



Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

DEHRADUN CAMPUS

PROJECT REPORT

**APP FOR
STUDENTS**

**B.Tech (CSE)
4th Semester**

SUBMITTED BY:

NAME: Pradeep Singh

UNIVERSITY ROLL No.: 2219247

CLASS ROLL No.: 44

SECTION: C1

COURSE: B.Tech (CSE)

SUBMITTED TO:

Ms. Manvi Bohra

Dept. of Computer science &
Engineering

Graphic Era Hill University



Graphic Era HILL UNIVERSITY

Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)

DEHRADUN CAMPUS

CERTIFICATE

University Roll No.: **2219247**

Class Roll No.: **44**

Certified that **PRADEEP SINGH** has developed mini project on “**APP FOR STUDENTS**” for the CS 4th semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

DATE-30/06/2024

Ms. Manvi Bohra

Class Co-ordinator

CSE- C1-IV-Sem

(CSE Department) GEHU

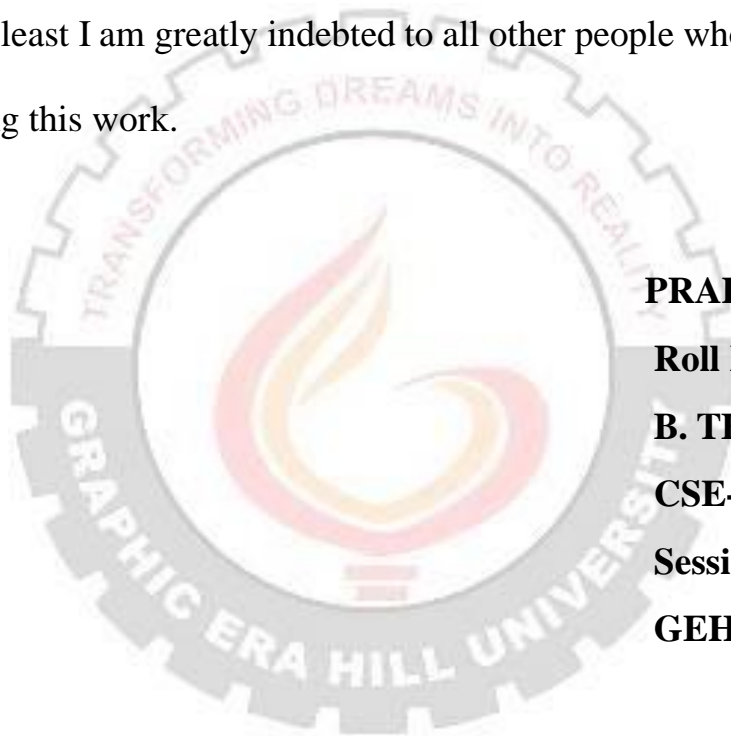
Dehradun

ACKNOWLEDGMENT

I would like to thank my parents for their continuing support and encouragement. I also wish to thank them for providing us with the opportunity to reach this far in our studies.

I would like to thank particularly our project Co- Ms. Manvi Bohra for her patience, support, and encouragement throughout the completion of this project and having faith in me.

At last, but not least I am greatly indebted to all other people who directly or indirectly helped us during this work.



PRADEEP SINGH

Roll No.- 2219247

B. TECH

CSE-C1-IV-Sem

Session- 2023-2024

GEHU, Dehradun

TABLE OF CONTENTS

1. Executive Summary

2. Introduction

3. Objectives

4. Literature Review

- 4.1 Overview of Note-Taking Applications for Students
- 4.2 Relevant Technologies and Tools Used

