

## WebApp Project Overview

**Objective:** Create a web application where users can upload a CSV file with financial data, and the application will display charts (e.g., pie, bar charts) based on the uploaded data.

**Tech Stack:** React (Frontend), Flask (Backend)

### Project Breakdown

#### Backend (Flask)

1. **Setup Flask Project:** Initialize a new Flask project.
2. **CSV Upload Endpoint:** Create an endpoint to handle CSV file uploads. This endpoint should save the uploaded file and parse it to extract financial data.
3. **Data Processing:** After extracting the data, perform any necessary data processing or transformation to prepare it for chart generation.
4. **Chart Generation:** Use a Python library (e.g., Matplotlib, Plotly) to generate charts based on the processed data. You could generate the charts as images or JSON data that can be rendered client-side.
5. **Endpoint to Serve Charts:** Create an endpoint to serve the generated charts to the frontend. Depending on your approach in step 4, this could involve sending image files or JSON data.

#### Frontend (React)

1. **Setup React Project:** Initialize a new React application using **create-react-app** or your preferred setup.
2. **File Upload Component:** Implement a component to upload CSV files. This component should allow users to select a CSV file and upload it to the backend.
3. **Display Charts:** After the CSV is uploaded and processed, display the charts generated by the backend. This could involve rendering images or using a JavaScript charting library (e.g., Chart.js, React-Chartjs-2) if you're sending JSON data from the backend.
4. **User Feedback:** Implement loading indicators and error messages to enhance user experience during file upload and data processing.

#### Additional Details

- **CSV File Structure:** Define a simple, consistent structure for the CSV files that your application will accept (e.g., columns for date, category, amount).
- **Security Considerations:** Implement basic security measures, such as validating the uploaded file's MIME type and handling errors gracefully.

#### What to submit for assessment:

1. Code containing implementations mentioned above. Please share the code via google drive/Dropbox/Onedrive. Please don't put it in public github repositories.
2. Report explaining:
  1. Code walk through,
  2. WebApp Architecture,
  3. Readme in MD/PDF format for us to review and run the code on our machine.

Please make the report within 5 pages. We will ask you for a 20 min presentation after you have submitted the code and report after internal review