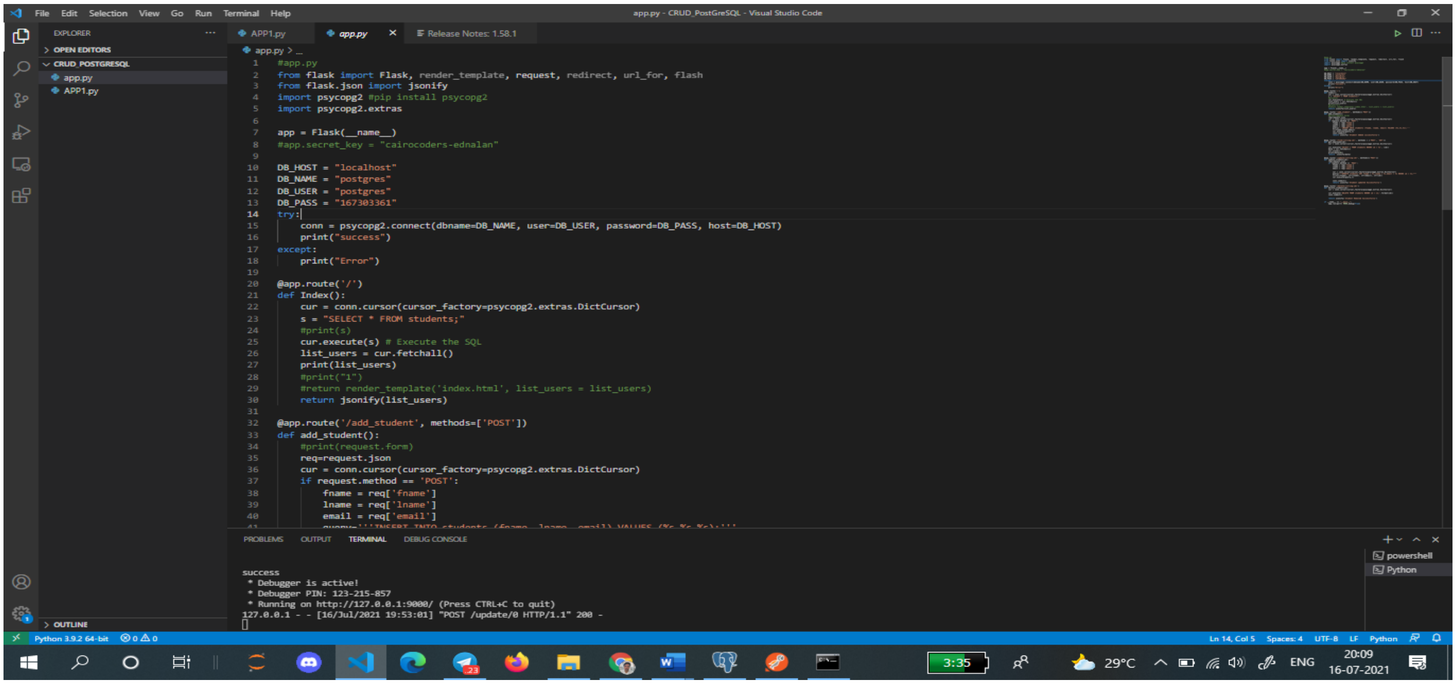
Operations on PostgreSql using Flask and postman

Here the main theme is basic operations on PostgreSql using flask and this type can be used in web applications Postman will take a part which html and css will do.

Github Link: https://github.com/krishnasai7500/CRUD-Operations-in-POSTGRE-SQL-using-Flask-and-postmen

Requirements :

