**Project Summary – Aegis**

Aegis automates the following workflow:

1. **Extract structured data from a consistent-format PDF file.**
2. **Automate interaction with a public website** ("https://www.thevirtualagent.co.za/"):
   * Navigate to the site.
   * Sign in
   * Navigate to the correct page
   * Fill in ‘ID’ field
   * Submit and extract up to 3 cell phone numbers from the results page.
3. **Generate an Excel file (.xlsx)** containing:
   * The extracted ID from the PDF.
   * The cell phone numbers scraped from the website per ID
4. **Save excel. And open it afterwards.**

There should be an option for this to toggle on the GUI weather to open the file after it is finished or not.

**⚙️ Technical Requirements**

* **GUI**: Use **CustomTkinter** for a clean, modern desktop interface.
  + Functionality: PDF selection (either browse or drag and drop file) and a "Start" button to trigger automation.
  + Display status messages: “Processing...”, “Done!”, or error messages.
* **PDF Parsing**: Use **PyMuPDF (fitz)** based on known text labels/structure.
* **Web Automation**: Use **Playwright** to handle navigation, input, and scraping.
* **Excel File Handling**: Use **openpyxl** to create and update .xlsx files.
* **Email Sending**: Use Python’s built-in smtplib and email modules.
* **Deployment**: Final product must be a **Windows standalone executable (.exe)** built with **PyInstaller**.
* **Offline-Only**: Must run fully locally, no external server dependencies.
* **Error Handling**: Add basic handling for file I/O, automation, and email issues.

**🧱 Development Flow (Step-by-Step)**

Walk me through each of these steps and help me set everything up from setting up a repo to installing dependencies. Help me develop this from start to finish please.

1. **PDF Data Extraction**
   * Start with code to extract required fields (e.g. ID) using PyMuPDF.
   * I’ll provide the PDF structure when needed.
2. **CustomTkinter GUI**
   * Build a basic UI for file selection and process initiation.
3. **Playwright Automation**
   * Automate the form-filling, submission, and phone number scraping from "Vurtual Agent".
4. **Excel File Writing**
   * Output the collected data into a formatted .xlsx file.
5. **Email Sending**
   * Add logic to send the Excel file as an email attachment.
6. **Integration & Feedback**
   * Combine components and add real-time GUI feedback and error messages.
7. **Packaging**
   * Show me how to bundle the app using PyInstaller into a .exe for Windows.