



OTP Verification System

Python Capstone Project

Prashantsagar Uppara (S8083)

Contents

Description	Slide
Project Charter	3
IDE Used	5
Libraries Used	6
Process Flow	7
Robust Features Inbuilt In The System	8
Challenges Faced	9
User Friendly Graphical User Interface	10
Code Snippet	11

Project Charter

Problem Statement	Project Requirements
<ul style="list-style-type: none">• Developing an OTP (One-Time Password) verification system in Python. The system should generate a 6-digit OTP and send it to the user's email address for verification• Upon receiving the OTP, the user should enter it into the system for validation• If the entered OTP matches the generated OTP, access should be granted; otherwise, access should be denied	<ul style="list-style-type: none">• Implement a function to generate a 6-digit OTP randomly• Develop a function to simulate sending the OTP to the user's email address• Create a function to prompt the user to enter the OTP received in their email• Implement a function to verify if the entered OTP matches the generated OTP• Ensure proper error handling and user-friendly prompts throughout the system• Allow the user to retry OTP entry in case of incorrect input

Project Charter


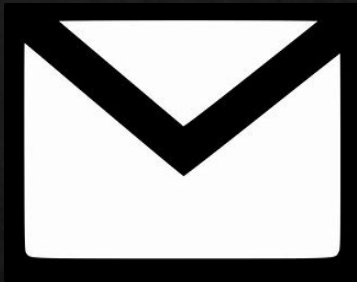
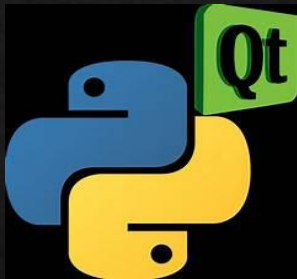
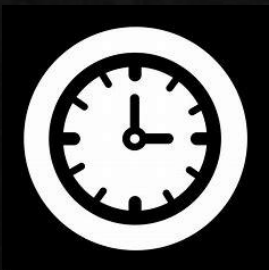

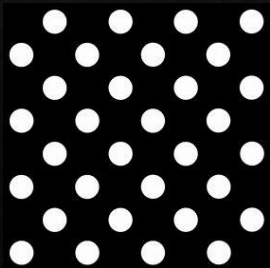
Project Objective	Project Scope
<ul style="list-style-type: none">• Creating an OTP verification system capable of sending OTP to the user and verifying it	<ul style="list-style-type: none">• Creating python-based OTP verification system• Building user friendly & secure GUI, if possible• Documentation & test cases