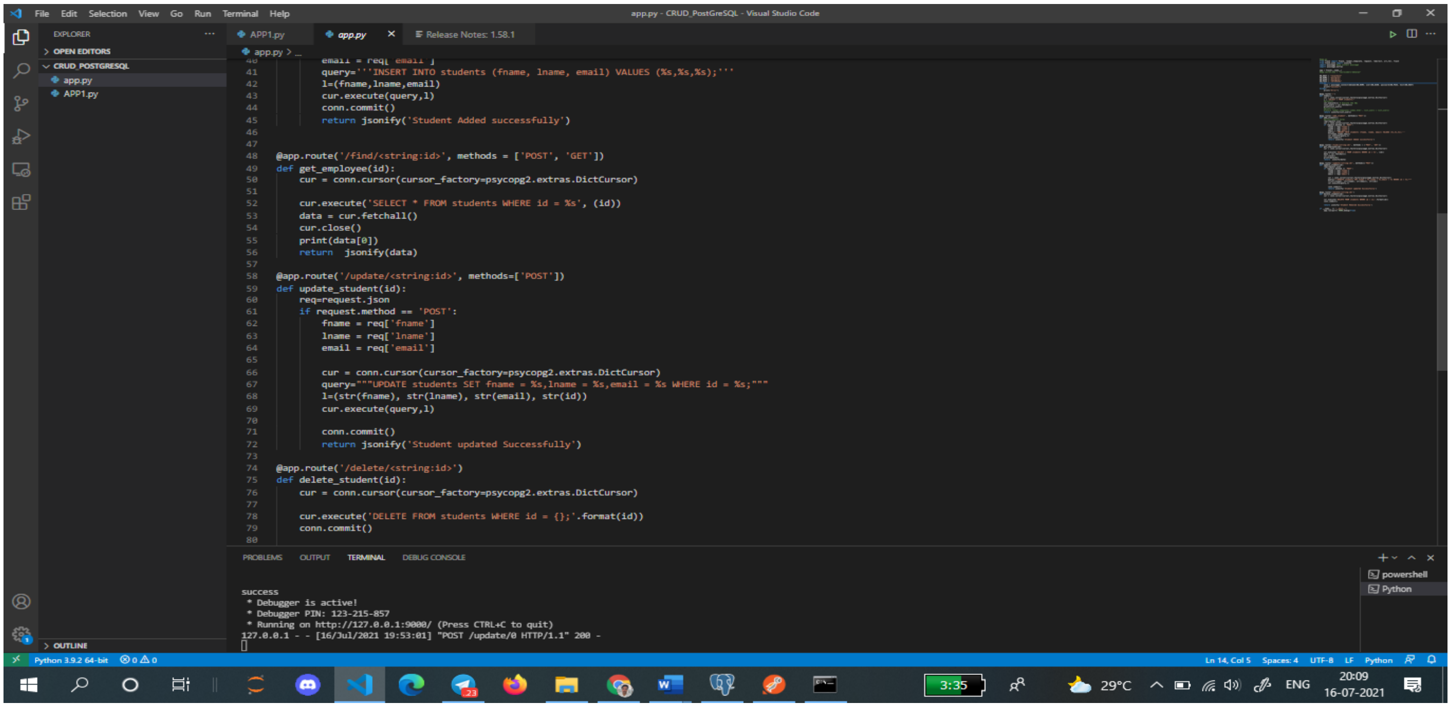
1. **Flask**
2. **PostgreSql database**
3. **Postman**



File Edit Selection View Go Run Terminal Help
app.py - CRUD_PostGreSQL - Visual Studio Code
EXPLORER
OPEN EDITORS
CRUD_POSTGRESQL
app.py
APP1.py
OUTLINE
APP1.py
app.py
Release Notes: 1.58.1
app.py > ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
Python
Python 3.9.2 64-bit ⊗ 0 ⚠ 0
Ln 14, Col 5 Spaces: 4 UTF-8 LF Python
3:35
29°C
ENG
20:09
16-07-2021

```python
#app.py
from flask import Flask, render_template, request, redirect, url_for, flash
from flask.json import jsonify
import psycopg2 #pip install psycopg2
import psycopg2.extras

app = Flask(__name__)
#app.secret_key = "cairocoders-ednalan"

DB_HOST = "localhost"
DB_NAME = "postgres"
DB_USER = "postgres"
DB_PASS = "167303361"
try:
    conn = psycopg2.connect(dbname=DB_NAME, user=DB_USER, password=DB_PASS, host=DB_HOST)
    print("success")
except:
    print("Error")

@app.route('/')
def Index():
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    s = "SELECT * FROM students;"
    #print(s)
    cur.execute(s) # Execute the SQL
    list_users = cur.fetchall()
    print(list_users)
    #print("1")
    #return render_template('index.html', list_users = list_users)
    return jsonify(list_users)

@app.route('/add_student', methods=['POST'])
def add_student():
    #print(request.form)
    req=request.json
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
    if request.method == 'POST':
        fname = req['fname']
        lname = req['lname']
        email = req['email']
        query='''INSERT INTO students (fname, lname, email) VALUES (%s,%s,%s);'''
```

