## Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

**QUIZ APPLICATION**

A red and white sign

Description automatically generated with low confidence

**Supervised By: Submitted By:**

Raveesh Samkaria VANDANA

2210931016(G-14)

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

**Abstract**

The Quiz Core app is a web-based platform designed to offer users an engaging and interactive quiz experience. Developed using HTML, CSS, and JavaScript, Quiz Core provides a user-friendly interface where participants can test their knowledge across various topics.

It aims to offer an interactive and engaging experience while testing users' knowledge. It includes the question related to front end engineering to boost up your knowledge. It includes 5 subjects related to that and get marks output your answer . It includes different questions and outputs the answer at same time whether you gave right or wrong

**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. Quiz application play a crucial role in this experience, serving as a central repository for addressing common queries and concerns users may have.**

**Traditionally, Quiz application 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 functional Quiz 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 responsive Quiz 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 this project is to design and develop a modern, responsive, and visually Quiz question page using CSS. CSS is a utility-first CSS framework known for its simplicity, flexibility, and efficiency, making it an ideal tool for rapidly prototyping and styling web interfaces without the need for writing custom CSS.**

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

**1. Create a Responsive Layout: Utilize CSS's system and responsive design utilities to craft a layout that seamlessly adapts to various screen sizes and devices, ensuring an optimal viewing experience for users across desktop and mobile platforms.**

**2. Style the questions: Employ CSS's extensive collection of utility classes for typography, spacing, and colour to style individual questions, creating a cohesive and visually appealing design that aligns with the overall aesthetic of the website or application.**

**3.Implement Interactive Accordion Behaviour : Utilize CSS's utility classes to implement an interactive accordion behaviour for collapsible answers, allowing users to expand and collapse individual questions to view their corresponding answers with ease and efficiency.**

**4. Enhance User Experience: By combining the power of CSS with thoughtful design choices and intuitive interactive elements, aim to enhance the overall user experience of the Quiz App, making it easy for users to find relevant information and navigate the content seamlessly.**

**5. Promote Best Practices : Throughout the project, emphasize best practices for structuring HTML markup, organizing CSS classes, and optimizing code maintainability and scalability, empowering developers to apply these principles to their own projects effectively.**

**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**

1. **Educational Purposes**: It's a great way to learn and understand web development technologies. Creating a quiz project helps developers practice their skills in HTML for structuring content, CSS for styling and layout, and JavaScript for interactivity and functionality.
2. **Interactive Learning**: Interactive quizzes engage users more effectively than static content. They provide immediate feedback, which enhances the learning experience.
3. **Skill Assessment**: Developing a quiz project allows developers to assess their proficiency in front-end web development. It requires understanding of DOM manipulation, event handling, and styling elements to create an engaging and functional user interface.
4. **Practical Application**: Quizzes are commonly used in e-learning platforms, job assessment tests, and various other educational contexts. Developing a quiz project provides practical experience in building a real-world application that can be deployed and used by others.
5. **Customization and Adaptability**: By building a quiz project from scratch, developers have the freedom to customize the design, features, and functionality according to their specific requirements. This allows for experimentation and creativity in implementing various quiz formats and features.
6. **Portfolio Building**: A well-designed quiz project can serve as a valuable addition to a developer's portfolio, showcasing their skills and creativity to potential employers or clients.
7. **Community Contribution**: Open-source quiz projects can be shared with the developer community, contributing to the collective knowledge and providing learning resources for others.

Overall, a quiz project developed with HTML, CSS, and JavaScript can be significant both as a learning exercise and as a practical application with various potential uses.

## ****Problem definition and requirements****

**Problem Definition**

**The problem at hand is the need to design and develop a modern and user-friendly Quiz application page for a website or application. While quizes are essential for providing users and coming generation with answers to common questions related to different subjects , they are often overlooked in terms of design and functionality. The challenge lies in creating a Quiz application that not only effectively addresses users' questions but also enhances the overall browsing experience, promoting engagement and satisfaction.**

**Key aspects of the problem include:**

1. **User Interface Design**:
   * Design an intuitive and visually appealing interface using CSS.
   * 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. **Question Bank**:
   * Prepare a set of questions and corresponding options for the quiz.
   * Store questions and options in an array or JSON format for easy retrieval and manipulation.
5. **User Interaction**:
   * Enable user interaction by implementing event listeners for buttons, radio buttons, checkboxes, etc.
   * Allow users to select answers, submit responses, and navigate through questions..
   * Ensure that the quiz application 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 **Quiz application**  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. CSS 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 the **Quiz application**  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 **Quiz application**  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 the **Quiz application**  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 the **Quiz application**

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

**Algorithms used**

The algorithm used for implementing the accordion behaviour on the FAQ page can vary depending on the developer's preferences and the specific requirements of the project.

