Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

INTERNSHIP REPORT
On
"FRONT-END DEVELOPMENT "

Submitted in partial fulfilment of the requirement for the award of the Degree of

Bachelor of Engineering
In
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted by

RAGHAVENDRA M H

1DA21CS107

Under the Guidance of

Dr. Shamshekhar S.Patil

Associate Professor,
Dept. of CSE,
Dr.AIT, Bengaluru-56.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
2023-24

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

Aided By Govt. of Karnataka

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

Certified that the Internship Program Entitled "**FRONT-END DEVELOPMENT**", carried out by **Raghavendra M H** bearing **1DA21CS107** bonafide student of Dr. Ambedkar Institute of Technology, Bengaluru – 560056 in partial fulfilment for the award of **Degree in Bachelor of Engineering** in **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** of Dr. Ambedkar Institute ofTechnology during the academic year 2022-23.

Signature of the Mentor
Dr. Shamshekhar S.Patil,
Mentor,
Dr.AIT, Bengaluru – 56.

Signature of the HOD
Dr.Siddaraju, Professor and
HEAD,Department of CSE,
Dr.AIT, Bengaluru – 56.

ACKNOWLEDGEMENT

The satisfaction that accompanies this internship would be incomplete without the mention of the people who made it possible. Without their constant guidance and encouragement, our efforts would have gone in vain.

I consider myself privileged to express my gratitude and respect towards all those who guided me through the internship, especially **Raghavendra M H**.

I would like to express our gratitude to **Dr. C .Nanjundaswamy , Principal, Dr. A.I.T**, for providing the congenial environment to work in.

I would also like to express our profuse gratitude **to Dr. Siddaraju, HOD of the Dept. of Computer Science & Engineering at Dr. AIT**, for giving the support, encouragement, and providing the required facilities necessary for the completion of this internship.

As a token of gratitude, I acknowledge our sincere gratefulness to the internal guide, **Dr. Shamshekhar S.Patil, Associate Professor in the Dept. of CSE at Dr. AIT**, for his unlimited support and encouragement provided throughoutthe process.

I also express our gratitude and sincere thanks to all the teaching and non-teaching staff of the **Computer Science& Engineering Department.**

Finally, yet importantly, I would like to express our heartfelt thanks to my beloved parents for their blessings and my friends for their help and wishes for the successful completion of this education-oriented internship on creating a product website.

Raghavendra M H

1DA21CS107

ABSTRACT

This internship report encapsulates the immersive experience gained during a comprehensive web development internship. The report provides a detailed overview of the knowledge, skills, and insights acquired while working on various projects and tasks related to web development.

The internship primarily focused on honing skills in front-end development technologies. It involved hands-on experience in HTML, CSS, JavaScript, and popular frameworks/libraries such as React.js and Angular.js for creating dynamic and responsive user interfaces.

Throughout the internship, practical applications of web development concepts were explored, including versioncontrol with Git, responsive web design principles, API integration, and testing methodologies.

The report also discusses challenges faced, lessons learned, and the overall impact of the internship on personal and professional growth. It concludes with reflections on the significance of continuous learning and adaptation in the rapidly evolving field of web development.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

ABOUT COMPANY:

Mevi Technology Company stands as a beacon of technological innovation in today's dynamic business environment. This comprehensive report provides an in-depth analysis and insight into the core facets of Mevi Technology, shedding light on its origins, key technological advancements, corporate culture, and future prospects.

Mevi Technology Company, established in the year of 2022 in Tumakuru has emerged as a trailblazer in the realm of IT sector particularly for students who are struggling to emerge to the Technology . Headquartered in Tumkur Karnataka, the company has rapidly grown from its humble beginnings to become a formidable player in the globaltechnology landscape.

An analysis of Mevi Technology's market presence and global impact is crucial for understanding the company's position in the competitive landscape. This section provides an overview of Mevi's market share, strategic partnerships, and the geographical regions where it has made significant inroads.

Fig 1.Company staffs

INTRODUCTION TO WEB DEVELOPMENT

1.1 Introduction to web development

Web development refers to the creating, building and maintaining of websites. It includes aspects such as web design, web programming, and database management.Web development is the creation of an application that works over the interneti.e website.Web development is closely related to the job of designing the features and functionality ofapps.

The word Web Development is made up of two words, that is:

- Web: It refers to websites, web pages or anything that works over the internet
- Development: It refers to building the application from scratch

Web development encompasses various aspects such as front-end development, back-end development, and full-stack development.

1.1 Types of web development

1. Front-End development
2. Back-End development
3. Full-stack development

1.2.1 Introduction to front-end development

The part of a website where the user interacts directly is termed as front-end.

Front-end development refers to the process of creating the visual and interactive elements of a website or application that users interact with directly.

Front-end development includes using languages such as HTML for content structure, CSS for styling and JavaScript for interactivity of a website.

1.2.2 Introduction to back-end development

Back-end refers to the server side of web development.

Back-end development involves working on the server-side operations, databases and server logic to ensure the website functions correctly.

Back-end development utilizes programming languages like Python, PHP, Ruby, Node.js etc.

1.2.3 Introduction to full-stack development

Full-stack development combines both front-end and back-end development skills.

