**Software Requirements**

**Specification**

**for**

**QUIZ APP TO TRACK KNOWLEDGE**

**Prepared by**

**BAWYASRI K - 23CSR031**

**DEEPIKA M - 23CSR040**

**DHANUSH S - 23CSR043**

**Date: February 27,2025**

# Table of Contents

[Tableof Contents ii](#_TOC_250033)

[Revision History ii](#_TOC_250032)

1. [Introduction 1](#_TOC_250031)
   1. [Purpose 1](#_TOC_250030)
   2. [Document Conventions 1](#_TOC_250029)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250028)
   4. [Product Scope 1](#_TOC_250027)
   5. [References 1](#_TOC_250026)
2. [Overall Description 2](#_TOC_250025)
   1. [Product Perspective 2](#_TOC_250024)
   2. [Product Functions 2](#_TOC_250023)
   3. [User Classes and Characteristics 2](#_TOC_250022)
   4. [Operating Environment 2](#_TOC_250021)
   5. [Design and Implementation Constraints 2](#_TOC_250020)
   6. [User Documentation 2](#_TOC_250019)
   7. [Assumptions and Dependencies 3](#_TOC_250018)
3. [ExternalInterfaceRequirements 3](#_TOC_250017)
   1. [User Interfaces 3](#_TOC_250016)
   2. [Hardware Interfaces 3](#_TOC_250015)
   3. [Software Interfaces 3](#_TOC_250014)
   4. [CommunicationI nterfaces 3](#_TOC_250013)
4. [SystemFeatures 4](#_TOC_250012)
   1. User selects a quiz category [4](#_TOC_250011)
   2. User answers quiz questions [4](#_TOC_250010)
   3. User receives quiz results [4](#_TOC_250010)
   4. User exits or retakes the quiz [4](#_TOC_250010)
5. [OtherNonfunctional Requirements 4](#_TOC_250009)
   1. [Performance Requirements 4](#_TOC_250008)
   2. [Safety Requirements 5](#_TOC_250007)
   3. [Security Requirements 5](#_TOC_250006)
   4. [Software QualityAttributes 5](#_TOC_250005)
6. [Other Requirements 5](#_TOC_250003)

[AppendixA: Glossary 5](#_TOC_250002)

[AppendixB:AnalysisModels 5](#_TOC_250001)

[AppendixC:ToBeDeterminedList 6](#_TOC_250000)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Quiz App is designed to provide an engaging and educational platform where users can test their knowledge across various topics, track scores, and receive performance feedback. The application supports a user-friendly interface for quiz-taking and results tracking.

## Document Conventions

This document adheres to the IEEE 830-1998 standard format for Software Requirements Specification. All requirements are numbered and categorized to facilitate easy reference. Terminology follows industry standards, and technical terms are explained where necessary. Formatting includes headings, subheadings, and bullet points to enhance readability. Any assumptions and dependencies are explicitly stated.

## Intended Audience and Reading Suggestions

* **Developers** - To understand system functionalities.
* **Testers** - To verify the system meets the specified requirements.
* **Project Managers** - To oversee project scope and progress.
* **End Users** - To understand the application's capabilities.

## Product Scope

The Quiz App is a web-based application designed to help users enhance their knowledge in various domains through interactive quizzes. The app enables users to select topics, answer questions, receive instant feedback, and track their progress over time. The system ensures an engaging and educational experience through a user-friendly interface.

## References

1. Online Quiz Application Best Practices - <https://www.edtechmagazine.com>
2. Web-Based Learning Systems - <https://www.educause.edu>

# Overall Description

## Product Perspective

The Quiz app is a standalone web-based application that provides a unified platform for users to take test and track their score. It integrates external APIs for real-time data and is optimized for desktop and mobile use. The system follows a microservices architecture to allow seamless updates and scalability. It ensures high availability and reliability for users worldwide.

## Product Functions

* Users can select quizzes from various knowledge domains.
* Randomized questions ensure a fresh experience every time.
* Score tracking allows users to monitor their progress.
* Responsive design ensures usability on both desktop and mobile devices.
* Future updates may include leaderboards and user authentication for progress tracking.

## User Classe sand Characteristics

The application is designed for two primary user classes: general users and administrators. General users are individuals who wish to test their knowledge and track their progress over time. They interact with the application by selecting quiz categories, answering questions, and viewing results. Administrators, on the other hand, manage quiz content, including adding, modifying, or removing questions. Both user classes benefit from an intuitive interface that simplifies their interactions with the system.