This algorithm follows these steps:

1. Selects start quiz.

2. Asks for 5 subjects for the user.

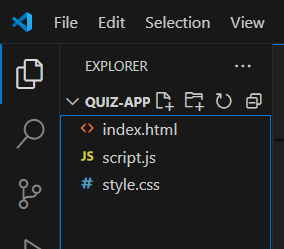
3. Question get displayed .

4. Select the correct answer for the question.

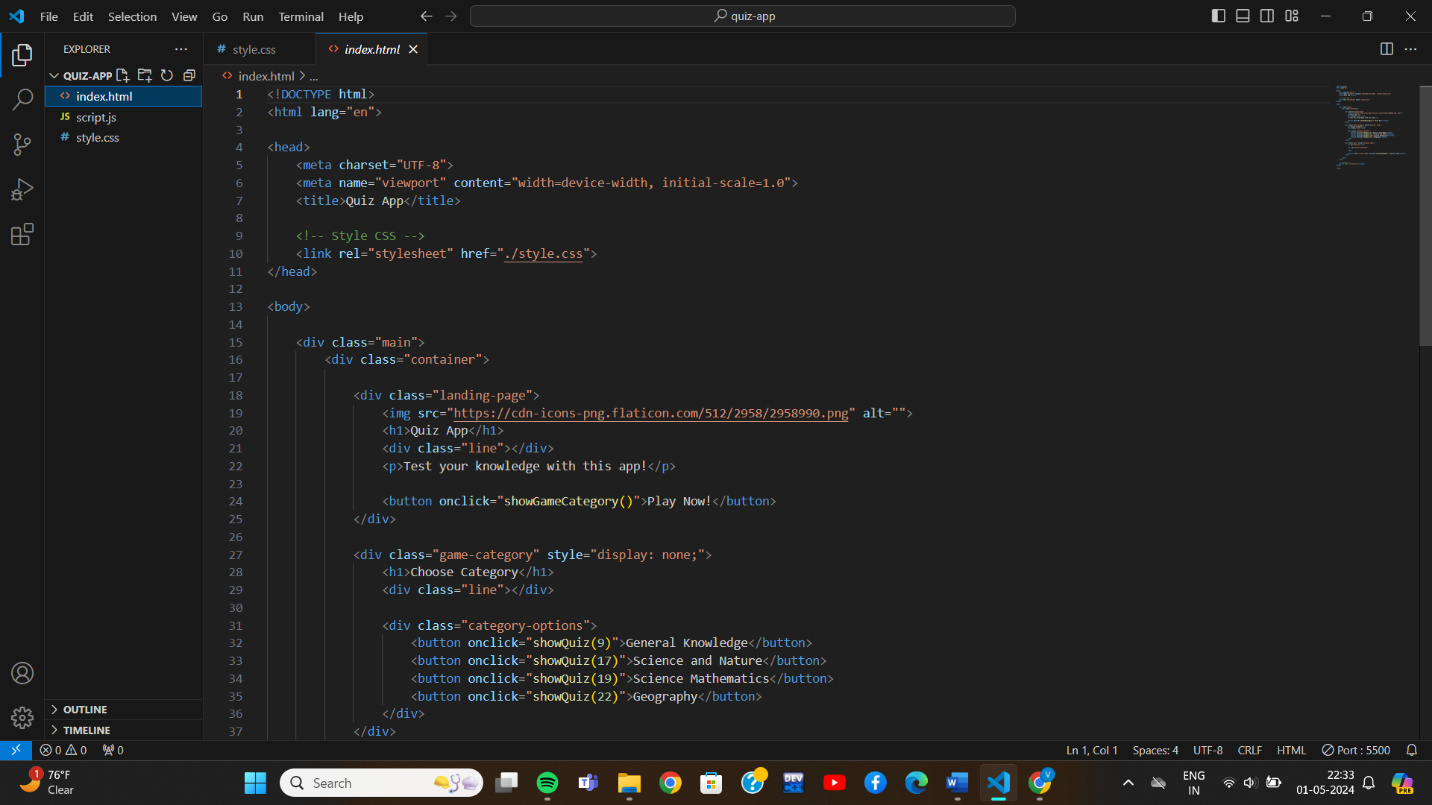
5. If the answer is correct it will give a correct. and if wrong it gives incorrect.

