Notes Keeping App

Document Version Control

Date Issued	Version	Description	Author
11/01/23	1.0.0	Server initial setup	Ngamlenmang Touthang
12/01/23	1.0.1	User authentication added	Ngamlenmang Touthang
13/01/23	1.0.2	Notes controller and routes added	Ngamlenmang Touthang
14/01/23	1.0.3	Swagger docs added	Ngamlenmang Touthang
15/01/23	1.0.4	Client template with responsive added	Ngamlenmang Touthang
16/01/23	1.0.5	Client side authentication UI added	Ngamlenmang Touthang
18/01/23	1.0.6	Search feature added	Ngamlenmang Touthang
19/01/23	1.0.7	Cookie update	Ngamlenmang Touthang
02/02/23	1.0.8	Documents updated	Ngamlenmang Touthang

Contents

Abstract

A web-based application designed to simplify note taking, the app offers an easy-to-use platform for organizing and saving information. Users can create, edit, customize note background and font colour and also search through their notes quickly and efficiently with its title. With the ability to access their notes form any device with the internet connectivity, this application is perfect for both personal and professional use.

1. Introduction

1.1. Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model of coding. The document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project.
- List and describe the non-functional attributes like
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2. Scope

The HLD documentation presents the structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HDL uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. General Description

2.1. Product Perspective

The Notes keeping application is a web application system which allow users to take notes. In simple words, the note keeping app can be described as a web-based notes taking web app where a user can store all of his significant notes and retrieve them from any location at any time.

2.2. Problem Statement

Design a Notes Keeping application using the MERN (MongoDB, Express, React and Node) stack

2.3. Proposed Solution

The proposed solution for a notes-keeping web application include the following features

1. User Accounts: Allow users to create an account and sign in to access their notes from any device.
2. Note creation and editing: Provide an intuitive and user-friendly interface for users to create, edit and delete notes.
3. Customizable background and font colour: Provide modifiable note card background and font colour.
4. Search functionality: implement a search bar to help users quickly find specific notes base on the notes title name.
5. Mobile Compatibility: Ensure that the application is accessible and functional on both desktop and mobile devices.
6. Security and privacy: Implement robust security measures to protect users data and ensure the privacy of their notes.
7. Timestamp: to help the user better understand when the note was written, the creation and modification dates is provided to every note.

2.4. Further Improvements

The notes keeping app can be further improve with the following list or features

1. Categorization and tagging: allow users to categorize their notes and add tags for easy organization and retrieval
2. Reminders: Enable users to set reminders for specific notes and receive notification on the due date.
3. Note sharing: Offer the ability for users to share their notes with others, either by sending a link or directly sharing the note.
4. OAuth authentication: Integrate the authentication with google or facebook account login system.
5. Masonry layout: Implement masonry layout for proper alignment of the note containers.

2.5. Technical Requirements

This document addresses some of the requirements for proper functioning of the notes-keeping app

- User Authentication: The application should provide secure user authentication to ensure only authorized users have access to their notes.
- Notes storage: The application should have a secure database for storing user notes, along with any metadata such as creation and modification dates.
- User Interface: The application should have a user-friendly interface for creating, editing and viewing notes, with options for organizing them.
- Accessibility: The application should be accessible from any device with an internet connection, such as desktop, laptop tablet and smartphone.

- Security: The application should implement security measures to protect user data, such as encryption for password in transit and at rest, and secure database access.
- Scalability: The application should be designed to scale to accommodate a growing number of users and increasing amounts of data.
- Performance: The application should have fast and responsive performance, with low latency and high availability.

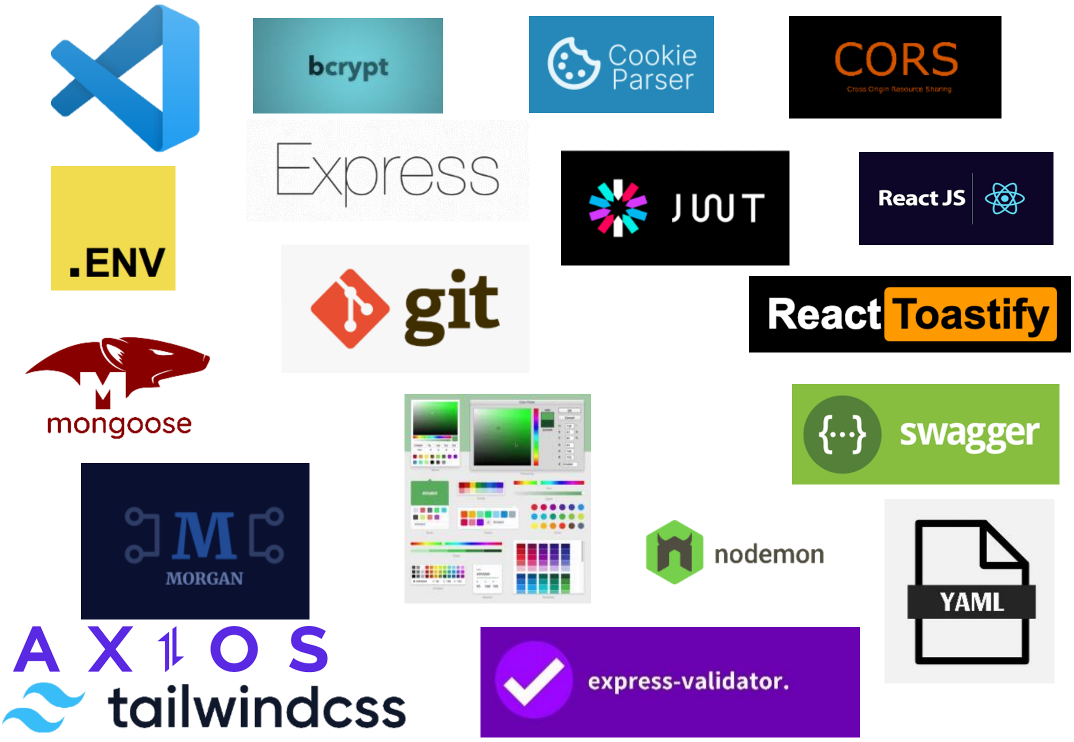
2.6. Data Requirements

The following data requirement should be considered:

