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 : ASHISH SAKTHI D
Seat No **:** 180
Project ID **:** 20
Project Title **:** Online Course System
Domain **:** XR Studio

Technical Components :

Component	Stack
Front End	Angular
Back End	Express.js , Node.js
Database	MongoDB
API	Rest API

Problem Statement :

The problem is the absence of a comprehensive, user-friendly platform for individuals seeking task-based online courses.

Current Scenario : Current offerings lack features such as user authentication, progress tracking, and interactive elements like task submission and instructor chat boards. This results in fragmented user experiences and limited engagement.

Solution : There is a need for a responsive and scalable web application that integrates user authentication, course creation and management, user reviews and ratings, output visualization, progress tracking, task submission with video capabilities, and an instructor chat board. By addressing these needs, the platform aims to provide a seamless and interactive learning environment for users, fostering greater engagement and facilitating effective learning outcomes.

Work Flow:

1. Requirement Gathering

User Research: Conduct surveys and interviews to understand the needs of students, instructors, and administrators. Focus on features like course creation, enrollment, reviews, and payment processing.

User Stories & Features:
- Students can search, enroll, and track progress in courses.
- Instructors can create, edit, and manage courses.
- Administrators can manage users, courses, and reviews.

Technical Requirements:
- Define specific requirements for MongoDB, Express.js, Angular, and Node.js.

2. Design Phase

User Interface Design:
- Create wireframes and prototypes focusing on user-friendly navigation and accessibility.
- Design using Adobe XD or Figma with a focus on Material Design principles for Angular components.

Database Design:
- Design a MongoDB schema to manage users, courses, reviews, payments, and more.
- Optimize for querying and indexing, focusing on scalability.

Architecture Design:
- Choose compatible versions of Angular, Node.js, and Express.js for optimal performance.
- Define a modular project structure to ensure maintainability and scalability.

3. Development

Frontend Development:
- Implement a responsive UI using Angular, HTML, CSS, and TypeScript.
- Utilize Angular CLI for creating and managing components and services.
- Implement features like course browsing, dashboards, and payment interfaces.

Backend Development:
- Develop RESTful APIs using Express.js for operations like course management, user authentication, and review management.
- Implement user authentication using Passport.js with strategies like JWT for secure login.

Database Implementation:
- Set up MongoDB with Mongoose ORM to simplify interactions and schema design.
- Implement data validation and schema constraints.

Integration:
- Use Angular's HttpClient module to handle communication between the frontend and backend, ensuring smooth data exchange.

4. Testing

Unit Testing:
- Write unit tests for Angular components and services using Jasmine and Karma.
- Ensure test coverage for all critical paths in the application.

Integration Testing:
- Test API endpoints and the integration of frontend and backend using tools like Postman or Newman.

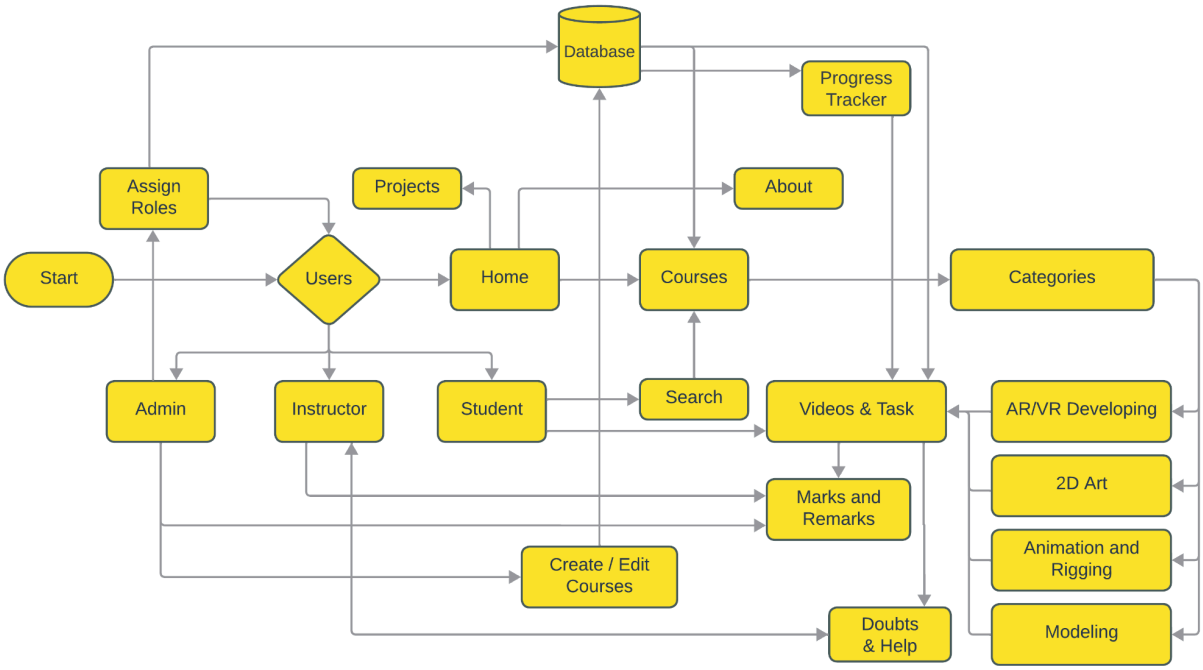
End-to-End Testing:
- Use Protractor to automate end-to-end tests, ensuring the entire flow works as expected.

5. Deployment

Environment Setup:
- Set up staging and production environments using AWS or Heroku.
- Configure environment variables and settings for different deployment stages.

Deployment Process:
- Use Docker for containerization, ensuring consistent environments across development, testing, and production.
- Implement Kubernetes for orchestration, managing scaling, and deployment across clusters.

FlowChart:



ER Diagram: