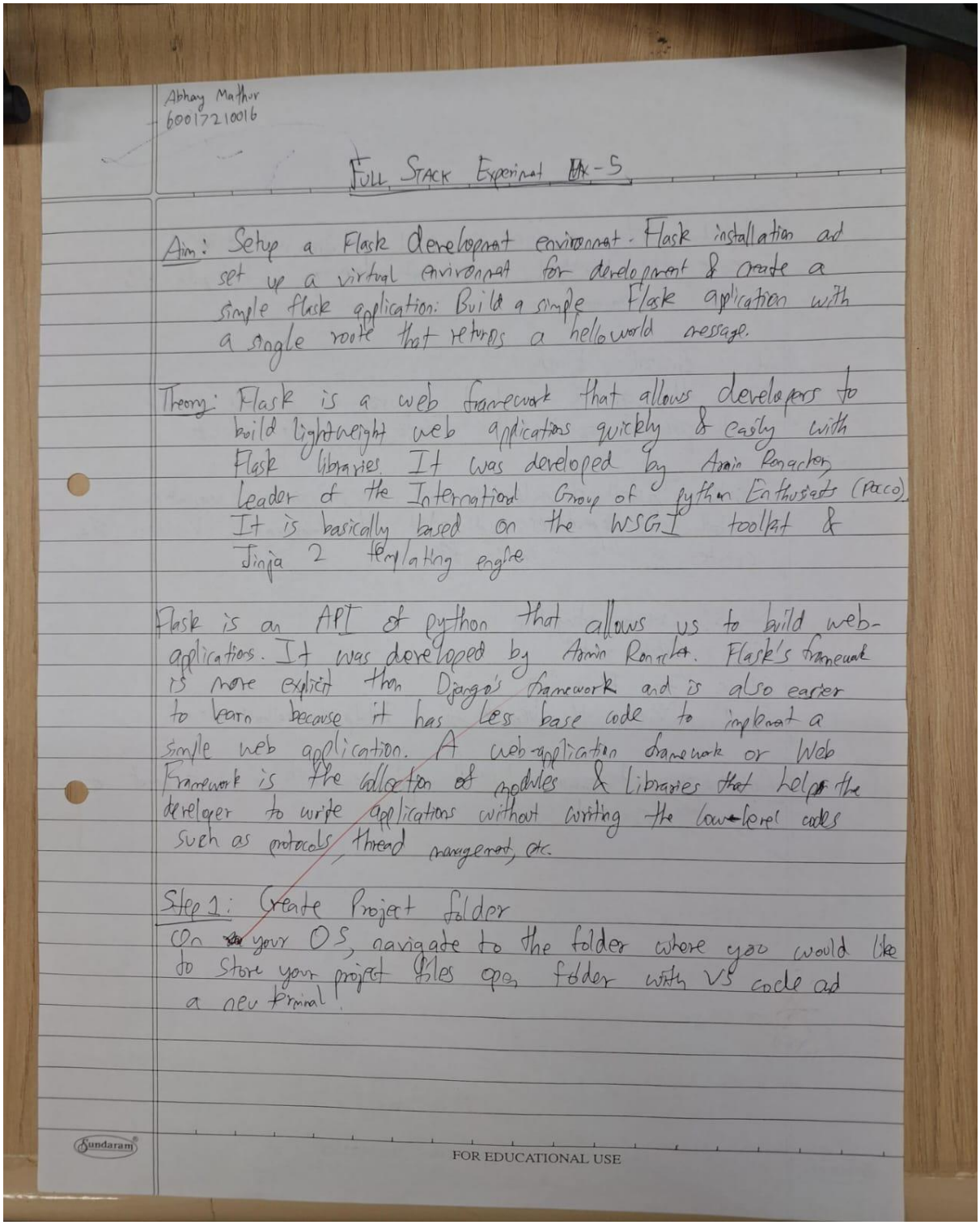
Full Stack Experiment 5

Name: Abhay Mathur
SAPID: 60017210016
Batch: AIML A1

Theory:



Abhay Mathur
60017210016

FULL STACK Experiment No-5

Aim: Setup a Flask development environment. Flask installation and set up a virtual environment for development & create a simple flask application. Build a simple Flask application with a single route that returns a hello world message.

