NoteHub: Collaborative Note-Taking Web App

Computer Science 343 Project 2 Report

Group Members:

Adam Gradner - 26128934

Ben St Arnaud - 25116835

Caeden Telfer - 25526693

Jordy Ward - 24913707

Regalo Dube - 23684534

Table of Contents

Introduction

NoteHub is a modern, collaborative web application designed to streamline the process of notetaking and sharing. In today's fast-paced digital world, the need for an efficient and intuitive platform for real-time collaboration has never been greater. *NoteHub* addresses this need by allowing users to create, edit, and share notes with others seamlessly. With built-in markdown support, it enables users to format their notes with ease, making them more structured and visually appealing in the app.

The application is built with a sleek, minimalist design and offers a comfortable user experience. The user interface, developed using React.js and styled with Tailwind CSS, ensures responsiveness across various devices while maintaining a clean layout. Whether on a desktop or mobile device, users can effortlessly interact with the app's features, ensuring a smooth and efficient workflow.

NoteHub also integrates real-time collaboration through WebSockets, allowing multiple users to edit the same note simultaneously. This feature, combined with user-friendly authentication and note management, makes it an ideal tool for teams and individuals looking to stay organized and collaborate efficiently.

In this report, we will discuss the development process, key features, and design choices that make *NoteHub* a robust and user-centric application.

System Requirements

Hardware Requirements

NoteHub is a web application that runs on any device with a modern web browser and an internet connection. Minimum requirements:

- **Device**: Computer, laptop, tablet, or smartphone.
- **Web Browser**: Latest version of Google Chrome, Firefox, Edge, or Safari.
- **Internet Connection**: Required for real-time collaboration.

Software Requirements

Front-End

- **Framework**: React.js with Tailwind CSS for styling.
- **Hosting**: Deployed on **Vercel** for fast, scalable front-end delivery.

Back-End

- **Framework**: Node.js and Express.js for the API.
- **Hosting**: Back-end hosted on Render for scalable server infrastructure.

Database

- **DBMS**: PostgreSQL, hosted on **Supabase** for real-time data management.

NoteHub leverages cloud services like Vercel, Render and Supabase to ensure a smooth, scalable, and secure user experience across all platforms.

Version Control with GitFlow

To streamline our development process, we implemented the GitFlow branching model. We used a dev branch for ongoing development efforts. For individual features, like feature/update-profile, we created separate branches that branched off from dev. Upon completing a feature, we merged it back into the dev branch, addressing any merge conflicts that occurred during the integration. Finally, after thorough testing and validation, we merged the dev branch into the main/master branch, ensuring that our production code remains stable and current.

Project Architecture

NoteHub follows a classic client-server architecture using the PERN stack (PostgreSQL, Express.js, React.js, and Node.js). The front end is built with **React.js** and styled using **Tailwind CSS**, providing a responsive and user-friendly interface.

The back end is powered by **Node.js** and **Express.js**, which handles the application's business logic and communicates with the database through a **REST API**. The REST API allows the front-end to perform CRUD operations for notes, users, and categories. It also supports real-time collaboration via **WebSockets**.

The **PostgreSQL** database is hosted on **Supabase**, ensuring fast and efficient data storage. The front end is deployed on **Vercel**, while the back end is hosted on **AWS (EC2)**, ensuring scalability and reliability.

This architecture enables smooth interactions between the client and server, ensuring real-time note-taking and seamless collaboration.

Database Design

- **ER Diagram**: *see Appendix A for ER Diagram*

Database Schema Overview

The database consists of four tables: categories, notes, users, and user_notes. It is normalized to **3NF/BCNF** to eliminate redundancy and ensure data integrity.

Tables and Relationships

- **categories**: Stores categories with category_id as the primary key and a name field.
- **notes**: Contains note data, including title, content, timestamps, and a foreign key (category_id) linking each note to a category. Each note belongs to one category (many-to-one relationship).

- **users**: Stores user details with user_id as the primary key, along with fields for username, email, password, and an optional user_avatar.
- **user_notes**: A junction table managing the many-to-many relationship between users and notes. It includes a composite primary key (user_id, note_id) and an is_creator boolean to indicate if the user created the note.

