

Task 36: File Upload with Multer

Problem Statement:

Your task is to create an Express.js application that handles file uploads using Multer and stores metadata in a MongoDB database using Mongoose. Additionally, create a React frontend to upload files and display metadata.

Problem Description:

In this task, you will set up a MongoDB database connection using Mongoose and configure Multer for file uploads. You will implement routes to handle file uploads and store metadata in the database. You will also create a React frontend to upload files and display metadata.

Your Express.js application should include the following features:

1. Connect to MongoDB: Use Mongoose to connect to a MongoDB database.
2. Configure Multer: Set up Multer for handling file uploads.
3. Define a Schema: Create a schema for storing file metadata with fields for filename and path.
4. Create Routes:
 - POST /upload: Handle file uploads and store metadata in the database.
 - GET /files: Retrieve and display file metadata.

Your React application should include the following features:

1. File Upload Form: A form to upload files.
2. Submit Data: Functionality to send a POST request to upload files.
3. Display Data: Functionality to fetch and display file metadata.

Submission Guidelines:

- 1. Deploy both frontend and backend on vercel
- 2. Upload both frontend and backend files on github.
- make a readme file in github frontend repo and paste the links of deployed frontend , deployed backend and backend github repo.
- **Add ScreenShots from mongodb in Readme file.**
- Submit the Frontend repo link in Dashboard
- Example frontend repo link:
<https://github.com/aakashk1905/DummyFrontEnd>