# Placement Empowerment Program

## Cloud Computing and DevOps Centre

**Write a Python Script to Monitor an Application : Create a Python script that sends periodic HTTP requests to your application and alerts you if it’s down.**

Name: OVIYA G Department: IT



# Introduction

Ensuring the availability of your application is critical for maintaining user satisfaction and trust. Monitoring your application proactively can help you detect issues before they impact your users. By creating a Python script, you can automate the process of checking the application's health. The script will periodically send HTTP requests to your application and alert you if it detects any downtime or issues, enabling you to take immediate action.

# Objectives

The primary objective of this Python script is to provide an automated and efficient way to monitor your application's availability. The script will:

Send periodic HTTP requests to your application's endpoint.

Check the response status to determine if the application is running correctly.

Send alerts (e.g., email, SMS, or log entry) if the application is down or returning unexpected response.

# Step-by-Step Overview

## Step 1: Install Python from Microsoft Store

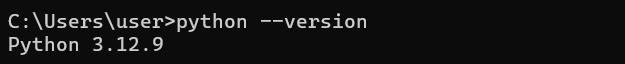
1. Open the **Microsoft Store** on your computer.
2. In the search bar, type **"Python"** and press **Enter**.
3. Find the latest version of Python (e.g., **Python 3.x.x**), and click on it.
4. Click the **Install** button to install Python on your system.
   * This will automatically add Python to your system’s PATH environment variable.
   * 

## Step 2: Verify Python Installation

1. Open the **Command Prompt (CMD)**:
2. Type the following command to verify that Python is installed:

### python --version

1. This should return the version of Python installed, e.g., Python 3.x.x.
2. If you see the version number, Python is correctly installed.



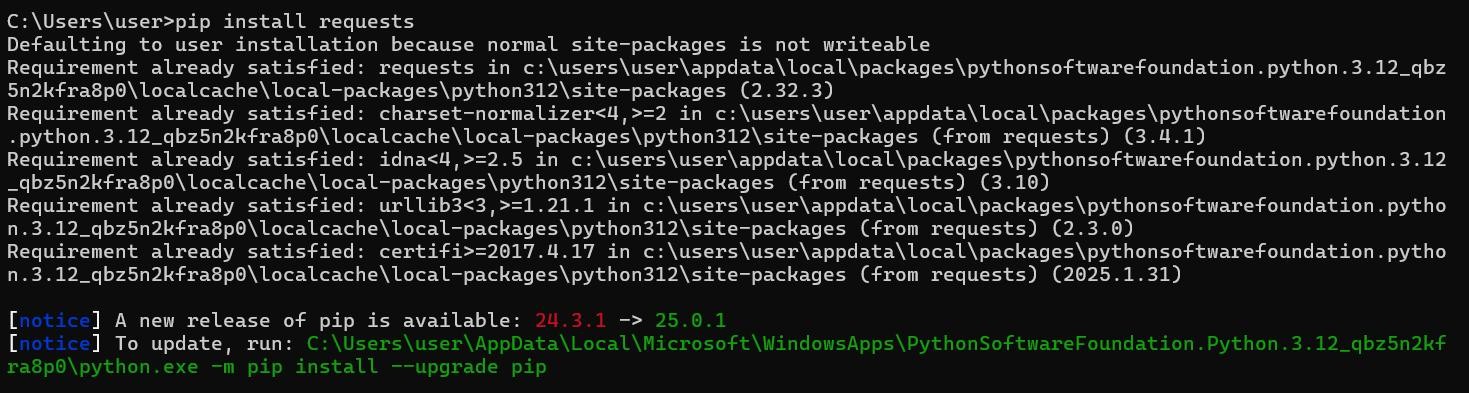
## Step 3: Install Required Libraries (requests, smtplib)

