## **Front End Engineering-II**

Project Report

Semester-IV (Batch-2022)

**Quiz App**



**Supervised By: Submitted By:**

Raveesh Samkaria Vineet Saini

2210990963

G-14

**Department of Computer Science and Engineering**

## **Chitkara University Institute of Engineering & Technology,**

## **Chitkara University, Punjab**

**Abstract**

The Quiz App is a web-based application that provides users with engaging and interactive quizzes. Built using Bootstrap, the app offers a responsive design that ensures a seamless user experience across various devices. This responsiveness is achieved through Bootstrap's components and utilities, which allow the app to adapt to different screen sizes without compromising on functionality or visual appeal. One of the key features of the Quiz App is its user-friendly interface, which is designed to be intuitive and easy to navigate.

Bootstrap's components are used to create a clean and organized layout, making it simple for users to find and take quizzes. Additionally, the app offers customizable quiz options, allowing users to create quizzes tailored to their preferences. This includes setting the number of questions, time limits, and scoring options.

The Quiz App also provides an interactive quiz-taking experience, with features such as real-time feedback, progress tracking, and score calculation. Users can see their results immediately after completing a quiz, providing them with instant gratification and motivation to continue using the app. Furthermore, an admin dashboard is available to manage quizzes, view quiz results, and monitor user activity, making it easy for educators, trainers, and organizations to create and manage quizzes for their audience.

1. **Introduction**
   1. **Background**

The quiz website "Quiz" is designed to offer a wide range of quizzes to users of all ages. The website aims to provide entertaining and educational content that caters to diverse interests and knowledge levels. Quiz's target audience includes students, professionals, and trivia enthusiasts looking for a fun and interactive way to test their knowledge and learn new facts.

Operating in the competitive online entertainment and education space, Quiz aims to differentiate itself by offering a user-friendly interface, a vast library of quizzes on various topics, and engaging features such as leaderboard rankings and social sharing options. The website also seeks to monetize through targeted advertising and premium quiz content for users seeking an ad-free experience or exclusive quizzes.

With a focus on user engagement and content quality, Quiz aims to become a go-to destination for quiz enthusiasts and a valuable resource for those looking to learn, have fun, and challenge themselves with quizzes.

* 1. **Objectives**

1. The objective of a quiz website is to provide an engaging and interactive platform for users to participate in quizzes. The website aims to entertain and educate users on various topics while offering a fun and challenging experience. Key objectives of a quiz website may include:
2. Engagement: To attract users and keep them engaged with interesting and diverse quizzes.
3. Education: To provide informative content through quizzes that help users learn new things or test their knowledge.
4. Entertainment: To offer a fun and enjoyable experience for users of all ages.
5. Interaction: To encourage interaction among users through features such as sharing quiz results and competing with friends.
6. Monetization: For commercial quiz websites, the objective may include generating revenue through advertising, premium quiz content, or other monetization strategies.
7. Branding: To build and promote a brand through quizzes that align with the brand's values and target audience.
8. Overall, the objective of a quiz website is to create a dynamic and engaging platform that attracts and retains users while providing valuable content and entertainment.
   1. **Significance**

The significance of a quiz website lies in its ability to offer a range of benefits to both users and website owners. Some key points of significance include:

Engagement: Quiz websites provide a highly engaging and interactive experience for users, encouraging them to spend more time on the site.

Education: Quizzes can be educational, helping users learn new information or reinforce existing knowledge in a fun and interactive way.

Entertainment: Quizzes are inherently entertaining, providing users with a fun and enjoyable experience.

Traffic Generation: A well-designed quiz website can attract a large number of visitors, increasing traffic to the site.

