A project report on

**Mail verification using SMTP protocol**

**Course code: CSE 420**

**Course Title: Mobile and Wireless Communication (Sessional)**

**Level-4 Semester-I**

**Team Members:**

Md : Sazzad Hossain(1902024)

Mehedi Hasan (1902044)

Md.Moshiur Rahman (1702060)

Md:Jamil Hossain(1902058)

Ahsanul Zahid (1902007)

**Department Of**

**Computer Science and Engineering**



**Hajee Mohammad Danesh Science and Technology University,5200**

**Dinajpur**

1. **Introduction:**

This project involves generating a 6-digit one-time password (OTP) and sending it via email to authenticate users. The OTP is generated randomly using Python libraries and sent to the provided email address. The user must then enter the OTP they receive to gain access. They have 3 attempts to enter the correct OTP before being denied access. If correct, a confirmation email is sent stating verification was successful. This helps prevent unauthorized access to accounts and ensures only authorized individuals can access private information or services.  
The project aims to develop a system for Gmail verification using the SMTP (Simple Mail Transfer Protocol) protocol. Email verification is a crucial aspect of user authentication and communication in various applications. Implementing Gmail verification through SMTP offers a reliable and widely used method for ensuring the validity of email addresses.

**2. Objectives:**

* To implement Gmail verification using the SMTP protocol.
* To develop a user-friendly interface for email verification.
* To ensure secure and efficient communication between the application and Gmail servers.
* To handle error cases and provide appropriate feedback to users.

**3. Methodology:**

The project involves the following steps:

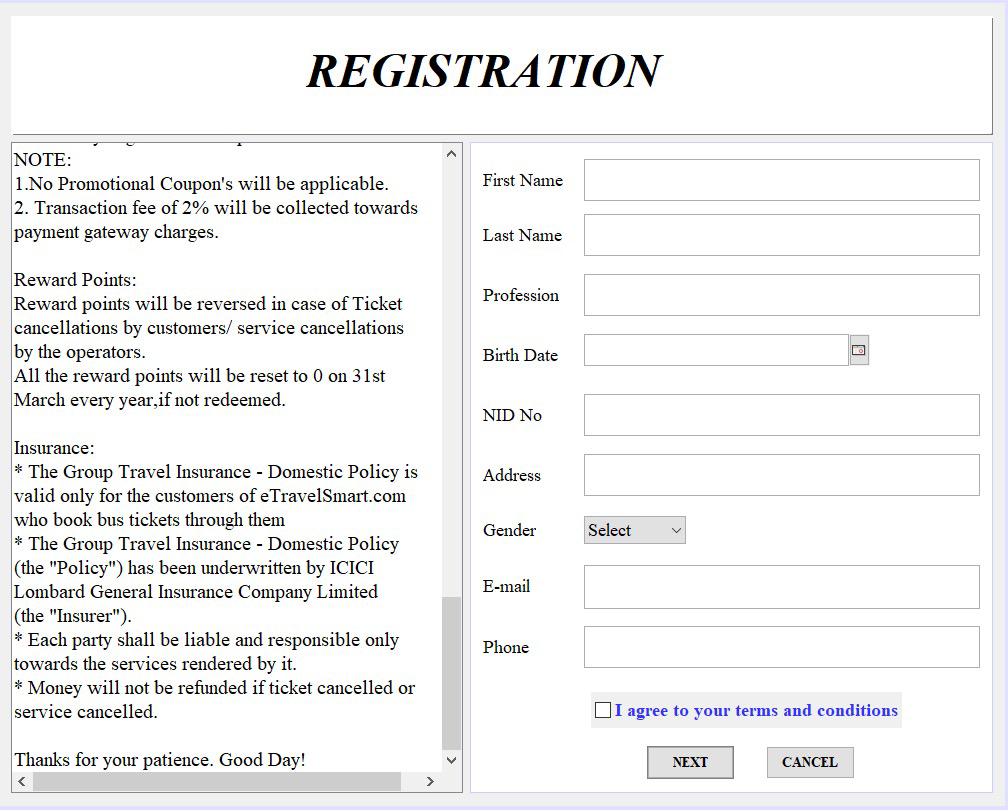
* **User Interface Design:** Designing an intuitive user interface for email verification, where users can input their email addresses and receive verification status.
* **SMTP Configuration:** Configuring the application to communicate with Gmail's SMTP server using the appropriate credentials and settings.
* **Email Sending:** Sending a verification email to the provided Gmail address using the SMTP protocol.
* **Email Parsing:** Parsing the response received from Gmail's SMTP server to determine the verification status.
* **Error Handling:** Implementing error handling mechanisms to address issues such as invalid email addresses, network errors, and SMTP server errors.
* **Feedback Mechanism:** Providing clear feedback to users regarding the verification status of their email addresses.

**4. Implementation:**

The project can be implemented using a programming language such as Python along with libraries like smtplib for SMTP communication and email for composing and sending emails. Below is a high-level overview of the implementation:

* **SMTP Configuration:** Set up the application to establish a connection with Gmail's SMTP server using the appropriate host, port, and authentication credentials (username and password).
* **Email Composition:** Compose a verification email containing a unique verification link or code.
* **Email Sending:** Use the SMTP protocol to send the verification email to the user's Gmail address.
* **Response Handling:** Parse the response received from the SMTP server to determine whether the email was successfully sent and accepted by Gmail.
* **Verification Status:** Provide feedback to the user indicating whether the email address is valid and verified.

**Registration and OTP generator page:**

 **Fig 1:Registration Page**

**Validity Check and set password:**

****

**Fig 2: Validity Check and set password**

**5. Conclusion:** The project demonstrates the implementation of Gmail verification using the SMTP protocol in Python. By following the SMTP standards and utilizing the **smtplib** library, we can establish a connection to Gmail's SMTP server, compose and send verification emails, and confirm successful delivery.

**6. Future Enhancements:**

* Implement email address validation to ensure the correctness of email addresses.
* Handle errors and exceptions gracefully to improve the robustness of the system.
* Explore additional features such as email template customization and email tracking.

In conclusion, this project provides a foundation for implementing email verification systems using SMTP protocols, which can be extended and adapted for various applications and use cases.