

Week 3 Project Report
Database (MII212501)



**PawPoint: Veterinary Clinic Appointment Scheduling and Management
Application**

Written By:

Deira Aisya Refani (24/532821/PA/22539)

Ratu Faiha Salsabilla (24/532756/PA/22533)

Aufaa Azzahra Aryawan (24/546854/PA/23215)

Universitas Gadjah Mada

Faculty Of Mathematics and Natural Science

Department of Computer Science and Electronics

2025

1. User CRUD

```
PS C:\Users\rfsr2> cd C:\Users\rfsr2\Downloads\dbProject
PS C:\Users\rfsr2\Downloads\dbProject> $headers = @{ "Content-Type" = "application/json" }
PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   username = "faiha"
>>   password = "123"
>>   role = "pet_owner"
>>   email = "faiha@example.com"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/users" -Method Post -Headers $headers -Body $body

message
=====
User created

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/users" -Method Get

created_at : Thu, 20 Nov 2025 20:46:30 GMT
email      : faiha@example.com
password   : 123
role       : pet_owner
user_id    : 1
username   : faiha
```

User CRUD manages the users of the PawPoint system – admin, staff.

Create (POST /users)

- Receives JSON with username, password, role, email
- Inserts a new record into the *User* table
- Returns a success message and HTTP status 201

```
@app.post("/users")
def create_user():
    data = request.json
    query = """
        INSERT INTO User (username, password, role, email)
        VALUES (%s, %s, %s, %s)
    """
    conn = get_connection()
    with conn:
        with conn.cursor() as cur:
            cur.execute(query, (data["username"], data["password"], data["role"], data.get("email")))
            conn.commit()
    return jsonify({"message": "User created"}), 201
```

Read (GET /users & GET /users/<user_id>)

- Returns all users or a specific user by ID
- Queries *User* table and sends results as JSON

```
@app.get("/users")
def get_users():
    conn = get_connection()
    with conn:
        with conn.cursor() as cur:
            cur.execute("SELECT * FROM User")
            return jsonify(cur.fetchall())

@app.get("/users/<int:user_id>")
def get_user(user_id):
    conn = get_connection()
    with conn:
        with conn.cursor() as cur:
            cur.execute("SELECT * FROM User WHERE user_id=%s", (user_id,))
            return jsonify(cur.fetchone())
```

```

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-WebRequest -Uri "http://127.0.0.1:5000/users" -Method Get -Headers $headers

created_at : Thu, 20 Nov 2025 20:52:08 GMT
email      : faiha@example.com
password   : 123
role       : pet_owner
user_id    : 2
username   : faiha

created_at : Thu, 20 Nov 2025 21:01:06 GMT
email      : faiha2@example.com
password   : 123
role       : pet_owner
user_id    : 6
username   : faiha2

created_at : Thu, 20 Nov 2025 21:40:20 GMT
email      : dr.smith@example.com
password   : vet123
role       : vet
user_id    : 7
username   : dr_smith

```

Update (PUT /users/<user_id>)

- Updates username, role, email for a given user_id
- Executes SQL *UPDATE* query and confirms update

```

@app.put("/users/<int:user_id>")
def update_user(user_id):
    data = request.json
    query = """
        UPDATE User SET username=%s, email=%s, role=%s
        WHERE user_id=%s
    """
    conn = get_connection()
    with conn:
        with conn.cursor() as cur:
            cur.execute(query, (data["username"], data.get("email"), data["role"], user_id))
            conn.commit()
            return jsonify({"message": "User updated"})

```

Delete (DELETE /users/<user_id>)

- Deletes the user with the given ID
- Executes SQL *DELETE* query and confirms deletion

```

@app.delete("/users/<int:user_id>")
def delete_user(user_id):
    conn = get_connection()
    with conn:
        with conn.cursor() as cur:
            cur.execute("DELETE FROM User WHERE user_id=%s", (user_id,))
            conn.commit()
            return jsonify({"message": "User deleted"})

```

```

PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   username = "faiha_updated"
>>   email = "faiha2@example.com"
>>   role = "pet_owner"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject>
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/users/1" -Method Put -Headers $headers -Body $body

message
-----
User updated

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/users/1" -Method Delete

message
-----
User deleted

```

2. Pet CRUD

Pet CRUD manages pets and are linked to their owners.

