

Class 230125

이메일 발송 처리

이메일 발송 기능 활용

- 단순 메일 용도
- 상품 구매 내역 전달
- 예약 확인 정보 전달
- 회원 가입 시 인증 처리(AJAX)
- 기타 많은 부분에서 다양하게 활용

SMTP(simple mail transfer protocol) 서버 구성 없이 메일 사용

- 네이버 메일 이용
- 구글 메일 이용
- 기타 메일 서비스 이용



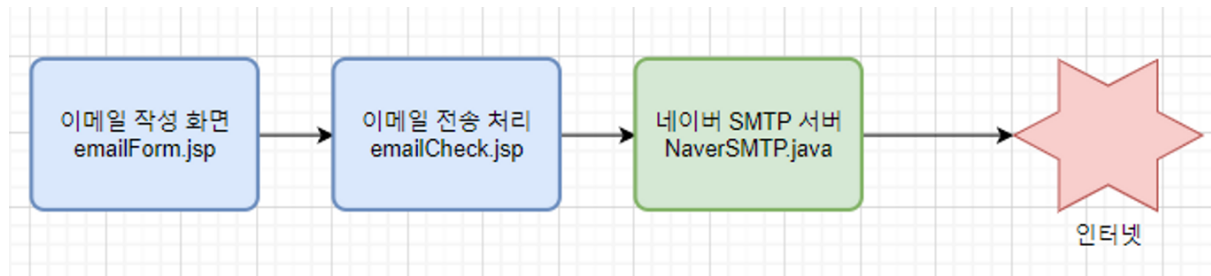
네이버 / 구글 등의 메일 서비스를 이용하면 이메일 발송을 쉽게 이용할 수 있지만 한 번 설정한 전송인의 메일 주소를 변경할 수 없는 단점이 있다.

네이버 이메일

POP3 / IMAP

- **SMTP**
 - 메일 발신에 사용
- **POP3**
 - 메일 수신에 사용
 - 메일 서버에 저장된 메일을 사용자의 컴퓨터나 스마트폰으로 가져와서 확인 할 수 있게 해 줌. 이 때 주로 outlook 과 같은 프로그램을 주로 사용.
- **IMAP**
 - 메일 수신에 사용
 - POP3와 다른 점은 중앙 서버에서 동기화가 이루어지기 때문에 같은 계정으로 연결 된 모든 장치에서 똑같은 내용이 보여지게 됨. 예를 들면 스마트 폰에서 메일을 삭제하면 태블릿에서도 같이 지워짐.

메일 전송 프로세스



- JavaMail API (compat) 1.4.7 jar 필요
- JavaBeans(TM) Activation Framework 1.1.1 jar 필요

▼ emailForm.jsp (이메일 작성 폼 만들기)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>emailForm.jsp</title>
</head>
<body>
<h3>이메일 전송하기</h3>
<form action="emailCheck.jsp" method="post">
<table>

<tr><td>
    보내는 사람 :
    <input type="text" name="from" value="ro1864@naver.com" readonly>
</td></tr>

<tr><td>
    받는 사람 :
    <input type="text" name="to">
</td></tr>

<tr><td>
    제목 :
    <input type="text" name="subject">
</td></tr>

<tr><td>
    형식 :
    TEXT<input type="radio" name="format" value="text" checked>
    HTML<input type="radio" name="format" value="html">
</td></tr>

<tr><td>
    <textarea rows="5" name="content"></textarea>
</td></tr>

<tr><td>
    <button>전송하기</button>
</td></tr>

</table>
</form>
</body>
</html>
  
```

▼ emailCheck.jsp (이메일 처리 페이지 만들기)

```

<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="email.NaverSMTP"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.Map"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");

    Map<String, String> emailInfo = new HashMap<>();

    emailInfo.put("from", request.getParameter("from")); // 보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
    emailInfo.put("to", request.getParameter("to")); // 받는 사람
  %>
  
```

```

emailInfo.put("subject", request.getParameter("subject")); // 제목
emailInfo.put("format", request.getParameter("format").equals("html")
    ? "text/html;charset=UTF-8"
    : "text/plain;charset=UTF-8"); // 메일의 형식을 설정
String content = request.getParameter("content");

if(request.getParameter("format").equals("text")) {
    emailInfo.put("content", content); // 내용
} else {
    // 줄바꿈 변경
    content = content.replace("\r\n", "<br>");

