Srs How well do you know .

Below is a \*\*Software Requirements Specification (SRS)\*\* document for the \*“How Well Do You Know?”\* quiz web application, based on all the details and clarifications provided. This document follows a typical SRS format to ensure comprehensive coverage of the requirements. Feel free to request any modifications or additional sections if needed.

---

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to document the functional and non-functional requirements for the \*“How Well Do You Know?”\* web application. This application aims to provide quizzes across various categories and subcategories, supporting both fun and educational purposes.

## 1.2 Scope

- \*\*Name of the Application:\*\* \*How Well Do You Know?\*

- \*\*Primary Objectives:\*\*

1. Allow users to register, log in, and create or take quizzes.

2. Provide multiple categories (and subcategories) of quizzes, including educational, skill-based, and fun trivia.

3. Maintain a leader board and comprehensive quiz analytics for users.

4. Offer an admin panel with the ability to manage categories, subcategories, and approve user-created quizzes.

## 1.3 Definitions, Acronyms, and Abbreviations

- \*\*Admin:\*\* A user with special privileges to manage the platform, including quizzes, categories, and user-generated content.

- \*\*General User:\*\* A regular user who can create, attempt quizzes, and view analytics.

- \*\*MCQ (Multiple Choice Questions):\*\* A question format where a user selects one correct answer from a set of options.

- \*\*UI:\*\* User Interface.

- \*\*SRS:\*\* Software Requirements Specification.

- \*\*XAMPP:\*\* A local server solution (Apache, MariaDB, PHP, Perl) used for development and testing.

## 1.4 References

- Bootstrap 5 documentation: [https://getbootstrap.com/docs/5.3/](https://getbootstrap.com/docs/5.3/)

- UI reference for MCQ style: [https://bbbootstrap.com/snippets/bootstrap-5-multiple-choice-custom-radio-buttons-46709123](https://bbbootstrap.com/snippets/bootstrap-5-multiple-choice-custom-radio-buttons-46709123)

- Sample multi-choice structure: [https://bootstrapr.io/bs3/multiple-choice.html](https://bootstrapr.io/bs3/multiple-choice.html)

---

# 2. Overall Description

## 2.1 Product Perspective

The \*“How Well Do You Know?”\* application will be a web-based platform designed to run on desktop and mobile browsers. It will be developed using:

- \*\*Frontend:\*\* HTML, CSS, Bootstrap, and Vanilla JavaScript (no jQuery).

- \*\*Backend:\*\* Plain PHP.

- \*\*Database:\*\* MySQL.

- \*\*Version Control:\*\* Git and GitHub.

The application will have two main user roles: Admins and General Users. Both roles can create quizzes (subject to guidelines and admin approval). However, Admins have additional privileges such as managing categories, subcategories, and user-generated quizzes.

## 2.2 Product Functions

1. \*\*User Registration and Login\*\*

- Users can sign up and log in securely to create or attempt quizzes.

2. \*\*Quiz Creation\*\*

- Both Admins and General Users can create MCQ quizzes with up to 4 options per question.

- Quizzes can include images or multimedia.

- Quiz creation includes selecting a category and subcategory from a hierarchical structure.

3. \*\*Quiz Approval\*\*

- Any quiz created by a General User must go through an approval process by Admin before becoming publicly available.

4. \*\*Quiz Attempt\*\*

- Users can take quizzes, view their score, and get instant feedback.

- Timer functionality (1 minute per question) for certain skill assessment quizzes.

5. \*\*Leaderboard\*\*

- Track user rankings globally and by category.

- Monthly reset of the leaderboard and a permanent overall leaderboard.

6. \*\*Quiz Analytics\*\*

- Detailed analytics including time taken, correct/incorrect answers, performance trends, and the option to export as PDF or Excel.

7. \*\*User Profiles\*\*

- Display user details, quiz performance stats, and created quizzes.

8. \*\*Admin Management\*\*

- Admin can manage categories, subcategories, user-generated quizzes, user accounts, and high-level analytics.

## 2.3 User Classes and Characteristics

1. \*\*General User\*\*

- Can sign up, log in, create MCQ quizzes (subject to guidelines and admin approval), attempt quizzes, and view analytics.

- Tech-savvy level: Basic.

2. \*\*Admin\*\*

- Has elevated privileges to manage categories, subcategories, approve or reject user-created quizzes, view or delete problematic quizzes, and view platform analytics.

- Tech-savvy level: Intermediate.

## 2.4 Operating Environment

- \*\*Client-Side:\*\* Modern web browsers on desktop (Windows, macOS, Linux) and mobile (Android, iOS).

