Technical Task: Implement OCR in a MERN Stack Application

Objective:
Integrate OCR functionality into an existing MERN application, allowing users to upload images containing handwritten text and extract the text content.

Requirements:

1. Backend Development:

- Create an endpoint in the Express.js application for receiving image uploads.
- Use a Node.js OCR library like Tesseract.js or node-tesseract to process the uploaded images and extract handwritten text.
- Implement validation to ensure that the uploaded file is an image (e.g., PNG, JPEG) and handle potential errors.
- Store the extracted text or results in a MongoDB database for future retrieval.
- Design the API to receive and return data in JSON format.

2. Frontend Development (React):

- Create a new or integrate OCR functionality into an existing React component.
- Develop a user interface that allows users to upload images for OCR processing.
- Implement a file input field for selecting and uploading images.
- Use Axios or another HTTP client library to send the image to the backend OCR API.
- Display the extracted text results on the user interface.

3. Database Integration (MongoDB):

- Design a MongoDB schema to store the OCR results. A basic schema could include fields like `originalImage`, `extractedText`, `timestamp`, and a unique identifier for each record.
- Create MongoDB models and configure them to interact with the database from the Express application.

4. User Experience:

- Ensure a user-friendly interface with clear instructions and error handling.
- Provide feedback to users on the OCR process (e.g., "Processing..." while waiting for results).
- Allow users to view the extracted text and manage their OCR results.

5. Deployment:

- Deploy the MERN stack application, including the backend Express server and the React frontend, to a hosting platform of your choice which is free and share the link
- Set up the necessary environment variables and configurations for deployment.

Optional Enhancements:

- Implement user authentication and authorization to secure OCR functionality.
- Optimize OCR performance and accuracy.
- Implement real-time image processing and OCR using WebSockets.