LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

SRS DOCUMENT OF KEEPER APPP

SUBMITTED BY:

Name- Abhishek Kumar Sah

Roll no- 67

Registration no- 12210288

Section- K21DA

Github Link-
https://github.com/updatedabhi/keeper-app

Submitted to: Ashish Kumar

1. Introduction

The Keeper app is a web-based note-taking application developed using the React JavaScript library. It serves as a digital platform for users to capture, organize, and manage their thoughts, ideas, and tasks. With Keeper, users can create, view, and delete notes seamlessly within a modern and intuitive user interface. By leveraging the power of React, Keeper delivers a responsive and dynamic user experience, catering to the needs of individuals seeking a reliable tool for note-taking and organization.

2. Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive outline of the functional and non-functional requirements of the Keeper app. This document serves as a blueprint for the development team, guiding them in the implementation of features, interactions, and system constraints. Additionally, the SRS document serves as a communication tool among stakeholders, ensuring clarity and alignment regarding the app's objectives and capabilities. By detailing the requirements in a structured manner, this document facilitates effective collaboration, development, and testing processes, ultimately leading to the successful delivery of the Keeper app.

3. Scope

The scope of the Keeper app encompasses a range of features and components aimed at enabling users to create, view, and manage notes efficiently. From note creation to deletion, the app provides a seamless user experience within a well-defined interface. The inclusion of header and footer sections enhances the app's usability and branding, while the note-taking area serves as the core component for interacting with notes. By defining the app's scope, stakeholders gain clarity on the intended functionalities and user interactions, guiding development efforts toward meeting user expectations effectively.

4. Functional Requirements

4.1. Note Creation

Requirement Explanation: The ability for users to create new notes is essential for capturing information effectively.

Description Elaboration: Users are provided with input fields for entering both the title and content of the note, ensuring flexibility and customization.

Acceptance Criteria Detailing: Upon completing the note creation process by clicking the "Add" button, the newly created note is seamlessly integrated into the list of existing notes, ensuring a smooth user experience.

4.2. Note Deletion

Requirement Explanation: The feature to delete existing notes empowers users to manage their notes efficiently, removing irrelevant or outdated information.

Description Elaboration: Each note displayed in the app interface is equipped with a dedicated "DELETE" button, enabling users to initiate the deletion process conveniently.

Acceptance Criteria Detailing: Upon clicking the "DELETE" button associated with a specific note, the corresponding note is promptly removed from the list, reflecting real-time updates and maintaining data consistency across the application.

4.3. Note Display

Requirement Explanation: The capability to display all created notes ensures that users have easy access to their stored information.

Description Elaboration: Notes are presented in a structured list format within the app interface, featuring both their titles and content for comprehensive visibility.

Acceptance Criteria Detailing: Users can effortlessly navigate through the list of notes, viewing each note's details seamlessly without encountering any performance issues or display inconsistencies.

5. Non-functional Requirements

5.1. User Interface

Requirement Explanation: A user-friendly and responsive interface enhances the overall user experience, fostering engagement and usability.

Description Elaboration: The app's interface is designed with a focus on simplicity and clarity, facilitating intuitive navigation and interaction for users of all skill levels.

Acceptance Criteria Detailing: Users can interact with the app effortlessly, experiencing smooth transitions and responsive feedback across various devices and screen sizes, ensuring consistency and accessibility.

5.2. Performance

Requirement Explanation: Optimal performance is crucial for delivering a seamless and efficient user experience, minimizing loading times and delays.

Description Elaboration: The app's performance is optimized to ensure swift execution of note-related operations, including creation, deletion, and display.

Acceptance Criteria Detailing: Users can perform note-related tasks swiftly and seamlessly, even when dealing with a large volume of notes, without experiencing any degradation in performance or responsiveness.

6. System Requirements

Technology Overview: The Keeper app is developed using the React JavaScript library, leveraging its capabilities for building modern and dynamic user interfaces.

Compatibility: The app is compatible with a wide range of modern web browsers, ensuring broad accessibility and cross-platform functionality for users.

Server-side Considerations: As a client-side application, Keeper does not have specific server-side requirements, relying on client-side technologies for its operation and functionality.

7. User Interface Design

Header Design: The header section of the app prominently displays the app name, serving as a visual anchor for users and reinforcing brand identity.

Footer Design: The footer section of the app showcases the copyright year, providing users with information regarding ownership and legal rights.

Note-taking Area Layout: The note-taking area of the app features a clean and intuitive layout, incorporating input fields for note title and content, along with action buttons for creating and deleting notes.

8. Conclusion

In conclusion, this comprehensive Software Requirements Specification (SRS) document serves as a detailed guide for the development, testing, and deployment of the Keeper app. By elaborating on the functional and non-functional requirements, system constraints, and user interface design considerations, this document provides stakeholders and development teams with a clear roadmap for building a robust and user-friendly note-taking application. Through adherence to the outlined requirements and principles, the Keeper app aims to deliver a superior user experience, empowering users to organize and manage their notes efficiently and effectively.