



Project Title	Web-Based Blogging Application
Technologies	MERN
Domain	Education
Project Level	Hard
Organization	iNeuron Intelligence Private Limited

Table

Table

1. Problem Statement:	2
1.1. Overview of Painting Application and MERN Stack?	2
1.1.1 Expected features of the Painting Application	2
1.2. Project Objective	3
1.3. Scope of The Project	3
1.4. Functional and Non-Functional Requirements: -	4
1.4.1. Functional Requirements	4
1.4.2. Non-Functional Requirements	4
1.5. Use Case Table	4
2. Project Evaluation Metrics:	4
2.2. Database:	5
2.3. API Details or User Interface:	5
2.4. Deployment:	5
2.5. Solutions Design:	5
2.6. System Architecture:	5
2.7. Optimization of solutions:	5
3. Submission requirements:	
3.1. High-level Document:	5

3.2. Low-level document:.....	5
3.3. Architecture:.....	5
3.4. Wireframe:.....	6
3.5. Project code:	6
3.6. Detail project report:	6
3.7. Project demo video:	6
3.8. The project LinkedIn a post:	6

1. Problem Statement:

Design a Painting Web Application.

1.1.Overview of Painting Application and MERN Stack?

Mongo DB Express React and Node JS are referred to as MERN. Web developers frequently employ this technological stack while building applications. Using only JavaScript and JSON, the MERN architecture makes it simple to build a three-tier architecture (front end, back end, database).

An application called a painting allows a user to create drawings by utilizing a pencil, text, and shapes in a variety of methods and styles. Kids and students utilize this well-liked application to develop relationships with computers in the classroom.

Microsoft Paint is one of the best examples that school children enjoy so much for showcasing their creativity. It is also the tool of choice for teachers to effortlessly and interactively teach pupils how to operate a mouse.

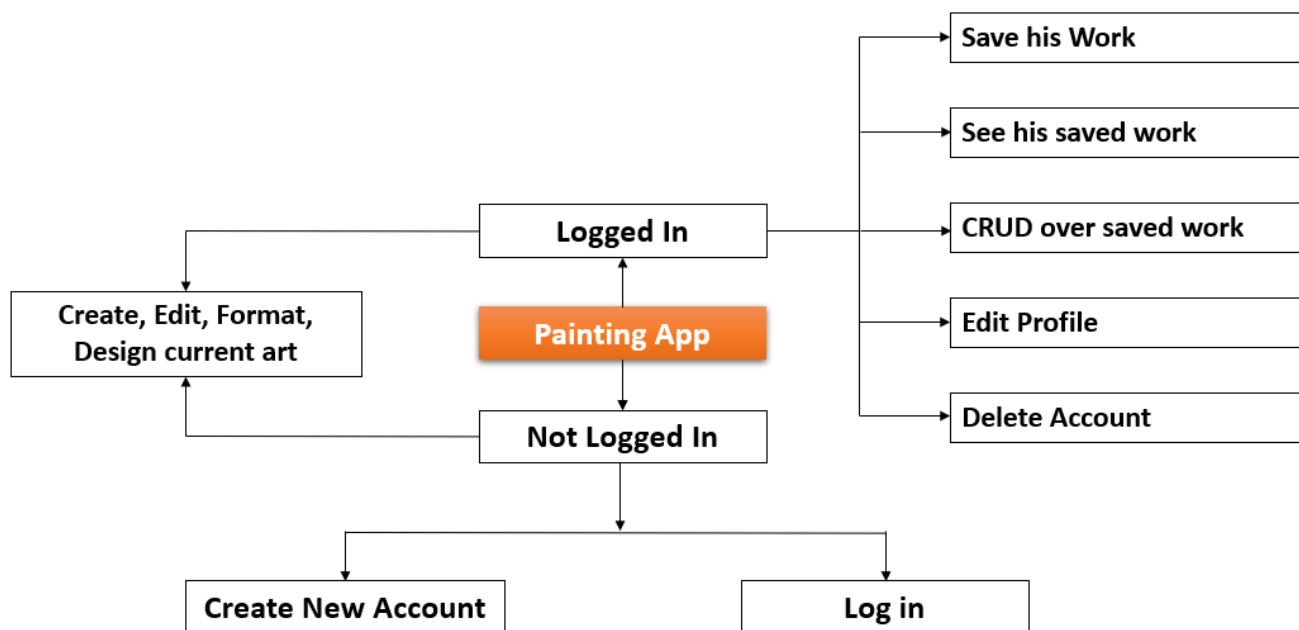
1.1.1 Expected features of the Painting Application

Here is a detailed list of the expected features which should be available in your Painting Application-

- The website will just have one page and have a highly **straightforward, interactive, and user-friendly user experience.**
- To save the users' drawings, a **login** mechanism is necessary; otherwise, the data will be deleted once the website is closed.
- Without logging in**, a user **cannot save, retrieve, or update** an existing file.
- A person who is **logged in** can change his name and password in his profile. He will be able to carry out **CRUD operations** on previously saved or already existing files.
- A user that is logged in can view all of his **saved files** at once.
- The user interface needs to be **simple and intuitive.**

- g. A **toolbar** should be provided so that the user has access to all of the tools that are available for use.
- h. You can use different tool bars for **basic, formatting, editing, and shapes**, among other things.
- i. A menu should be accessible from which the user can **save** their work in progress or access their **profile, files, log in/out, establish an account, and stored pictures**.
- j. On the screen, each tool should be clearly visible.
- k. The website ought to be **fully responsive**.

Low Level user flow



1.2. Project Objective

- **Painting Application:** For schoolchildren, a straightforward, cost-free painting application.
- **User-friendliness:** The application should have a very user-friendly interface so that anyone of any age can use it with ease.

1.3. Scope of The Project

1. Everyone is creative, and we all want to make things that can be conveniently used and engaging. This is the ideal setting for them because everyone wants to spend time with their children doing productive things and should also involve them in such activities.
2. This application will facilitate kids' computer interaction.
3. Through this software, they can play and express their creativity.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** The application must allow users to sign up using their email, username, and password. Users must be able to self-register or, if they already have an account, log in immediately after opening the application. The user should be able to produce drawings if they skip this stage, but they won't be able to save them. The user's email address will serve as the account's unique identification on the blogging application.

2. **Save:** The software allows logged-in users to store their work.

3. **CRUD:** The logged-in user can carry out CRUD operations on both his profile and his saved work.

5. **Dashboard:** The user can execute CRUD operations while viewing his profile and all of his created projects in one location.

1.4.2. Non-Functional Requirements

1. **Privacy:** Only logged in user can perform CRUD operation on their profile data and on saved work, no one else can access those details.

2. **Robustness:** In case the user's system crashes, a backup of his saved work must be stored on remote database servers to enable recoverability.

3. **Performance:** The application must be lightweight and must loads the painting app instantly.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
Dashboard	View all saved works and profile	User
Toolbar	Display all tools	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.

Demo link: [Project sample link](#)

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.

Demo link: [Linkedin post sample link](#)