Document Signing Application similar to DocuSign

Objective:
Create a web-based document signing application that allows users to upload a PDF, specify different fields in the doc (Name, Email, Digital signature, etc.), and download a consolidated PDF document with all the fields superimposed.

Detailed Requirements:
Frontend (React):

- React application for the frontend.
- Implement user authentication for secure access.
- Design a user-friendly interface with the following features: - Document upload form with the ability to upload PDF files. - Name / Email / Signature capture component. - Button to initiate the signing process. - A list or display area for the signed documents.
- Make API requests to the Node.js backend for document upload, signing, and retrieving signed documents.
- Implement error handling and display appropriate messages to the user.
- Deploy the React frontend to a hosting platform (e.g., Netlify, Vercel, or GitHub Pages).

Backend (Node.js):

- Create a Node.js application using the Express framework.
- Implement user authentication using a library of your choice (e.g., Passport.js, Firebase Authentication).
- Set up API routes for the following actions: - User registration and login. - Document upload (PDF files). - Signature capture and merging with the uploaded document. - Retrieving signed documents.
- Use a library like `pdf-lib` or `pdfkit` to generate signed PDF documents by merging the uploaded documents with the digital signatures.
- Implement signature validation for uploaded documents to ensure document integrity.
- Handle errors gracefully and provide clear error messages.
- Deploy the Node.js backend to a hosting service (e.g., Heroku, AWS, DigitalOcean).

Submission:

- The candidate should provide a link to the GitHub repository containing the code for the project.
- Include clear documentation on how to set up and run the application.
- During the interview, the candidate should demonstrate the application's functionality, including document upload, signing, and retrieval.
- Use any resources available.

Evaluation Criteria:

- Code quality, organization, and readability.
- Secure user authentication and data handling.
- Effective use of React for the frontend.
- API design and functionality on the Node.js backend.
- Document generation and signature verification.
- Error handling and user-friendly messages.
- Deployment of both frontend and backend.
- [illegible] unication and explanation of design decisions. This assignment tests the [illegible]'s ability to create a complete document signing application using Node.js and [illegible] focusing on both frontend and backend development, as well as secure user [illegible] tion and document handling.