    // 내용을 담을 템플릿html 경로
    String templatePath = application.getRealPath("/EmailDesign/emailDesign.html");
    BufferedReader br = new BufferedReader(new FileReader(templatePath));

    String htmlContent = "";
    String line;
    while((line = br.readLine()) != null) {
        htmlContent += line;
    }
    br.close();
    htmlContent = htmlContent.replace("__CONTENT__", content);

    emailInfo.put("content", htmlContent); // 내용
}

try {
    NaverSMTP smtpServer = new NaverSMTP(); // 메일 전송 객체 생성
    smtpServer.emailSending(emailInfo);
    out.print("이메일 전송 성공");
} catch(Exception e) {
    out.print("이메일 전송 실패");
    e.printStackTrace();
}
%>

```

▼ NaverSMTP.java (네이버 메일 서버 전송 클래스)

```

package email;

import java.util.Map;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class NaverSMTP {
    private final Properties serverInfo; // 서버 정보
    private final Authenticator auth; // 인증 정보

    public NaverSMTP() {
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.naver.com"); // 메일 서버 지정
        serverInfo.put("mail.smtp.port", "465"); // 포트 번호 거의 대부분 통일
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class",
            "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2"); // 맥에서 반드시

        auth = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("ro1864", "n5flanwnd.");
            }
        };
    }

    public void emailSending(Map<String, String> mailInfo) throws MessagingException {
        // 1. 세션 생성 (네이버/다음/구글 메일 서버에 사용자 인증)
        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);
        // 2. 메시지 작성
    }
}

```

```

        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from")));
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to")));
        msg.setSubject(mailInfo.get("subject"));
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));
        // 3. 전송
        Transport.send(msg);
    }
}

```

모달 띄우기 (bootstrap & JQuery)

▼ emailForm.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String msg = request.getParameter("msg");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>emailForm.jsp</title>
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQf"
</head>
<body>
<h3>이메일 전송하기</h3>
<form action="emailCheck.jsp" method="post">
<table>

    <tr><td>
        <label class="form-label">보내는 사람</label>
        <input class="form-control" type="text" name="from" value="ro1864@naver.com" readonly>
    </td></tr>

    <tr><td>
        <label class="form-label">받는 사람</label>
        <input class="form-control" type="text" name="to">
    </td></tr>

    <tr><td>
        <label class="form-label">제목</label>
        <input class="form-control" type="text" name="subject">
    </td></tr>

    <tr><td>
        <label class="form-label">형식</label>
        TEXT<input class="" type="radio" name="format" value="text" checked>
        HTML<input class="" type="radio" name="format" value="html">
    </td></tr>

    <tr><td>
        <label class="form-label">내용</label>
        <textarea class="form-control" rows="5" name="content"></textarea>
    </td></tr>

    <tr><td>
        <button class="btn btn-info">전송하기</button>
    </td></tr>

</table>
</form>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-5" id="exampleModalLabel">메시지</h1>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<%= msg!=null && msg.equals("success") ? "메일 전송을 성공했습니다." : "메일 전송을 실패했습니다." %>
</div>
<div class="modal-footer">

```

```

        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">닫기</button>
        <!-- <button type="button" class="btn btn-primary">Save changes</button> -->
    </div>
</div>
</div>
<!-- JavaScript Bundle with Popper -->
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxg0cBQBxU=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenuU1KfDBIe4zVF0s0G1h" crossorigin="anonymous"></script>
<script>
$(function(){
    var msg = '<%= msg %>';
    console.log(msg);
    if(msg != 'null') {
        $('#exampleModal').modal('show');
    }
});
</script>
</body>
</html>

