## **ONLINE TEST APPLICATION SYSTEM**

## **Course-end Project 1**

**Specification Document: The Online Test Application system creates an application that enables users to provide online tests, review them, and display the results.**

**Project and Developer Details**

**Project Name: Online Test Application system.**

**Developer: Mr Innocent Malwane.**

**Contact Information:**

- Email: innocented00@gmail.com

- Phone: +27 68 127 3330

**Sprints and Task Achievements**

**Sprint 1**

**Duration: 1 week**

**Task 1:**

- Set up the development environment (IDE, version control).

- Define project structure (folders, files).

- Plan the overall architecture (backend, frontend).

- Create a detailed task breakdown (user stories, tasks).

**Task 2:**

- Design the JSON format for quiz questions.

- Develop the server-side API for fetching quiz questions.

- Implement client-side rendering of quiz questions.

- Create a basic interface for users to answer quiz questions.

**Task 3:**

- Design the user interface for reviewing quiz answers.

- Implement the functionality for users to navigate between questions.

- Allow users to submit their quiz answers.

**Sprint Goal**: Implement the fundamental components of the Quiz Module, allowing users to access and answer quiz questions, while also setting up the basic structure and project environment.

**Sprint 2**

**Duration: 1 week**

**Task 1:**

- Implement answer validation on the client side.

- Add a timer for quizzes with time constraints.

- Test the review module thoroughly.

**Task 2:**

- Design the JSON format for quiz results (user details, answers).

- Implement the server-side API for receiving and processing quiz results.

- Develop the client-side interface for users to submit quiz results.

**Sprint Goal**: Complete the Review Module, enabling users to review their quiz answers with validation and implement the first phase of the Result Module to handle user submissions.

**Sprint 3**

**Duration: 1 week**

**Task 1:**

- Implement the scoring mechanism for evaluating quiz results.

- Display the results to users immediately after submission.

- Test the result module thoroughly.

**Task 2:**

- Implement user authentication (login, registration).

- Integrate a database for storing quiz questions, user details, and results.

- Ensure secure handling of user data.

**Task 3:**

- Document the project, including code documentation and user guides.

- Perform a final code review.

- Address any outstanding issues or bugs.

- Set up deployment to a staging environment for testing.

- Deploy the application to a production environment.

**Sprint Goal:** Finalize the Result Module, implementing the scoring mechanism and result display. Additionally, integrate user authentication, database storage, and ensure the overall security of the application. Complete documentation, testing, and deployment for a production-ready application.

**Algorithms and Flowcharts**

**ALGORITHM FOR CREATING AN ONLINE TEST APPLICATION SYSTEM.**

**1. Requirement Gathering**

- Gather detailed requirements for the online test application, including user stories, features, and functionalities.

**2. Specification Document**

- Create a comprehensive document outlining the product's capabilities, appearance, and user interactions.

**3. Technology Stack Selection**

- Choose the technology stack: Frontend (HTML, CSS, JavaScript, possibly React or Angular).

**4. JSON Design**

- Design a JSON file to store the quiz questions and answers.

**5. Setting Up GitHub Repository**

- Create a GitHub repository for version control and collaborative development.

**6. Frontend Development**

- Start with basic UI development: create the homepage, quiz display pages, review page and results page.

- Implement responsive design and user-friendly interfaces.

**7. Testing and Debugging**

- Conduct extensive testing, including unit tests, integration tests, and user answer selection tests.

- Debug and resolve any issues or bugs identified during testing.

**8. Documentation**

- Document the entire development process, including architecture, technologies used, and how to operate the application.

