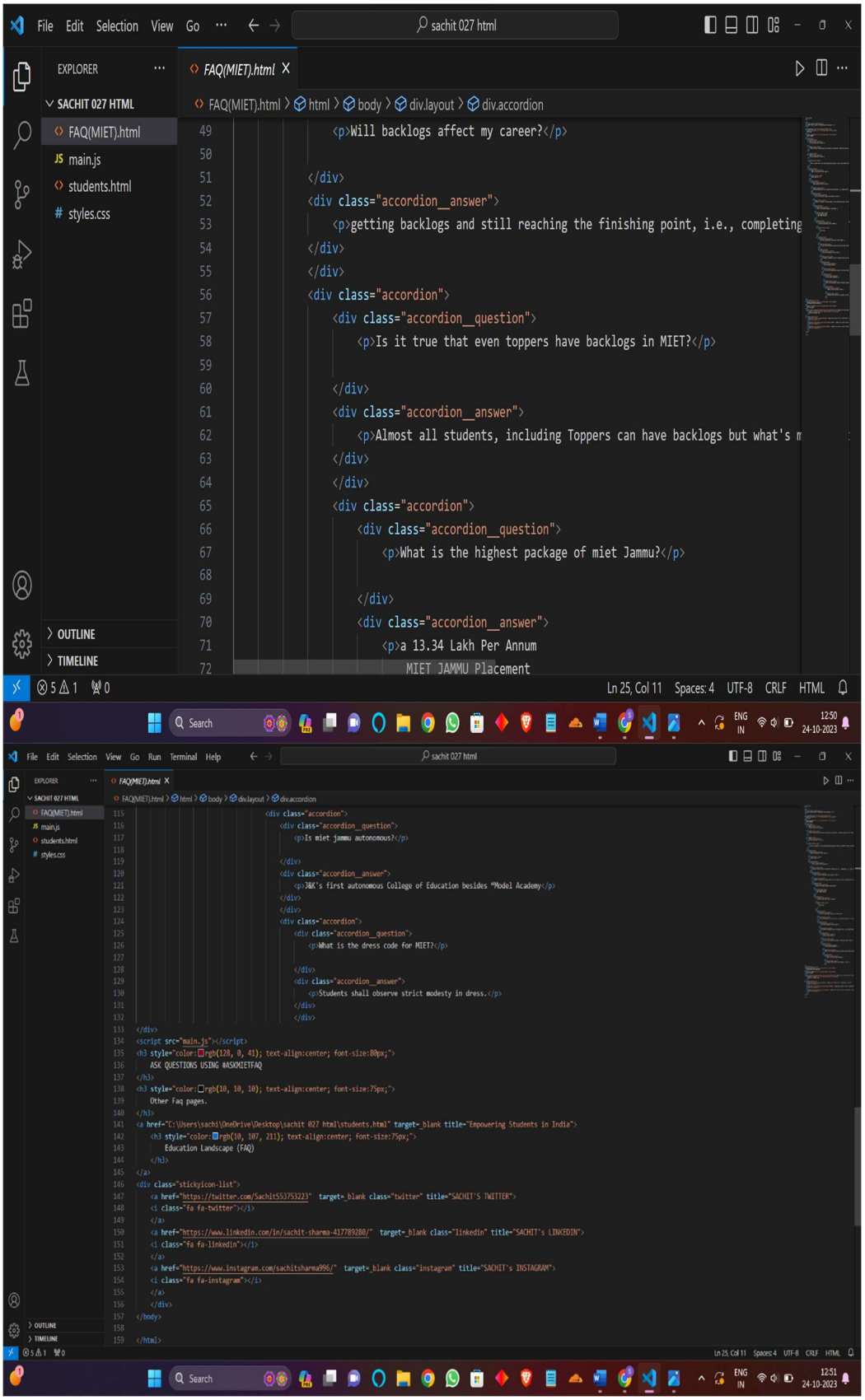
