

## **Front End Engineering-II**

Project Report

Semester-IV (Batch-2022)

### **WEB AGENCY PORTFOLIO**



**Supervised By:**

Raveesh Samkaria

**Submitted By:**

VANDANA

2210931016(G-14)

**Department of Computer Science and Engineer** Chitkara University  
Institute of Engineering & Technology, Chitkara University, Punjab

## **Abstract**

In the digital era, a compelling online presence is essential for businesses succeed. This project showcases a modern and responsive web agency portfolio developed using React, a popular JavaScript library for building user interfaces. The portfolio serves as a professional showcase for a web development agency, highlighting its services, past projects, and providing a means for potential clients to make contact.

## **Table of Contents**

1. Introduction

2. Problem Definition and Requirements

3. Proposed Design / Methodology

4. Results

## **Introduction**

In today's digital landscape, user experience reigns supreme. As web developers, our goal is not just to convey information but to do so in a manner that is intuitive, engaging, and visually appealing. One often-overlooked yet crucial aspect of this user experience is the Quiz core aap.

This is a “QUIZ” app which gives the information about CSE core subjects. First of all, No registration page by using his/her name unique roll number, by creating username and password required and may save your time . No need to register, they need to login every time to access their account.

By following along with this project, you will gain valuable insights into the following key aspects:

1. Structuring Layout: We'll utilize CSS's and javascript responsive design utilities to create a flexible and visually appealing layout that adapts seamlessly to various screen sizes and devices.
2. Styling Elements: CSS's extensive collection of utility classes, we'll style individual question to achieve a cohesive and aesthetically pleasing design.
3. Implementing Interactivity: We'll employ CSS's to create an interactive accordion behaviour, allowing users to expand and collapse individual questions to view their corresponding answers.

.

## **Background**

In today's fast-paced digital landscape, websites and applications serve as primary touchpoints for businesses and individuals alike. As such, providing users with an intuitive and seamless browsing experience is paramount to success. web agency portfolio a crucial role in this experience, serving as a central repository for addressing common queries and concerns users may have.

Traditionally, web agency portfolio have been viewed as mundane and static elements of a website, often overlooked in favor of more visually striking features. However, with the growing emphasis on user-centric design and accessibility, the importance of designing an engaging and web agency portfolio application cannot be overstated.

**CSS** – a revolutionary framework that has transformed the way developers approach web design. Unlike traditional CSS frameworks, which rely on pre-defined components and complex styling rules, CSS embraces a utility-first approach, providing developers with a vast array of utility classes that can be used to style elements directly in the markup.

**JavaScript** is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language. it is a programming language that allows you to create dynamic and interactive website content. It enables you to implement complex features on web pages, such as displaying timely updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc.

Against this backdrop, this project seeks to explore the potential of CSS in designing a modern and web agency portfolio application . By harnessing the power of CSS's utility classes, we aim to create a visually stunning and

highly functional Quiz application that not only addresses users' common queries but also enhances the overall browsing experience.

Through this project, we endeavor to demonstrate the versatility and efficiency of CSS in web design, providing developers with practical insights and techniques for creating polished and user-friendly interfaces. Whether you're a seasoned developer looking to streamline your workflow or a beginner eager to learn the basics of CSS, this project offers valuable lessons and inspiration for designing exceptional Quiz application that captivate and delight users.

## **Objectives**

The objective of developing a web agency portfolio using React encompasses several key goals aimed at both technical excellence and business success. These objectives can be categorized into enhancing user experience, showcasing the agency's capabilities, supporting business growth, and ensuring maintainability and scalability.

### **Through this project, we aim to achieve the following objectives:**

#### **1.Demonstrate Our Expertise:**

**Highlight Competencies:** Showcase our technical skills and creativity through an engaging portfolio that presents our best work and range of services, including web design, development, and SEO.

**2.Prove Capabilities:** Provide concrete examples of past projects to demonstrate our ability to deliver high-quality, innovative solutions tailored to client needs.

#### **3.Enhance Client Engagement:**

**4.Interactive Experience:** Create an interactive and user-friendly interface that captivates visitors and encourages them to explore our services and portfolio.

**5.Effective Communication:** Facilitate easy communication through integrated contact forms and clear calls-to-action, making it simple for potential clients to reach out and initiate projects.

#### **6.Optimize for Performance:**

**7.Fast Loading Times:** Implement performance best practices, including lazy loading and efficient rendering using React's virtual DOM, to ensure fast load times and a smooth browsing experience.

**8.Responsive Design:** Ensure the portfolio is fully responsive, providing an optimal viewing experience across all devices, from desktops to smartphones.

By achieving these objectives, this project aims to demonstrate the capabilities of CSS in designing polished and user-friendly questions related to Cse that not only address users' common queries but also contribute to a positive and engaging browsing experience. Whether you're a beginner learning the basics of CSS or a seasoned developer seeking to optimize your workflow, this project provides valuable insights and practical techniques for creating exceptional web interfaces.

## **Significance**

Developing a web agency portfolio using React holds substantial significance for several reasons, reflecting both technical and business advantages. Here's a detailed look at its importance:

- **Enhanced User Experience:**
- **Responsiveness:** A responsive design ensures that the portfolio is accessible and visually appealing across all devices, from desktops to smartphones. This broad accessibility is crucial in an era where users may access websites from a variety of devices.
- **Interactivity:** React's component-based architecture allows for the creation of dynamic and interactive user interfaces, providing a more engaging user experience. Features such as smooth scrolling, animated transitions, and interactive elements can be easily implemented.
- **Maintainability and Scalability:**
- **Component-Based Architecture:** React promotes a modular approach where each part of the portfolio is built as an independent component. This makes the codebase easier to manage, debug, and scale. New features can be added or existing ones modified without affecting the entire application.
- **Reusability:** Components can be reused across different parts of the application, reducing development time and effort.
- **Performance Optimization:**
- **Efficient Rendering:** React's virtual DOM efficiently updates and renders only the components that change, enhancing performance and providing a smoother user experience.
- **Lazy Loading:** React's support for lazy loading helps in loading only the necessary components and images when they are needed, reducing initial load times and improving overall performance.
- **SEO and Business Growth:**
- **SEO-Friendly:** A well-structured, responsive, and fast-loading website improves search engine rankings, making it easier for potential clients to find the agency online.
- **Professional Image:** A modern, aesthetically pleasing portfolio enhances the agency's professional image, instilling confidence in potential clients. It serves as a showcase of the agency's skills and capabilities, potentially leading to more business opportunities.
- **Technological Edge:**



- **Modern Framework:** Using React, a leading JavaScript library, positions the web agency at the forefront of modern web development practices. This not only attracts clients looking for cutting-edge solutions but also demonstrates the agency's commitment to staying updated with industry trends.
- **Community Support:** React has a vast community and robust ecosystem, offering a wealth of resources, libraries, and tools that can be leveraged to enhance the portfolio and streamline development processes.
- **Client Engagement:**
- **Interactive Elements:** Features such as testimonials, project showcases, and service highlights can be made interactive, providing visitors with a deeper understanding of the agency's work and capabilities.
- **Ease of Contact:** An integrated contact form with validation and real-time feedback facilitates easy communication between potential clients and the agency, improving lead generation and customer engagement.
- **Future-Proofing:**
- **Scalable Infrastructure:** As the agency grows and its portfolio expands, React's architecture ensures that the website can scale efficiently. New services, projects, or features can be added without overhauling the existing system.
  - **Continuous Improvement:** React's active development and frequent updates ensure that the framework

## **Problem definition and requirements**

### **Problem Definition**

In today's digital landscape, web development agencies must have a compelling online presence to attract and retain clients. Traditional static websites often fail to deliver the dynamic, interactive, and responsive experiences that modern users expect. Moreover, maintaining and scaling these websites can become increasingly challenging as the agency grows and evolves. There is a need for a more advanced solution that addresses these limitations while showcasing the agency's skills and services effectively.

Key aspects of the problem include:

### 1. User Interface Design:

- Design an intuitive and visually appealing interface using react.
- Create layouts for questions, options, buttons, and feedback messages.
- Ensure consistency in styling across different elements of the quiz.

### 2. Question Presentation:

- Display questions and options clearly and prominently.
- Use appropriate HTML elements such as `<div>`, `<ul>`, `<li>`, and `<input>` for structuring questions and options.

### 3. Interactivity:

- Implement interactive features using JavaScript to allow users to select answers, submit responses, and navigate through questions.

## **Requirements**

Requirements for the Future of Web Design:

### 1. HTML Structure:

- Define the overall structure of the quiz application using HTML elements.
- Organize elements such as questions, options, buttons, and feedback messages.

### 2. CSS Styling:

- Apply CSS styles to create an appealing and user-friendly interface.
- Design layout, typography, colors, and other visual elements to enhance the user experience.

### 3. JavaScript Functionality:

- Implement JavaScript functions to handle user interactions and quiz logic.
- Create functions to load questions, display options, check answers, calculate scores, and navigate between questions.

### 4. Fast Loading Time:

- Using react Json loading becomes faster.

#### 5. **User Interaction:**

- Enable user interaction by implementing event listeners for buttons, radio buttons, checkboxes, etc.
- Ensure that the Web agency Portfolio is responsive and works well on various devices and screen sizes.
- Use CSS media queries to adapt the layout and styling for different viewport sizes.

#### 6. **Accessibility:**

- Ensure accessibility by providing appropriate HTML markup, such as labels for form elements and ARIA attributes.
- Test the application with screen readers and keyboard navigation to ensure it is usable by people with disabilities.

#### 7. **Testing and Debugging:**

- Test the quiz application thoroughly to identify and fix any bugs or issues.
- Conduct usability testing to gather feedback and improve the user experience.

### **Hardware requirements**

For a project focused on designing and developing a Web agency Portfolio using CSS, the hardware requirements are relatively minimal. Here's a basic outline of the hardware needed:

1. Computer : Any modern computer capable of running web development tools and browsers should suffice. This includes desktops, laptops, or even tablets with appropriate development environments installed.

2. Operating System: The choice of operating system depends on the developer's preference. React Js development is platform-agnostic, meaning it can be used on Windows, macOS, or Linux systems.

3.API: API stands for **A**pplication **P**rogramming **I**nterface.

A Web API is an application programming interface for the Web.

A Browser API can extend the functionality of a web browser.

A Server API can extend the functionality of a web server.

## **Proposed design and methodology**

### **Design for Quiz Application :**

1. First view page consist of starting the quiz and asking for different subjects for the quiz.

2. Questions: Different question consist after selecting the subject according to you will appear on screen and after completing one by one you will get the output whether it is correct or incorrect.

3. Answer Display:

- Answers to the questions will be displayed in an accordion-style layout, ensuring that only one answer is visible at a time to minimize clutter.

- When a user clicks on a question, the corresponding answer will smoothly expand below it, providing a seamless and intuitive user experience.

- A subtle animation or transition effect can be applied to enhance the interaction and make the accordion behaviour more engaging.

4. Visual Elements:

- A minimalist design approach will be adopted, with a focus on clean lines, ample white space, and subtle colour accents to maintain visual interest.

- Use of a cohesive colour palette that complements the website or application's branding will help create a harmonious visual identity.

- Clear typography with readable font sizes and styles will ensure that content is easy to understand and accessible to all users.

5. Mobile Responsiveness:

- The design will be fully responsive, ensuring optimal display and usability across a wide range of devices, including smartphones and tablets.

## **Methodology**

1. Implement the accordion behaviour using JavaScript or a JavaScript framework such as React.

- Write code to toggle the visibility of answer content when a question is clicked, ensuring smooth transitions and accessibility.

2. Responsive Design:

- Ensure that Web agency Portfolio is fully responsive, adapting seamlessly to different screen sizes and devices.

- Test the responsiveness of the page across various breakpoints and make adjustments as needed to optimize the user experience.

3. Accessibility Considerations:

- Incorporate accessibility best practices into the design and development process to ensure that the Web agency Portfolio is usable by all users, including those with disabilities.

- Use semantic HTML markup, ARIA attributes support to enhance accessibility.

#### 4. Testing and Quality Assurance:

- Conduct thorough testing of Web agency Portfolio across different browsers, devices, and assistive technologies to identify and address any compatibility issues.

- Perform usability testing with real users to gather feedback on the clarity, effectiveness, and usability of Web agency Portfolio

By following this methodology, you can systematically design, develop, and deploy a high-quality Web agency Portfolio that effectively addresses users' common questions and enhances the overall user experience.

### **Algorithms used**

Building a web agency portfolio with React involves several algorithms and techniques to ensure responsiveness, performance optimization, interactivity, and maintainability. Here are some key algorithms and techniques commonly used: This algorithm follows these steps:

#### **1. Virtual DOM Diffing Algorithm:**

**React's Reconciliation Algorithm:** React uses a virtual DOM to keep a lightweight copy of the actual DOM in memory. When a component's state or props change, React calculates the difference (diff) between the previous virtual DOM and the new virtual DOM. It then efficiently updates the actual DOM with only the changes, minimizing direct DOM manipulation and improving performance.

**CSS Media Queries:** Media queries are used to apply different styles based on the device's screen size. This ensures that the layout adapts to various screen sizes, providing a good user experience on both desktop and mobile devices.

**Flexbox and Grid Layout:** These CSS layout models are used to create flexible and responsive grid layouts. They allow elements to be dynamically arranged based on screen size and available space.

**Intersection Observer API:** This API is used to defer loading of off-screen images and components until they come into the viewport. This technique reduces initial load times and improves performance by only loading necessary content when it is needed.

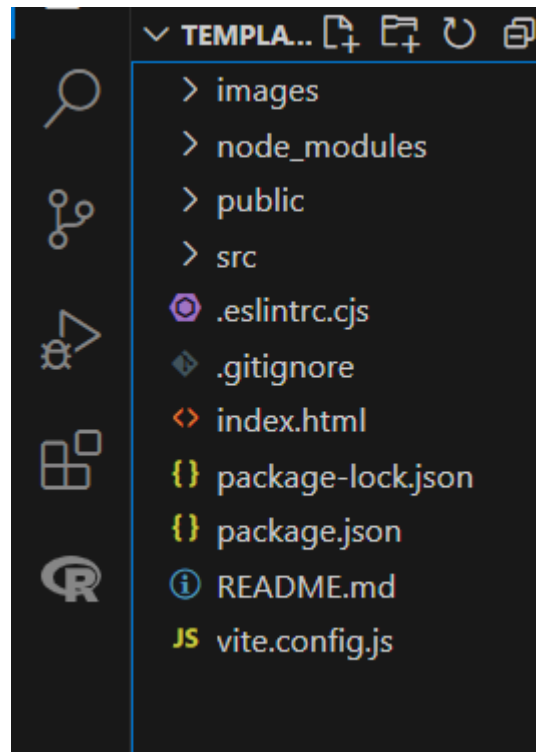
**React Router:** This library is used to manage navigation within the single-page application (SPA). It allows for dynamic routing, enabling users to navigate between different sections of the portfolio without triggering a full page reload.

**useState and useEffect Hooks:** React hooks are used to manage state and side effects. **useEffect** handles side effects such as data fetching, subscriptions, and manual DOM manipulations.

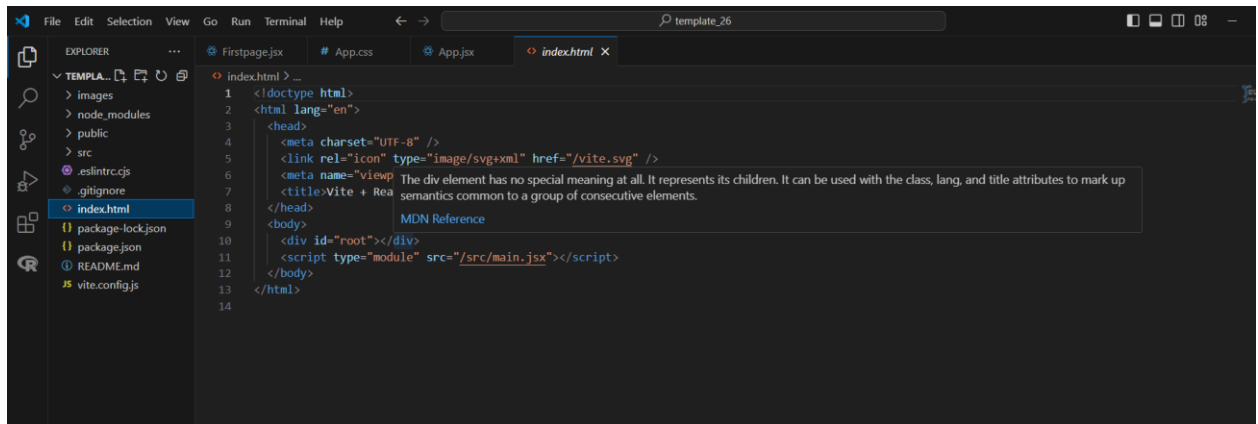
**Context API:** For managing global state, the Context API is used to provide state that can be accessed by any component in the component tree, avoiding the need to pass props down through multiple levels.



## **FILE STRUCTURE:-**



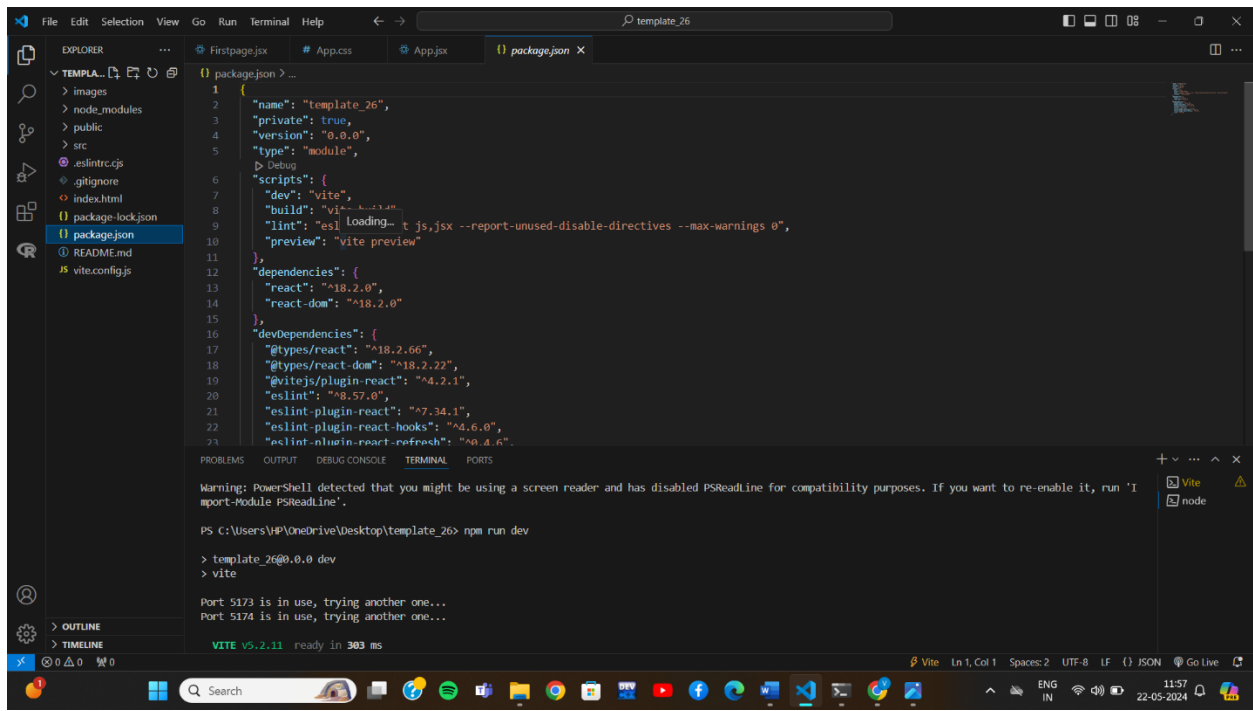
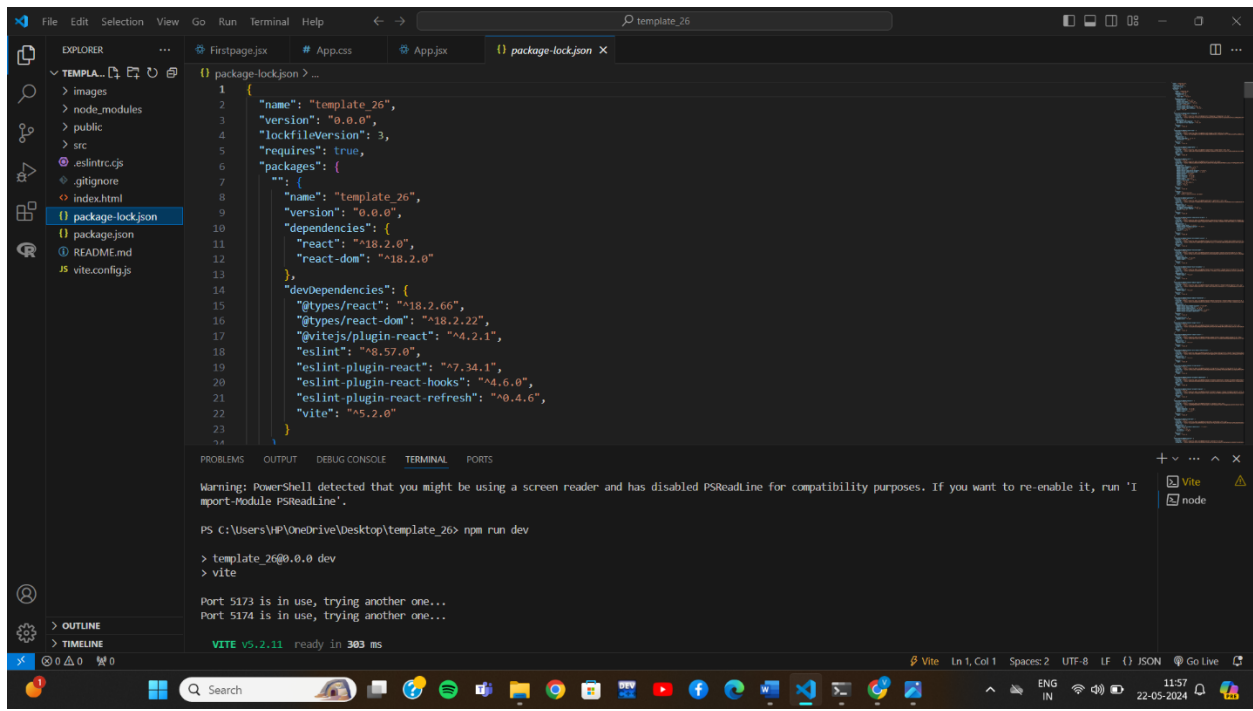
## Code

