PROJECT DOCUMENTATION

ON

AI Chemist: Pioneering the Future of Chemical Science with Gemini Vision Pro

Submitted in partial fulfillment of the requirements

for the completion of the internship program

GOOGLE CLOUD GENERATIVE AI VIRTUAL INTERNSHIP PROGRAM

BY

Riya Maurya

Table of Contents

1. Introduction

Project Overview:

The **AI Chemist** application is designed to revolutionize the field of chemical research by leveraging the power of artificial intelligence. It assists chemists by analyzing chemical inputs, identifying tablets from images, and generating detailed experimental recommendations using the Gemini 1.5 Flash model. The application aims to enhance research efficiency and foster innovation through intelligent, data-driven guidance.

Motivation:

The increasing complexity of chemical research demands innovative tools to accelerate discoveries and streamline processes. **AI Chemist** addresses this need by providing chemists with an AI-driven assistant capable of offering insightful recommendations, thereby reducing the time and effort required in the research process.

2. Project Overview

Description:

AI Chemist is a mobile application that integrates AI technology to assist chemists in various research tasks. It allows users to input chemical data, upload images of tablets, and receive detailed analysis and experimental suggestions. The app uses the Gemini 1.5 Flash model to process inputs and generate outputs tailored to the specific needs of chemists.

Target Audience:

This application is primarily designed for chemists, pharmaceutical researchers, and materials scientists who require precise and reliable data to inform their research and experiments.

Use Cases

- **Pharmaceutical Research**: Identifying potential drug compounds and synthesis routes.
- **Green Chemistry**: Developing eco-friendly chemical solutions.
- **Polymer Science**: Designing polymers with specific properties for industrial applications.

3. Objectives

Primary Goal

To assist chemists in research by providing AI-driven insights and recommendations based on chemical inputs and image data.

Specific Objectives

- Analyze chemical inputs to generate tailored recommendations.
- Identify tablets from images and provide detailed information.
- Suggest experimental procedures based on input data.

4. Technical Architecture

System Design

The **AI Chemist** application is structured to process user inputs (text and images) through a streamlined pipeline. The user interacts with a front-end interface built with Streamlit, where they can input chemical data and upload images. This data is then sent to the backend, where the Gemini 1.5 Flash model processes it and returns detailed analysis and recommendations.

Technology Stack

- **Front-End**: Streamlit for building the user interface.
- **Back-End**: Python, with the Google Gemini 1.5 Flash model for AI processing.
- **Environment Management**: Python-dotenv for managing environment variables.
- **Image Processing**: Pillow for handling and processing images.
- **PDF Handling**: PyPDF2 for extracting content from PDF documents.

Project Structure

- **images/**: Contains images used in the user interface.
- **.env**: Stores the Google API key securely.
- **app.py**: The main application file containing both the model integration and the Streamlit UI code.
- **requirements.txt**: Lists all the necessary libraries and dependencies for the project.

5. Implementation Details

Implementation Steps for AI Chemist Project in Visual Studio Code

1. Set Up the Project Environment

- Install **Visual Studio Code (VS Code)** if not already installed.
- Install **Python** and ensure it's added to your system's PATH.
- Open VS Code and create a new folder for your project.

2. Create Project Files

- Within your project folder, create the following files and directories:
  - **app.py:** The main application file containing the Streamlit UI and model integration code.
  - **.env:** A file to securely store your Google API key.
  - **requirements.txt:** A file listing the required Python libraries.
  - **images/:** A directory to store images used in the user interface.

3. Set Up Virtual Environment

4. Install Required Libraries

- With the virtual environment activated, install the required libraries listed in requirements.txt:

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

5. Configure Google API Key

- Open the .env file and add your Google API key:

```
GOOGLE_API_KEY=your_google_api_key_here
```

6. Implement the Application Logic

- Write your application code in app.py, which includes loading the Gemini model, processing inputs, and displaying results using Streamlit.

7.Run the Application

- In the terminal, run the Streamlit app using the following command:

```
streamlit run app.py
```

The app will open in your default web browser, where you can interact with the UI.

6. Testing and Results

Testing Strategy

- **Manual Testing**: The application was tested manually by uploading different images of tablets and providing varied chemical inputs. The responses were evaluated for accuracy and relevance.
- **Automated Testing**: Basic unit tests were written to ensure that core functions like get_gemini_response() and input_image_setup() work as expected.

Sample Inputs and Outputs

- **Input**: An image of a tablet and a chemical compound name.
- **Output**: Detailed analysis of the tablet's composition and potential applications in pharmaceuticals.

Performance Metrics

- **Response Time**: The AI model processes input and generates responses within a few seconds.
- **Accuracy**: The model's suggestions were accurate in identifying tablets and providing relevant recommendations.

7. Challenges and Solutions

Key Challenges

- **API Deprecation**: The initial model (Gemini Pro Vision) was deprecated, requiring a switch to Gemini 1.5 Flash.
- **Integration Issues**: Handling image data and integrating it with the AI model posed challenges, particularly in formatting the data correctly.

Solutions

- **Model Update**: Updated the code to use the latest Gemini 1.5 Flash model, ensuring continued functionality.
- **Data Handling**: Improved the input_image_setup() function to correctly process and format image data for the AI model.

8. Conclusion

The **AI Chemist** application successfully demonstrates the potential of AI in advancing chemical research. By providing chemists with an AI-powered assistant capable of analyzing inputs and offering experimental recommendations, this project paves the way for more efficient and innovative research practices. Future developments could include expanding the model's capabilities to cover a broader range of chemical analysis and integrating additional data sources.

9. References

- NLP: https://www.tutorialspoint.com/natural_language_processing/index.htm
- Generative AI: https://en.wikipedia.org/wiki/Generative_artificial_intelligence
- About Gemini: https://deepmind.google/technologies/gemini/#introduction
- Gemini API: https://ai.google.dev/gemini-api/docs/get-started/python
- Gemini Demo: https://colab.research.google.com/github/google/generative-ai-docs/blob/main/site/en/gemini-api/docs/get-started/python.ipynb
- Streamlit: https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/