**BANNARI AMMAN INSTITUTE OF TECHNOLOGY**

**An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade**

**Sathyamangalam - 638401 Erode District, Tamil Nadu, India**

STUDENT NAME: HARISH KUMAR A

SEAT NO : 281

PROJECT ID : 18

PROJECT TITLE : COURSE REGISTRATION

|  |  |
| --- | --- |
| **Component** | **Tech Stack** |
| Frontend | React JS |
| Backend | Node JS |
| Database | MongoDB(NOSQL Database) |
| API | OpenAPI |

**Technical Components:**

**Implementation Timeline:**

,,,,

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Deadline** | **Status** | **Notes** |
| Stage 1 | 25.07.2024 | Completed | Planning and Requirement gathering |
| Stage 2 |  | Under review | Design and Prototyping |
| Stage 3 |  | Not started | DB Designing |
| Stage 4 |  | Not started | Backend Implementation |
| Stage 5 |  | Not started | Testing & Implementation |

**INTRODUCTION:**

This document outlines the design and functionalities of the COURSE REGISTRATION, a web application developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) to address the current inefficiencies in course registration and management at our college.

**PROBLEM STATEMENT:**

Our college currently relies on a decentralized system for course registration, primarily

through email communication and Google Forms. This approach leads to several challenges:

* User Authentication: Students and administrators should be able to log in securely with their credentials.
* User Regulations: Students should select regulations.
* Department and semesters: Student should select respective department and semesters.
* Course: Courses are available based on the department and semesters.

**PROJECT-FLOW:**

1. **User Authentication:**

**Login Page**:

* + - Students log in using their email and password.
  + Implement form validation for email and password fields.
  + Send login request to the backend API for authentication.

**Registration Page (Optional)**:

* If self-registration is allowed, provide a form for students to register.
* Fields include email, password, name, and other necessary information.

1. **Dashboard:**

* Student Dashboard
* **Regulation Selection**
* Department Selection
* Semester Selection
* **Available Courses**
* Course Registration

**3. Course Management (Admin):**

* **Admin Dashboard** (remains the same):
  + Display all courses, view eligible students, and manage course details.
* **Course Management** (remains the same):
  + Add, edit, delete courses.
  + Manage course details like course code, name, credits, and prerequisites.

**4. User Profile:**

* **Profile Page**:
  + Display user profile details such as name, email, department, regulation, and semester.
  + Provide an option to edit profile details if necessary.
* **Update Profile**:
  + Implement form validation and send an update request to the backend API.
  + Ensure data consistency and security.

**5. Database Schema Design:**

* **Users**: Collection for storing user details (students and admins).
* **Courses**: Collection for storing course details.
* **Registrations**: Collection for storing registration details (linking students to courses).

**STAGES OF DEVELOPMENT:**

The development process will involve several stages:

1. **Planning and Requirement Gathering:**

• Define project goals, objectives, and scope.

• Conduct stakeholder interviews to gather requirements and understand user needs.

• Create a detailed project plan outlining timelines, milestones, and deliverables.

• Identify key features, functionalities, and user roles based on requirements.

1. **Design and Prototyping:**

• Develop UI/UX mock ups and wireframes to visualize the application's layout and navigation.

• Design the user interface using design principles such as responsiveness, accessibility, and aesthetics.

• Review and refine prototypes based on feedback from stakeholders and usability testing.

1. **Database Design:**

• Define the MongoDB database schema including collections, documents, and relationships.

• Determine data storage requirements, indexing strategies, and data access patterns.

• Optimize the database design for efficient data retrieval, storage, and scalability.

1. **Backend Development:**

• Set up the Node.js environment and install necessary packages/modules.

• Develop server-side logic using Express.js for routing, middleware, and API development.

• Implement user authentication using JWT (JSON Web Tokens) for secure access control.

• Create RESTful APIs for CRUD (Create, Read, Update, Delete) operations on skills, users, sessions, etc.

• Integrate the backend with the MongoDB database for data storage and retrieval.

1. **Frontend Development:**

• Initialize a Vue.js project structure and configure necessary plugins/libraries.

• Develop frontend components, views, and layouts using React.js, HTML, and CSS.

• Implement state management using React for managing application-wide state and data.

• Ensure responsiveness and cross-browser compatibility for a seamless user experience.