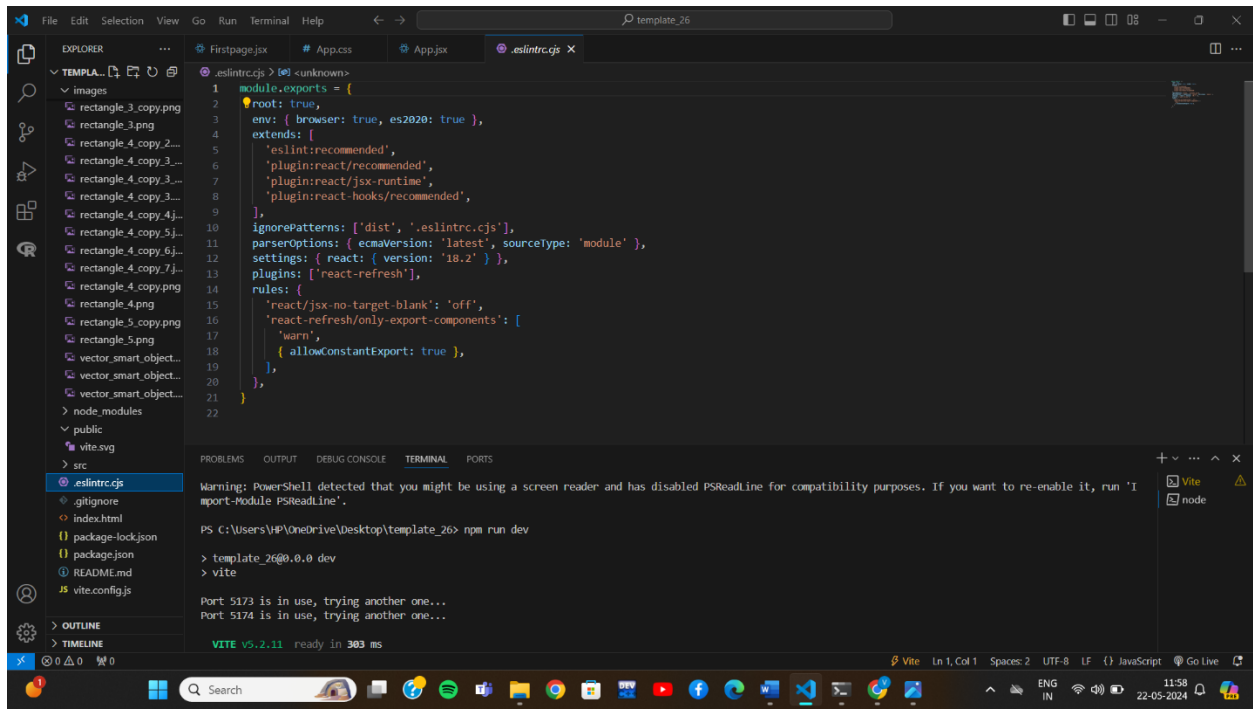
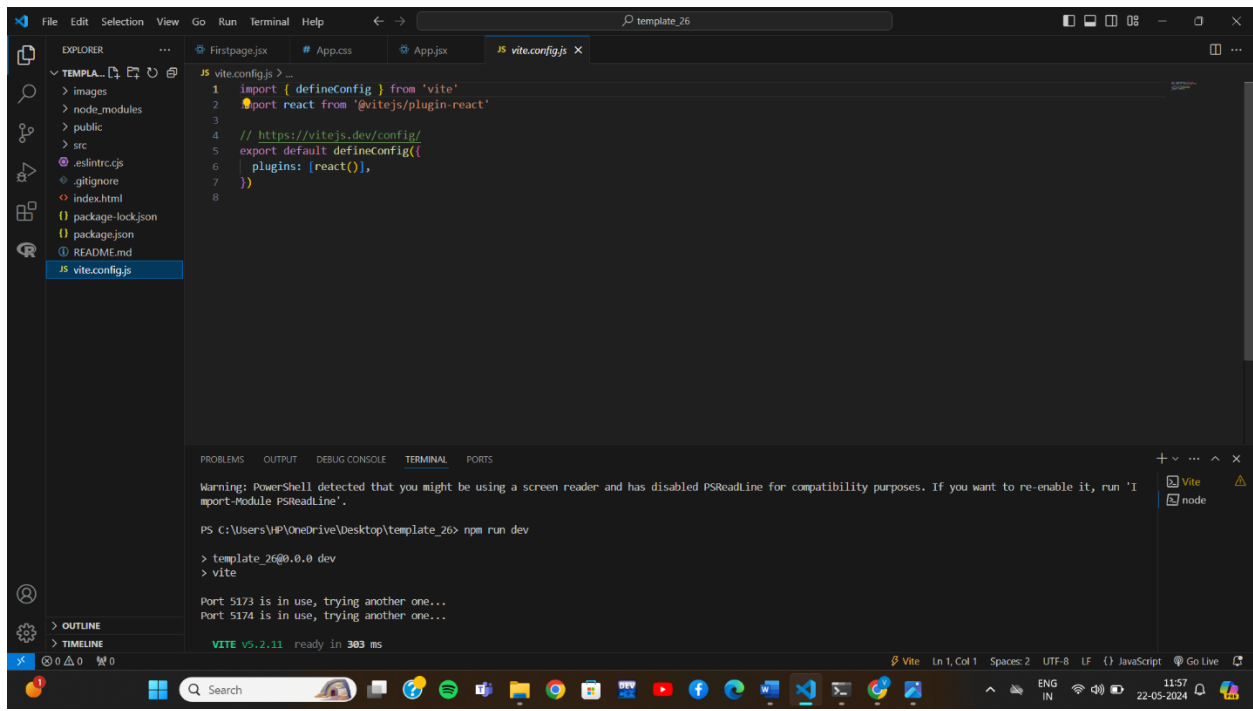


```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6     <meta name="viewport" />
7     <title>Vite + React</title>
8   </head>
9   <body>
10    <div id="root"></div>
11    <script type="module" src="/src/main.js"></script>
12  </body>
13 </html>
14
```

The div element has no special meaning at all. It represents its children. It can be used with the class, lang, and title attributes to mark up semantics common to a group of consecutive elements.

[MDN Reference](#)