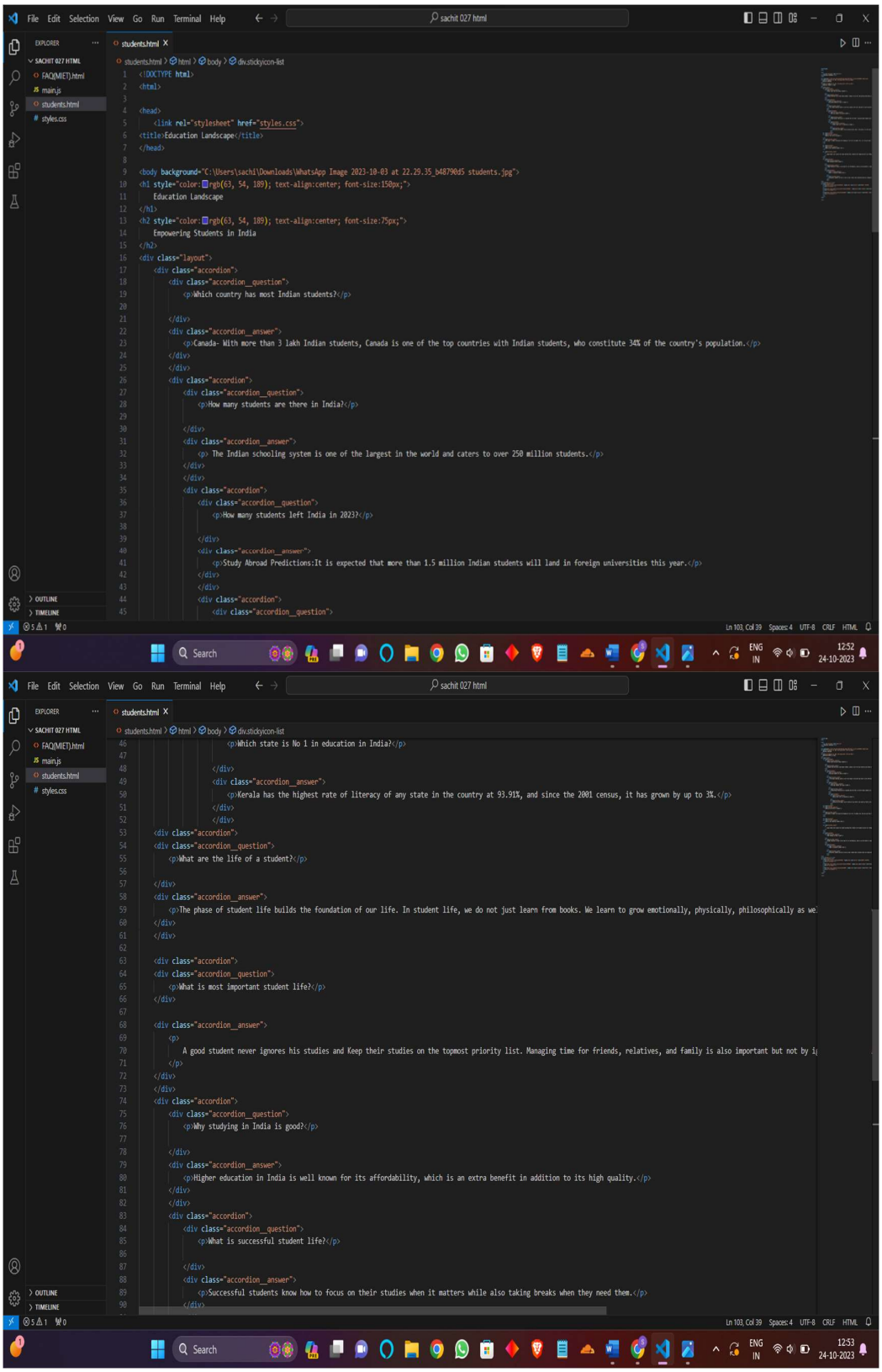


