STUDENT NAME : VASANTHAKUMAR G

PROJECT ID : 11

SEAT NO: 225

PROJECT TITLE : EVENT MANAGEMENT

COMPONENTS	TECH STACK
Front End	Angular(Js Framework)
Back End	Express.js(Web framework for Node.js) Node.js(Javascript runtime environment)
Database	MongoDB(NOSQL Database)
API	REST ful API / GraphQL APIs

ROBLEM STATEMENT:

The current process of managing these events is often cumbersome and inefficient due to the reliance on manual, paper-based forms and workflows.

- **Manual Form Filling and Submission:** Faculty members are required to fill out various event-related forms manually.
- **Inefficient Data Management:** Maintaining hard copies of forms leads to difficulties in data retrieval, storage, and management. The IQAC team, responsible for approving and managing events, faces challenges in tracking the status and history of submissions.
- **Lack of Real-time Updates:** Faculty members do not have access to real-time updates on the status of their submissions. Manual communication methods (such as phone calls or emails) are used to check the status, leading to inefficiencies.
- **Approval and Review Process:** The IQAC team manually reviews and approves the forms, which is time-consuming. There is no streamlined workflow for reviewing and approving event requisitions and related forms.

SCOPE OF THE PROJECT:

A web-based Event Management system that automates the event management process, making it easier for faculty to submit event-related forms and for the IQAC team to review and manage these submissions.

SYSTEM OVERVIEW:

1. **Admin:**

Admins in the Event Management System can manage user roles, create, view, edit, and delete events, and oversee form submissions by approving or rejecting them. They can also download forms and generate detailed reports summarizing event details and form statuses, enhancing coordination and documentation of event-related activities.

2. **Faculty:**

Faculty members in the Event Management System can view details of all upcoming, ongoing, and past events, allowing them to stay informed and manage their participation. They can fill out and submit various necessary forms for event requirements, such as those for audio equipment,

accessories, photography, venue setup, accommodations, and transportation. These submissions are done online, streamlining the process and reducing the need for physical paperwork. Faculty can also track the status of their submitted forms, receiving notifications about approvals or rejections, and any necessary feedback from the admin. This system enhances their ability to organize and participate in events efficiently and effectively.

FUNCTIONAL FEATURES:

1. **User Roles and Permissions:**

   Admins can add, remove, and manage faculty users and can assign specific roles and permissions to users, controlling their access to different parts of the system.

2. **Event Management:**

   Event Creation: Admins can create new events by entering details such as event name, date, time, venue, and organizer information.

   Event Viewing: Both admins and faculty can view a comprehensive list of all events, including upcoming, ongoing, and past events.

   Event Editing: Admins can update event details as needed, ensuring all information is current and accurate.

   Event Deletion: Admins can remove events that are no longer necessary or have been canceled.

3. **Form Management:**

   Form Submission: Faculty can fill out and submit various forms required for event organization, such as audio, accessories, photography, venue setup, accommodations, and transportation.

   Form Viewing: Admins can view all submitted forms to oversee event requirements.

   Form Approval/Rejection: Admins can approve or reject submitted forms, providing necessary feedback and reasons for rejection if applicable.

   Form Downloading: Admins can download forms in PDF or DOC format for offline use or record-keeping.

4. **Report Generation:**

   Generate Reports: Admins can generate detailed reports summarizing event details, form submissions, and approval statuses.

Export Data: Reports can be exported in various formats, such as Excel or PDF, for further analysis or presentation.

5. **Secure Storage for faculty submissions:**

Upon submission, faculty responses are securely stored in the database, ensuring data integrity and confidentiality.

NON-FUNCTIONAL REQUIREMENTS:

- **Security:**

The system should adhere to industry-standard security protocols to safeguard sensitive student data and maintain confidentiality.

- **Performance:**

The platform should be capable of handling concurrent user interactions and survey submissions without significant degradation in performance.

Response times for loading surveys, submitting responses, and generating reports should be optimized to enhance user experience.

