WOC 7.0 NodeJs

aDApt—A Collaborative plartform for students

Overview of the project:

The project involves creating a collaborative platform for students using **NodeJs**, **HTML**, **CSS**, **JS** and **React**. The platform features a shared resource library where students can upload and **access study materials**, a **Q&A manager** for posting and resolving academic **doubts**, a directory of important email IDs and websites, and a **Lost & Found manager** for tracking misplaced items. The system includes an admin panel for managing categories and requests, with potential for direct messaging in the Q&A manager.

Why This Project?

Practical Application:

- Builds a real-world, student-centric platform addressing common needs like resource sharing and communication.

Skill Development:

- Hands-on experience with web technologies and frameworks.
- Strengthens knowledge in database design, API development, and user interface design.

Community Impact:

- Encourages collaboration among students and streamlines common processes like managing doubts or lost items.

Problem-Solving:

- Opportunity to design innovative solutions, such as automated category management and item tracking.

Concepts covered in this project

- **Backend Framework (e.g., Node.js)**:
  - Asynchronous and event-driven, ideal for handling multiple user interactions simultaneously.
  - Large ecosystem of npm libraries for rapid development and scalability.
- **Database (e.g., MongoDB or PostgreSQL)**:
  - NoSQL/Relational flexibility for managing diverse datasets like resources, Q&A, and lost & found items.
- **Express.js:**
  - Building RESTful APIs
  - Middleware and routing
- **Socket IO:**
  - Instant updates when new resources are uploaded to the library.
  - Live updates on Lost & Found items when added or marked as found.
  - Real-time communication between students and admins for requests.

Overall Development Pipeline

1. **Planning and Requirement Analysis**: Identify core features and define the tech stack.
2. **UI/UX Design**: Create wireframes for intuitive and responsive user interfaces.
3. **Frontend Development**: Build dynamic and interactive components.
4. **Backend Development**: Set up APIs and database models for handling resources, Q&A, and user data.
5. **Integration**: Connect frontend and backend for seamless functionality.
6. **Deployment**: Host the platform on a cloud service (e.g., AWS, Vercel) for public access.

Tools and Technologies Required

1. **Frontend**: HTML, CSS, JavaScript.
2. **Backend**: Node.js, Express.js.
3. **Database**: MongoDB or PostgreSQL.
4. **Version Control**: Git and GitHub.
5. **Deployment**: AWS, Vercel, or Netlify.
6. **Package Manager**: npm.

Learning Resources:

- Node installation - https://nodejs.org/en/
- Node basics - https://www.youtube.com/watch?v=BLl32FvcdVM
- Express Basics - https://www.youtube.com/watch?v=7H_QH9nipNs
- HTML/CSS/JS basics - https://www.w3schools.com/ and https://javascript.info/
- Node, Express and Mongo all in one - https://www.youtube.com/watch?v=W1Kttu53qTg
- Learn Socket IO - https://socket.io/docs/v4/
- Learn React-Chai aur react | with projects - YouTube
- Learn MongoDB-Master MONGODB in ONE VIDEO: Beginner to Advanced Course For Beginners in Hindi | MPrashant - YouTube

Phase-wise division of the project:

Phase 0: Preparation and Basics

1. Get familiar with HTML, CSS, JavaScript, and Node.js.
2. The React code will be provided from the beginning, so there is no need to focus on learning React.
3. Understand the basics of RESTful APIs and JSON for backend communication.
4. Learn Git and GitHub for version control and collaboration.
5. Explore MongoDB or PostgreSQL for database management and queries.

Phase 1: Home Page and Authentication (Student, Admin Login)

1. Design a responsive home page layout with navigation menus.
2. Set up authentication with role-based access for students and admins.
3. Implement secure login, registration, and password hashing.
4. Add email verification for new student registrations.
5. Important Email IDs and Website Library:
    a. Create a list of important email addresses and websites for academic and admin use.
    b. Add a search function for quick access to information.

Phase 2: Core Functionality

1. Shared Resource Library :
    - Students can upload and download study materials (notes, assignments, etc.).
    - Include a search option to easily find resources.
2. Q&A Manager:
    - Students can ask questions.
    - Users can upload answers to questions.
    - Sort questions into categories for easy browsing.
    - Admin panel to manage categories and allow students to request new ones.
3. Lost & Found Manager:
    - Add categories like CEP, LT, Canteen, etc., for lost items.
    - Students can upload photos of lost items and mark them as "found" or "not found."
    - Automatically delete photos marked as "found" after a set time.

Project Setup Guidelines and Installation Instructions:

- Download NodeJs from https://nodejs.org/en/download
- Install npm package Socket IO https://www.npmjs.com/package/socket.io

Mentors:

- **Devang Vaghani**
  - Email: 202201208@daiict.ac.in
  - Contact: +91 70692 90029

- **Vashisht Purani**
  - Email: 202412077@daiict.ac.in
  - Contact: +91 84909 95868

- **Brinda Sorathiya**
  - Email: 202301182@daiict.ac.in
  - Contact: +91 82007 21981

- **Nidhi Dodiya**
  - Email: 202303009@daiict.ac.in
  - Contact: +91 76219 83270