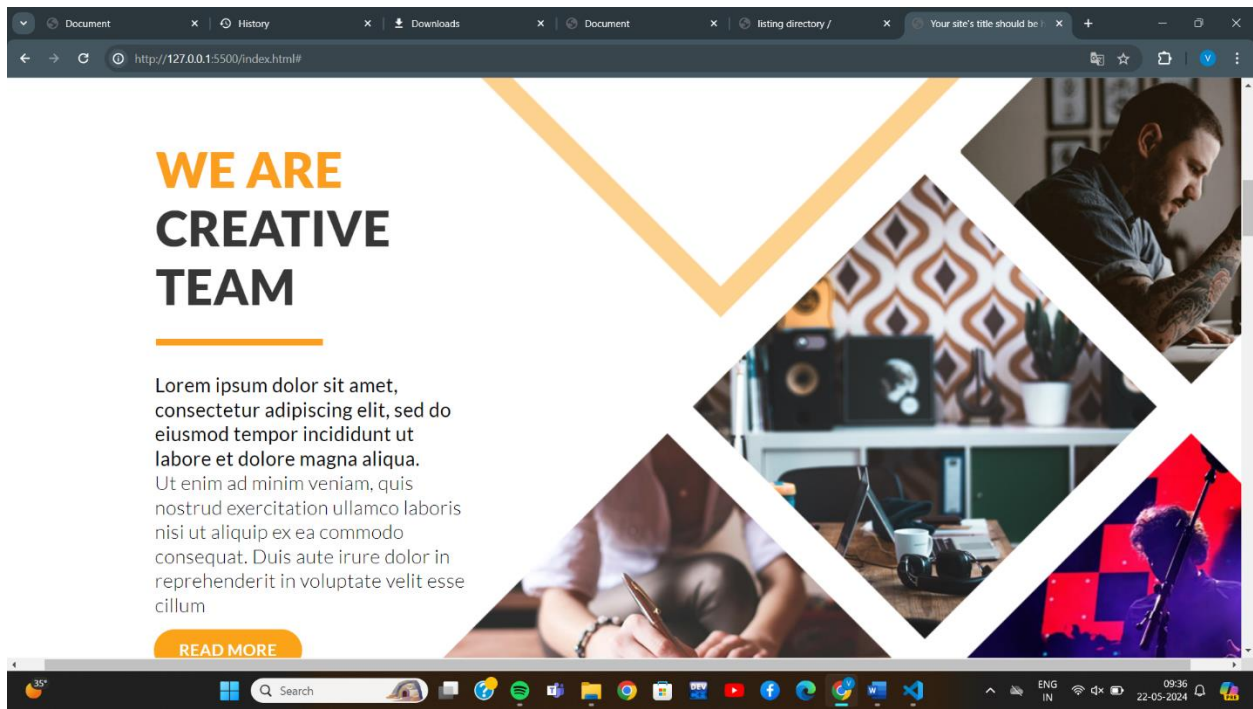
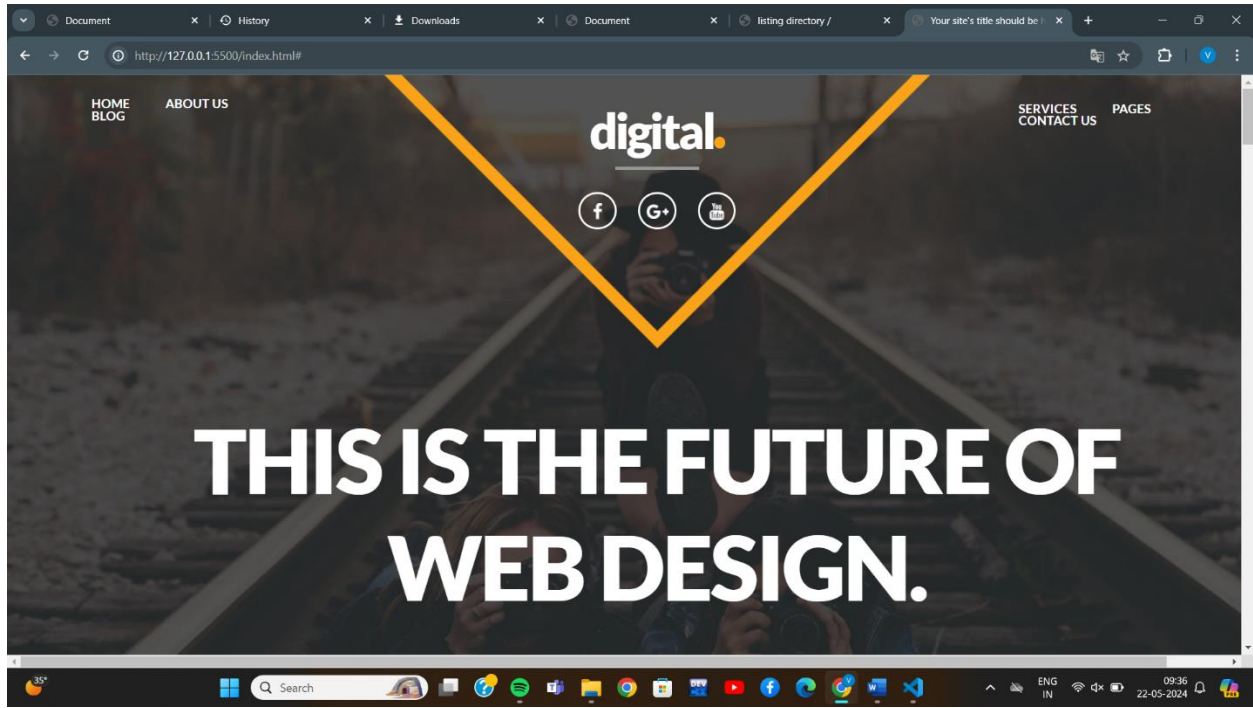


