Problem Statement 1

1. API Data Retrieval and Storage: You are tasked with fetching data from an external REST API, storing it in a local SQLite database, and displaying the retrieved data. The API provides a list of books in JSON format with attributes like title, author, and publication year.

The objective of this problem is used to save the book information data in the sqlite.I am assuming the as said in the problem to use any external Rest API,I will use publicly available api of the OpenLibrary as the information required is available there and also for the information i used some search engines instead of blind copy paste.I will break the problem codebase into multiple modules
api_client.py:- connecting external api and normalising the information in data manner
database.py-managing database operations
main.py-the main module controlling flow of operation

2.Data Processing and Visualization: Given a dataset containing information about students' test scores, fetch the data from an API, calculate the average score, and create a bar chart to visualize the data.

The objective of the task is to fetch the student information data from api then calculate the avg score and then creating bar charts to visualize ,I will use here publicly available Mocki data.The pipeline will for the program is data api fetching,data validation,average calculation then visualization.I will break the problem in multiple modules for readability

Api_client.py :- fetching and normalising datas
processor.py:- computational of average
visualizer.py:- to do the visualization in bar and plots
main.py:-main module controlling the flow

We are using the api available
https://dummyjson.com/products

We treat rating as a “test score”
We treat title as a “student identifier”

As it is a products rating the students name might appear as product one as we are using it as the names

3. CSV Data Import to a Database: Write a Python script that reads data from a CSV file containing user information (e.g., name, email) and inserts it into a SQLite database.

Ans:- Since a CSV dataset file was not provided in the task given,i created a sample CSV data of name and email. The pipeline which it follows, data from this CSV was read and inserted into a SQLite database. This demonstrates the complete CSV-to-database workflow in a simple way and as mentioned in task.
create_csv.py:- to create a dummy csv with name and email
main.py- main module which does the operation of reading the csv and inserting in database

4. Send a link to the most complex Python code you have written

https://github.com/piyush182004/Learnytics-Assists

5. Send a link to the most complex database code you have written

https://github.com/piyush182004/DATABASE-PROJECT

Problem statement- assignment 2

1. Where would you rate yourself on (LLM, Deep Learning, AI, ML). A, B, C [A = can code independently; B = can code under supervision; C = have little or no understanding]

I rate myself B can code under supervision,because in current situation everyone wants their production and work efficient and faster so i use ai models like copilot for the coding part,but the thing which matters is how i can understand the system workflow and pipelines,AI is there to assist me but the real production and understanding should be done by me and also libraries and models changes gets updated which also changes the code from the previous ones often times we are not able to do those ,and speaking of concepts there's always some new researches and topics that comes in this new era of AI/ML,Deep Learning,LLM,so i rate myself ‘B’ for the both coding and understanding ,I at my current stage will prefer a senior person to help me to understand new topics pipelines and systematic workflow and AI assisting coding agent to help me to code faster and efficient way.

2. What are the key architectural components to create a chatbot based on LLM? Please explain the approach on a high-level

The key architectural components to create a chatbot based on LLMs are:-

1)UI:- the frontend interface where user can write down their questions and the system then sends it to the backend to function.
2)Processing Layer:- where the it functions to clean the text like removing whitespaces making the sentence simpler standardize.
3)orchestration layer:-it works as decision maker which it analyzes the questions and understand the type of conversation ,and makes decision whether the answer present in internal documents
4)embedding model:- here the user question gets converted into the embedding which are basically set of numbers,these embedding helps to represent the meaning of the question
5)vector database:-these databases help to store,manage and search high dimensional embeddings,after the embedding model the vector database searches for sections where the embeddings are type of similar to the user question.
6)LLM core:- after the question goes through all the stages ,the llm receives the user question along with the context retrieved from the vector database it generates a response to it.
7)post response processing:- here it checks the tone,safety and clarity of the response generated
8)the final output gets shown in the ui to the user

3. Please explain vector databases. If you were to select a vector database for a hypothetical problem (you may define the problem) which one will you choose, and why?

Ans:- Vector databases help to store,manage,search high dimensional arrays called embeddings or vectors. In older systems like SQL like it used to use text matching keywords to search but in vector database it uses semantic like search based on similarity .It uses mathematic applications like:-
Cosine similarity
Euclidean distance
Dot product

The vector database has components like:-

Embeddings:- the numerical representation of the text,image or audio or any kind of data.It helps of capture similarity.

Indexing Algorithm:- To get the Approximate Nearest Neighbour using algorithms like HNSW (Hierarchical Navigable Small World graphs),IVF (Inverted File Index),PQ (Product Quantization) to do the search faster and efficient .

Metadata Filtering:-It stores the extra information along side with each vectors,for example tags categories,topics .It helps to filter results,narrow down search,make the responses more relevant.

Low Latency Retrieval:- It optimizes for fast similarity search in the vectors set,to execute for faster response.

If I have to select a vector database for any hypothetical situation or task I will choose the FAISS ((Facebook AI Similarity Search) which is an open source vector search library because :-

- It is fast and efficient,It can deliver very fast compared to other vector databases.
- It has full control over indexing and storage.
- It is lightweight and serverless,which can work locally in our own server.

<u>Simple Hypothetical situation</u>
Suppose our vector database has 3 sentences along with the vectors:-

AccuKnox is a cloud native security platform.
[100,101,102,....]

AccuKnox incorporates AI security for modern threats ,and is recognised in top AI security startup.
[110,115,120…]

The top trending anime is Jujutsu Kaisen.
[300,305,320,...]

Now we have these vectors in FAISS

<u>User searches for a query</u>
What is AccuKnox?

This query gets converted into vectors:-

[98,96,99….]

How FAISS works?
Now FAISS doesn't look at word.It calculates distance between query vectors and stored vectors.

- Sentence 1 closer to query
- Sentence 2 closer to query
- Sentence 3 not at all closer to query

Result:-
FAISS returns the first two words:-
AccuKnox is a cloud native security platform.
AccuKnox incorporates AI security for modern threats ,and is recognised in top AI security startup.

And ignores the word:-
The top trending anime is Jujutsu Kaisen.