Fig3.2Code of HTML SS2

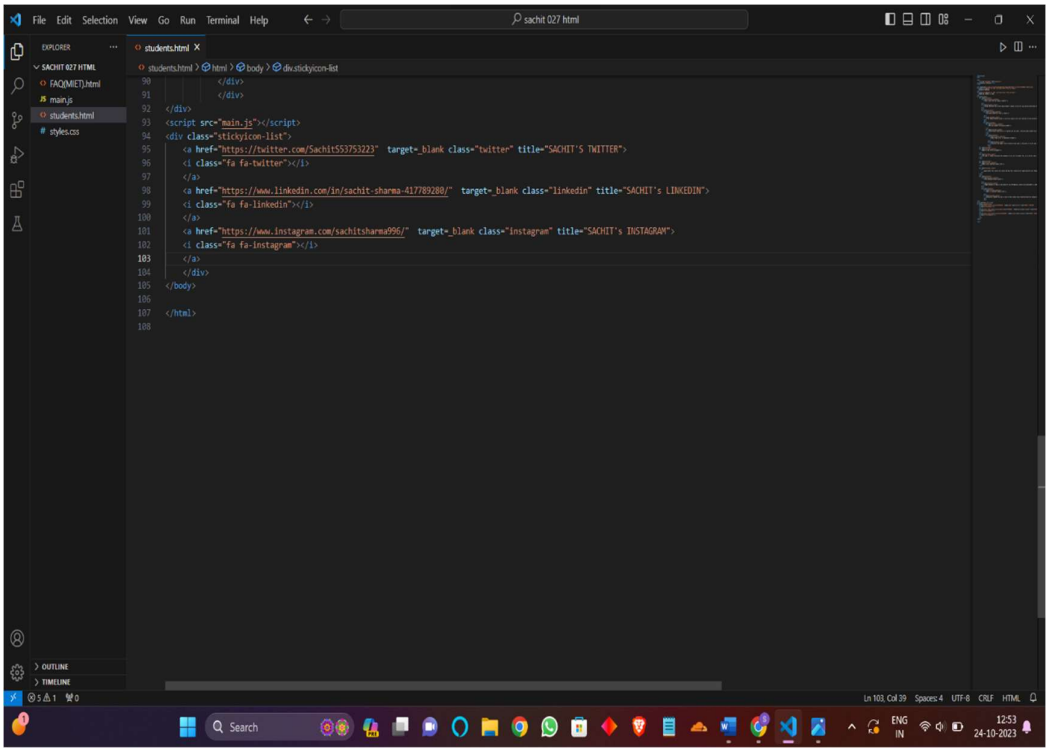
HTML CODE 2:



```
<!DOCTYPE html>
<html>

<head>
    <link rel="stylesheet" href="styles.css">
<title>Education Landscape</title>
</head>

<body background="C:\Users\sachi\Downloads\WhatsApp Image 2023-10-03 at 22.29.35_b48790d5 students.jpg">
<h1 style="color:rgb(63, 54, 189); text-align:center; font-size:150px;">
    Education Landscape
</h1>
<h2 style="color:rgb(63, 54, 189); text-align:center; font-size:75px;">
    Empowering Students in India
</h2>
<div class="layout">
    <div class="accordion">
        <div class="accordion__question">
            <p>Which country has most Indian students?</p>

        </div>
        <div class="accordion__answer">
            <p>Canada- With more than 3 lakh Indian students, Canada is one of the top countries with Indian students, who constitute 34% of the country's population.</p>
        </div>
        </div>
        <div class="accordion">
            <div class="accordion__question">
                <p>How many students are there in India?</p>

            </div>
            <div class="accordion__answer">
                <p> The Indian schooling system is one of the largest in the world and caters to over 250 million students.</p>
            </div>
            </div>
            <div class="accordion">
                <div class="accordion__question">
                    <p>How many students left India in 2023?</p>

                </div>
                <div class="accordion__answer">
                    <p>Study Abroad Predictions:It is expected that more than 1.5 million Indian students will land in foreign universities this year.</p>
                </div>
                </div>
                <div class="accordion">
                    <div class="accordion__question">
```

```
                        <p>Which state is No 1 in education in India?</p>

                    </div>
                    <div class="accordion__answer">
                        <p>Kerala has the highest rate of literacy of any state in the country at 93.91%, and since the 2001 census, it has grown by up to 3%.</p>
                    </div>
                    </div>
    <div class="accordion">
    <div class="accordion__question">
        <p>What are the life of a student?</p>

    </div>
    <div class="accordion__answer">
        <p>The phase of student life builds the foundation of our life. In student life, we do not just learn from books. We learn to grow emotionally, physically, philosophically as we
    </div>
    </div>

    <div class="accordion">
    <div class="accordion__question">
        <p>What is most important student life?</p>
    </div>

    <div class="accordion__answer">
        <p>
            A good student never ignores his studies and Keep their studies on the topmost priority list. Managing time for friends, relatives, and family is also important but not by i
        </p>
    </div>
    </div>
    <div class="accordion">
        <div class="accordion__question">
            <p>Why studying in India is good?</p>

        </div>
        <div class="accordion__answer">
            <p>Higher education in India is well known for its affordability, which is an extra benefit in addition to its high quality.</p>
        </div>
        </div>
        <div class="accordion">
            <div class="accordion__question">
                <p>What is successful student life?</p>

            </div>
            <div class="accordion__answer">
                <p>Successful students know how to focus on their studies when it matters while also taking breaks when they need them.</p>
            </div>
```

