AI-Powered Personal Assistant Project

This project will help you build an AI-powered personal assistant using Python. It will showcase your ability to integrate various technologies and demonstrate skills in Python, machine learning, and natural language processing.

Key Features:

1. Natural Language Understanding (NLU): Use libraries like NLTK, SpaCy, or Transformer-based models (like GPT) to process and understand user input.
2. Task Management: Allow users to create, update, and delete tasks using natural language commands. Store tasks in a local database (e.g., SQLite).
3. Calendar Integration: Integrate with a calendar API (like Google Calendar) to add and retrieve events. Provide reminders and notifications for upcoming events.
4. Weather Information: Fetch and display current weather information using a weather API (e.g., OpenWeatherMap).
5. Web Scraping for News: Scrape news headlines from popular websites and display them to the user.
6. Voice Interface: Integrate a speech-to-text and text-to-speech system using libraries like SpeechRecognition and pyttsx3 to enable voice commands and responses.

Technical Stack:

Python: Core programming language.

NLTK/SpaCy/Transformers: For natural language processing.

SQLite: For local database storage.

APIs: For calendar and weather information.

BeautifulSoup/Requests: For web scraping.

SpeechRecognition/pyttsx3: For voice interface.

Why This Project?

- Comprehensive Skill Set: Demonstrates your ability to work with APIs, perform web scraping, and implement machine learning models.
- Practical Application: Shows you can build applications that solve real-world problems.
- Impressive to Employers: Highlights your ability to integrate various technologies into a cohesive project.

Steps to Get Started:

1. Set up your development environment: Install Python and Visual Studio Code on your Mac.
2. Create a new project directory and set up a virtual environment.
3. Install necessary libraries: nltk, spacy, transformers, sqlite3, requests, beautifulsoup4, SpeechRecognition, pyttsx3.
4. Start with the NLU module: Implement basic natural language understanding using SpaCy or NLTK.
5. Implement task management: Create a simple SQLite database to store tasks and allow CRUD operations.
6. Integrate calendar API: Use the Google Calendar API to add and retrieve events.
7. Fetch weather information: Use the OpenWeatherMap API to get current weather data.
8. Web scraping for news: Use BeautifulSoup and Requests to scrape news headlines.
9. Implement the voice interface: Use SpeechRecognition for speech-to-text and pyttsx3 for text-to-speech functionality.
10. Test and refine each module, then integrate them into a cohesive application.