1. In **Command Prompt (CMD)**, type the following command to install the

**requests** library:

### pip install requests

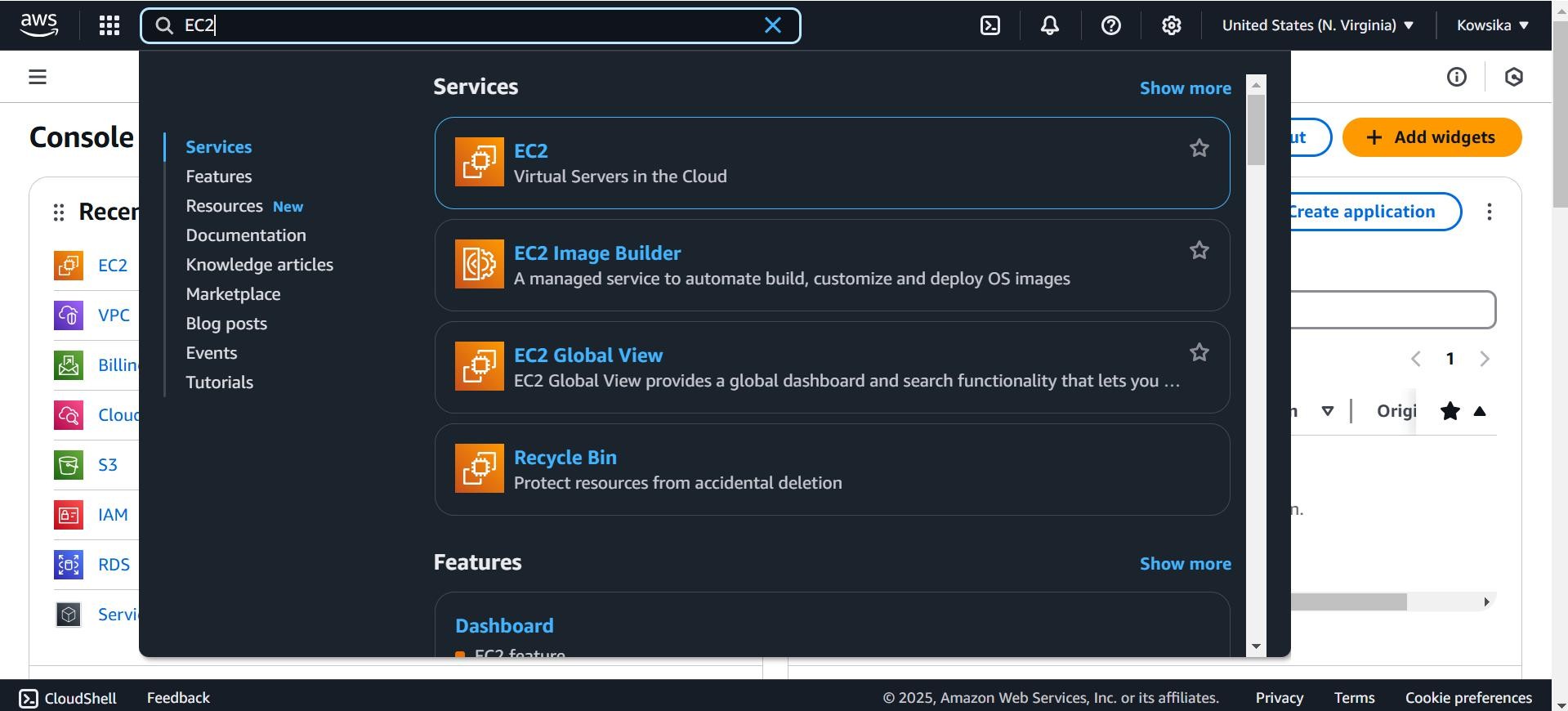
1. The **smtplib** library is included with Python by default, so no installation is needed for it.

