

NextgenLead_4

app.py

```
import psycopg2
from flask import Flask, request, jsonify
from flask_restful import Api
from sqlalchemy import Column, String, Integer, Date, BOOLEAN, BIGINT, Text, create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker
from sqlalchemy.pool import NullPool

app = Flask(__name__)
api = Api(app)

# Database setup
Base = declarative_base()
database_url = "postgresql://postgres:1234@localhost:5432/postgres"

# Disable SQLAlchemy pool using NullPool
engine = create_engine(database_url, echo=True, poolclass=NullPool)
Session = sessionmaker(bind=engine)
session = Session()

# Define the ProductEnquiryForms_4 model
class ProductEnquiryForms_4(Base):
    __tablename__ = 'productenquiryforms_4'

    CustomerName = Column("customername", String)
    Gender = Column("gender", String)
    Age = Column("age", Integer)
    Occupation = Column("occupation", String)
    MobileNo = Column("mobilenumber", BIGINT, primary_key=True)
    Email = Column("email", String)
    VehicleModel = Column("vehiclemodel", String)
    State = Column("state", String)
    District = Column("district", String)
    City = Column("city", String)
    ExistingVehicle = Column("existingvehicle", String)
    DealerState = Column("dealerstate", String)
    DealerTown = Column("dealertown", String)
    Dealer = Column("dealer", String)
    BriefAboutEnquiry = Column("briefaboutenquiry", Text)
    ExpectedDateOfPurchase = Column("expecteddateofpurchase", Date)
    IntendedUsage = Column("intendedusage", String)
    SentToDealer = Column("senttodealer", BOOLEAN)
    DealerCode = Column("dealercode", String)
    LeadId = Column("leadid", String)
    Comments = Column("comments", Text)
    CreatedDate = Column("createddate", Date)
    IsPurchased = Column("ispurchased", BOOLEAN)

# Define the CustomerDetails_4 model
```

```

class CustomerDetails_4(Base):
    __tablename__ = 'customerdetails_4'

    LeadId = Column("leadid", String)
    CustomerName = Column("customername", String)
    MobileNo = Column("mobilen", BIGINT, primary_key=True)
    City = Column("city", String)
    Dealer = Column("dealer", String)
    DealerCode = Column("dealercode", String)
    SentToDealer = Column("senttodealer", BOOLEAN)

# Create the tables in PostgreSQL
Base.metadata.create_all(engine)

# Route to post records to both tables
@app.route('/post_records', methods=['POST'])
def post_records():
    data = request.get_json()
    try:
        # Add record to ProductEnquiryForms_4
        product_record = ProductEnquiryForms_4(
            CustomerName=data['customername'],
            Gender=data['gender'],
            Age=data['age'],
            Occupation=data['occupation'],
            MobileNo=data['mobilen'],
            Email=data['email'],
            VehicleModel=data['vehiclemodel'],
            State=data['state'],
            District=data['district'],
            City=data['city'],
            ExistingVehicle=data['existingvehicle'],
            DealerState=data['dealerstate'],
            DealerTown=data['dealertown'],
            Dealer=data['dealer'],
            BriefAboutEnquiry=data['briefaboutenquiry'],
            ExpectedDateOfPurchase=data['expecteddateofpurchase'],
            IntendedUsage=data['intendedusage'],
            SentToDealer=data['senttodealer'],
            DealerCode=data['dealercode'],
            LeadId=data['leadid'],
            Comments=data['comments'],
            CreatedDate=data['createddate'],
            IsPurchased=data['ispurchased']
        )
        session.add(product_record)

        # Add record to CustomerDetails_4
        customer_record = CustomerDetails_4(
            LeadId=data['customer_leadid'],
            CustomerName=data['customername'],
            MobileNo=data['mobilen'],
            City=data['customer_city'],
            Dealer=data['customer_dealer'],
            DealerCode=data['customer_dealercode'],

```

```

        SentToDealer=data['customer_senttodealer']
    )
    session.add(customer_record)

    session.commit()
    return jsonify({"message": "Records added successfully!"}), 201
except Exception as e:
    session.rollback()
    return jsonify({"error": str(e)}), 500

# Route to get all records from both tables
@app.route('/get_records', methods=['GET'])
def get_records():
    try:
        product_records = session.query(ProductEnquiryForms_4).all()
        customer_records = session.query(CustomerDetails_4).all()

        product_list = [record.__dict__ for record in product_records]
        customer_list = [record.__dict__ for record in customer_records]

        return jsonify({
            "ProductEnquiryForms_4": product_list,
            "CustomerDetails_4": customer_list
        }), 200
    except Exception as e:
        return jsonify({"error": str(e)}), 500

# Route to update a record in ProductEnquiryForms_4
@app.route('/update_record/<int:mobile_no>', methods=['PUT'])
def update_record(mobile_no):
    data = request.get_json()
    try:
        record = session.query(ProductEnquiryForms_4).filter_by(MobileNo=mobile_no).fi
        if record:
            for key, value in data.items():
                setattr(record, key, value)
            session.commit()
            return jsonify({"message": "Record updated successfully!"}), 200
        else:
            return jsonify({"message": "Record not found!"}), 404
    except Exception as e:
        session.rollback()
        return jsonify({"error": str(e)}), 500

# Route to delete a record from ProductEnquiryForms_4
@app.route('/delete_record/<int:mobile_no>', methods=['DELETE'])
def delete_record(mobile_no):
    try:
        record = session.query(ProductEnquiryForms_4).filter_by(MobileNo=mobile_no).fi
        if record:
            session.delete(record)
            session.commit()
            return jsonify({"message": "Record deleted successfully!"}), 200
        else:
            return jsonify({"message": "Record not found!"}), 404

```

```

except Exception as e:
    session.rollback()
    return jsonify({"error": str(e)}), 500

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

```

for post method
json data

```

{
    "customername": "Radha",
    "gender": "Female",
    "age": 25,
    "occupation": "Teacher",
    "mobilenumber": 9876543210,
    "email": "radha@example.com",
    "vehiclemodel": "Model Y",
    "state": "Uttar Pradesh",
    "district": "Mathura",
    "city": "Vrindavan",
    "existingvehicle": "Scooter",
    "dealerstate": "Uttar Pradesh",
    "dealertown": "Mathura",
    "dealer": "Dealer A",
    "briefaboutenquiry": "Interested in a car",
    "expecteddateofpurchase": "2024-11-15",
    "intendedusage": "Personal",
    "senttodealer": false,
    "dealercode": "VRN123",
    "leadid": "LEAD_RADHA",
    "comments": "Looking for an affordable car",
    "createddate": "2024-09-07",
    "ispurchased": false,

    "customer_leadid": "LEAD_RADHA",
    "customer_city": "Vrindavan",
    "customer_dealer": "Dealer A",
    "customer_dealercode": "VRN123",
    "customer_senttodealer": false
}

```

- **GET /get_records**
- **URL:** `http://localhost:5000/get_records`
 - Fetches all records from `productenquiryforms_4` and `customerdetails_4`.
- **PUT /update_record/<mobilenumber>**
- **URL:** `http://localhost:5000/update_record/9876543210`

```
{  
  "CustomerName": "New Radha"  
}
```

- **DELETE** /delete_record/<mobilen>
- **URL:** http://localhost:5000/delete_record/9876543210