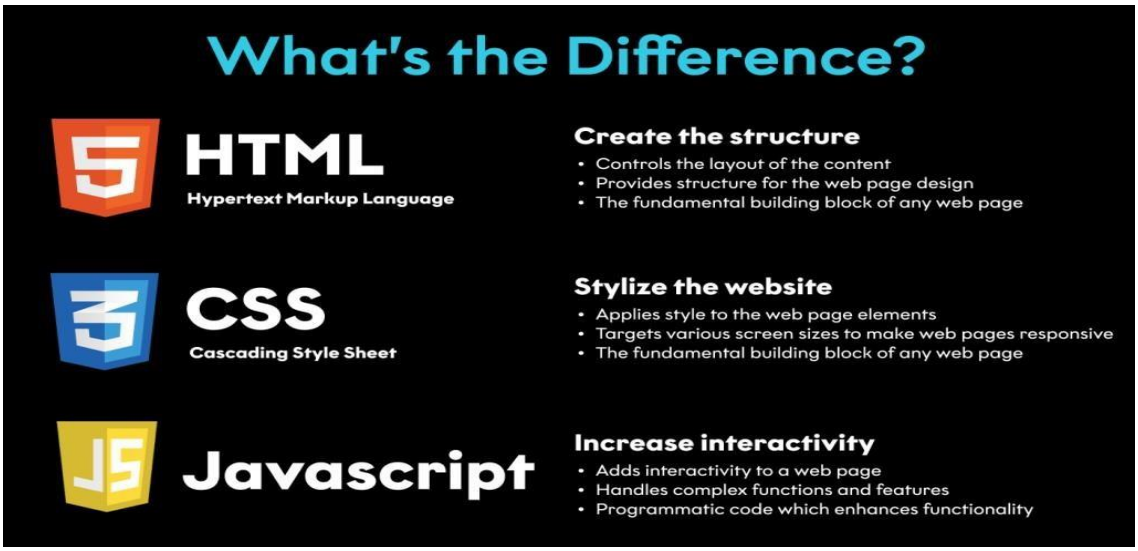
Full-stack development enables the developers to work on both client and server sides of a web application

Chapter 2

REQUIREMENT SPECIFICATION

Various tools and technology used for front-end development are listed below:

- HTML (Hypertext and Mark-up language)
- CSS (Cascading Style Sheets)
- JavaScript



2.1 HARDWARE REQUIREMENTS:

The section of hardware configuration is an important task related to the software development insufficient random-access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

- Process : Pentium IV and above
- Processor speed : 2.4Hz onwards
- System memory : 2 GB minimum 4 GB recommended
- Hard disk: 80Gb
- RAM : 2 MB(minimum)

2.2 SOFTWARE REQUIEREMENTS:

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric Progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system.

- Operating System : Windows 7 and above
- Working Environment : Google Chrome and Visual Studio Code

Chapter 3

DESCRIPTION

3.1 HTML

HTML is the combination of hypertext and mark-up language.

HTML (Hypertext Markup Language) is a fundamental language used in front- end web development. It provides the structure and content of web pages.

Here's an overview of HTML's role in front-end development

1. **Structure:**

HTML defines the structure of web pages using elements like <html>, <head>, <body>, <header>, <footer>, <nav>, <section>, <article>, etc. These elements organize content semantically, allowing browsers and search engines to understand the hierarchy and purpose of different parts of the page.

2. **Content creation:**

HTML allows you to create content by using various tags. For instance, <p> is used for paragraphs, <h1>-<h6> for headings, <ul> and <ol> for lists, <a> for links, <img> for images, <table> for tabular data etc.

3. **Semantic markup:**

HTML5 introduced semantic elements that help define the meaning of content more precisely. Using elements like <header>, <nav>, <section>, and <article> provides clearer semantics for both developers and searchengines, enhancing accessibility and SEO.

4. Forms:

HTML includes elements for creating interactive forms, such as <form>, <input>, <textarea>, <select>, <button>, etc., allowing users to input data and interact with websites.

5. Multimedia integration:

HTML includes elements for creating interactive forms, such as <form>, <input>, <textarea>, <select>, <button>, etc., allowing users to input data and interact with websites.

6. Accessibility:

Properly structured HTML helps in creating accessible websites by using elements and attributes that assist screen readers and other assistive technologies.

