Low Level Design (LLD) MERN

Note App

Siddharth

Rukadikar

Document Version Control

Date Issued	Version	Description	Author
2025-04-07	1.0	LLD	Siddharth Rukadikar

Contents

Abstract

This document provides a detailed low-level design for the MERN Note App. The application enables users to authenticate, create, read, update, delete, pin, and categorize notes through a responsive user interface. The backend is built with Node.js and Express, utilizing MongoDB for data storage and JWT for authentication.

1 Introduction

1.1 Purpose of this Low-Level Design Document

The purpose of this document is to present a comprehensive description of the MERN Note App, detailing its functionalities, system interfaces, constraints, and expected behavior. This document serves as a guide for mentors, reviewers, and fellow learners evaluating the project on the iNeuron platform.

1.2 Scope

The MERN Note App is a full-stack web application designed to manage user notes. Key features include:

- User authentication (signup/login)
- CRUD operations for notes
- Note pinning and add tags.
- Responsive design for various devices
- Toast notifications for user feedback

1.3 Constraints

- The application relies on JWT for authentication.
- Notes are stored in MongoDB.
- The frontend is built with React and Redux Toolkit.
- The backend is built with Node.js and Express

1.4 Risks

- Potential security vulnerabilities associated with JWT if not handled properly.
- Data loss risk if MongoDB is not properly backed up.
- Performance issues with large datasets if indexing is not optimized.

1.5 Out of Scope

- Real-time collaboration features.
- Integration with external note-taking services.
- Offline functionality.

2 Technical specifications

2.1 Architecture

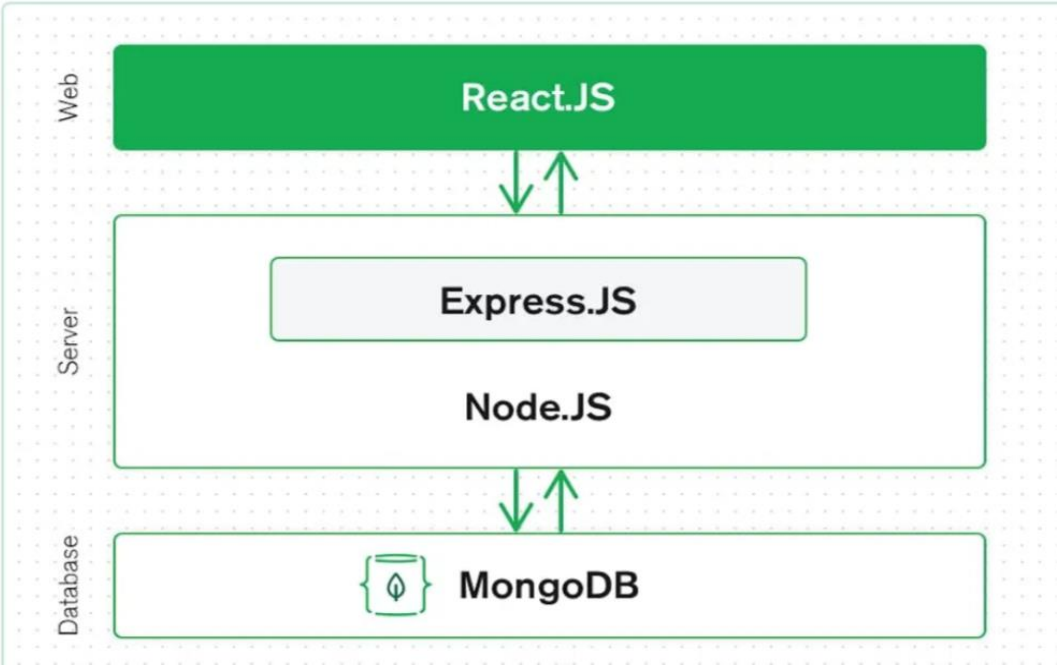
Frontend: React with Redux Toolkit for state management.

Backend: Node.js with Express.

