Project Title	White Board Web Application
Technologies	MERN
Domain	Education
Project Level	Hard

Table

Table of Contents

1. Problem Statement:

Design a web based White Board Application.

1.1. Overview of White Board Application and MERN Stack?

Mongo DB, Express, React, and Node JS are all referred to as MERN. These four are currently among the most widely utilized technologies in the web development industry.

A web application called a "white board" is used mostly by users who want to instruct their students online. A user may interact and comprehend the concepts more readily with the aid of this web app by effortlessly drawing, writing, and other actions on the screen. User can simply jot it down and present it to their classmates; they don't need a produced PowerPoint to instruct them.

Note: - You can use any white board application to get a concept of the subject and a reference for it.

1.1.1 Features to be included in the White Board Application

Here is a comprehensive list of the characteristics that your web app should include. -

a. The website will be a single-page application.
b. A person can access the website directly and use it without any login or signup.

c. The user will be given the ability to alter the color of the pen, the backdrop color, the thickness of the pen, the shapes, and many other options via a **hamburger menu**.

d. Any drawing can be made immediately on the **whiteboard** using the **mouse, a finger, or a light pen**.

e. The user should be able to erase or delete a specific area or object that he had created in addition to clearing the entire whiteboard.

f. The user can utilize the whiteboard to do **UNDO** and **REDO** operations.

g. If a user wants to save his work, he must first **check in** or **create a new account** before saving it as a pdf or in any other format.

h. If a user is logged into his account, he can access his saved work.

i. User can use **CRUD operations** on his previously saved work.

1.1.2 List of expected option in hamburger menu-

a. **Pen** with multiple color option

c. **Highlighter** with multiple color option to highlight a selected area

d. Option to insert an **image** in the whiteboard

e. Option to write some **text** over the whiteboard

f. Option to insert some common **shapes** like circle, rectangle, line, arrow etc.

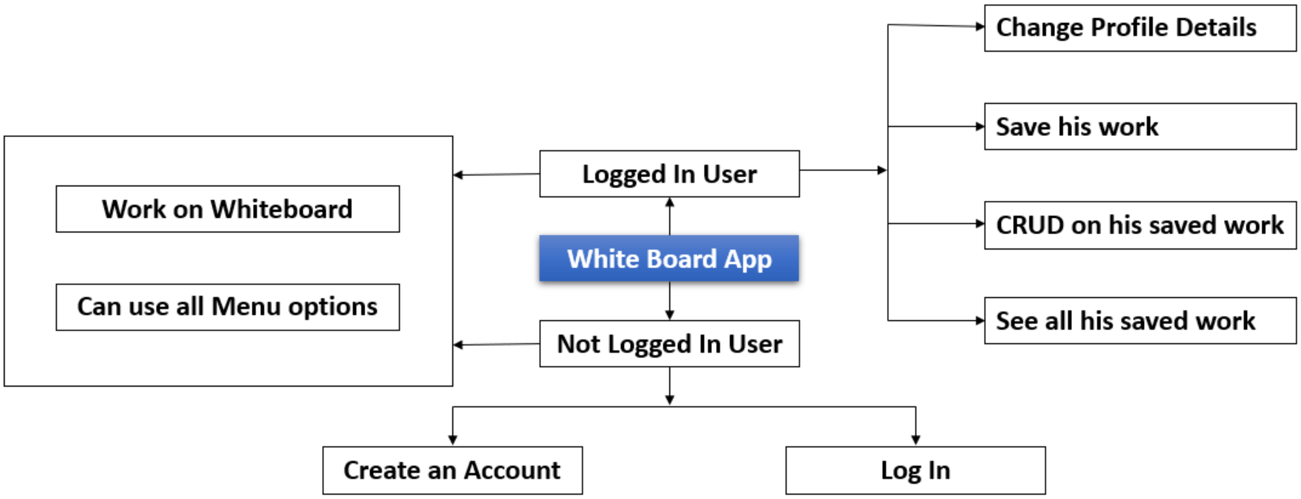
g. **Undo** and **Redo** option

h. Option to create **new blank whiteboard**.

i. **Eraser** to wipe out a part or whole whiteboard.

Note: Additional features from your side is welcomed.

Low Level user flow



1.2. Project Objective

• **Whiteboard Application:** A straightforward and user-friendly whiteboard application that allows users to generate content with ease and teachers to utilize it to increase student interaction and knowledge sharing.

• **User-friendliness:** The project should be exceedingly user-friendly, making it possible for even a newbie to use it.

• **Free / No account needed:** It is free for users to use and does not require an account to use.

1.3. Scope of The Project

1. As the popularity of online education increases, so does the demand for whiteboard applications.

2. To improve user experience, this project will support text, images, and shapes.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** When he wishes to save his progress or when he first lands on the page, the user must be able to register for the web app using an Email, Username, and Password in order to access the account-creation features.

2. **Creating New Whiteboard:** If the user is logged in, he can access all of his previously saved whiteboards, change them, and save them in addition to creating new ones.

3. **Undo and Redo:** It is a modest but extremely effective function that ought to be available to aid consumers.

5. **Save:** The user that is logged in can conduct **CRUD** operations on his saved work.

1.4.2. Non-Functional Requirements

1. **Privacy:** Only users that are logged in can use their saved files to conduct CRUD operations.

2. **Robustness:** To provide recoverability and prevent data loss in the event that the user's system crashes, a backup of the user's whiteboard must be kept on distant database servers.

3. **Performance:** The website must load the whiteboard quickly and be lightweight.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
User Dashboard	Display all the saved whiteboard and user details	User
Whiteboard	Display tools and options to the user to use whiteboard	User

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.