# **Functional Requirements Document for "Shifter"**

# **1. Overview**

This document defines the functional requirements for the "Shifter" application. The purpose is to ensure a shared understanding of the key features and functionality of the app, facilitating smooth development and implementation.

# **2. Functional Requirements**

**2.1 User Interface**

* **Requirement**: The application should have an intuitive and user-friendly interface.
  + Support for both light and dark modes.
  + Clear navigation and easy access to key features such as daily questions, dashboard, and settings.
  + Consistent design across Android and iOS platforms.

**2.2 User Authentication and Profile Management**

* **Requirement**: Users must be able to register, log in, and manage their profiles securely.
  + **Registration**: Via email, Google, or social media.
  + **Login**: Support for email/password and social login.
  + **Profile Management**:
    - View and edit personal information (e.g., name, avatar, preferences).
    - Set and update topics of interest.

**2.3 Question Management**

* **Requirement**: The system should handle questions and answers dynamically.
  + **Question Types**:
    - Multiple-choice questions (MCQs).
    - Open-ended questions.
  + **Features**:
    - Randomized question delivery.
    - Ability to save questions for review later.
    - Display correct answers and explanations post-quiz.
    - Maintain a history of attempted questions and performance.

**2.4 Dashboard**

* **Requirement**: The dashboard should provide users with insights into their progress.
  + **Features**:
    - Progress tracking: Graphs and stats showing daily, weekly, and monthly performance.
    - Leaderboard: Display rankings among users globally or within groups.
    - Badges: Visual rewards for milestones (e.g., streaks, high scores, consistency).

**2.5 Client Support**

* **Requirement**: Users should have access to a support system for resolving issues.
  + In-app support via chat or email.
  + A FAQ section for common questions and troubleshooting.
  + Feedback system for suggestions or reporting bugs.

**2.6 Security**

* **Requirement**: The application must prioritize user data security.
  + **Authentication**: Implement OAuth2 or JWT for secure user sessions.
  + **Data Protection**: Encrypt sensitive data (e.g., passwords, personal information).
  + **Database Security**: Use secure connections (e.g., SSL/TLS).
  + Regular updates to address vulnerabilities.

**2.7 Performance**

* **Requirement**: The application must deliver a fast and seamless experience.
  + Load times for screens should be under 2 seconds.
  + Questions should load instantly.
  + Support offline access for preloaded content.

**2.8 Notifications**

* **Requirement**: Send notifications to encourage user engagement.
  + Daily reminders for quizzes.
  + Alerts for streaks and achievements.
  + Updates for new features or important announcements.

**2.9 Gamification**

* **Requirement**: Add gamification elements to enhance user engagement.
  + Badges and achievements for milestones.
  + Streaks for daily activity.
  + Leaderboards for friendly competition among users.

**2.10 Monetization**

* **Requirement**: Prepare the platform for future monetization.
  + **Premium Features**:
    - Ad-free experience.
    - Exclusive question sets.
    - Advanced analytics for progress tracking.
  + Subscription plans and one-time purchases.

**2.11 Analytics**

* **Requirement**: Provide insights into app usage and user behavior.
  + **For Users**:
    - Detailed reports on individual performance.
  + **For Admins**:
    - Data on user engagement, popular topics, and app performance.

**2.12 Accessibility**

* **Requirement**: Ensure the app is accessible to all users.
  + Support for screen readers.
  + Adjustable text sizes and themes.
  + Multi-language support for a diverse user base.

**2.13 Scalability**

* **Requirement**: The system must be designed to scale with user growth.
  + Efficient database queries to handle increasing volumes of questions and users.
  + Load balancing for server-side operations.

# **3. Non-Functional Requirements**

* **Reliability**: The app should have 99.9% uptime.
* **Maintainability**: Modular code structure to facilitate updates and bug fixes.
* **Portability**: The app must function smoothly across Android and iOS devices.