```

▼ emailCheck.jsp (이메일 처리 페이지 만들기)

```

<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="email.NaverSMTP"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.Map"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");

    Map<String, String> emailInfo = new HashMap<>();

    emailInfo.put("from", request.getParameter("from"));    // 보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
    emailInfo.put("to", request.getParameter("to"));        // 받는 사람
    emailInfo.put("subject", request.getParameter("subject")); // 제목
    emailInfo.put("format", request.getParameter("format").equals("html")
        ? "text/html;charset=UTF-8"
        : "text/plain;charset=UTF-8");    // 메일의 형식을 설정
    String content = request.getParameter("content");

    if(request.getParameter("format").equals("text")) {
        emailInfo.put("content", content); // 내용
    } else {
        // 줄바꿈 변경
        content = content.replace("\r\n", "<br>");

        // 내용을 담은 템플릿html 경로
        String templatePath = application.getRealPath("/EmailDesign/emailDesign.html");
        BufferedReader br = new BufferedReader(new FileReader(templatePath));

        String htmlContent = "";
        String line;
        while((line = br.readLine()) != null) {
            htmlContent += line;
        }
        br.close();
        htmlContent = htmlContent.replace("__CONTENT__", content);

        emailInfo.put("content", htmlContent); // 내용
    }

    try {
        NaverSMTP smtpServer = new NaverSMTP();    // 메일 전송 객체 생성
        smtpServer.emailSending(emailInfo);
        response.sendRedirect("emailForm.jsp?msg=success");
    } catch(Exception e) {
        response.sendRedirect("emailForm.jsp?msg=fail");
        e.printStackTrace();
    }
%>

```

응용 - 회원가입 시 가입 축하 메일 보내기

▼ market(DB)/user(table) 생성

```
CREATE TABLE `user` (
  `to` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `pass` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `name` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `content` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  PRIMARY KEY (`to`) USING BTREE
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
;
```

▼ userFrom.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    String msg = request.getParameter("msg");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>join.jsp</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQf"
</head>
<body>
<!-- nav bar -->
<nav class="navbar navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Test</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="na
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Login</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Logout</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Join</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown link
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">1</a></li>
            <li><a class="dropdown-item" href="#">2</a></li>
            <li><a class="dropdown-item" href="#">3</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>

<div class="container">
  <form action="userCheck.jsp" method="post">
    <div class="form-group">
      <label class="form-label">이메일</label>
      <input class="form-control" type="email" name="to">
    </div>
    <div class="form-group">
      <label class="form-label">비밀번호</label>
      <input class="form-control" type="password" name="pass">
    </div>
    <div class="form-group">
      <label class="form-label">이름</label>
      <input class="form-control" type="text" name="name">
    </div>
    <div class="form-group">
      <label class="form-label">내용</label>
      <input class="form-control" type="text" name="content">
    </div>
    <div class="form-group text-end">
      <p></p>
      <button class="btn btn-lg btn-dark">전송</button>
    </div>
  </form>
</div>
```

```

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="exampleModalLabel">
          <%= msg!=null && msg.equals("success") ? "회원가입 완료" : "회원가입 실패"%>
        </h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <%= msg!=null && msg.equals("success") ? "환영합니다." : "다시 시도하세요."%>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">닫기</button>
        <!-- <button type="button" class="btn btn-primary">Save changes</button> -->
      </div>
    </div>
  </div>
</div>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+800xUf+/Im1MZjXxxgOcBQBxU=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M" crossorigin="anonymous"></script>
<script>
$(function(){
  var msg = '<%= msg %>';
  console.log(msg);
  if(msg != 'null') {
    $('#exampleModal').modal('show');
  }
});
</script>
</body>
</html>

```

▼ userCheck.jsp

```

<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.Map"%>
<%@page import="email.NaverSMTP"%>
<%@page import="jdbc.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");

    String to = request.getParameter("to");
    String pass = request.getParameter("pass");
    String name = request.getParameter("name");
    String content = request.getParameter("content");
    boolean result = new UserDAO().join(to, pass, name, content);

    if(result) {
        Map<String, String> emailInfo = new HashMap<>();

        emailInfo.put("format", "text/html;charset=UTF-8"); // 메일의 형식을 설정

        // 내용을 담은 템플릿html 경로
        String templatePath = application.getRealPath("welcome.html");
        BufferedReader br = new BufferedReader(new FileReader(templatePath));

        String htmlContent = "";
        String line;
        while((line = br.readLine()) != null) {
            htmlContent += line;
        }
        br.close();
        htmlContent = htmlContent.replace("__CONTENT__", content);
        htmlContent = htmlContent.replace("__NAME__", name);
        htmlContent = htmlContent.replace("__EMAIL__", to);
        htmlContent = htmlContent.replace("__PASSWORD__", pass);

        emailInfo.put("content", htmlContent); // 내용
        emailInfo.put("from", "ro1864@naver.com"); // 보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
        emailInfo.put("to", to); // 받는 사람
        emailInfo.put("subject", "가입을 축하합니다."); // 제목

        try {
            NaverSMTP smtpServer = new NaverSMTP();
            smtpServer.emailSending(emailInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

```