Database: MongoDB

Authentication: JWT-based authentication

Architecture Diagram:

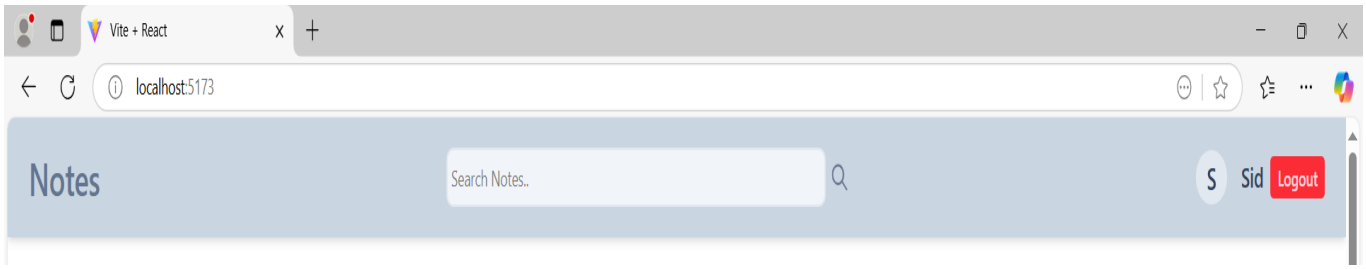


2.2 Component Design

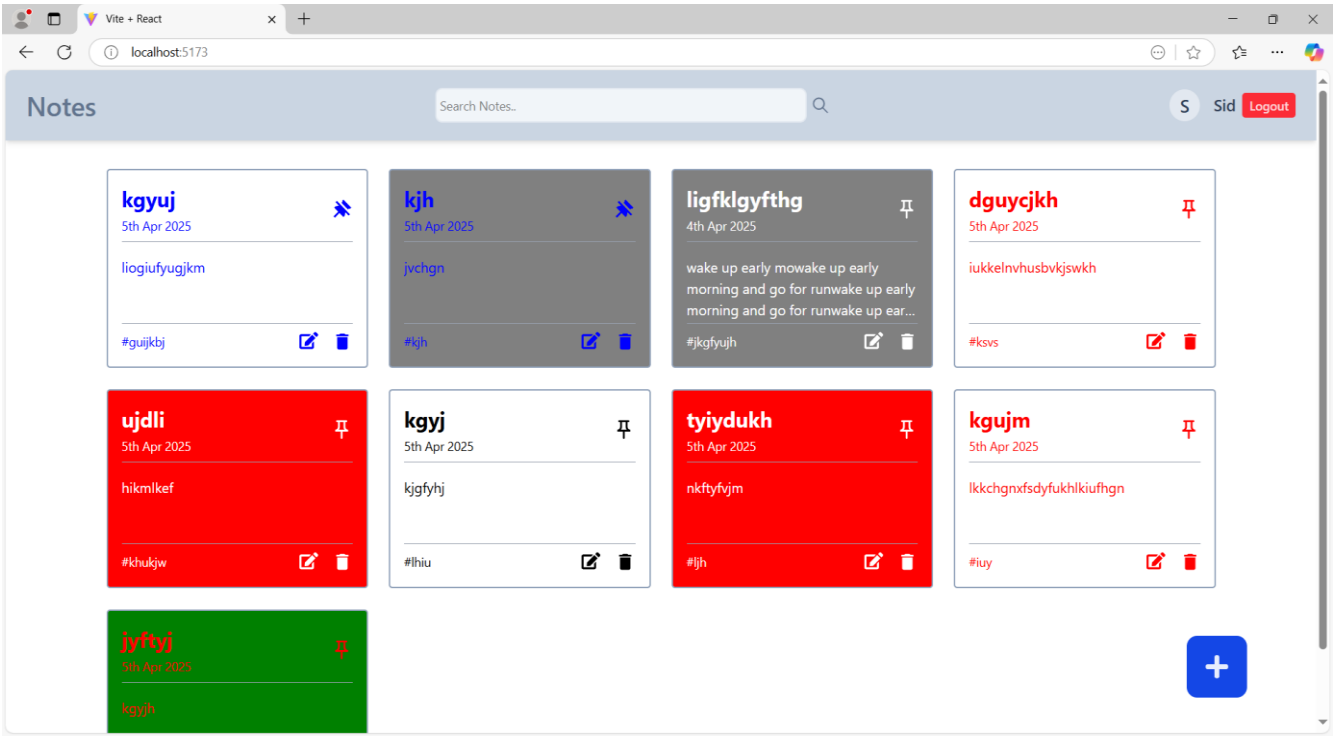
- **Navbar:** Contains the application logo, search bar, and user profile section.
- **Main Content Area:** Displays notes in a card format with options to pin, edit, or delete.
- **Note Modal:** Popup for creating or editing notes.

