

## ✓ 1. System Requirements

- Python 3.8 or higher
- Pip (Python package manager)
- Git (optional, if you want to clone directly from GitHub)

## ✓ 2. Set Up Project Directory

1. Open your terminal or command prompt.
2. Create a new directory and navigate into it:

```
mkdir stock-assistant  
cd stock-assistant
```

3. (If you have the files in GitHub):

```
git clone <REPO_URL> .
```

4. Or, manually copy the provided Python file into this directory and name it, e.g., app.py.

## ✓ 3. Create a Virtual Environment (Recommended)

```
python -m venv venv  
source venv/bin/activate    # On Windows: venv\Scripts\activate
```

## ✓ 4. Create a Requirements File

Create a file named requirements.txt and paste the following inside:

```
txt  
CopyEdit  
flask  
yfinance  
matplotlib  
mplfinance  
reportlab  
pandas  
langchain  
openai
```

### Note:

- llm\_utils is a **custom module** likely in the same project directory — make sure it's present or create a dummy one for testing.

- langchain\_community.llms is part of the LangChain framework and assumes OpenAI API integration.

## ✓ 5. Install Dependencies

Run:

```
pip install -r requirements.txt
```

If any package fails to install, make sure:

- You are using a compatible Python version
- You are connected to the internet

## ✓ 6. Check Folder Structure

Ensure your folder contains at least:

```
stock-assistant/  
├── app.py  
├── llm_utils.py    # or adjust imports if it's different  
├── templates/  
│   └── index.html  # For rendering the main page  
├── static/         # Auto-generated for charts and reports  
└── requirements.txt
```

Create templates/index.html with a basic form if not provided.

## ✓ 7. Set the Secret Key (Optional)

You can set the FLASK\_SECRET\_KEY in a .env file, or leave the default:

```
export FLASK_SECRET_KEY="your_custom_secret" # For Linux/macOS  
set FLASK_SECRET_KEY="your_custom_secret"    # For Windows
```

Or you can hardcode it in the script for testing:

```
app.secret_key = "super_secret_key_123"
```

## ✓ 8. Run the App

From your terminal, run:

```
python app.py
```

If using Flask's CLI directly:

```
export FLASK_APP=app.py
export FLASK_ENV=development
flask run
```

Visit in your browser:

<http://127.0.0.1:5000/>

## ✓ 9. Interacting with the App

- Use the web UI to enter stock-related queries.
- Try inputs like:
  - "Show me the chart for AAPL"
  - "Generate report for RELIANCE"
  - "What's the dividend for TSLA?"

## ✓ 10. Important Notes

- The `llm_utils.py` file must exist and include a working `llm` or `OpenAI` call.
- If you're not using `LangChain/OpenAI` features, **you can comment out or remove**:

```
python
CopyEdit
from langchain_community.llms import OpenAI
from llm_utils import llm
```

- Flask will generate `*.png` and `*.pdf` reports in the `static/` directory.

## 🔧 Optional Enhancements

- Use `.env` files for secrets with the `python-dotenv` package.
- Deploy to services like **Heroku**, **Render**, or **AWS Elastic Beanstalk** for public access.