CRUD Operations with MySQL and FastAPI

Group Members

Rahul Govind Kumar-25PGAI0003

Ria Singh-25PGPAI0006

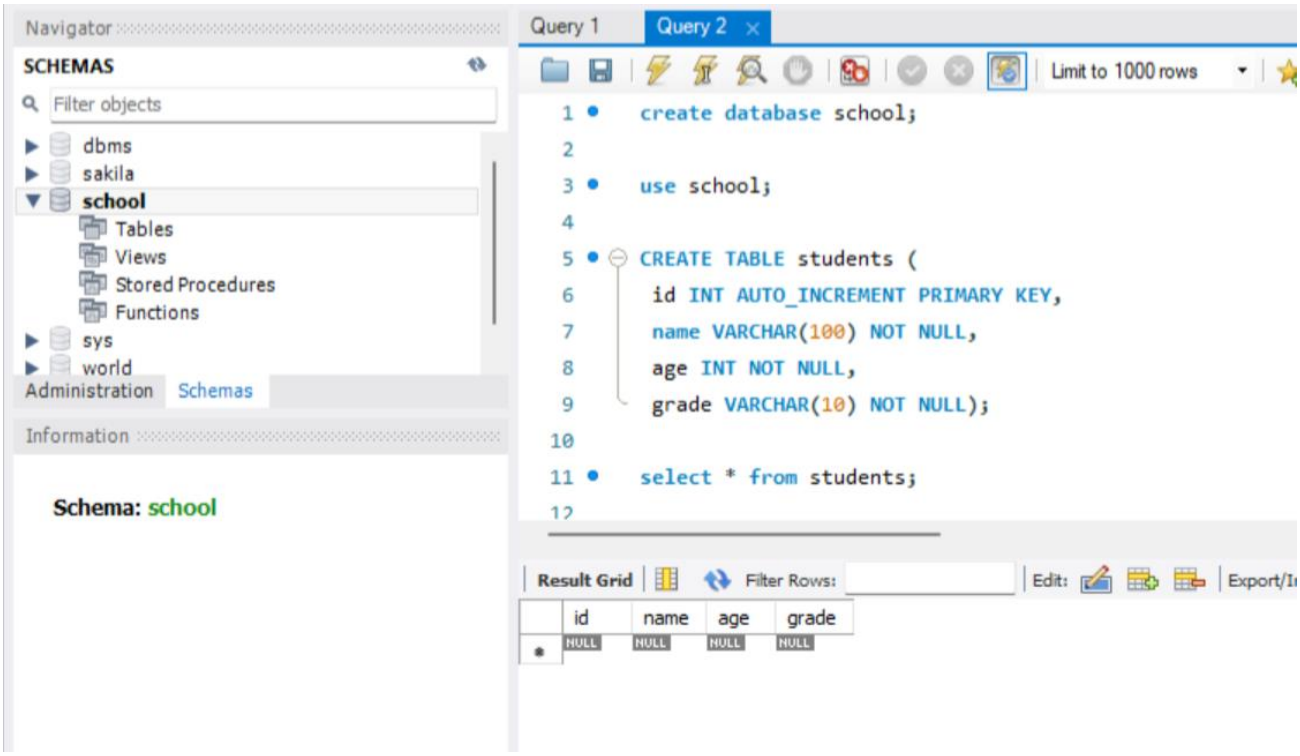
Anurag Sahu-25PGAI0143

Nishanth Pandey –25PGPAI0042

Mihir Kumar- 25PGPAI0046

GitHub Repository Link:
https://github.com/rahulorihiki/CRUD_Operations_With_MYSQL_and_FastAPI

1. **Set Up MySQL Database**
    a. Install and configure MySQL Server.
    b. Create a database ‘school’
    c. Create table ‘students’ with id, name, age, grade.



2. **Implement CRUD Wrapper Functions**
    a. Install mysql-connector-python: pip install mysql-connector-python
    b. Create a Python module database.py to implement the database connection and CRUD operations.
    c. Implement the CRUD Functions.

database.py × main.py

database.py > create_connection

```python
import mysql.connector
from mysql.connector import Error

def create_connection():
    connection = None
    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="root",
            database="school"
        )
        print("Connection to MySQL DB successful")
    except Error as e:
        print(f"The error '{e}' occurred")
    return connection

def execute_query(connection, query, values=None):
    cursor = connection.cursor()
    try:
        if values:
            cursor.execute(query, values)
        else:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Query executed successfully
INFO:     127.0.0.1:54141 - "POST /students/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:54142 - "GET /students/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:54346 - "GET /students/ HTTP/1.1" 200 OK
```

database.py × main.py

database.py > create_connection

```python

def execute_query(connection, query, values=None):
    cursor = connection.cursor()
    try:
        if values:
            cursor.execute(query, values)
        else:
            cursor.execute(query)
        connection.commit()
        print("Query executed successfully")
    except Error as e:
        print(f"The error '{e}' occurred")

def fetch_query(connection, query, values=None):
    cursor = connection.cursor(dictionary=True)
    result = None
    try:
        if values:
            cursor.execute(query, values)
        else:
            cursor.execute(query)
        result = cursor.fetchall()
        return result
    except Error as e:
        print(f"The error '{e}' occurred")
    return None

def create_student(connection, name, age, grade):
    query = "INSERT INTO students (name, age, grade) VALUES (%s, %s, %s)"
```