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>This is a Heading</h>
<p>This is a Paragraph</p>
</body>
</html>
```

Output:

This is a Heading This is a Paragraph

3.2 CSS (cascading style sheet)

CSS (cascading style sheet) is a fundamental technology used in front-end development. Its responsible for controlling the presentation, layout, and design of web pages.

Here are key aspects of CSS in front-end development:

1. Styling and Design:

- Selectors: CSS selectors help target HTML elements to apply style.
- Properties and values: control various styling attributes such as color, font, size, spacing, borders, background and more.
- Responsive design: use media queries to create responsive layouts that adapt to different screen seizes and devices'

2. Layout:

- Box model: understanding how elements rendered in terms of their content, padding, borders,and margins.
- Flexbox and grid: powerful layout models layout that offer more control and flexibility in arranging elements within the container.

3. Compatibality and Cross-Border Support:

- Ensuring CSS code works consistently across different browsers and versions.
- Using vendor prefixes for specific properties to ensure compatibility with older browser.

4. Animations and Transitions:

- Utilizing CSS transitions and animations to create engaging and interactive elements withoutJavaScript.

5. Preprocessors:

- Tools like Sass, LESS, or stylus offer features like variables, nesting, and functions, enhancing the capability of CSS.

6. Frameworks and Libraries:

- Using CSS frameworks like bootstrap, foundation, or tailwind CSS to expedite development by providing pre-built components and style.

7. Optimization and Performance:

- Minifying and composing CSS files to reduce loading time.
- Employing techniques like CSS Sprites or icon fonts to optimize rendering performance.

8. Maintainability and Scalability:

- Writing modular, reusable CSS code to maintain consistency and scalability across large projects.
- Employing methodologies like BEM (bold element modifier) or SMACSS (scalable and modulararchitecture for CSS) for better organization and structure

3.3 JAVASCRIPT

JavaScript is a programming language commonly used in front-end web development to create interactive and dynamic user experiences within web browsers. It's one of the three core technologies of the web along with HTML(Hypertext Markup Language) and CSS (Cascading Style Sheets).

JavaScript's role in front-end development:

1. Enhancing user interactivity:

JavaScript allows developers to create interactive elements on web pages. This includes things like form validation, pop-up alerts, sliders, accordions, and other dynamic content that responds to user actions.

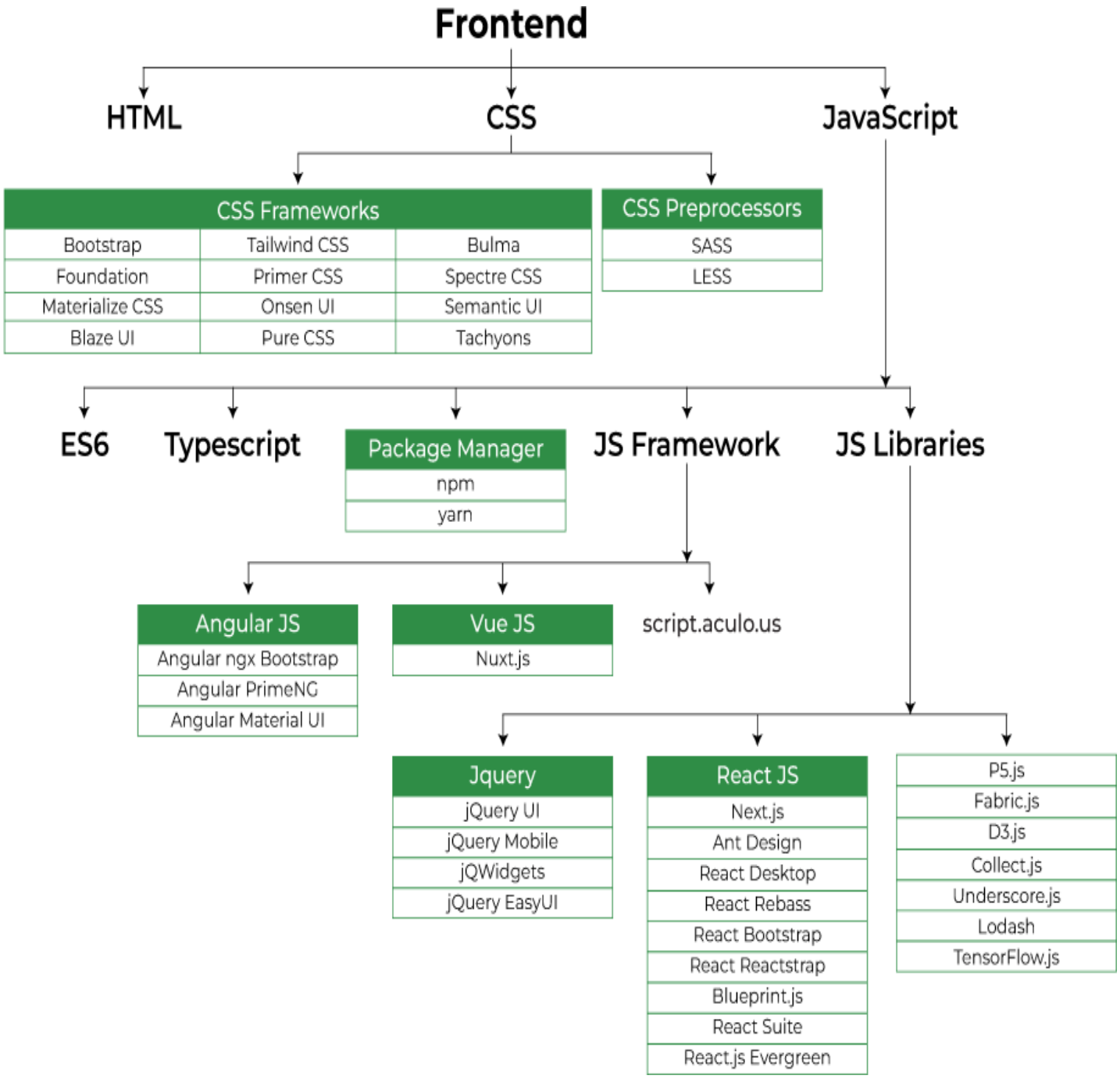
2. Manipulating the DOM:

The Document Object Model (DOM) represents the structure of an HTML document. JavaScript provides the ability to manipulate this DOM, allowing developers to dynamically change the content, structure, and style of web pages based on user interactions or other events.

3. Handling Asynchronous Events:

JavaScript is particularly useful for handling asynchronous tasks, such as fetching data from servers without having to reload the entire page. This is commonly done using AJAX (Asynchronous JavaScript and XML) or more modern techniques like the Fetch API or Axios to interact with APIs and update the page content dynamically.

3.4 Front-end development road map



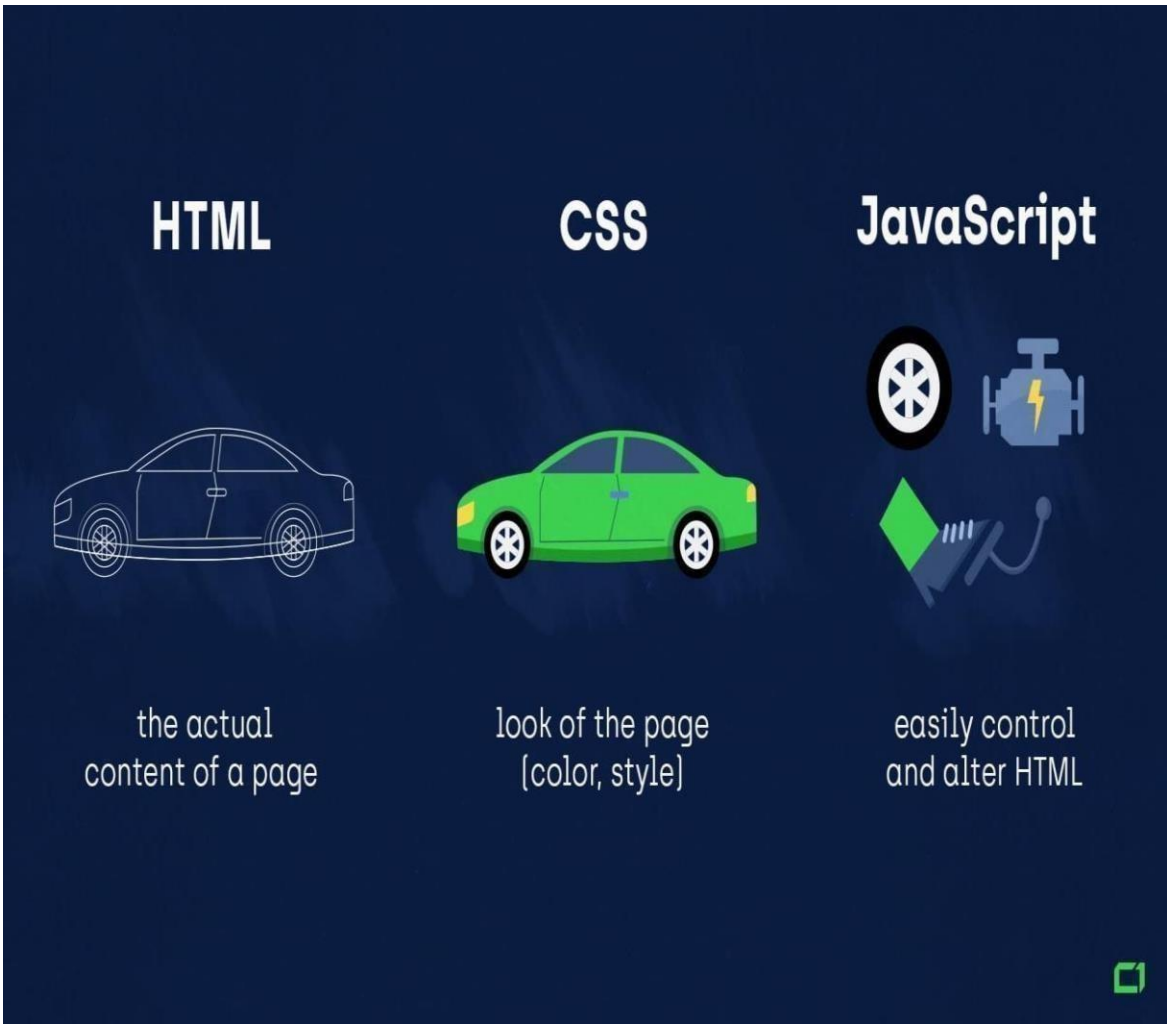


Fig 3.4.1 The below picture shows difference between HTML, CSS, and JavaScript

Chapter 4

INTERNSHIP PROJECT

4.1 Introduction:

The Temperature Converter HTML/CSS project is a simple yet practical web application that allows users to converttemperatures between different units, such as Celsius, Fahrenheit, and Kelvin. This project serves as a handy toolfor quickly converting temperatures without the need for manual calculations.

Features:

1. User-friendly Interface: The project boasts an intuitive and user-friendly interface, making it easy for users to navigate and input temperature values.

2. Conversion Options: Users can select from various temperature units including Celsius, Fahrenheit, and Kelvin. The converter dynamically updates the converted values as users input different temperatures.

3. Real-time Conversion: The conversion process occurs in real-time, providing instant feedback to users as they input temperature values or switch between units.

4. Responsive Design: The project is designed with responsiveness in mind, ensuring compatibility across different devices and screen sizes, including desktops, laptops, tablets, and smartphones.

5. Styling with CSS: The project utilizes Cascading Style Sheets (CSS) to enhance the visual appeal of the temperature converter, offering a polished and modern look.

Overall, the Temperature Converter HTML/CSS project is a practical and educational endeavor that combines simplicity with functionality, making it a valuable asset for both developers and end-users alike.

Background:

The background of the Temperature Converter HTML/CSS project can stem from various motivations or needs:

1. **Educational Purposes:** The project might have been initiated as a learning exercise for individuals seeking to enhance their skills in web development. Creating a temperature converter provides an excellent opportunity to practice HTML, CSS, and potentially JavaScript, as well as to understand the concepts behind temperature conversions.

2. **Practical Utility**: The need for a simple and accessible tool for temperature conversion could have driven the development of this project. Such a converter can be useful for students, professionals, or anyone needing to quickly convert temperatures between different units for various purposes, such as academic studies, scientific research, or everyday tasks.

3. **Portfolio Building**: For aspiring web developers or designers, building projects like the Temperature Converter can serve as valuable additions to their portfolios. It demonstrates their proficiency in frontend technologies like HTML and CSS, as well as their ability to create functional and visually appealing web applications.

4. **Community Contribution**: The project might have been developed with the intention of contributing to the open- source community. By making the code publicly available, developers can share their work with others, encourage collaboration, and allow for further improvement or customization by the community.

