

Assignment 02: Full Stack CRUD Application – MERN

Due Date: Sunday, 10-Aug-2025

Objective:

Develop a full stack web application using **React** for the frontend and **Node.js with Express and MongoDB** for the backend. The application must implement full **Create, Read, Update, and Delete (CRUD)** functionality for a chosen data model (e.g., product, portfolio project, book, employee, or any other etc.).

Project Requirements

Core Functionalities

1. Add New Item
 2. View All Items
 3. Edit/Update Item
 4. Delete Item
 5. View Details of a Selected Item (Dynamic Read by ID)
-

Technical Specifications

Frontend – React

- Component-based architecture
- React Router DOM for **SPA routing**
 - /items – List view
 - /items/:id – **Dynamic detail view** ✓
 - /add-item – Form to add
 - /edit-item/:id – Edit form
- useState and useEffect for state & lifecycle

- Axios for API calls
- Tailwind CSS for styling
- Client-side validation of forms

Backend – Node.js + Express + MongoDB

- RESTful API with the following routes:
 - GET /api/items – Get all items
 - GET /api/items/:id
 - POST /api/items – Add new item
 - PUT /api/items/:id – Update item
 - DELETE /api/items/:id – Delete item
 - Use Mongoose for MongoDB schema & model
 - Express middleware (CORS, JSON parser)
 - Database: MongoDB Atlas (preferred) or local MongoDB
-

Suggested Folder Structure

React Frontend

```
/src
├── /components
│   ├── /layouts
│   │   ├── Navbar.jsx
│   │   └── Footer.jsx
│   └── /views
│       ├── HomeView.jsx
│       ├── ItemListView.jsx
│       ├── ItemFormView.jsx
│       └── ItemDetailView.jsx
├── /pages
│   ├── HomePage.jsx
│   ├── AddItemPage.jsx
│   ├── EditItemPage.jsx
│   └── ViewItemsPage.jsx
├── App.jsx
└── main.jsx
```

Node Backend

```
├── /models
│   └── itemModel.js
├── /controllers
│   └── itemController.js
├── /routes
│   └── itemRoutes.js
└── index.js
```

Submission Requirements

Submit a **single PDF file** that includes:

1. GitHub Repository Link

- Separate frontend and backend repos
- Code should follow clean structure
- In case not uploaded on GitHub, then upload your code to Google Drive, and paste drive link in pdf file (**Recommended is GitHub Repository Link**)

2. Screenshots

- Full UI pages: List, Add, Edit, Detail View
- Full VS Code Screen showing its file folder structure details (**both for React and Node js applications – separately**)

3. Source Code Snippets

- **React**
 1. All API Calls
 2. One React component (e.g., ItemDetails.jsx)
 3. Dynamic Routing
- **Node**
 1. All controllers
 2. Models
 3. Routes / dynamic routes