Normalization

The schema is fully normalized to **BCNF**, ensuring that all tables maintain data integrity by preventing redundancy, and each non-key attribute is functionally dependent on its primary key.

Authentication and Security

In ***NoteHub***, we implemented authentication using ***JWTs (JSON Web Tokens)*** to provide secure access to the application. Users must have a valid JWT to carry out actions like creating, editing, or deleting notes. After logging in, the token is created by encrypting the user's data with a secret key, ensuring secure storage and enabling stateless authentications for all subsequent requests. Moreover, JWTs help detect any unauthorized alterations, preserving data integrity and offering an added layer of security to safeguard data transmissions throughout the app.

Password Hashing Considerations

We chose **bcrypt** for password hashing due to its security and widespread support. **bcrypt** is derived from Blowfish, but specifically designed for securely hashing passwords. Unlike Blowfish, which is a general-purpose encryption algorithm, bcrypt incorporates salting and is intentionally slow to resist brute-force attacks. This makes bcrypt ideal for secure password storage in modern authentication systems.

By using **bcrypt** in combination with JWT-based authentication, *NoteHub* ensures robust protection for user credentials and API access, aligning with industry best practices for password security and user authentication.

User Interface

Design Principles

The user interface (UI) of *NoteHub* is designed to prioritize simplicity and usability, ensuring an intuitive experience for users. Built using **React.js** and styled with **Tailwind CSS**, the design follows a clean, minimalist approach to reduce clutter and provide users with a seamless note-taking experience. Key UI/UX principles such as consistency, feedback, and easy navigation were applied to ensure that users can effortlessly interact with the app's features, from creating notes to collaborating in real-time.

Screenshots

Visual examples of the key pages, including the login screen, notes list, note editor, and user profile, are provided in the appendix. These screenshots demonstrate the sleek and functional layout, along with the real-time collaboration features enabled by WebSockets.

Responsive Design

The application is fully responsive, adapting to various screen sizes and devices. The use of **Tailwind CSS** ensures that the layout adjusts smoothly whether accessed from a mobile phone, tablet, or desktop. Elements like the navigation bar, buttons, and note editor dynamically resize and reflow to provide an optimal user experience regardless of the device being used.

Accessibility

Accessibility was a key consideration in the design of *NoteHub*. The app incorporates features such as keyboard navigation, screen reader compatibility, and sufficient color contrast to ensure it is usable by individuals with varying abilities. By adhering to modern web accessibility standards, we aimed to make *NoteHub* inclusive for all users.

For detailed screenshots of the interface, please refer to the appendix.

Features and Functionalities

Core Features

- **Note Creation, Editing, and Deletion**: Users can easily create, edit, and delete notes. Each note is stored in a PostgreSQL database, and changes are automatically reflected in the user's notes list.
- **Real-Time Collaboration**: Using **WebSockets**, multiple users can collaborate on a single note in real time. Changes made by one user are instantly visible to others without the need for refreshing the page.
- **Markdown Support**: Notes can be formatted using **Marked.js**, which allows users to utilize markdown syntax for headings, lists, code blocks, and more, making their notes structured and easier to read.
- **User Registration, Login, and Profile Management**: Users can create accounts, log in securely, and manage their profiles. This includes updating personal details and avatars, with authentication handled through hashed and salted passwords.

Additional Features

- **Password Reset via Email**: If a user forgets their password, they can request a password reset link via email, ensuring they can regain access to their account quickly.

These features collectively create a dynamic and collaborative note-taking platform that offers a seamless user experience across various devices.

Implementation Details

Front-End

The front-end of *NoteHub* was built using **React.js** for a modular structure and styled with **Tailwind CSS** for responsive design. The front-end communicates with the backend through REST API calls to handle user actions like note creation and editing. Components were designed for easy navigation and interaction.

Back-End

The backend, built with **Node.js** and **Express.js**, manages the app's logic and processes API requests. Data is stored in a **PostgreSQL** database (via Supabase), and the backend handles all CRUD operations for users and notes, ensuring smooth interaction between the UI and database.

Real-Time Collaboration

WebSockets enable real-time note collaboration, allowing multiple users to edit notes simultaneously, with changes reflected instantly for all participants.

Third-Party Libraries

- **Marked.js**: Adds markdown support for note formatting.
- **Auth.js**: Handles secure user authentication.
- **Dotenv**: Manages environment variables securely.
- **WebSocket API**: Enables real-time collaborative editing.
- **BCrypt:** Used to securely store passwords in database via password hashing

Testing and Validation

Unit Tests

Unit tests were executed through **GitLab CI/CD**, focusing on key functionality such as user creation, login, and handling incorrect user details. These tests ensured that each component of the authentication system performed as expected, validating edge cases like invalid logins and ensuring proper responses from the API.

Due to limitations on the shared docker runner, members ran the front-end test before committing or pushing a change to the front end. This executes a script that conducts the frontend build locally. Backend integration tests were performed on the shared Docker Runner.

See appendix B for pipeline showing build test frontend and test backend

Manual Testing

Manual testing was performed to validate user experience, covering UI responsiveness, real-time collaboration, and profile management. This ensured that the app functions correctly across various devices and scenarios.

CI/CD Integration

Using **GitLab CI/CD**, we automated the API testing process. Unit and integration tests were run on every commit, and **GitLab secret variables** were used to securely manage environment variables, ensuring secure and consistent deployments across different environments.

Conclusion

NoteHub successfully meets its goal of creating a collaborative, real-time note-taking web app. With features like markdown support, secure authentication using JWT and bcrypt, and WebSocket-enabled collaboration, the platform offers a seamless and efficient user experience. The use of the PERN stack, alongside cloud services like Vercel and Supabase, ensures the app is scalable and reliable.

Despite challenges such as WebSocket integration, we overcame them through effective teamwork and adopted best practices like GitFlow and CI/CD. While NoteHub is fully functional, future improvements such as enhanced OAuth support and notifications can be explored. Overall, this project has been a rewarding experience in full-stack development.