• Integrate frontend components with backend APIs for data fetching and manipulation.

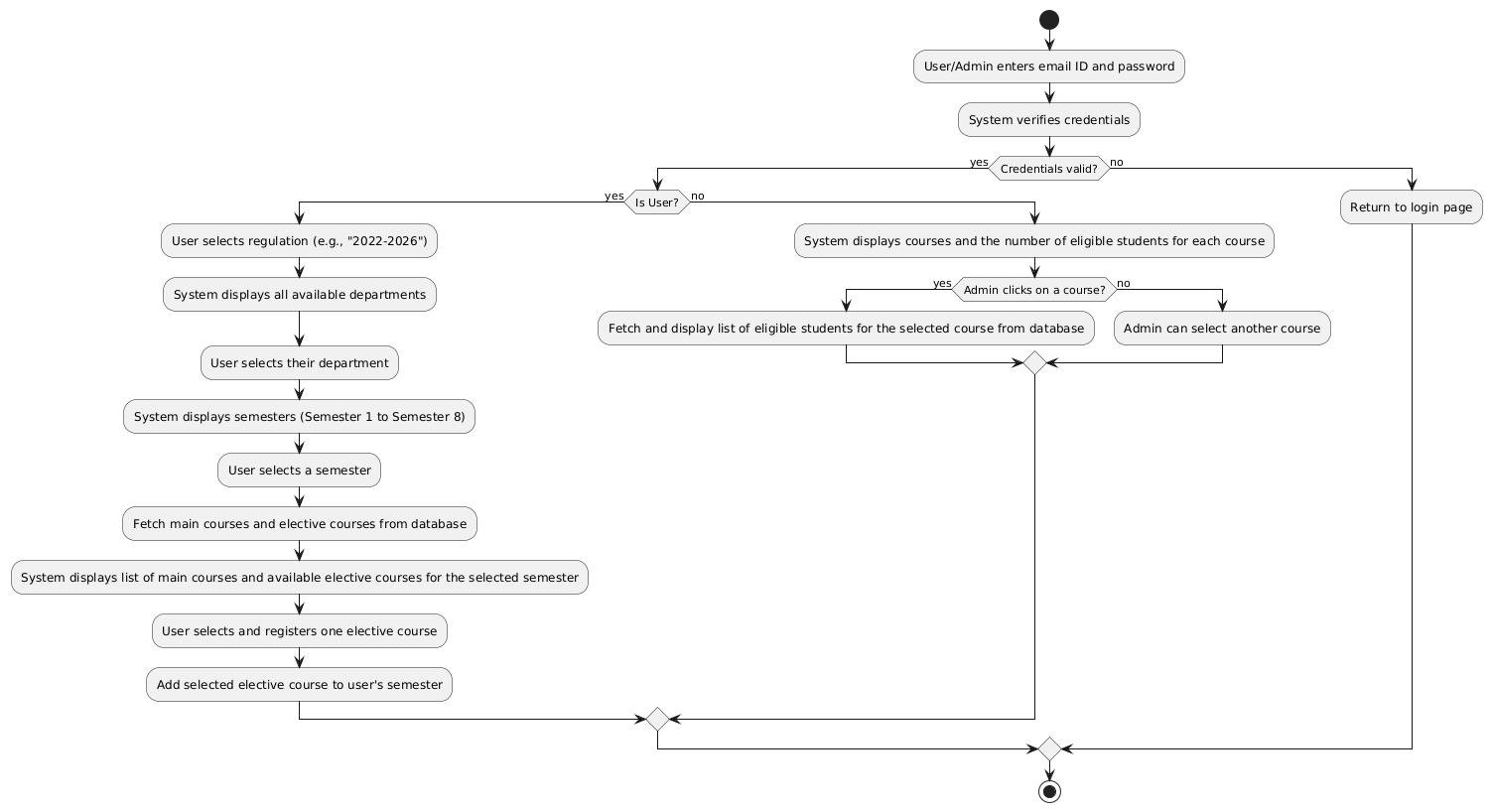
1. **User Acceptance Testing (UAT):**

• Conduct UAT sessions with representative users including students, faculty, and administrators.

• Validate system functionalities, usability, and performance against predefined acceptance criteria.

• Gather feedback from users to identify bugs, usability issues, and areas for improvement.

**FLOWCHART:**

****