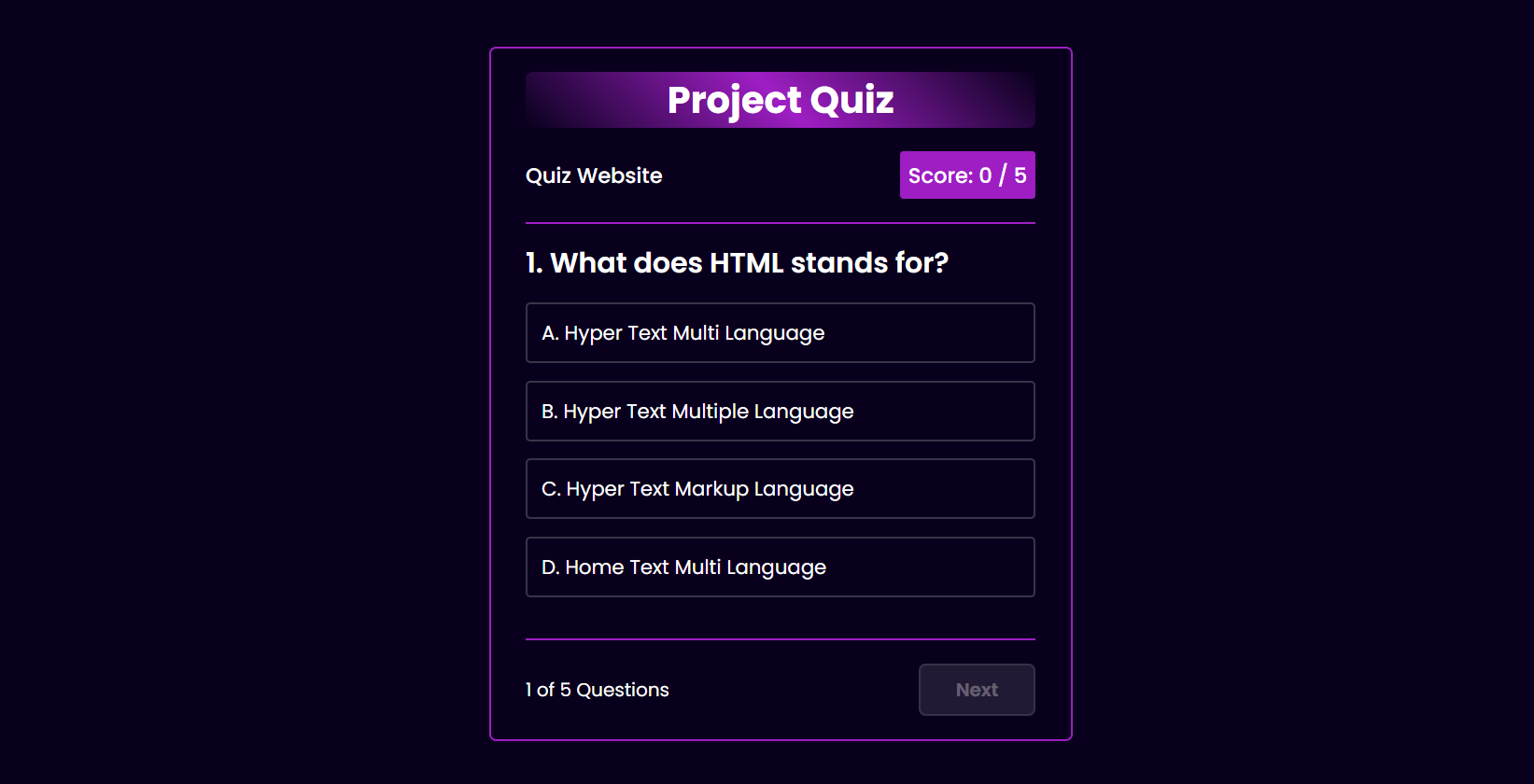
User Interaction: Quizzes encourage users to interact with the website and share their results with others, fostering a sense of community.

Monetization: Quiz websites can be monetized through various means, such as advertising, sponsored quizzes, or premium content.

Brand Building: For businesses, quiz websites can help build brand awareness and promote products or services in a subtle and engaging way.

Data Collection: Quizzes can be used to collect valuable data about users, such as their preferences and interests, which can be used for targeted marketing or content personalization.





1. **Problem Definition and Requirements**
   1. **Problem Statement**

Despite the popularity of online quizzes, existing quiz websites often lack the features and user experience necessary to fully engage users and provide a satisfying quiz-taking experience. Many quiz websites have limited quiz options, lack customization features, and fail to provide real-time feedback and scoring, leading to a less-than-optimal user experience.

Additionally, existing quiz websites may not effectively cater to the diverse interests and knowledge levels of users, limiting their appeal and engagement. There is a need for a quiz website that offers a wide range of quizzes on various topics, allows users to customize their quiz experience, provides real-time feedback and scoring, and fosters a sense of community through interactive features.