5. Implementation Details

- 5.1 HTML, CSS
- 5.2 React
- 5.3 Local Storage
- 5.3 User Interface

6. Key Features

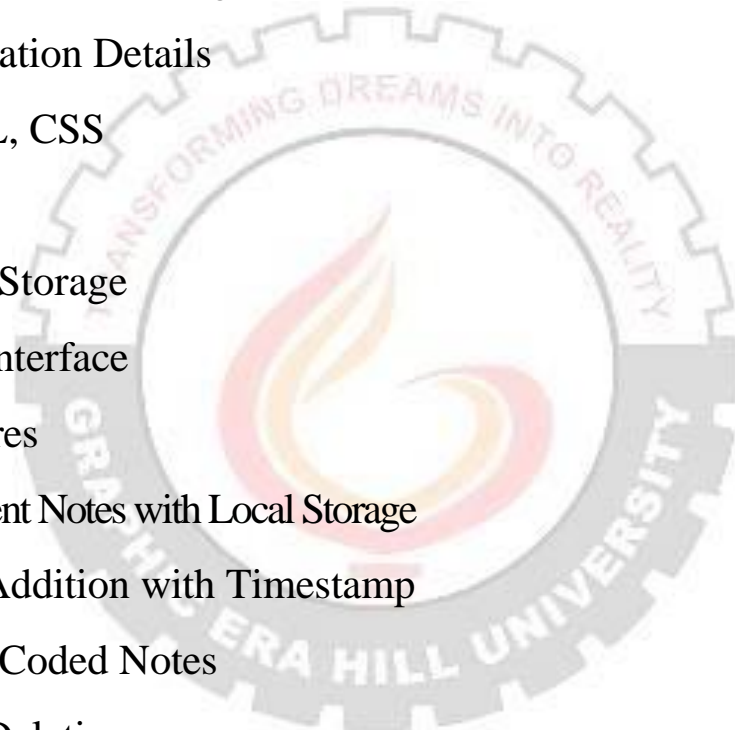
- 6.1 Persistent Notes with Local Storage
- 6.2 Note Addition with Timestamp
- 6.3 Color-Coded Notes
- 6.4 Note Deletion

7. System Workflow

- 7.1 User Interaction
- 7.2 Note Storage
- 7.3 Note Display and Management

8. User Instructions

- 8.1 Adding a Note
- 8.2 Deleting a Note



- Customizing the Note Color

9. Challenges and Solutions

- 9.1 Handling State Management
- 9.2 Implementing Local Storage
- 9.3 Ensuring a Responsive Design

10. Future Enhancements

- 10.1 Cloud Syncing
- 10.2 Rich Text Formatting
- File and Photo Upload

11. Conclusion

12. References

Executive Summary

This report details the development of a React-based note-taking application specifically designed for students. The app allows students to create, delete, and customize notes with timestamps and color coding, ensuring notes persist across sessions through local storage. This tool aims to enhance student productivity by providing an organized, digital platform for capturing and managing notes.

Introduction

Effective note-taking is a critical skill for academic success. Digital note-taking applications provide an accessible and organized platform for students to capture and retain information. This project aims to create a user-friendly, customizable note-taking application using React, designed specifically to meet the needs of students.

Objectives

The primary objectives of this project are:

- To develop a note-taking application using React tailored for students.
- To ensure notes persist across browser sessions using local storage.
- To provide functionalities for adding, deleting, and customizing notes.
- To display the timestamp for each note.
- To create a responsive design that works across various devices.

Literature Review

Overview of Note-Taking Application for Students

Note-taking applications for students often include features like cloud synchronization, multimedia integration, and organizational tools to enhance learning and productivity. These tools help students manage their academic workload, keep track of important information, and collaborate with peers.

Relevant Technologies and Tools Used

Modern note-taking applications utilize web technologies such as HTML, CSS, JavaScript, and frameworks like React. These tools provide a robust platform for building responsive, interactive applications. Local storage and cloud storage solutions ensure data persistence and accessibility, making it easier for students to access their notes from different devices.

Implementation Details

HTML & CSS

HTML (Hypertext Markup Language) forms the structure of the application. It defines the elements on the page, including headings, paragraphs, buttons, and forms, providing the basic framework for the user interface.

CSS (Cascading Style Sheets) is used to style the HTML elements, providing a visually appealing and consistent look and feel. CSS is essential for ensuring that the application is responsive and accessible on different devices.