Contributions

Breakdown of work

Adam Gradner: Front end and User Experience, Testing, Partial Documentation

Ben St Arnaud: Backend functionality and CI/CD, Testing

Caeden Telfer: Database and Authentication Implementation, Testing

Regalo Dube: Documentation, Authentication Research, CI/CD and Code Review

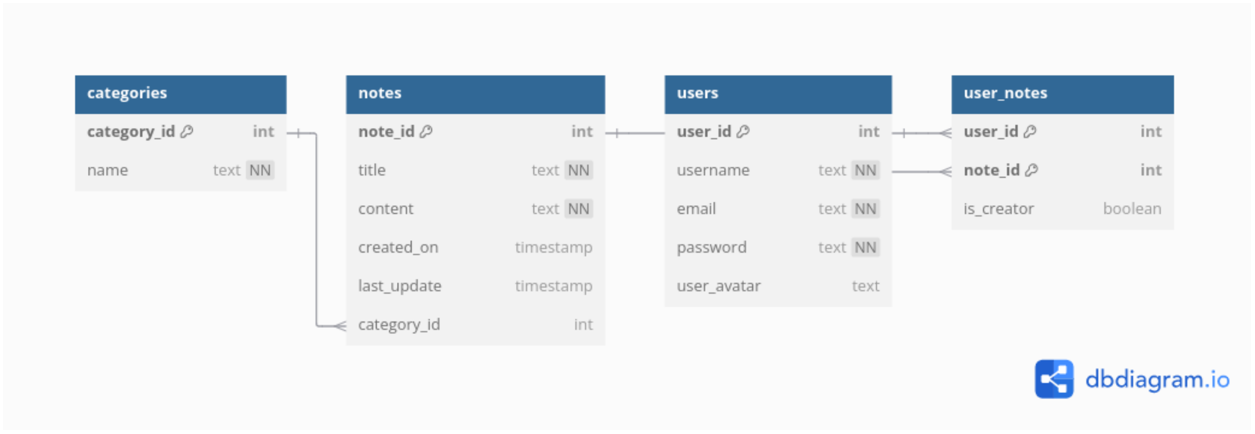
Jordan Ward: UI Improvements, Testing

Git Graph

See Appendix for Git Graph and Git Pipeline Results

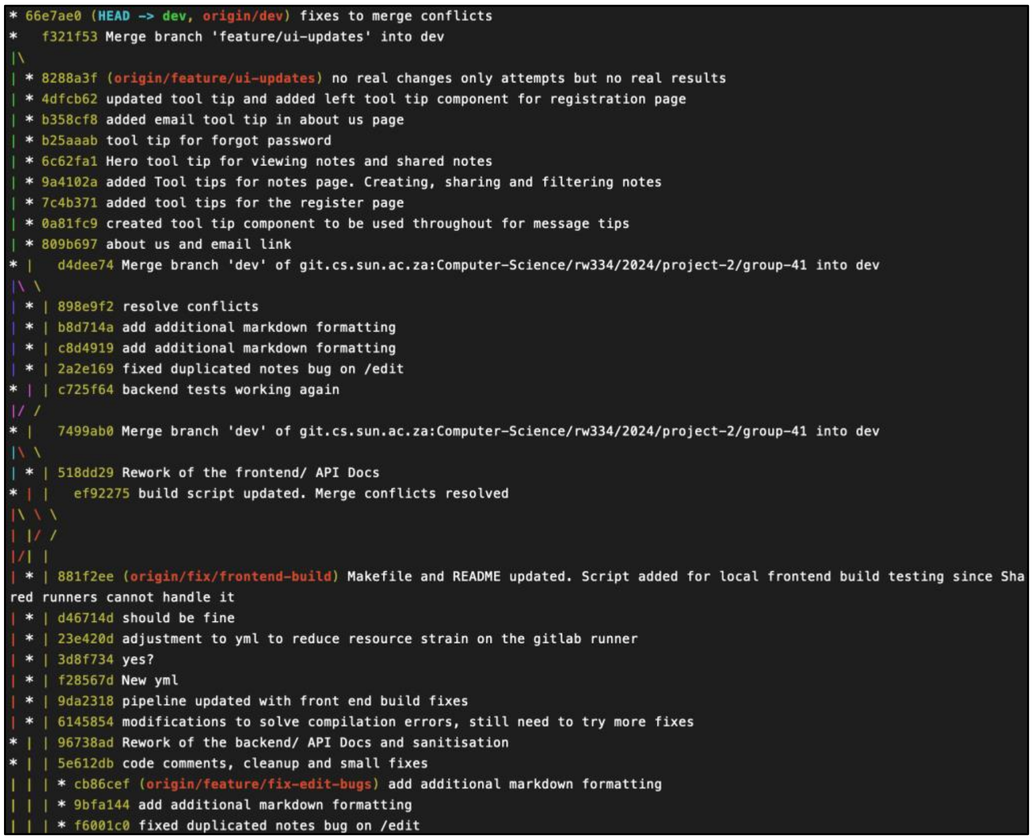
Appendices

Appendix A: Database Entity-Relationship Diagram.



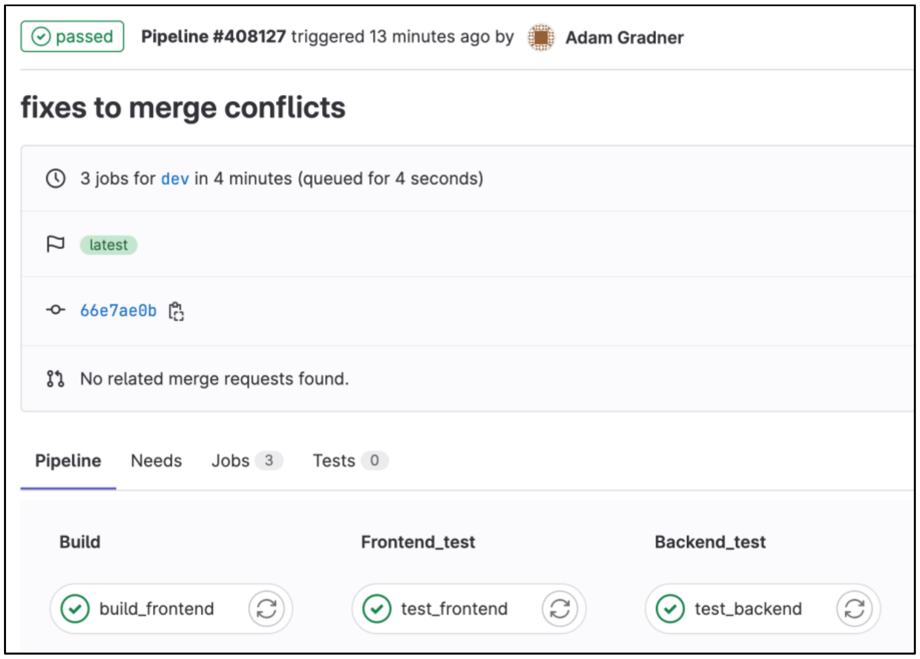
NoteHub Entity-Relationship Diagram: This diagram illustrates the relationships between entities in the database across different tables. As shown, the 'users' table stores all information about current users of NoteHub. When a user registers, they are assigned a unique user_id, which serves as the primary key. When a user creates a note, their user_id is linked to it, as indicated by the foreign key relationship in the 'user_notes' table. Notes are associated with different categories, and this relationship is shown by the foreign key in the 'notes' table, which links to the 'categories' table. This overall structure ensures referential integrity, allowing efficient management of user and note data, while also supporting scalability as more users and categories are added.