Screenshots:

Navbar:



Home Page:


Vite + React
localhost:5173
Notes
Search Notes..
S
Sid
Logout
kgyuj
5th Apr 2025
liogiufyugjkm
#guijkbj
kjh
5th Apr 2025
jvchgn
#kjh
ligfklgyfthg
4th Apr 2025
wake up early mowake up early morning and go for runwake up early morning and go for runwake up ear...
#jkgfyujh
dguycjkh
5th Apr 2025
iukkelnvhusbvkjswkh
#ksvs
ujdli
5th Apr 2025
hikmlkef
#khukjw
kgyj
5th Apr 2025
kjgfyhj
#lhiu
tyiydukh
5th Apr 2025
nkftyfvjm
#ljh
kgujm
5th Apr 2025
lkkchgnxfsdyfukhlkiufhgn
#iuy
jyftyj
5th Apr 2025
kgyjh

2.3 Database Design

User Schema:

- Email: String(unique)
- Password: String(hashed)
- createdAt:Date
- updatedAt: Date

Note Schema:

- **userId:**String
- **title:**String
- **content:** String
- **tags:**[String]
- **isPinned:** Boolean
- **bgColor:** String(default(white))
- **fontColor:** String(default(black))
- **createdAt:**Date
- **updatedAt:**Date

2.4 API Design

User Authentication:

- POST /api/auth/signup: Registers a new user
- POST /api/auth/login: Authenticates a user and returns a JWT.

Notes Management:

- GET /api/note/all : Retrieves all notes for the authenticated user.
- POST /api/note/add : Creates a new note
- POST /api/note/edit/:noteId : Updates an existing note.
- DELETE /api/note/delete/:noteId : Deletes a note.
- GET /api/note/getNote/:noteId : Retrieves single clicked note for the authenticated user.
- PUT /api/note/update-note-pinned/:noteId : update status of ispinned .
- GET /api/note/search : Retrieves all notes based on search query for the authenticated user.

2.5 Logging

- Logging is currently not implemented in this version of the MERN Note App For future

enhancement, the following logging solutions can be considered:

- Server-side Logging : using libraries like Winston or Morgan for capturing API events and errors.
- Client-side Logging : using tools like Sentry to track frontend exceptions and user interactions.

2.6 Deployment

- **Backend:** Deployed on a platform like Heroku or AWS EC2.
- **Frontend:** Deployed on platforms like Vercel or Netlify.
- **Database:** Hosted on MongoDB Atlas.

3 Technology stack

Front End	HTML/CSS/JS/React
Backend	Nodejs ExpressJs
Database	MongoDB
Deployment	AWS

4 Proposed Solution

The application will provide a seamless note-taking experience with features such as:

- Secure user authentication using JWT.
- Efficient state management with Redux Toolkit.
- Responsive UI designed with Tailwind CSS.
- CRUD operations for notes with options to pin.
- Toast notifications for immediate user feedback.