5. **Personal Interest or Challenge:** Sometimes, developers undertake projects simply out of personal interest or as a challenge to themselves. Creating a temperature converter could have been a fun and engaging endeavor to explore and experiment with frontend web development techniques.

Overall, the background of the Temperature Converter HTML/CSS project likely involves a combination of educational, practical, and personal motivations, driven by the desire to create a useful and visually appealing tool for temperature conversion.

4.2 Objective:

The objective of the Temperature Converter HTML/CSS project can be summarized as follows:

1. **Provide Functionality:** The primary goal is to create a web-based tool that allows users to convert temperatures between different units (e.g., Celsius, Fahrenheit, Kelvin) easily and accurately. The project aims to offer a seamless and intuitive user experience, enabling users to input temperatures and obtain the converted values effortlessly.

2. **Enhance User Experience:** The project seeks to provide a user-friendly interface that simplifies the temperature conversion process. This involves designing an intuitive layout, implementing real-time conversion updates, and ensuring responsiveness across various devices and screen sizes. By prioritizing user experience, the project aims to make temperature conversion accessible and efficient for users.

3. **Demonstrate Technical Skills:** For developers involved in creating the project, the objective may include showcasing their proficiency in frontend web development technologies such as HTML and CSS. Through the implementation of clean and well-structured code, adherence to best practices, and attention to detail in design and functionality, developers aim to demonstrate their technical abilities and expertise.

4. **Educational Purpose:** The project may also serve as an educational resource for individuals learning web development. By providing a practical example of building a web application from scratch, complete with HTML structure, CSS styling, and potentially basic JavaScript functionality, the objective is to help learners understand key concepts and techniques in frontend development.

5. **Encourage Collaboration and Contribution:** Depending on the project's nature, another objective might be to foster collaboration and contribution within the developer community. By making the project open-source and inviting others to review, contribute enhancements, or report issues, the objective is to create a collaborative environment where developers can learn from each other and collectively improve the project.

4.3 Project Structure:

The project structure of the Temperature Converter HTML/CSS project typically consists of the following components:

1. HTML File: This file defines the structure and content of the web page. It includes elements such as input fields for entering temperature values, dropdown menus for selecting temperature units, buttons for initiating conversions, and areas to display the converted results. The HTML file serves as the backbone of the project, organizing the various elements that make up the user interface.

2. CSS File: The CSS file contains styling rules and declarations that define the visual appearance of the temperature converter. This includes properties such as colors, fonts, margins, paddings, borders, and layout arrangements. CSS is used to enhance the aesthetics of the project, ensuring a visually appealing and cohesive design that aligns with the project's objectives and target audience.

3. Images : If the project incorporates any graphical elements such as icons, logos, or background images, these would be stored in a separate directory within the project structure. Images can be used to enhance the visual appeal of the temperature converter and contribute to the overall user experience.

4. JavaScript : While the primary functionality of the temperature converter can be achieved using HTML and CSS alone, JavaScript may be employed to add dynamic behavior or additional features to the project. For example, JavaScript can be used to perform real-time temperature conversions as users input values, validate user inputs, handle events such as button clicks or dropdown selections, or implement advanced functionalities like unit conversion formulas.

Overall, the project structure of the Temperature Converter HTML/CSS project is organized and well-defined, with separate files and directories for HTML, CSS, JavaScript, images, documentation, and potentially other assets, facilitating easy maintenance, collaboration, and further development.

Chapter 5

TRAINING AND MENTORSHIP

During the course of my internship, I had the opportunity to contribute to the development of a Memory Card Game utilizing HTML, CSS, and JavaScript. Throughout the project lifecycle, emphasis was placed on fostering a supportive learning environment through training and mentorship initiatives.

The training program aimed to equip team members with a comprehensive understanding of web development fundamentals, particularly focusing on the intricacies of HTML, CSS, and JavaScript. Collaborative efforts were undertaken with fellow development teams to explore the nuances of these languages and frameworks, enabling the design and implementation of specialized training modules tailored to the specific requirements of the project.

Continuous learning and professional growth were integral components of the internship experience. Team members were encouraged to remain abreast of emerging trends and best practices within the field of web development. This was facilitated through the provision of regular training sessions, workshops, and opportunities for engagement with industry conferences and resources, thereby fostering an environment conducive to ongoing skill enhancement and career development.

Effective communication and collaboration were paramount throughout the internship duration. Recognizing the collaborative nature of frontend development, an emphasis was placed on establishing open channels of communication across teams. This facilitated seamless collaboration, knowledge sharing, and the exchange of ideas, ultimately contributing to the project's success.

Furthermore, a robust incident response plan was developed to address any challenges or issues encountered during the development process promptly. This plan outlined clear protocols for issue identification, escalation, and resolution, ensuring effective communication and coordination among team members and stakeholders, thereby mitigating project risks and maintaining project momentum.

