AIDI2004 – AI in Enterprise systems

Assignment #3

DURHAM COLLEGE

SUCCESS MATTERS

Group 9

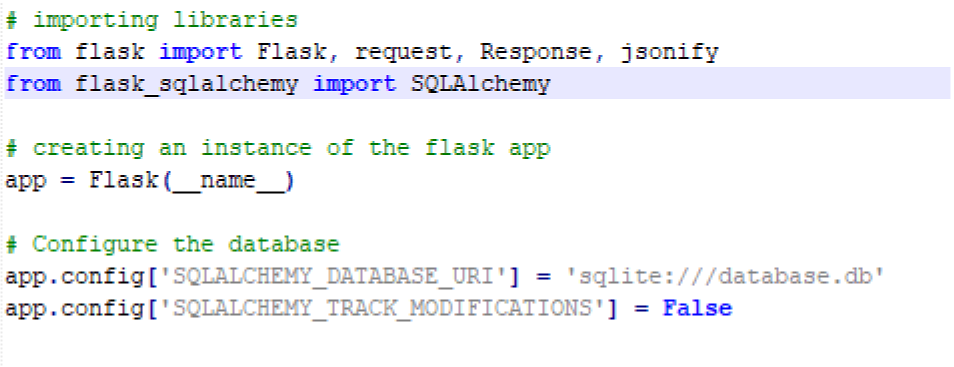
NAME: Dheeraj Reddy

STUDENT NUMBER: 100811270

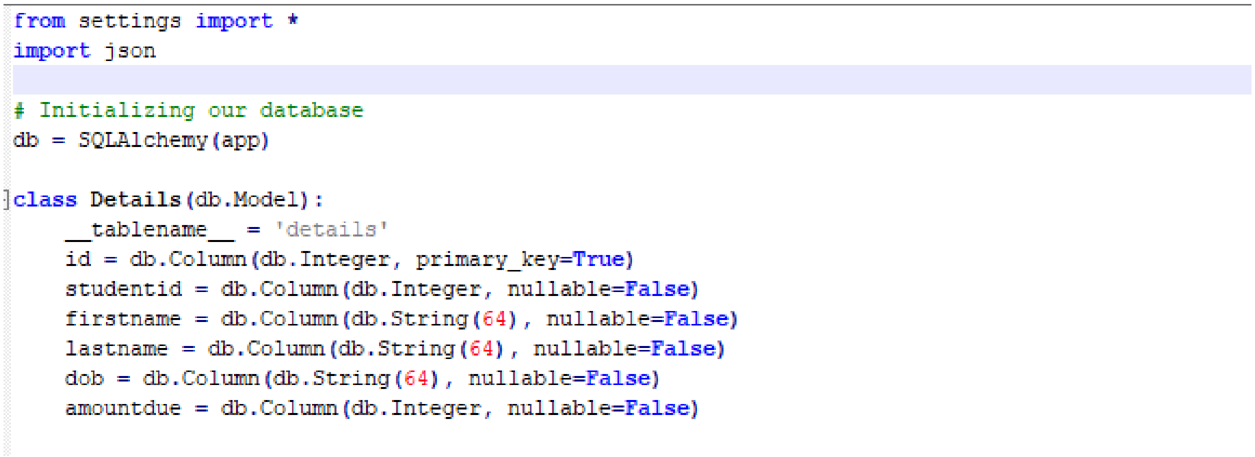
NAME: Ayush Parolia

STUDENT NUMBER: 100799278

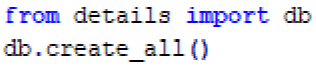
Step 1: Creating Database



```
# importing libraries
from flask import Flask, request, Response, jsonify
from flask_sqlalchemy import SQLAlchemy

# creating an instance of the flask app
app = Flask(__name__)

# Configure the database
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

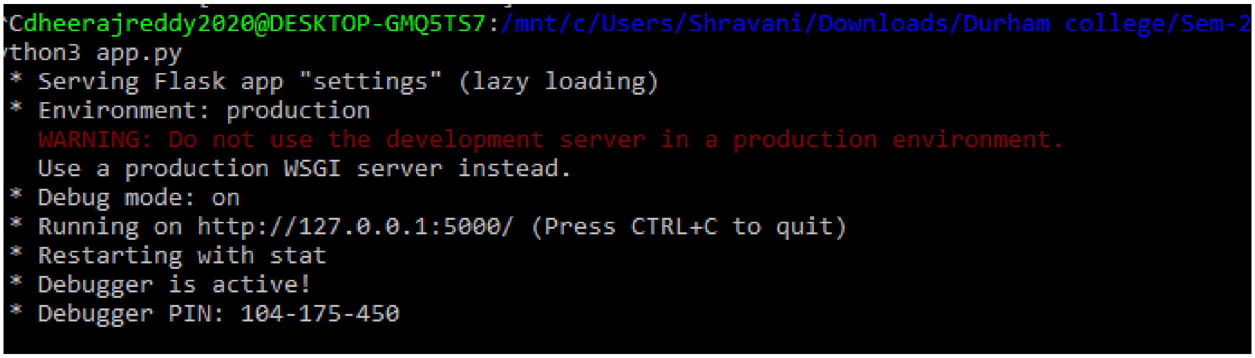


