TASK -4

a) Write a brief README file that includes instructions on how to set up and run your Flask application.

Environment Setup Install Python 3.x:

Download and install Python 3.x from the official website:https://www.python.org/ftp/python/3.12.4/python-3.12.4-amd64.exe

Install flask using pip:

```
pip install django
```

Install MySQL and Set Up a Local Database

Download and install MySQL from the official MySQL website. Set up a local MySQL database:

```
CREATE DATABASE sample;
```

Task 1: Django API Development:

1. Creating new django Project.

   ```
   django-admin startproject myproject
   ```

2. Creating new django App.

```
cd myproject python manage.py startapp myapp
```

a. Create a route /hello:

http://127.0.0.1:8000/hello/ output: "Hello World!"

b. b. Implement a route /users to retrieve a list of users from a MySQL database:

http://127.0.0.1:8000/users/ output:

Name Email Age Uttam Yadav uttamcseau@gmail.com 23 Viraj Singh viraj@gmail.com 22 Radha Yadav radha@gmail.com 23

c. Implement a route /new_user to render an HTML page to accept input from the user and store the information in the database:

http://127.0.0.1:8000/new_user/ output:

Name: --------------

Email: ----------------

Age: -----

```
Submit
```

d. Create a route /users/ that retrieves a specific user's details from the database:

http://127.0.0.1:8000/user/<1>/ output:

User Details Name: Uttam Yadav Email: uttamcseau@gmail.com Age: 23

http://127.0.0.1:8000/user/<2>/ output:

User Details Name: Viraj Singh Email: viraj@gmail.com Age: 22

http://127.0.0.1:8000/user/<2>/ output:

User Details Name: radha Email: radha@gmail.com Age: 23

e. Add error handling:

def user_detail(request,id): try: user = User.objects.get(pk=id) return render(request, 'user_detail.html', {'user': user}) except User.DoesNotExist: return HttpResponse("User not found", status=404)

to run the file use

```
python manage.py runserver
```

b) . Include information about the database schema and how to populate it with sample data.

Set up the MySQL database:

i) create migration

```
python manage.py makemigrations
```

ii) appyl migrate

```
python manage.py migrate
```

c) Command in SQL

CREATE DATABASE users; Create table myapp_user(id (int, primary key)

- name (varchar)
- email (varchar)
- role (varchar)
-);

insert into myapp_user values(5,'Aniket','aniket@gmail.com','Data Engineer'); select * from myapp_user; select * from myapp_user where id=3;

d) Git Work flow

git init git add -A git commit -a -m "important changes" git remote add origin "https://github.com/uttamcse/steptech.git" git push -u origin master git pull origin master git checkout -b steptech_assignment git push origin steptech_assignment

Deliverables

1.)

```
git init git add -A git commit -a -m "important changes" git remote add origin
"https://github.com/uttamcse/steptech.git" git push -u origin master git pull origin
master git checkout -b steptech_assignment git push origin steptech_assignment
```

2. Insert sample data into the "users" table.

```
insert into myapp_user values(5,'Aniket','aniket@gmail.com','Data Engineer');
```