5 User I/O workflow

User Authentication:

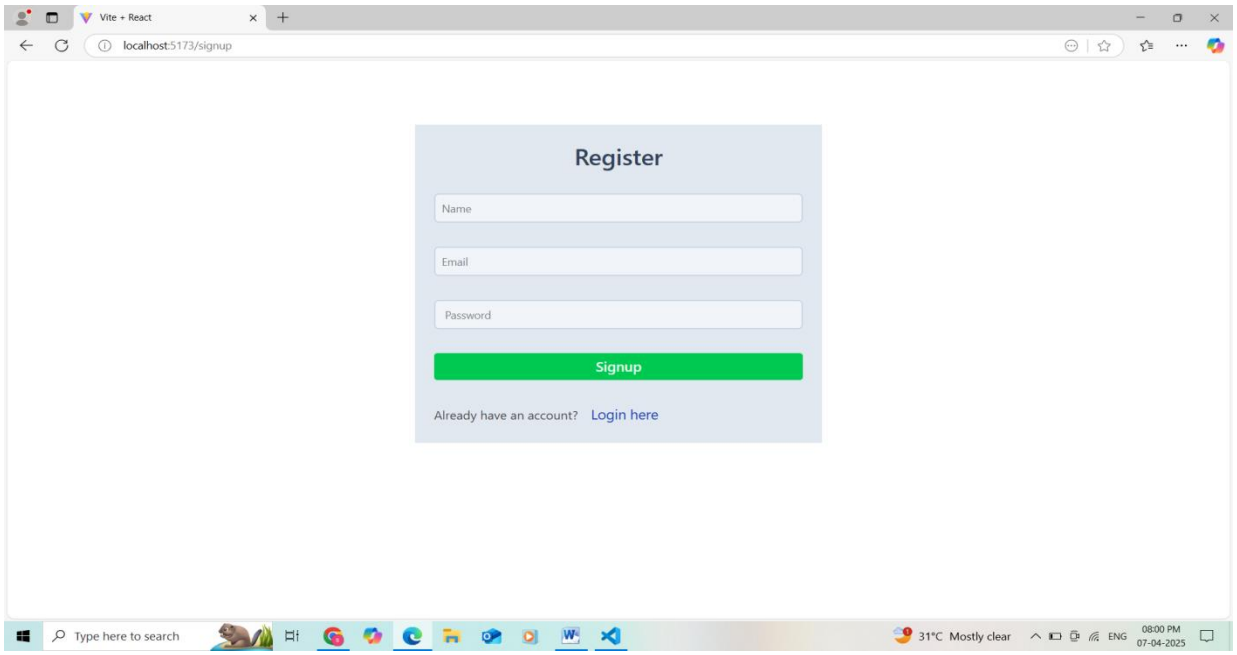
- User signs up or logs in.
- Upon successful authentication, the user is redirected to the dashboard.

Notes Management:

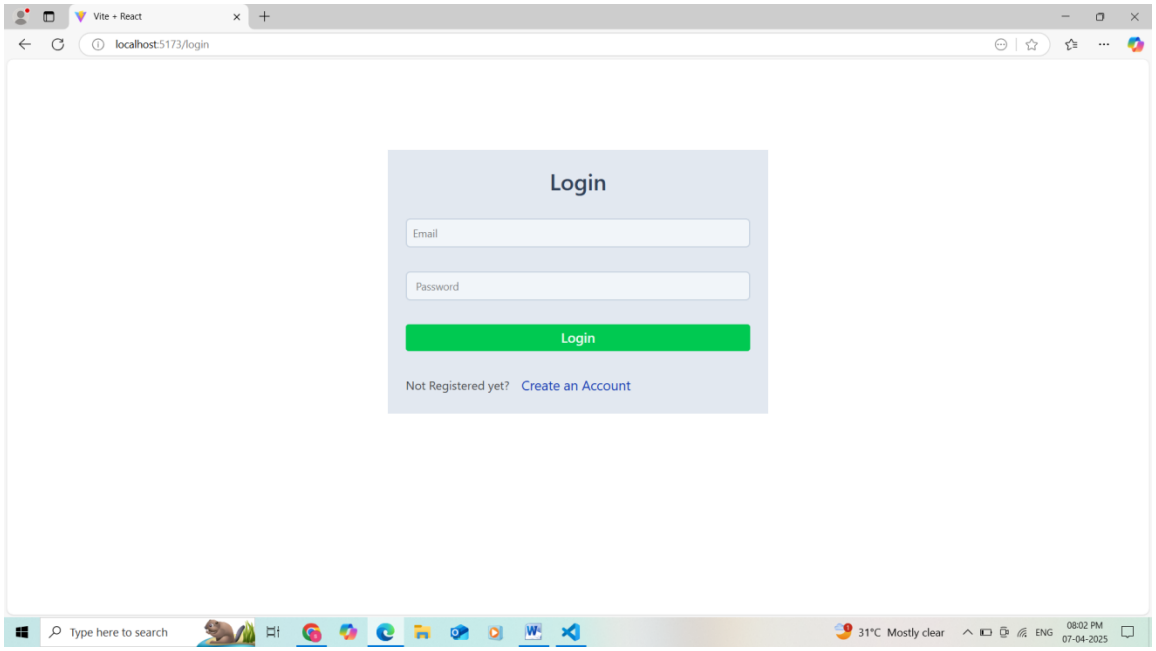
- User can create a new note using the Note Modal.
- Existing notes are displayed in the Main Content Area.
- User can edit, delete, pin, or categorize notes