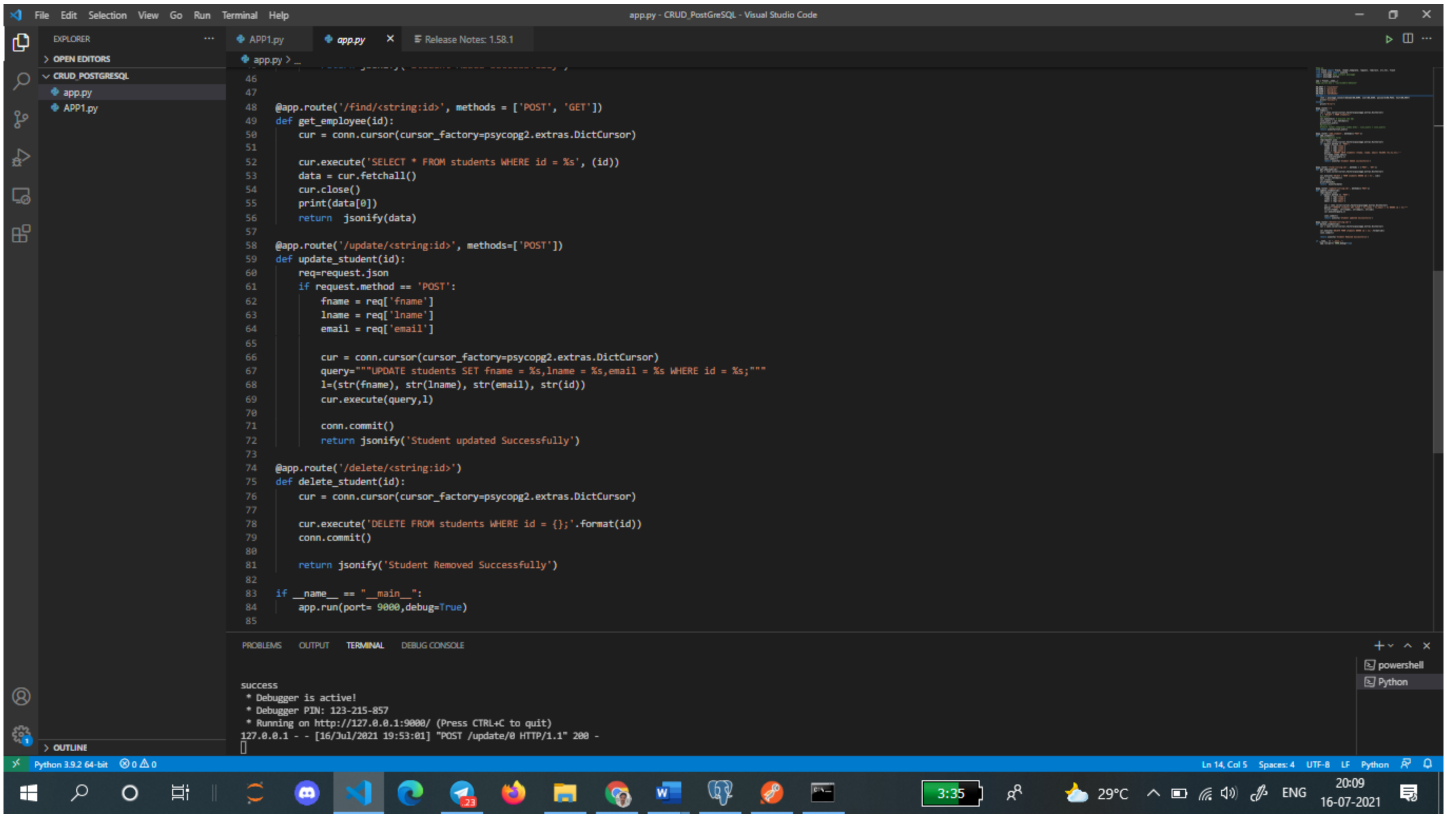
```
success
 * Debugger is active!
 * Debugger PIN: 123-215-857
 * Running on http://127.0.0.1:9000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Jul/2021 19:53:01] "POST /update/0 HTTP/1.1" 200 -
```