Furthermore, existing quiz websites may not fully leverage the potential for monetization through targeted advertising, premium quiz content, and other revenue streams. There is an opportunity to create a quiz website that not only provides a great user experience but also offers lucrative monetization options for the website owner.

In light of these challenges and opportunities, there is a need for a comprehensive quiz website that addresses the shortcomings of existing platforms and provides a feature-rich, engaging, and customizable quiz-taking experience for users of all ages and interests.

* + 1. **Software Requirements**

The software requirements for the Quiz Website project include the following:

* + - Front-end Framework: Utilize Bootstrap for responsive design and layout.
    - Back-end Development: Use a server-side language like Node.js, Python, or PHP to handle quiz logic and data management.
    - Database: Implement a database system (e.g., MySQL, MongoDB) to store quiz questions, user data, and quiz results.
    - User Authentication: Include user authentication functionality to manage user accounts and access to quizzes.
    - Quiz Creation: Implement a feature that allows users to create and customize quizzes with different question types (multiple choice, true/false, etc.) and scoring options.
    - Quiz Taking: Develop a user-friendly interface for taking quizzes, displaying questions, and providing real-time feedback and scoring.
    - Result Display: Show users their quiz results, including scores and correct/incorrect answers, with the option to share results on social media.
    1. **Hardware Requirements**

The hardware requirements for running and developing the Random Quote Generator project are relatively minimal and include the following:

* + - Server: A server capable of running the chosen back-end technology (e.g., Node.js, Python, PHP). The server should have sufficient CPU and RAM to handle the expected traffic and workload.
    - Database Server: A separate server or instance for the database system (e.g., MySQL, MongoDB). The database server should also have sufficient resources to handle database operations efficiently.
    - Storage: Adequate storage space for storing quiz questions, user data, and other content. Consider using solid-state drives (SSDs) for improved performance.
    - Networking: A stable internet connection with sufficient bandwidth to handle incoming and outgoing traffic to the website.
    - Backup Solution: Implement a backup solution to regularly back up data and ensure data integrity and availability in case of hardware failure or data loss.

1. **Proposed Design/Methodology**
   * 1. **Design Concept**

User-Centric Design: The design should prioritize the user experience, with a clean and intuitive interface that makes it easy for users to find and take quizzes.

Responsive Design: Utilize Bootstrap to ensure the website is responsive and works seamlessly across devices of all sizes, including desktops, tablets, and smartphones.

Engaging Visuals: Use visually appealing graphics, colors, and typography to enhance the overall look and feel of the website and make it more engaging for users.

Clear Navigation: Implement a clear and user-friendly navigation system that allows users to easily browse and find quizzes based on their interests.

Interactive Elements: Include interactive elements such as animations, hover effects, and interactive quiz components to enhance user engagement and make the quiz-taking experience more enjoyable.

Consistent Branding: Maintain consistent branding throughout the website, including the use of logos, colors, and fonts, to reinforce the website's identity and create a cohesive user experience.

* 1. **Methodology**

Requirement Analysis: Gather and analyze the requirements for the quiz website, including features, user expectations, and business goals.

Design Planning: Plan the overall design of the website, including the user interface, navigation structure, and visual elements.

Front-end Development: Use Bootstrap to develop the front-end of the website, focusing on creating a responsive and visually appealing design.

Back-end Development: Develop the back-end of the website using a suitable server-side language (e.g., Node.js, Python, PHP) and database system (e.g., MySQL, MongoDB).

Integration: Integrate the front-end and back-end components to ensure they work seamlessly together.

Testing: Conduct thorough testing of the website to identify and fix any bugs or issues, ensuring the website functions as intended across different devices and browsers.

Deployment: Deploy the website to a server or hosting platform, making it accessible to users.

Maintenance: Regularly maintain and update the website to ensure it remains secure, up-to-date, and meets the evolving needs of users and the business.

Optimization: Continuously optimize the website for performance, user experience, and search engine visibility to attract and retain users.

Feedback and Iteration: Collect feedback from users and stakeholders to identify areas for improvement and make iterative updates to the website.