1. User information: username, password and email for profile information
2. Notes: each note should have a title and note description
3. Dates: each note should also maintain timestamp of created time and updated time.

2.7. Tools used

JavaScript programming language is the main programming language to be used along with some of the libraries and frame works.



- Vscode is used as IDE
- Backend end developmet is done using Nodejs and some of the frame work and libraries like express, mongoose, cookie-parser, bcryptjs, jsonwebtoken, dotenv, json, webtoken, swagger, nodemon, validator, yamljs and morgan
- Front end development is done using Reactjs, TailwindCss , React toastify
- Axios is used to make http/https request
- GitHub is used as version control system
- MongoDB is used to retrieve, insert, delete, and update the database.
- MongoDB atlas is use for storing the data.
- Netlify and Railway is used for deployment of the application.

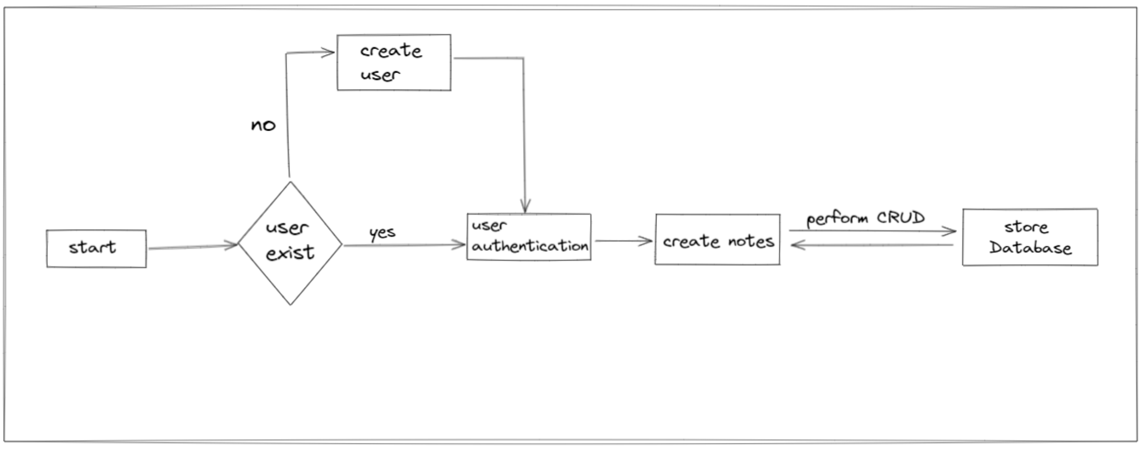
3. Design Details

3.1. Process Flow

In order to allow the users to add, retrieve delete and update the note, we will use the MERN stack to build it.

Below is the process flow diagram is as shown below.

Proposed methodology



1.1. Event log/Notification

The system should log/notify every event so that the user will what the app is running into.

- The system app will log all the event that occurs in the system.

1.2. Error handling

Should errors be encounterd, an explaination will displayed as to what went wrong? An error will be defined as accordingly. The system will display all the error in the user interface

1.3. Performance

The Notes keeper web application should be able to handle a number of request and performed within specific time frame and any users with Internet connection and browser should be able to use it anywhere they are.

1.4. Reusability

The code written and the components used should have the ability to be reused with no problems.

1.5. Application Compatibility

The different components for this project will be using a JavaScript frontend and backend libraries and framework. Each component will have its own task to perform, and it is the job of the JavaScript to ensure proper transfer of information.

1.6. Resource Utilization

When any request is performed. It will likely use all the processing resources available until that function is finished.

1.7. Deployment

The Notes keeper app deploy using Railway (backend), Netlify (frontend) and database (MongodDB Atlas) via github which is software version control system.

4. KPIs (Key Performance Indicators)

Key Performance Indicators (KPIs) are metrics used to measure the success and effectiveness of a product or system. For The notes keeping app, Some relevant KPIs may include:

1. User Retention: The percentage of users who continue to use the app over time, indicating satisfaction with its performance and features
2. Active User Count: The number of unique users who access the app regularly, indicating its popularity and usefulness
3. Note Creation and Editing Activity: The number of notes created and edited by users, indicating their level of engagement and use of the app
4. Search Usage: The frequency and effectiveness of the search functionality, indicating the usefulness of these features for users
5. Feedback and Rating: user feedback, ratings and reviews can provide valuable insight into the app's performance and areas of improvement.
6. Data Privacy and security: Measures to ensure the safety and security of user data, such as encryption and secure storage are critical to the success of the app.

This KPIs can provide valuable information in-order to evaluate the performance and impact of the notes keeping application.

5. Conclusion

In Summary, the notes keeper app should focus on delivering an effective and efficient solution for users to manage and organize their notes. Key features such as note creation, editing, searching and storing should be prioritized along with a user-friendly interface, data privacy and personalization.