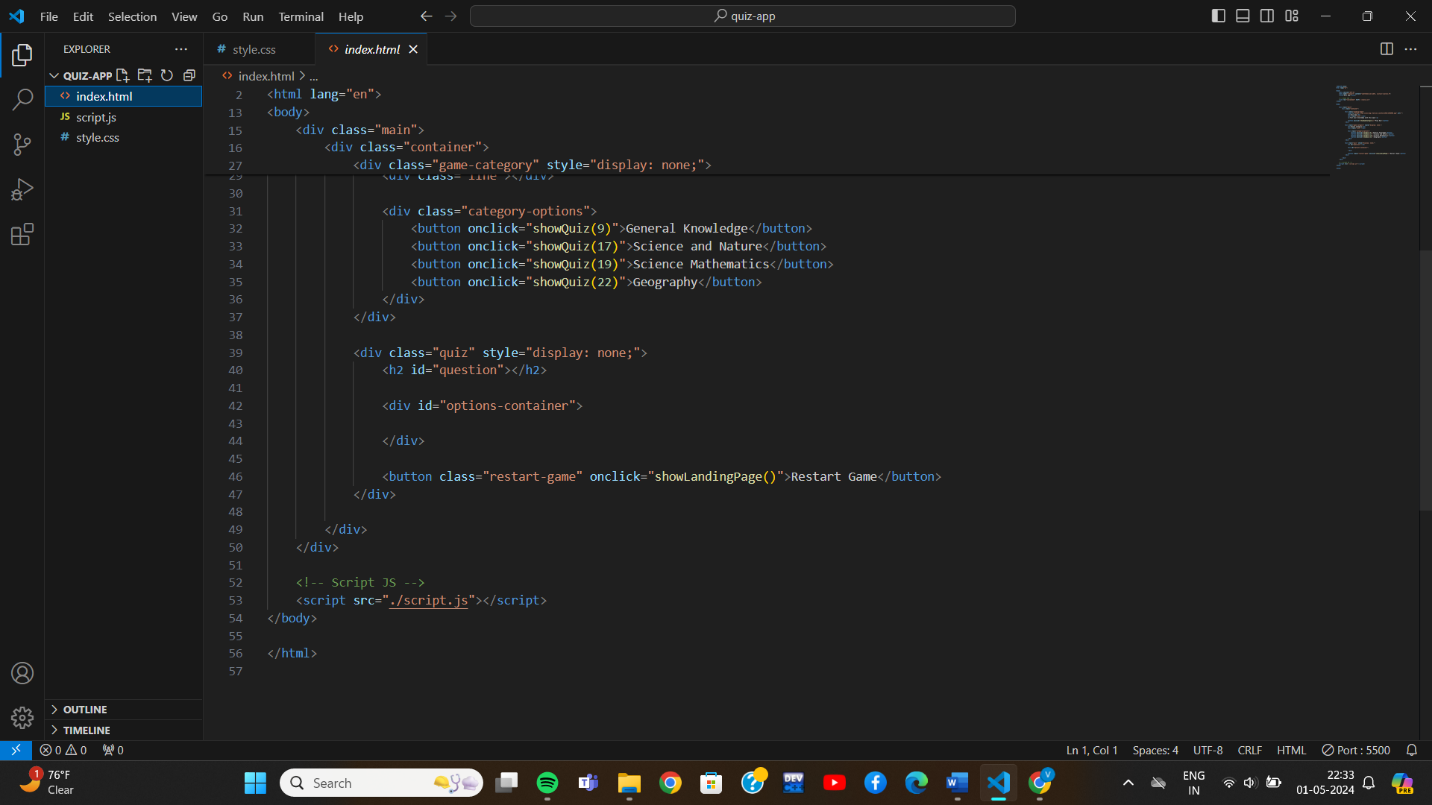
This algorithm provides a simple and effective way to implement accordion behaviour on the **Quiz application**  using vanilla JavaScript. However, frameworks like React can also be used to achieve similar functionality with additional features and optimizations.

