**Q: Hospital management system - Design a class to manage patient records, including admissions, treatments, and discharge.**

**Approach to solve:**

1.Understand the problem statement clearly and find out the all the end points need to be created .

2.Figure out the proper datasturctures to manipulate the data in the api

3.Figure out the database design with proper list of tables and appropriate constraints to solve the problem.

**Pseudocode:**

**ADMIT\_PATIENT (HTTP POST method):**

Step 1: Extract the required data from the JSON payload.

Step 2: If the data format is incorrect, display a message saying "Invalid request format."

Step 3: Create a database connection using the set\_connection() method. If the connection is created successfully, insert the extracted data into the "patients" table and save the changes. If an exception is raised, roll back the transaction and display a message saying "Failed to admit the patient."

Step 4: Return a JSON response with the newly created patient's ID and a success message.

**GET\_ADMISSIONS (HTTP GET method):**

Step 1: Create a database connection using the set\_connection() method. If the connection is created successfully, execute a SELECT query to retrieve all the patients from the "patients" table that have not been discharged yet. If an exception is raised, display a message saying "Failed to retrieve patient admissions."

Step 2: Loop through the retrieved patients and create a list of dictionaries containing the patient's ID, name, date of birth, gender, and admission date.

Step 3: Return a JSON response with the list of patient admissions.

**ADD\_TREATMENT (HTTP POST method):**

Step 1: Extract the required data from the JSON payload.

Step 2: If the data format is incorrect, display a message saying "Invalid request format."

Step 3: Create a database connection using the set\_connection() method. If the connection is created successfully, execute a SELECT query to check if the patient with the given ID exists and has not been discharged yet. If the patient is found, insert the extracted data into the "treatments" table and save the changes. If an exception is raised, roll back the transaction and display a message saying "Failed to add the treatment."

Step 4: Return a JSON response with a success message.

**DISCHARGE\_PATIENT (HTTP PUT method):**

Step 1: Extract the required data from the JSON payload.

Step 2: If the data format is incorrect, display a message saying "Invalid request format."

Step 3: Create a database connection using the set\_connection() method. If the connection is created successfully, execute a SELECT query to check if the patient with the given ID is currently admitted and has not been discharged yet. If the patient is found, update the admission record in the "admissions" table with the discharge date and diagnosis, and save the changes. If an exception is raised, roll back the transaction and display a message saying "Failed to discharge the patient."

Step 4: Return a JSON response with a success message.

**GET\_PATIENT\_BY\_ID (HTTP GET method):**

Step 1: Create a database connection using the set\_connection() method. If the connection is created successfully, execute a SELECT query to retrieve the patient with the given ID from the "patients" table. If the patient is not found, raise a PatientNotFound exception.

Step 2: Execute a SELECT query to retrieve all the admission records for the patient from the "admissions" table. If no admission records are found, return a JSON response with just the patient's ID, name, and date of birth.

Step 3: Loop through the admission records and create a list of dictionaries containing the admission ID, admission date, discharge date, and diagnosis.

Step 4: Add the list of admission records to a dictionary containing the patient's ID, name, and date of birth.

Step 5: Return a JSON response with the dictionary containing the patient's information and admission records.