Appendix B: Git logs or contribution summaries.



```
* 66e7ae0 (HEAD -> dev, origin/dev) fixes to merge conflicts
*   f321f53 Merge branch 'feature/ui-updates' into dev
|\
| * 8288a3f (origin/feature/ui-updates) no real changes only attempts but no real results
| * 4dfcb62 updated tool tip and added left tool tip component for registration page
| * b358cf8 added email tool tip in about us page
| * b25aaab tool tip for forgot password
| * 6c62fa1 Hero tool tip for viewing notes and shared notes
| * 9a4102a added Tool tips for notes page. Creating, sharing and filtering notes
| * 7c4b371 added tool tips for the register page
| * 0a81fc9 created tool tip component to be used throughout for message tips
| * 809b697 about us and email link
* |   d4dee74 Merge branch 'dev' of git.cs.sun.ac.za:Computer-Science/rw334/2024/project-2/group-41 into dev
|\ \
| * | 898e9f2 resolve conflicts
| * | b8d714a add additional markdown formatting
| * | c8d4919 add additional markdown formatting
| * | 2a2e169 fixed duplicated notes bug on /edit
* | | c725f64 backend tests working again
|/ /
* |   7499ab0 Merge branch 'dev' of git.cs.sun.ac.za:Computer-Science/rw334/2024/project-2/group-41 into dev
|\ \
| * | 518dd29 Rework of the frontend/ API Docs
* | |   ef92275 build script updated. Merge conflicts resolved
|\ \ \
| |/ /
|/| |
| * | 881f2ee (origin/fix/frontend-build) Makefile and README updated. Script added for local frontend build testing since Sha
red runners cannot handle it
| * | d46714d should be fine
| * | 23e420d adjustment to yml to reduce resource strain on the gitlab runner
| * | 3d8f734 yes?
| * | f28567d New yml
| * | 9da2318 pipeline updated with front end build fixes
| * | 6145854 modifications to solve compilation errors, still need to try more fixes
* | | 96738ad Rework of the backend/ API Docs and sanitisation
* | | 5e612db code comments, cleanup and small fixes
| | | * cb86cef (origin/feature/fix-edit-bugs) add additional markdown formatting
| | | * 9bfa144 add additional markdown formatting
| | | * f6001c0 fixed duplicated notes bug on /edit
```

Figure 1: Git Log Tree for "Dev" Branch



passed **Pipeline #408127** triggered 13 minutes ago by **Adam Gradner**

fixes to merge conflicts

3 jobs for dev in 4 minutes (queued for 4 seconds)

latest

66e7ae0b

No related merge requests found.