```
from settings import *
import json

# Initializing our database
db = SQLAlchemy(app)

class Details(db.Model):
    __tablename__ = 'details'
    id = db.Column(db.Integer, primary_key=True)
    studentid = db.Column(db.Integer, nullable=False)
    firstname = db.Column(db.String(64), nullable=False)
    lastname = db.Column(db.String(64), nullable=False)
    dob = db.Column(db.String(64), nullable=False)
    amountdue = db.Column(db.Integer, nullable=False)
```



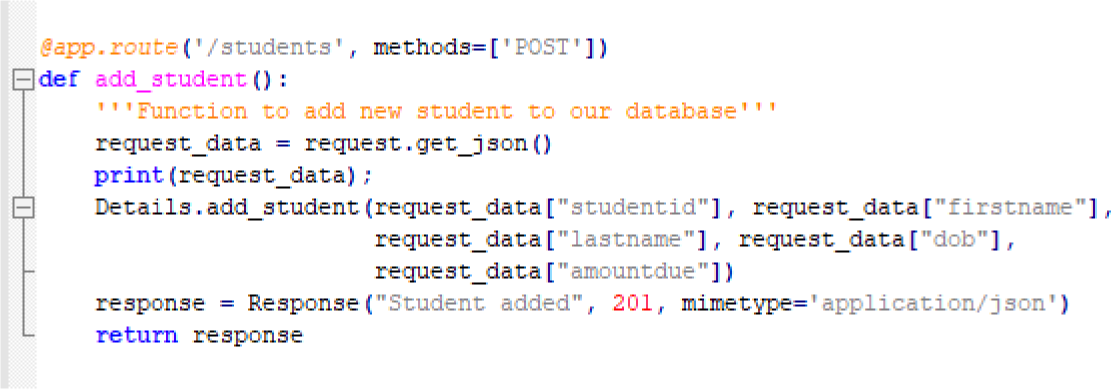
```
from details import db
db.create_all()
```

Step 2: Creating Basic Flask app



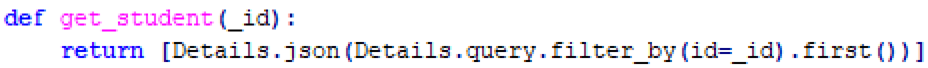
```
Cdheerajreddy2020@DESKTOP-GMQ5TS7:/mnt/c/Users/Shravani/Downloads/Durham college/Sem-2
thon3 app.py
 * Serving Flask app "settings" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 104-175-450
```

Step 3: For Create



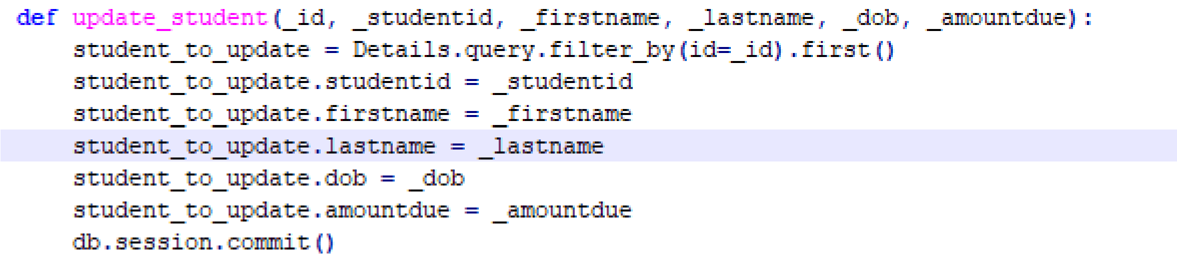
```
@app.route('/students', methods=['POST'])
def add_student():
    '''Function to add new student to our database'''
    request_data = request.get_json()
    print(request_data);
    Details.add_student(request_data["studentid"], request_data["firstname"],
                        request_data["lastname"], request_data["dob"],
                        request_data["amountdue"])
    response = Response("Student added", 201, mimetype='application/json')
    return response
```

Step 4: Read



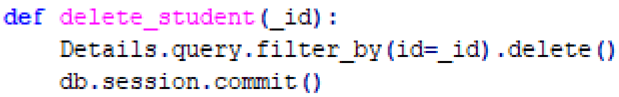
```
def get_student(_id):
    return [Details.json(Details.query.filter_by(id=_id).first())]
```

Step 5: Update



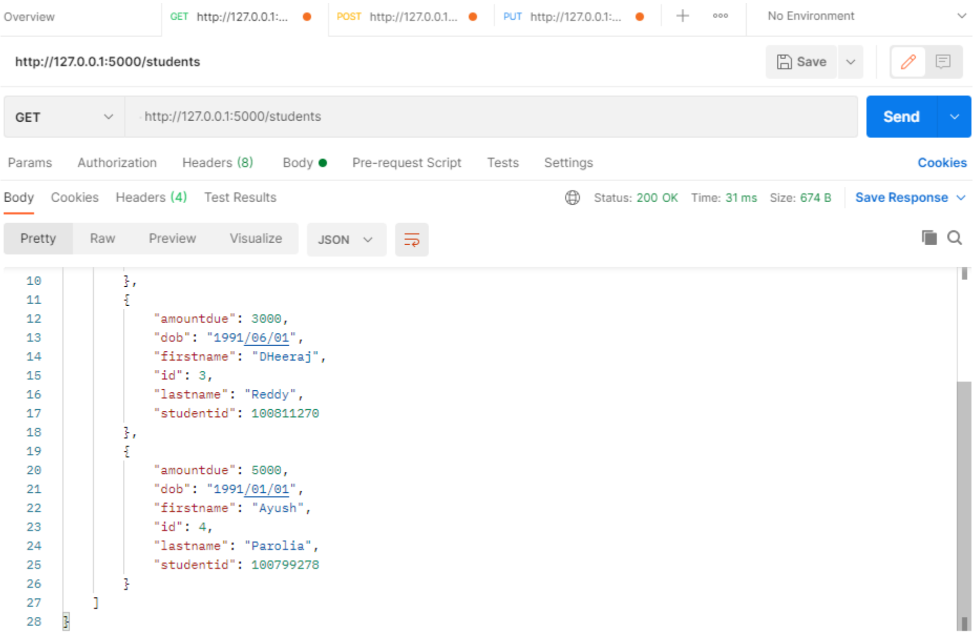
```
def update_student(_id, _studentid, _firstname, _lastname, _dob, _amountdue):
    student_to_update = Details.query.filter_by(id=_id).first()
    student_to_update.studentid = _studentid
    student_to_update.firstname = _firstname
    student_to_update.lastname = _lastname
    student_to_update.dob = _dob
    student_to_update.amountdue = _amountdue
    db.session.commit()
```

Step 6: For Delete