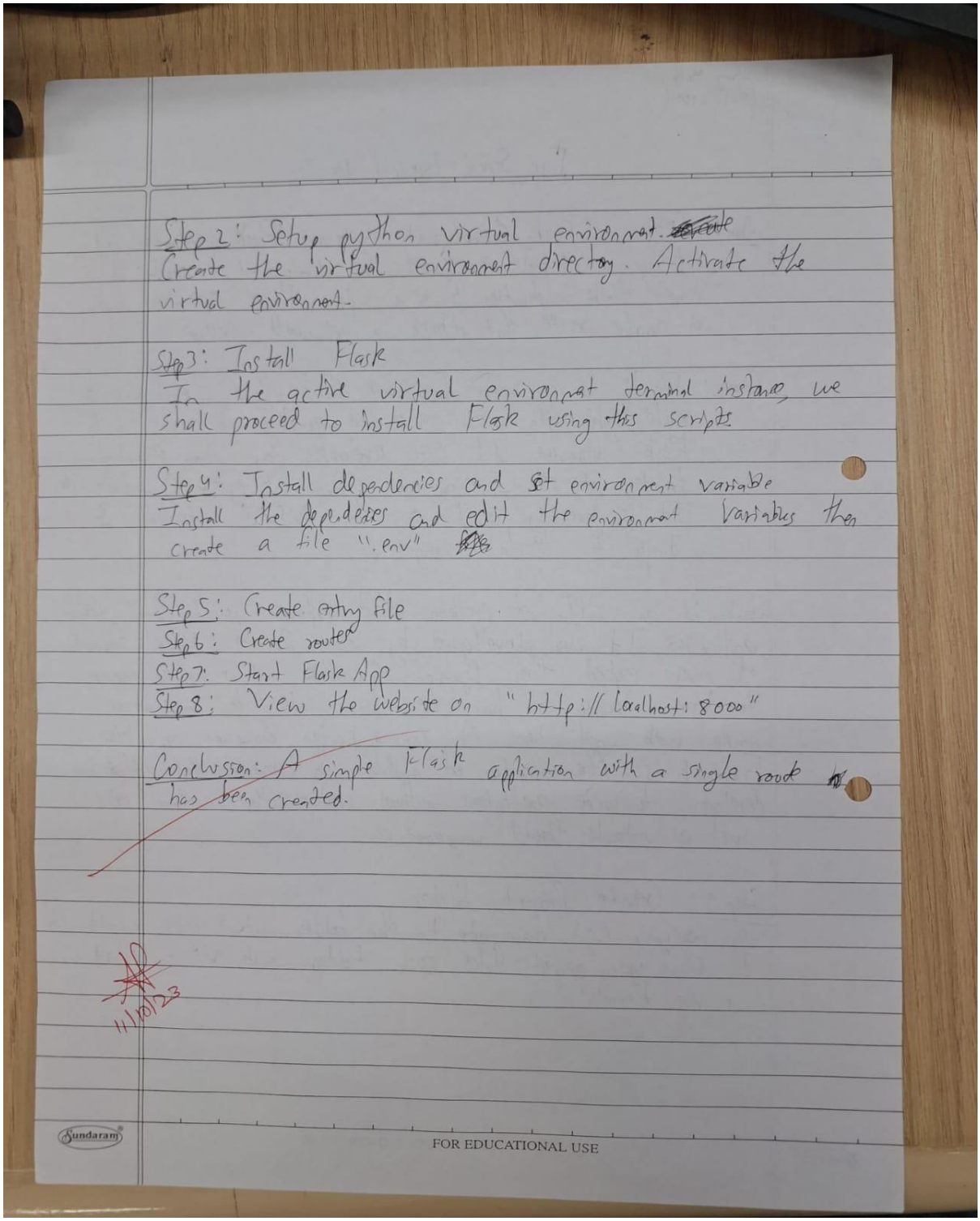
Theory: Flask is a web framework that allows developers to build lightweight web applications quickly & easily with Flask libraries. It was developed by Armin Ronacher, leader of the International Group of Python Enthusiasts (Pocco). It is basically based on the WSGI toolkit & Jinja 2 templating engine

Flask is an API of python that allows us to build web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web application. A web-application framework or Web Framework is the collection of modules & libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc.

Step 1: Create Project folder
On your OS, navigate to the folder where you would like to store your project files open folder with VS code and a new terminal.

Sundaram

FOR EDUCATIONAL USE



Step 2: Setup python virtual environment. ~~Create~~
Create the virtual environment directory. Activate the virtual environment.

Step 3: Install Flask
In the active virtual environment terminal instance, we shall proceed to install Flask using this scripts

Step 4: Install dependencies and set environment variable
Install the dependencies and edit the environment variables then create a file ".env"

Step 5: Create entry file
Step 6: Create routes
Step 7: Start Flask App
Step 8: View the website on "http://localhost:8000"

