Technical Challenge

Due Date: 23rd April, 11:59 PM (EST)

Github Project Link : Please provide a Github repo link to the project.

Objective: Create a very simple web application, including both frontend and backend, that enables users to view and explore the invoice/payment data provided in the CSV file. The main goal is to **extract and present meaningful insights about the companies in a clear and intuitive format.**

Technologies Available :

- **Frontend :** Any (React, Python, Flask, Express, etc.)
- **Backend :** Any
- **Version Control :** Git
- **Data** : Provided to you in email (CSV file format)

*Any additional technology can be used on top of this, like OpenAI, Cursor, Github Copilot, etc. Tools like Replit, Vercel, Render, or Heroku for hosting are also **permitted**.*

Tasks:

1. **Data Parsing :** Load and parse the Excel file (you can convert it to JSON or load it in-memory). Handle any edge cases however you like, but please explain the choices you made.
    a. Feel free to convert it to JSON, load it in-memory, or use whichever approach suits your solution.
    b. Handle any missing or malformed rows if necessary (e.g the invoice reference section should all be formatted like so (YEAR-REFERENCE NUMBER).
    c. Store the data in a convenient format for further processing (e.g., JSON in memory, database, etc.).
2. **Viewing Company Results :** Let the user pick a company (dropdown or simple buttons is fine).
    a. Identify the **companies**.
    b. Let the user pick one of these companies via a dropdown or simple buttons.
3. **Insight for Each Company**: For each selected company:
    - **List of invoices** with:
        - Invoice Number
        - Invoice Date
        - Invoice Amount
        - Paid Amount (e.g. is there anyone that hasn't paid the full amount?)
        - Days to Pay (e.g. Payment Date - Invoice Date)

    - **Average Days to Pay**

○ **Monthly Totals** (useful to show aggregated invoice amounts or paid amounts per month; you choose how to present these — table or chart)

○ **Late Invoices**

- **Define your own logic** for “late” (e.g., >30 days from InvoiceDate).
- **Explain your reasoning** for this threshold in your documentation or code comments.

4. **Visualizing the results**
    a. Include at least one chart (bar chart, line chart, or any style) that offers insights, such as: Average Days to Pay per month, or Company revenue over time, etc.

5. **Deploy it Live :** You must give us a live link where we can view the dashboard.

6. **Testing and Documentation:** Test the functionality and document the process.

Deliverables:

1. A complete workflow of code.
2. Link of the web application attached.
3. GitHub Link.

Evaluation Criteria:

1. **Accuracy in Data Structuring**: How well you parse, categorize, and present the data.

2. **Compatibility and Integration**: Smooth integration of front end and back end and clean, maintainable code.

3. **Interface Usability**: Simplicity and clarity of the user interface, responsiveness to different user selections (e.g., changing companies).

4. **Innovation and Adaptability**: Uniqueness of insights or additional features. Demonstration of creativity in problem-solving.

Note: In the evaluation criteria, creativity of ideas and quality of code will be **highly** valued. The focus isn't solely on achieving perfect end results, but on demonstrating thoughtful approaches and learning throughout the project. This includes how the candidate handles challenges and applies creative problem-solving skills.