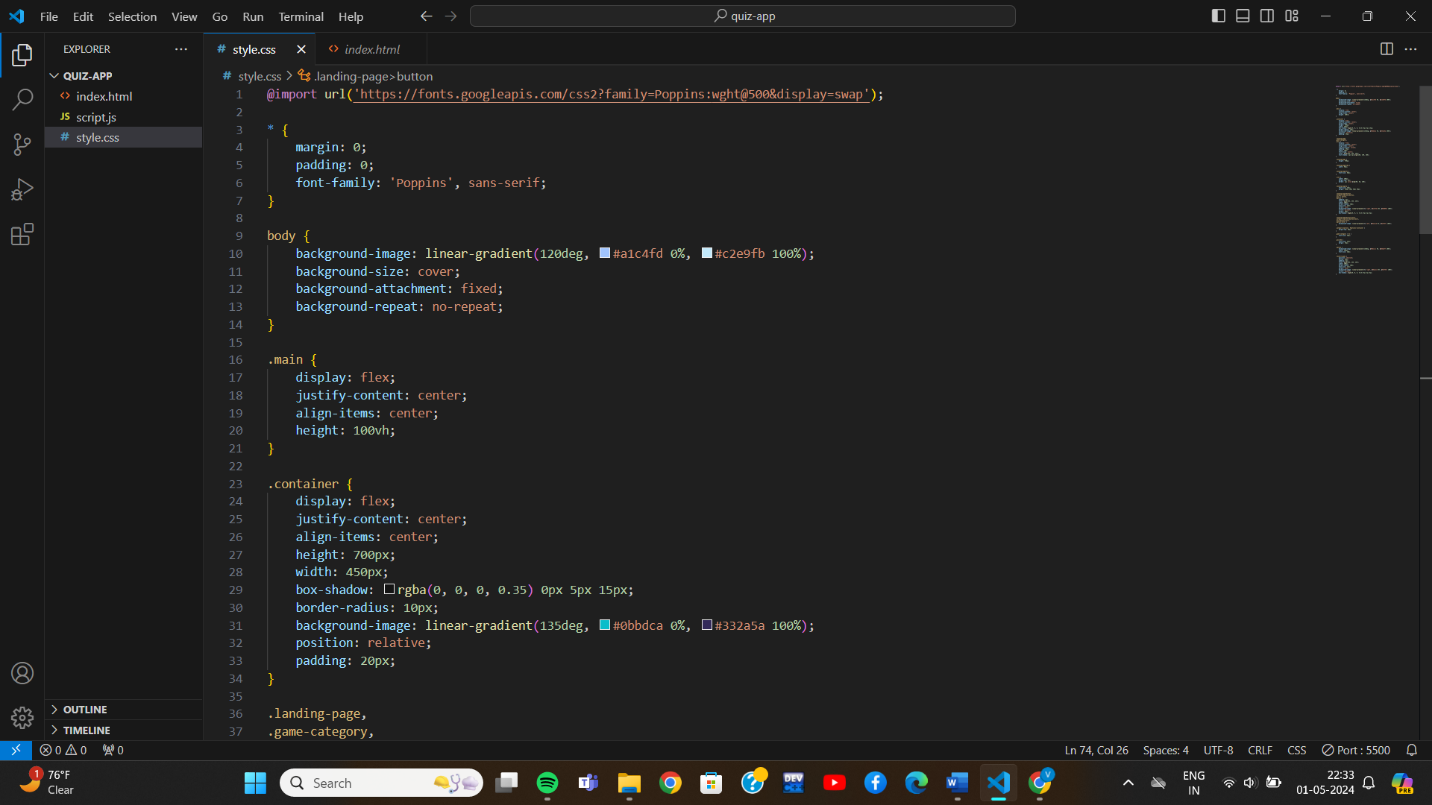
**FILE STRUCTURE:-**