Chapter 6

CHALLENGES

1. Data Management and Retrieval:

- Challenge: Efficiently retrieving and managing card data for the game.

- Solution: Implementing asynchronous API calls or local data storage methods to fetch and store card information. Utilizing tools like Axios or local storage for seamless data management.

2. Dynamic Updates for Card Deck:

- Challenge: Handling dynamic updates for the card deck, including shuffling and resetting.

- Solution: Developing a dynamic system that updates the card deck based on user interactions or game events. Utilizing JavaScript functions to shuffle cards and reset the game state as needed.

3. User Interaction and Game Logic:

- Challenge: Managing user interactions and implementing game logic for card matching and gameplay.

- Solution: Implementing event listeners and game logic functions using JavaScript to handle card flips, matches, and game progression. Utilizing algorithms to check for matching pairs and update the game state accordingly.

4. State Management:

- Challenge: Managing the state of the game, including current card positions, matched pairs, and game progress.

- Solution: Leveraging JavaScript variables and objects to track the game state dynamically. Utilizing state management libraries or custom functions to ensure synchronized updates across the game interface.

5. Performance Optimization:

- Challenge: Optimizing game performance to minimize loading times and enhance user experience.

- Solution: Implementing techniques such as preloading images, optimizing code for efficient rendering, and minimizing resource usage to improve game performance. Utilizing browser developer tools for performance profiling and optimization.

6. Error Handling and Feedback Mechanism:

- Challenge: Effectively handling errors and providing feedback to users during gameplay.

- Solution: Implementing error handling functions to gracefully manage unexpected events or user errors.Providing informative feedback messages and options for users to retry or reset the game as needed.

7. Responsive Design and Cross-Device Compatibility:

- Challenge: Ensuring that the game is accessible and functions correctly across different devices and screen sizes.

- Solution: Utilizing responsive design principles and media queries to create a visually appealing and user- friendly game interface. Conducting testing on various devices and browsers to identify and address any compatibility issues.

8. Testing and Quality Assurance:

- Challenge: Ensuring that the game functions seamlessly and meets quality standards.

- Solution: Developing comprehensive unit tests and integration tests to validate the functionality and performance of different game features. Conducting user testing to gather feedback and make necessary improvements before final release.

Chapter 7
REFLECTIONS

1. Comprehensive Learning Journey:

- Engaging in the project demanded a comprehensive understanding of Restful API calls, web application development, and user interface design.

- The amalgamation of React JS, HTML, CSS, and JavaScript presented a holistic learning experience that stretched my technical capabilities.

2. Challenges as Learning Opportunities:

- Every challenge encountered during the project became an opportunity for growth.

- Overcoming API charging points complexities, integrating to CPO portal using pagination, and ensuring consistency in frontend development were pivotal moments that strengthened problem-solving skills.

3. Team Collaboration and Communication:

- The importance of effective team collaboration and communication became evident throughout the project.

- Establishing a culture of open communication facilitated the alignment of diverse skills and perspectives within the development teams, fostering a collaborative environment.

4. User-Centric Design:

- Emphasizing UI/UX design principles underscored the significance of user-centric development.

- Iterative feedback sessions and usability testing helped refine the system, ensuring it met the specific needs of end-users in the college management context.

5. Adapting to Evolving Technologies:

- Staying abreast of evolving technologies such as React.js, Redux.js, and other frontend frameworks required a commitment to continuous learning.

- This experience emphasized the need to adapt to new tools and practices in the ever-evolving landscape of software development.

6. Importance of Rigorous Testing:

- The project reinforced the importance of rigorous testing, both automated and manual, to identify and resolve bugs promptly.

- A systematic approach to debugging and troubleshooting contributed significantly to the overall stability and reliability of the system.

7. Project Management and Time Allocation:

- Managing the development phases required effective project management and time allocation.

- Prioritizing tasks, setting milestones, and ensuring regular progress updates were crucial aspects that contributed to the successful completion of the project.

8. Future Directions and Continuous Improvement:

- The reflections on this project extend beyond its completion.

- Recognizing the continuous evolution of technology, there is a commitment to staying informed, embracing new challenges, and implementing lessons learned in future projects.

- The journey does not end here but serves as a stepping stone for continuous improvement and growth.

Chapter 8
RECOMMENDATIONS

1. User Experience Enhancement:

- Engage in comprehensive feedback sessions with users to identify areas for improving the game's user experience. Incorporate user insights to enhance gameplay mechanics and interface design, making the game more intuitive and enjoyable.

2.Performance Optimization:

- Explore optimization techniques such as preloading assets and implementing efficient rendering algorithms to enhance game performance. Regular performance monitoring and analysis will help identify and address any bottlenecks or lag issues.

3. Scalability Considerations:

- Assess the scalability of the game to accommodate potential future updates or expansions. Plan for scalability enhancements to ensure the game can handle increased user traffic and additional features seamlessly.

4. Enhanced Security Measures:

- Implement security measures to protect the game from potential vulnerabilities and ensure the safety of user data. Regular security audits and updates will help mitigate evolving threats and maintain the integrity of the game.