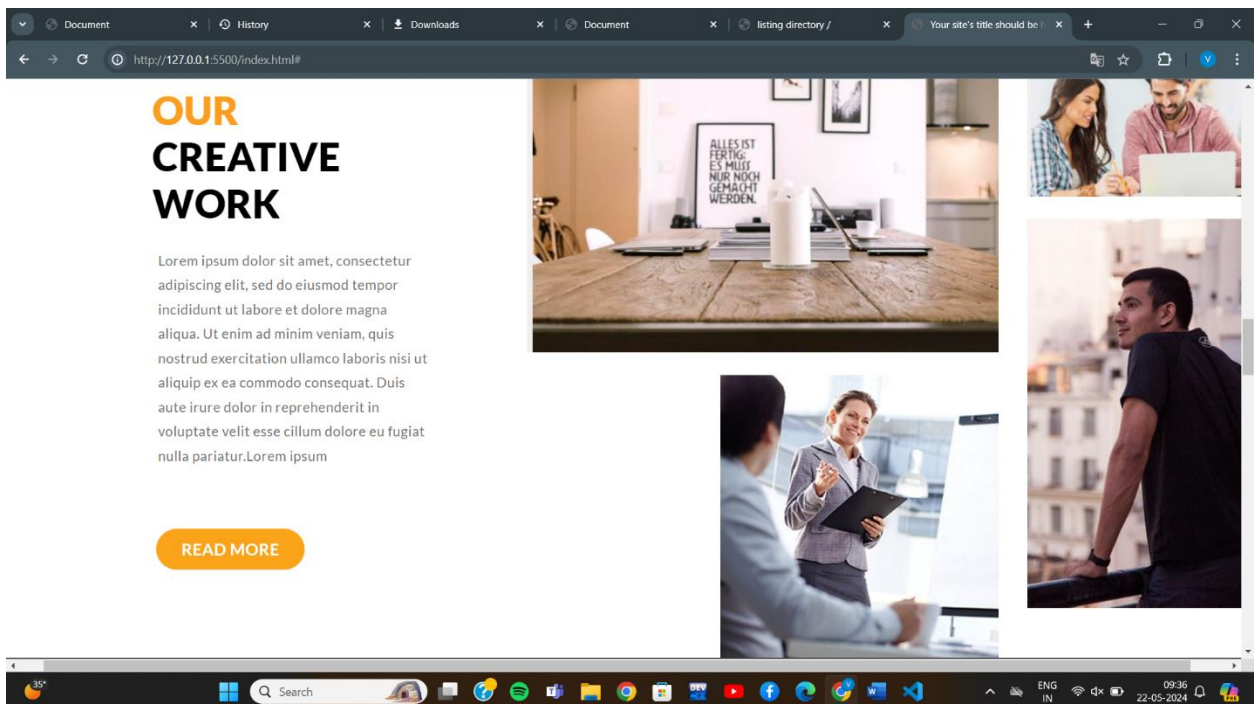
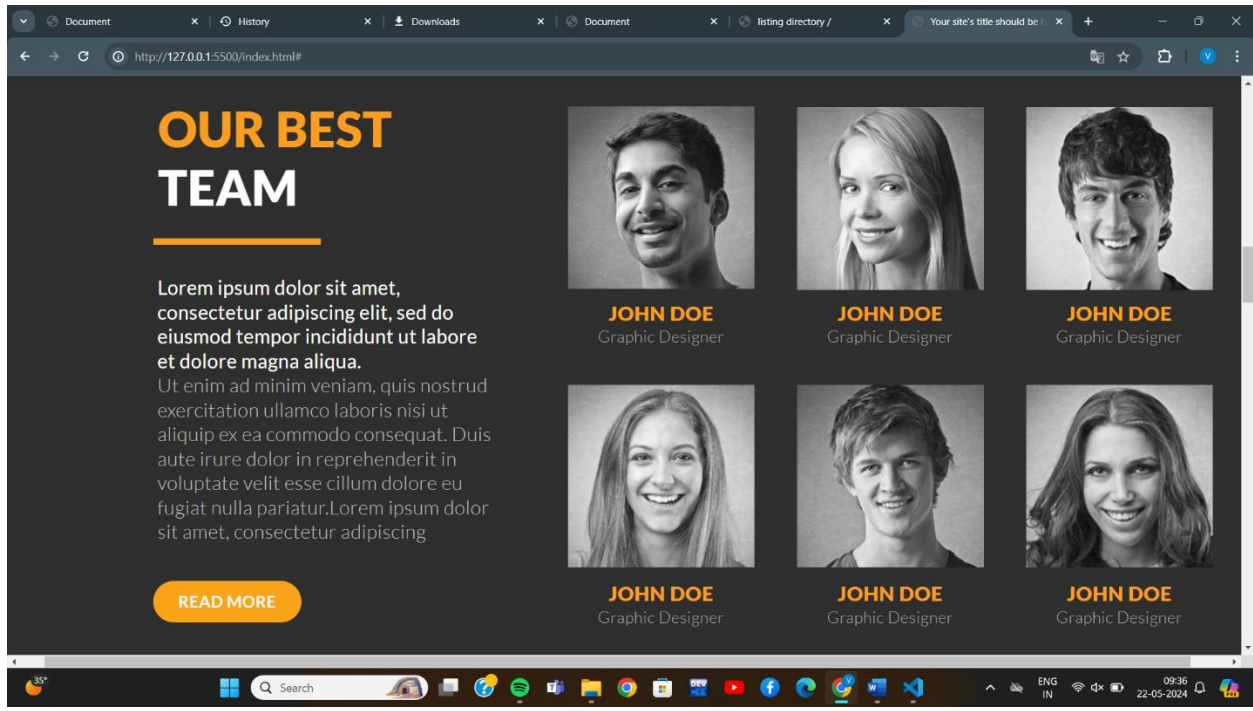
## TERMINAL:

```
2122 ... resolved: https://registry.npmjs.org/array-includes/->array-includes-3.1.7.tgz ;  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.  
  
PS C:\Users\HP\OneDrive\Desktop\digital-agency-main\digital-agency-main> npm run dev  
  
> the-coding-journey@0.0.0 dev  
> vite  
  
Port 5173 is in use, trying another one...  
Port 5174 is in use, trying another one...  
Port 5175 is in use, trying another one...  
Port 5176 is in use, trying another one...
```

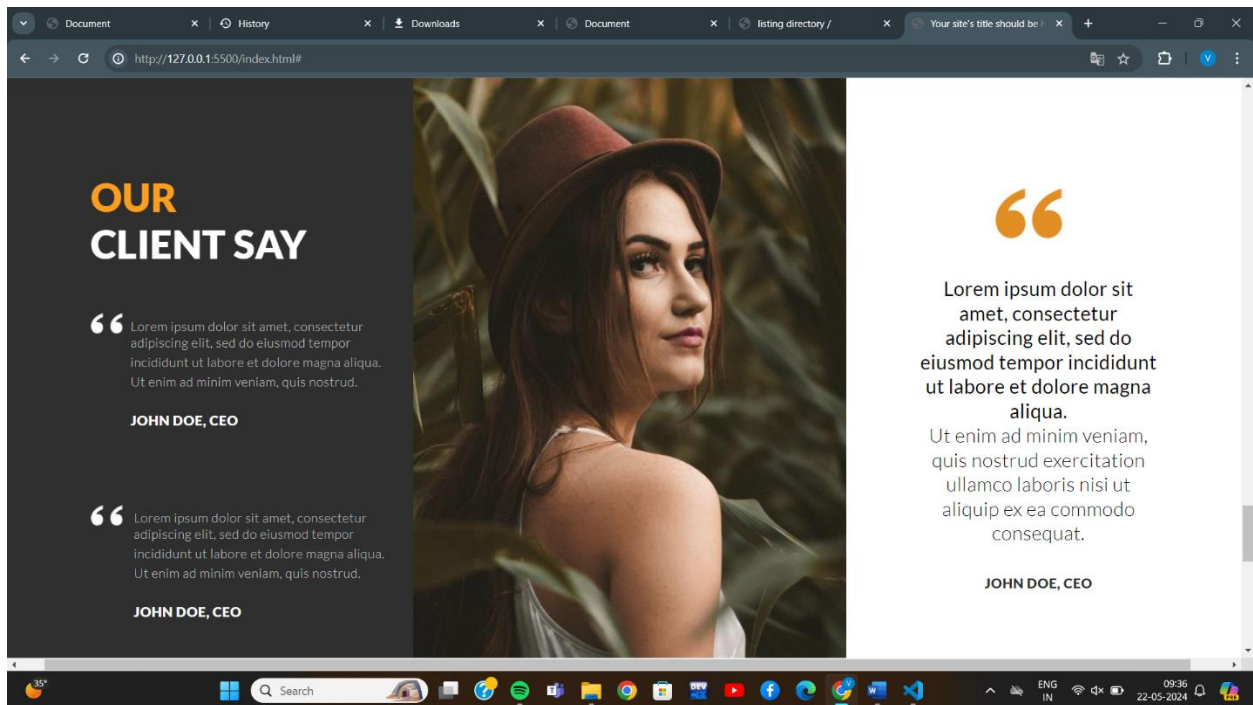
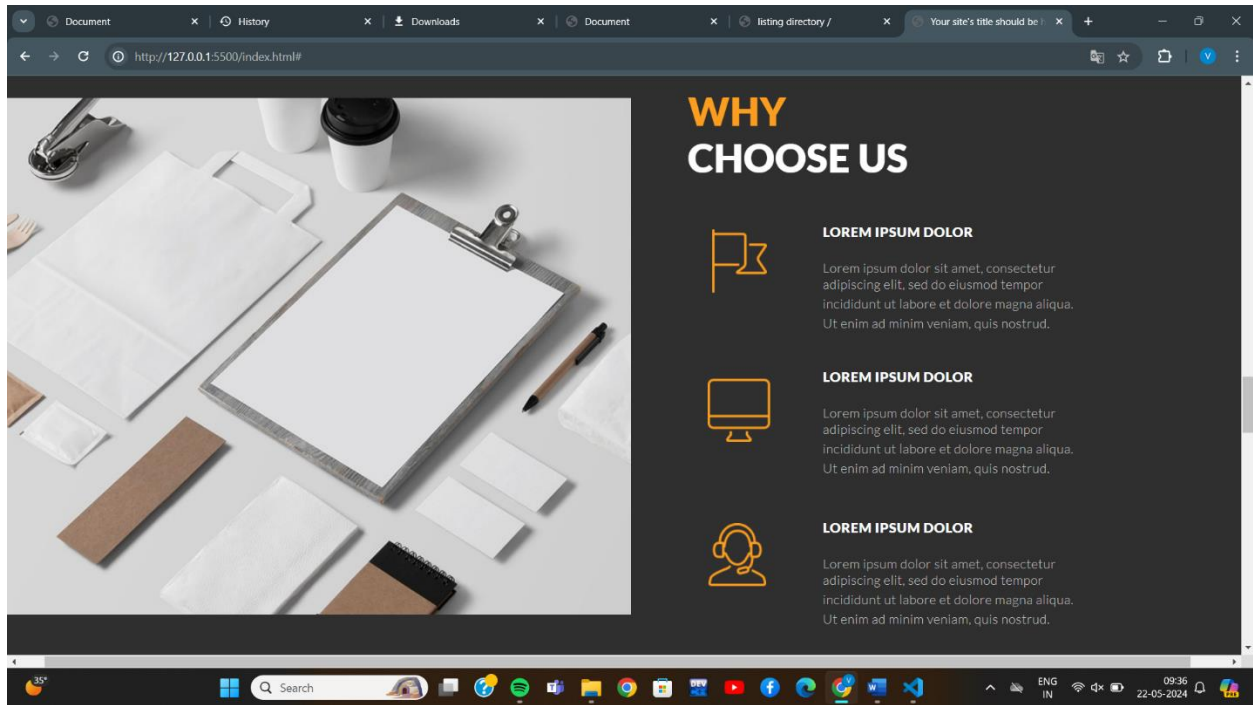
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
  
