

Based on the following code create code that will have it shown in html

```
# Import necessary modules and models from sqlalchemy
from sqlalchemy import create_engine, func
from sqlalchemy.orm import sessionmaker
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy import Column, Integer, String, Float, Date, ForeignKey
from sqlalchemy.orm import relationship

Base = declarative_base()

class PetOwner(Base):
    __tablename__ = 'pet_owners'
    id = Column(Integer, primary_key=True)
    name = Column(String(50))
    address = Column(String(100))
    phone = Column(String(20))

class Pet(Base):
    __tablename__ = 'pets'
    id = Column(Integer, primary_key=True)
    name = Column(String(50))
    weight = Column(Float)
    height = Column(Float)
    owner_id = Column(Integer, ForeignKey('pet_owners.id'))
    owner = relationship('PetOwner', backref='pets')

class Veterinarian(Base):
    __tablename__ = 'veterinarians'
    id = Column(Integer, primary_key=True)
    name = Column(String(50))
    address = Column(String(100))
    phone = Column(String(20))

class Visit(Base):
    __tablename__ = 'visits'
    id = Column(Integer, primary_key=True)
    pet_id = Column(Integer, ForeignKey('pets.id'))
    vet_id = Column(Integer, ForeignKey('veterinarians.id'))
    date = Column(Date)
    pet = relationship('Pet', backref='visits')
    vet = relationship('Veterinarian', backref='visits')

class Billing(Base):
    __tablename__ = 'bills'
    id = Column(Integer, primary_key=True)
    pet_id = Column(Integer, ForeignKey('pets.id'))
    vet_id = Column(Integer, ForeignKey('veterinarians.id'))
    amount = Column(Float)
    address = Column(String(100))
    insurance = Column(String(50))
    visit_id = Column(Integer, ForeignKey('visits.id'))
    pet = relationship('Pet', backref='bills')
    vet = relationship('Veterinarian', backref='bills')

# Create the database engine and session
engine = create_engine('sqlite:///DBproj.db')
Session = sessionmaker(bind=engine)
session = Session()

# Define the command-line menu
def main_menu():
    while True:
        print("\n--- Veterinary Clinic Management System ---")
        print("1. View pet visit history")
        print("2. View billing information for a pet owner")
        print("3. Add a new pet")
        print("4. Schedule a visit")
        print("5. Exit")
        choice = input("Enter your choice (1-5): ")
        if choice == '1':
            view_visit_history()
        elif choice == '2':
            view_billing_info()
        elif choice == '3':
            add_new_pet()
        elif choice == '4':
            schedule_visit()
        elif choice == '5':
            print("Exiting the system.")
            break
        else:
            print("Invalid choice. Please select a valid option.")

def view_visit_history():
    pet_id_input = input("Enter the Pet ID: ").strip()
    try:
        pet_id = int(pet_id_input)
    except ValueError:
        print("Invalid Pet ID. Please enter numbers only.")
        return
    visit = session.query(Visit).filter_by(petID=pet_id).all()
    if not visit:
        print("No visits found for this pet.")
    for visit in visit:
        vet = session.query(Veterinarian).filter_by(vetID=visit.vetID).first()
        print(f"Visit ID: {visit.visitID}, Vet: {vet.vetName}, Date: {visit.date}")

def view_billing_info():
    owner_id_input = input("Enter the Owner ID: ").strip()
    try:
        owner_id = int(owner_id_input)
    except ValueError:
        print("Invalid Owner ID. Please enter numbers only.")
        return
    bills = session.query(Billing).filter_by(ownerID=owner_id).all()
    if not bills:
        print("No billing records found for this owner.")
    for bill in bills:
        print(f"Receipt ID: {bill.receiptID}, Amount: ${bill.billAmount}, Address: {bill.billAddress}, Insurance: {bill.billInsurance}, Visit ID: {bill.visitID}")

def add_new_pet():
    name = input("Enter the pet's name: ").strip()
    weight_input = input("Enter the pet's weight: ").strip()
    height_input = input("Enter the pet's height: ").strip()
    owner_id_input = input("Enter the owner's ID: ").strip()
    try:
        weight = float(weight_input)
        height = float(height_input)
        owner_id = int(owner_id_input)
    except ValueError:
        print("Invalid input. Weight and height should be numbers, owner ID should be an integer.")
        return
    owner = session.query(PetOwner).filter_by(ownerID=owner_id).first()
    if not owner:
        print("Owner not found.")
        return
    pet = Pet(petName=name, petWeight=weight, petHeight=height, ownerID=owner_id)
    session.add(pet)
    session.commit()
    print(f"Pet {name} added successfully.")

def schedule_visit():
    pet_id_input = input("Enter the Pet ID: ").strip()
    vet_id_input = input("Enter the Vet ID: ").strip()
    date = input("Enter the visit date (YYYY-MM-DD HH:MM:SS): ").strip()
    try:
        pet_id = int(pet_id_input)
        vet_id = int(vet_id_input)
    except ValueError:
        print("Invalid input. Pet ID and Vet ID should be integers.")
        return
    pet = session.query(Pet).filter_by(petID=pet_id).first()
    vet = session.query(Veterinarian).filter_by(vetID=veterinarian_id).first()
    if not pet:
        print("Pet not found.")
    if not vet:
        print("Veterinarian not found.")
    visit = Visit(petID=pet_id, vetID=veterinarian_id, date=date)
    session.add(visit)
    session.commit()
    print(f"Visit scheduled for pet ID {pet_id} with vet ID {vet_id} on {date}.")

# Run the main menu if __name__ == "__main__":
if __name__ == "__main__":
    main_menu()
```

ChatGPT said:

To display this veterinary clinic system functionality in **HTML** (using a web interface instead of a command-line menu), you'll need to use a web framework like **Flask**.