File Edit Selection View Go Run Terminal Help
app.py - CRUD_PostGreSQL - Visual Studio Code
EXPLORER
OPEN EDITORS
CRUD_POSTGRESQL
app.py
APP1.py
OUTLINE
APP1.py
app.py
Release Notes: 1.58.1
app.py > ...
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
powershell
Python
Python 3.9.2 64-bit ⊗ 0 ⚠ 0
Ln 14, Col 5 Spaces: 4 UTF-8 LF Python
3:35
29°C
ENG
20:09
16-07-2021

```python
        email = req['email']
        query='''INSERT INTO students (fname, lname, email) VALUES (%s,%s,%s);'''
        l=(fname,lname,email)
        cur.execute(query,l)
        conn.commit()
        return jsonify('Student Added successfully')


@app.route('/find/<string:id>', methods = ['POST', 'GET'])
def get_employee(id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('SELECT * FROM students WHERE id = %s', (id))
    data = cur.fetchall()
    cur.close()
    print(data[0])
    return  jsonify(data)

@app.route('/update/<string:id>', methods=['POST'])
def update_student(id):
    req=request.json
    if request.method == 'POST':
        fname = req['fname']
        lname = req['lname']
        email = req['email']

        cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
        query="""UPDATE students SET fname = %s,lname = %s,email = %s WHERE id = %s;"""
        l=(str(fname), str(lname), str(email), str(id))
        cur.execute(query,l)

        conn.commit()
        return jsonify('Student updated Successfully')

@app.route('/delete/<string:id>')
def delete_student(id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('DELETE FROM students WHERE id = {};'.format(id))
    conn.commit()

```

```
success
 * Debugger is active!
 * Debugger PIN: 123-215-857
 * Running on http://127.0.0.1:9000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Jul/2021 19:53:01] "POST /update/0 HTTP/1.1" 200 -
```