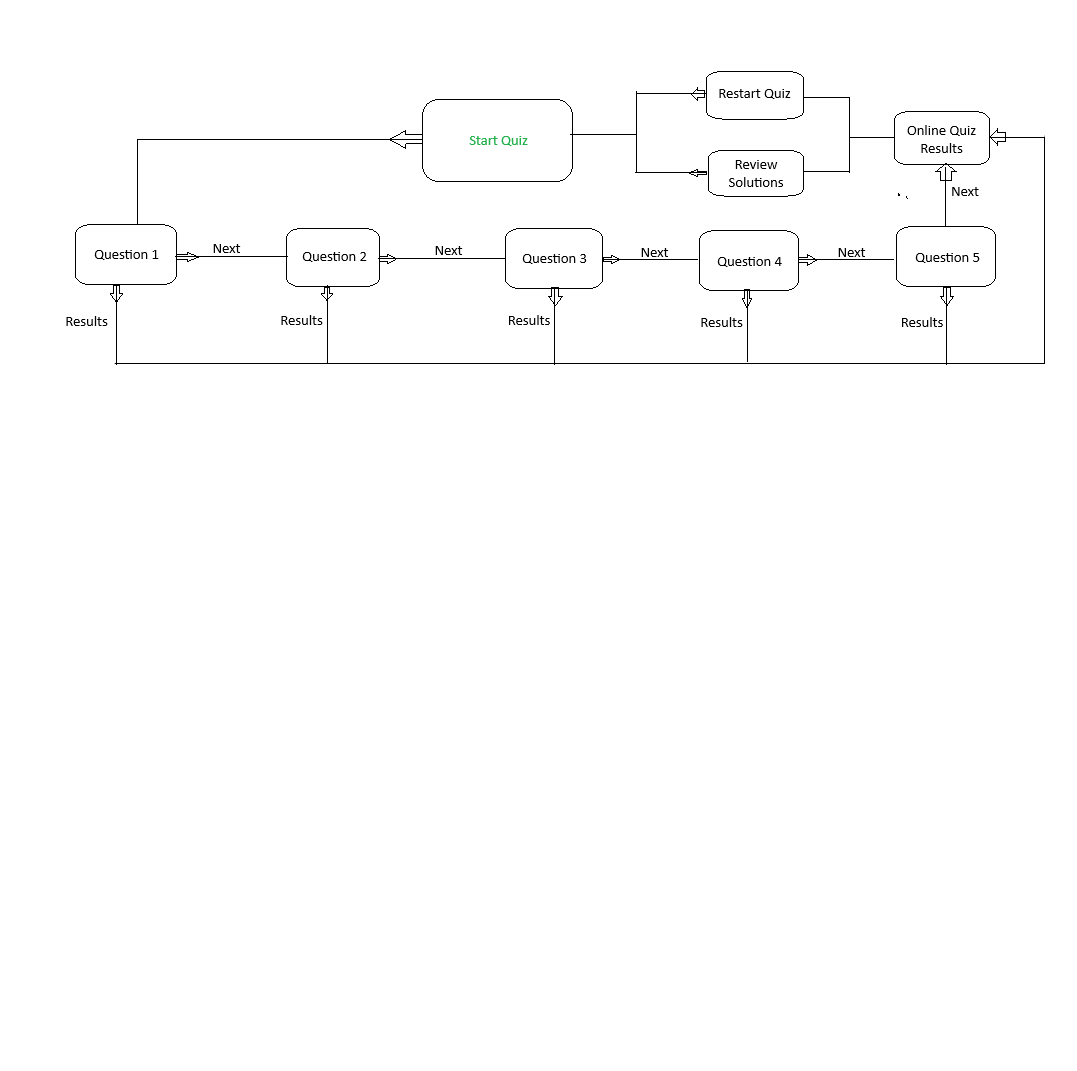
**9. Stakeholder Presentation Preparation**

- Prepare for the presentation by organizing documentation and showcasing the working application or prototype.

**10. Deployment**

- Deploy the application on a server or a hosting platform for users to access.

**Flowchart of the Application**

****

**Core Concepts Used**

**1. Frontend Development:**

- HTML/CSS/JavaScript: Fundamental technologies for creating the user interface and handling client-side interactions.

- Frontend Frameworks: Such as React, Angular, or Vue.js for building dynamic and responsive user interfaces.