**HTML**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

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

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.rtl.min.css" integrity="sha384-dpuaG1suU0eT09tx5plTaGMLBsfDLzUCCUXOY2j/LSvXYuG6Bqs43ALlhIqAJVRb" crossorigin="anonymous">

<title>Quiz</title>

</head>

<body>

<nav class="navbar navbar-expand-lg fixed-top font-size-lg">

<div class="container-fluid">

<a class="navbar-brand" href="#">Quiz.</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarTogglerDemo02"

aria-controls="navbarTogglerDemo02" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarTogglerDemo02">

<ul class="navbar-nav me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link" aria-current="page" href="#">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Services</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">About</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Contact</a>

</li>

</ul>

</div>

</div>

</nav>

<div class="content website-start">

<div class="quiz-about">

<div class="quiz-about1">Quiz Website</div>

<p class="quiz-about2">Lorem ipsum dolor sit amet consectetur adipisicing elit. Cumque dicta eaque obcaecati alias sit, sequi minus tenetur autem fugit velit! Amet iste nihil ullam quae.</p>

<button type="button" class="btn btn-outline-light btn-lg" onclick="redirectToURL()">Quizz</button>

</div>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous">

</script>

<script>

function redirectToURL(){

window.location.href='http://127.0.0.1:5500/index2.html';

}

</script>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<link rel="stylesheet" href="style2.css">

<title>Quiz App</title>

</head>

<body>

<main class="main">

<header class="header">

<!-- <a href="#" class="logo">Quiz.</a>

<nav class="navbar">

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Services</a>

<a href="#">Contact</a>

</nav> -->

</header>

<div class="container">

<section class="quiz-section">

<div class="quiz-box">

<h1>Project Quiz</h1>

<div class="quiz-header">

<span>Quiz Website</span>

<span class="header-score">Score: 0 / 5</span>

</div>

<h2 class="question-text">What does HTML stands for?</h2>

<div class="option-list">

<!-- <div class="option">

<span>A. Hyper Text Multi Language</span>

</div>

<div class="option">

<span>B. Hyper Text Multiple Language</span>

</div>

<div class="option">

<span>C. Hyper Text Markup Language</span>

</div>

<div class="option">

<span>D. Home Text Multi Language</span>

</div> -->

</div>

<div class="quiz-footer">

<span class="question-total">1 of 5 Questions</span>

<button class="next-btn">Next</button>

</div>

</div>

<div class="result-box">

<h2>Quiz Result</h2>

<div class="percentage-container">

<div class="circular-progress">

<span class="progress-value">0%</span>

</div>

<span class="score-text">Your Score 0 out of 5</span>

</div>

<div class="buttons">

<button class="tryAgain-btn">Try Again</button>

<button class="goHome-btn">Go To Home</button>

</div>

</div>

</section>

<section class="home">

<div class="home-content">

<h1>Let's Quiz !</h1>

<button class="start-btn">Start Quiz</button>

</div>

</section>

</div>

</main>

<div class="popup-info">

<h2>Quiz Guide</h2>

<span class="info">1. Lorem ipsum dolor sit amet adipisicing.</span>

<span class="info">2. Lorem ipsum dolor sit amet consectetur adipisicing.</span>

<span class="info">3. Lorem ipsum dolor sit amet here consectetur adipisicing.</span>

<span class="info">4. Lorem ipsum dolor sit ahead amet consectetur adipisicing.</span>

<span class="info">5. Lorem sit amet occur consectetur adipisicing.</span>

<div class="btn-group">

<button class="info-btn exit-btn">Exit Quiz</button>

<a href="#" class="info-btn continue-btn">Continue</a>

</div>

</div>

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

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

</body>

</html>

**CSS**

\*{

margin: 0;

padding: 0;

}

.navbar-custom{

background-color: rgb(77, 119, 156);

}

.body {

background-color: seagreen;

}

.website-start {

background-image: url(assets/images/background.jpg);

height: 100vh;

z-index: 100;

display: flex;

flex-direction: column;

}

.website-start .quiz-about{

justify-content: center;

align-items: center;

color: white;

display: flex;

flex-direction: column;

padding-top: 10%;

padding-inline: 25%;

}

.quiz-about1{

display: flex;

flex-direction: column;

justify-content: center;

padding-bottom: 3%;

align-items: center;

font-size: 80px;

text-align: center;

}

.quiz-about p{

text-align: center;

padding-bottom: 2%;

font-size: 20px;

justify-content: center;

align-items: center;