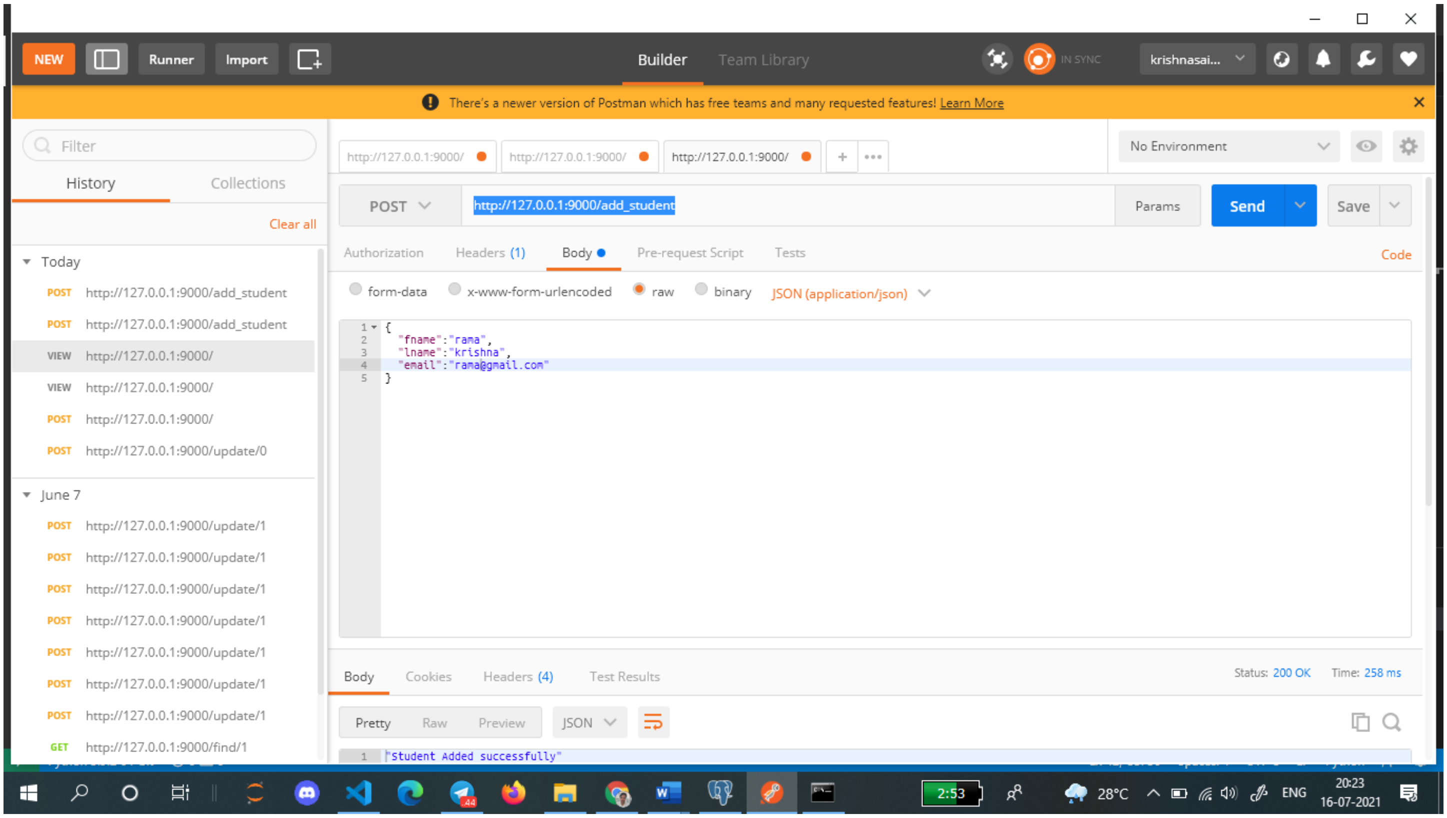


```python
@app.route('/find/<string:id>', methods = ['POST', 'GET'])
def get_employee(id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('SELECT * FROM students WHERE id = %s', (id))
    data = cur.fetchall()
    cur.close()
    print(data[0])
    return  jsonify(data)

@app.route('/update/<string:id>', methods=['POST'])
def update_student(id):
    req=request.json
    if request.method == 'POST':
        fname = req['fname']
        lname = req['lname']
        email = req['email']

        cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
        query="""UPDATE students SET fname = %s,lname = %s,email = %s WHERE id = %s;"""
        l=(str(fname), str(lname), str(email), str(id))
        cur.execute(query,l)

        conn.commit()
        return jsonify('Student updated Successfully')

@app.route('/delete/<string:id>')
def delete_student(id):
    cur = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)

    cur.execute('DELETE FROM students WHERE id = {};'.format(id))
    conn.commit()

    return jsonify('Student Removed Successfully')

if __name__ == "__main__":
    app.run(port= 9000,debug=True)
```

```
success
 * Debugger is active!
 * Debugger PIN: 123-215-857
 * Running on http://127.0.0.1:9000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Jul/2021 19:53:01] "POST /update/0 HTTP/1.1" 200 -
```

Here 5 roots are there in flask application

1.index 2.add_Student 3.get_Student 4.Update_student 5.delete_student



Today

```
POST  http://127.0.0.1:9000/add_student
POST  http://127.0.0.1:9000/add_student
VIEW  http://127.0.0.1:9000/
VIEW  http://127.0.0.1:9000/
POST  http://127.0.0.1:9000/
POST  http://127.0.0.1:9000/update/0
```

June 7

```
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
POST  http://127.0.0.1:9000/update/1
GET   http://127.0.0.1:9000/find/1
```

POST http://127.0.0.1:9000/add_student

○ form-data ○ x-www-form-urlencoded ● raw ○ binary JSON (application/json)

```json
{
  "fname":"rama",
  "lname":"krishna",
  "email":"rama@gmail.com"
}
```