**2. Backend Development:**

- JavaScript Object Notation (JSON): Representing structured data and transmitting data in the web application.

**3. Client-Server Communication:**

- HTTP/HTTPS Protocols: Use to transmit data between the client and server.

- AJAX/Fetch API: To make asynchronous requests to the server from the client.

**4. Web Security:**

- HTTPS: Ensure secure communication over the web.

- Input Validation: Validate user inputs on both the client and server sides.

- Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) Protection: Implement measures to prevent these common security vulnerabilities.

**5. Quiz Module:**

- JSON Handling: Design and implement a structured format for storing quiz questions and answers.

- Client-Side Rendering: Use JavaScript frameworks to dynamically render quiz questions on the client side.

**6. Review and Result Modules:**

- Client-Side Navigation: Enable users to navigate through questions during the review.

- Result Calculation: Implement algorithms to calculate and display quiz results.

**7. User Experience (UX) and User Interface (UI) Design:**

- Responsive Design: Ensure the application works seamlessly across different devices and screen sizes.

- User-Friendly Interface: Design an intuitive and easy-to-use interface for a positive user experience.

**8. Testing:**

- Unit Testing: Test individual components in isolation.

- Integration Testing: Verify that different components work together as expected.

- End-to-End Testing: Simulate user interactions to test the entire application flow.

**9. Deployment:**

- Cloud Services: Use services like AWS, Azure, or Heroku for hosting.

- Containerization: Docker for containerizing the application.

- Continuous Integration/Continuous Deployment (CI/CD): Automate the deployment process for efficient updates.

**GitHub Repository**

You can verify the project completion by visiting the GitHub repository:

**[GitHub Repository]** [https://github.com/Dodge4k/Online-Application-Test-System-Phase-4.git]

**Application Source Code**

**All Source code files are in the GitHub Repo. Inclusion of the files will make this Specification Doc too long**

**Start of code (Index)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Online Test Application</title>

    <link rel="stylesheet" href="style.css"/>

</head>

<body>

    <div class="container">

        <div class="leftSide">

        <img src="https://static-00.iconduck.com/assets.00/logo-js-icon-2048x2048-ptuzd8a3.png"alt="quiz logo"class="tilt">

