Documentation for Word-to-PDF Conversion Web App

This documentation provides an overview of the Word-to-PDF Conversion web application. The project leverages modern technologies, with the **frontend** deployed on **Vercel** and the **backend** hosted on **AWS EC2** using a **Dockerized FastAPI** service. The app provides a seamless user experience for converting Word documents into PDF format.

Features

- **User-Friendly Interface**: Intuitive frontend for uploading Word files and downloading converted PDFs.
- **High-Performance Backend**: Fast and reliable file conversion using FastAPI.
- **Scalable Deployment**: Deployed using Docker containers for scalability and portability.
- **Cross-Platform Support**: Accessible via web browsers on any device.

Tech Stack

Frontend

- **Framework**: React.js
- **Deployment Platform**: Vercel
- **Key Features**:
    - File upload interface.
    - Real-time status updates during conversion.
    - Responsive design for desktop and mobile users.

Backend

- **Framework**: FastAPI
- **Hosting**: AWS EC2 instance
- **Containerization**: Docker
- **Key Features**:
    - API endpoints for handling file upload and conversion.
    - Efficient processing of Word documents into PDF format.
    - Error handling for unsupported file formats.

System Architecture

1. User Interaction

- Users interact with the frontend deployed on **Vercel**, where they upload Word documents.
- The frontend sends the uploaded file to the backend via a REST API.

2. Backend Processing

- The backend, hosted on an **AWS EC2 instance**, receives the file through an API endpoint.
- The FastAPI service processes the Word document using a conversion library (e.g., `docx2pdf`, `pikepdf`, or similar tools).
- The converted PDF is saved temporarily and sent back to the frontend for download.

3. Deployment Workflow

- **Frontend**: Continuous Deployment on Vercel with GitHub integration.
- **Backend**: Dockerized FastAPI application deployed on an AWS EC2 instance for scalability and isolation.

API Endpoints

1. File Conversion

Endpoint: `/convert-to-pdf`
Method: `POST`
Description: Accepts a Word document (`.doc` or `.docx`) and returns a converted PDF.
Payload:

```
StreamingResponse(io.BytesIO(pdf_content),media_type="application/pdf")
```

Response:

- **Success (200)**: Returns the converted PDF file.
- **Error (400)**: Unsupported file type or processing error.

2. Encrypted File Conversion

Endpoint: `/convert-to-encrypted-pdf`
Method: `POST`
Description: Accepts a Word document (`.doc` or `.docx`) and returns a Encrypted converted PDF.
Payload:

```
StreamingResponse(io.BytesIO(pdf_content),media_type="application/pdf")
```

Response:

- **Success (200)**: Returns the converted PDF file.
- **Error (400)**: Unsupported file type or processing error.

Setup and Deployment

Frontend (Vercel)

1. Clone the frontend repository:

   ```
   bash
   Copy code
   git clone Saksham-21/Doc_to_PDF
   cd frontend
   ```

2. Install dependencies:

   ```
   bash
   Copy code
   npm install
   ```

3. Deploy on Vercel:
   - Connect the repository to Vercel.
   - Configure the API endpoint URL in environment variables.

Backend (AWS EC2 with Docker)

1. Clone the backend repository:

   ```
   bash
   Copy code
   git clone Saksham-21/Doc_to_PDF
   cd backend
   ```

2. Build the Docker image:

   ```
   bash
   Copy code
   docker build -t word-to-pdf-backend .
   ```

3. Run the Docker container:

   ```
   bash
   Copy code
   docker run -p -d 8000:8000 word-to-pdf-backend
   ```

4. Configure the EC2 instance:
    - Open port 8000 in the security group for public access.
    - Set up the EC2 instance with necessary libraries and dependencies.

Usage

1. **Upload**: Navigate to the frontend app and upload a Word file.
2. **Convert**: Click the convert button, which sends the file to the backend API.
3. **Download**: Once the conversion is complete, download the PDF file.

Future Improvements

- Add support for additional file types (e.g., `.txt`, `.rtf`).
- Implement authentication and user profiles.
- Add batch processing for multiple files.

Contributors

- Saksham Singla (Developer)

License