QUESTION : 4

**1. Data Flow Diagram**

The data flow diagram will illustrate how the application interacts with the external COVID-19 statistics API and how the data is processed and displayed to the user.

**Key Components:**

* **User Input:** The user specifies the region (country, state, or city).
* **API Request:** The application sends a request to the COVID-19 statistics API with the specified region.
* **API Response:** The API returns the latest data on COVID-19 cases, recoveries, and deaths.
* **Data Processing:** The application processes the response data and extracts the relevant information.
* **Display:** The processed data is displayed to the user in a user-friendly format.

**2. Pseudocode and Implementation**

**Pseudocode:**

BEGIN

Display a prompt for the user to input a region (country, state, or city)

User inputs the region

Send an API request to the COVID-19 statistics API with the specified region

IF the API request is successful THEN

Extract the number of cases, recoveries, and deaths from the API response

Display the COVID-19 statistics to the user

ELSE

Display an error message indicating that the data could not be retrieved

END IF

END

**Implementation (Python):**

import requests

def fetch\_covid\_stats(region):

url = f"https://disease.sh/v3/covid-19/{region}"

response = requests.get(url)

if response.status\_code == 200:

data = response.json()

cases = data.get('cases')

recoveries = data.get('recovered')

deaths = data.get('deaths')

return cases, recoveries, deaths

else:

return None

def display\_stats(region):

stats = fetch\_covid\_stats(region)

if stats:

cases, recoveries, deaths = stats

print(f"COVID-19 Statistics for {region}:")

print(f"Cases: {cases}")

print(f"Recoveries: {recoveries}")

print(f"Deaths: {deaths}")

else:

print(f"Could not retrieve data for {region}.")

if \_\_name\_\_ == "\_\_main\_\_":

region = input("Enter the region (country/state/city): ")

display\_stats(region)

**3. Documentation**

The documentation will include:

* **API Integration:** A detailed explanation of how the application interacts with the COVID-19 statistics API, including the endpoints used and the structure of the API requests and responses.
* **Data Fetching:** The method used to fetch data from the API, including error handling and response validation.
* **Data Display:** How the data is processed and formatted before being displayed to the user.

**4. Assumptions and Potential Improvements**

**Assumptions:**

* The API is reliable and provides up-to-date data.
* The user inputs a valid region name that the API can recognize.

**Potential Improvements:**

* **Error Handling:** Improve error handling by providing more detailed error messages and suggestions for correcting user input.
* **Caching:** Implement caching to reduce the number of API requests and improve performance.
* **Visualization:** Add data visualization features, such as graphs or charts, to provide a more interactive experience.
* **Localization:** Support multiple languages for displaying the COVID-19 statistics.