## Operating Environment

The Quiz App is a web-based application that runs on modern web browsers, including Chrome, Firefox, Edge, and Safari. It is designed to be responsive and fully functional across desktop and mobile devices. The system does not require additional software installation, making it accessible from any device with an internet connection. JavaScript must be enabled in the browser to ensure smooth interaction. The application is optimized for a fast and seamless user experience across different platforms.

## Design and Implementation Constraints

The Quiz App is developed using HTML, CSS, Bootstrap, and JavaScript, ensuring a lightweight and responsive design. Since it is a web-based application, it requires a stable internet connection to function correctly. The initial version does not include backend support, meaning user progress is not saved across sessions. Future updates may introduce database integration for storing user scores and progress. Additionally, compatibility across different browsers and devices is prioritized to enhance accessibility.

## User Documentation

The application will provide a comprehensive user guide detailing how to start and complete quizzes. This guide will include step-by-step instructions on selecting topics, submitting answers, and interpreting results. An FAQ section will be available to address common queries related to scoring, feedback, and technical issues. Future versions may include tutorial videos to assist new users in navigating the platform effectively.

## Assumptions and Dependencies

The application depends on a stable internet connection to function properly. Since no backend storage is implemented in the initial version, user progress will not be saved beyond a single session. The system relies on JavaScript-based functionalities for user interactions, meaning disabling JavaScript may result in limited usability. Future enhancements may include database integration for persistent user tracking.

# External Interface Requirements

## User Interfaces

The Quiz App features a clean and intuitive user interface designed for seamless navigation. The home page provides users with a list of available quiz categories to choose from. The quiz interface displays randomized questions along with multiple-choice answers, allowing users to submit their selections. A results page summarizes user performance by displaying scores, correct answers, and feedback. The responsive design ensures accessibility on both desktop and mobile devices, providing a consistent user experience.

## Hardware Interfaces

The portal supports standard desktop, laptop, tablet, and smartphone devices. It is optimized for various screen resolutions. No additional hardware is required for operation. The platform runs on cloud infrastructure for high availability. Server-side components handle user requests efficiently.

## Software Interfaces

The Quiz App is developed using HTML, CSS, Bootstrap, and JavaScript, ensuring a responsive and interactive user experience. The system runs on modern web browsers and does not require additional software installations. Future updates may introduce a backend system using databases to store user progress and authentication details. The application may also integrate third-party APIs for additional quiz content or analytics. Data transfer and storage will follow standard security protocols to ensure data integrity and protection.

## Communications Interfaces

The application operates over the internet using standard HTTP/HTTPS protocols to ensure secure data transmission. All user interactions with the system occur through web-based interfaces, eliminating the need for local installations. Future versions may include WebSocket communication for real-time interactions, such as timed quizzes or multiplayer quiz challenges. Email notifications may be introduced to provide users with quiz results and progress updates. Secure encryption methods will be implemented to safeguard user data during communication.

# System Features

### **User selects a quiz category**

The user begins by selecting a category from a list of available quiz topics. This ensures that they engage with questions relevant to their interests or areas of study. The categories are displayed in an intuitive interface, allowing easy navigation. Once a category is chosen, the system loads the corresponding set of randomized questions for the quiz session.

### **User answers quiz questions**

After selecting a category, the system presents a series of multiple-choice questions. The user reads each question carefully and selects an answer from the given options. The interface provides a seamless experience, allowing users to move forward to the next question once they submit an answer. Immediate feedback is provided after each submission to enhance learning.

### **User receives quiz results**

Upon completing the quiz, the system calculates the user's final score based on the number of correct answers. The results page displays a summary, including the total score, correct and incorrect answers, and personalized feedback. Users can review their performance to identify strengths and areas for improvement.

### **User exits or retakes the qui**

After reviewing their results, the user has the option to exit the application or retake the quiz. If they choose to retake, they can either attempt the same category again with a new set of randomized questions or select a different category. This feature allows users to continually test and improve their knowledge in various subjects.

# Other Nonfunctional Requirements

### **Performance Requirements**

The Quiz App should ensure fast and efficient performance to provide a seamless user experience. The system must load quiz pages within two seconds and process user responses instantly. The application should handle multiple concurrent users without significant delays or crashes. Performance optimization techniques such as caching and efficient code execution should be implemented to enhance responsiveness.

### **Safety Requirements**

The system must ensure the integrity and reliability of user interactions. Users should be prevented from submitting multiple answers to the same question. Error handling mechanisms should be in place to alert users in case of failed submissions. The application should provide proper guidance in case of connectivity issues to prevent data loss or quiz disruption.