        </div>

    <div class="rightSide"><br>

        <h1>JavaScript Online Quiz</h1><br>

        <ul style="list-style-type:circle">

            <li><b>A Quiz on JavaScript</b></li>

            <li><b>Consists Of 5 Questions</b></li>

            <li><b>Answer All Questions</b></li>

        </ul>

        <a href="questions.html">Start Quiz</a>

        </div>

    </div>

</body>

<script src="js/jquery-3.5.1.min.js"></script>

<script src="quiz.js"></script>

<script src="script.js"></script>

</html>

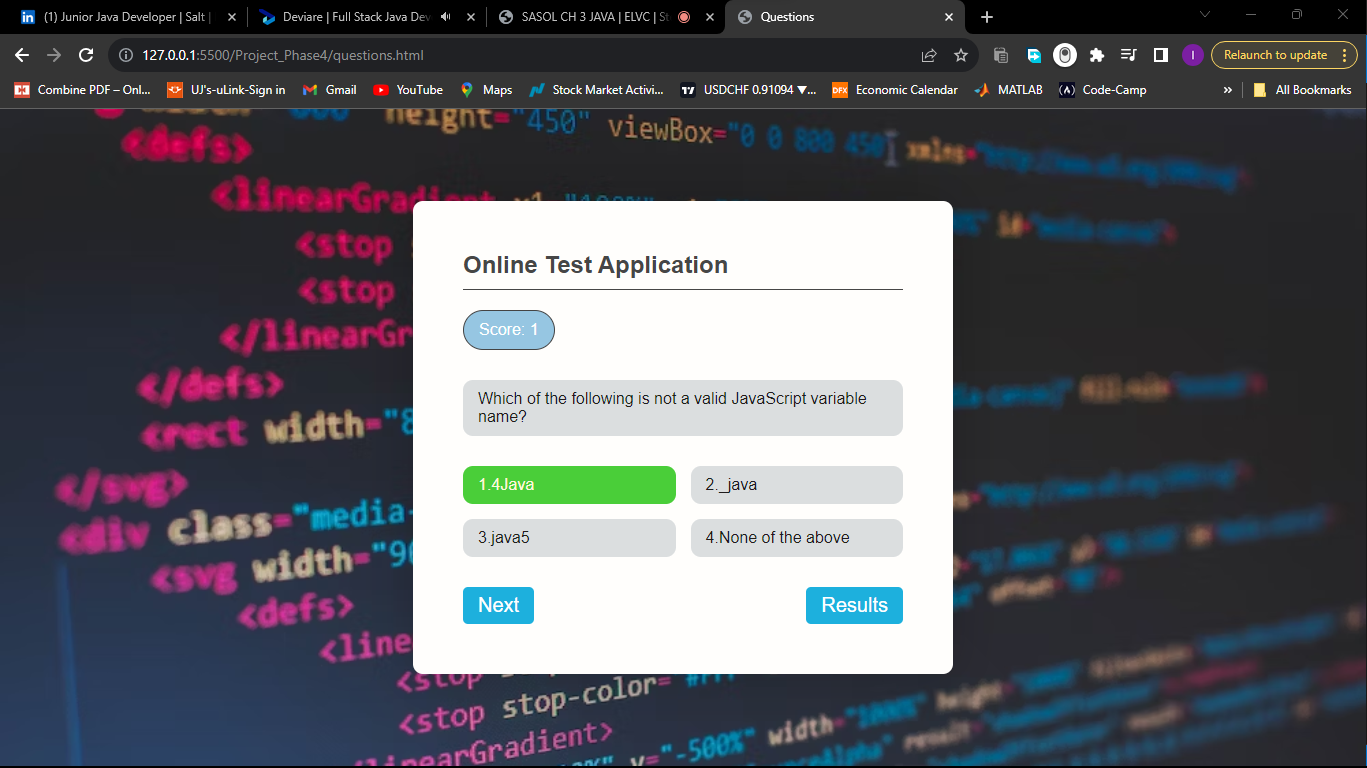
**End of code**

**Screenshots of the