**PRACTICAL NO 3**

**AIM:** Create a Simple REST Service.

**SOLUTION:**

**A) Simple Rest Service**

**INPUT:**

from flask import Flask, request, jsonify

app = Flask(\_\_name\_\_)

@app.route('/add', methods=['GET'])

def add\_numbers():

    num1 = float(request.args.get('num1'))

    num2 = float(request.args.get('num2'))

    # Calculate the sum

    result = num1 + num2

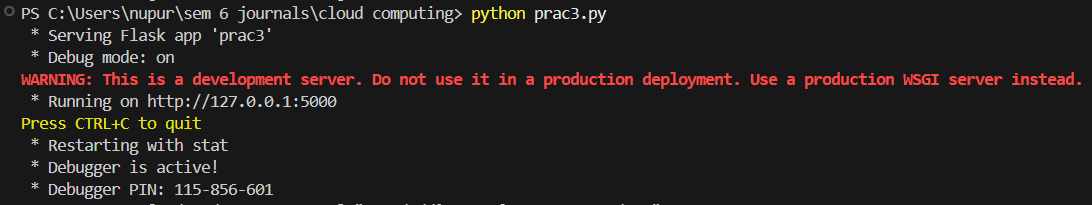
    # Return the result as JSON

    return jsonify({"result": result})

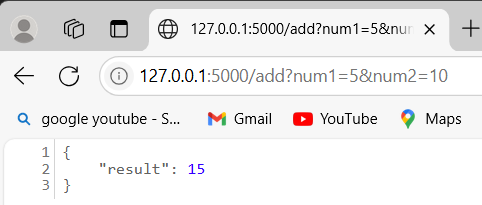
if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**OUTPUT:**



Type this URL in brower: [127.0.0.1:5000/add?num1=5&num2=10](http://127.0.0.1:5000/add?num1=5&num2=10)



**B) create a simple rest service to demonstrate CRUD operations**  
  
**Step 1 :** Install THE FOLLOWING module via pip in THE command prompt :

Command: pip install Flask Flask-SQLAlchemy mysqlclient

**Step 2 :** Open mysql workbench And create a New database ‘player\_db’ And a table ‘PLAYERS’ as follows:  
Command:  
CREATE DATABASE player\_db;

USE player\_db;

CREATE TABLE players (

id INT AUTO\_INCREMENT PRIMARY KEY,

player\_name VARCHAR(100) NOT NULL,

age INT NOT NULL,

country VARCHAR(100) NOT NULL,

runs INT NOT NULL

);

**Step 3 :** Write the following codes **app.py**

from flask import Flask, request, jsonify

from flask\_sqlalchemy import SQLAlchemy

app = Flask(\_\_name\_\_)

# Configure MySQL database connection (no password required if MySQL is set up like that)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql://root:root123@localhost/player\_db'

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db = SQLAlchemy(app)

# Define the Player model

class Player(db.Model):

    \_\_tablename\_\_ = 'players'

    id = db.Column(db.Integer, primary\_key=True)

    player\_name = db.Column(db.String(100), nullable=False)

    age = db.Column(db.Integer, nullable=False)

    country = db.Column(db.String(100), nullable=False)

    runs = db.Column(db.Integer, nullable=False)

# Create tables when app starts

with app.app\_context():

    db.create\_all()

# Create a player

@app.route('/player', methods=['POST'])

def create\_player():

    data = request.json

    player = Player(\*\*data)

    db.session.add(player)

    db.session.commit()

    return jsonify({"message": "Player created successfully"}), 201

# Get all players

@app.route('/players', methods=['GET'])

def get\_players():

    players = Player.query.all()

    return jsonify([{

        "id": p.id, "player\_name": p.player\_name, "age": p.age,

        "country": p.country, "runs": p.runs

    } for p in players])

# Update a player

@app.route('/player/<int:id>', methods=['PUT'])

def update\_player(id):

    player = Player.query.get(id)

    if not player:

        return jsonify({"message": "Player not found"}), 404

    data = request.json

    for key, value in data.items():

        setattr(player, key, value)

    db.session.commit()

    return jsonify({"message": "Player updated successfully"})

# Delete a player

@app.route('/player/<int:id>', methods=['DELETE'])

def delete\_player(id):

    player = Player.query.get(id)

    if not player:

        return jsonify({"message": "Player not found"}), 404

    db.session.delete(player)

    db.session.commit()

    return jsonify({"message": "Player deleted successfully"})

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**test\_app.py**  
from app import db, Player, app

def create\_player():

    new\_player = Player(player\_name="MS Dhoni", age=35, country="India", runs=120)

    db.session.add(new\_player)

    db.session.commit()

    print(f"Player created: {new\_player.player\_name}")

def get\_all\_players():

    players = Player.query.all()

    print("All players in DB:")

    for player in players:

        print(f"{player.id}: {player.player\_name}, {player.age}, {player.country}, {player.runs}")

def update\_player(player\_id):

    player = db.session.get(Player, player\_id)

    if player:

        player.player\_name = "Rohit Sharma"

        db.session.commit()

        print(f"Updated player {player.id} ")

    else:

        print("Player not found!")

def delete\_player(player\_id):

    player = db.session.get(Player, player\_id)

    if player:

        db.session.delete(player)

        db.session.commit()

        print(f"Deleted player {player.player\_name}")

    else:

        print("Player not found!")

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

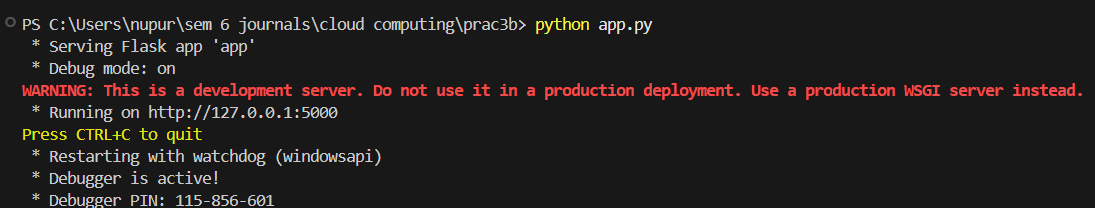
    with app.app\_context():

        create\_player()

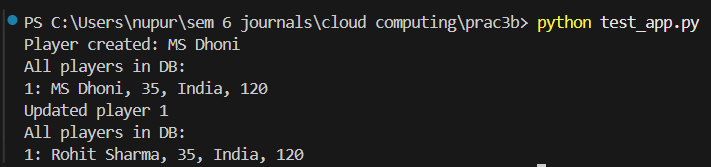
        get\_all\_players()

        update\_player(1)

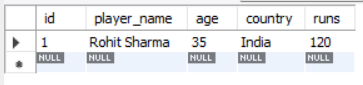
        get\_all\_players()

**Step 4 :** First run the app.py file  


**Step 5 :** now run the test\_app.py file



YOU will see that a new Player will Be created And you can Also update , delete or get all the players details  
  
In the Mysql workbench run the select command to check if the changes are getting reflected in the database  
command: select \* from players;



you can also see the details in the browser  
