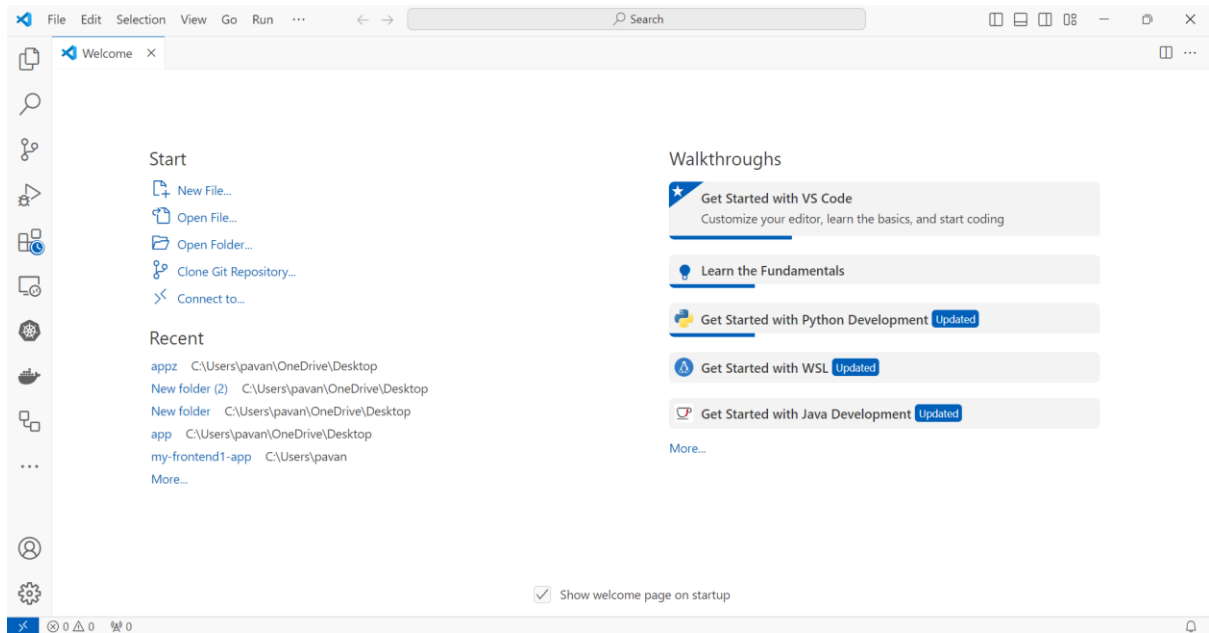
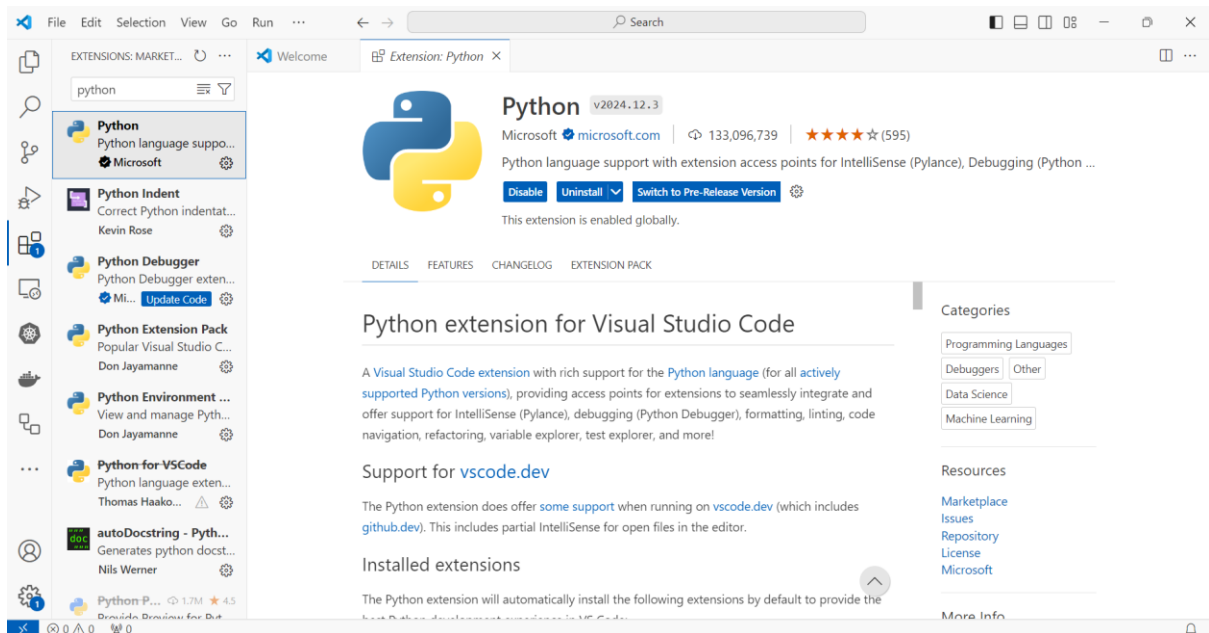