```
def delete_student(_id):
    Details.query.filter_by(id=_id).delete()
    db.session.commit()
```

Step 7 : Show all records



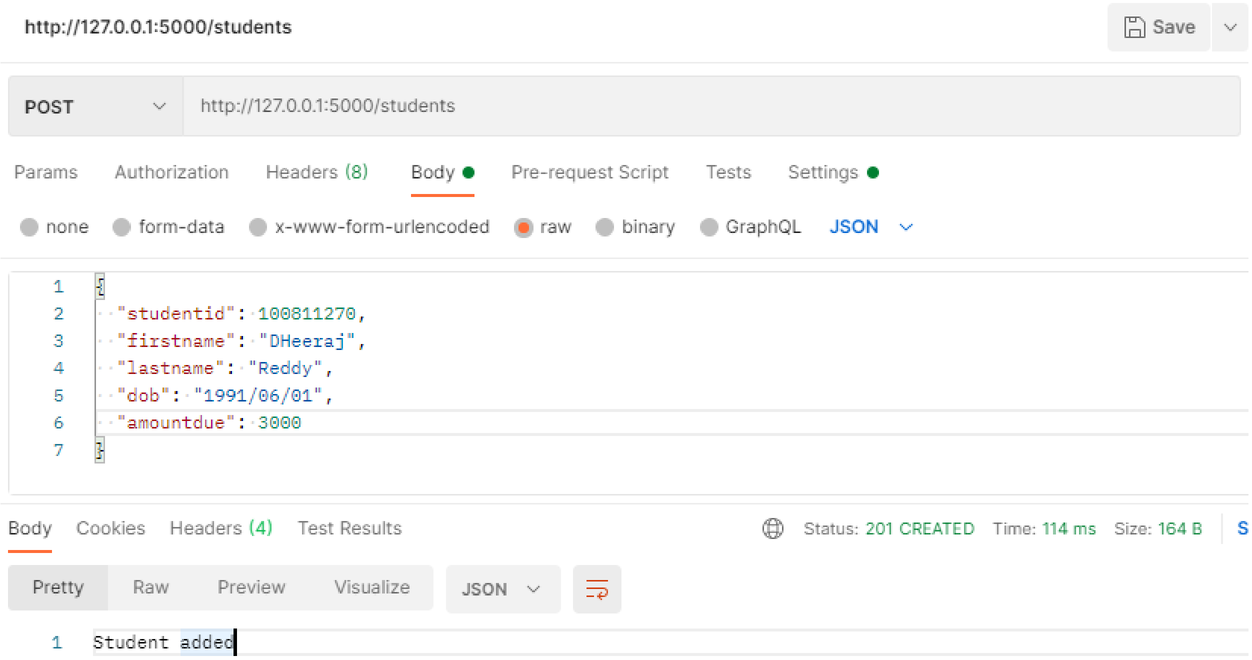
http://127.0.0.1:5000/students

GET http://127.0.0.1:5000/students

Status: 200 OK Time: 31 ms Size: 674 B

```
        },
        {
            "amountdue": 3000,
            "dob": "1991/06/01",
            "firstname": "DHeeraj",
            "id": 3,