}

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700;800;900&display=swap');

\*{

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'poppins', sans-serif;

}

.body{

color: #fff;

background: #09001d;

overflow: hidden;

}

.header {

position: fixed;

top: 0;

left: 0;

width: 100%;

padding: 20px 10%;

background: transparent;

display: flex;

justify-content: space-between;

align-items: center;

z-index: 100;

}

.logo{

font-size: 32px;

color: #fff;

text-decoration: none;

font-weight: 700;

filter: drop-shadow(0 0 5px #09001d)

}

.navbar a{

font-size: 18px;

color: #fff;

text-decoration: none;

font-weight: 500;

margin-left: 35px;

transition: .3s;

}

.navbar :hover{

color: #d81b80f3;

}

.main{

background: url(./assets/images/background.jpg);

background-repeat: no-repeat;

min-height: 100vh;

background-size: cover;

background-position: center;

overflow-x: hidden;

}

.main.active{

filter: blur(15px)

}

.container{

display: flex;

height: 100vh;

width: 200%;

/\* background:#09001d; \*/

}

.home{

position: relative;

left: -50%;

width: 100%;

height: 100vh;

display: flex;

color: #fff;

justify-content: center;

align-items: center;

}

.home-content{

max-width: 600px;

display: flex;

align-items: center;

flex-direction: column;

}

.home-content h1 {

font-size: 78px;

font-weight: 700;

text-shadow: 0 0 10px rgba(0, 0, 0, .3);

}

.home-content p {

font-size: 16px;

text-align: center;

text-shadow: 0 0 10px rgba(0, 0, 0, .3);

margin-bottom: 30px;

}

.home-content .start-btn {

width: 200px;

height: 60px;

background: #a620ccf3;

border: 2px solid #fff;

outline: none;

border-radius: 6px;

box-shadow: 0 0 10px #a620ccf3;

font-size: 18px;

color: #fff;

letter-spacing: 1px;

font-weight: 600;

cursor: pointer;

transition: .5s;

}

.home-content .start-btn:hover{

background-color: #09001d;

box-shadow: none;

}

.popup-info{

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%) scale(.9);

width: 500px;

background: #fff;

border-radius: 6px;

padding: 10px 25px;

opacity: 0;

pointer-events: none;

transition: .3s ease;

}

.popup-info.active{

opacity: 1;

pointer-events: auto;

transform: translate(-50%, -50%) scale(1);

}

.popup-info h2 {

font-size: 50px;

color: #a620ccf3;

}

.popup-info .info {

display: inline-block;

font-size: 16px;

color: #333;

font-weight: 500;

margin: 4px 0;

}

.popup-info .btn-group {

display: flex;

justify-content: space-between;

align-items: center;

border-top: 1px solid #999;

margin-top: 10px;

padding: 15px 0 7px;

}

.popup-info .btn-group .info-btn {

display: inline-flex;

justify-content: center;

align-items: center;

width: 130px;

height: 45px;

background: #a620ccf3;

border: 2px solid #a620ccf3;

outline: none;

border-radius: 6px;

text-decoration: none;

font-size: 16px;

color: white;

font-weight: 600;

box-shadow: 0 0 10px rgba(0, 0, 0, .1);

cursor: pointer;

transition: .5s;

}

.popup-info .btn-group .info-btn:nth-child(1) {

background: transparent;

color: #a620ccf3;

}

.popup-info .btn-group .info-btn:nth-child(1):hover {

background: #a620ccf3;

color: white;

}

.popup-info .btn-group .info-btn:nth-child(2):hover{

background: #bf25eaf3;

border-color: #bf25eaf3;

}

.quiz-section {

position: relative;

left: -50%;

/\* left: 0; \*/

width: 100%;

z-index: 100;

background: #09001d;

color: #fff;

display: flex;

justify-content: center;

align-items: center;

transition: .8s ease-in-out;

transition-delay: .25s;

}

.quiz-section.active{

left: 0;

}

.quiz-section .quiz-box {

position: relative;

width: 500px;

background: transparent;

border: 2px solid #a620ccf3;

border-radius: 6px;

display: flex;

flex-direction: column;

padding: 20px 30px;

opacity: 0;

pointer-events: none;

transform: scale(.9);

transition: .3s ease;

transition-delay: 0s;