Screenshots:

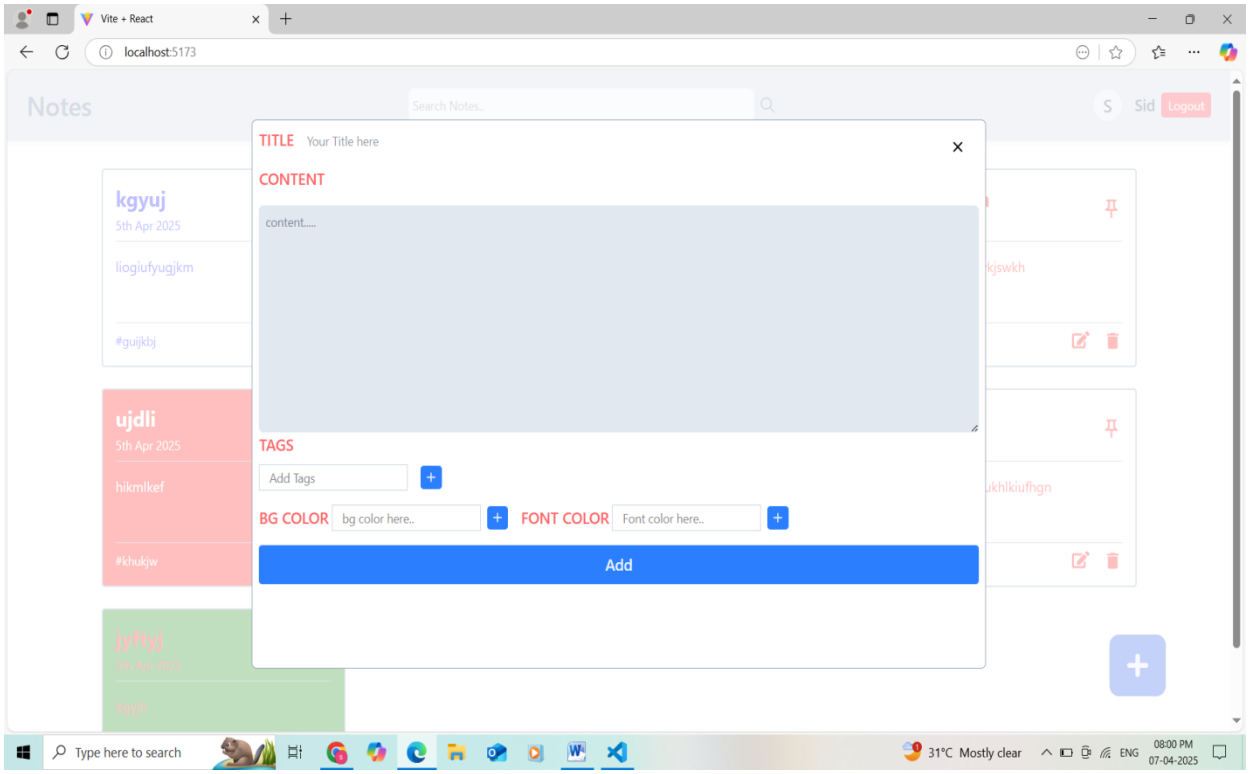
Register:



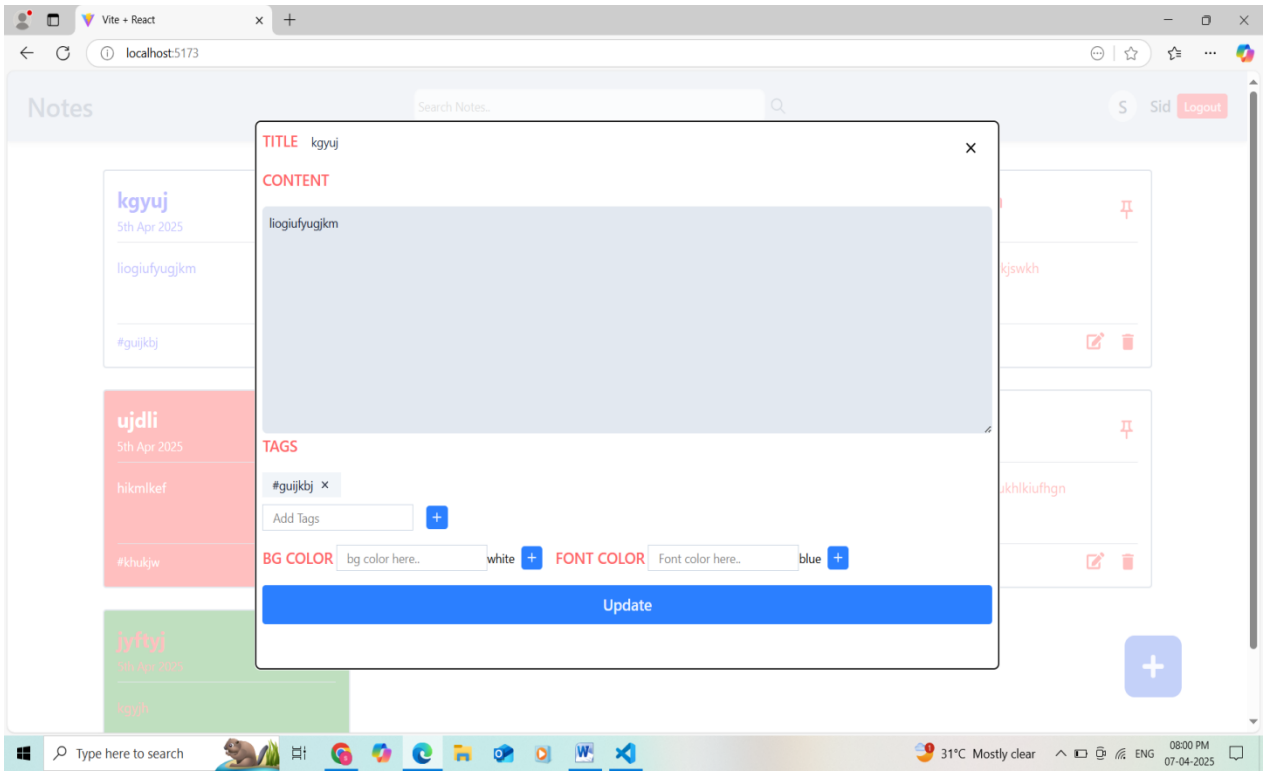
Login:



