



**Student Name:** Shruthi N

**Seat No:** 152

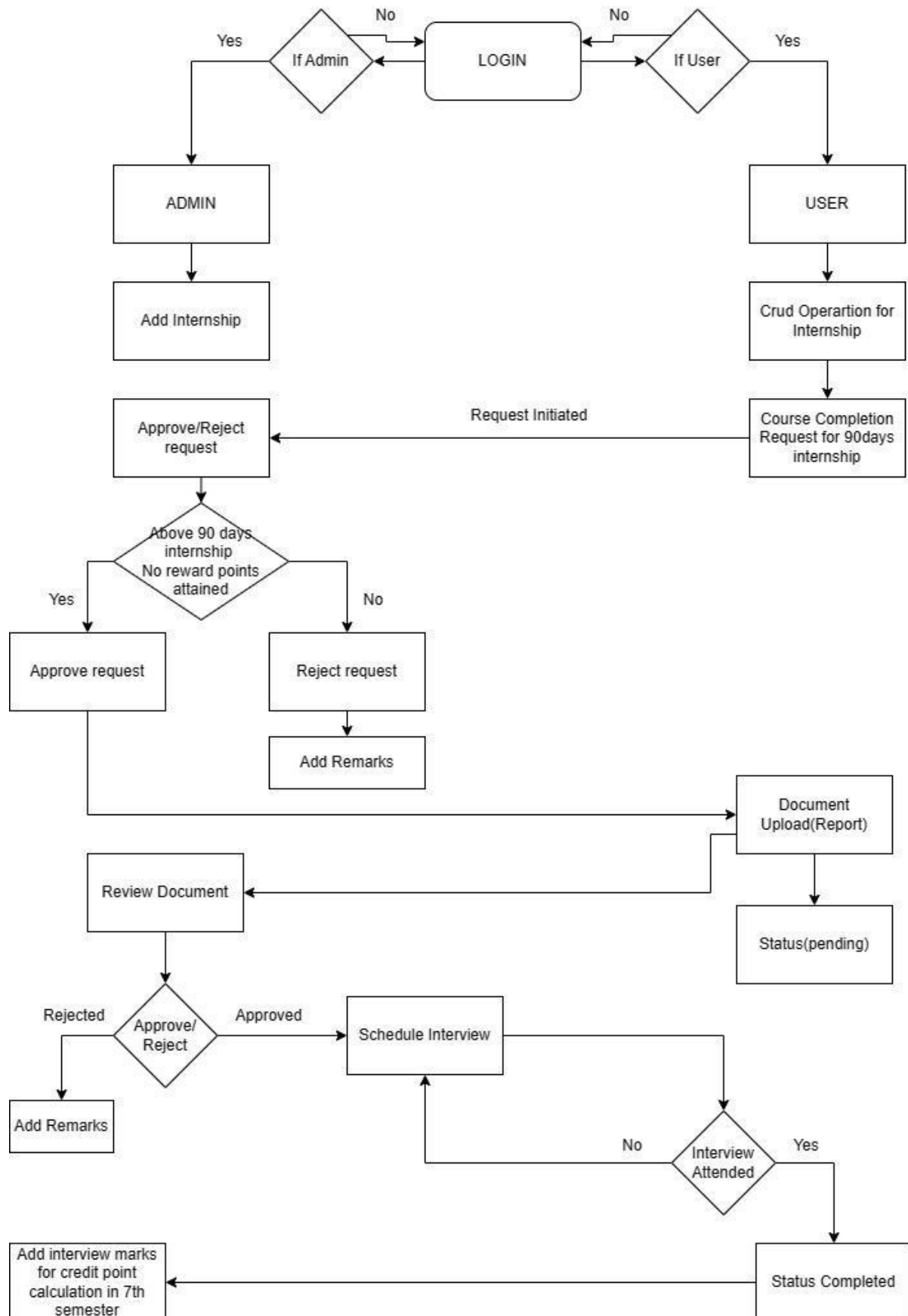
**Project ID:** 16

**Project title:** Internship course exemption

**Technical Components:**

Component	Tech Stack
Backend	Express.js(Web framework for Node.js) Node.js(Javascript runtime environment)
Frontend	Angular JS
Database	MongoDB(NOSQL Database)
API	REST Ful API / GraphQL APIs

## FLOWCHART:



## **For Users:**

### **User Login:**

- Users access the system through a login interface with their credentials.