Create (POST /pets)

- Receives JSON with owner_id, name, species, breed, age, gender
- Inserts a new pet in the *Pet* table

```

PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>> name = "Fluffy"
>> species = "Dog"
>> breed = "Poodle"
>> age = 3
>> owner_id = 2
>> } | ConvertTo-Json

```

Read (GET /pets & GET /pets/<pet_id>)

- Returns all pets or a specific pet

Update (PUT /pets/<pet_id>)

- Updates the pet's information (name, species, breed, age, gender, owner_id)

```

PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>> name = "Fluffy_updated"
>> species = "Dog"
>> breed = "Poodle"
>> age = 4
>> owner_id = 2
>> } | ConvertTo-Json

```

Delete (DELETE /pets/<pet_id>)

- Deletes the pet record from the database

```
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/pets/1" -Method Put -Head
message
-----
Pet updated

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/pets" -Method Get
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/pets/1" -Method Delete
message
-----
Pet deleted
```

3. Veterinarian CRUD

Veterinarian creation and initial read

- Creation (POST):
 - A JSON body is prepared with the details for a new vet, including user_id = 7, name = “Dr.Smith”, phone = “0812345678”, specialization = “Surgery”, and a schedule.
 - Invoke-RestMethod is used with the POST method to <http://127.0.0.1:5000/vets>.
 - Result: the server returns the message Vet created
- Read (GET):
 - A subsequent call with the GET method to the same URL is executed to retrieve the list of vets.
 - Result: The output shows two vet records: the newly created one (vet_id: 2, user_id:7) and a previously existing one (vet_id: 1, user_id: 2).

```
PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   user_id = 7
>>   name = "Dr. Smith"
>>   phone = "0812345678"
>>   specialization = "Surgery"
>>   schedule = "Mon-Fri 9am-5pm"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets" -Method Post -H
message
-----
Vet created

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets" -Method Get -H
name           : Dr. Smith
phone          : 0812345678
schedule       : Mon-Fri 9am-5pm
specialization : Surgery
user_id        : 2
vet_id         : 1

name           : Dr. Smith
phone          : 0812345678
schedule       : Mon-Fri 9am-5pm
specialization : Surgery
user_id        : 7
vet_id         : 2
```

- Update (PUT)
 - A new JSON body is created with updated information for a vet, changing the name to “Dr. John Smith”, specialization to “Dermatology”, and updating the phone and schedule.
 - Invoke-RestMethod is used with the PUT method targeting the specific vet ID: <http://127.0.0.1:5000/vets/2>.
 - Result: The server confirms with the message Vet updated.
- Read (GET)
 - A final GET request confirms the update.
 - Result: The output now shows vet id: 2 has the updated name and specializaion: Dr. John Smith and Dermatology.

```
PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   name = "Dr. John Smith"
>>   phone = "0812345679"
>>   specialization = "Dermatology"
>>   schedule = "Mon-Fri 10am-6pm"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject>
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets/2" -Method Put -H

message
-----
Vet updated

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets" -Method Get

name           : Dr. Smith
phone          : 0812345678
schedule       : Mon-Fri 9am-5pm
specialization : Surgery
user_id        : 2
vet_id         : 1

name           : Dr. John Smith
phone          : 0812345679
schedule       : Mon-Fri 10am-6pm
specialization : Dermatology
user_id        : 7
vet_id         : 2
```