}

.quiz-section .quiz-box.active{

opacity: 1;

pointer-events: auto;

transform: scale(1);

transition: 1s ease;

transition-delay: 1s;

}

.quiz-box h1{

font-size: 32px;

text-align: center;

background: linear-gradient(45deg, transparent, #a620ccf3, transparent);

border-radius: 6px;

}

.quiz-box .quiz-header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 20px 0;

border-bottom: 2px solid #a620ccf3;

margin-bottom: 15px;

}

.quiz-header span {

font-size: 18px;

font-weight: 500;

}

.quiz-header .header-score {

background: #a620ccf3;

border-radius: 3px;

padding: 7px;

}

.quiz-box .question-text {

font-size: 24px;

font-weight: 600;

}

.option-list .option {

width: 100%;

padding: 12px;

background: transparent;

border: 2px solid rgba(255, 255, 255, .2);

border-radius: 4px;

font-size: 17px;

margin: 15px 0;

cursor: pointer;

transition: .3s;

}

.option-list .option:hover{

background: rgba(255, 255, 255, .1);

border-color: rgba(255, 255, 255, .1);

}

.option-list .option.correct {

background: #09001d;

color: #00a63d;

border-color: #00a63d;

}

.option-list .option.incorrect {

background: #09001d;

color: #a60045;

border-color: #a60045;

}

.option-list .option.disabled {

pointer-events: none;

}

.quiz-box .quiz-footer {

display: flex;

justify-content: space-between;

align-items: center;

border-top: 2px solid #a620ccf3;

padding-top: 20px;

margin-top: 20px;

}

.quiz-footer .quiz-total {

font-size: 16px;

font-weight: 600;

}

.quiz-footer .next-btn {

width: 100px;

height: 45px;

background: rgba(255, 255, 255, .1);

border: 2px solid rgba(255, 255, 255, .1);

outline: none;

border-radius: 6px;

font-size: 16px;

color: rgba(255, 255, 255, .3);

font-weight: 600;

cursor: pointer;

pointer-events: none;

transition: .5s;

}

.quiz-footer .next-btn.active {

pointer-events: auto;

background: #a620ccf3;

border-color: #a620ccf3;

color: #fff;

}

.quiz-footer .next-btn.active:hover{

background: #bf25eaf3;

border-color: #bf25eaf3;

}

.quiz-section .result-box {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 500px;

background: transparent;

border: 2px solid #a620ccf3;

border-radius: 6px;

padding: 20px;

display: flex;

flex-direction: column;

align-items: center;

opacity: 0;

pointer-events: none;

transition: .3s ease;

}

.quiz-section .result-box.active {

opacity: 1;

pointer-events: auto;

transform: translate(-50%, -50%) scale(1);

}

.result-box h2 {

font-size: 52px;

}

.result-box .percentage-container {

width: 300px;

display: flex;

flex-direction: column;

align-items: center;

margin: 20px 0 40px;

}

.percentage-container .circular-progress {

position: relative;

width: 250px;

height: 250px;

background: conic-gradient(#a620ccf3 3.6deg, rgba(255, 255, 255, .1) 0deg);

border-radius: 50%;

display: flex;

justify-content: center;

align-items: center;

}

.percentage-container .circular-progress::before{

content: '';

position: absolute;

width: 210px;

height: 210px;

background: #09001d;

border-radius: 50%;

}

.circular-progress .progress-value {

position: relative;

font-size: 45px;

font-weight: 600;

}

.percentage-container .score-text {

font-size: 26px;

font-weight: 600;

margin-top: 20px;

}

.result-box .buttons {

display: flex;

}

.buttons button {

width: 130px;

height: 47px;

background: #a620ccf3;

border: 2px solid #a620ccf3;

outline: none;

border-radius: 6px;

font-size: 16px;

color: #fff;

font-weight: 600;

margin: 0 20px 20px;

cursor: pointer;

transition: .5s;

}

.buttons button:nth-child(1):hover {

background: #bf25eaf3;

border-color: #bf25eaf3;

}

.buttons button:nth-child(2) {

background: transparent;

color: #a620ccf3;

}

.buttons button:nth-child(2):hover {

background: #a620ccf3;

color: #fff;