5. Feedback Loop Optimization:

- Streamline the process for collecting and integrating user feedback into the game's development cycle. Improving feedback loops will enable faster responsiveness to user needs and preferences, driving continuous improvement.

6. User Education and Training:

- Develop tutorials or in-game guides to help players understand the game mechanics and features better. Providing resources for player education will enhance the overall gaming experience and increase player engagement.

7. API Communication Enhancements:

- Optimize API calls for smoother data retrieval and integration within the game. Implement caching mechanisms and refine data retrieval strategies to minimize latency and improve overall performance.

8. Continuous Integration/Continuous Deployment (CI/CD):

- Strengthen the CI/CD pipeline by introducing automated testing for game features and updates. Automated testing ensures the stability and reliability of new game releases and content updates.

9. Consideration of Accessibility Standards:

- Evaluate and ensure compliance with accessibility standards to make the game accessible to players with diverse needs. Implementing accessibility features will broaden the game's audience and enhance inclusivity.

10. Documentation for Development:

- Enhance documentation related to game development, providing comprehensive explanations for developers working on future enhancements. Detailed documentation facilitates collaboration and accelerates development efforts.

11. Cross-Platform Compatibility:

- Conduct thorough testing for cross-platform compatibility to ensure the game functions consistently across various devices and operating systems. Addressing compatibility issues will improve the accessibility and usability of the game.

Chapter 9
CODE

INDEX.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Memory Card Game | Raghavendra</title>
    <link rel="stylesheet" href="style.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div class="wrapper">
      <ul class="cards">
        <li class="card">
          <div class="view front-view">
            <img src="images/que_icon.svg" alt="icon">
          </div>
          <div class="view back-view">
            <img src="images/img-1.png" alt="card-img">
          </div>
        </li>
        <li class="card">
          <div class="view front-view">
```

```
<img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-6.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-3.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-2.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
```

```
</div>
  <div class="view back-view">
          <img src="images/img-1.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-5.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-2.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
```

```
<img src="images/img-6.png" alt="card-img">



 </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-3.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
          <img src="images/img-4.png" alt="card-img">
        </div>
      </li>
      <li class="card">
        <div class="view front-view">
          <img src="images/que_icon.svg" alt="icon">
        </div>
        <div class="view back-view">
```

```
<img src="images/img-5.png" alt="card-img">
      </div>
    </li>



<li class="card">
      <div class="view front-view">
        <img src="images/que_icon.svg" alt="icon">
      </div>
      <div class="view back-view">
        <img src="images/img-4.png" alt="card-img">
      </div>
    </li>
    <li class="card">
      <div class="view front-view">
        <img src="images/que_icon.svg" alt="icon">
      </div>
      <div class="view back-view">
        <img src="images/img-4.png" alt="card-img">
      </div>
    </li>
    <li class="card">
      <div class="view front-view">
        <img src="images/que_icon.svg" alt="icon">
      </div>
      <div class="view back-view">
```

```
 <img src="images/img-4.png" alt="card-img">
          </div>
        </li>
        <li class="card">
          <div class="view front-view">


     <img src="images/que_icon.svg" alt="icon">
          </div>
          <div class="view back-view">
            <img src="images/img-4.png" alt="card-img">
          </div>
        </li>
        <li class="card">
          <div class="view front-view">
            <img src="images/que_icon.svg" alt="icon">
          </div>
          <div class="view back-view">
            <img src="images/img-4.png" alt="card-img">
          </div>
        </li>
      </ul>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

Style.css

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap');
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}
body{
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  background: #6563FF;
}
.wrapper{
  padding: 25px;
  border-radius: 10px;
  background: #F8F8F8;
  box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}
.cards, .card, .view{
  display: flex;
  align-items: center;
  justify-content: center;
```

```
}
.cards{
  height: 400px;
  width: 400px;
  flex-wrap: wrap;
  justify-content: space-between;
}
.cards .card{
  cursor: pointer;
  list-style: none;
  user-select: none;
  position: relative;
  perspective: 1000px;
  transform-style: preserve-3d;
  height: calc(100% / 4 - 10px);
  width: calc(100% / 4 - 10px);
}
.card.shake{
  animation: shake 0.35s ease-in-out;
}
@keyframes shake {
  0%, 100%{
    transform: translateX(0);
  }
  20%{
    transform: translateX(-13px);
```

```
}
  40%{
    transform: translateX(13px);
  }
  60%{
    transform: translateX(-8px);
  }
  80%{
    transform: translateX(8px);
  }
}
.card .view{
  width: 100%;
  height: 100%;
  position: absolute;
  border-radius: 7px;
  background: #fff;
  pointer-events: none;
  backface-visibility: hidden;
  box-shadow: 0 3px 10px rgba(0,0,0,0.1);
  transition: transform 0.25s linear;
}
.card .front-view img{
  width: 19px;
}
.card .back-view img{
```

```
 max-width: 45px;
}
.card .back-view{
  transform: rotateY(-180deg);
}
.card.flip .back-view{
  transform: rotateY(0);
}
.card.flip .front-view{
  transform: rotateY(180deg);
}

@media screen and (max-width: 700px) {
  .cards{
    height: 350px;
    width: 350px;
  }
  .card .front-view img{
    width: 17px;
  }
  .card .back-view img{
    max-width: 40px;
  }
}
```

```
@media screen and (max-width: 530px) {
  .cards{
    height: 300px;
    width: 300px;
  }
  .card .front-view img{
    width: 15px;
  }
  .card .back-view img{
    max-width: 35px;
  }
}
```