IDE Used

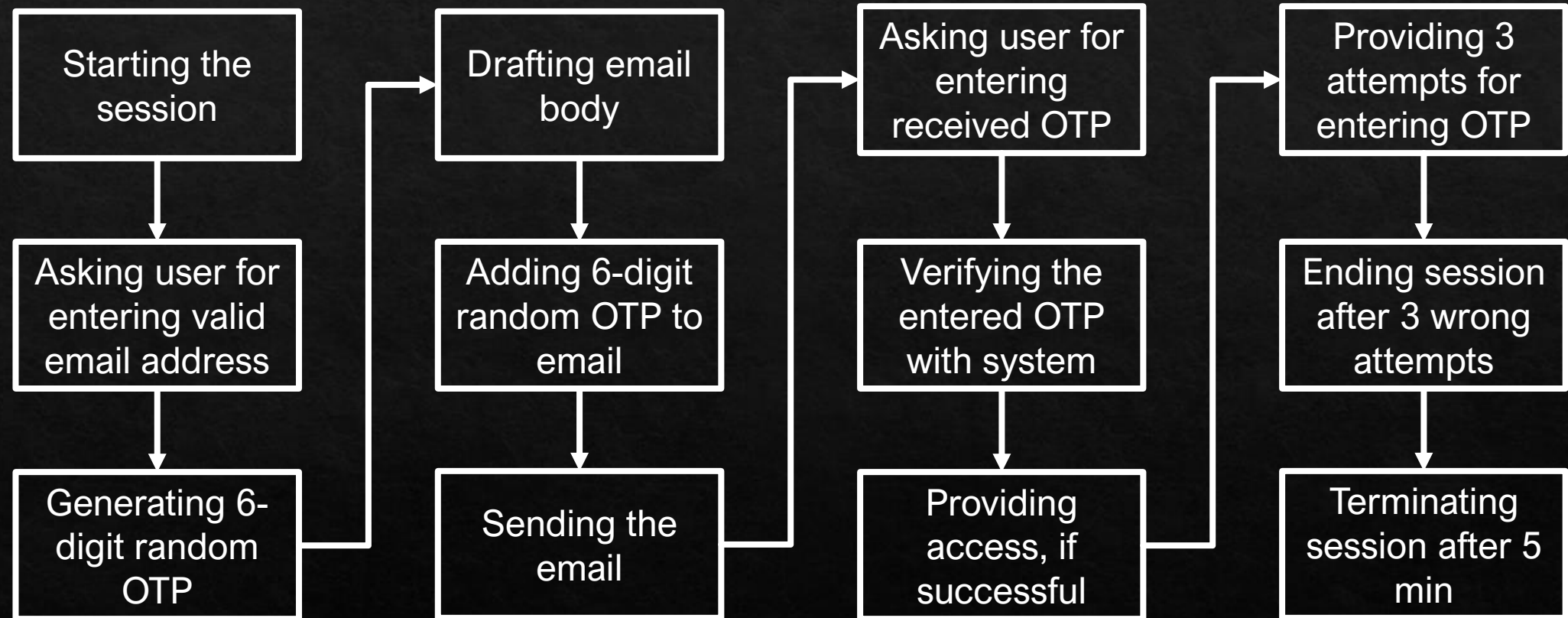


Jupyter Notebook

Libraries Used

		
SMTPLIB Simple Mail Transfer Protocol	email 4.0.2 (MIME) Multipurpose Internet Mail Extensions	PyQt6 Python bindings for Qt version 6
		
Time	re Regular Expressions	random Generate pseudo-random numbers

Process Flow



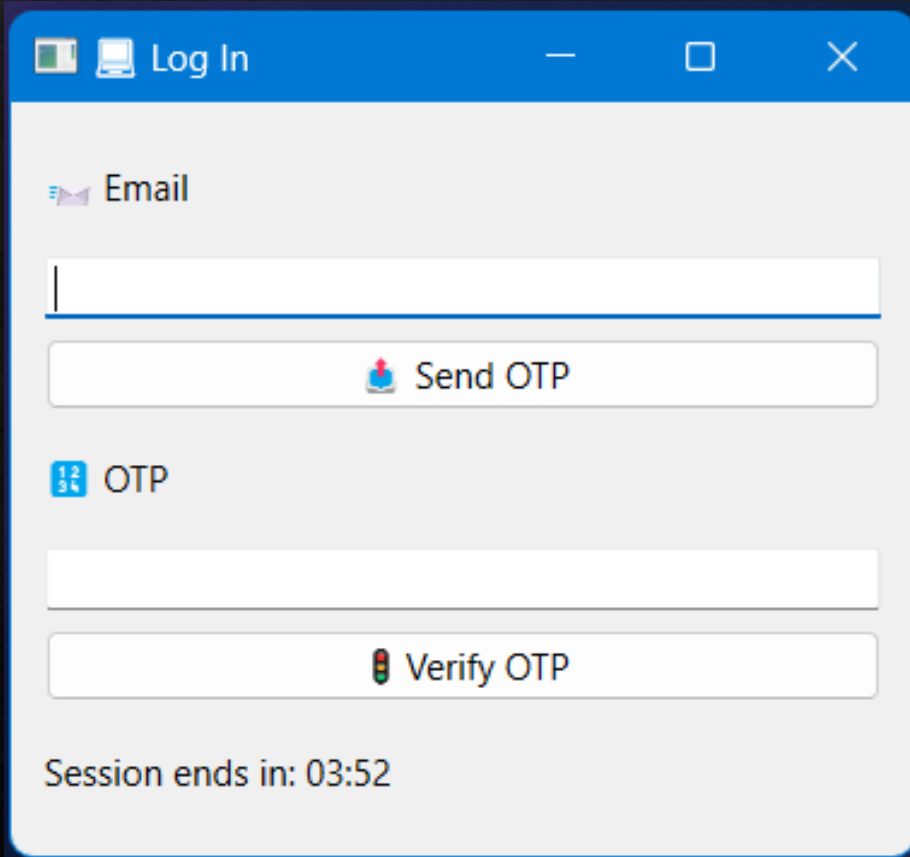
Robust Features Inbuilt In The System

- ◆ Each session is valid for 5 minutes, after which the session terminates automatically & no OTP can be generated once the OTP is sent to the entered email address
- ◆ Entered email format is validated before OTP generation, random strings are not accepted
- ◆ After sending the OTP to email, message is shown to the user
- ◆ 3 attempts are provided for entering OTP, after each attempt input box is frozen for 30 sec for ensuring that the user checks & enters right OTP
- ◆ After 3 attempts, session is terminated
- ◆ Random strings & characters are not accepted as OTP

