**Web Application Requirements Document**

**1. Introduction**

**1.1. Project Overview**

Develop a web-based platform for creating, sharing, and playing interactive quizzes and games.

**1.2. Scope of the Application**

* Define the overall scope of the web application and what it will or will not include.

**2. Functional Requirements**

1. **User Management**

* **User Registration and Login: Allow users to register and log in using email or social media accounts.**
* **User Profiles: Enable users to create and manage their profiles.**

1. **Quiz Creation**

* **Quiz Builder: Provide an intuitive interface for creating quizzes with multiple question types (multiple choice, true/false, short answer).**
* **Question Bank: Allow users to save and reuse questions.**
* **Media Support: Enable adding images, videos, and audio to questions.**

1. **Quiz Management**

* **Quiz Library: Allow users to save, edit, and delete their quizzes.**
* **Privacy Settings: Enable users to set quizzes as public or private.**
* **Tags and Categories: Allow users to categorize and tag quizzes for easy searching.**

1. **Game Play**

* **Live Games: Support real-time quiz sessions where participants can join using a game code.**
* **Self-Paced Games: Allow users to play quizzes at their own pace.**
* **Leaderboard: Display scores and rankings during and after the game.**

1. **Reporting and Analytics**

* **Performance Reports: Provide detailed reports on quiz performance for both creators and participants.**
* **Data Export: Allow users to export quiz results in various formats (CSV, PDF).**

1. **Social Features**

* **Comments and Ratings: Enable users to comment on and rate quizzes.**
* **Sharing: Allow users to share quizzes via social media and email.**

1. **Administrative Features**

* **User Management: Admins can manage users, including banning and unbanning accounts.**
* **Content Moderation: Admins can review and remove inappropriate content.**

1. **Technical Requirements**

* **Responsive Design: Ensure the platform is accessible on both desktop and mobile devices.**
* **Scalability: Design the system to handle a large number of concurrent users.**
* **Security: Implement robust security measures to protect user data and prevent cheating.**

**3. Non-Functional Requirements**

* **Performance: Ensure fast load times and smooth interactions.**
* **Usability: Provide an intuitive and user-friendly interface.**
* **Accessibility: Ensure the platform is accessible to users with disabilities.**

**4. Technical Requirements**

**4.1. Frontend Requirements**

* Technologies and frameworks (e.g., React, Vue.js)

**4.2. Backend Requirements**

* Server-side technologies and frameworks (e.g., Node.js, Django, Ruby on Rails)
* API requirements (REST, GraphQL)
* Database requirements (e.g., PostgreSQL, MongoDB)

**4.3. Third-Party Services and Integrations**

* List third-party services that the web app must integrate with (e.g., payment gateways, social login APIs).

**4.4. Hosting and Infrastructure**

* Preferred hosting platforms (e.g., AWS, Azure)
* CDN usage and caching strategies

**5. User Interface and User Experience Requirements**

**5.1. UI/UX Guidelines**

* Define any design constraints, branding, or style guides to be followed.
  + E.g., *The application should follow Material Design guidelines.*

**5.2. Wireframes/Mockups**

* Provide links to or descriptions of the wireframes, mockups, or design prototypes.

**5.3. Navigation**

* Define the site’s navigation structure and user flow.

**6. Data and Database Requirements**

**6.1. Data Storage**

* Define what kind of data the application will store and manage (e.g., user data, transaction history).

**6.2. Data Relationships**

* Define the relationships between different data entities (e.g., user profiles and orders).

**6.3. Backup and Recovery**

* Define the backup strategy and disaster recovery plan.
  + E.g., *The system must create daily backups and store them in a secure location.*

**7. API Requirements**

**7.1. Internal APIs**

* Define the requirements for internal APIs used within the system.

**7.2. External APIs**

* List the external APIs that the system must interact with and any security protocols involved.
  + E.g., *The application must interact with the Google Maps API to display location data.*

**8. Deployment and Hosting Requirements**

**8.1. Hosting Environment**

* Define the hosting environment and expected server resources (e.g., cloud hosting, virtual machines, or dedicated servers).

**8.2. CI/CD Pipeline**

* Specify the tools and processes for Continuous Integration and Continuous Deployment (CI/CD).

**8.3. Deployment Process**

* Describe how the application will be deployed (manual or automated, environments like staging and production).

**9. Compliance and Legal Requirements**

**9.1. Legal Requirements**

* Any legal obligations, such as intellectual property considerations, data ownership, or licensing.

**9.2. Data Protection**

* Specific requirements for protecting user data, including data encryption and user consent processes.

**9.3. Privacy Policies**

* Outline the privacy policies that must be implemented and shown to users (e.g., GDPR compliance).

**10. Testing and Quality Assurance**

**10.1. Testing Environment**

* Define the environments that will be used for testing (e.g., dev, staging, production).

**10.2. Test Types**

* Unit testing
* Integration testing
* UI/UX testing
* Load and performance testing
* Security testing (e.g., penetration testing)

**10.3. Acceptance Criteria**

* Define the criteria for successfully passing each stage of testing.

**11. Risks and Assumptions**

**11.1. Risks**

* Potential risks that might affect the project (e.g., third-party service outages, scaling issues).

**11.2. Assumptions**

* Assumptions made during the requirements gathering phase (e.g., availability of necessary tools, reasonable user growth).

**12. Timeline and Milestones**

**12.1. Development Phases**

* Break down development into phases with start and end dates.

**12.2. Key Milestones**

* Define major milestones, such as:
  + Feature completion
  + Alpha release
  + Beta release
  + Launch

**13. Conclusion**

**13.1. Summary**

* Provide a final summary of the key takeaways from the requirements document.

This template offers a comprehensive structure to outline all the requirements needed to successfully develop and deploy a web application. You can customize and add to these sections as necessary to fit the specific needs of your project.