Fig3.3Code of HTML SS3



CSS CODE:

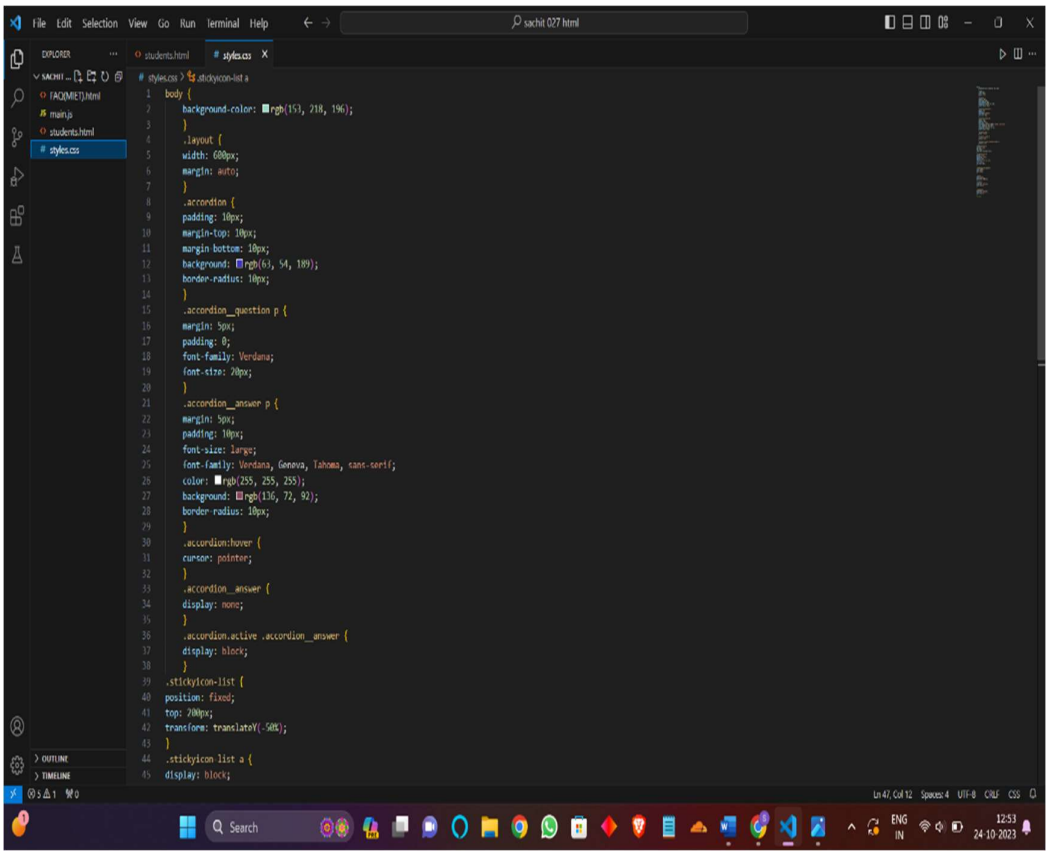
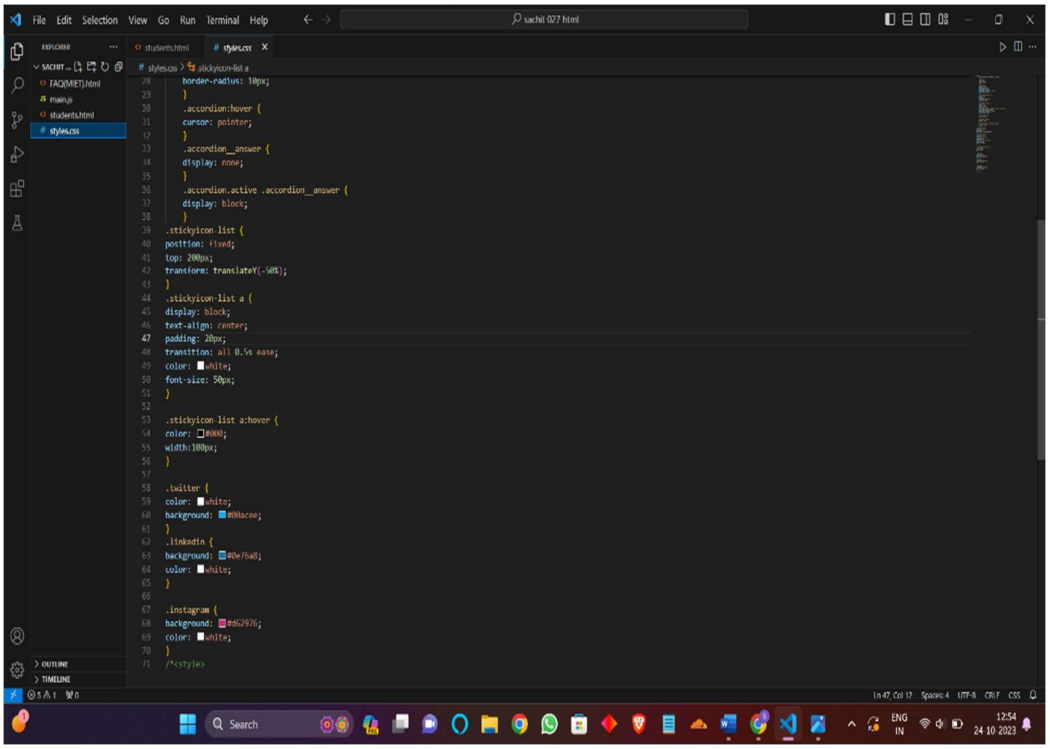


