Financial Document QA Assistant: Project Summary

This summary outlines the steps taken to build a user-friendly app that reads financial documents and answers questions, explained simply for everyone to understand. The app was developed to meet specific project requirements and helps users analyze financial data.

Step 1: Setting Up the Workspace

What I Did: I created a project folder at `financial-doc-qa-assistant` to store all files. I set up a virtual environment to keep the app's tools organized and separate from other programs on my computer.

Tools Used: Python and a virtual environment.

Step 2: Installing the Main Software (Ollama)

What I Did: I installed Ollama (version 0.1.11) on my Windows computer using `OllamaSetup.exe`. I checked it worked by running a command in Command Prompt to display the version.

Tools Used: Command Prompt and Ollama software.

Step 3: Downloading the Language Model (Llama3)

What I Did: I used Command Prompt to download the `llama3` model via Ollama. I saw progress messages like "pulling manifest" and "success" during the download (4–8GB). I tested it by asking, "What is 2+2?" and got the answer 4.

Tools Used: Command Prompt and Ollama's `llama3` model.

Step 4: Preparing the App's Tools

What I Did: In a new PowerShell terminal in PyCharm, I activated the virtual environment and installed the `ollama` Python library (version 0.5.4), along with `streamlit`, `pdfplumber`, and `pandas`. I verified all installations worked correctly.

Tools Used: PyCharm, PowerShell, and Python libraries (`ollama`, `streamlit`, `pdfplumber`, `pandas`).

Step 5: Testing the App with a Financial Document

What I Did: I ran the app using `streamlit run app.py` in the PowerShell terminal, opening a webpage at `http://localhost:8501`. I uploaded `Test_Balance_Sheet.pdf` clicked "Process Documents," and tested questions:

- "What is the total value of XYZ Enterprises' assets as of March 31, 2025?" (Answer: $970,000)
- "How much is the company's total current liabilities?" (Answer: $145,000)
- "What is the net value of Property, Plant, and Equipment after depreciation?" (Answer: $450,000)

All answers were correct, and I took screenshots of the webpage showing these results.

Tools Used: Streamlit app, `Test_Balance_Sheet.pdf`, and a web browser.

Step 6: Saving Screenshots for Proof

What I Did: I created a `screenshots` folder in my project directory and saved images of the app's webpage showing each question and answer (e.g., `assets_question.png`).

Tools Used: Windows screenshot tool (`Win + Shift + S`) and a `screenshots` folder.

Step 7: Sharing the Project on GitHub

What I Did: I created a public GitHub repository named `financial-document-qa-assistant` I uploaded `app.py`, `Test_Balance_Sheet.pdf`, `Sample_Balance_Sheet.pdf`, `requirements.txt`, and the `screenshots` folder. I updated `README.md` with instructions on how to use the app, including setup steps and example questions.

Tools Used: Git, GitHub, and PowerShell.

Step 8: Submitting the Project

What I Did: I prepared an email or portal submission with the GitHub repository URL, explaining that the app processes financial documents and answers questions (e.g., total assets: $970,000). I included the screenshots in the `screenshots` folder as proof.

Tools Used: Email or submission portal.

Conclusion

The Financial Document QA Assistant is complete, successfully reading financial documents and providing accurate answers. By setting up the workspace, installing Ollama and `llama3`, adding necessary tools, testing with real questions,

and sharing on GitHub, I've built a reliable and professional app. The screenshots and clear instructions ensure anyone can use it, demonstrating my ability to create effective, user-friendly tools.