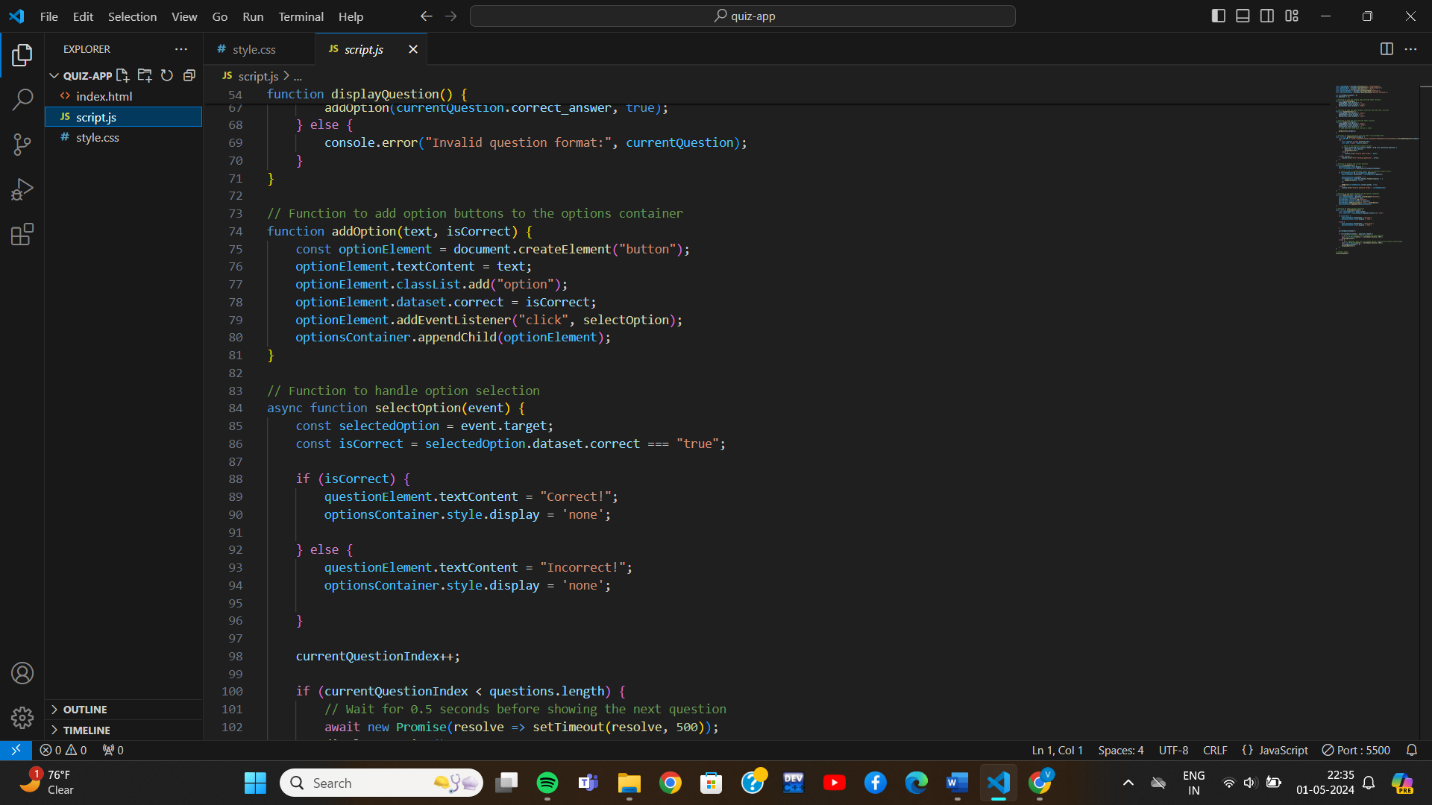
****

**Code**

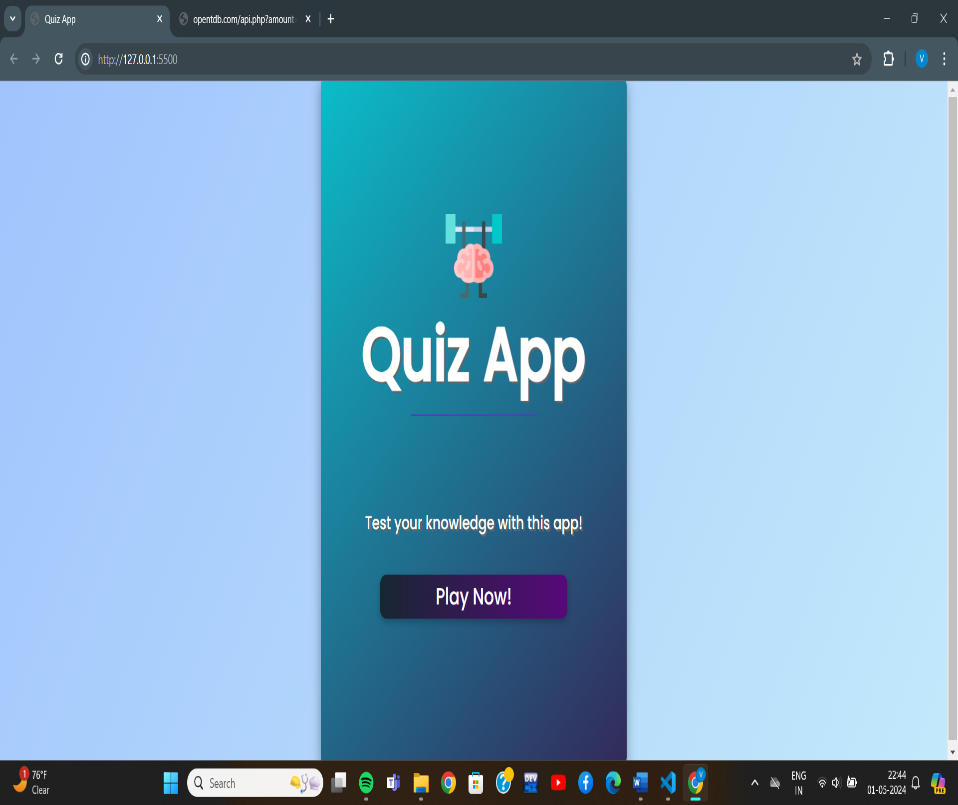