- Delete (DELETE)
 - Invoke-RestMethod is used with the DELETE method targeting <http://127.0.0.1:5000/vets/2>.
 - Result: The server confirms with the message Vet deleted.
- Read (READ)
 - A final GET request confirms the deletion.
 - Result: Only the original veterinarian (vet_id: 1, Dr. Smith) remains in the output.

```
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets/2" -Method Delete

message
-----
Vet deleted

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/vets" -Method Get

name           : Dr. Smith
phone          : 0812345678
schedule       : Mon-Fri 9am-5pm
specialization : Surgery
user_id        : 2
vet_id         : 1
```

4. Appointments CRUD

Clinic creation and appointment creation

- Clinic creation (POST)
 - A JSON body is prepared to create a clinic named Happy Pets Clinic.
 - Result: The server returns Clinic created.
 - A GET request confirms the clinic is created with clinic id: 1.
- Appointment Creation (POST)
 - The first appointment is created with details like pet_id, vet_id, clinic_id and notes="regular checkup".
 - Result: The server returns Appointment created.
 - A GET request shows this appointment with an appointment_id of 4 and others.

```
PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   name = "Happy Pets Clinic"
>>   address = "123 Pet Street"
>>   phone = "0812345679"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject> $headers = @{ "Content-Type" = "application/json" }
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/clinics" -Method Post -
message
-----
Clinic created

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/clinics" -Method Get
address      clinic_id name      phone
-----
123 Pet Street      1 Happy Pets Clinic 0812345679
```

Appointment update and second creation

- Update (PUT)
 - A JSON body is prepared to update appointment ID 4, changing the appointment_date and setting the notes to Vaccination.
 - The PUT method is used on <http://127.0.0.1:5000/appointments/4>.
 - Result: The server confirms Appointment updated.
 - A GET request confirms appointment ID 4 now has the notes Vaccination.
- Second creation (POST)
 - a second appointment is created with notes= "Follow-up" and a different date
 - Result: The server returns Appointment created.

```

PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   pet_id = 9
>>   vet_id = 9
>>   clinic_id = 1
>>   appointment_date = "2025-11-22 11:00:00"
>>   notes = "Vaccination"
>>   status = "Scheduled" # optional, but good to include
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/appointments/4" -Method Put -H

message
-----
Appointment updated

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/appointments" -Method Get

appointment_date : Sat, 22 Nov 2025 11:00:00 GMT
appointment_id   : 4
clinic_id       : 1
notes           : Vaccination
pet_id          : 9
status          : Scheduled
vet_id          : 9

```

Appointment deletion and final read

- Delete (DELETE):
 - The DELETE method is used to remove a specific appointment:
http://127.0.0.1:5000/appointments/1.
 - Result: The server confirms Appointment deleted.
- Read (GET):
 - The final GET request shows the remaining appointments.
 - Result: Two records remain: the updated Vaccination appointment (appointment_id: 4) and the new follow-up appointment (appointment_id: 5)

```

PS C:\Users\rfsr2\Downloads\dbProject> $body = @{
>>   pet_id = 9
>>   vet_id = 9
>>   clinic_id = 1
>>   appointment_date = "2025-11-23 10:00:00"
>>   notes = "Follow-up"
>>   status = "Scheduled"
>> } | ConvertTo-Json
PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/appointments" -Method Post -Headers $headers -Body $body

message
-----
Appointment created

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/appointments/1" -Method Delete

message
-----
Appointment deleted

PS C:\Users\rfsr2\Downloads\dbProject> Invoke-RestMethod -Uri "http://127.0.0.1:5000/appointments" -Method Get

appointment_date : Sat, 22 Nov 2025 11:00:00 GMT
appointment_id   : 4
clinic_id       : 1
notes           : Vaccination
pet_id          : 9
status          : Scheduled
vet_id          : 9

appointment_date : Sun, 23 Nov 2025 10:00:00 GMT
appointment_id   : 5
clinic_id       : 1
notes           : Follow-up
pet_id          : 9
status          : Scheduled
vet_id          : 9

```