            "lastname": "Reddy",
            "studentid": 100811270
        },
        {
            "amountdue": 5000,
            "dob": "1991/01/01",
            "firstname": "Ayush",
            "id": 4,
            "lastname": "Parolia",
            "studentid": 100799278
        }
    ]
}
```

Step 8 : Postman



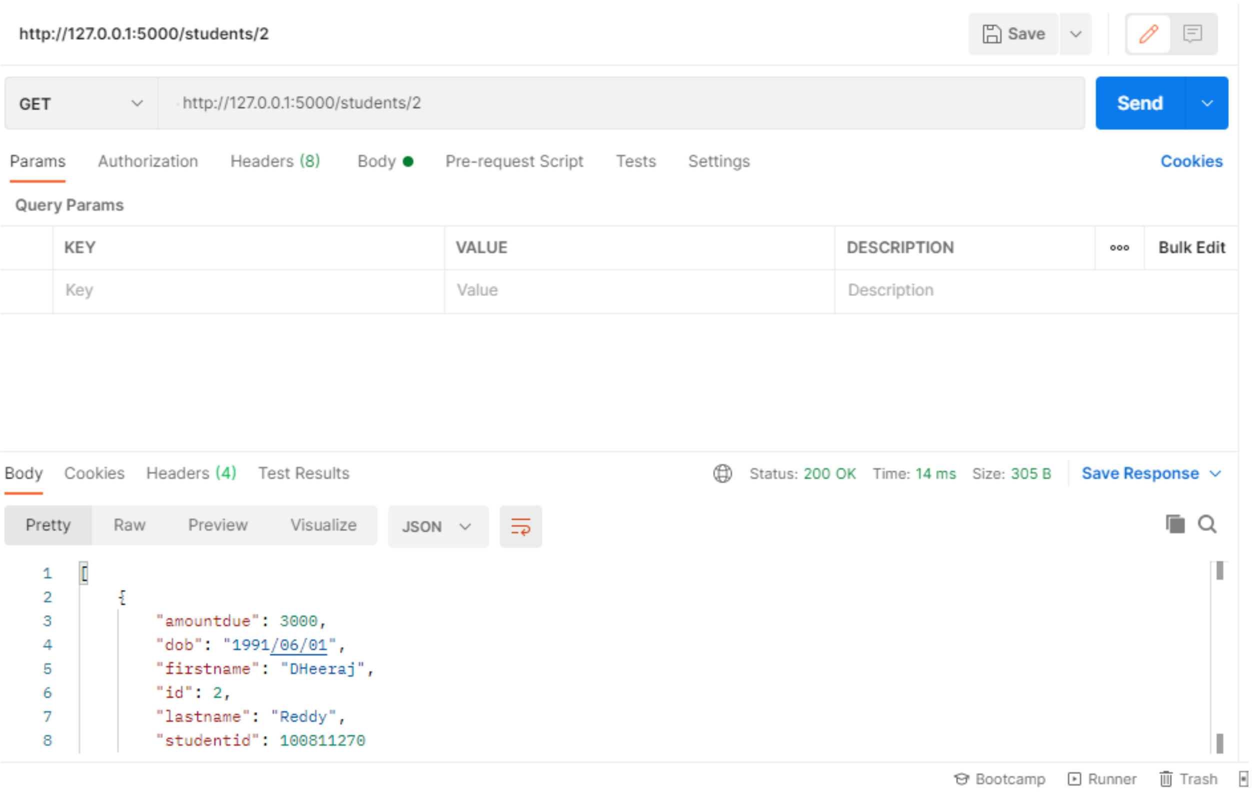
http://127.0.0.1:5000/students

POST http://127.0.0.1:5000/students