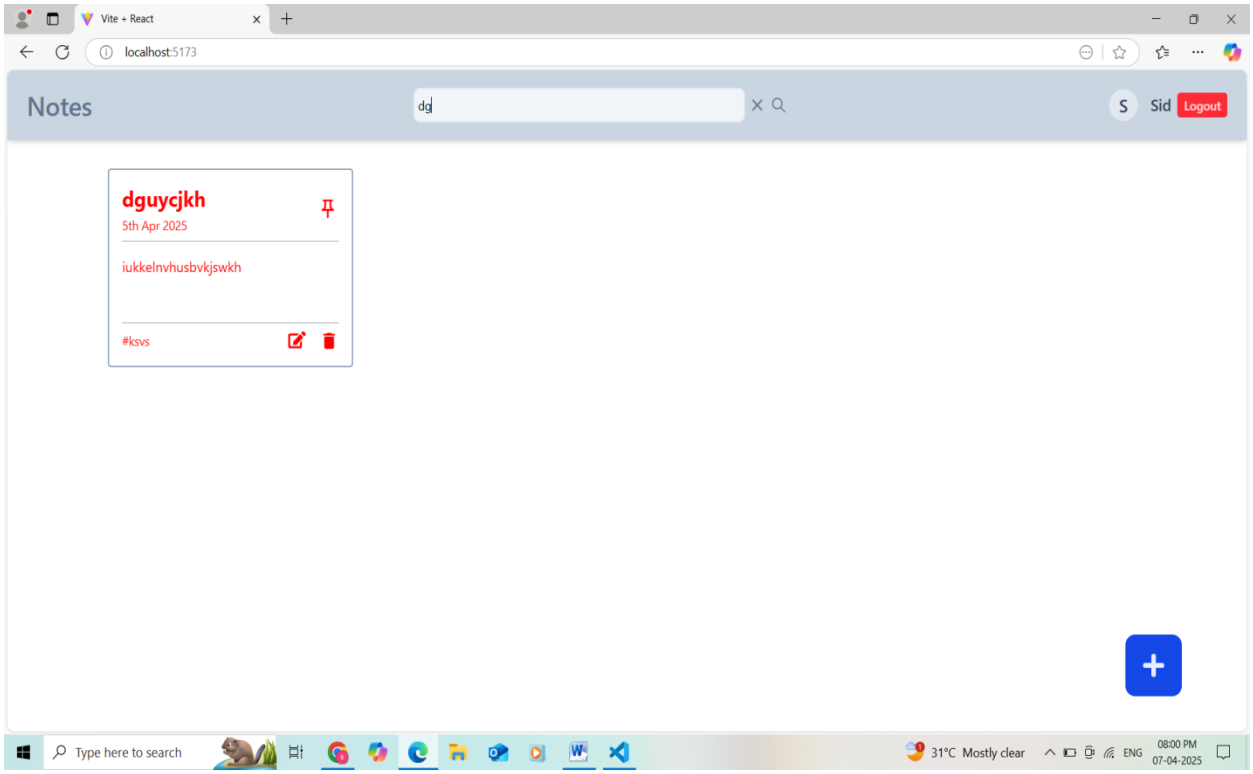
Add Note:



Edit Note:



Search Note:



6 Exceptional Scenarios:

Step	Execution	Mitigation	Module
Login	Invalid credentials	Show toast error message	Auth
Note Save	Missing fields	Show validation message	Notes
Delete	Note Not Found	Show:”Note not found”	Notes
Load	Server Not reachable	Display fallback UI with retry	Global/error

7 Test cases

Test case	Steps to perform test case	Module	Pass/Fail
TC01	Login With Valid user	Auth	Redirect to home
TC02	Create a new note	Notes	Note created
TC03	Pin a note	Notes	Note pinned
TC04	Delete a note	Notes	Note deleted
TC05	Edit a note	Notes	Update note content

8 Key performance indicators (KPI)

- Time and workload reduction using the app.
- Number of times a user accesses the app.
- Time between login and first note creation.
- Storage efficiency and performance with larger note counts.

9 Conclusion

The MERN Note App Low-Level Design outlines a modular and scalable architecture for building a user-friendly note management system. By leveraging modern web technologies such as React, Node.js, and MongoDB, along with features like JWT authentication and responsive design, the application ensures both functionality and security. This document will guide developers through implementation and help maintain consistency throughout the project lifecycle.