Script.js

```
const cards = document.querySelectorAll(".card");
let matched = 0;
let cardOne, cardTwo;
let disableDeck = false;
function flipCard({target: clickedCard}) {
    if(cardOne !== clickedCard && !disableDeck) {
        clickedCard.classList.add("flip");
        if(!cardOne) {
            return cardOne = clickedCard;
        }
        cardTwo = clickedCard;
        disableDeck = true;
        let cardOneImg = cardOne.querySelector(".back-view img").src,
        cardTwoImg = cardTwo.querySelector(".back-view img").src;
        matchCards(cardOneImg, cardTwoImg);
    }
}
function matchCards(img1, img2) {
    if(img1 === img2) {
        matched++;
        if(matched == 8) {
            setTimeout(() => {
                return shuffleCard();
            }, 1000);
        }
```

```
    cardOne.removeEventListener("click", flipCard);
        cardTwo.removeEventListener("click", flipCard);
        cardOne = cardTwo = "";
        return disableDeck = false;
    }
    setTimeout(() => {
        cardOne.classList.add("shake");
        cardTwo.classList.add("shake");
    }, 400);
    setTimeout(() => {
        cardOne.classList.remove("shake", "flip");
        cardTwo.classList.remove("shake", "flip");
        cardOne = cardTwo = "";
        disableDeck = false;
    }, 1200);
}
function shuffleCard() {
    matched = 0;
    disableDeck = false;
    cardOne = cardTwo = "";
    let arr = [1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8];
    arr.sort(() => Math.random() > 0.5 ? 1 : -1);
    cards.forEach((card, i) => {
        card.classList.remove("flip");
        let imgTag = card.querySelector(".back-view img");
        imgTag.src = `images/img-${arr[i]}.png`;
```

```
card.addEventListener("click", flipCard);
    });
}
shuffleCard();
cards.forEach(card => {
    card.addEventListener("click", flipCard);
});
```

