Software Requirements Specification (SRS)
GitHub Simple Sign-In Page

1. Introduction

1.1 Purpose
This SRS describes the functionality and constraints for a simple GitHub sign■in page (i.e., https://github.com/login). It targets developers, QA teams, and stakeholders, providing clear guidance on behavior, validation, and security for a basic authentication interface.

1.2 Scope
Covers user inputs, validation, authentication workflows, error handling, passkey integration, and navigational elements. Does not delve into social login, two■factor flows beyond initial credential prompts, or backend API design.

1.3 Definitions
- User: Someone attempting to log in using GitHub credentials.
- Passkey: A FIDO/WebAuthn-based authentication alternative to passwords.
- 2FA: Two■Factor Authentication, may be part of subsequent flows not in scope here.

2. Overall Description

2.1 Product Perspective
The sign■in page is a standalone interface in the GitHub web app ecosystem.

2.2 User Needs
- Quick access: username/email + password or passkey.
- Clear error messaging.
- Recovery via “Forgot password?”.
- Call-to-action for new users.
- Compliance with security best practices.

2.3 Assumptions & Dependencies
- Backend authentication service handles credential validation.
- HTTPS is mandatory.
- JavaScript required for passkey support.

3. Functional Requirements

The system must support:
- FR1: Display username/email and password fields.
- FR2: Sign in button.
- FR3: “Forgot password?” link.
- FR4: “Create an account” link.
- FR5: Sign in with a passkey option.
- FR6: Error messages for invalid login.
- FR7: Redirect on success.
- FR8: JS-enabled passkey sign-in.

4. Non-Functional Requirements

- Performance: Page load < 1s on broadband.
- Security: HTTPS, sanitized inputs, masked password.
- Usability: Accessible labels, responsive layout.
- Accessibility: WCAG AA compliance.
- Localization: Multi-language text support.

5. External Interface Requirements

5.1 UI Layout
- Username/email and password fields.
- Sign in button.
- “Forgot password?” and “Create account” links.
- Passkey option.

5.2 API Interaction
- Credential validation endpoint.
- Responses: 200 OK, 401 Unauthorized, 429 Too Many Requests, 500 Internal Error.

6. Validation Criteria

- Load page and confirm elements.
- Valid credentials → successful login.
- Invalid credentials → error message.
- Passkey available → works correctly.
- “Forgot password?” and “Create account” work.
- Accessibility checks (screen reader, keyboard).
- Cross-device rendering.

7. Future Enhancements

- Social login (OAuth).
- 2FA flows.
- Device recognition prompts.
- Enterprise SSO.