}

**JavaScript**

const startBtn = document.querySelector('.start-btn');

const popupInfo = document.querySelector('.popup-info');

const exitBtn = document.querySelector('.exit-btn');

const main = document.querySelector('.main');

const continueBtn = document.querySelector('.continue-btn');

const quizSection = document.querySelector('.quiz-section');

const quizBox = document.querySelector('.quiz-box');

const resultBox = document.querySelector('.result-box');

const tryAgainBtn = document.querySelector('.tryAgain-btn');

const goHomeBtn = document.querySelector('.goHome-btn');

function redirectToURL() {

window.location.href = "http://127.0.0.1:5500/index2.html";

}

startBtn.onclick = () => {

popupInfo.classList.add('active');

main.classList.add('active');

}

exitBtn.onclick = () => {

popupInfo.classList.remove('active');

main.classList.remove('active');

}

continueBtn.onclick = () => {

quizSection.classList.add('active');

popupInfo.classList.remove('active');

main.classList.remove('active');

quizBox.classList.add('active');

showQuestions(0);

questionCounter(1);

headerScore();

}

tryAgainBtn.onclick = () => {

quizBox.classList.add('active');

nextBtn.classList.remove('active');

resultBox.classList.remove('active');

questionCount = 0;

questionNumb = 1;

userScore = 0;

showQuestions(questionCount);

questionCounter(questionNumb);

headerScore();

}

goHomeBtn.onclick = () => {

quizSection.classList.remove('active');

nextBtn.classList.remove('active');

resultBox.classList.remove('active');

questionCount = 0;

questionNumb = 1;

userScore = 0;

showQuestions(questionCount);

questionCounter(questionNumb);

headerScore();

}

let questionCount = 0;

let questionNumb = 1;

let userScore = 0;

const nextBtn = document.querySelector('.next-btn');

nextBtn.onclick = () => {

if(questionCount < questions.length - 1){

questionCount++;

showQuestions(questionCount);

questionNumb++;

questionCounter(questionNumb);

nextBtn.classList.remove('active');

}

else{

showResultBox();

}

}

const optionList = document.querySelector('.option-list');

function showQuestions(index) {

const questionText = document.querySelector('.question-text');

questionText.textContent = `${questions[index].numb}. ${questions[index].question}`;

let optionTag = `<div class="option"><span>${questions[index].options[0]}</span></div>

<div class="option"><span>${questions[index].options[1]}</span></div>

<div class="option"><span>${questions[index].options[2]}</span></div>

<div class="option"><span>${questions[index].options[3]}</span></div>`;

optionList.innerHTML = optionTag;

const option = document.querySelectorAll('.option');

for (let i = 0; i < option.length; i++) {

option[i].setAttribute('onclick', 'optionSelected(this)');

}

}

function optionSelected(answer) {

let userAnswer = answer.textContent;

let correctAnswer = questions[questionCount].answer;

let allOptions = optionList.children.length;

console.log(correctAnswer);

if(userAnswer == correctAnswer) {

answer.classList.add('correct');

userScore += 1;

headerScore();

}

else{

answer.classList.add('incorrect');

for (let i=0; i< allOptions;i++) {

if(optionList.children[i].textContent == correctAnswer) {

optionList.children[i].setAttribute('class', 'option correct');

}

}

}

for (let i=0; i< allOptions;i++) {

optionList.children[i].classList.add('disabled');

}

nextBtn.classList.add('active');

}

function questionCounter(index){

const questionTotal = document.querySelector('.question-total');

questionTotal.textContent = `${index} of ${questions.length} Questions`;

}

function headerScore() {

const headerScoreText = document.querySelector('.header-score');

headerScoreText.textContent = `Score: ${userScore} / ${questions.length}`;

}

function showResultBox(){

quizBox.classList.remove('active');

resultBox.classList.add('active');

const scoreText = document.querySelector('.score-text');

scoreText.textContent = `Your Score ${userScore} out of ${questions.length}`;

const circularProgress = document.querySelector('.circular-progress');

const progressValue = document.querySelector('.progress-value');

let progressStartValue = -1;

let progressEndValue = (userScore / questions.length) \* 100;

let speed = 20;

let progress = setInterval(() => {

progressStartValue++;

progressValue.textContent = `${progressStartValue}%`;

circularProgress.style.background = `conic-gradient(#a620ccf3 ${progressStartValue \* 3.6}deg, rgba(255, 255, 255, .1) 0deg)`;

