

You've been asked to build and deploy a basic task management (To-Do List) app using a specific tech stack called MERN, which means using tools for both the frontend (what users see) and backend (the logic and data storage). The app should let users add, view, edit, and delete tasks. You'll use a free online database service (MongoDB), and once the app is working, you need to put it live using platforms like Vercel or GitHub Pages. Lastly, to show that your app works well, you need to write two simple tests for the user interface and two for the backend functionality.

✓ Project Objective

Build and deploy a "To-Do List" application

This should be a full-stack web app allowing users to manage tasks. The core operations should follow CRUD principles.

🔧 Technology Stack (MERN Stack)

- **MongoDB** – Database for storing tasks
- **Express.js** – Backend framework for handling API requests
- **React.js** – Frontend library to build the user interface
- **Node.js** – Runtime environment for the backend server

This stack will allow you to create a modern web app with a reactive UI and a RESTful API.

🔄 Functionality: CRUD

You must implement **Create, Read, Update, Delete** functionality:

- **Create** – Add new tasks
- **Read** – Fetch/display all tasks
- **Update** – Edit existing tasks (e.g., mark as complete)

- **Delete** – Remove tasks

The frontend should consume REST APIs created in Express/Node, which in turn interact with MongoDB.

Database: MongoDB Free Cluster

Use **MongoDB Atlas**:

- Set up a **free-tier cluster**
 - Create a **tasks** collection
 - Connect it securely to your Node.js/Express backend
-

Deployment

You must **deploy** the application:

- **Frontend:**
 - Can be deployed to **Vercel** (preferred for React apps)
 - Optionally GitHub Pages (though Vercel is better for full-stack projects)
- **Backend:**
 - Can be deployed to **Render**, **Railway**, or **Cyclic** (free Node.js hosting options)

If using Vercel, consider separating front and back into two projects or host the API as serverless functions.

Testing: 2 Test Cases Each (Frontend & Backend)

Write at least:

- **2 frontend test cases** (e.g., using **Jest + React Testing Library**):
 - Example: Add a task and check if it appears
 - Example: Mark a task as complete and check status update
- **2 backend test cases** (e.g., using **Jest + Supertest** for Express routes):
 - Example: Test **POST /tasks** creates a task
 - Example: Test **DELETE /tasks/:id** deletes the task

These tests help prove that both UI and API logic are working correctly.

Summary

You are expected to:

1. Build a To-Do list app using **MERN stack**
2. Implement **CRUD** features
3. Use a **MongoDB Atlas Free Cluster**
4. Deploy the app (frontend/backend) on **Vercel or GitHub Pages**
5. Include **at least 2 test cases each** for frontend and backend