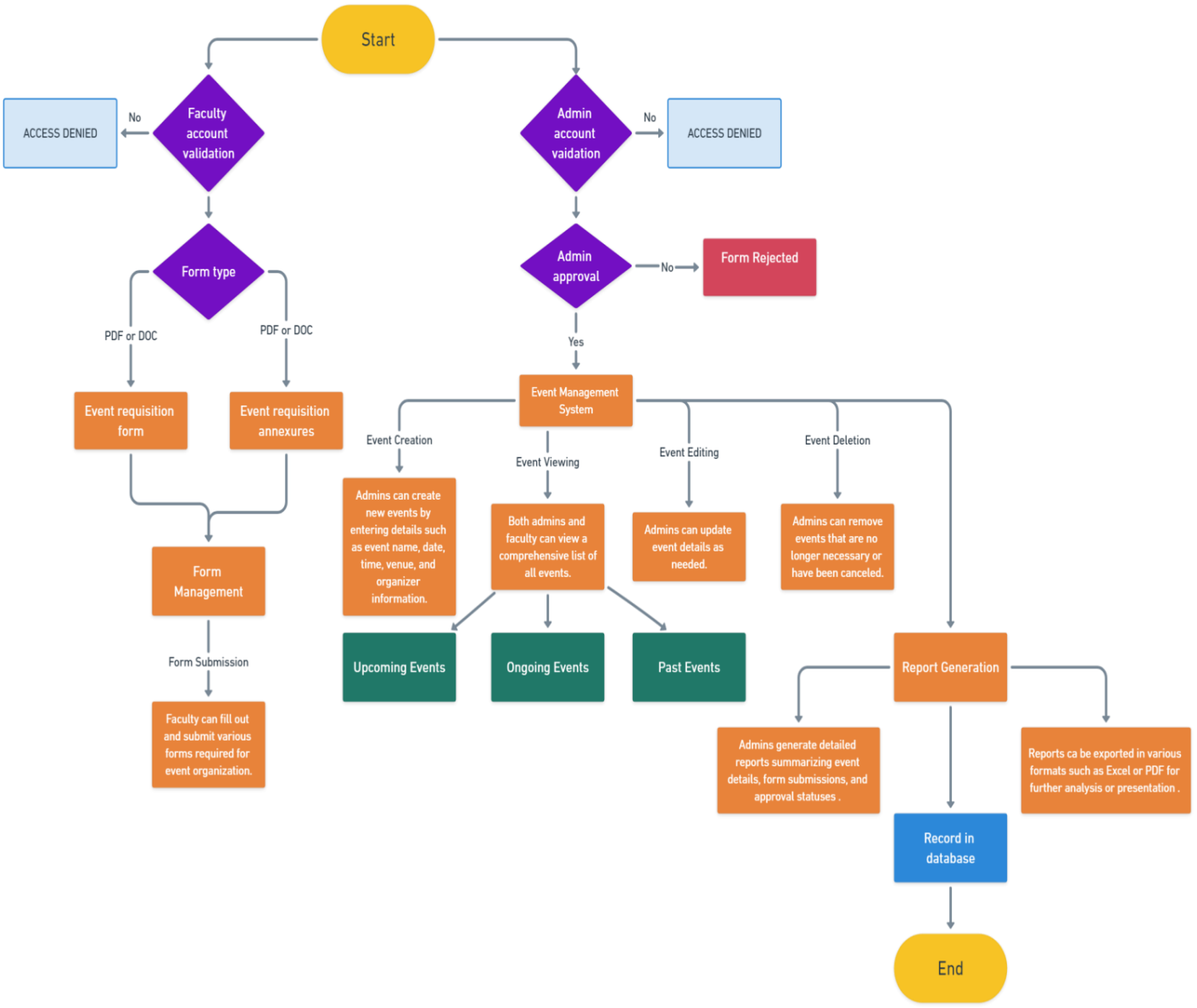
- **Scalability:**

Scalability measures should ensure that the platform remains responsive and efficient, even during periods of high usage or growth.

- **Accessibility:**

The platform should be accessible across various devices and web browsers to accommodate the diverse needs of users.

FLOW CHART:


Start
Faculty account validation
No
ACCESS DENIED
Admin account vaidation
No
ACCESS DENIED
Form type
PDF or DOC
PDF or DOC
Event requisition form
Event requisition annexures
Form Management
Form Submission
Faculty can fill out and submit various forms required for event organization.
Admin approval
No
Form Rejected
Yes
Event Management System
Event Creation
Admins can create new events by entering details such as event name, date, time, venue, and organizer information.
Event Viewing
Both admins and faculty can view a comprehensive list of all events.
Upcoming Events
Ongoing Events
Past Events
Event Editing
Admins can update event details as needed.
Event Deletion
Admins can remove events that are no longer necessary or have been canceled.
Report Generation
Admins generate detailed reports summarizing event details, form submissions, and approval statuses .
Reports ca be exported in various formats such as Excel or PDF for further analysis or presentation .
Record in database
End