AI Chemist

by Bedanta Gautom

Introduction :-

AI Chemist is a state-of-the-art mobile application designed to revolutionize chemical research by offering personalized solutions and experimental recommendations powered by the advanced Gemini Pro AI model. This innovative app evaluates user input, laboratory conditions, and research objectives to provide tailored experiment designs, optimized chemical synthesis routes, and insightful data analysis. By leveraging artificial intelligence, AI Chemist helps researchers streamline their processes, reduce trial and error, and enhance research outcomes. The app's goal is to significantly boost research efficiency and foster innovation in the chemistry field, offering intelligent, data-driven guidance that accelerates discovery and problem-solving, while supporting a more efficient and cost-effective approach to experimentation.

Key Features of AI Chemist -

- *Personalized Chemical Solutions:*
  AI Chemist evaluates a variety of inputs from the user, including research goals, laboratory conditions, and available materials. Using this data, it generates customized chemical solutions that fit the user's specific needs. Whether the goal is to synthesize a novel compound or optimize a reaction process, the AI model provides step-by-step guidance on the most effective approaches.

- *Experimental Recommendations:*
  The app suggests optimized experimental protocols based on real-time input. By analyzing the variables in play—such as temperature, pressure, solvents, catalysts, and reaction conditions—it proposes experimental designs that maximize success and efficiency. This makes it easier for researchers to conduct complex experiments without the risk of failure due to improper setups.

- *Chemical Synthesis Routes:*
  AI Chemist offers intelligent routes for chemical synthesis, identifying potential pathways for the creation of new compounds. It provides a clear, organized synthesis pathway with detailed instructions and safety measures, ensuring that chemists are guided through every step of the process, from starting reagents to final products. This feature saves time, reduces trial-and-error, and opens up new avenues for research and discovery.

- *Insightful Data Analysis:*
  The app doesn't just provide recommendations for experiments—it also plays a crucial role in post-experiment analysis. Users can input experimental data, and the AI will process this information to identify patterns, trends, and anomalies. With this feature, users receive meaningful insights into their experimental outcomes, including potential improvements, yield optimizations, or novel interpretations of the data that can lead to breakthroughs.

- *Research Optimization:*
  AI Chemist aims to streamline the research process by suggesting time-saving solutions, predicting possible challenges, and offering alternative routes or modifications to experimental plans. It helps reduce costs by identifying more efficient methods or reagents, and also aids in the exploration of new chemical frontiers. The result is an environment where researchers can innovate more rapidly and confidently.

- *Enhanced Collaboration:*
  The app provides a platform for collaborative research by offering a shared space where users can access common experimental setups, share data, and exchange findings. This aspect is especially valuable for teams working in multidisciplinary environments or across geographical locations. AI Chemist's ability to unify diverse research objectives enhances global collaboration in chemistry.

- *User-Friendly Interface:*
  Despite its sophisticated technology, AI Chemist is designed to be intuitive and user-friendly. Researchers, whether beginners or seasoned professionals, can easily input their parameters and receive meaningful results without the need for deep technical expertise in artificial intelligence or machine learning. The seamless interface ensures that chemists can focus on their research rather than the tool itself.

Objectives :-

The process of using AI Chemist involves a series of seamless interactions between the user, the mobile app's frontend, backend, and the advanced Gemini Pro AI model. Here's a detailed breakdown of how the system works:

1. *User Interaction with the UI:*
   The user begins by interacting with the app's user interface (UI), where they input specific data relevant to their experiment or research. This could include details like chemical compounds, laboratory conditions (e.g., temperature, pressure), research objectives, or experimental constraints. The UI is designed to be intuitive and user-friendly, making it easy for both novice and experienced researchers to provide the necessary information.

2. *Input Transmission to Backend:*
   Once the user submits their input through the UI, the app collects this data and transmits it to the backend of the application. This transmission is securely handled using the Google API key, which ensures that the data exchange is both authenticated and encrypted, maintaining the privacy and security of the user's input.

3. *API Call to Gemini Pro:*
   After the backend receives the user input, it forwards the data to the **Gemini Pro pre-trained model** via an API call. The Gemini Pro model, which has been trained on vast amounts of chemical data, then processes the input by analyzing the various factors such as chemical properties, lab conditions, and research goals. This step leverages the advanced capabilities of the AI model to generate highly relevant and accurate experimental recommendations, synthesis routes, or data analysis based on the input provided.

4. *Processing and Output Generation:*
   The **Gemini Pro model** processes the input in real time, using its deep learning algorithms to evaluate the user's requirements. Based on its training, the model generates customized solutions, such as optimal experimental designs, chemical reaction pathways, or insights drawn from experimental data. This output is both intelligent and context-aware, taking into account the latest research trends and chemical principles to deliver highly relevant results.

5. *Results Displayed on the Frontend:*
   Once the model has processed the data and generated its output, the results are sent back to the frontend of the application. The frontend then formats the output, ensuring that it is presented clearly and in a user-friendly manner. Whether it’s a suggested experimental protocol, a synthesis route, or a data analysis report, the results are displayed in an easy-to-read format that helps the user understand and apply the information. The UI ensures that the data is presented logically, allowing users to quickly interpret and act on the AI-generated insights.

Project Flow :-

Requirements Specification

- Create a requirements.txt file to list the required libraries.
- Install the required libraries

Initialization of Google API Key

- Generate Google API Key
- Initialize Google API Key

Interfacing with Pre-trained Model

- Load the Gemini Pro pre-trained model
- Implement a function to get gemini response
- Implement a function to read PDF content
- Write a prompt for gemini model

Model Deployment

- Integrate with Web Framework
- Host the Application

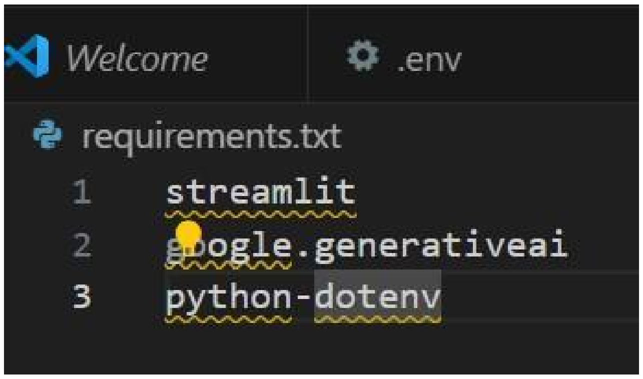
Project Structure :-

- *images folder*: It is established to store the images utilized in the user interface.
- *.env file*: It securely stores the Google API key.
- *app.py*: It serves as the primary application file housing both the model and Streamlit UI code.
- *requirements.txt*: It enumerates the libraries necessary for installation to ensure proper functioning.
- Additionally, ensure proper file organization and adhere to best practices for version control.