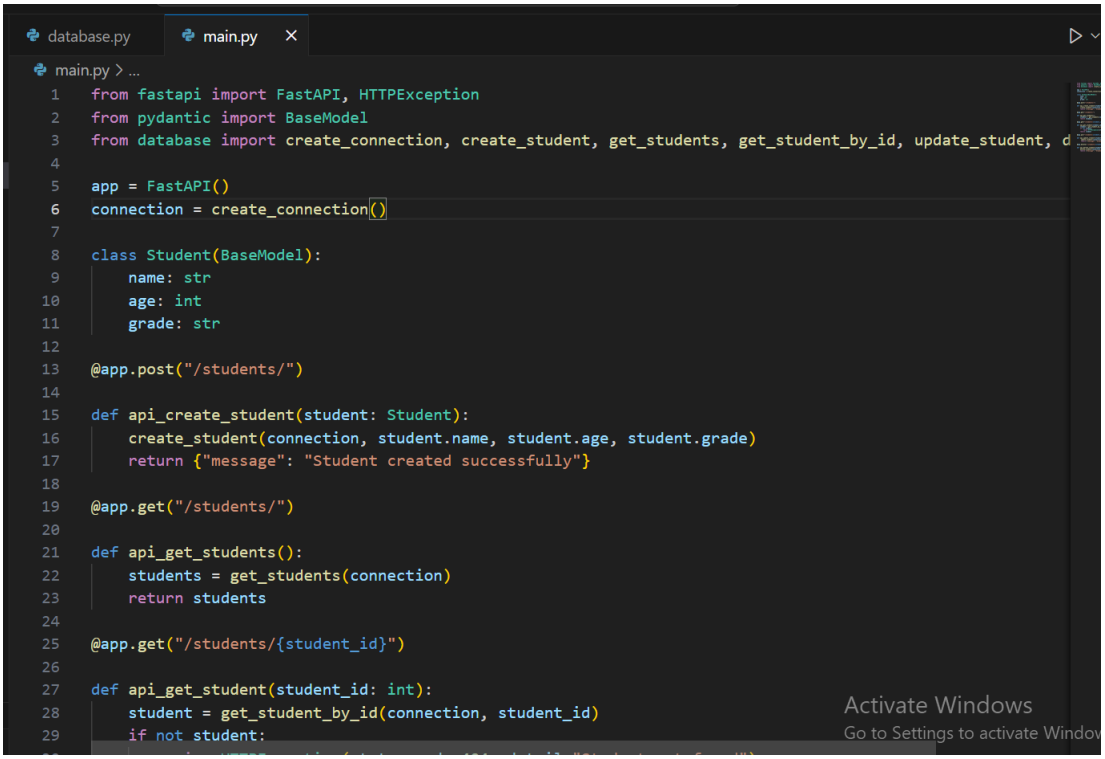
Ln 6, Col 9 Spaces: 4

database.py × main.py

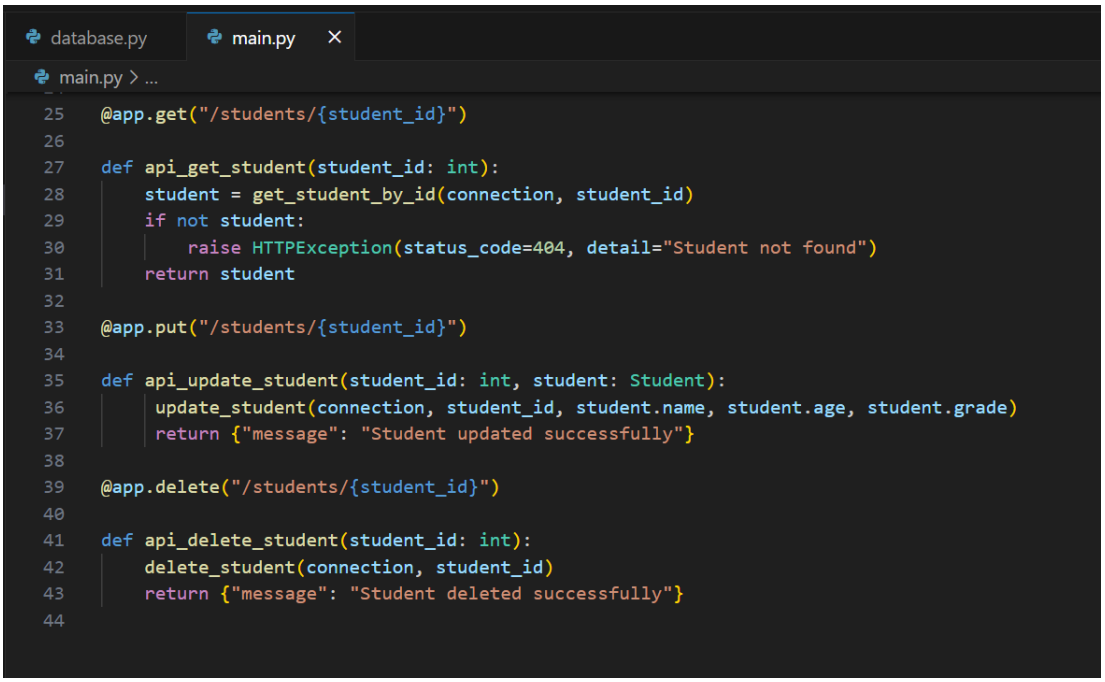
database.py > create_connection

```python

def create_student(connection, name, age, grade):
    query = "INSERT INTO students (name, age, grade) VALUES (%s, %s, %s)"
    values = (name, age, grade)
    execute_query(connection, query, values)

def get_students(connection):
    query = "SELECT * FROM students"
    return fetch_query(connection, query)

def get_student_by_id(connection, student_id):
    query = "SELECT * FROM students WHERE id = %s"
    values = (student_id,)
    return fetch_query(connection, query, values)

def update_student(connection, student_id, name, age, grade):
    query = "UPDATE students SET name = %s, age = %s, grade = %s WHERE id = %s"
    values = (name, age, grade, student_id)
    execute_query(connection, query, values)

def delete_student(connection, student_id):
    query = "DELETE FROM students WHERE id = %s"
    values = (student_id,)
    execute_query(connection, query, values)

