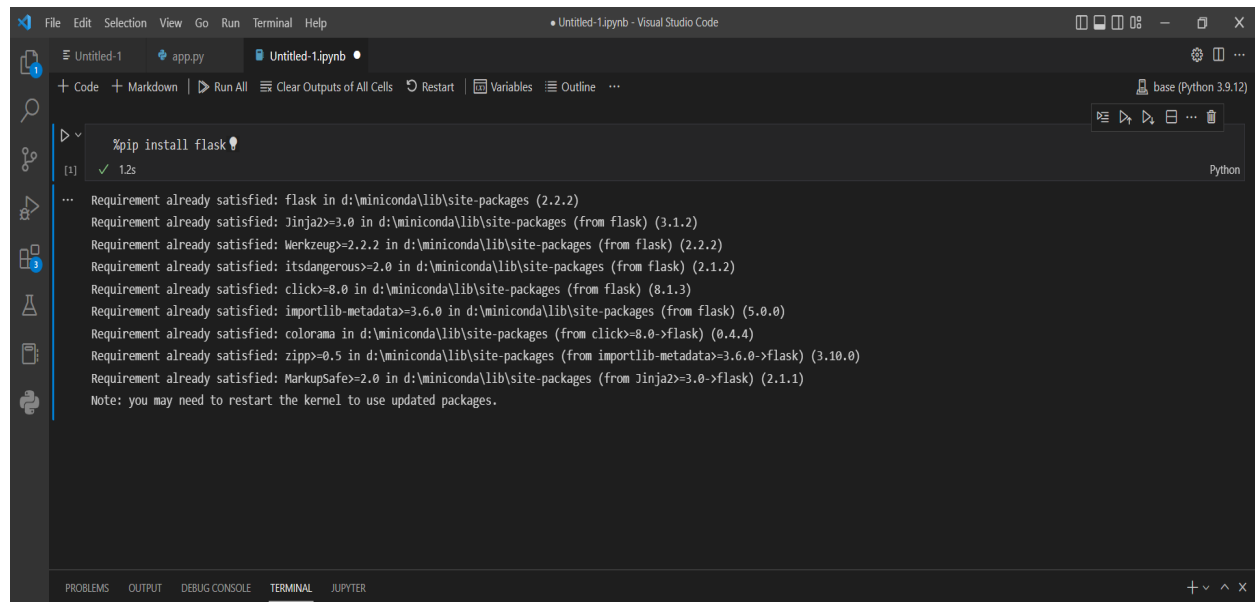


IMPLEMENTING WEB APPLICATION CREATING APIs IN FLASK

Date	12 November 2022
Team ID	PNT2022TMID35589
Project Name	Containment Zone Alerting Application



The screenshot shows a Jupyter Notebook interface within Visual Studio Code. The terminal output displays the command `%pip install flask` and its successful execution. Below the command, a list of requirements is shown, all of which are already satisfied in the current environment. The requirements include Flask (2.2.2), Jinja2 (3.1.2), Werkzeug (2.2.2), itsdangerous (2.0), click (8.1.3), importlib-metadata (5.0.0), colorama (0.4.4), zipp (3.10.0), and MarkupSafe (2.1.1). A note at the bottom suggests restarting the kernel to use the updated packages.

```
%pip install flask
[1] ✓ 12s

...
Requirement already satisfied: flask in d:\miniconda\lib\site-packages (2.2.2)
Requirement already satisfied: Jinja2>=3.0 in d:\miniconda\lib\site-packages (from flask) (3.1.2)
Requirement already satisfied: Werkzeug>=2.2.2 in d:\miniconda\lib\site-packages (from flask) (2.2.2)
Requirement already satisfied: itsdangerous>=2.0 in d:\miniconda\lib\site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in d:\miniconda\lib\site-packages (from flask) (8.1.3)
Requirement already satisfied: importlib-metadata>=3.6.0 in d:\miniconda\lib\site-packages (from flask) (5.0.0)
Requirement already satisfied: colorama in d:\miniconda\lib\site-packages (from click>=8.0->flask) (0.4.4)
Requirement already satisfied: zipp>=0.5 in d:\miniconda\lib\site-packages (from importlib-metadata>=3.6.0->flask) (3.10.0)
Requirement already satisfied: MarkupSafe>=2.0 in d:\miniconda\lib\site-packages (from Jinja2>=3.0->flask) (2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def homepage():
    return "Welcome to Flask API Server!"

if __name__ == '__main__':
    app.run()
```

The image shows the Visual Studio Code interface with a Python file named `app.py` open. The code defines a Flask application with a single route `/` that returns "Welcome to Flask API Server!". The terminal at the bottom shows the command `python app.py` being executed, which results in an `ImportError` because the `flask` module is not found. The error message indicates that the module is partially initialized, likely due to a circular import. The terminal also shows the Flask server starting on `http://127.0.0.1:5000` and serving the application.

```
1
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route("/")
6 def homepage():
7     return "Welcome to Flask API Server!"
8 if __name__ == '__main__':
9     app.run()
10
11
12
```

from flask import Flask
File "d:\HANDYYY\7th Semester\IBM\codes\flask.py", line 1, in <module>
from flask import Flask, request
ImportError: cannot import name 'Flask' from partially initialized module 'flask' (most likely due to a circular import) (d:\HANDYYY\7th Semester\IBM\codes\flask.py)
PS C:\Users\ASUS> & D:/miniconda/python.exe "d:/HANDYYY/7th Semester/IBM/codes/app.py"
* Serving Flask app 'app'
* 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://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [18/Nov/2022 16:56:45] "GET / HTTP/1.1" 200 -