Below is a basic **Flask app** that replicates the functionality of your CLI app and renders it as HTML:

---

## 1. Install Flask

Open your VS Code terminal and run:

```
bash
Copy code
pip install flask
```

---

## 2. Create app.py

This is your web-based version of the veterinary clinic system:

```
python
Copy code
from flask import Flask, render_template, request, redirect, url_for
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from alchemyBase import Base, Pet, PetOwner, Clinic, Company, Veterinarian,
Visit, Billing

app = Flask(__name__)

# Database setup
engine = create_engine('sqlite:///DBproj.db')
Session = sessionmaker(bind=engine)
session = Session()

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/visit_history', methods=['GET', 'POST'])
def visit_history():
    if request.method == 'POST':
        pet_id = request.form.get('pet_id')
        visits = session.query(Visit).filter_by(petID=pet_id).all()
        return render_template('visit_history.html', visits=visits,
pet_id=pet_id)
    return render_template('visit_history.html', visits=None)

@app.route('/billing_info', methods=['GET', 'POST'])
def billing_info():
    if request.method == 'POST':
        owner_id = request.form.get('owner_id')
        bills = session.query(Billing).filter_by(ownerID=owner_id).all()
        return render_template('billing_info.html', bills=bills,
owner_id=owner_id)
```

```

        return render_template('billing_info.html', bills=None)

@app.route('/add_pet', methods=['GET', 'POST'])
def add_pet():
    if request.method == 'POST':
        name = request.form['pet_name']
        weight = float(request.form['pet_weight'])
        height = float(request.form['pet_height'])
        owner_id = int(request.form['owner_id'])

        owner = session.query(PetOwner).filter_by(ownerID=owner_id).first()
        if not owner:
            return "Owner not found."

        new_pet = Pet(petName=name, petWeight=weight, petHeight=height,
ownerID=owner_id)
        session.add(new_pet)
        session.commit()
        return redirect(url_for('home'))

    return render_template('add_pet.html')

@app.route('/schedule_visit', methods=['GET', 'POST'])
def schedule_visit():
    if request.method == 'POST':
        pet_id = int(request.form['pet_id'])
        vet_id = int(request.form['vet_id'])
        date = request.form['date']

        pet = session.query(Pet).filter_by(petID=pet_id).first()
        vet = session.query(Veterinarian).filter_by(vetID=vet_id).first()

        if not pet or not vet:
            return "Invalid Pet or Vet ID."

        new_visit = Visit(petID=pet_id, vetID=vet_id, date=date)
        session.add(new_visit)
        session.commit()
        return redirect(url_for('home'))

    return render_template('schedule_visit.html')

if __name__ == '__main__':
    app.run(debug=True)

```

---

### 3. Create HTML Templates

Inside a `templates/` folder, create these files:

**home.html**

```

html
Copy code
<h1>Veterinary Clinic Management</h1>

```

```

<ul>
  <li><a href="{{ url_for('visit_history') }}">View Pet Visit
History</a></li>
  <li><a href="{{ url_for('billing_info') }}">View Billing Info</a></li>
  <li><a href="{{ url_for('add_pet') }}">Add New Pet</a></li>
  <li><a href="{{ url_for('schedule_visit') }}">Schedule Visit</a></li>
</ul>

```

### **visit\_history.html**

```

html
Copy code
<h2>Visit History</h2>
<form method="POST">
  <input type="text" name="pet_id" placeholder="Enter Pet ID" required>
  <input type="submit" value="Search">
</form>

{% if visits %}
  <h3>Visits for Pet ID {{ pet_id }}</h3>
  <ul>
    {% for visit in visits %}
      <li>Visit ID: {{ visit.visitID }}, Vet ID: {{ visit.vetID }}, Date:
{{ visit.date }}</li>
    {% endfor %}
  </ul>
{% elif visits is not none %}
  <p>No visits found.</p>
{% endif %}

```

### **billing\_info.html**

```

html
Copy code
<h2>Billing Information</h2>
<form method="POST">
  <input type="text" name="owner_id" placeholder="Enter Owner ID" required>
  <input type="submit" value="Search">
</form>

{% if bills %}
  <h3>Billing for Owner ID {{ owner_id }}</h3>
  <ul>
    {% for bill in bills %}
      <li>Receipt ID: {{ bill.receiptID }}, Amount: {{ bill.billAmount }},
Insurance: {{ bill.billInsurance }}</li>
    {% endfor %}
  </ul>
{% elif bills is not none %}
  <p>No billing records found.</p>
{% endif %}

```

### **add\_pet.html**

```

html

```

Copy code

```
<h2>Add New Pet</h2>
<form method="POST">
  <input type="text" name="pet_name" placeholder="Pet Name" required><br>
  <input type="number" step="any" name="pet_weight" placeholder="Weight"
required><br>
  <input type="number" step="any" name="pet_height" placeholder="Height"
required><br>
  <input type="number" name="owner_id" placeholder="Owner ID" required><br>
  <input type="submit" value="Add Pet">
</form>
```

### **schedule\_visit.html**

html

Copy code

```
<h2>Schedule Visit</h2>
<form method="POST">
  <input type="number" name="pet_id" placeholder="Pet ID" required><br>
  <input type="number" name="vet_id" placeholder="Vet ID" required><br>
  <input type="text" name="date" placeholder="YYYY-MM-DD HH:MM:SS"
required><br>
  <input type="submit" value="Schedule Visit">
</form>
```

---

Would you like a ZIP of the full working project structure to import into VS Code?