- \*\*Server-Side:\*\* XAMPP on Windows 11 (development phase).

- \*\*Deployment:\*\* Initially local (via XAMPP), potentially hosted online on GitHub Pages or a web hosting service with PHP/MySQL support.

## 2.5 Design and Implementation Constraints

- \*\*Plain PHP:\*\* No frameworks like Laravel or CodeIgniter.

- \*\*No jQuery:\*\* Strictly Vanilla JS + Bootstrap for frontend interactions.

- \*\*Bootstrap:\*\* Either served via local files or via CDN.

- \*\*GitHub:\*\* Version control and potential deployment.

- \*\*Security Constraints:\*\*

- User authentication must be secure (hashed passwords).

- Quizzes containing hateful or harmful content (e.g., Islamophobic content) must be rejected by Admin.

## 2.6 Assumptions and Dependencies

- Users have access to stable internet when taking quizzes.

- Admins will actively manage content to ensure no inappropriate or hateful quizzes.

- The local environment runs on XAMPP with MySQL properly configured.

---

# 3. System Features and Requirements

## 3.1 Functional Requirements

Below is a high-level breakdown of the core functional requirements. Each requirement is labeled for reference (FR-1, FR-2, etc.).

### 3.1.1 User Management

- \*\*FR-1: User Registration\*\*

- The system shall allow a new user to register using a unique email and a secure password.

- \*\*FR-2: User Login\*\*

- The system shall allow registered users to log in with their credentials.

- \*\*FR-3: Password Encryption\*\*

- User passwords shall be hashed and stored securely in the database.

- \*\*FR-4: User Profile Management\*\*

- The system shall allow users to view and edit their profile (name, email, profile picture, etc.).

### 3.1.2 Quiz Management

- \*\*FR-5: Quiz Creation by Users\*\*

- The system shall allow registered users to create new quizzes by entering:

1. Title of the quiz.

2. Category -> Subcategory.

3. Each question (text, optional image).

4. Exactly 4 answer options per question.

5. Mark one correct answer for each question.

- \*Constraint:\* 4 options only, no partial marks.

- \*\*FR-6: Quiz Creation Guidelines\*\*

- The system shall display guidelines prohibiting hateful or harmful content.

- The system shall reject quizzes with offensive content.

- \*\*FR-7: Quiz Approval\*\*

- The system shall route quizzes created by a General User to Admin for approval.

- Admin must approve a quiz before it is published and visible to all.

- \*\*FR-8: Quiz Attempt\*\*

- The system shall allow users to attempt any published quiz.

- \*\*FR-9: Scoring\*\*

- The system shall award points for correct answers and apply a penalty for wrong answers.

- No partial scoring is allowed.

### 3.1.3 Categories and Subcategories

- \*\*FR-10: Category Hierarchy\*\*

- The system shall organize quizzes into categories and subcategories (e.g., Education > Math > Algebra).

- \*\*FR-11: Category Management\*\*

- The system shall allow only Admins to create, edit, or delete categories and subcategories.

### 3.1.4 Timer (For Skill Assessment Quizzes)

- \*\*FR-12: Timer Implementation\*\*

- The system shall enforce a 1-minute timer per question in skill assessment quizzes.

- \*\*FR-13: Timer Expiry\*\*

- Once the timer for a question expires, the user’s answer is locked or marked incorrect.

### 3.1.5 Leaderboard

- \*\*FR-14: Leaderboard Display\*\*

- The system shall display leaderboards that rank users based on total points.

- The system shall provide options for global ranking, category-specific ranking, and monthly resets.

### 3.1.6 Analytics

- \*\*FR-15: User Quiz Analytics\*\*

- The system shall provide analytics for each quiz attempt, showing time taken, correct vs. incorrect answers, and improvement trends.

- \*\*FR-16: Export Analytics\*\*

- The system shall allow users to export their performance data as PDF or Excel.

### 3.1.7 Admin Panel

- \*\*FR-17: Quiz Approval/Deletion\*\*

- The admin can approve, reject, or delete quizzes.

- \*\*FR-18: Analytics Dashboard\*\*

- The admin can view platform-wide analytics, including number of quizzes, number of users, popular categories, etc.

## 3.2 Non-Functional Requirements

### 3.2.1 Usability

- \*\*NFR-1: Consistent UI/UX\*\*

- The system shall use a clean and consistent Bootstrap-based layout.

- Refer to [bbbootstrap.com](https://bbbootstrap.com/snippets/bootstrap-5-multiple-choice-custom-radio-buttons-46709123) for MCQ styling.