```

        response.sendRedirect("userForm.jsp?msg=success");
    } else {
        response.sendRedirect("userForm.jsp?msg=fail");
    }
}
%>

```

▼ NaverSMTP.java

```

package email;

import java.util.Map;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class NaverSMTP {
    private final Properties serverInfo; // 서버 정보
    private final Authenticator auth; // 인증 정보

    public NaverSMTP() {
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.naver.com"); //메일 서버 지정
        serverInfo.put("mail.smtp.port", "465"); // 포트 번호 거의 대부분 통일
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class",
            "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2"); //맥에서 반드시

        auth = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("아이디", "비밀번호");
            }
        };
    }

    public void emailSending(Map<String, String> mailInfo) throws MessagingException {
        // 1. 세션 생성 (네이버/다음/구글 메일 서버에 사용자 인증)
        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);
        // 2. 메시지 작성
        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from")));
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to")));
        msg.setSubject(mailInfo.get("subject"));
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));
        // 3. 전송
        Transport.send(msg);
    }
}

```

구글 이메일 (지메일)

지메일 앱 비밀번호 발급

구글 계정 관리 → 보안 → 앱 비밀번호 → 메일 / 윈도우 컴퓨터 → 생성
 → GmailSMTP.java의 계정 비밀번호로 입력

응용 - 회원가입 시 가입 축하 메일 보내기

▼ market(DB)/user(table) 생성

```
CREATE TABLE `user` (  
  `to` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `pass` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `name` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `content` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  PRIMARY KEY (`to`) USING BTREE  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
;
```

▼ userFrom.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>  
  
<%  
    String msg = request.getParameter("msg");  
%>  
  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>join.jsp</title>  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQh" data-bbox="151 430 886 445" type="text/css">  
</head>  
<body>  
    <!-- nav bar -->  
    <nav class="navbar navbar-dark bg-dark">  
        <div class="container-fluid">  
            <a class="navbar-brand" href="#">Test</a>  
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" data-bbox="151 505 886 515">  
                <span class="navbar-toggler-icon"></span>  
            </button>  
            <div class="collapse navbar-collapse" id="navbarNavDropdown">  
                <ul class="navbar-nav">  
                    <li class="nav-item">  
                        <a class="nav-link active" aria-current="page" href="#">Login</a>  
                    </li>  
                    <li class="nav-item">  
                        <a class="nav-link" href="#">Logout</a>  
                    </li>  
                    <li class="nav-item">  
                        <a class="nav-link" href="#">Join</a>  
                    </li>  
                    <li class="nav-item dropdown">  
                        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">  
                            Dropdown link  
                        </a>  
                        <ul class="dropdown-menu">  
                            <li><a class="dropdown-item" href="#">1</a></li>  
                            <li><a class="dropdown-item" href="#">2</a></li>  
                            <li><a class="dropdown-item" href="#">3</a></li>  
                        </ul>  
                    </li>  
                </ul>  
            </div>  
        </div>  
    </nav>  
  
    <div class="container">  
        <form action="userCheck.jsp" method="post">  
            <div class="form-group">  
                <label class="form-label">이메일</label>  
                <input class="form-control" type="email" name="to">  
            </div>  
            <div class="form-group">  
                <label class="form-label">비밀번호</label>  
                <input class="form-control" type="password" name="pass">  
            </div>  
            <div class="form-group">  
                <label class="form-label">이름</label>  
                <input class="form-control" type="text" name="name">  
            </div>  
        </form>  
    </div>  
</body>  
</html>
```

```

        <div class="form-group">
            <label class="form-label">내용</label>
            <input class="form-control" type="text" name="content">
        </div>
        <div class="form-group text-end">
            <p></p>
            <button class="btn btn-lg btn-dark">전송</button>
        </div>
    </form>
</div>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="exampleModalLabel">
                    <%= msg!=null && msg.equals("success") ? "회원가입 완료" : "회원가입 실패"%>
                </h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <%= msg!=null && msg.equals("success") ? "환영합니다." : "다시 시도하세요."%>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">닫기</button>
                <!-- <button type="button" class="btn btn-primary">Save changes</button> -->
            </div>
        </div>
    </div>
</div>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxg0cBQBxU=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1K" crossorigin="anonymous"></script>
$(function(){
    var msg = '<%= msg %>';
    console.log(msg);
    if(msg != 'null') {
        $('#exampleModal').modal('show');
    }
});
</script>
</body>
</html>

