Gehad Khaled 7980

Lab 2 Networks

The Email Client Application is a Python-based GUI application that allows users to send and receive emails. It uses
the smtplib and imaplib libraries for email communication and tkinter for the graphical user interface. The application is designed to provide a simple and intuitive interface for managing email communication.

Features

1. **Send Emails**: Users can send emails by providing the sender's email, password, recipient's email, subject, and body.

2. **Receive Emails**: Users can fetch the latest email from their inbox and view its content.
3. **Graphical User Interface**: A user-friendly GUI built with tkinter for easy interaction.
4. **Error Handling**: Displays appropriate error messages for invalid inputs or failed operations.
5. **Notifications**: Desktop notifications for received emails using the plyer library.

Application Structure

The application is organized into the following files:

- **gui.py**: Contains the main GUI logic and integrates the send and receive functionalities.

- send.py: Implements the send_mail function for sending emails using the SMTP protocol.
- recieve.py: Implements the receive_mail function for fetching the latest email using the IMAP protocol.

Installation

Prerequisites

1. Python 3.6 or later installed on your system.
2. Internet connection for email communication.

Steps

1. Downlaod the code
2. Navigate to the project directory:

```
cd email-client
```

3. Install the required dependencies:

```
pip install -r requirements.txt
```

How to Run the Application

1. Open a terminal or command prompt.
2. Navigate to the directory containing the gui.py file.
3. Run the application:

```
python gui.py
```

4. The GUI window will open, allowing you to send and receive emails.

Usage Instructions

1. **Sending Emails**:
    - Enter the sender's email and password.

- Enter the recipient's email, subject, and body of the email.
- Click the "Send Email" button.
- A success or error message will be displayed based on the operation's outcome.

2. **Receiving Emails**:
    - Enter the sender's email and password.
    - Click the "Receive Email" button.
    - The latest email's content will be displayed in a message box, and a desktop notification will appear.

Dependencies

The application relies on the following Python libraries:

1. **smtplib**: For sending emails using the SMTP protocol.
2. **imaplib**: For receiving emails using the IMAP protocol.
3. **email**: For parsing email messages.
4. **tkinter**: For creating the graphical user interface.
5. **plyer**: For displaying desktop notifications.

Testing Process and Results

Testing Scenarios

1. **Send Email**:
   - Tested sending emails to various recipients using Gmail and Outlook accounts.

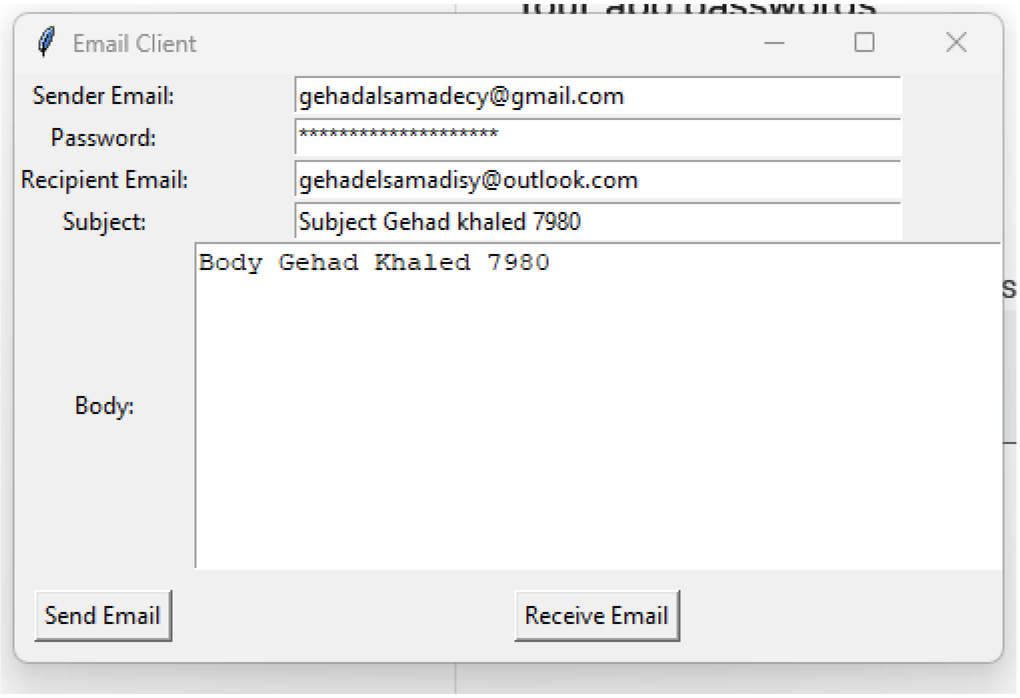
   - Verified that the email is delivered successfully and appears in the recipient's inbox.

2. **Receive Email**:
   - Tested fetching the latest email from Gmail and Outlook accounts.
   - Verified that the email content is displayed correctly in the GUI and as a desktop notification.
3. **Error Handling**:
   - Tested with invalid email credentials.
   - Verified that appropriate error messages are displayed.
4. **GUI Functionality**:
   - Tested all input fields and buttons for proper functionality.

    - Verified that the GUI responds correctly to user actions.

Results

- All features worked as expected.
- Emails were sent and received successfully.
- Error messages were displayed for invalid inputs or failed operations.
- The GUI was responsive and user-friendly.



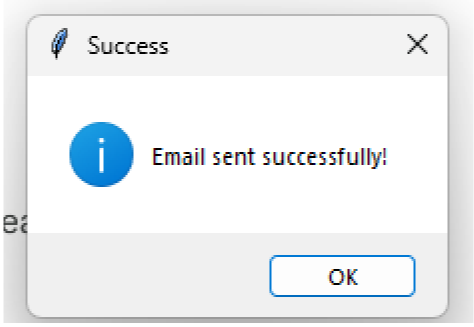
Sender Email: gehadalsamadecy@gmail.com

Password: ********************

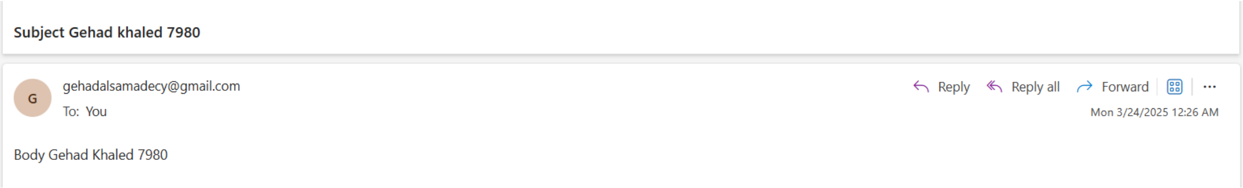
Recipient Email: gehadelsamadisy@outlook.com

Subject: Subject Gehad khaled 7980

Body: Body Gehad Khaled 7980



Email sent successfully!



Subject Gehad khaled 7980

gehadalsamadecy@gmail.com
To: You

Mon 3/24/2025 12:26 AM

Body Gehad Khaled 7980