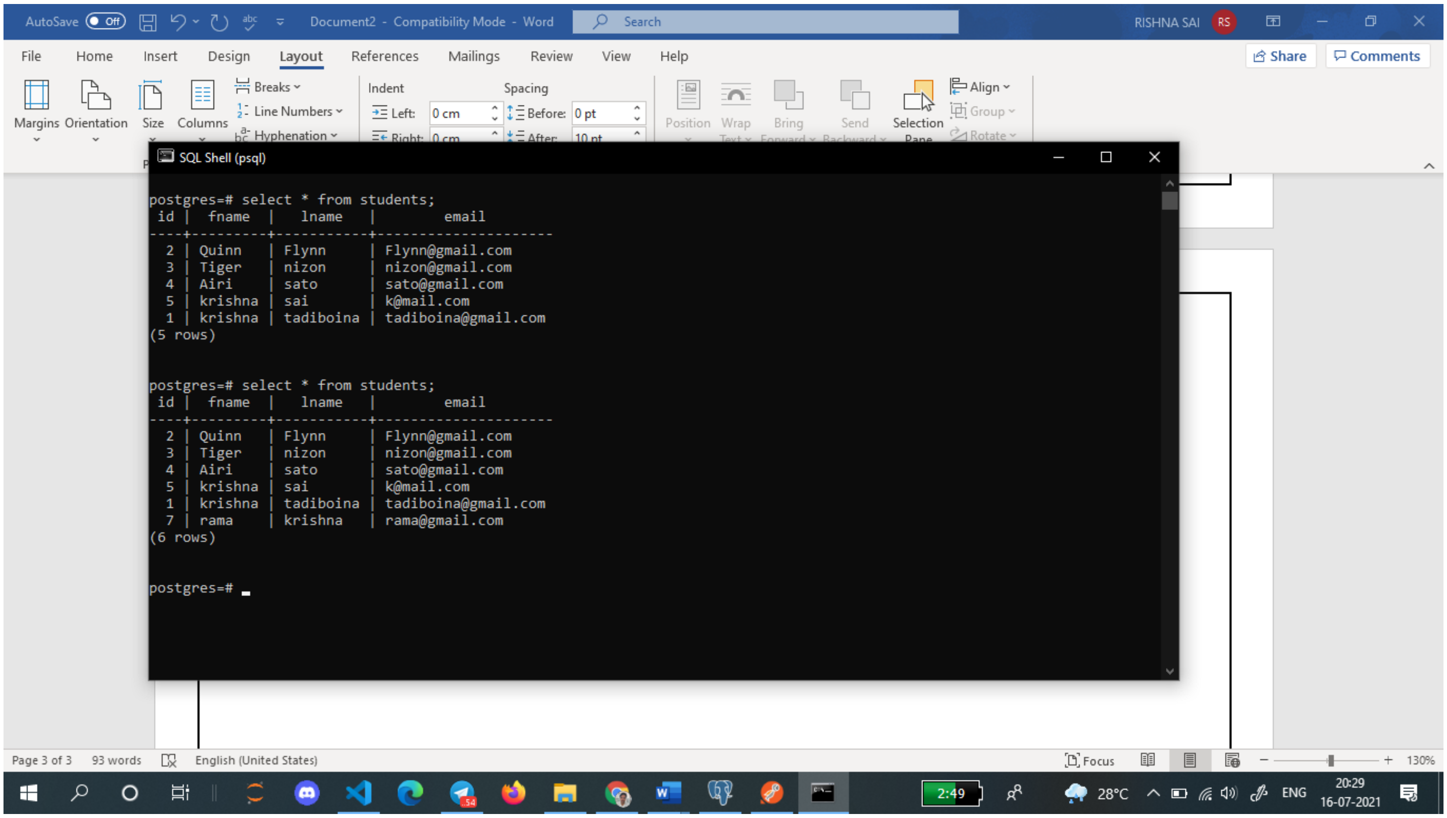
Status: 200 OK Time: 258 ms

```
"Student Added successfully"
```

Here the flask will run under the ip address http://127.0.0.1:80/

Here the data will sent in form of json format

We can do any kind of operation with this process



```
postgres=# select * from students;
 id |  fname  |   lname   |        email
----+---------+-----------+---------------------
  2 | Quinn   | Flynn     | Flynn@gmail.com
  3 | Tiger   | nizon     | nizon@gmail.com
  4 | Airi    | sato      | sato@gmail.com
  5 | krishna | sai       | k@mail.com
  1 | krishna | tadiboina | tadiboina@gmail.com
(5 rows)


postgres=# select * from students;
 id |  fname  |   lname   |        email
----+---------+-----------+---------------------
  2 | Quinn   | Flynn     | Flynn@gmail.com
  3 | Tiger   | nizon     | nizon@gmail.com
  4 | Airi    | sato      | sato@gmail.com
  5 | krishna | sai       | k@mail.com
  1 | krishna | tadiboina | tadiboina@gmail.com
  7 | rama    | krishna   | rama@gmail.com
(6 rows)


postgres=#
```

This is the output in postgre sql prompt

Thanking you :)

By T Krishna sai