Requirement Specification :-

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.

<u>Create a requirements.txt file to list the required libraries:</u>



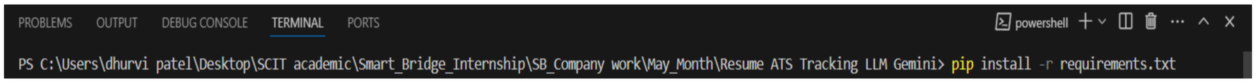
Welcome .env

requirements.txt

```
streamlit
google.generativeai
python-dotenv
```

- streamlit: Streamlit is a powerful framework for building interactive web applications with Python.
- streamlit_extras: Additional utilities and enhancements for Streamlit applications.
- google-generativeai: Python client library for accessing the GenerativeAI API, facilitating interactions with pre-trained language models like Gemini Pro.
- python-dotenv: Python-dotenv allows you to manage environment variables stored in a .env file for your Python projects.
- PyPDF2: It is a Python library for extracting text and manipulating PDF documents.
- Pillow: Pillow is a Python Imaging Library (PIL) fork that adds support for opening, manipulating, and saving many different image file formats.

<u>Install the required libraries:</u>



```
PS C:\Users\dhurvi patel\Desktop\SCIT academic\Smart_Bridge_Internship\SB_Company work\May_Month\Resume ATS Tracking LLM Gemini> pip install -r requirements.txt
```

- Open the terminal.
- Run the command: pip install -r requirements.txt
- This command installs all the libraries listed in the requirements.txt file

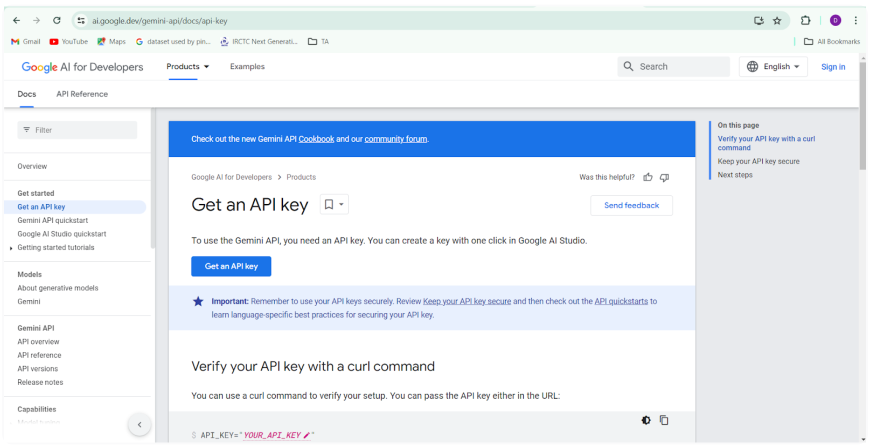
Initialization of Google API Key :-

The Google API key is a secure access token provided by Google, enabling developers to authenticate and interact with various Google APIs. It acts as a form of identification, allowing users to access specific Google services and resources. This key plays a crucial role in authorizing and securing API requests, ensuring that only authorized users can access and utilize Google's services.

Generate Google API Key:

- Click the provided link to access the following webpage.

  Link: https://ai.google.dev/gemini-api/docs/api-key



Check out the new Gemini API Cookbook and our community forum.

Get an API key

To use the Gemini API, you need an API key. You can create a key with one click in Google AI Studio.

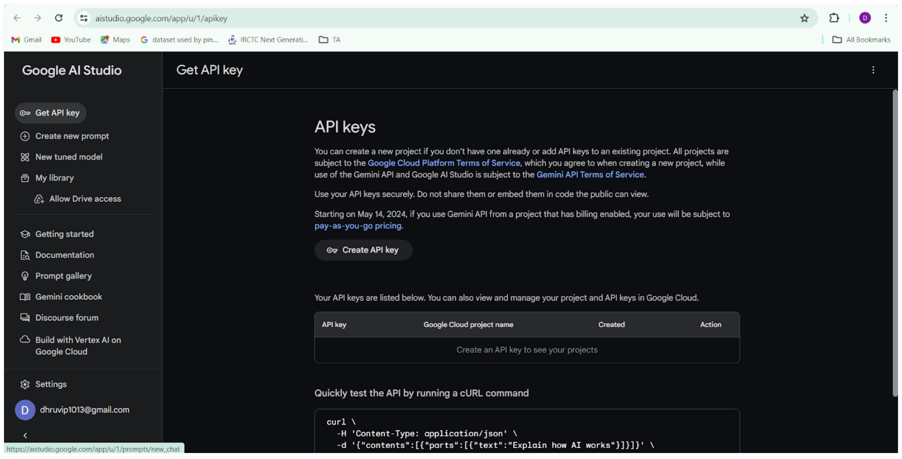
Important: Remember to use your API keys securely. Review Keep your API key secure and then check out the API quickstarts to learn language-specific best practices for securing your API key.

Verify your API key with a curl command

You can use a curl command to verify your setup. You can pass the API key either in the URL:

```
$ API_KEY="YOUR_API_KEY"
```

- After signing in to your account, navigate to the 'Get an API Key' option. Clicking on this option will redirect you to another webpage as shown below.



API keys

You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the Google Cloud Platform Terms of Service, which you agree to when creating a new project, while use of the Gemini API and Google AI Studio is subject to the Gemini API Terms of Service.

