



EMAIL-CLIENT SOFTWARE

In java

Project Overview

This project aims to explore the world of email client software, delving into its history, key features, popular applications, and the factors to consider when choosing the right one.

Project Objectives

- **Understand the Evolution of Email:** Trace the journey of email from its early days to its current state.
- **Explore Key Features of Email Clients:** Analyze essential features such as inbox management, organization, search capabilities, and more.
- **Compare Popular Email Clients:** Evaluate leading email clients like Outlook, Gmail, Thunderbird, and others based on their features, user interface, and pricing.
- **Factors to Consider When Choosing an Email Client:** Discuss criteria such as security, compatibility, customization options, and mobile integration.
- **Future Trends in Email Clients:** Explore emerging trends and technologies that may shape the future of email.

Project Structure

1. Introduction

- Brief overview of email and its significance
- Definition of email client software
- Importance of choosing the right email client

2. History of Email

- Early development of email
- Key milestones and innovations
- Evolution of email clients

3. Key Features of Email Clients

- Inbox management and organization
- Search and filtering capabilities
- Calendar and scheduling
- Contact management
- Customization options
- Mobile integration

4. Popular Email Clients

- Outlook

- Gmail
- Thunderbird
- Other notable options
- Comparative analysis of features, user interface, and pricing

Code

```
import java.util.*;

import javax.mail.*;

import javax.mail.internet.*;

public class EmailClient {

    public static void main(String[] args) {

        try {

            // Set up email properties

            Properties props = new Properties();

            props.put("mail.smtp.auth", "true");

            props.put("mail.smtp.starttls.enable", "true");

            props.put("mail.smtp.host", "smtp.gmail.com"); // Replace with your SMTP server

            props.put("mail.smtp.port", "587"); // Replace with your SMTP port


            // Create authentication object

            Authenticator auth = new Authenticator() {

                protected PasswordAuthentication getPasswordAuthentication() {

                    return new PasswordAuthentication("your_email@gmail.com",

"your_password"); // Replace with your email and password

                }

            };


            // Create session

            Session session = Session.getInstance(props, auth);
```

```
// Compose email

Message message = new MimeMessage(session);

message.setFrom(new InternetAddress("your_email@gmail.com")); // Replace with
your email

message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse("recipient_email@example.com")); // Replace with recipient's email

message.setSubject("Test Email");

message.setText("This is a test email sent using Java Mail API.");


// Send email

Transport.send(message);


System.out.println("Email sent successfully!");
} catch (MessagingException e) {
    e.printStackTrace();
}
}
}
```