****

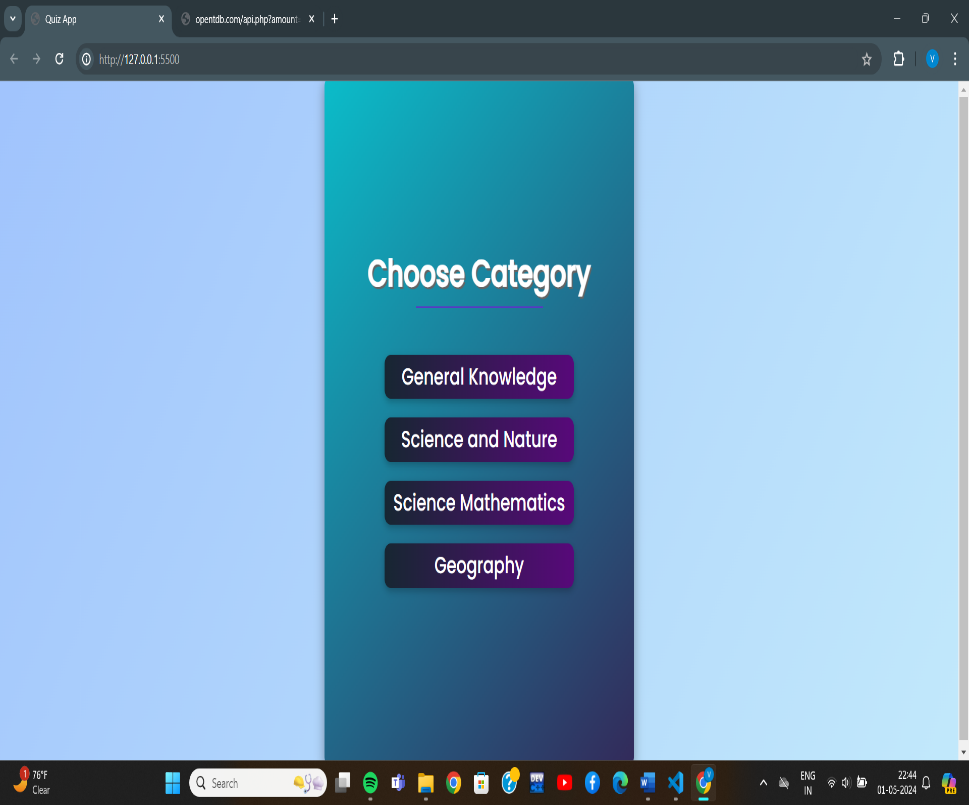
****

****

****

**Results**

****

****