**Product Requirements Document (PRD)**

**Project: GitHub Repository Finder**

**Version: 1.0**

**Date: 02 july 2023**

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Product Overview

2. User Requirements

2.1 User Goals

2.2 User Stories

3. Functional Requirements

3.1 Search Functionality

3.2 Display of Repository List

4. Non-Functional Requirements

4.1 Usability

4.2 Performance

4.3 Security

5. Technical Requirements

5.1 Technology Stack

5.2 GitHub API Integration

6. Constraints

7. Assumptions

8. Risks and Mitigation Strategies

9. Release Plan

10. User Acceptance Criteria

11. Appendix

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the requirements for the development of the GitHub Repository Finder. It outlines the functionalities and features that the application should provide to users.

**1.2 Product Overview**

The GitHub Repository Finder is a web-based application that allows users to search and retrieve GitHub repositories based on a specific username. By entering a username, users can access a list of repositories associated with that username, providing a convenient way to explore and discover relevant repositories.

**2. User Requirements**

**2.1 User Goals**

- Users should be able to search for GitHub repositories by entering a username.

- Users should be able to view a list of repositories associated with the entered username.

**2.2 User Stories**

- As a user, I want to be able to enter a username in a text field to search for repositories.

- As a user, I want to see a list of repositories associated with the entered username.

- As a user, I want to view repository details such as name, description, and number of stars.

- As a user, I want the list of repositories to be displayed in a user-friendly and intuitive manner.

**3. Functional Requirements**

**3.1 Search Functionality**

- The application should provide a text input field where users can enter a GitHub username.

- Users should be able to initiate a search by clicking a "Search" button or pressing Enter.

- The application should handle empty or invalid username inputs gracefully, providing appropriate error messages or prompts.

**3.2 Display of Repository List**

- The application should retrieve repository data from the GitHub API based on the entered username.

- The list of repositories should be displayed in a clear and organized manner, showing relevant information such as repository name, description, and number of stars.

- Each repository in the list should be clickable, allowing users to navigate to the repository's GitHub page for further details.

**4. Non-Functional Requirements**

**4.1 Usability**

- The user interface should be intuitive and user-friendly, enabling users to easily understand and navigate the application.

- The application should provide clear feedback to users during search and loading processes.

- The user interface should be responsive and optimized for different screen sizes and devices.

**4.2 Performance**

- The application should retrieve repository data efficiently and display the results in a timely manner.

- The search process should have acceptable response times, even for users with a large number of repositories.

**4.3 Security**

- The application should handle user inputs securely and protect sensitive data.

- The integration with the GitHub API should adhere to authentication and authorization mechanisms to ensure data privacy and security.

**5. Technical Requirements**

**5.1 Technology Stack**

- Front-end: HTML, CSS, JavaScript

- GitHub API for repository data retrieval

**5.2 GitHub API Integration**

- The application should integrate with the GitHub API to retrieve repository data based on the entered username.

- The integration should follow the API's guidelines and best practices for authentication and data retrieval.

**6. Constraints**

- The application will be limited to retrieving and displaying public repositories associated with the entered username.

**7. Assumptions**

- Users have a basic understanding of GitHub and the concept of repositories.

- Users have a valid GitHub account if they want to access private repositories.

**8. Risks and Mitigation Strategies**

- Risk: Dependency on the GitHub API for repository data.

Mitigation Strategy: Monitor the GitHub API status and have fallback mechanisms in place to handle API outages or limitations.

**9. Release Plan**

- Initial Release: Basic search functionality and repository list display.

- Future Releases: Enhancements such as pagination, sorting options, and additional repository details.

**10. User Acceptance Criteria**

- Users can enter a username and initiate a search for repositories.

- Users receive a list of repositories associated with the entered username.

- The list of repositories is displayed in an organized and intuitive manner, showing relevant details.

- Clicking on a repository in the list navigates the user to the respective GitHub page.

**11. Appendix**

Include any additional diagrams, mockups, or supporting materials relevant to the PRD.

Please note that this PRD template serves as a starting point. Feel free to modify and add more details specific to your project's requirements.