Project Title	**College Event Management**
Technologies	MERN
Domain	Education
Project Level	Difficult

Table

Table

1. Problem Statement:

The College Event Management System is used to maintain information about various college activities, such as social fests, specialty fests, college day celebrations, seminars, project expos, and courses, etc. It provides information on events, maintains student involvement, sports branch facts, and moreover provides college triumphs. In the previous paradigm, all the data had to be visible in a physical document. It takes a lot of time and effort to go at certain information simultaneously when looking through any data; it is difficult to even consider accessing it. Therefore, in order to solve this problem, a web application can be used to make the process easier, safer, and less prone to errors. Increased data effectiveness will be achieved through this framework.

2. Features:

- Registration
- Login
- Add event details
- Update/Delete event details
- Add/Delete departments
- View/Search event details
- View/Search departments
- Generate event timetables
- Generate event performance graphs
- View event performance graphs
- Logout

2.1. Report Generation:

- It generates the report on Event concluded.
- It provides filter report on events and department wise
- You can easily export PDF for the all event reports.

3. Project Evaluation metrics:

3.1. 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 similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so 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.

3.2. 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.

3.3. 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.

3.4. Deployment:

Deploy the application on your preferred service.

3.5. Solutions Design:

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

3.6. System Architecture:

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

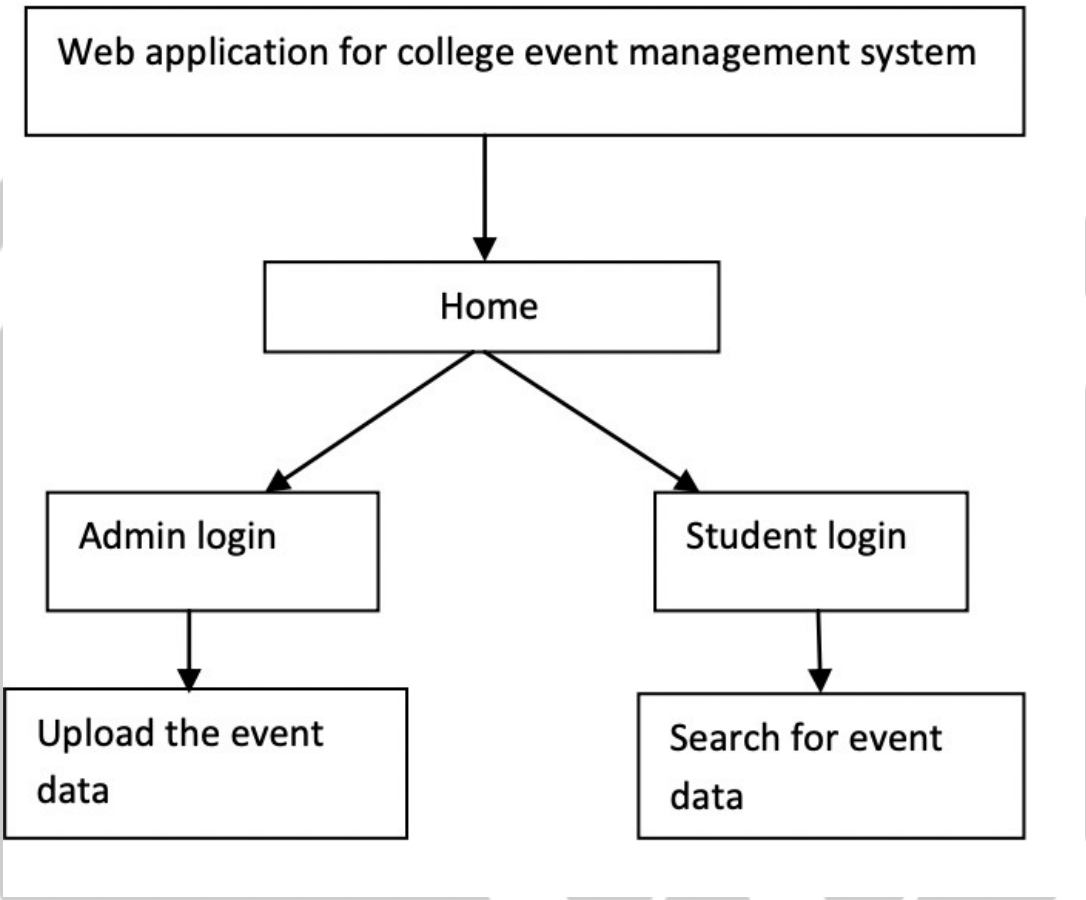
3.7. Optimization of solutions:

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

4. Submission requirements:

4.1. 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

4.2. 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

4.3. 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

4.4. Wireframe:

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

Demo link: Wireframe Document Link

4.5. 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

4.6. 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

4.7. Project demo video:

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