# Email Sender

## Introduction:

This project involves the creation of an Automated Email Sender Application using Java. The application reads recipient details from a CSV file and sends personalized emails using a predefined template. It utilizes JavaMail API for sending emails via an SMTP server. The primary objective of this project is to automate the process of sending emails, making it efficient and scalable for various use cases such as marketing campaigns, notifications, and more.

## Project Structure

### 2.1 Configuration and Metadata Files

* **.classpath**: Eclipse class path file.
* **.project**: Eclipse project file.
* **pom.xml**: Maven project file containing dependencies and build configurations.
* **.settings/org.eclipse.jdt.core.prefs**: Eclipse Java Development Tools settings.
* **.settings/org.eclipse.m2e.core.prefs**: Eclipse Maven settings.

### 2.2 Source Code

* **EmailSender.java**: Main class responsible for configuring and sending emails.
* **MainApp.java**: Entry point for the application.

### 2.3 Resources

* **config.properties**: Configuration properties file containing email server settings.
* **email\_template.txt**: Template file used for formatting emails.
* **recipients.csv**: CSV file containing the list of email recipients.

### 2.4 Compiled Classes and Build Artifacts

* **target/classes/**: Directory containing compiled class files and build artifacts.
* **target/classes/META-INF/MANIFEST.MF**: Manifest file for the compiled JAR.
* **target/classes/META-INF/maven/**: Maven-related metadata for the project.

## Project Team

* Dhanush Jayadevan (Team Leader)
* Prajakta Chavan
* Ananya Rathore
* Shivansh Sharma
* Vivek Som

## Project Technologies

* Java
* JavaMail API
* Maven
* CSV File Handling
* SMTP Protocol

## Project Setup

### Prerequisites:

* Java Development Kit (JDK) 1.8 or higher
* Apache Maven
* An SMTP email account (e.g., Gmail)

### Installation Steps:

1. **Clone the Repository:**

git clone <repository-url>

cd <repository-directory>

1. **Configure Email Credentials:**

* Update the config.properties file with your email ID and app password.

1. **Set Up Maven:** Ensure Maven is installed and configured in your environment. Navigate to the project directory and run:

mvn clean install

1. **Run the Application:** Execute the MainApp.java file from your IDE or use the following Maven command:

mvn exec:java -Dexec.mainClass="com.example.emailsender.MainApp"

### Configuration Details:

* **config.properties** should contain the SMTP configuration: **operties** should contain the SMTP configuration:

## Project Structure:



## Source Code:

**MainApp.java**





**EmailSender.java**



### config.properties

Contains configuration settings such as SMTP server details, authentication credentials, and other email-related configurations. Sample Configuration:

### email\_template.txt

Defines the format and content of the emails. The template supports placeholders for dynamic content. Sample Template:

### recipients.csv

A CSV file listing the email recipients. Each row typically contains recipient details such as name and email address. Sample CSV:

## 

## Pom.xml



## Project Resources/References:

* **JavaMail API Documentation**: <https://javaee.github.io/javamail/>
* **Maven Documentation**: <https://maven.apache.org/guides/index.html>
* **SMTP Configuration for Gmail**: <https://support.google.com/mail/answer/7126229>

## Project Risks:

* **Security Risks:** Exposing email credentials in the code can lead to security vulnerabilities. Mitigation: Use environment variables or secure credential storage.
* **Email Quotas:** Sending a large number of emails can exceed the SMTP provider's quota limits. Mitigation: Monitor email sending limits and implement throttling if necessary.
* **Network Issues:** Network connectivity problems can disrupt the email sending process. Mitigation: Implement retry mechanisms and error handling to manage network failures.

## Additional Comments:

### Security Considerations

* **Credentials Management:** It's crucial to handle email credentials securely. Avoid hardcoding sensitive information such as email addresses and passwords directly in the source code. Instead, consider using environment variables or secure vaults to manage sensitive data.
* **App-Specific Passwords:** If using Gmail, ensure to use an app-specific password instead of the main account password to enhance security.

### Error Handling

* **Robust Error Handling:** Implement robust error handling to manage potential issues such as network failures, SMTP server issues, or invalid recipient addresses. This can include retry mechanisms, logging errors for troubleshooting, and notifying administrators in case of critical failures.

### Scalability

* **Email Throttling:** For large email campaigns, implement email throttling to avoid hitting SMTP server limits or being marked as spam. This can be achieved by spacing out email sends or batching them.

### Enhancements

* **HTML Email Support:** Consider enhancing the application to support HTML emails, allowing for richer and more engaging email content.
* **Dynamic Content:** Further enhance email personalization by incorporating more dynamic content based on recipient data.

### Testing

* **Test Environment:** Set up a testing environment to validate the email sending functionality without sending emails to real recipients. Use mock SMTP servers or test email accounts to ensure everything works as expected before going live.

### Compliance

* **Regulatory Compliance:** Ensure compliance with email sending regulations such as GDPR, CAN-SPAM, and other applicable laws. This includes providing recipients with a way to opt out and respecting their privacy preferences.

### Documentation

* **User Documentation:** Provide comprehensive user documentation to help users understand how to configure and use the application. This can include setup guides, configuration details, and troubleshooting tips.

These additional comments aim to provide insights and considerations to further enhance the security, functionality, and reliability of the EmailSender project.