- \*\*NFR-2: Easy Navigation\*\*

- The system shall provide clear navigation menus for accessing categories, subcategories, user profiles, and admin panel.

### 3.2.2 Reliability

- \*\*NFR-3: Server Uptime\*\*

- The system shall run on a reliable server environment (local XAMPP for development, potential external hosting).

- \*\*NFR-4: Data Backup\*\*

- Database backups should be implemented to protect user data.

### 3.2.3 Performance

- \*\*NFR-5: Page Load\*\*

- Each page shall load within 3 seconds on a standard broadband connection.

- \*\*NFR-6: Scalability\*\*

- The database design should accommodate increasing numbers of quizzes and users.

### 3.2.4 Security

- \*\*NFR-7: Authentication & Authorization\*\*

- The system shall restrict certain pages (admin panel, quiz creation) to authorized users only.

- \*\*NFR-8: Input Validation\*\*

- The system shall validate all user inputs to prevent SQL injection, XSS, and other security vulnerabilities.

### 3.2.5 Maintainability

- \*\*NFR-9: Git Version Control\*\*

- All code shall be managed through GitHub for versioning and collaboration.

- \*\*NFR-10: Code Documentation\*\*

- Inline comments and basic documentation should be provided for all critical modules.

### 3.2.6 Portability

- \*\*NFR-11: Cross-Platform Support\*\*

- The frontend shall be accessible on various operating systems (Windows, macOS, Linux) and mobile platforms.

---

# 4. User Interfaces

Below is a brief overview of the main user interfaces:

1. \*\*Landing Page:\*\*

- A simple homepage with options to \*Sign Up\*, \*Log In\*, \*Browse Quizzes\*, or \*View Leaderboard\*.

2. \*\*Registration/Login Pages:\*\*

- User-friendly forms for new account registration and existing user login.

3. \*\*Dashboard (Post-Login):\*\*

- Shows user’s current stats, top quizzes, and recent quiz attempts.

4. \*\*Quiz Creation Page:\*\*

- A form to select category/subcategory, input question text, up to 4 options, and specify the correct answer.

- A summary of guidelines is shown before submission.

5. \*\*Quiz Attempt Page:\*\*

- Displays quiz questions one by one or all at once (TBD based on final design).

- Enforces the 1-minute timer for skill-based quizzes.

- Submits user’s answers for scoring.

6. \*\*Quiz Result and Analytics Page:\*\*

- Shows user’s score, correct/incorrect breakdown, time taken, and an option to export results as PDF or Excel.

7. \*\*Admin Panel:\*\*

- Contains sections for category management, quiz approval queue, and high-level analytics.

8. \*\*Leaderboard Page:\*\*

- Displays top users globally and per category.

- Indicates monthly resets and allows viewing historical data.

---

# 5. Use Cases (High-Level)

1. \*\*UC-1: Register an Account\*\*

\*\*Primary Actor:\*\* Visitor

\*\*Precondition:\*\* Visitor must have valid email address.

\*\*Main Flow:\*\*

1. Visitor clicks \*Sign Up\*.

2. Visitor provides email, password, name.

3. System validates and creates the user record.

4. System logs user in or redirects to login page.

2. \*\*UC-2: Create a Quiz\*\*

\*\*Primary Actor:\*\* General User or Admin

\*\*Precondition:\*\* User is logged in.

\*\*Main Flow:\*\*

1. User navigates to “Create Quiz” page.

2. User selects category and subcategory.

3. User inputs quiz title and questions with up to 4 options.

4. User marks correct answer, reviews guidelines, and submits.

5. Quiz goes into \*Pending\* state for Admin approval (if User is non-Admin).

3. \*\*UC-3: Approve Quiz\*\*

\*\*Primary Actor:\*\* Admin

\*\*Precondition:\*\* There is at least one pending quiz.

\*\*Main Flow:\*\*

1. Admin opens the “Quiz Approval” section.

2. Admin reviews quiz details, checks for any violations or errors.

3. Admin either approves, rejects, or deletes quiz.

4. Approved quiz becomes visible to all users.

4. \*\*UC-4: Attempt a Quiz\*\*

\*\*Primary Actor:\*\* Any logged-in user

\*\*Precondition:\*\* The quiz is approved and published.

\*\*Main Flow:\*\*

1. User selects a quiz to attempt.

2. For skill-based quizzes, a 1-minute timer starts per question.

3. User submits answers.

4. System calculates score and shows result.

5. \*\*UC-5: View Leaderboard\*\*

\*\*Primary Actor:\*\* Any user or visitor

\*\*Precondition:\*\* Quizzes have been taken and scores recorded.