### **Security Requirements**

The application should follow best security practices to protect user interactions and data integrity. Future versions may include authentication features to secure user accounts and maintain quiz history. All communications between the client and server (if applicable) should be encrypted using HTTPS protocols. Security measures should be taken to prevent unauthorized access or tampering with quiz content.

### **Software Quality Attributes**

The system should maintain high reliability and availability to ensure continuous user access. It should be designed for scalability, allowing the addition of new quiz categories and features without affecting performance. The user interface should be intuitive, ensuring a smooth and engaging experience for users. Future enhancements should maintain compatibility with various devices and browsers while adhering to accessibility standards.

# Other Requirements

The Quiz App should be adaptable for future enhancements, including the ability to integrate a backend database for storing user progress and performance history. The system should be designed with scalability in mind, allowing additional quiz categories and question banks to be incorporated seamlessly. The application should also comply with accessibility guidelines to ensure usability for users with disabilities. Furthermore, localization options should be considered to support multiple languages, expanding the reach of the application to a global audience. Future versions may include monetization features, such as premium quizzes or subscription-based access for advanced content.

# Appendix A: Glossary

### **Quiz:** A set of multiple-choice or short-answer questions designed to test the user's knowledge in a specific subject or topic.

### **Score Tracking:** A system feature that records and displays the user's quiz performance, including correct answers and overall score.

### **Feedback:** Information provided to users after submitting an answer, indicating whether the response was correct or incorrect, and explaining the correct answer if needed.

### **User Interface (UI):** The graphical layout of the application, including screens, buttons, and navigation elements, that allow users to interact with the system.

### **Responsive Design:** An approach to web design ensuring that the application functions well on different devices, including desktops, tablets, and mobile phones.

### **Randomized Questions:** A feature that ensures quiz questions appear in a different order each time a user takes the quiz, providing a unique experience with every attempt.

# Appendix B: Analysis Models

### **System Architecture:** The Quiz App follows a client-server architecture where the frontend is developed using HTML, CSS, Bootstrap, and JavaScript, ensuring an interactive and responsive user experience. Future enhancements may include a backend database for storing quiz questions and user scores. The system is designed to be scalable, allowing for the addition of more quiz categories and advanced features without affecting performance.

### **Data Flow Diagram:** The application follows a structured data flow, starting with user input when selecting a quiz category. The system processes the request, fetches randomized questions, and displays them to the user. Upon submission, the answers are processed, and feedback is generated instantly. The final scores are then presented to the user at the end of the quiz session.

### **User Interaction Flow:** The user initiates the process by selecting a quiz topic from the available options. The system responds by retrieving relevant questions and displaying them in an interactive format. As the user submits answers, the system evaluates correctness and provides immediate feedback. At the end of the session, a performance summary is generated, allowing users to analyze their progress and determine areas for improvement.

### **User selects a quiz category:** The user begins by selecting a category from a list of available quiz topics. This ensures that they engage with questions relevant to their interests or areas of study. The categories are displayed in an intuitive interface, allowing easy navigation. Once a category is chosen, the system loads the corresponding set of randomized questions for the quiz session.

### **User answers quiz questions:** After selecting a category, the system presents a series of multiple-choice questions. The user reads each question carefully and selects an answer from the given options. The interface provides a seamless experience, allowing users to move forward to the next question once they submit an answer. Immediate feedback is provided after each submission to enhance learning.

### **User receives quiz results:** Upon completing the quiz, the system calculates the user's final score based on the number of correct answers. The results page displays a summary, including the total score, correct and incorrect answers, and personalized feedback. Users can review their performance to identify strengths and areas for improvement.

### **User exits or retakes the quiz:** After reviewing their results, the user has the option to exit the application or retake the quiz. If they choose to retake, they can either attempt the same category again with a new set of randomized questions or select a different category. This feature allows users to continually test and improve their knowledge in various subjects.

# Appendix C: To Be Determined List

### **Supported Question Types:** The system currently supports multiple-choice questions, but future updates may introduce fill-in-the-blank, true/false, and short-answer questions.

### **User Authentication and Profiles:** The initial version does not include user authentication; however, future enhancements may allow users to create accounts and track their quiz history.

### **Quiz Timer and Difficulty Levels:** The system does not yet include a timer for quizzes or categorized difficulty levels. Future versions may implement timed quizzes and adjustable difficulty settings.

### **Backend Integration:** Currently, the system operates without a backend database. A future iteration may integrate a database to store user scores and quiz performance for long-term tracking.

### **Multiplayer and Challenge Mode:** Future updates may introduce a multiplayer feature where users can challenge friends to quizzes and compare scores in real-time.