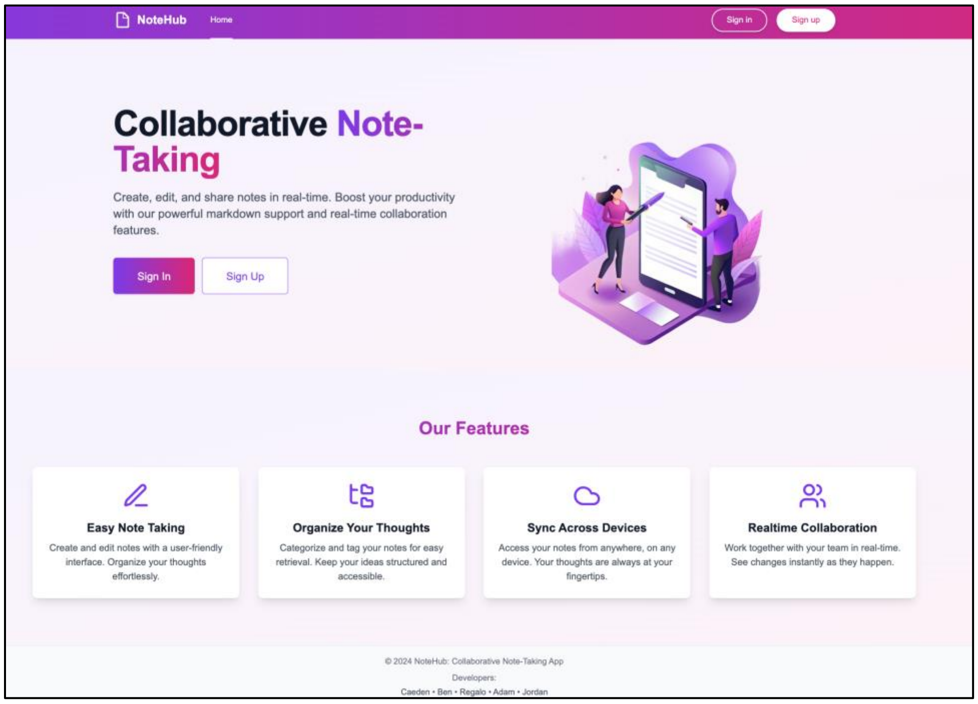
Figure 2: Git CI/CD Pipeline Tests

Appendix C: External libraries used.

Library	**Version**	**Purpose**
Express	4.21.0	Used to create the backend server and handle API routing for the web application.
Socket.io	4.3.1	Implemented real-time, bidirectional communication between the server and the client for features like collaborative note editing.
bcrypt	5.1.1	Used for hashing passwords to securely store user credentials in the database.
jsonwebtoken (JWT)	9.0.0	Implemented user authentication by generating and verifying JSON Web Tokens for secure API access.
uuid	10.0.0	Used to generate unique identifiers for various elements such as users and notes.
cors	2.8.5	Used to enable Cross-Origin Resource Sharing (CORS) for secure communication between the backend and frontend hosted on different origins.
dotenv	16.4.5	Used to load environment variables from a .env file for secure and configurable application settings.
validator	13.12.0	Used for input validation to ensure that user inputs, such as email addresses,

		are correctly formatted and safe.
nodemailer	6.9.15	Used for sending emails from the backend, such as for user registration and password recovery notifications.
supertest	7.0.0	Used for testing the API endpoints of the application, ensuring that they return the correct responses.
yjs	13.6.19	Used to enable real-time collaboration and synchronization of shared document states between users.
y-websocket	2.0.4	Integrated with WebSocket to handle the synchronization of Yjs documents for real-time collaborative editing features.
@supabase/supabase-js	2.45.4	Used for interacting with the Supabase backend as a database service, handling authentication, storage, and real-time capabilities.
nodemon	3.1.7	Used in development to automatically restart the server when file changes are detected.
concurrently	9.0.1	Used to run multiple development processes (e.g., server and frontend) concurrently during the development phase.

Appendix D: Examples of Web-App Features



Collaborative Note-Taking

Create, edit, and share notes in real-time. Boost your productivity with our powerful markdown support and real-time collaboration features.

Our Features

Easy Note Taking
Create and edit notes with a user-friendly interface. Organize your thoughts effortlessly.