Challenges Faced

- ◇ Generating appropriate email body with subject & email body
- ◇ Generating GUI form
- ◇ Integrating GUI with Jupyter notebook
- ◇ Integrating considered features in the code & GUI
- ◇ Generating time-based sessions & making it work parallelly with other considered features

User Friendly Graphical User Interface



Log In

Email

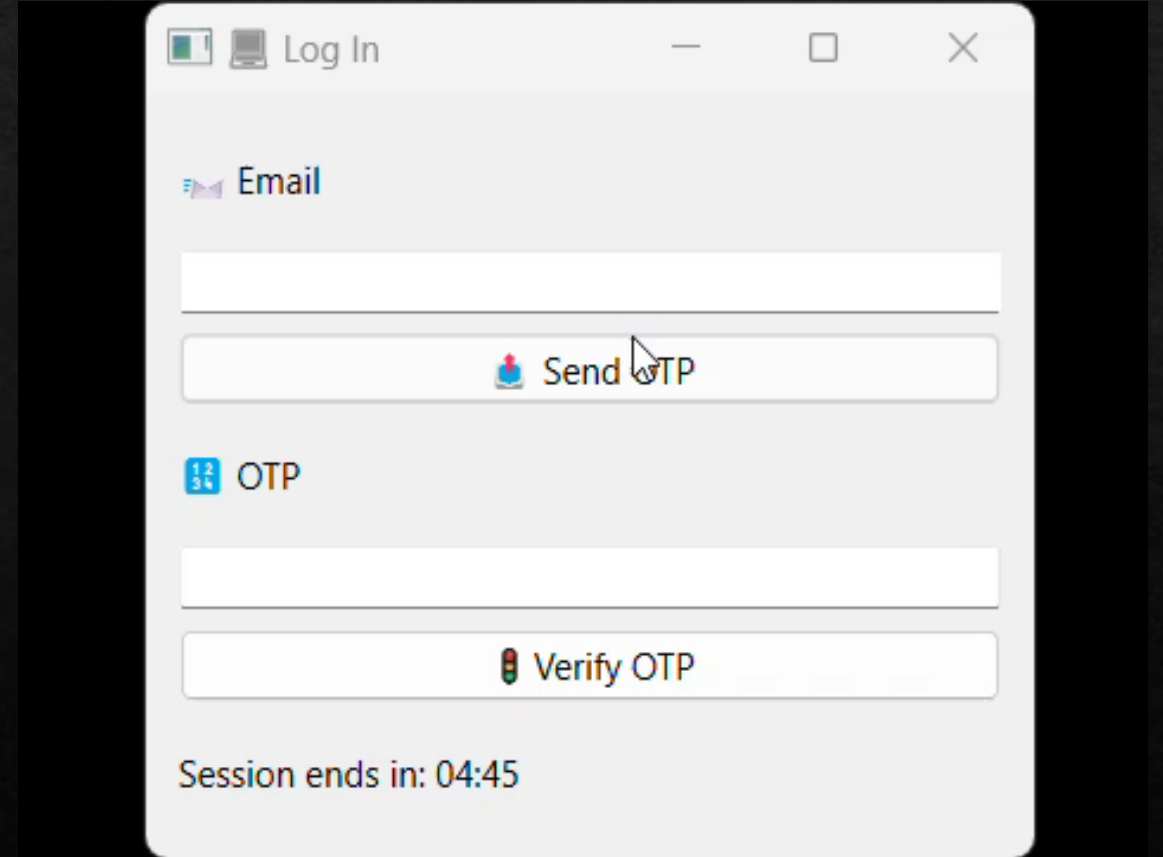
Send OTP

OTP

Verify OTP

Session ends in: 03:52

GUI (OTP verification form)