Use your API keys securely. Do not share them or embed them in code the public can view.

Starting on May 14, 2024, if you use Gemini API from a project that has billing enabled, your use will be subject to pay-as-you-go pricing.

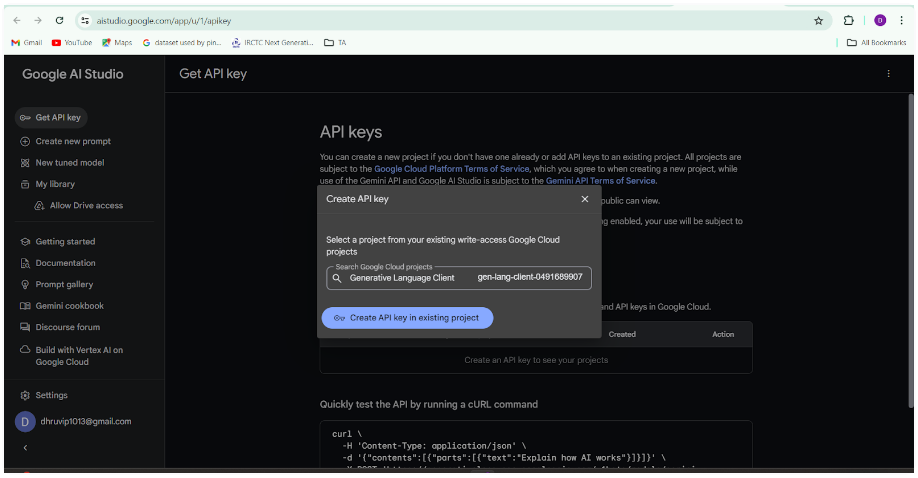
Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

API key	Google Cloud project name	Created	Action
Create an API key to see your projects			

Quickly test the API by running a cURL command

```
curl \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' \
```

- Next, click on 'Create API Key' and choose the generative language client as the project. Then, select 'Create API key in existing project'.



Get API key

API keys

You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the Google Cloud Platform Terms of Service, which you agree to when creating a new project, while use of the Gemini API and Google AI Studio is subject to the Gemini API Terms of Service.

public can view.

g enabled, your use will be subject to

Create API key

Select a project from your existing write-access Google Cloud projects

Search Google Cloud projects: Generative Language Client gen-lang-client-0491689907

Create API key in existing project

nd API keys in Google Cloud.

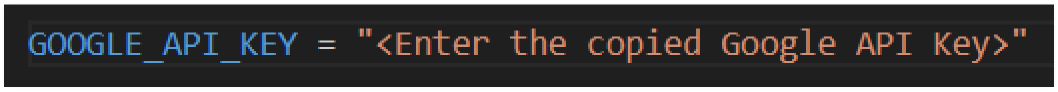
Created	Action
Create an API key to see your projects	

Quickly test the API by running a cURL command

```
curl \
  -H 'Content-Type: application/json' \
  -d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' \
```

- Copy the newly generated API key as it is required for loading the Gemini Pro pre-trained model.

Initialize Google API Key:



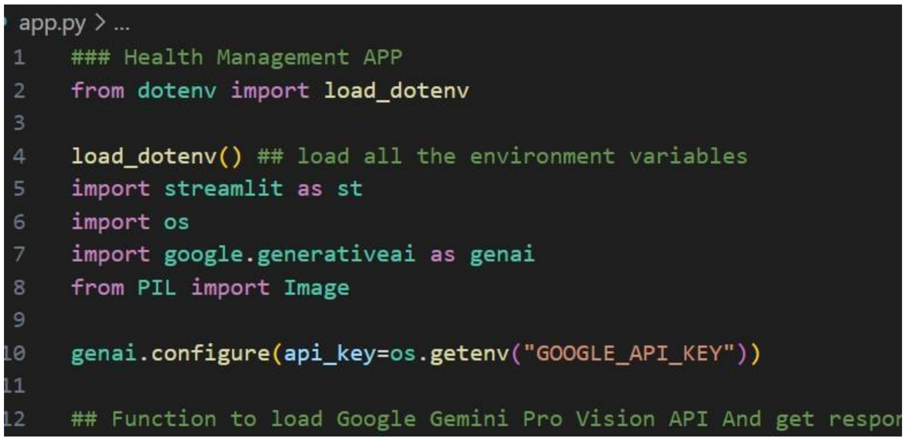
```
GOOGLE_API_KEY = "<Enter the copied Google API Key>"
```

- Create a .env file and define a variable named GOOGLE_API_KEY.
- Assign the copied Google API key to this variable.
- Paste the API key obtained from the previous steps here.

Interfacing with Pre-trained Model :-

To interface with the pre-trained model, we'll start by creating an app.py file, which will contain both the model and Streamlit UI code.

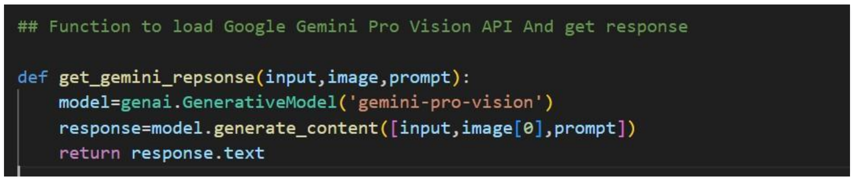
Load the Gemini Pro API:



```
app.py > ...
### Health Management APP
from dotenv import load_dotenv

load_dotenv() ## load all the environment variables
import streamlit as st
import os
import google.generativeai as genai
from PIL import Image

genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))

## Function to load Google Gemini Pro Vision API And get respor
```

This code snippet is for initializing a health management application using Streamlit, an open-source app framework, and Google Generative AI services. The script starts by loading environment variables from a .env file using the load_dotenv() function from the dotenv package. It then imports necessary libraries: streamlit for creating the web app interface, os for accessing environment variables, google.generativeai for utilizing Google's Generative AI capabilities, and PIL.Image for image processing. The genai.configure() function is called to set up the Google Generative AI API with the API key retrieved from the environment variables, ensuring secure and authorized access to the AI services.

