System Requirement Specification Document

1. Introduction

This document outlines the system requirements for the development of the user interface (UI) of the AwesomeQA platform's login page. The login page is a crucial component of the platform, serving as the gateway for users to access their accounts and utilize the various functionalities offered by AwesomeQA.

2. Purpose

The purpose of this document is to define the functional and non-functional requirements of the UI for the AwesomeQA login page. It aims to provide clear guidance for developers, designers, and stakeholders involved in the development process.

3. Scope

This specification covers the UI requirements for the login page of the AwesomeQA platform. It includes both frontend and backend requirements necessary for the page to function efficiently and securely.

4. Functional Requirements

4.1. User Authentication:

- Users should be able to log in using their registered email address and password.

- Provide an option for users to recover/reset their password in case they forget it.

4.2. Input Validation:

- Validate user inputs for email address and password fields to ensure they meet the required format and length criteria.

- Display appropriate error messages if validation fails.

4.3. Security:

- Implement secure transmission of login credentials over HTTPS.

- Implement measures to prevent brute-force attacks, such as account lockout after a certain number of failed login attempts.

- Store user passwords securely using hashing algorithms.

4.4. User Experience:

- Design the login page to be intuitive and user-friendly.

- Provide feedback to users upon successful or unsuccessful login attempts.

4.5. Remember Me Option:

- Include an option for users to stay logged in by enabling a "Remember Me" feature.

4.6. Accessibility:

- Ensure that the login page is accessible to users with disabilities by following WCAG guidelines.

5. Non-Functional Requirements

5.1. Performance:

- The login page should load quickly to provide a seamless user experience.

- Handle concurrent user logins efficiently without performance degradation.

5.2. Compatibility:

- Ensure cross-browser compatibility for popular web browsers such as Chrome, Firefox, Safari, and Edge.

- Ensure compatibility with different devices, including desktops, laptops, tablets, and smartphones.

5.3. Scalability:

- Design the login system to be scalable to accommodate an increasing number of users over time.

5.4. Security:

- Implement industry-standard security protocols to protect user data from unauthorized access or breaches.

- Regularly update security measures to address emerging threats and vulnerabilities.

5.5. Reliability:

- Ensure high availability of the login page to minimize downtime.

- Implement backup and recovery mechanisms to prevent data loss.

6. Constraints

6.1. The UI design should align with the existing branding and visual identity of the AwesomeQA platform.

6.2. The development should adhere to the technology stack and frameworks already in use by the AwesomeQA development team.

7. Assumptions

7.1. Users have already registered for an account on the AwesomeQA platform.

7.2. The backend infrastructure necessary for user authentication and account management is already in place.

8. Dependencies

8.1. The availability of backend APIs for user authentication and password management.

8.2. Timely collaboration and communication between frontend and backend development teams.

9. Revision History

Version	Description	Date	Author
1.0	Initial version	2024-05-15	Michelle

This System Requirement Specification document provides a comprehensive overview of the requirements for developing the UI of the AwesomeQA login page. It serves as a guide for the development team to ensure the successful implementation of the login functionality while meeting the needs and expectations of users and stakeholders.