Chapter 10

SNAPSHOTS

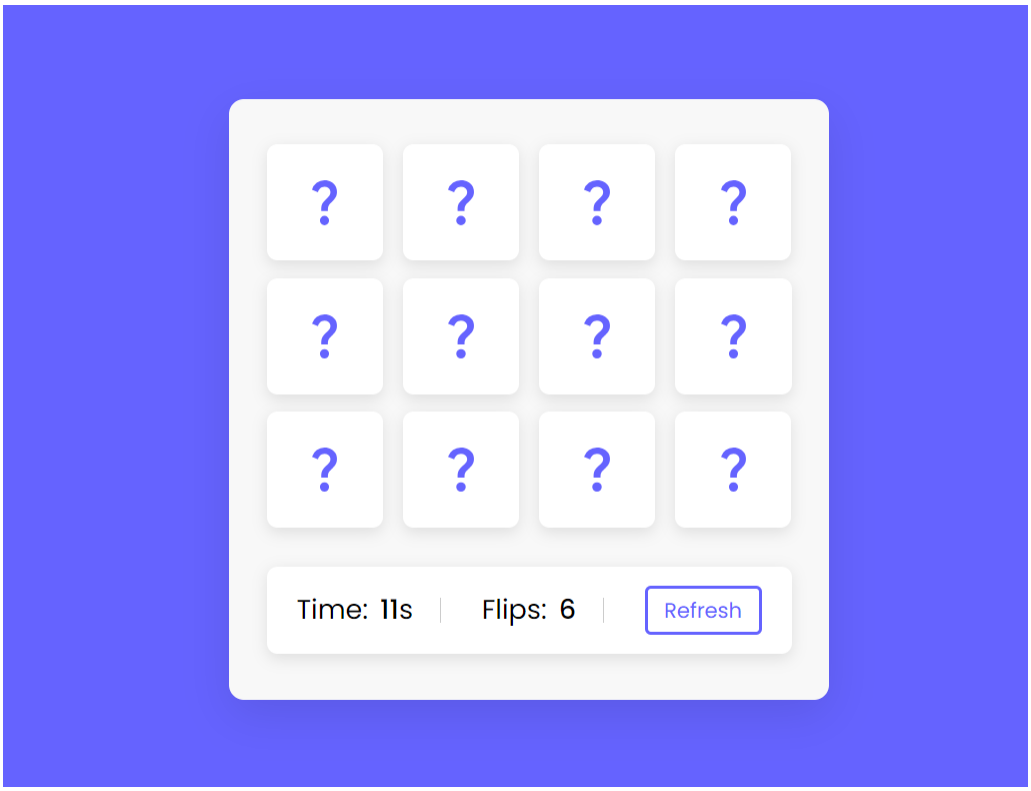


Fig 10.1

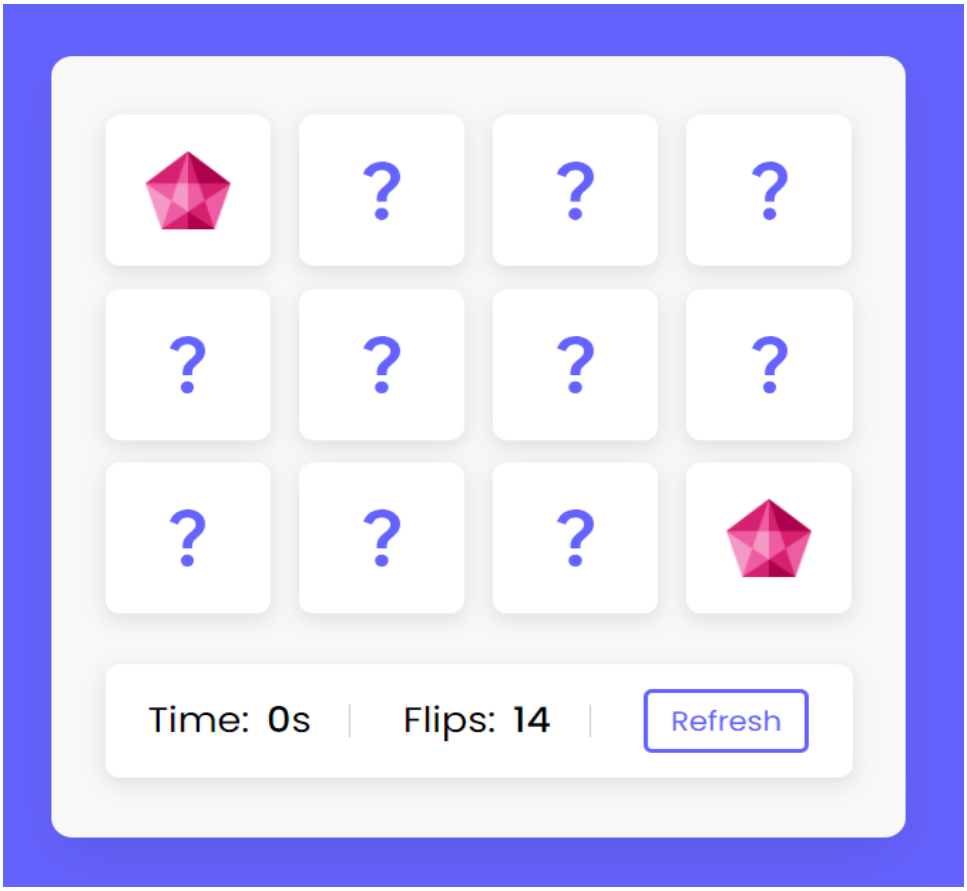


Fig 10.2

Certificate

CONCLUSION

Front-end web development serves as a pivotal element in the ever-evolving digital ecosystem. Through adept utilization of HTML, CSS, and JavaScript, developers craft engaging interfaces that transcend conventional boundaries. These interfaces are responsive, ensuring seamless user experiences across a myriad of devices and platforms. Additionally, the integration of progressive web applications (PWAs) and single-page applications (SPAs) underscores developers' commitment to innovation and functionality, further enhancing the user journey.

The continuous evolution of front-end frameworks and libraries empowers developers to push the boundaries of creativity and functionality. By leveraging these tools, they can create dynamic and interactive user interfaces that captivate audiences and drive engagement. Moreover, front-end developers prioritize accessibility, ensuring that digital experiences are inclusive and usable by individuals of all abilities. This dedication to inclusivity underscores the ethical responsibility of developers to create technology that is accessible to everyone.

As the digital landscape evolves, front-end developers must remain agile and adaptable, embracing emerging trends and technologies. By fostering a culture of continuous learning and collaboration, developers can stay ahead of the curve and deliver cutting-edge solutions. Ultimately, front-end development plays a vital role in shaping the future of web experiences, paving the way for richer, more immersive, and more accessible digital interactions.