Port 5173 is in use, trying another one...  
Port 5174 is in use, trying another one...  
Port 5175 is in use, trying another one...  
Port 5176 is in use, trying another one...  
  
VITE v5.0.8 ready in 509 ms  
  
→ Local: http://localhost:5177/  
→ Network: use --host to expose  
→ press h + enter to show help
```

## Results

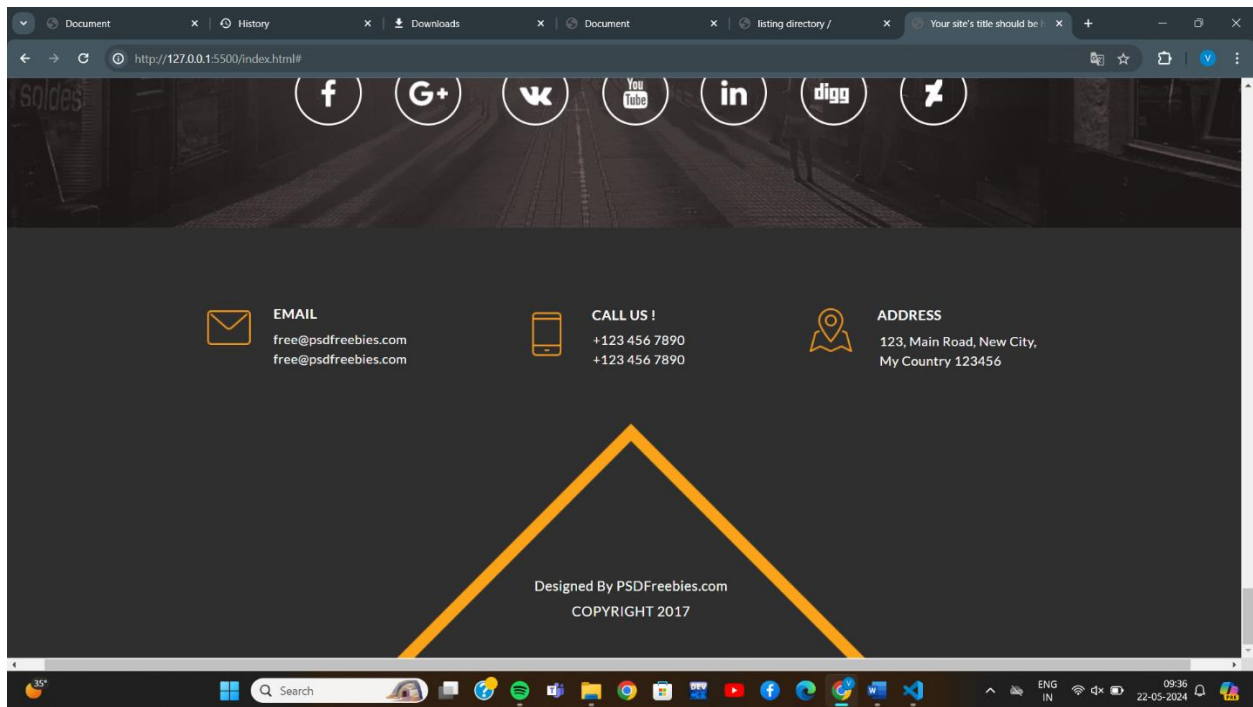












## References:

1. React JS: <https://react.dev/learn>
2. Tailwind CSS: <https://tailwindcss.com/docs/>
3. Google Fonts: <https://fonts.google.com/>
4. W3 Schools: <https://www.w3schools.com/>