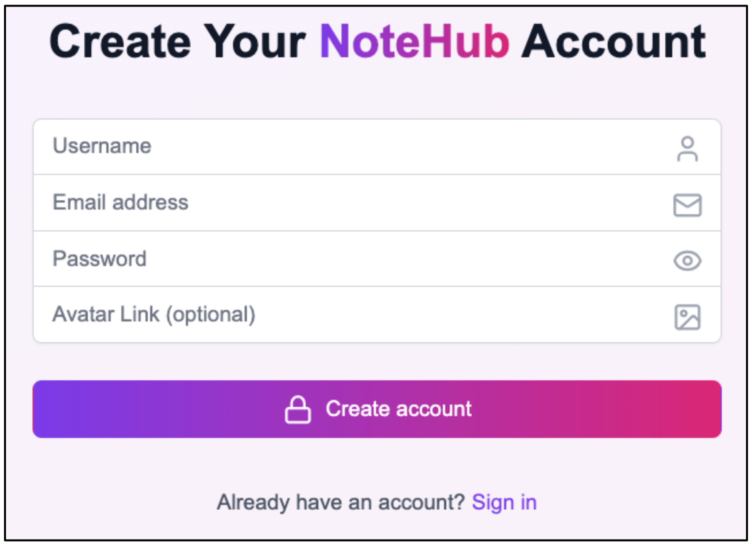
Organize Your Thoughts
Categorize and tag your notes for easy retrieval. Keep your ideas structured and accessible.

Sync Across Devices
Access your notes from anywhere, on any device. Your thoughts are always at your fingertips.

Realtime Collaboration
Work together with your team in real-time. See changes instantly as they happen.

© 2024 NoteHub: Collaborative Note-Taking App
Developers:
Caeden • Ben • Regalo • Adam • Jordan