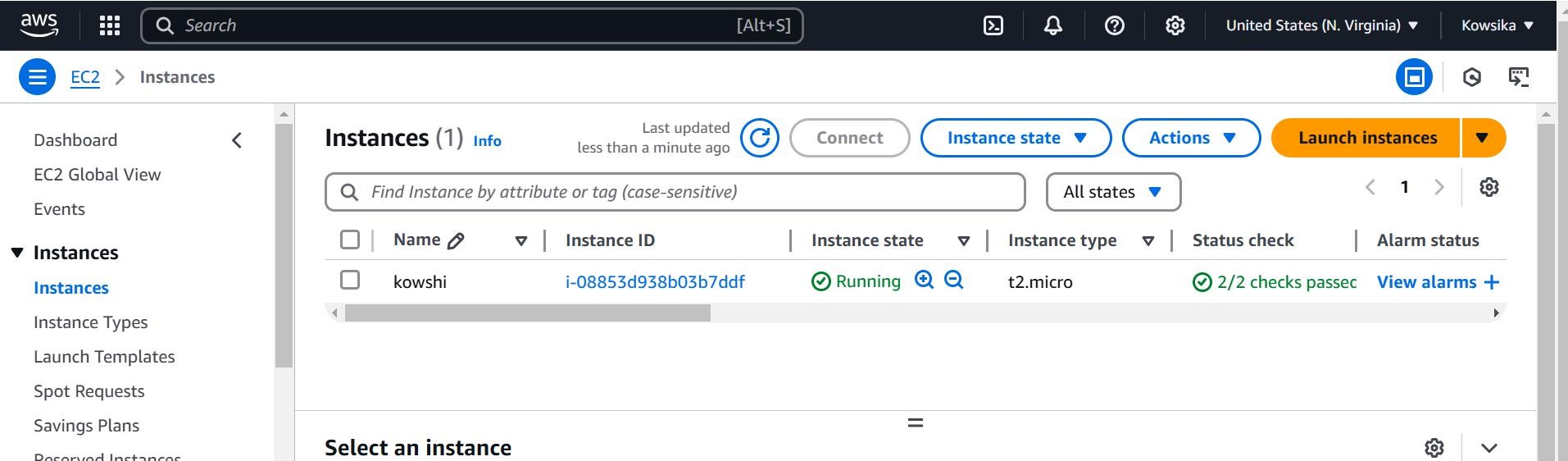


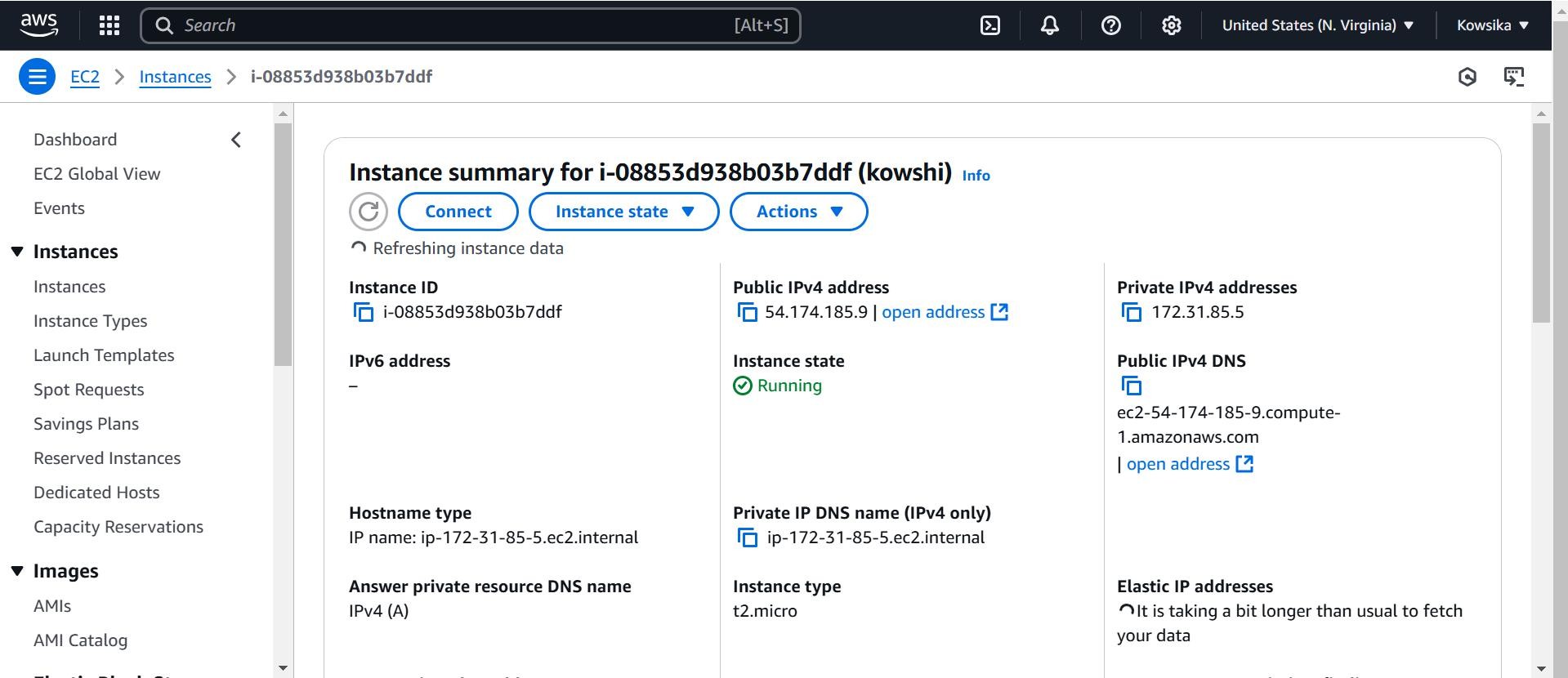
1. ​

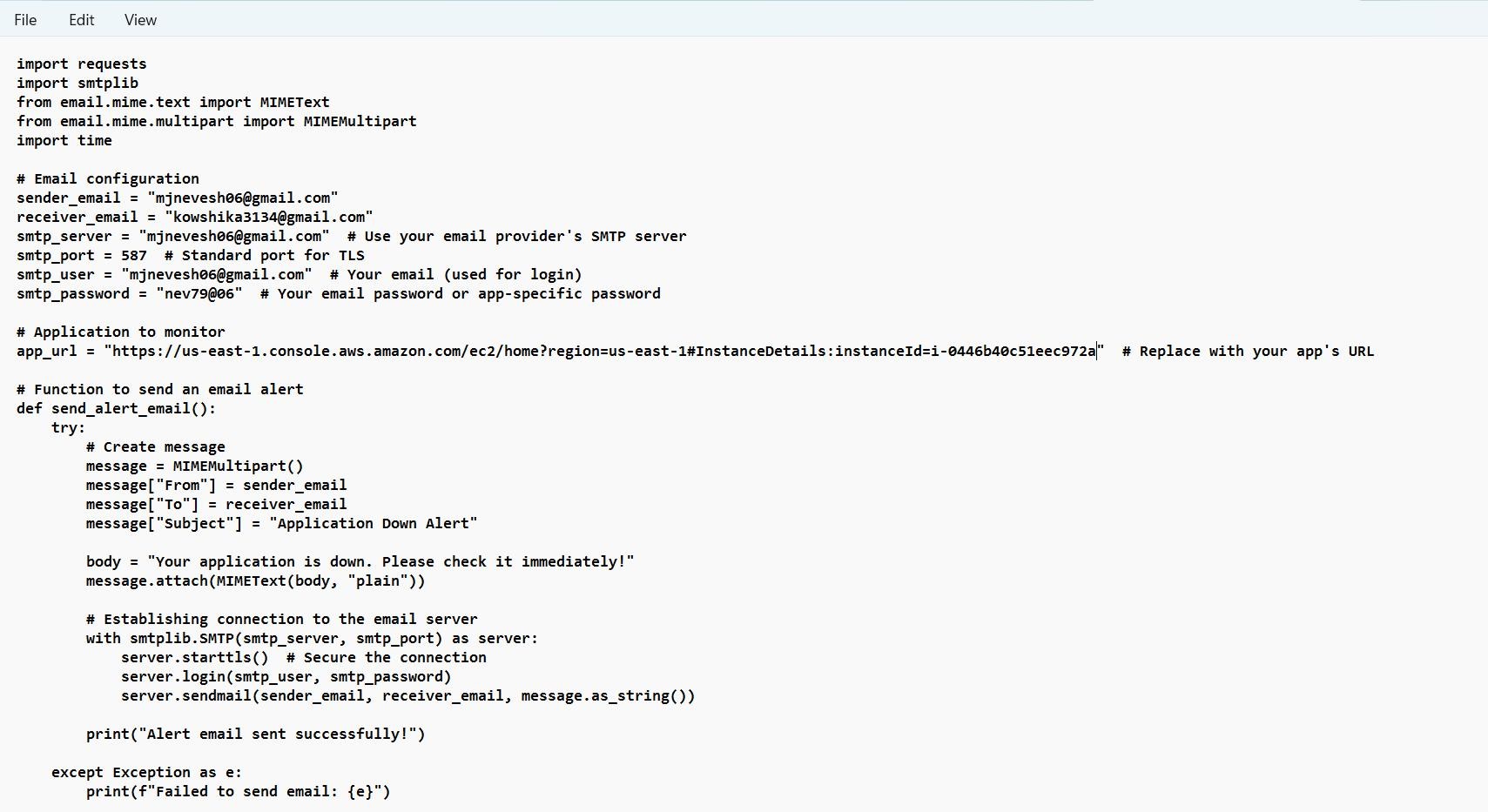
## Step 4: Write the Python Script

1. Create a EC2 Instance
2. Open any **text editor** (e.g., Notepad, VS Code).
3. Copy and paste the Python script to monitor your EC2 instance (from your PoC).
4. Change [your\_email@example.com](mailto:your_email@example.com) to your actual Gmail address (e.g., [your\_email@gmail.com).](mailto:your_email@gmail.com)
5. Set smtp\_user to your **Gmail** address as well.
6. Enter your **app-specific password** (not your Gmail password) for the smtp\_password field. If you don't have an app-specific password, you can create one in your Google Account settings (in the **Security** section under **App passwords**)
7. Also Change the app\_url to your Instance URL
