**보고서**



|  |  |
| --- | --- |
| **제출일자** | 2022년 04월 12일 (화) |
| **교과목** | 클라우드시스템 |
| **담당교수** | 박유현 교수님 |
| **학과** | 컴퓨터소프트웨어공학과 |
| **이름(학번)** | 박상현 (20183145) |
| **학년** | 3학년 |

**[로그인 구현하기]**

|  |
| --- |
| **Index.html** |
| <html>  **<title>Login\_20183145 박상현</title>**  <head>  **<meta charset="utf-8">**  </head>  <body>  **<form action="login.jsp"method="post">**  <p><h2>Login</h2></p>  <p>  ID **<input type="text"name="id"placeholder="Input ID"required>**  </p>  <p>  P/W **<input type="password"name="pw"placeholder="Input P/W">**  </p>  <p>Master ID : park, P/W : 20183145</p>  **<input type="submit"value="Enter">**  </form>  </body>  </html> |

|  |
| --- |
| **login.jsp** |
| <%  request.setCharacterEncoding("utf-8");  **String id = request.getParameter("id");**  String enterId = "park";  String pw = request.getParameter("pw");  String enterPw = "20183145";  boolean Enter = false;  %>  <%  if (id.equals(enterId)&&pw.equals(enterPw)){  Enter = true;  **out.println("Login Succese!");**  out.println("ID : " + id);  out.println("PW : " + pw);  }  else{  Enter = false;  out.println("Login fail");  }  %> |

|  |
| --- |
| **HTML Login 페이지** |
| 텍스트이(가) 표시된 사진  자동 생성된 설명 |

|  |
| --- |
| **Login에 성공했을 때** |
|  |

|  |
| --- |
| **Login에 실패했을 때** |
|  |

|  |
| --- |
| **HTML** |

* **<title>** : 웹 페이지 이름 지정
* **<meta>** : HTML에 대한 여러 정보를 정의한다
  + **charset="utf-8"** : HTML의 문자 인코딩 방식을 utf-8로 지정한다
* **<form action="login.jsp" method="post">**

post 방식으로 login.jsp에 URL을 전달한다

* **<input type="text" name="id" placeholder="Input ID" required>**

text 방식으로, submit에 필수 입력을 요구한다

* **<input type="submit" value="Enter">**

해당 submit를 클릭하면 login.jsp에 넘어간다

|  |
| --- |
| **JSP** |

* **<% ... %>** : Java 코드 입력 구간
* **<%= ... %>** : Java 변수 출력
* **Void request.setCharacterEncoding("utf-8")**

문자 인코딩을 utf-8로 지정한다

* **String request.getParameter(VALUE)**

HTML에서 받아온 파라미터인 VALUE 값을 문자열로 반환한다

* **out.println(STR)** : STR을 화면에 출력한다

**[Google Login API 구현하기]**

|  |
| --- |
| **Index.html** |
| <html>  <title>Login\_20183145 박상현</title>  <head>  <meta charset="utf-8">  **<meta name ="google-signin-client\_id" content="YOUR\_CLIENT\_ID.apps.googleusercontent.com">**  **<script src="https://apis.google.com/js/platform.js" async defer></script>**  </head>  <body>  <form action="login.jsp" method="post">  <p><h2>Login</h2></p>  <p>  ID <input type="text" name="id" placeholder="Input ID" required>  </p>  <p>  P/W <input type="password" name="pw" placeholder="Input P/W">  </p>  <p>Master ID : park, P/W : 20183145</p>  **<div class="g-signin2" data-onsuccess="onSignIn" data-theme="dark"></div>**  **<script>**  function onSignIn(googleUser) {  // Useful data for your client-side scripts:  var profile = googleUser.getBasicProfile();  console.log("ID: " + profile.getId());  console.log('Full Name: ' + profile.getName());  console.log('Given Name: ' + profile.getGivenName());  console.log('Family Name: ' + profile.getFamilyName());  console.log("Image URL: " + profile.getImageUrl());  console.log("Email: " + profile.getEmail());    // The ID token you need to pass to your backend:  var id\_token = googleUser.getAuthResponse().id\_token;  console.log("ID Token: " + id\_token);  }  **</script>**  <input type="submit" value="Enter">  </form>  </body>  </html> |

|  |
| --- |
| **HTML Login 페이지** |
|  |

* **<meta name ="google-signin-client\_id" content="YOUR\_CLIENT\_ID.apps.googleusercontent.com">**

content에 부여받은 Client ID API를 넣는다

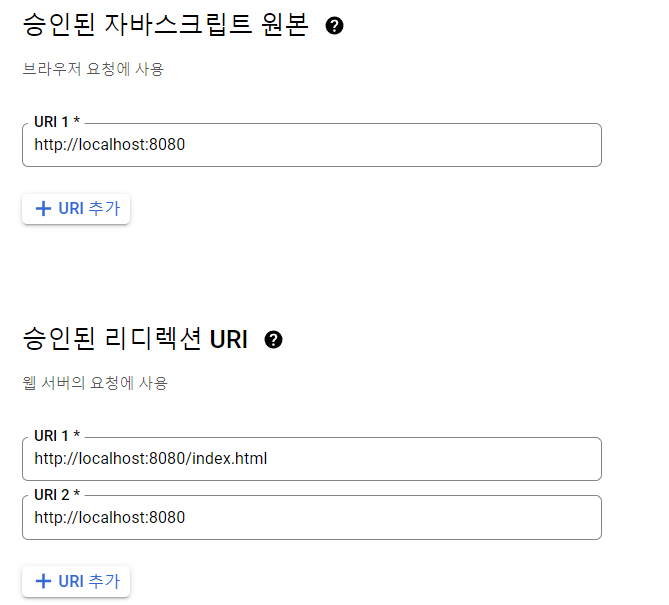
* + **Google Login API 사용**

1. 사용자 인증 정보를 통해 사용자 인증정보를 만든다



1. OAuth 2.0 클라이언트 ID에서 브라우저 및 웹 서버의 요청에 사용할 URL을 입력한다

//공인 IP자리에 localhost로 대체한다



1. <https://developers.google.com/identity/sign-in/web/>에 접속하여 Google에서 제공하는 코드를 html파일에 추가한다