# Quiz Quest

# Documentation

1. **Introduction**

**1.1 Purpose**

This document specifies the requirements for Quiz Quest, an advanced front-end quiz website designed to offer an engaging and interactive quiz experience. The purpose of Quiz Quest is to provide users with a responsive and dynamic quiz platform with features such as a leaderboard, dark mode, question categories, difficulty levels, animated transitions, sound effects, responsive timers, and score-sharing options.

**1.2 Scope**

Quiz Quest is a web-based application built using HTML, CSS, JavaScript, and Bootstrap. Its primary functions include:

* Interactive quiz functionality with multiple-choice questions.
* Selection of question categories and difficulty levels.
* Score tracking and leaderboard system.
* Animated transitions and UI enhancements.
* Sound effects and haptic feedback for mobile users.
* Dark mode toggle with user preference saving.
* Review of answers after quiz completion.
* Dynamic question loading from an API or JSON file.
* Social media score-sharing features.

Future enhancements include backend integration for user authentication and a multiplayer quiz mode.

**1.3 Overview**

This document details the system’s functional and non-functional requirements, overall system description, and design constraints. It is intended for stakeholders, developers, and testers involved in the project.

1. **Overall Description**

**2.1 Product Perspective**

Quiz Quest is designed as a standalone web application that mimics commercial quiz platforms. The system is modular, allowing for future integration of additional features such as user authentication and multiplayer modes.

* 1. **Product Functions**

**Quiz Functionality:**

* Users can select a category before starting the quiz.
* Multiple-choice questions are displayed with a progress indicator.
* Users receive immediate feedback on correct and incorrect answers.

**Leaderboard System:**

* Scores are sorted in descending order.
* Displays the top 5 players.
* Shows "Your Rank" for comparison.

**Dark Mode Toggle:**

* Users can switch between light and dark themes.
* Theme preference is saved in local storage.

**Animated Transitions & UI Enhancements:**

* Smooth fade-in animations for questions and options.
* Highlighting correct & wrong answers (green for correct, red for wrong).
* Hover effects on buttons for better UX.

**Sound Effects & Haptic Feedback:**

* Plays a sound when selecting an answer (different for correct & wrong answers).
* Haptic feedback support for mobile users.

**Responsive Timer Enhancements:**

* Countdown progress bar for visual tracking.
* Bonus points for answering quickly.

**Score Sharing & Achievements:**

* Allows sharing scores on Twitter and Facebook.
* Unlockable badges for milestones (e.g., "Scored 100 points", "Completed 10 quizzes").

**Review Answers After Quiz Completion:**

* Users can review correct answers after finishing the quiz.
* Highlights user’s chosen answer with explanations.

**Dynamic Question Loading:**

* Questions are fetched from an API or JSON file.
* Implements randomized question selection for variety.
* Future-ready for backend integration.

1. **Specific Requirements**

**3.1 Functional Requirements**

**FR1**: The system shall allow users to select quiz categories and difficulty levels.

**FR2**: The system shall display multiple-choice questions with real-time feedback.

**FR3**: The system shall track scores and display a leaderboard.

**FR4**: The system shall allow users to enable/disable dark mode.

**FR5**: The system shall animate transitions between questions.

**FR6**: The system shall play different sounds for correct and incorrect answers.

**FR7**: The system shall provide haptic feedback for mobile users.

**FR8**: The system shall include a progress bar for countdown timing.

**FR9**: The system shall allow users to share scores on social media.

**FR10**: The system shall allow users to review answers post-quiz.

**3.2 Non-Functional Requirements**

**NFR1**: The user interface shall be simple and intuitive, using Bootstrap for responsiveness.

**NFR2**: The system shall be compatible with modern web browsers and support various screen sizes.

**NFR3**: The quiz website shall load pages efficiently within a reasonable time frame.

**NFR4**: The system shall save user preferences (e.g., dark mode, past scores) in local storage.

1. **Use Cases**

**Use Case 1: Quiz Attempt**

| **Field** | **Details** |
| --- | --- |
| **Use Case Title** | Quiz Attempt |
| **Primary Actor** | User |
| **Description** | A user selects a quiz category and difficulty level, then answers multiple-choice questions within a time limit. After completing all questions, the final score and leaderboard ranking are displayed. |
| **Preconditions** | User has selected a quiz category and difficulty level. |
| **Actions** | 1. User selects a quiz category and difficulty level.  2. System displays multiple-choice questions with a timer.  3. User selects an answer.  4. System provides feedback and moves to the next question.  5. Quiz ends after all questions are answered.  6. System displays the final score and leaderboard ranking. |
| **Alternative Paths** | - User can quit the quiz anytime, but the score will not be recorded.  - If the timer runs out, the system auto-submits the current answer and moves to the next question. |
| **Postconditions** | The final score and leaderboard ranking are displayed. |
| **Exceptions** | - User exits the quiz before completing all questions.  - System error occurs, preventing score calculation. |

