Software Requirements Specification (SRS)

Name of the application : **Thiraiarangam**

Index

5.5 User Load
5.6 Security Constraints
5.7 Functional Constraints
5.8 Performance Constraints
5.9 Design Constraints

Deliverables
6.1 Wireframes
6.2 Front-End and Back-End Code

6.3 Functional and Integrated Website
6.4 Test Results

Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

- This project aims to develop a social networking platform for users to log, rate, and review movies/tasks.

1.2 Scope

- This Website will allow the user to manage the watchlist,write reviews and interact socially through comments and ratings.

1.3 Audience

- This document (Software Requirement Specification) is made for developers, testers, project managers, users, leadership teams, sales, and marketing.

---

2. Functional Requirements

2.1 User Authentication

- Mentioned features like sign-up, login, logout, and password reset.

2.2 Search Functionality

- Search functionality supports for the Collection, company , keyword, Movie, Multi, Person, TV

2.3 Item/Task Management

- Explain how users can add, edit, or delete items/tasks (like movies).

2.4 Reviews and Ratings

- Users need to be logged in to their account and they can write the review, edit or delete.Rating will be given out of five starts. They can rate the movie out of 5.

2.5 Watchlists/Task Lists

- Users need to log in to their respective account and by searching the movie they can find the movie name and its movie details. They will be provided with the button named as the ‘Add to Watchlist’ by pressing that they can add them to the watch later or watch list.Other than Watchlist they are allowed to make the new list and they can also add the movie to that list. Eg; Users can create the new Motivational Movie List and they can add the movies in that list like Steve Jobs, Social Network etc…

2.6 Social Features

- In this platform users are allowed to share their thoughts and idea among the Society. Like other social media they can follow , comment and view the activities of others.

2.7 Recommendations

- Based on their top searches they made in the website Recommendation will be shown. This will increase the chance of getting the good movies to the right people.

---

3. Non-Functional Requirements

3.1 Usability

- The website must be easy to navigate and responsive on all devices.

3.2 Performance

- Search should be faster and loading page should be there to make it more interactive

3.3 Security

- The application will implement basic security measures to protect user credentials and ensure secure authentication. All user passwords will be hashed using the **bcrypt** algorithm before storage in the database, preventing unauthorized access even if the database is compromised. Additional security features may be added in future iterations as needed

3.4 Scalability

- The website will be hosted on cloud platforms like **Vercel**,which provide automatic scaling based on traffic. The database will be optimized with **indexes** and **pagination** to improve performance as the data grows. Basic **caching** for static resources will be implemented, and **Redis** may be used for frequently accessed data if needed. The system is designed to be easily scalable, with the ability to upgrade hosting or database resources as traffic increases.

3.5 Maintainability

- The code will be clean, clear, and well-structured, with thorough documentation to ensure that new developers can easily understand and contribute to the project.

---

4. System Architecture

- How the system will be built:
    - **Front-End:** The front-end will be developed using HTML for structuring the content, CSS for styling, and JavaScript for interactivity. We will use Next.js to build dynamic pages.
    - **Back-End:** The back-end will be developed using **Python** with the **Flask** framework, which is lightweight and flexible, making it ideal for small to medium-sized applications. The back-end will handle user authentication, data processing, and communication with the database. The database will be **PostgreSQL** for structured data storage, supporting efficient querying and scalability.
    - **Hosting:** The application will be hosted on **Vercel**, which provides seamless deployment for front-end projects. Vercel also supports serverless functions for the back-end, which will allow the application to scale efficiently with minimal maintenance

---

5. Constraints

1. Time Constraints:

- The project must be completed within a week **(7 Days)** limiting the scope of features and complexity that can be implemented.

2. Budget Constraints:

- The budget for hosting and services is limited, which may restrict the use of advanced tools or high-cost cloud services. The application will be hosted on **Vercel**

3. Scalability Constraints:

- Due to the project's limited scope, scalability will be considered in a basic form. Advanced scaling techniques such as sharding, microservices, or complex load balancing will not be implemented at this stage.

4. Technology Constraints:

- The project will be developed using specific technologies, such as **HTML**, **CSS**, **JavaScript**, **Next.js**, **Python (Flask)**, and **PostgreSQL**. Other technologies (e.g., advanced frameworks or databases) may not be considered due to time limitations or complexity.

5. User Load:

- The application is designed to handle a limited number of concurrent users, as it is primarily intended for use as a task for club entry. High user traffic or heavy data loads will not be a focus for this project.

6. Security Constraints:

- Basic security features such as **password encryption** and **session management** will be implemented, but advanced security measures (e.g., SSL certificates, complex encryption methods) may not be fully implemented due to time and resource limitations.

7. Functional Constraints:

- Not all advanced features (e.g., sophisticated recommendation algorithms, complex social networking features) will be included due to the limited scope of the project. The primary goal is to implement core features like user authentication, task addition, and basic reviews.

8. Performance Constraints:

- The website will be optimized for performance, but due to the limited resources and time, performance optimizations like load balancing, advanced caching strategies, or microservices will not be implemented.

9. Design Constraints:

- The user interface will follow basic design principles but may not include advanced features such as complex animations, accessibility features, or cross-platform design beyond responsive web design.

---

6. Deliverables

The outputs expected from the project:

1. Wireframes
2. Front-end and back-end code
3. A functional and integrated website
4. Test results

Contact Information

Developer:
Name: Sivabharathi V S
Email: vssivabharathi@gmail.com
Phone: +91 6383116892