```
{
  "studentid": 100811270,
  "firstname": "DHeeraj",
  "lastname": "Reddy",
  "dob": "1991/06/01",
  "amountdue": 3000
}
```

Status: 201 CREATED Time: 114 ms Size: 164 B

```
Student added
```



http://127.0.0.1:5000/students/2

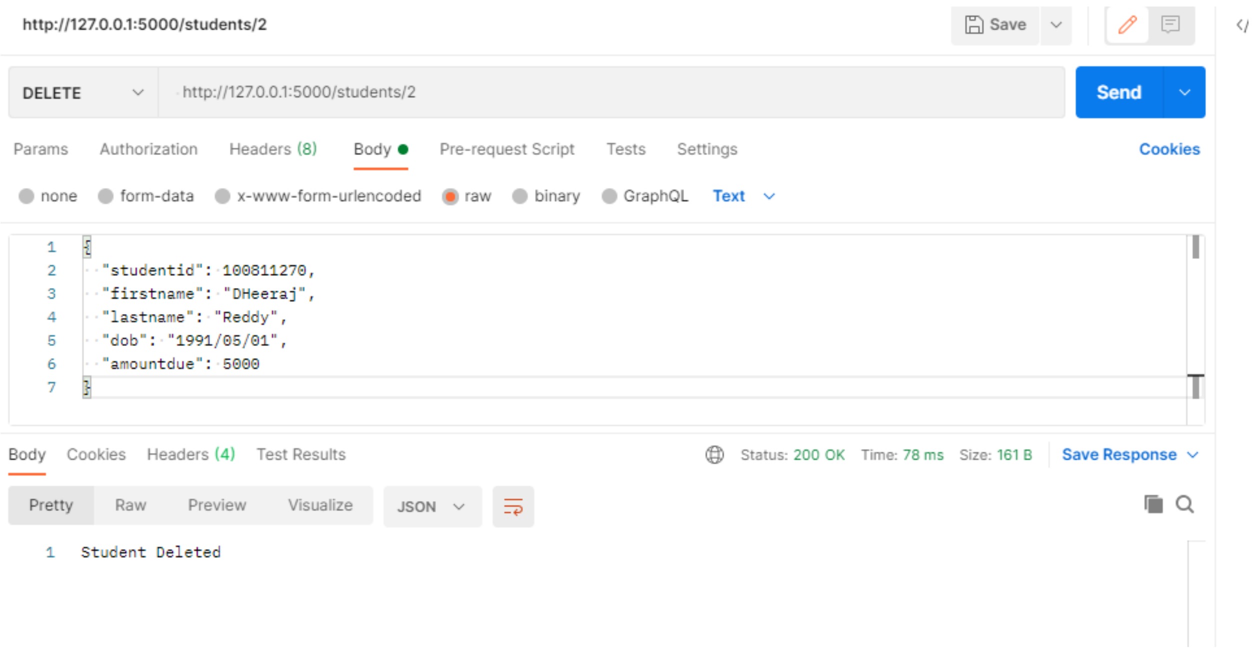
GET http://127.0.0.1:5000/students/2 Send

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 14 ms Size: 305 B

