**Task: PDF Editor Application with React.js and Apryse PDFTron Webviewer**

**Objective:**

Create a simple PDF editor web application using React.js and integrate the Apryse PDFTron Webviewer SDK. The application should allow users to load PDFs, .doc, and .docx files, edit text on these files, and draw shapes. Additionally, implement a basic authentication system using React Hook Form and local environment variables.

Time Limit: The test is expected to be completed within 2 hours, however you have 2 days to complete this provided task.

**Requirements:**

1. Create a new React.js app using create-react-app, using the LTS Node.js version.
2. All npm modules should be on the latest version.
3. Use the latest version of Apryse PDFTron Webviewer.
4. Implement a basic authentication system with a randomly generated token.
5. Use React Hook Form for the authentication form, which consists of an input field for the token, a submit button, proper validation, an error message, and a success message.  
   The PDFTron Webviewer should launch in demo mode, as licenses will not be provided.

**Features to Implement:**

Load PDFs, .doc, and .docx files using the Apryse PDFTron Webviewer API.

Enable text editing capabilities for PDFs, .doc, and .docx files using the Webviewer API.

Provide shape drawing functionalities on PDFs, .doc, and .docx files using the Webviewer API.

**Authentication Setup:**

Generate a random token and use it as the secret token for authentication.

Create a .env file in the root directory of the React.js app.

Add the generated token to the .env file with the following format:

REACT\_APP\_SECRET\_TOKEN=your\_generated\_token\_here

**Instructions:**

1. Set up the React.js app using create-react-app.
2. Integrate the Apryse PDFTron Webviewer SDK into the app.
3. Implement the PDF editing features: loading files, text editing, and shape drawing using the Webviewer API.
4. Create a React Hook Form for the authentication form with proper validation and messages.
5. Implement the authentication system using the random token and local environment variables.
6. Test the app to ensure all features are working correctly.
7. Create a detailed README.md file with the following information:
   1. Setup instructions: How to install dependencies and run the app locally.
   2. How to use the app: Instructions on loading files, editing text, and drawing shapes.
   3. How to test the app: Instructions to verify the authentication and all the features.

**Submission:**

Compress the entire project folder into a zip file, excluding the node\_modules directory and send the zip file to us