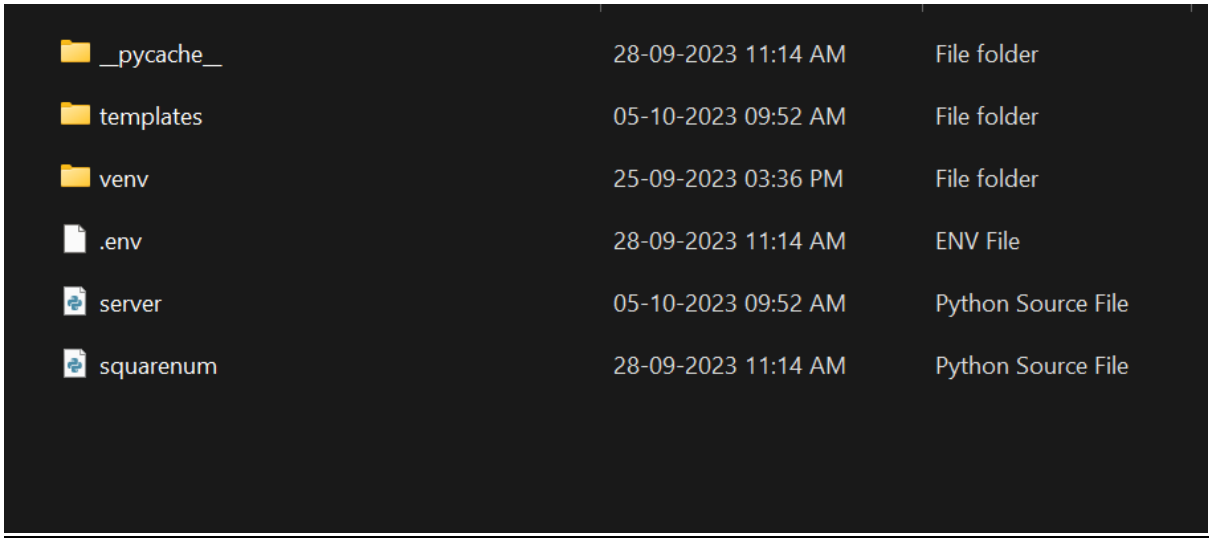
Conclusion: A simple Flask application with a single route has been created.

11/10/23

Sundaram

FOR EDUCATIONAL USE

Code:



__pycache__	28-09-2023 11:14 AM	File folder
templates	05-10-2023 09:52 AM	File folder
venv	25-09-2023 03:36 PM	File folder
.env	28-09-2023 11:14 AM	ENV File
server	05-10-2023 09:52 AM	Python Source File
squarenum	28-09-2023 11:14 AM	Python Source File

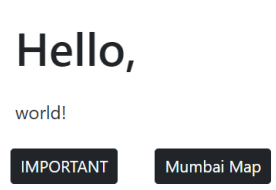
```
PS C:\Users\a21ma\OneDrive\Desktop\Sem5 Codes\Full Stack\Experiment 5, 6 & 7 (Flask)> python -m venv venv
PS C:\Users\a21ma\OneDrive\Desktop\Sem5 Codes\Full Stack\Experiment 5, 6 & 7 (Flask)> .\venv\Scripts\activate
(venv) PS C:\Users\a21ma\OneDrive\Desktop\Sem5 Codes\Full Stack\Experiment 5, 6 & 7 (Flask)> pip install flask

  Collecting flask
    Using cached flask-3.0.0-py3-none-any.whl (99 kB)
  Collecting Werkzeug>=3.0.0 (from flask)
    Downloading werkzeug-3.0.1-py3-none-any.whl (226 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 226.7/226.7 kB 2.3 MB/s eta 0:00:00
  Collecting Jinja2>=3.1.2 (from flask)
    Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
  Collecting itsdangerous>=2.1.2 (from flask)
    Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
  Collecting click>=8.1.3 (from flask)
    Using cached click-8.1.7-py3-none-any.whl (97 kB)
  Collecting blinker>=1.6.2 (from flask)
    Using cached blinker-1.6.3-py3-none-any.whl (13 kB)
  Collecting colorama (from click>=8.1.3->flask)
    Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
  Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
    Using cached MarkupSafe-2.1.3-cp311-cp311-win_amd64.whl (17 kB)
  Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
  Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.1 blinker-1.6.3 click-8.1.7 colorama-0.4.6 fl
  ask-3.0.0 itsdangerous-2.1.2

  [notice] A new release of pip is available: 23.1.2 -> 23.3.1
```

```
(venv) PS C:\Users\a21ma\OneDrive\Desktop\Sem5 Codes\Full Stack\Experiment 5, 6 & 7 (Flask)> pip install python
-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.0
```

```python
from flask import Flask
app = Flask(__name__)
print("Initialized the app! With a name of: ",__name__)

@app.route('/', methods=['GET']) #there is no need to specify the GET method since it is by default called by flask anyway
@app.route('/home') #you can stack multiple routes onto one function so multiple paths will load the same page
def home():
    print("Hey, I'm visiting the home page!")
    return '''
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
        crossorigin="anonymous" />
    <h1 style=margin:20>Hello,</h1>
    <p style=margin:20 >world!<p>
    <a href="/important"
        <button type="button" class="btn btn-dark btn-sm mx-3">IMPORTANT</button>

    <a href="/mumbai"
        <button type="button" class="btn btn-dark btn-sm mx-3">Mumbai Map</button>

    '''
```

```
(venv) PS C:\Users\a21ma\OneDrive\Desktop\Sem5 Codes\Full Stack\Experiment 5, 6 & 7 (Flask)> flask run
Initialized the app! With a name of:  server
 * Serving Flask app 'server.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8000
Press CTRL+C to quit
Hey, I'm visiting the home page!
127.0.0.1 - - [26/Oct/2023 10:55:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Oct/2023 10:55:29] "GET /favicon.ico HTTP/1.1" 404 -
```

Output:



Conclusion: A simple Flask application with a single route has been created