```
[
    {
        "amountdue": 3000,
        "dob": "1991/06/01",
        "firstname": "DHeeraj",
        "id": 2,
        "lastname": "Reddy",
        "studentid": 100811270
```



http://127.0.0.1:5000/students/2

DELETE http://127.0.0.1:5000/students/2 Send

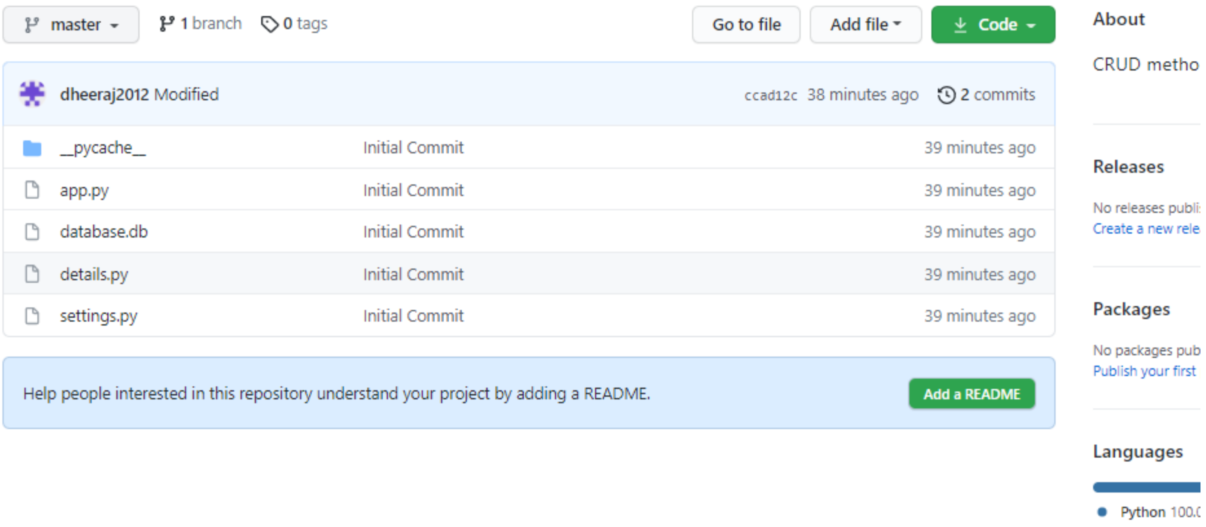
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL Text

```
{
  "studentid": 100811270,
  "firstname": "DHeeraj",
  "lastname": "Reddy",
  "dob": "1991/05/01",
  "amountdue": 5000
}
```

Status: 200 OK Time: 78 ms Size: 161 B

```
Student Deleted
```

Step 9: Github repo link : https://github.com/dheerajreddy2020/CRUD_in_Flask



dheeraj2012 Modified ccad12c 38 minutes ago 2 commits

__pycache__	Initial Commit	39 minutes ago
app.py	Initial Commit	39 minutes ago
database.db	Initial Commit	39 minutes ago
details.py	Initial Commit	39 minutes ago
settings.py	Initial Commit	39 minutes ago

Help people interested in this repository understand your project by adding a README.

Step 10 : Collaboration Plan

Task	**Responsible**	**Time**
Flask analysis	Dheeraj	2 Hrs
Flask app creation	Dheeraj, Ayush	4 Hrs
Postman testing	Ayush	3 Hrs
Document Creation	Dheeraj, Ayush	1 Hr