**Software Requirements Specification (SRS)**

**Project Title:** Social Media Dashboard

**Prepared by:** Ankit Raj

**Date:** 19/01/2025

**Table of Contents**

1. **Introduction**
   1. 1.1 Purpose
   2. 1.2 Scope
   3. 1.3 Definitions, Acronyms, and Abbreviations
   4. 1.4 References
2. **Overall Description**
   1. 2.1 Product Perspective
   2. 2.2 Product Features
   3. 2.3 User Classes and Characteristics
   4. 2.4 Operating Environment
   5. 2.5 Assumptions and Dependencies
3. **Functional Requirements**
   1. 3.1 User Authentication
   2. 3.2 Post Management
   3. 3.3 User Interaction
   4. 3.4 Analytics Dashboard
   5. 3.5 Notifications
   6. 3.6 Search and Filtering
4. **Non-Functional Requirements**
   1. 4.1 Performance Requirements
   2. 4.2 Security Requirements
   3. 4.3 Usability Requirements
   4. 4.4 Maintainability Requirements
5. **System Design Constraints**
6. **External Interface Requirements**
   1. 6.1 User Interfaces
   2. 6.2 APIs
   3. 6.3 Hardware Interfaces
   4. 6.4 Software Interfaces
7. **Appendices**
   1. 7.1 Future Enhancements
   2. 7.2 Glossary
   3. 7.3 References

**1. Introduction**

**1.1 Purpose** The purpose of this document is to outline the functional, non-functional, and technical requirements for the development of a Social Media Dashboard. The application will enable users to manage, view, and interact with social media content seamlessly.

**1.2 Scope** The Social Media Dashboard is a web application designed to facilitate users in creating and managing posts, engaging with other users, and analyzing user engagement through an interactive interface. The system aims to provide:

* Post creation, likes, comments, and sharing features.
* Real-time notifications and analytics.
* A responsive user interface for seamless interaction.

**1.3 Definitions, Acronyms, and Abbreviations**

* **UI**: User Interface
* **API**: Application Programming Interface
* **CRUD**: Create, Read, Update, Delete
* **JSON**: JavaScript Object Notation
* **SRS**: Software Requirements Specification

**1.4 References**

* React Documentation: <https://reactjs.org/docs/>
* Material-UI: <https://mui.com/>
* Firebase Documentation: <https://firebase.google.com/docs>

**2. Overall Description**

**2.1 Product Perspective** The Social Media Dashboard will act as an intermediary between users and social media functionalities. It is built as a single-page application (SPA) using React. The application will integrate with back-end services for authentication, database management, and API calls.

**2.2 Product Features**

* User Authentication (Sign up, login, and logout)
* User Profile Management
* Post creation, editing, and deletion
* Real-time notifications
* Comments and likes on posts
* Analytics dashboard for tracking engagement metrics

**2.3 User Classes and Characteristics**

* **Regular Users**: Individuals who will create and interact with posts.
* **Admin Users**: Manage user-generated content, moderate posts, and review analytics.

**2.4 Operating Environment**

* **Frontend**: ReactJS with Material-UI for styling.
* **Backend**: Node.js with Express framework.
* **Database**: MongoDB.
* **Hosting**: Deployed on platforms like Vercel/Netlify (frontend) and AWS/Heroku (backend).
* **Supported Browsers**: Latest versions of Chrome, Firefox, and Safari.

**2.5 Assumptions and Dependencies**

* Users will have internet access to use the application.
* All third-party APIs, such as Firebase, will be functional and integrated seamlessly.

**3. Functional Requirements**

**3.1 User Authentication**

* Users can register with an email and password.
* Users can log in and log out securely.
* Password reset functionality.

**3.2 Post Management**

* Users can create, edit, delete, and view posts.
* Posts can include text, images, or videos.

**3.3 User Interaction**

* Users can like, comment, and share posts.
* Users can follow/unfollow other users.

**3.4 Analytics Dashboard**

* Display post engagement metrics such as likes, comments, and shares.
* Show follower growth over time.

**3.5 Notifications**

* Real-time alerts for likes, comments, and follows.

**3.6 Search and Filtering**

* Users can search for other users or posts.
* Posts can be filtered based on relevance, date, or popularity.

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* The system should handle up to 10,000 concurrent users.
* Response time for API calls should not exceed 2 seconds.

**4.2 Security Requirements**

* Data encryption for sensitive user information.
* Secure user authentication with JWT.
* Protection against common vulnerabilities like SQL injection and XSS.

**4.3 Usability Requirements**

* The UI should be intuitive and accessible.
* Support for both desktop and mobile devices (responsive design).

**4.4 Maintainability Requirements**

* The codebase should follow industry-standard practices and be well-documented.
* Modular architecture for easy scalability.

**5. System Design Constraints**

* The application must use ReactJS for the frontend and Node.js for the backend.
* The database should be MongoDB, hosted on a cloud platform.
* Use Firebase for real-time notifications.

**6. External Interface Requirements**

**6.1 User Interfaces**

* Dashboard with an overview of analytics.
* Post creation interface with file upload capability.
* Profile page for user information.

**6.2 APIs**

* Third-party API integration for notifications (Firebase).
* RESTful APIs for CRUD operations on posts and user data.

**6.3 Hardware Interfaces**

* None; this is a web-based application.

**6.4 Software Interfaces**

* ReactJS frontend communicates with Node.js backend via RESTful APIs.
* MongoDB as the primary database.

**7. Appendices**

**7.1 Future Enhancements**

* Integration with external social media platforms.
* Dark mode and theme customization.
* Advanced analytics with AI-driven insights.

**7.2 Glossary**

* **SPA**: Single Page Application
* **JWT**: JSON Web Token

**7.3 References**

* [ReactJS Documentation](https://reactjs.org/docs/)
* [Node.js Documentation](https://nodejs.org/en/docs/)
* [MongoDB Documentation](https://www.mongodb.com/docs/)

End of Document