INTERNSHIP PROJECT-10 DOCUMENT

ON “Email Client Simulation (JAVA)”

Submitted by:

Navpreet Singh (INTERN)

Submitted To:-

Kanduri Abhinay (FOUNDER)

RITHIN VERMA (CTO)

INDEX

INTRODUCTION:-

The Email Client Simulation project is a Java-based application that demonstrates how to send emails programmatically using the JavaMail API. It simulates the behavior of an email client by allowing users to send emails with a subject and message body to a specific recipient. The project provides hands-on experience in working with SMTP (Simple Mail Transfer Protocol) and handling authentication securely. It also helps learners understand real-world email automation tasks such as notifications, alerts, and communication within applications.

SOFTWARE REQUIREMENTS

- **Programming Language:** Java (version 8 or above)
- **Libraries Used:** JavaMail API (javax.mail)
- **Development Environment:** Any Java IDE (Eclipse, IntelliJ IDEA, NetBeans) or command-line (javac, java)
- **Operating System:** Platform-independent (Windows, Linux, macOS)
- **Java Runtime Environment:** Java SE with JavaMail API configured
- **x** Required for connecting to the SMTP server

DESIGN:-

The system design is centered around a single main class, EmailClientSimulation, which encapsulates the entire logic of sending an email. The design focuses on separation of configuration and functionality

INPUT:-

Inputs are specified in the main method or taken dynamically through user prompts (optional modification).

- **Receiver Email:** Recipient's email address (e.g., "receiver@example.com")
- **Subject:** Subject line of the email (String)
- **Message Text**: Body/content of the email (String)

The sender's email and password are pre-configured in the code or securely fetched from environment variables.

IMPLEMENTATION:-

The project is implemented using JavaMail API and follows these steps:

- Define the sender's credentials (email and password).
- Configure SMTP properties for Gmail (host, port, TLS enable, authentication).
- Create a Session object using an authenticator that returns the sender's credentials.
- Construct the email using MimeMessage, specifying From, To, Subject, and Text fields.
- Send the email using Transport.send() method.
- Handle exceptions gracefully and display appropriate success or error messages.
- The program demonstrates connectivity to Gmail's SMTP server (smtp.gmail.com) using port 587.

TESTING:-

Testing was performed by executing multiple test cases under different scenarios:

- **Valid Receiver Email:** Email sent successfully.
- **Invalid Receiver Email:** Exception thrown and handled properly.
- **Incorrect Password:** Authentication failed, program displayed an error.
- **Network Disconnection:** Connection error handled gracefully.
- **Different Email Content:** Verified successful sending with different message lengths and subjects.

All tests confirmed correct configuration of SMTP properties and reliable email transmission under valid conditions.

ADVANTAGES:-

- Demonstrates real-world use of JavaMail API for communication.
- Provides clear understanding of SMTP configuration and authentication.
- Platform-independent and lightweight.

- Can be extended to include advanced features (attachments, CC/BCC, HTML templates).
- Useful for learning network programming and email automation in Java.
- Reinforces key concepts of Java I/O, networking, and API integration.

CONCLUSION:-

The Email Client Simulation project effectively simulates an email client's core functionality using JavaMail API. It illustrates how to send emails programmatically, manage SMTP configurations, and handle exceptions in network-based applications. This project provides practical exposure to Java networking, making it a valuable learning experience for students and developers interested in backend automation or communication-based systems.

REFERENCES:-

- Oracle JavaMail Documentation: https://javaee.github.io/javamail/
- Oracle Java Tutorials: https://docs.oracle.com/javase/tutorial/
- Gmail SMTP Configuration Guide: https://support.google.com/mail/answer/7126229
- GeeksforGeeks JavaMail API Tutorials: https://www.geeksforgeeks.org/javamail-api/