if(progressStartValue == progressEndValue) {

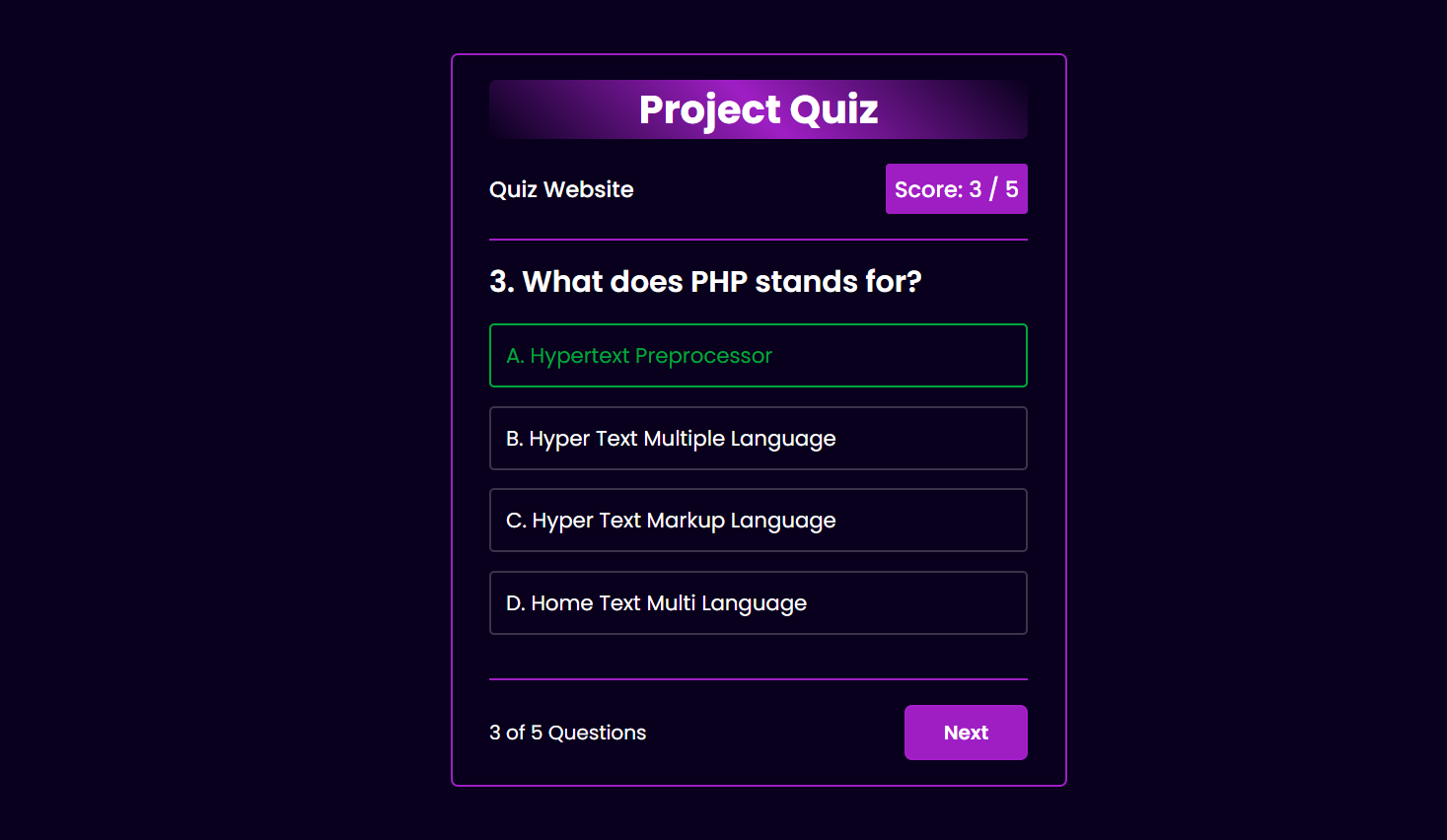
clearInterval(progress);

}

}, speed);

}

1. **Results**
   * 1. **Screenshots**





# **References**

1. <https://www.w3schools.com/>
2. <https://fonts.google.com/>
3. <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.3.0/css/all.min.css>
4. <https://www.youtube.com/>