Fig3.4 SS of CSS



Java Script Code:

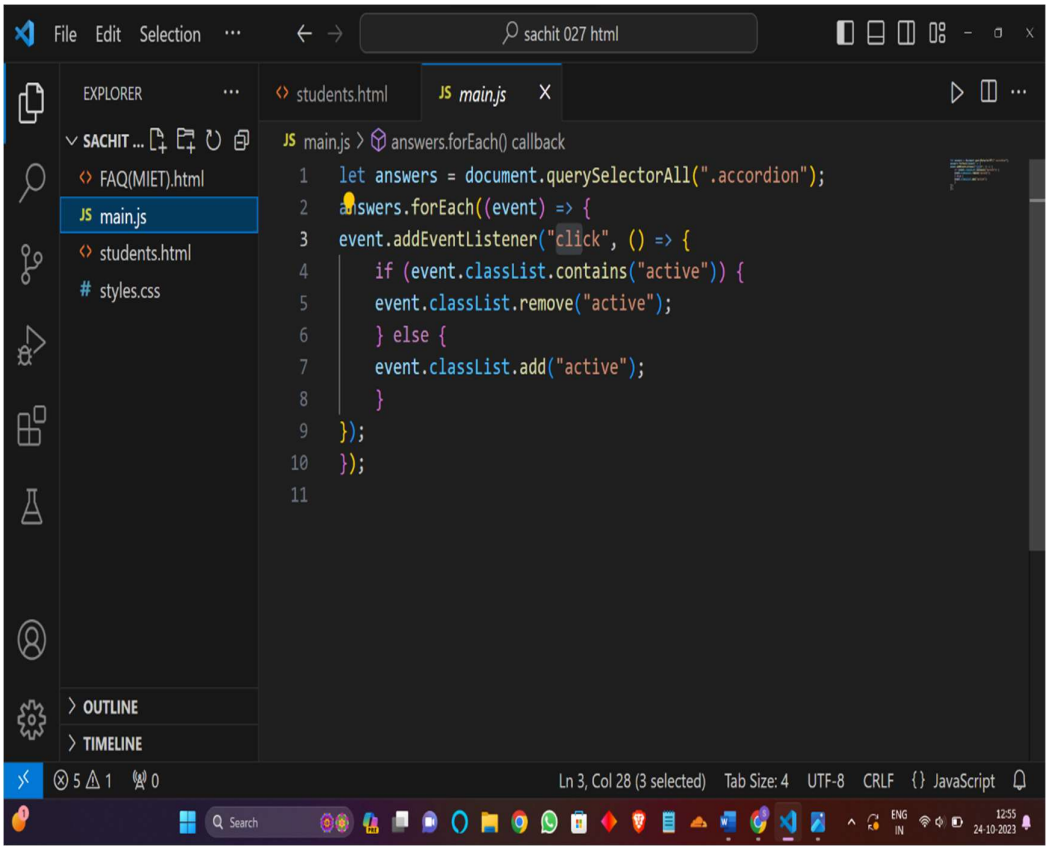


Fig3.5 SS of JAVA Script

3.4 OUTPUTS:

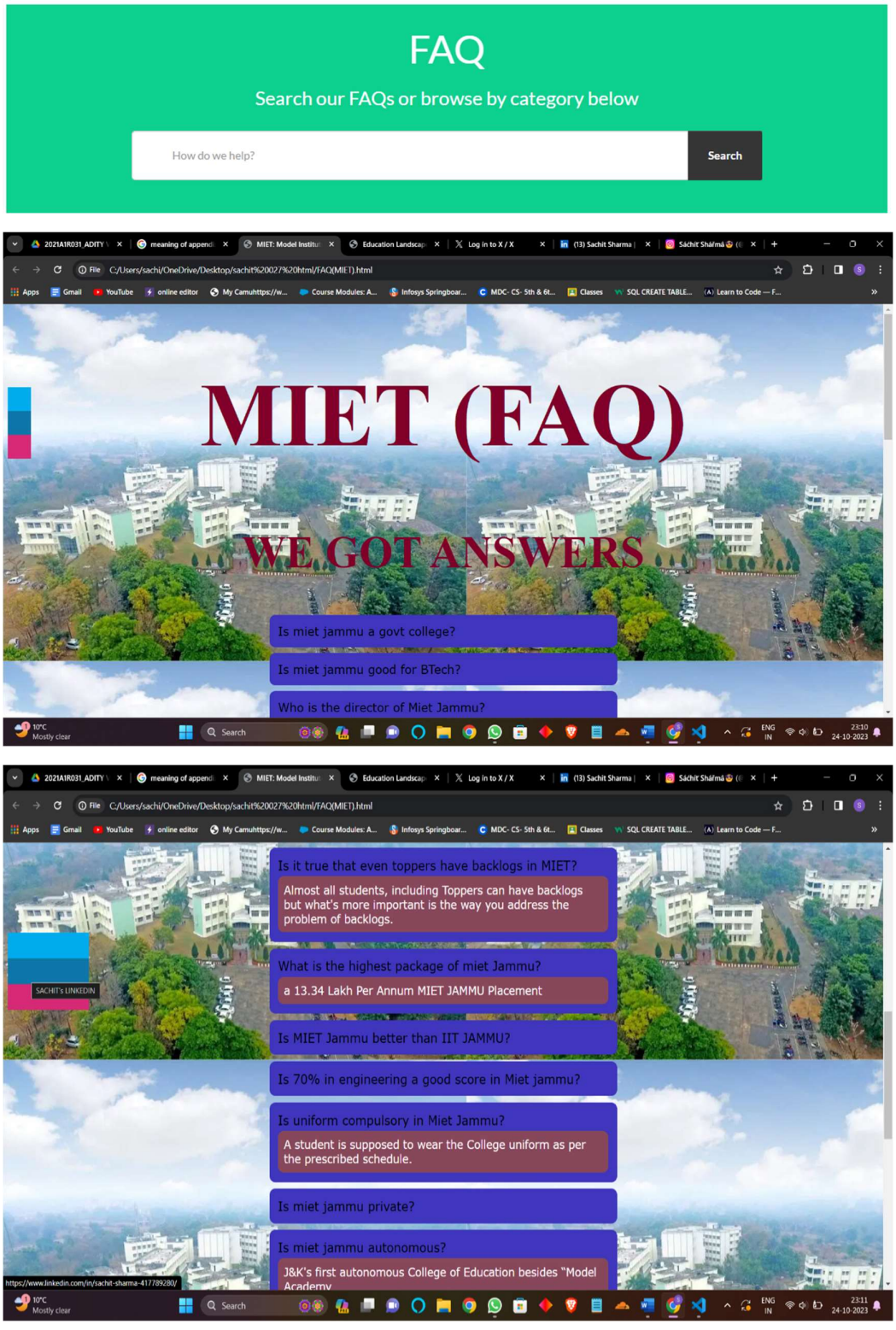


Fig3.6 Output SS1

Fig3.1 Output SS2

Fig3.8 Output SS3

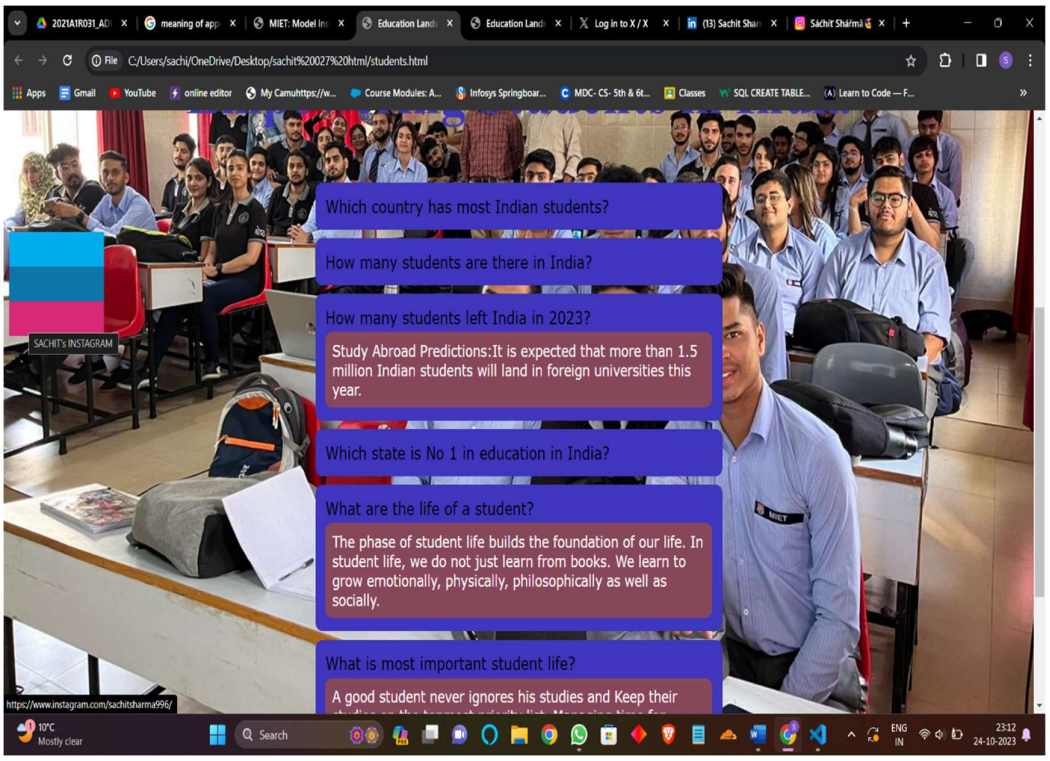


Fig3.9 Output SS4

3.5 Project Features

When working on a college-level FAQ webpage web development project, it's important to incorporate a range of features to create an effective and user-friendly experience. Here are some essential features to consider:

1. **Search Functionality: **

- Implement a search bar that allows users to quickly find answers to their questions by entering keywords.

2. **Accordion Style Sections: **

- Use accordions to organize the FAQs into collapsible sections, making it easy for users to access information without overwhelming them with too much text.

3. **Categorization: **

- Group FAQs into categories or topics to help users navigate and locate relevant

information.

4. **User-Friendly Navigation: **

- Ensure intuitive navigation with a clear and organized menu structure.

5. **Responsive Design: **

- Make the FAQ page responsive, so it works well on a variety of devices and screen sizes.

6. **Mobile Optimization: **

- Optimize the FAQ page for mobile devices, ensuring a seamless experience for users on smartphones and tablets.

7. **Feedback Mechanism: **

- Provide a way for users to submit feedback or ask additional questions not covered in the FAQs.

8. **Sorting and Filtering: **

- Allow users to sort FAQs by popularity, date, or category, and apply filters to refine search results.

9. **Rich Content Support: **

- Incorporate multimedia elements, such as images or videos, where relevant to enhance the user's understanding of certain FAQs.