Figure 1: Landing Page


Create Your NoteHub Account

Already have an account? Sign in

Figure 2: Register for NoteHub

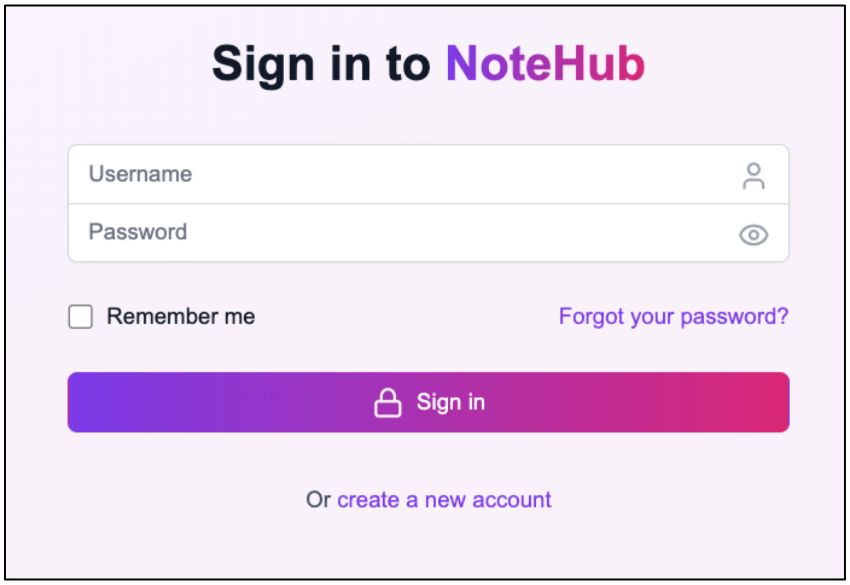


Figure 3: Sign In to NoteHub

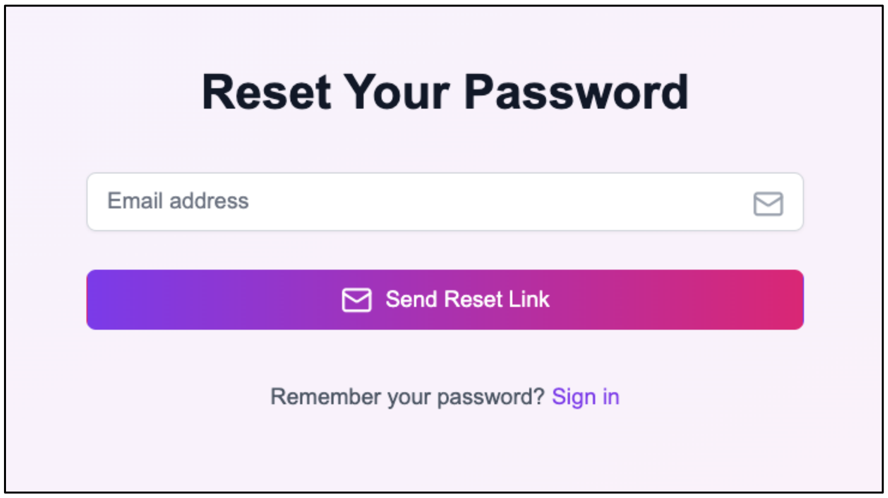


Figure 4: Reset NoteHub Password

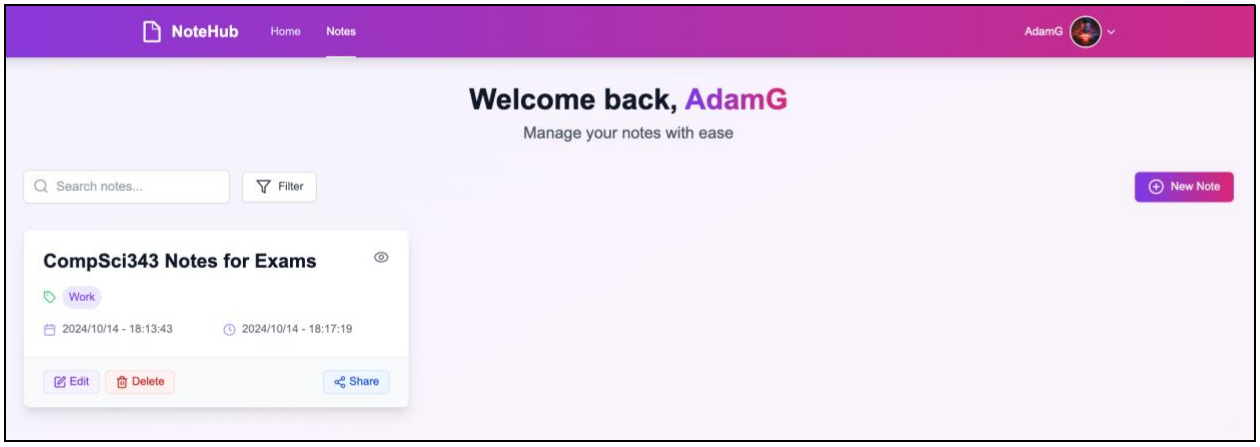
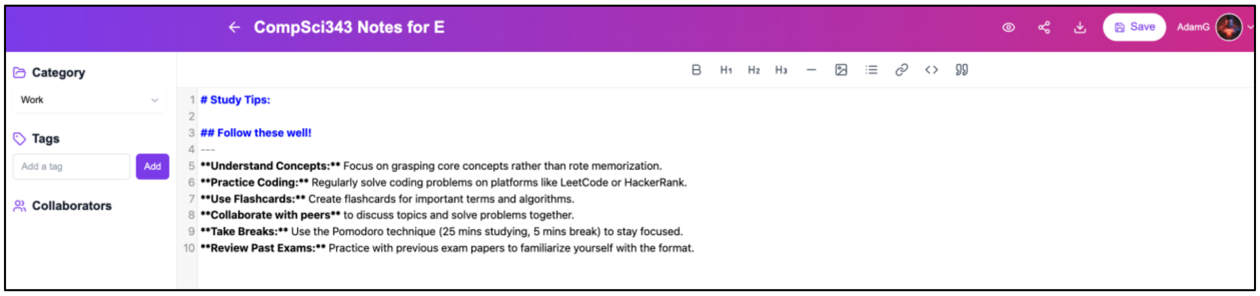


Figure 5: My Note Home Page



```
# Study Tips:

## Follow these well!
---
**Understand Concepts:** Focus on grasping core concepts rather than rote memorization.
**Practice Coding:** Regularly solve coding problems on platforms like LeetCode or HackerRank.
**Use Flashcards:** Create flashcards for important terms and algorithms.
**Collaborate with peers** to discuss topics and solve problems together.
**Take Breaks:** Use the Pomodoro technique (25 mins studying, 5 mins break) to stay focused.
**Review Past Exams:** Practice with previous exam papers to familiarize yourself with the format.
```

Figure 6: Editing using Marked