\*\*Main Flow:\*\*

1. User clicks “Leaderboard.”

2. System displays top users globally and by category.

3. System shows monthly reset countdown or archived scores.

6. \*\*UC-6: View Analytics\*\*

\*\*Primary Actor:\*\* Logged-in user

\*\*Precondition:\*\* User has completed one or more quizzes.

\*\*Main Flow:\*\*

1. User navigates to “Analytics” page.

2. System shows quiz completion history, correct/incorrect answers, time spent, and improvement trends.

3. User can export analytics in PDF or Excel format.

---

# 6. Data Model

A possible relational data schema (high-level; details may change during implementation):

1. \*\*Users\*\*

- user\_id (PK)

- name

- email

- password\_hash

- role (admin or user)

- profile\_picture (optional)

- created\_at

2. \*\*Categories\*\*

- category\_id (PK)

- category\_name

- parent\_category\_id (nullable for top-level categories)

3. \*\*Quizzes\*\*

- quiz\_id (PK)

- title

- created\_by (FK -> Users)

- category\_id (FK -> Categories)

- subcategory\_id (FK -> Categories)

- status (pending, approved, rejected)

- created\_at

- approved\_at (nullable if pending)

- approved\_by (FK -> Users, nullable)

4. \*\*Questions\*\*

- question\_id (PK)

- quiz\_id (FK -> Quizzes)

- question\_text

- question\_image (nullable)

5. \*\*Options\*\*

- option\_id (PK)

- question\_id (FK -> Questions)

- option\_text

- is\_correct (boolean)

6. \*\*User\_Quiz\_Attempts\*\*

- attempt\_id (PK)

- user\_id (FK -> Users)

- quiz\_id (FK -> Quizzes)

- score

- start\_time

- end\_time

7. \*\*User\_Answers\*\*

- user\_answers\_id (PK)

- attempt\_id (FK -> User\_Quiz\_Attempts)

- question\_id (FK -> Questions)

- chosen\_option\_id (FK -> Options)

- is\_correct (boolean)

---

# 7. Additional Requirements and Guidelines

1. \*\*Islamophobia or Hate Speech Filtering\*\*

- Quiz content must be moderated to disallow hateful or harmful content.

2. \*\*Hosting and Deployment\*\*

- Initially on local XAMPP for development.

- Potential hosting on GitHub or a standard PHP-supporting web host.

3. \*\*Documentation\*\*

- Basic API documentation for backend endpoints (for future expansions).

- A changelog once the app is deployed and updated.

4. \*\*Future Features (Not in Current Scope)\*\*

- Multi-language support.

- Automated test suites (PHPUnit).

- CI/CD pipelines.

- Staging environment.

---

# 8. Acceptance Criteria

1. \*\*User Registration and Login\*\*

- A new user can create an account and log in successfully.

- Passwords are securely hashed.

2. \*\*Quiz Creation & Approval\*\*

- A logged-in user can create a quiz with 4 options per question.

- Admin must approve the quiz before it’s accessible to other users.

3. \*\*Taking Quizzes\*\*

- Users can see a question’s text and multiple choice options.

- For skill-based quizzes, a timer ensures each question’s answer is locked after 1 minute.

4. \*\*Leaderboard & Analytics\*\*

- Leaderboard displays top scorers globally, by category, and resets monthly.

- User can view their personal performance history and export analytics.

5. \*\*Admin Panel\*\*

- Admin can create, edit, or remove categories and subcategories.

- Admin can approve or delete any quiz.

- Admin can see overall platform analytics (e.g., total quizzes, user count).

6. \*\*Security and Content Guidelines\*\*

- Offensive content is flagged and can be removed by Admin.

- SQL injection, XSS, and similar attacks are mitigated via input validation.

---

# 9. Conclusion

This SRS provides a clear blueprint for developing the \*“How Well Do You Know?”\* quiz web application. It covers both functional and non-functional requirements, user roles, data modeling, and acceptance criteria. The next steps involve:

1. \*\*UI/UX Wireframes (Optional):\*\* To define the look-and-feel based on Bootstrap components.

2. \*\*Database Schema Finalization:\*\* Setting up MySQL structures according to the proposed schema.

3. \*\*Development & Iteration:\*\* Implementing the features in milestones, continuously testing.

4. \*\*Deployment & Maintenance:\*\* Hosting the platform online and maintaining future changes via Git/GitHub version control.

With this specification, we have a comprehensive understanding of how the system should function, its constraints, and the user needs it aims to meet. If you have any further questions or need clarifications, feel free to let me know before we proceed with the development phase.

---

\*\*End of SRS\*\*