10. **Accessibility Features:**

- Ensure the page complies with web accessibility standards (e.g., WCAG) to make it usable by people with disabilities.

11. **SEO Optimization:**

- Apply SEO best practices to make the FAQ page more discoverable by search engines.

12. **Content Management System (CMS):**

- If feasible, consider using a CMS that simplifies content management and updates.

13. **User Authentication (Optional):**

- Implement user accounts if needed to provide personalized experiences or access to restricted FAQs.

14. **User Analytics:**

- Incorporate analytics tools to track user behavior and measure the effectiveness of the FAQs.

15. **Multi-Language Support (Optional):**

- If your audience includes speakers of multiple languages, offer translations or provide FAQs in different languages.

16. **Social Sharing: **

- Include social sharing buttons so users can easily share helpful FAQs with others.

17. **Print-Friendly Version:**

- Add a feature that enables users to print FAQs for offline reference.

18. **Comprehensive FAQ Management: **

- Admin tools for managing, adding, editing, and deleting FAQs and categories.

19. **User Ratings and Feedback:**

- Allow users to rate the helpfulness of FAQs and provide feedback for continuous improvement.

20. **Security Measures:**

- Implement security features to protect user data and prevent unauthorized access.

Chapter 4: Conclusion:

In conclusion, the development of the FAQ page for our college-level project has been a rewarding and educational experience. Through this project, we have achieved several key objectives and gained valuable insights into web development. This report has highlighted the key aspects of our project, including its purpose, design, development process, and the technologies used. We have successfully created an interactive and user-friendly FAQ page that serves as a valuable resource for our college community.

Key takeaways from this project include:

1. **Enhanced Web Development Skills:** We have honed our web development skills, including HTML, CSS, JavaScript, and responsive design principles. This project has provided hands-on experience and improved our proficiency in these essential technologies.

2. **User-Centered Design:** Our focus on user experience and user-centered design has resulted in a user-friendly FAQ page that addresses the needs and preferences of our college community.

3. **Problem Solving and Collaboration:** Developing the FAQ page required effective problem-solving skills and collaboration among team members. We have learned the importance of communication and teamwork in achieving project goals.

4. **Content Management:** We have gained insights into content management systems and the importance of keeping information up-to-date to ensure the FAQ page remains a relevant and valuable resource.

5. **Responsiveness:** Our page is responsive across various devices and screen sizes, which is critical in today's mobile-centric world. This experience has underscored the significance of responsive design in modern web development.

6. **Future Improvements:** As part of our ongoing commitment to excellence, we

recognize the need for continuous improvement. Future enhancements may include expanding the FAQ database, integrating a search feature, and improving accessibility.

In summary, the FAQ page project has been an enriching experience that has equipped us with practical web development skills and insights into creating user-centered, responsive web solutions. It not only serves the immediate needs of our college community but also provides a foundation for future improvements and learning opportunities. This project has demonstrated the value of applying our knowledge in a real-world context and has further fueled our passion for web development. We look forward to taking these lessons into future projects and continuing to refine our abilities in this dynamic field.

REFERENCES

It's essential to gather relevant information and references to inform your design and development choices. Here are some potential references and sources I have considered for my project:

[1] GEEK FOR GEEKS: https://www.geeksforgeeks.org/

[2] TUTORIALS TONIGHT: https://www.geeksforgeeks.org/

[3] HOSTINGER: https://www.hostinger.in/

[4] WIKIPEDIA: https://www.wikipedia.org/

[5]TUTORIALPOINT: https://www.tutorialspoint.com/index.htm

[6] FREECODECAMP: https://www.freecodecamp.org/

[7] W3SCHOOLS: https://www.w3schools.com/

[8] SCALER: https://www.scaler.com/

[9] CODECADEMY: https://www.codecademy.com/resources/docs

[10] MOZILLA DEVELOPER: https://developer.mozilla.org/en-US/

[11] EDUCATIVE: https://www.educative.io/learn/home

[12] FIREBASE: https://firebase.google.com/docs

[13] STACKOVERFLOW: https://stackoverflow.com/

Appendix

A. TECH STACK

1. HTML
2. CSS
3. JAVASCRIPT
4. VISUAL STUDIO CODE

B. FEATURES

1. User-Friendly Navigation
2. Responsive Design
3. Feedback Mechanism
4. Rich Content Support
5. Social Sharing