1. L1 - In Local Windows Machine Install and setup Environment Variable for Python
2. L2 - Install Visual Studio Code and Install Python and Terraform Extensions in VS Code
3. L3 - Create Python Console Application to randomly generate OTP kind of secure code
4. L4 - Create Python Console Application to read the contents of .json file and print in the VS Code python console output
5. L5 - Create Python Web Application to using Flask Web Application Framework

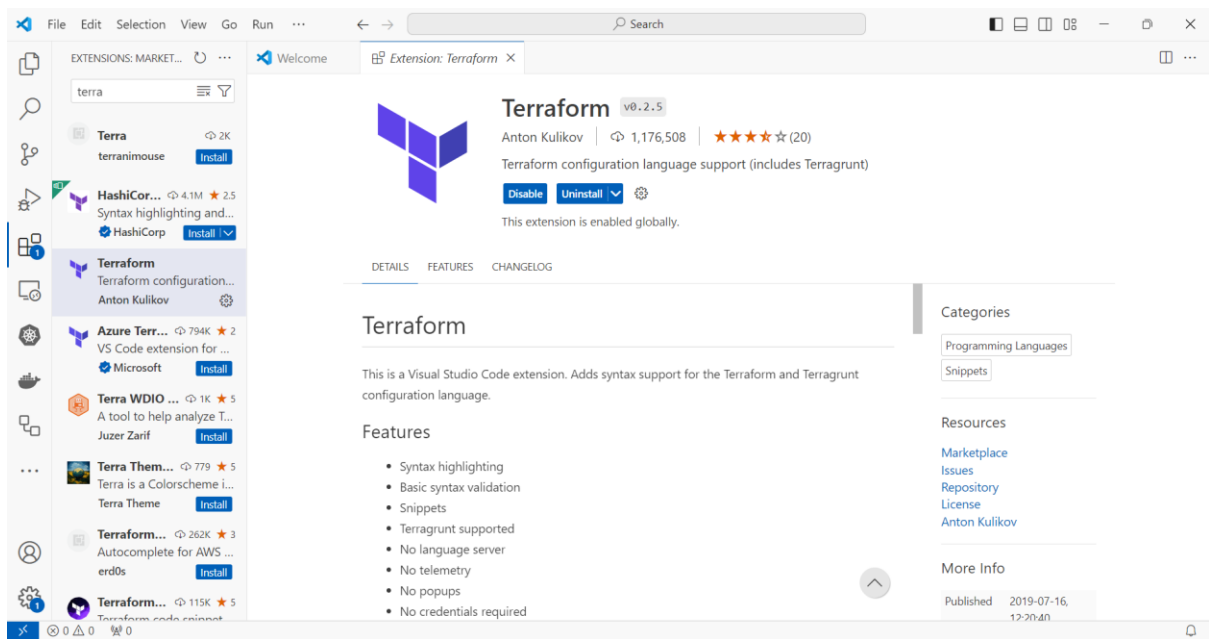
2. Visual Studio Code Installed



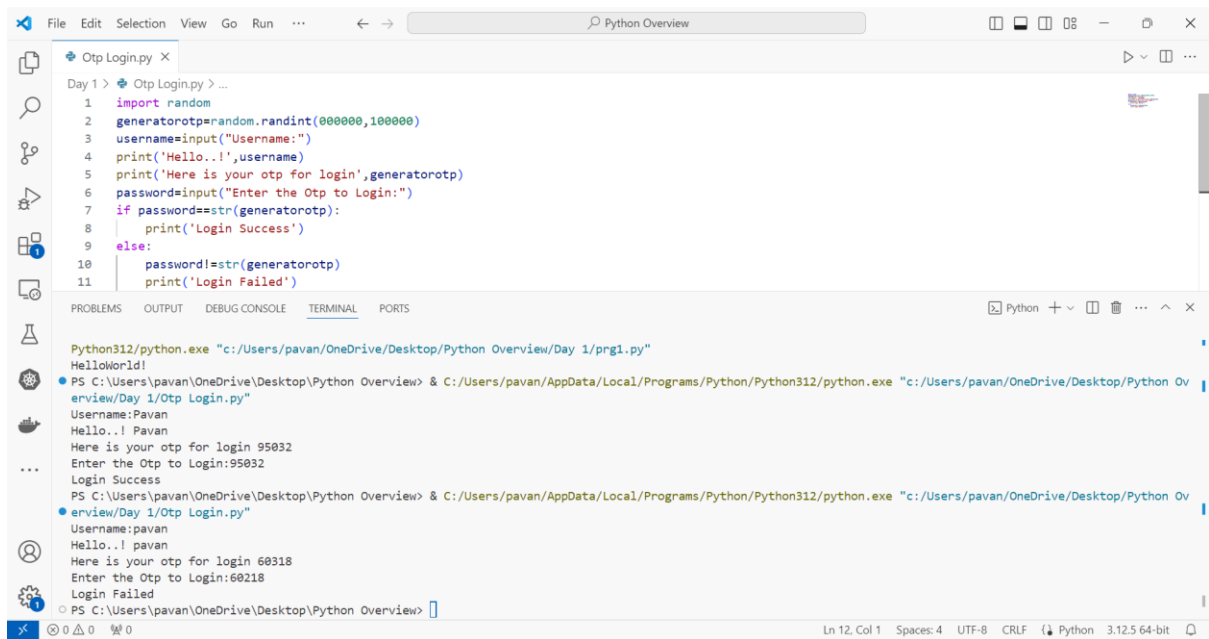
Python Extension



Terraform Extension



3. Python OTP generator



4. Read and print JSON file in python

The first screenshot shows a Python IDE with a file named `simple.json` open. The JSON content is as follows:

```
1 {
2   "name" : "John Doe",
3   "age" : 34,
4   "friends" : [
5     {
6       "name" : "Paul C",
7       "age" : 33
8     }
9   ],
10  "other_key" : null
11 }
```

The second screenshot shows the same IDE with a new file `opnjson.py` open. The Python code in this file reads the JSON file and prints its contents:

```
1 import json
2
3 file_name = "Day 1/simple.json"
4
5 with open(file_name, "r") as f:
6     data = json.load(f)
7
8 print(data)
```

Below the code editor, the terminal window shows the command executed and the output:

```
PS C:\Users\pavan\OneDrive\Desktop\Python Overview> & C:/Users/pavan/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/pavan/OneDrive/Desktop/Python Overview/Day 1/opnjson.py"
{'name': 'John Doe', 'age': 34, 'friends': [{'name': 'Paul C', 'age': 33}], 'other_key': None}
```

5. Flask Web Application

The screenshot displays a development environment for a Flask web application. The top section shows the VS Code interface with the Explorer sidebar on the left, listing files like `__pycache__`, `__init__.py`, `routes.py`, `venv`, `.flaskenv`, and `main.py`. The main editor window shows the `routes.py` file with the following code:

```
app > routes.py > index
1 from app import app
2
3 @app.route('/')
4 def index():
5     return "hello flask!"
```

Below the editor, the TERMINAL panel shows the execution of the application:

```
Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.4 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdangerous-2.2.0
(venv) PS C:\Users\pavan\OneDrive\Desktop\flask> pip install python-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.0.1-py3-none-any.whl.metadata (23 kB)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.0.1
(venv) PS C:\Users\pavan\OneDrive\Desktop\flask> flask run
 * Serving Flask app 'main.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://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [24/Aug/2024 19:36:54] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Aug/2024 19:36:54] "GET /favicon.ico HTTP/1.1" 404 -
```

The bottom section of the image shows a web browser window with the address bar set to `127.0.0.1:5000`. The browser's address bar and tabs are visible, and the page content displays "hello flask!"