React

React was chosen for its component-based architecture, allowing for efficient state management and reusability of UI components. This framework is ideal for building dynamic, interactive web applications.

Local Storage

Local storage is used to save notes in the browser, ensuring they persist across sessions without requiring a backend server. This approach provides a simple yet effective way to maintain data persistence.

User Interface

The user interface is designed to be intuitive and user-friendly, allowing students to quickly add, delete, and customize notes. The layout is responsive, ensuring usability across different devices, including desktops, tablets, and smartphones.

Key Features

Persistent Notes with Local Storage

Notes are stored in the browser's local storage, ensuring they remain available even after the browser is closed and reopened. This feature provides a reliable way for students to access their notes anytime.

Color Coded Notes

Students can choose a background color for each note, making it easier to categorize and visually organize their notes. Color coding can help in differentiating between subjects or priority levels.

Note Deletion

Students can easily delete notes they no longer need, helping them maintain a clutter-free workspace. The deletion process is straightforward, ensuring that students can quickly remove outdated or unnecessary notes.

System Workflow

User Interaction

Students interact with the application through a simple interface that allows them to add, delete, and customize notes. The UI is designed to be intuitive, making it easy for students to use the app without a steep learning curve.

Note Storage

Notes are stored in local storage in a JSON format. When the app is loaded, it retrieves and displays the saved notes. This approach ensures that notes are readily available and can be accessed quickly.

Note Display and Management

The app renders notes on the screen, allowing students to view, add, and delete them as needed. The interface updates in real-time as notes are modified, providing immediate feedback to the user.

User Instruction

Adding a Note

Students can add a new note by typing into the text area and clicking the "Add Note" button. They can also select a background color for the note before adding it. This process is designed to be quick and straightforward.

Deleting a Note

Each note has a delete button that students can click to remove the note from the list. This feature allows students to easily manage their notes and remove any that are no longer needed.

Customizing Note Color

Students can choose a color from the color picker before adding a note to customize its background. This feature helps in organizing notes by subject or priority.

Challenges and Solutions

This section anticipates and addresses potential challenges in the implementation of the website. The goal is to present a concise yet comprehensive approach to ensure the website's smooth operation and user satisfaction.

Handling State Management

Managing the state of notes and ensuring that updates propagate correctly through the app was a challenge. This was addressed by leveraging React's `useState` and `useEffect` hooks, which provided a robust solution for state management.

Implementing Local Storage

Ensuring that notes were properly saved and retrieved from local storage required careful handling of JSON data and `useEffect` to synchronize state changes. This implementation ensures data persistence and reliability.

Ensuring a Responsive Design

Designing a responsive UI that works well on various devices was achieved through the use of CSS flexbox and media queries. This approach ensures that the app is accessible and usable on different screen sizes..

Future Enhancements

Cloud Syncing

Adding cloud synchronization would allow students to access their notes from multiple devices, providing greater flexibility and convenience. This feature would also ensure that notes are backed up and can be retrieved even if the device is lost or damaged.

Rich Text Formatting

Providing rich text formatting options would enhance the note-taking experience, allowing students to style their text with bold, italics, and lists. This feature would make notes more readable and organized.

File and Photo Upload

Allowing students to upload photos and files would make the application even more versatile. This feature would enable students to attach images, documents, and other files to their notes, making it a comprehensive tool for managing study materials.

Conclusion

This project successfully demonstrates the creation of a React-based note-taking application with persistent storage and customizable notes. It provides a solid foundation for future enhancements such as cloud syncing, rich text formatting, and file uploads, making it an invaluable tool for students. The application is designed to be intuitive and user-friendly, helping students manage their academic tasks effectively.

References

1. <https://www.w3schools.com>
2. <https://developer.mozilla.org/en-US>
3. <https://stackoverflow.com>
4. [Introducing react.dev – React - GitHub Pages](#)
5. [Window: localStorage property - Web APIs | MDN - MDN Web Docs](#)