From : Sandip Kothari

[Sandip.kothari@gmail.com](mailto:Sandip.kothari@gmail.com)

Mobile : +91-9885887175

This document covers part 1 and part 2 of the exercise. The source code correspondent to this document contains in zaniaV1.zip.

Contents

[Part 1: Front End & Part 2: Back End 1](#_Toc170923702)

[Step 1: Set up the Backend 1](#_Toc170923703)

[1. First, create a virtual environment and install the necessary packages. 1](#_Toc170923704)

[2. Install necessary dependencies. 2](#_Toc170923705)

[3. Create flask app 2](#_Toc170923706)

[Step 2: Set up the Frontend 2](#_Toc170923707)

[1. Initialize the React Project 2](#_Toc170923708)s

[2. Install Necessary Dependencies 2](#_Toc170923709)

[3. Create a JSON File for Static Data 2](#_Toc170923710)

[4. Create Necessary Components 2](#_Toc170923711)

[5. Index Files 2](#_Toc170923712)

[6. Add Thumbnails 3](#_Toc170923713)

[Final Code Structure 3](#_Toc170923714)

# Part 1: Front End & Part 2: Back End

To build a simple full-stack application with React, Python, and PostgreSQL, we need to complete the following main steps:

1. Set up the backend with Python (using Flask) and PostgreSQL.
2. Create a frontend with React and implement the required functionality.
3. Add a README file to explain how to run the application.

High-level overview of the steps required to accomplish this:

## Step 1: Set up the Backend

We are going to create backend with Python/Flask and PostgreSQL:

### First, create a virtual environment and install the necessary packages.

Below steps are mentioned assuming windows 11 laptop and executed on powershell.

mkdir document\_backend

cd document\_backend

python -m venv venv

.\venv\Scripts\Activate

### Install necessary dependencies.

In document\_backend directory, create a file requirements.txt file and add below lines (text) in this file. This file we can keep on updating in future if there will be a need of new packages.

flask

flask-cors

psycopg2-binary

Now install this dependencies

pip install -r requirements.txt

### Create flask app

Create an app.py file (stored in document\_backend)

## Step 2: Set up the Frontend

We are going to create a full-stack application with React for the frontend.

### Initialize the React Project

Below steps are mentioned assuming windows 11 laptop and executed on powershell.

npx create-react-app document-app

cd document-app

### Install Necessary Dependencies

npm install react-dnd react-dnd-html5-backend react-bootstrap bootstrap

### Create a JSON File for Static Data

Create a file public/documents.json (stored in document-app/public/). This file has static data which is mentioned in the exercise.

### Create Necessary Components

* Create a file named App.js to set up the main application component.
* Create a file named DocumentCard.js for the document card component.
* Create a file named DocumentOverlay.js for the overlay component.
* We need to show the spinner while the image is loading, for that create a file named SpinnerComponent.js and modify the DocumentCard.js accordingly.

### Index Files

Following files are required in the document-app/src directory to bootstrap React application. These files are generated while initializing react project.

* index.js
* index.css : We can add global styles here
* App.css : This file has additional defined styles. By importing App.css file in App.js, the styles defined in App.css will be applied to React components.

### Add Thumbnails

Place thumbnail images for the document types in the public/thumbnail directory. Ensure each image file is named according to the type property in the documents array. Below thumbnails are added.

* bankdraft.png
* bill-of-lading.png
* invoice.png
* bank-draft-2.png
* bill-of-lading-2.png

# Final Code Structure

zania/

│

├── document\_backend/

│ ├── app.py

│ ├── requirements.txt

│ └── setup\_database.sql

│

├── document-app/

│ ├── public/

│ │ ├── thumbnails/

│ │ │ ├── bankdraft.png

│ │ │ ├── bill-of-lading.png

│ │ │ ├── invoice.png

│ │ │ ├── bank-draft-2.png

│ │ │ └── bill-of-lading-2.png

│ │ └── ...

│ ├── src/

│ │ ├── components/

│ │ │ ├── DocumentCard.js

│ │ │ ├── DocumentOverlay.js

│ │ │ ├── SpinnerComponent.js

│ │ └── App.js

│ │ └── App.css

│ │ └── index.js

│ │ └── index.css

│ ├── package.json

└── ...

└── README.md

# How to run ?

### Start the BackendmServer

cd zania\document\_backend

flask run

### Start the Frontend Server

cd zania\document-app

npm start

### Open the Application in a Browser

Open a web browser and navigate to <http://localhost:3000>.

### Test the Drag-and-Drop Functionality:

* Open your browser and go to http://localhost:3000.
* Try dragging and dropping the cards to reorder them.
* The cards should move to the new positions as you drag and drop them.