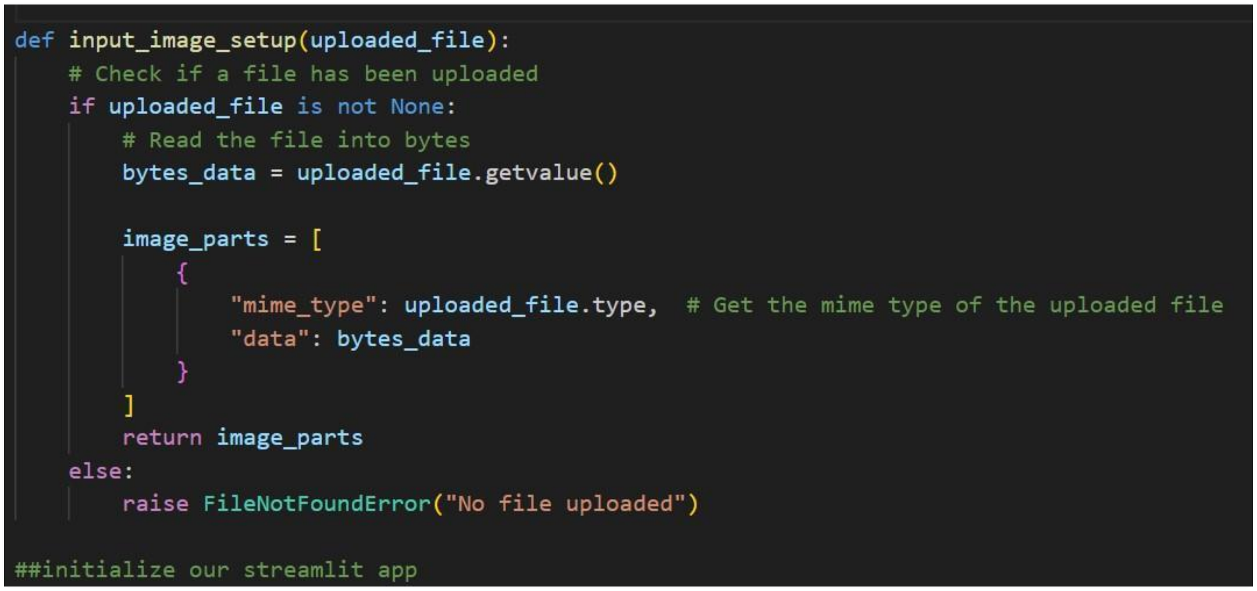
Implement a function to get gemini response:



```
## Function to load Google Gemini Pro Vision API And get response

def get_gemini_repsonse(input,image,prompt):
    model=genai.GenerativeModel('gemini-pro-vision')
    response=model.generate_content([input,image[0],prompt])
    return response.text
```

- The function get_gemini_response takes an input text as a parameter.
- It calls the generate_content method of the model object to generate a response.
- The generated response is returned as text.

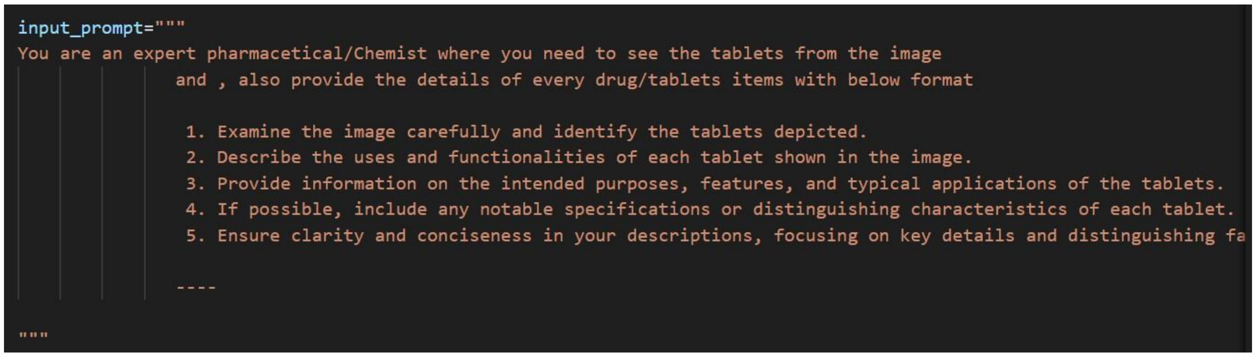
Implement a function to read the Image and set the image format for Gemini Pro model Input:



```python
def input_image_setup(uploaded_file):
    # Check if a file has been uploaded
    if uploaded_file is not None:
        # Read the file into bytes
        bytes_data = uploaded_file.getvalue()

        image_parts = [
            {
                "mime_type": uploaded_file.type,  # Get the mime type of the uploaded file
                "data": bytes_data
            }
        ]
        return image_parts
    else:
        raise FileNotFoundError("No file uploaded")

##initialize our streamlit app
```

The function input_image_setup processes an uploaded image file for a health management application. It first checks if a file has been uploaded. If a file is present, it reads the file's content into bytes and creates a dictionary containing the file's MIME type and its byte data. This dictionary is then stored in a list named image_parts, which is returned by the function. If no file is uploaded, the function raises a FileNotFoundError, indicating that an image file is required but not provided. This setup ensures that the uploaded image is correctly formatted and ready for further processing or analysis in the application.

Write a prompt for gemini model:



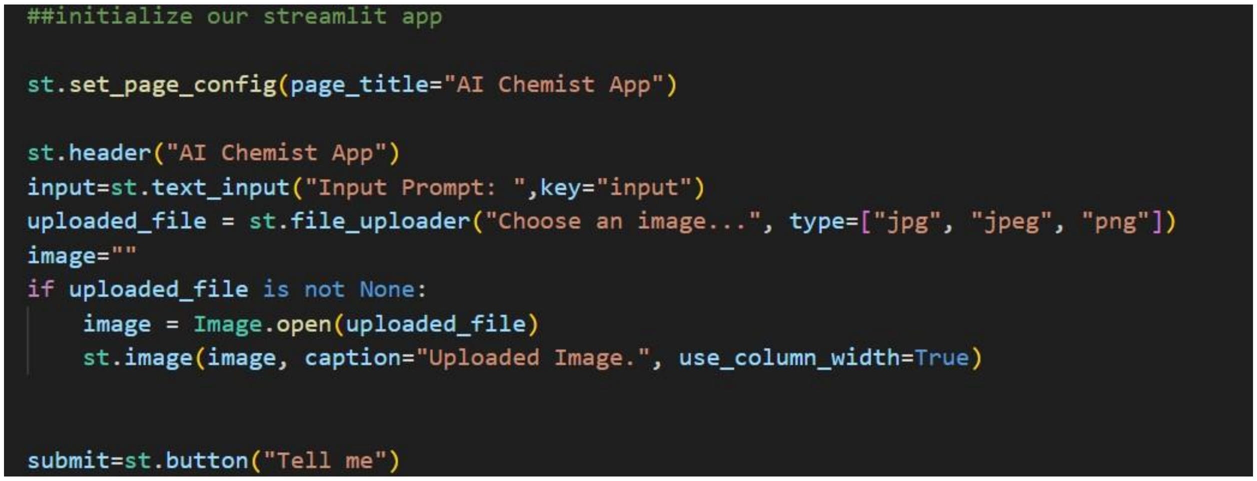
```python
input_prompt="""
You are an expert pharmacetical/Chemist where you need to see the tablets from the image
               and , also provide the details of every drug/tablets items with below format

                1. Examine the image carefully and identify the tablets depicted.
                2. Describe the uses and functionalities of each tablet shown in the image.
                3. Provide information on the intended purposes, features, and typical applications of the tablets.
                4. If possible, include any notable specifications or distinguishing characteristics of each tablet.
                5. Ensure clarity and conciseness in your descriptions, focusing on key details and distinguishing fa

               ----

"""
```

The variable input_prompt is a multi-line string designed as a prompt for a nutritionist AI model. It instructs the model to analyze an image of food items, identify each food item, and calculate the total calories. Additionally, the model is to provide a detailed breakdown of each food item with its respective calorie count. The expected output format is a numbered list where each item is listed alongside its calorie content, ensuring clarity and structured information for the user. This prompt is likely used in conjunction with an AI service that can process images and generate nutritional information based on the visual data provided.

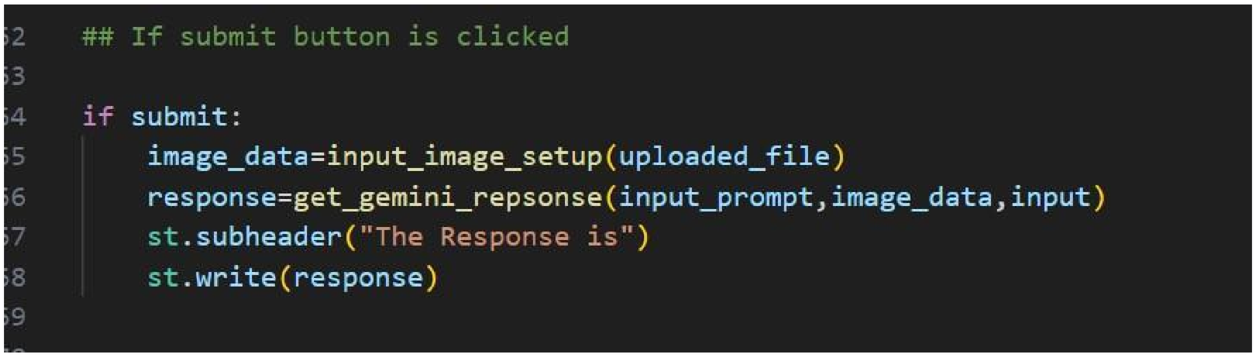
Model Deployment :-

We deploy our model using the Streamlit framework, a powerful tool for building and sharing data applications quickly and easily. With Streamlit, we can create interactive web applications that allow users to interact with our models in real-time, providing an intuitive and seamless experience.

Integrate with Web Framework:



```python
##initialize our streamlit app

st.set_page_config(page_title="AI Chemist App")

st.header("AI Chemist App")
input=st.text_input("Input Prompt: ",key="input")
uploaded_file = st.file_uploader("Choose an image...", type=["jpg", "jpeg", "png"])
image=""
if uploaded_file is not None:
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Image.", use_column_width=True)


submit=st.button("Tell me")
```



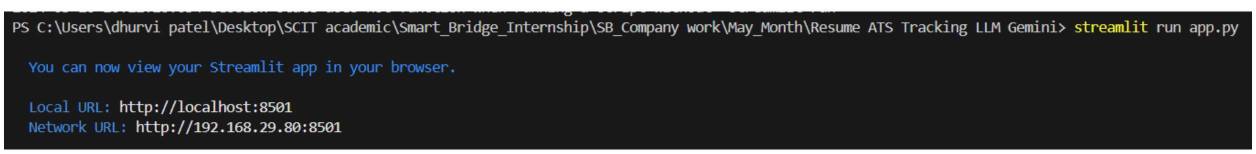
```python
## If submit button is clicked

if submit:
    image_data=input_image_setup(uploaded_file)
    response=get_gemini_repsonse(input_prompt,image_data,input)
    st.subheader("The Response is")
    st.write(response)

```

This code initializes a Streamlit application titled "AI Nutritionist App" by setting the page title and creating the app's header. It includes a text input field for users to enter a custom prompt and a file uploader for users to upload an image in JPG, JPEG, or PNG format. If an image is uploaded, it is opened using the PIL library and displayed within the app with a caption. A button labeled "Tell me the total calories" is also provided, which users can click to trigger the application's functionality for analyzing the uploaded image to calculate and display the total calorie content of the food items depicted.

Host the Application:

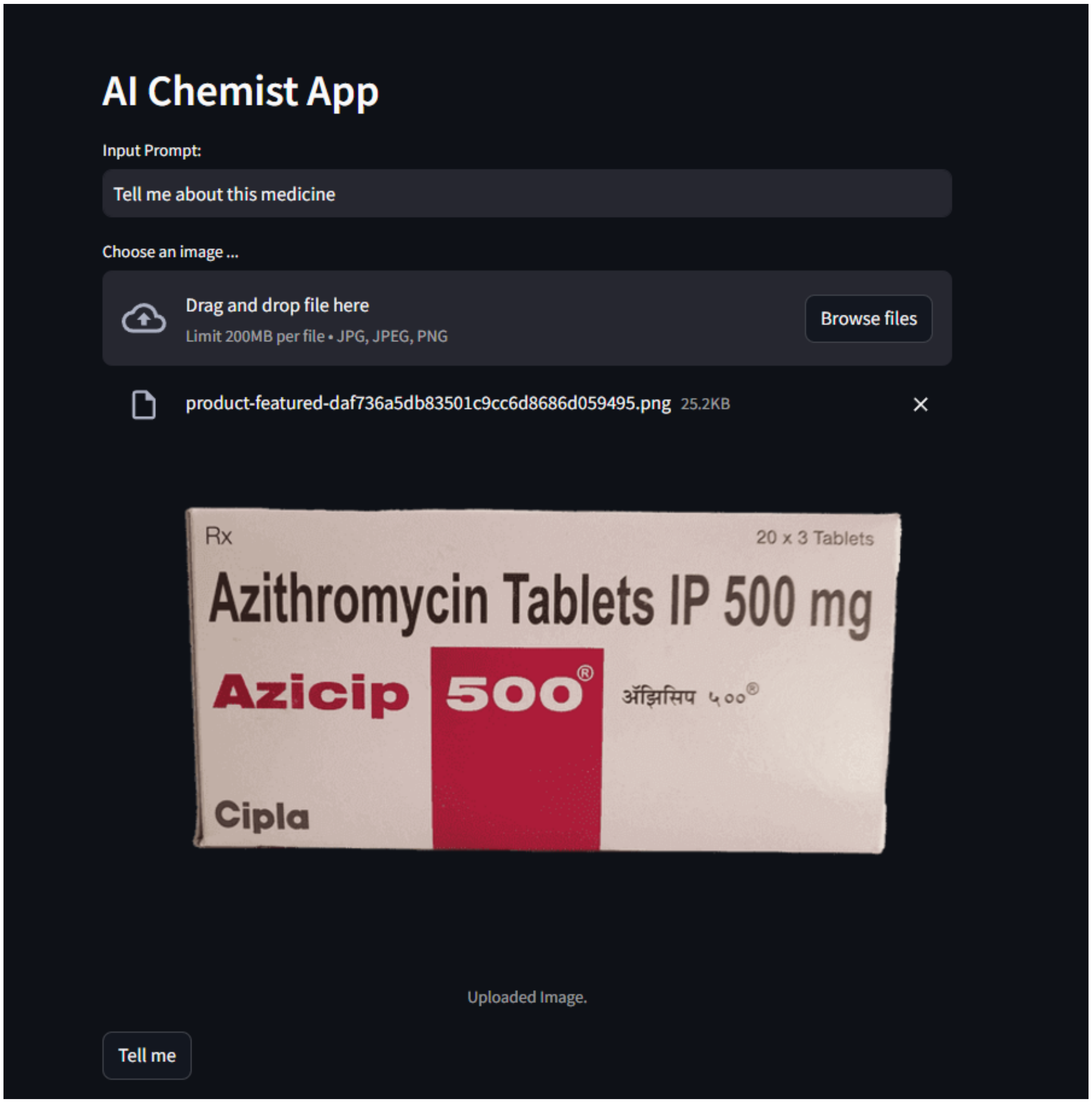
Launching the Application:

- To host the application, go to the terminal, type - streamlit run app.py
- Here app.py refers to a python script.



```
PS C:\Users\dhurvi patel\Desktop\SCIT academic\Smart_Bridge_Internship\SB_Company work\May_Month\Resume ATS Tracking LLM Gemini> streamlit run app.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.29.80:8501
```

Input 1:



AI Chemist App

Input Prompt:

Tell me about this medicine

Choose an image ...

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

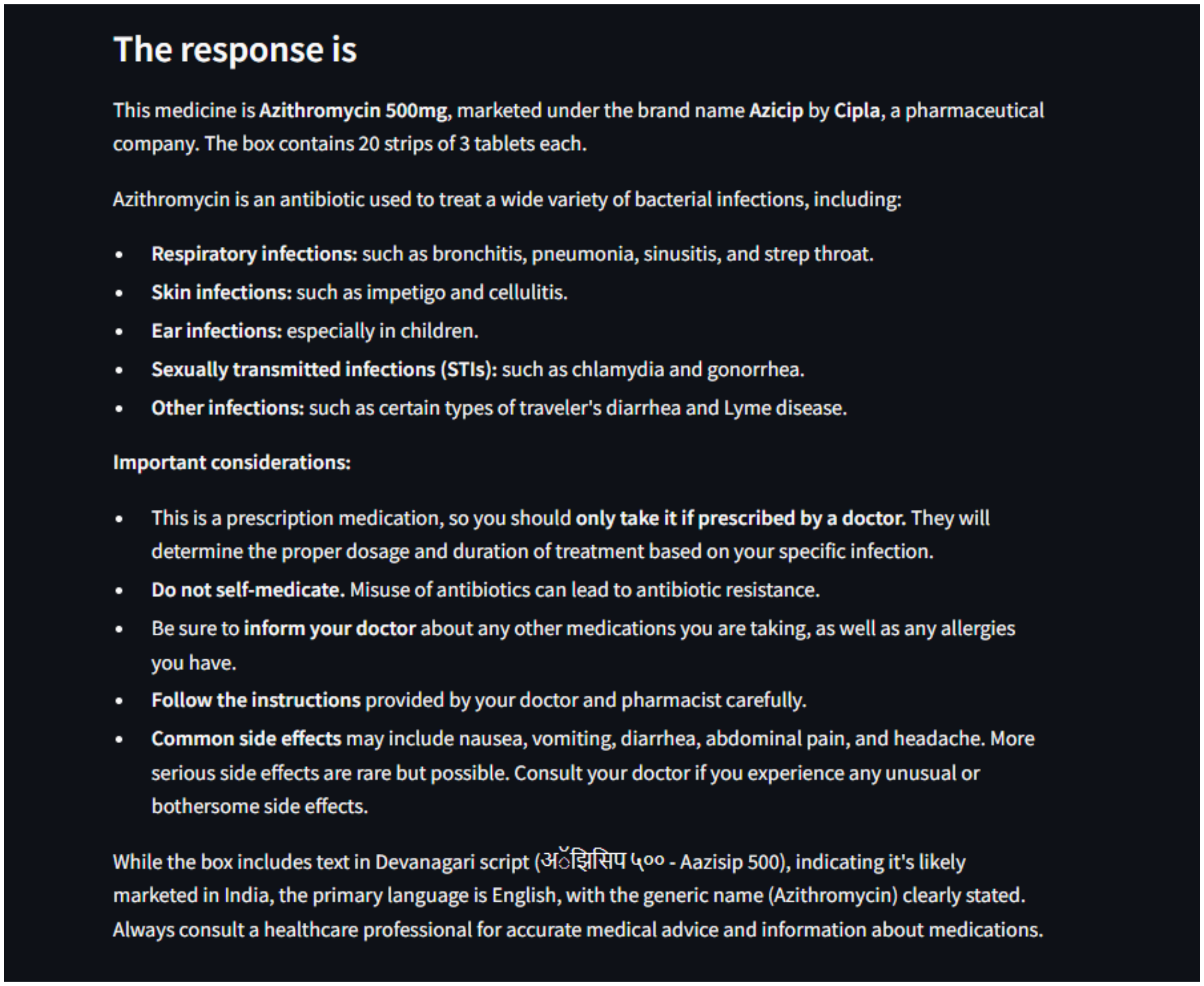
Browse files

product-featured-daf736a5db83501c9cc6d8686d059495.png 25.2KB

Uploaded Image.

Tell me

Output 1:



The response is

This medicine is **Azithromycin 500mg**, marketed under the brand name **Azicip** by **Cipla**, a pharmaceutical company. The box contains 20 strips of 3 tablets each.

Azithromycin is an antibiotic used to treat a wide variety of bacterial infections, including:

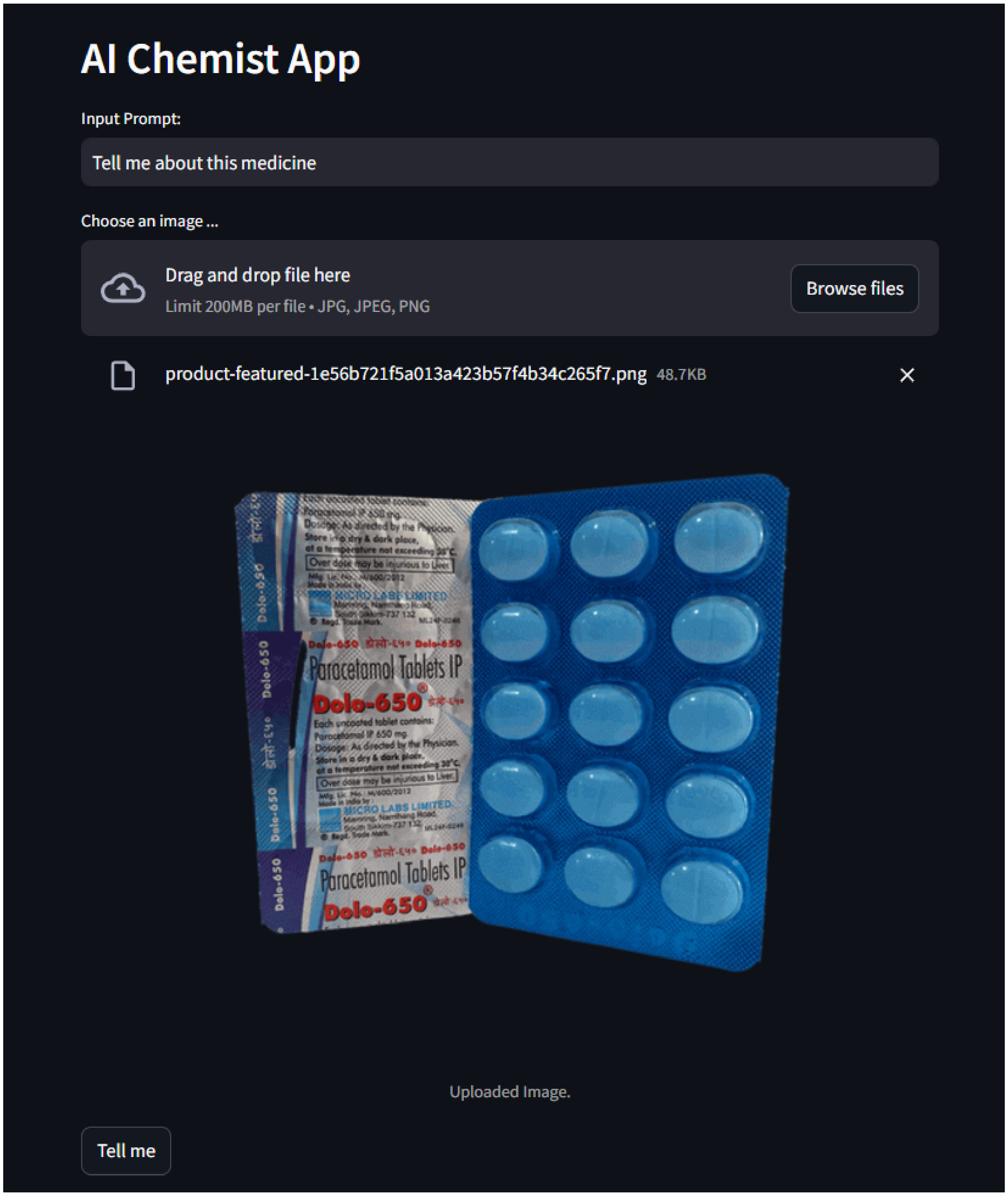
- **Respiratory infections:** such as bronchitis, pneumonia, sinusitis, and strep throat.
- **Skin infections:** such as impetigo and cellulitis.
- **Ear infections:** especially in children.
- **Sexually transmitted infections (STIs):** such as chlamydia and gonorrhea.
- **Other infections:** such as certain types of traveler's diarrhea and Lyme disease.