8. Save the file with a **.py** extension, e.g., monitor\_app.py









***Step 6: Run the Python Script***

1. In **Command Prompt (CMD)**, navigate to the folder where the Python script is saved using the cd command:

cd path\to\your\script\directory

1. Run the script with the following command:

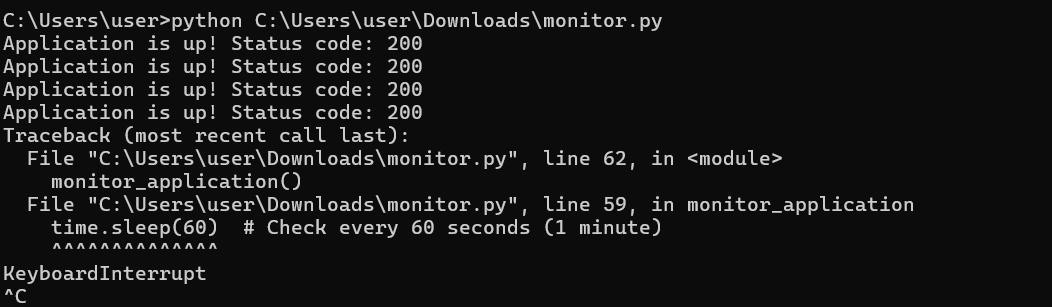
python monitor\_app.py

.



## Step 7: Stop the Script

To stop the script at any time, press **Ctrl + C** in the **Command Prompt** window.

.

Outcome

By creating and running a Python script to monitor your application, you will achieve the following outcomes:

**Proactive Monitoring:** Your application will be monitored continuously, allowing you to detect issues before they affect users. This proactive approach ensures minimal downtime and maintains user satisfaction.

**Automated Alerts:** The script will automatically send alerts if the application is down or not responding as expected. These alerts can be configured to notify you via email, SMS, or any other preferred method, enabling you to take immediate action.

**Improved Reliability:** Regular monitoring and timely alerts contribute to the overall reliability of your application. You'll be able to address potential problems quickly, reducing the risk of extended downtime.

**Enhanced User Experience:** By ensuring your application is always available, you provide a consistent and reliable experience for your users, which can lead to increased trust and loyalty.

**Data Insights:** Monitoring data can help you identify patterns and potential areas for improvement in your application. This information can be valuable for optimizing performance and planning future updates.