application in action**

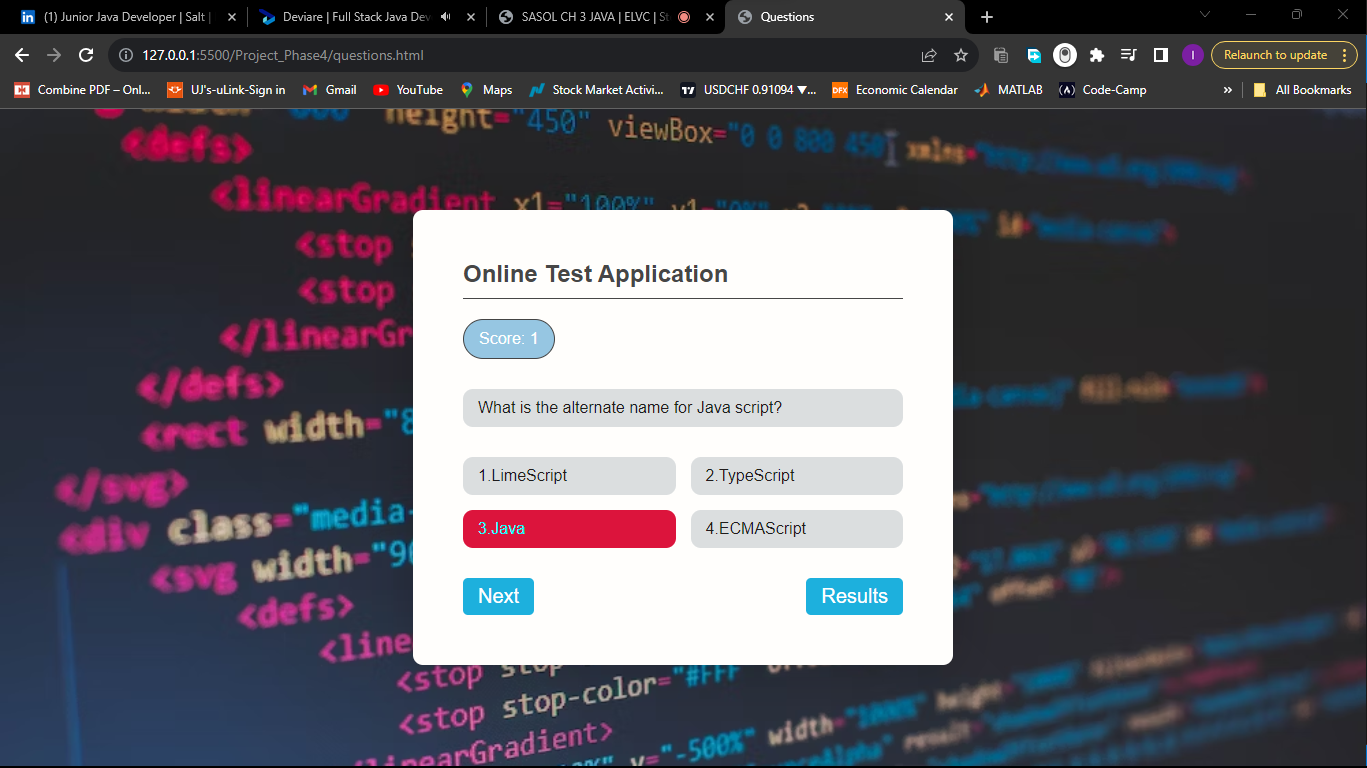
index page of the application

****

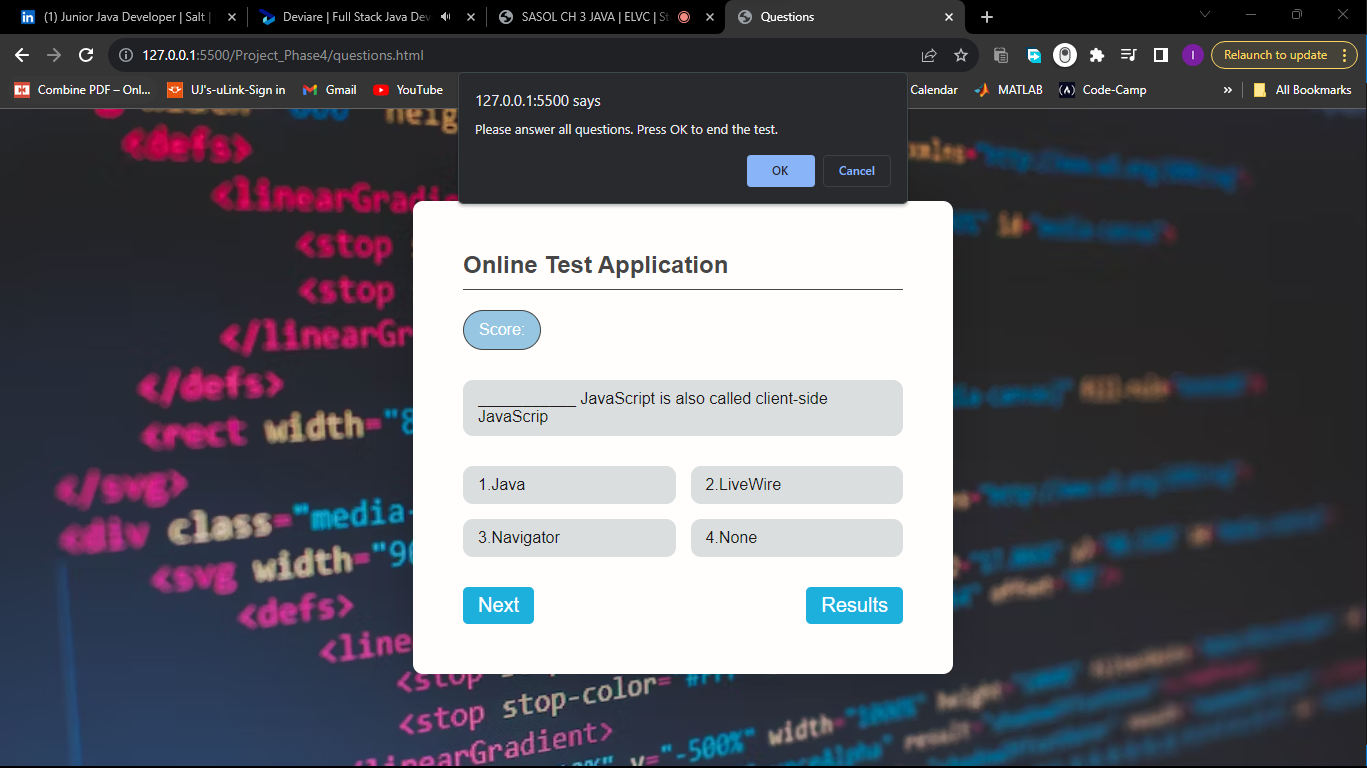
Correct answer selection

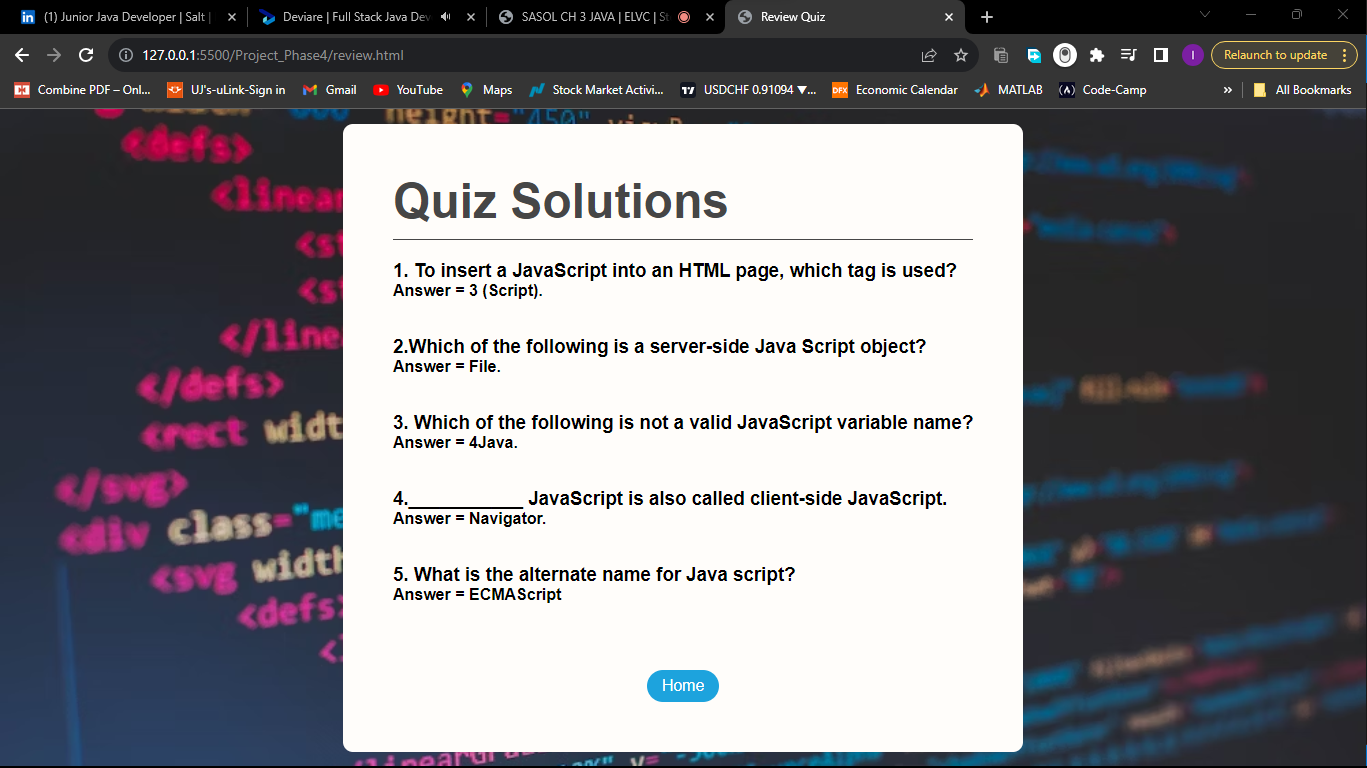