```

▼ userCheck.jsp

```

<%@page import="email.GmailSMTP"%>
<%@page import="java.io.FileReader"%>
<%@page import="java.io.BufferedReader"%>
<%@page import="java.util.HashMap"%>
<%@page import="java.util.Map"%>
<%@page import="email.NaverSMTP"%>
<%@page import="jdbc.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("utf-8");

    String to = request.getParameter("to");
    String pass = request.getParameter("pass");
    String name = request.getParameter("name");
    String content = request.getParameter("content");
    boolean result = new UserDAO().join(to, pass, name, content);

    if(result) {
        Map<String, String> emailInfo = new HashMap<>();

        emailInfo.put("format", "text/html;charset=UTF-8");    // 메일의 형식을 설정

        // 내용을 담을 템플릿html 경로
        String templatePath = application.getRealPath("welcome.html");
        BufferedReader br = new BufferedReader(new FileReader(templatePath));

        String htmlContent = "";
        String line;
        while((line = br.readLine()) != null) {
            htmlContent += line;
        }
        br.close();
        htmlContent = htmlContent.replace("__CONTENT__", content);
        htmlContent = htmlContent.replace("__NAME__", name);
        htmlContent = htmlContent.replace("__EMAIL__", to);
        htmlContent = htmlContent.replace("__PASSWORD__", pass);
    }
}

```

```

        emailInfo.put("content", htmlContent); // 내용
        emailInfo.put("from", "ro7330@gmail.com"); // 보내는 사람, 변동 불가, 반드시 본인의 네이버 메일 주소
        emailInfo.put("to", to); // 받는 사람
        emailInfo.put("subject", "[구글] 가입을 축하합니다."); // 제목

        try {
            GmailSMTP smtpServer = new GmailSMTP();
            smtpServer.emailSending(emailInfo);
        } catch (Exception e) {
            e.printStackTrace();
        }
        response.sendRedirect("userForm.jsp?msg=success");
    } else {
        response.sendRedirect("userForm.jsp?msg=fail");
    }
}
%>

```

▼ GmailSMTP.java

```

package email;

import java.util.Map;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class GmailSMTP {
    private final Properties serverInfo; // 서버 정보
    private final Authenticator auth; // 인증 정보

    public GmailSMTP() {
        serverInfo = new Properties();
        serverInfo.put("mail.smtp.host", "smtp.gmail.com"); // 메일 서버 지정
        serverInfo.put("mail.smtp.port", "465"); // 포트 번호 거의 대부분 동일
        serverInfo.put("mail.smtp.starttls.enable", "true");
        serverInfo.put("mail.smtp.ssl.enable", "true");
        serverInfo.put("mail.smtp.ssl.trust", "smtp.gmail.com");
        serverInfo.put("mail.smtp.auth", "true");
        serverInfo.put("mail.smtp.debug", "true");
        serverInfo.put("mail.smtp.socketFactory.port", "465");
        serverInfo.put("mail.smtp.socketFactory.class",
            "javax.net.ssl.SSLSocketFactory");
        serverInfo.put("mail.smtp.socketFactory.fallback", "false");

        serverInfo.put("mail.smtp.ssl.protocols", "TLSv1.2"); // 맥에서 반드시

        auth = new Authenticator() {
            @Override
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("아이디", "비밀번호");
            }
        };
    }

    public void emailSending(Map<String, String> mailInfo) throws MessagingException {
        // 1. 세션 생성 (네이버/다음/구글 메일 서버에 사용자 인증)
        Session session = Session.getInstance(serverInfo, auth);
        session.setDebug(true);
        // 2. 메시지 작성
        MimeMessage msg = new MimeMessage(session);
        msg.setFrom(new InternetAddress(mailInfo.get("from")));
        msg.addRecipient(Message.RecipientType.TO, new InternetAddress(mailInfo.get("to")));
        msg.setSubject(mailInfo.get("subject"));
        msg.setContent(mailInfo.get("content"), mailInfo.get("format"));
        // 3. 전송
        Transport.send(msg);
    }
}

```