**Use Case 2: Leaderboard Access**

| **Field** | **Details** |
| --- | --- |
| **Use Case Title** | Leaderboard Access |
| **Primary Actor** | User |
| **Description** | A user accesses the leaderboard to check top scores and their ranking compared to others. |
| **Preconditions** | User has completed at least one quiz. |
| **Actions** | 1. User navigates to the leaderboard section.  2. System displays the top 5 scores.  3. System highlights the user’s ranking compared to others. |
| **Alternative Paths** | - If no leaderboard data is available, the system informs the user.  - User can refresh the page to update the leaderboard. |
| **Postconditions** | User views leaderboard rankings. |
| **Exceptions** | - No scores available in the leaderboard.  - System fails to fetch leaderboard data. |

**Use Case 3: Dark Mode Toggle**

| **Field** | **Details** |
| --- | --- |
| **Use Case Title** | Dark Mode Toggle |
| **Primary Actor** | User |
| **Description** | A user can switch between light and dark mode, and the system saves their preference for future visits. |
| **Preconditions** | User is on any page of the website. |
| **Actions** | 1. User clicks the dark mode toggle button.  2. System updates UI to dark mode.  3. System saves preference in local storage. |
| **Alternative Paths** | - If the user refreshes the page, the system will restore the previously saved mode. |
| **Postconditions** | UI is updated to dark mode and preference is saved. |
| **Exceptions** | - Dark mode fails to activate due to a script error. |

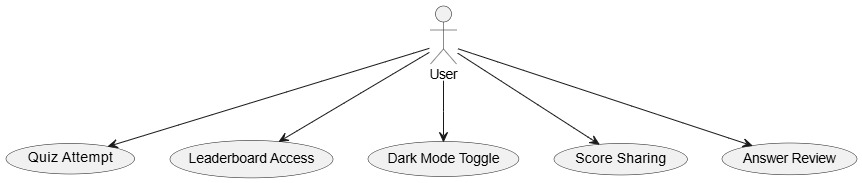
**Use Case 4: Score Sharing**

| **Field** | **Details** |
| --- | --- |
| **Use Case Title** | Score Sharing |
| **Primary Actor** | User |
| **Description** | A user can share their quiz score on social media after completing a quiz. |
| **Preconditions** | User has completed a quiz. |
| **Actions** | 1. User completes the quiz.  2. System provides an option to share the score on social media.  3. User selects the desired platform and confirms sharing. |
| **Alternative Paths** | - User can copy the score link manually if automatic sharing fails.  - User can choose to skip score sharing. |
| **Postconditions** | Score is shared successfully on the selected platform. |
| **Exceptions** | - User cancels the sharing action.  - Internet connectivity issue prevents sharing. |

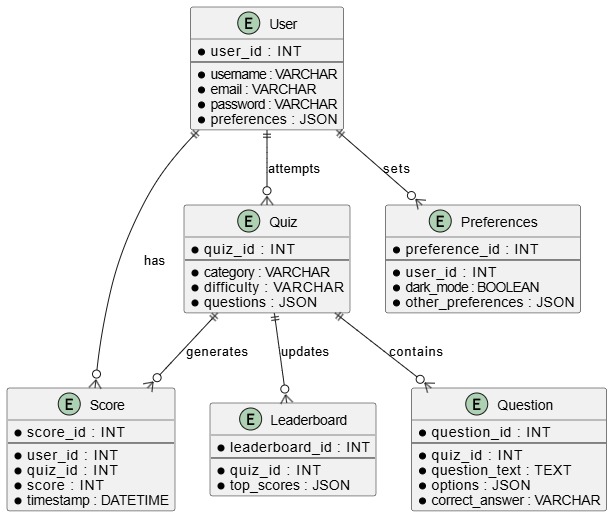
**Use Case 5: Answer Review**

| **Field** | **Details** |
| --- | --- |
| **Use Case Title** | Answer Review |
| **Primary Actor** | User |
| **Description** | After completing a quiz, a user can review their answers and see the correct ones. |
| **Preconditions** | User has completed a quiz. |
| **Actions** | 1. User completes the quiz.  2. System provides an option to review answers.  3. System highlights correct answers and user's selections. |
| **Alternative Paths** | - User can skip the review and proceed directly to the leaderboard. |
| **Postconditions** | User is able to review the correct and incorrect answers. |
| **Exceptions** | - System fails to display answer review due to a data error. |

1. **Use Case Diagram**



1. **ERD Diagram**



1. **Sequence Diagram**

A screenshot of a computer screen

AI-generated content may be incorrect.

1. **File Structure**

/home\_extracted/home/

│── index.html # Main entry page

│── quiz.html # Quiz interface

│── about.html # About page

│── feedback.html # Feedback form

│── Privacy\_Policy.html # Privacy policy

│── styles.css # Global styles

│── feedback.css # Feedback page styles

│── footer.css # Footer styles

│── about.css # About page styles

│── scrept.js # JavaScript logic for quiz

│── images/ # Image assets

│── background/ # Background images

1. **Future Enhancements**

* Backend integration for user authentication and persistent leaderboards.
* Additional question categories and difficulty levels.
* Multiplayer quiz mode.