### **Course Completion Request:**

- Users initiate a course completion request, indicating their intent to complete the 6-week course.

### **Submit Request & Document:**

- Users submit their request along with any necessary documentation, such as certificates or completion records from the course.

### **Review Document:**

- Admins review the submitted documents for completeness and relevance to the course requirements.

### **Approval/Rejection:**

- If approved, the admin schedules an interview.
- If rejected, feedback is provided to the user for improvements.

### **Schedule Interview:**

- An interview is scheduled for the user to assess their understanding and knowledge gained from the course.

### **Interview:**

- The user participates in the interview conducted by admins.

### **Outcome:**

- If the user passes the interview, they are granted course completion.
- If additional requirements are identified, the user may need to fulfill them and repeat the process.

## **For Admins:**

### **Admin Login:**

- Admins access the system through a login interface with their credentials.

### **Review Document:**

- Admins review the documents submitted by users to determine their eligibility for course completion.

### **Approval/Rejection:**

- Admins can approve the request and proceed to schedule an interview.
- If rejected, admins provide feedback to users for improvement.

### **Schedule Interview:**

- Admins schedule an interview with the user to assess their knowledge and understanding of the course material.

### **Interview:**

- Admins conduct the interview to evaluate the user's comprehension and proficiency in the course subject matter.

### **Outcome:**

- Based on the interview results, admins decide whether to grant course completion or request additional requirements from the user.

To break down the frontend and backend requirements needed for the described course completion process using the MEVN stack (MongoDB, Express.js, Vue.js, Node.js), we need to consider both the user-facing interface (frontend) and the server-side functionality (backend).

Here's a breakdown:

### Frontend Requirements:

#### 1. User Login Interface:

- A login page where users can input their credentials (username/email and password) to access the system.

#### 2. Course Completion Request Form:

- A form or interface where users can initiate a course completion request by providing relevant details such as their intent to complete the 6-week course.

#### 3. Document Submission:

- A feature allowing users to upload necessary documentation, such as certificates or completion records from the course, along with their completion request.

#### 4. Feedback Display:

- If a user's request is rejected, a mechanism to display feedback provided by admins for improvement.

#### 5. Interview Scheduling:

- A scheduling interface where users can view and confirm interview appointments set by admins.

#### 6. Interview Participation:

- An interface for users to participate in interviews conducted by admins.

#### 7. Outcome Notification:

- Upon completion of the interview process, a notification mechanism to inform users of the outcome (course completion or additional requirements).

### Backend Requirements:

#### 1. User Authentication:

- Backend functionality to handle user authentication and authorization for accessing the system and initiating course completion requests.

#### 2. Document Management:

- Functionality to manage document submissions from users, including storage, retrieval, and validation.

3. Request Processing:

- Backend logic to process course completion requests submitted by users, including validation and forwarding to admins for review.

4. Admin Interface:

- Backend support for admin accounts with appropriate permissions to review usersubmissions, schedule interviews, and provide feedback.

5. Interview Management:

- Functionality to manage interview scheduling, including date/time selection and notification mechanisms.

6. Outcome Decision Logic:

- Backend logic to evaluate interview results and make decisions on granting course completion or requesting additional requirements from users.

7. Notification System:

- Integration with a notification system to inform users of interview schedules and outcomes.

8. Logging and Monitoring:

- Implementation of logging and monitoring mechanisms to track user actions, system events, and potential errors for debugging and analysis purposes.

These frontend and backend requirements form the foundation for building a comprehensive system to facilitate the course completion process outlined in the scenario, utilizing the MEVN stack.