Log In

Email

Send OTP

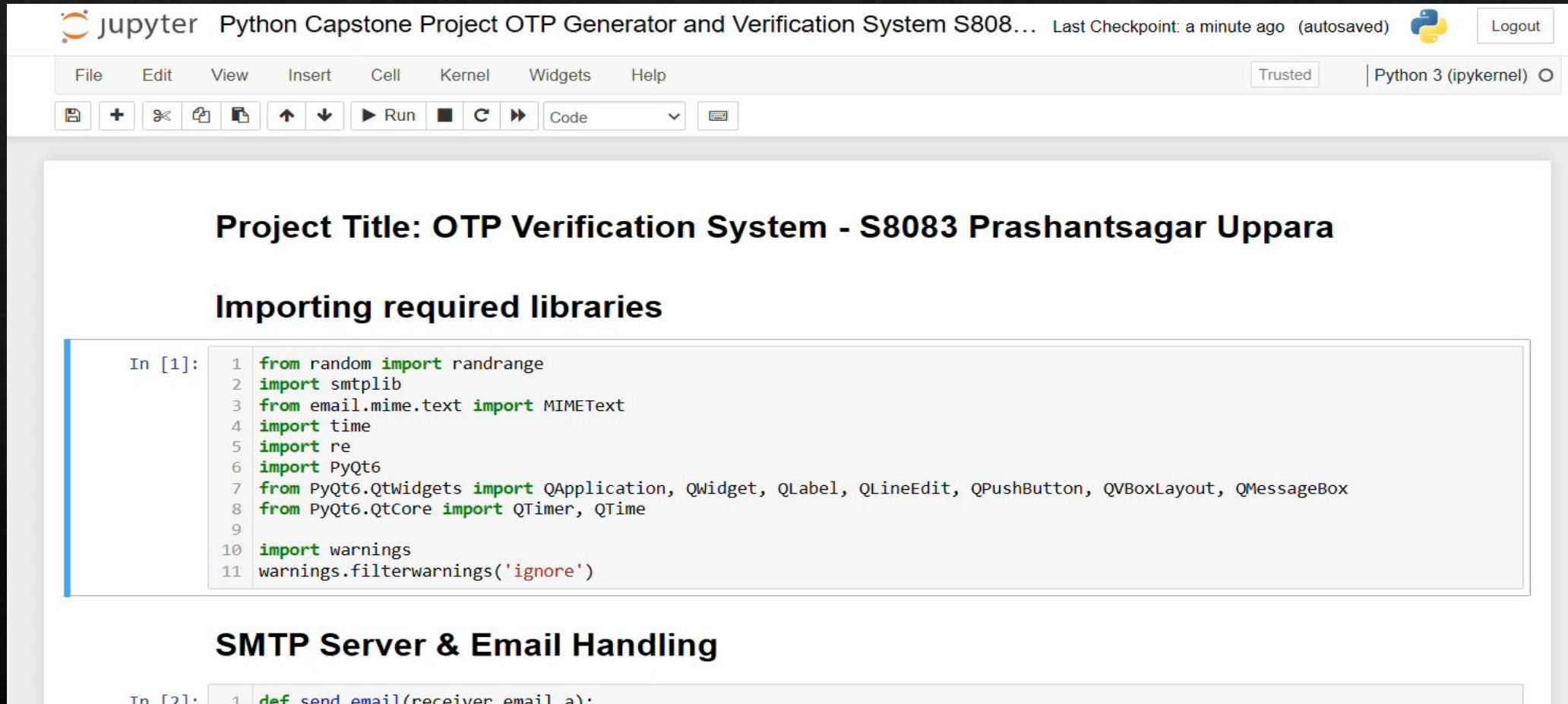
OTP

Verify OTP

Session ends in: 04:45

Play the above video to check working

Code Snippet



The screenshot shows a Jupyter Notebook interface with the title "Python Capstone Project OTP Generator and Verification System S808...". The interface includes a top bar with the Jupyter logo, the title, and a "Logout" button. Below the top bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar are buttons for "Trusted" and "Python 3 (ipykernel)". Below the menu bar is a toolbar with icons for saving, adding, deleting, and running code, along with a dropdown menu set to "Code".

The notebook content is displayed in a white area with a light gray border. It features two sections:

- Project Title: OTP Verification System - S8083 Prashantsagar Uppara**
- Importing required libraries**

The first code cell, labeled "In [1]:", contains the following Python code:

```
1 from random import randrange
2 import smtplib
3 from email.mime.text import MIMEText
4 import time
5 import re
6 import PyQt6
7 from PyQt6.QtWidgets import QApplication, QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout, QMessageBox
8 from PyQt6.QtCore import QTimer, QTime
9
10 import warnings
11 warnings.filterwarnings('ignore')
```

The second section, **SMTP Server & Email Handling**, shows the beginning of a code cell labeled "In [2]:":

```
1 def send_email(receiver_email,a):
```

Play the above video to check the code



Questions

Thankyou