```

3. **Develop RESTful APIs with FastAPI**
    a. Install FastAPI and Uvicorn: pip install fastapi uvicorn
    b. Create FastAPI Application by Creating a file named main.py
    c. Run the FastAPI Application: uvicorn main:app –reload



```python
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from database import create_connection, create_student, get_students, get_student_by_id, update_student, d

app = FastAPI()
connection = create_connection()

class Student(BaseModel):
    name: str
    age: int
    grade: str

@app.post("/students/")

def api_create_student(student: Student):
    create_student(connection, student.name, student.age, student.grade)
    return {"message": "Student created successfully"}

@app.get("/students/")

def api_get_students():
    students = get_students(connection)
    return students

@app.get("/students/{student_id}")

def api_get_student(student_id: int):
    student = get_student_by_id(connection, student_id)
    if not student:
```

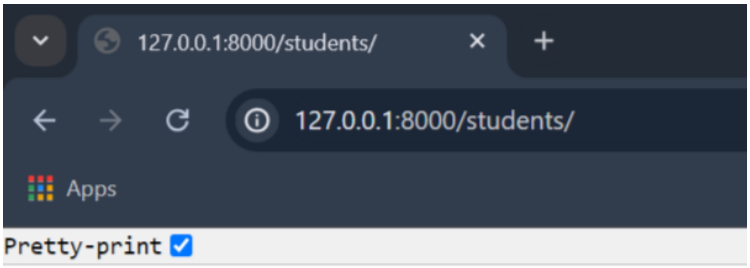



```python
@app.get("/students/{student_id}")

def api_get_student(student_id: int):
    student = get_student_by_id(connection, student_id)
    if not student:
        raise HTTPException(status_code=404, detail="Student not found")
    return student

@app.put("/students/{student_id}")

def api_update_student(student_id: int, student: Student):
    update_student(connection, student_id, student.name, student.age, student.grade)
    return {"message": "Student updated successfully"}

@app.delete("/students/{student_id}")

def api_delete_student(student_id: int):
    delete_student(connection, student_id)
    return {"message": "Student deleted successfully"}
```

```
PS C:\Users\PC\Desktop\DBMS Assignment 5> python -m uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\PC\\Desktop\\DBMS Assignment 5']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [14244] using WatchFiles
Connection to MySQL DB successful
INFO:     Started server process [12096]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

4. **Test the Application using cURL tool**
    a. Create a Student
    b. Get All Students
    c. Update a Student
    d. Delete a Student

```
C:\Users\PC>curl -X POST "http://127.0.0.1:8000/students/" -H "accept:application/json" -H "Content-Type: application/json" -d "{\"na
me\":\"John Doe\",\"age\":21,\"grade\":\"A\"}"
{"message":"Student created successfully"}
C:\Users\PC>curl -X GET "http://127.0.0.1:8000/students/" -H "accept:application/json"
[{"id":3,"name":"John Doe","age":21,"grade":"A"}]
C:\Users\PC>curl -X PUT "http://127.0.0.1:8000/students/3" -H "accept:application/json" -H "Content-Type: application/json" -d "{\"na
me\":\"Jane Doe\",\"age\":22,\"grade\":\"A+\"}"
{"message":"Student updated successfully"}
C:\Users\PC>curl -X DELETE "http://127.0.0.1:8000/students/3" -H "accept:application/json"
{"message":"Student deleted successfully"}
C:\Users\PC>
```




```
[
  {
    "id": 4,
    "name": "John Doe",
    "age": 21,
    "grade": "A"
  }
]
```