Important considerations:

- This is a prescription medication, so you should **only take it if prescribed by a doctor.** They will determine the proper dosage and duration of treatment based on your specific infection.
- **Do not self-medicate.** Misuse of antibiotics can lead to antibiotic resistance.
- Be sure to **inform your doctor** about any other medications you are taking, as well as any allergies you have.
- **Follow the instructions** provided by your doctor and pharmacist carefully.
- **Common side effects** may include nausea, vomiting, diarrhea, abdominal pain, and headache. More serious side effects are rare but possible. Consult your doctor if you experience any unusual or bothersome side effects.

While the box includes text in Devanagari script (अॅझिसिप ५०० - Aazisip 500), indicating it's likely marketed in India, the primary language is English, with the generic name (Azithromycin) clearly stated. Always consult a healthcare professional for accurate medical advice and information about medications.

Input 2:



AI Chemist App

Input Prompt:

Tell me about this medicine

Choose an image ...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

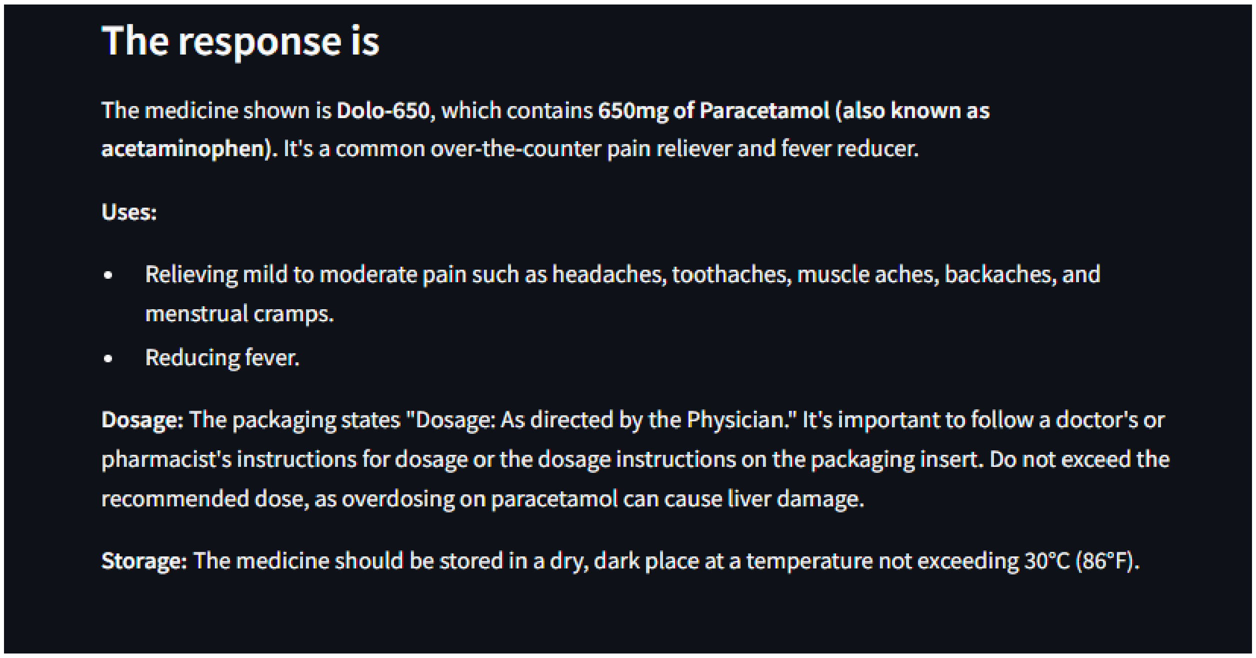
Browse files

product-featured-1e56b721f5a013a423b57f4b34c265f7.png 48.7KB

Uploaded Image.

Tell me

Output 2:



The response is

The medicine shown is **Dolo-650**, which contains **650mg of Paracetamol (also known as acetaminophen).** It's a common over-the-counter pain reliever and fever reducer.

Uses:

- Relieving mild to moderate pain such as headaches, toothaches, muscle aches, backaches, and menstrual cramps.
- Reducing fever.

Dosage: The packaging states "Dosage: As directed by the Physician." It's important to follow a doctor's or pharmacist's instructions for dosage or the dosage instructions on the packaging insert. Do not exceed the recommended dose, as overdosing on paracetamol can cause liver damage.

Storage: The medicine should be stored in a dry, dark place at a temperature not exceeding 30°C (86°F).

Conclusion :-

AI Chemist represents a groundbreaking leap in the intersection of artificial intelligence and chemical research. By leveraging the sophisticated capabilities of the **Gemini Vision Pro** model, it transforms how chemists and researchers approach experimentation, chemical synthesis, and data analysis. The app's ability to provide personalized, data-driven recommendations, generate optimized experimental designs, and propose innovative chemical synthesis routes opens new frontiers in scientific discovery.

Through intelligent processing and real-time guidance, AI Chemist enhances research efficiency, reduces trial-and-error, and fosters innovation across a wide range of chemical disciplines. By integrating cutting-edge AI with the practical needs of modern researchers, AI Chemist not only accelerates the pace of scientific advancements but also democratizes access to powerful computational tools, enabling both academic and industrial labs to achieve more with less effort and cost.

As a result, AI Chemist is poised to redefine the future of chemical science, empowering researchers to push the boundaries of what is possible, explore uncharted chemical territories, and ultimately contribute to innovations that can benefit industries ranging from pharmaceuticals to materials science and beyond. With its ability to merge AI with chemistry, AI Chemist is set to lead the way in advancing the next generation of research, innovation, and discovery.