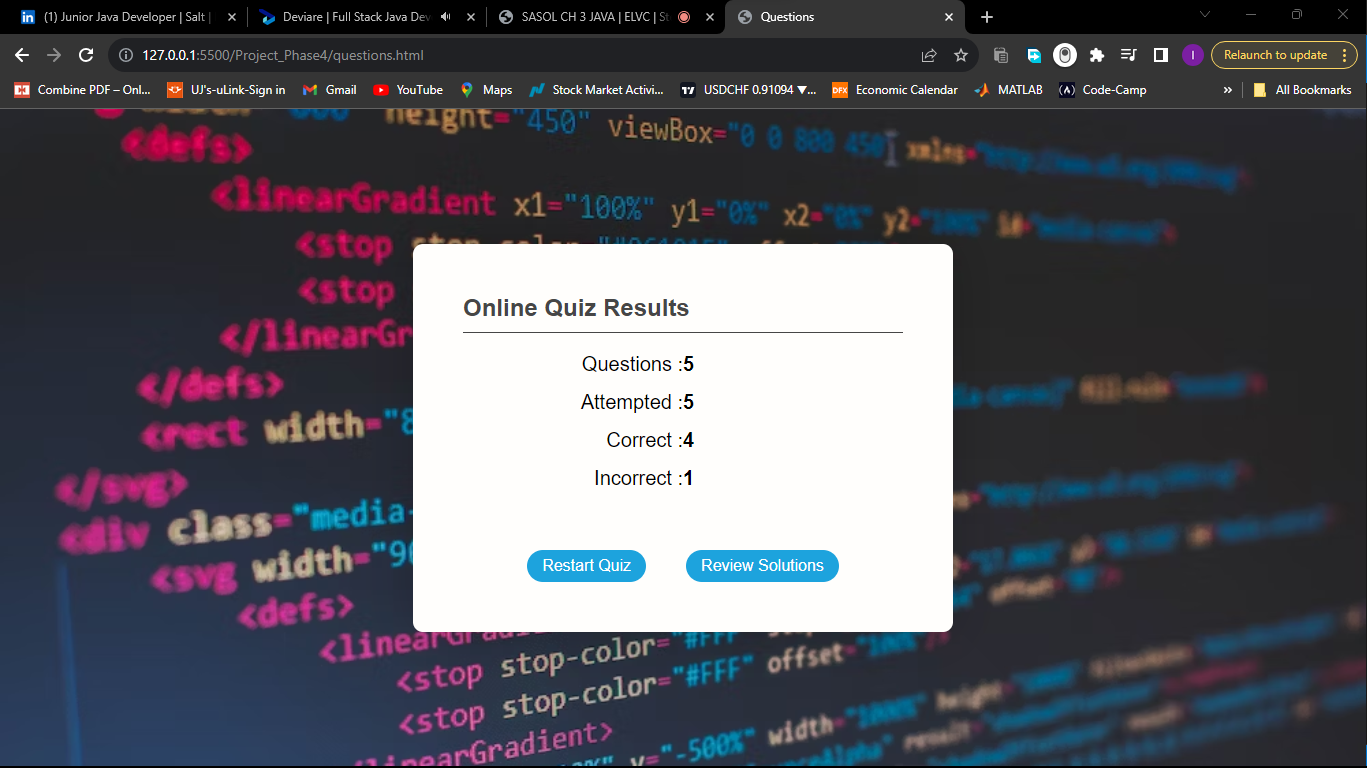
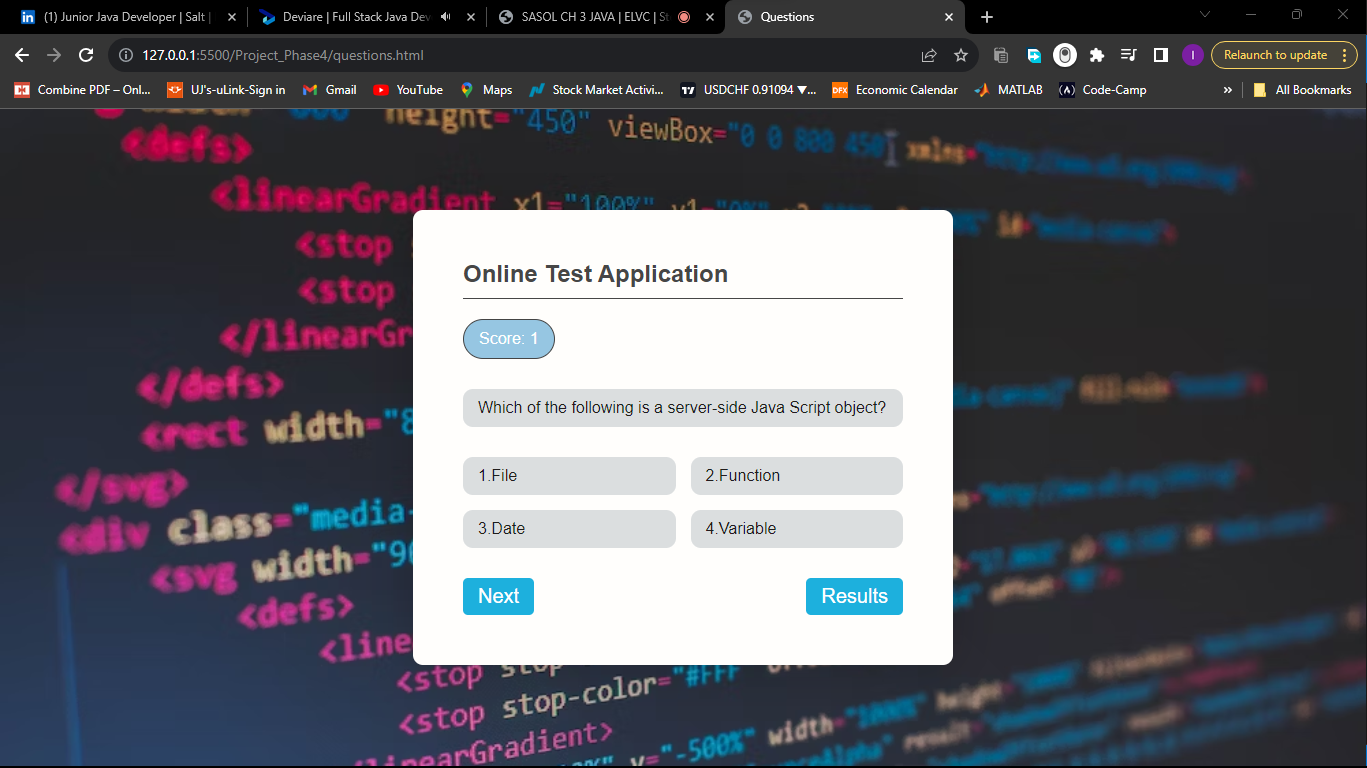
****

Incorrect answer selection

****

When you press results before answering all questions

****

****