<오픈소스SW활용 보고서>



**조원: 32177384 안상현, 32184787 표세훈**

**과목: 오픈소스SW활용**

**교수: 송 인 식 교수님**

* **프로젝트 배경**

최근 사이트에서는 본인 인증을 하는 사이트들이 대부분이다. 이번 프로젝트로는 회원가입을 통해 자신의 인적사항을 입력하여 본인의 학과, 학번, 아이디 및 비밀번호를 입력해 로그인을 하면 입력한 인적사항이 나오는 프로젝트를 제작했다.

* **시스템 구조 및 설계 내용**

서버는 장고를 활용해 개발, 이를 리눅스(Ubuntu)를 이용해 웹서버를 실행시켰다. 장고 내부에서 홈페이지는 Html, CSS, JS를 이용하여 로그인, 회원가입, 관리자 로그인 및 관리 페이지 그리고 결과가 뜨는 Info페이지를 제작, 홈페이지 사이 연결과 각종 논리 처리는 파이썬, 데이터 처리는 장고에서 기본 지원하는 SQLite를 이용해 구축하였다.

* **Front End**

먼저 로그인 페이지에서는 아이디와 비밀번호를 입력해야 로그인이 가능해진다. 회원가입에서 입력한 자신의 아이디와 비밀번호를 기입해야 한다. 만약 아이디 혹은 비밀번호를 잘못 입력했을 경우 서버측으로부터 error\_message가 날라오는데 이를 alert를 이용해 다시 입력하라는 메시지를 띄우게 했다.

회원가입에서는 자신의 이름, 아이디, 비밀번호 입력 및 확인, 학과와 학번을 기입할 수 있고 Checkpoint를 통해 자신이 학생, 관리가 그리고 교수인지를 체크할 수 있게 만들었다. 기입을 하지 않은 항목이 있을 경우 재입력을 알리는 메시지를 띄우고, 만약 이미 존재하는 ID가 있다면 중복된 ID가 존재한다는 메시지를 띄운다. 모든 항목의 기입, 2번 입력 비밀번호 모두 동일, ID의 중복이 없을 경우 서버로부터 0번 message를 받고 이를 이용해 회원가입이 완료됐다는 메시지를 띄우고 로그인 페이지로 이동하게 구축했다.

관리자 로그인 및 관리 페이지는 먼저 관리자의 아이디와 비밀번호를 입력해야 한다. 여기서도 앞의 로그인 페이지와 마찬가지로 아이디와 비밀번호를 잘못 입력했을 경우 메시지가 뜨게 했고 관리자 로그인 후 회원가입이 되어있는 사람들의 비밀번호를 제외한 모든 정보가 나와있다. 메시지 버튼을 클릭하면 회원가입을 요청한 사람의 정보가 뜨고 체크를 한 후 저장버튼을 누르면 회원가입이 승인이 되고 로그인이 가능해진다. 삭제 버튼을 클릭하면 회원가입&관리자 허가를 받은 사람들의 정보가 뜨고 체크를 한 후 저장버튼을 누르면 계정이 삭제된다. 즉 관리자가 직접 계정을 삭제할 수 있는 권한을 가지고 있다.

Info페이지에서는 회원가입에서 기입한 자신의 인적사항이 회원가입 승인과 로그인을 통해 데이터를 전송하여 띄우는 방식으로 개발을 했다.

* **Back End**

사용자가 홈페이지 주소를 입력에 서버에 접근하게 된다. 서버는 TCP 80 (http) 포트 로 접근을 포트포워딩을 통해 8000 포트로 넘겨준다. 이 8000포트는 Django 서버가 돌아가고 있는 포트이다. 요청이 Django 서버에 접근하게 되면 mysite/urls.py 에서 path(‘’, include(‘homepage.urls’))에 의해 homepage/url.py로 접근하게 된다. 여기서 path(‘’, views.login, name=’login’)에 의해 views.py 의 def login이 호출되게 되고 접근의 경우 request.method 가 ‘GET’이므로 해당하는 사이트인 homepage/Login.html을 사용자에게 보여주게 된다.

Login 사이트에서 사용자가 로그인을 시도하면 POST를 통해 서버 측으로 입력한 ID와 Password가 넘어오게 된다. 서버는 입력한 ID, Password, 관리자 허가를 체크한 후 앞의 조건 3개를 모두 만족하는 query data가 존재하면 그 데이터를 이용해 사용자에게 Information.html을 보여주게 된다. 만약 앞의 3개를 만족하는 query data가 없으면 다시 login.html을 보여주게 되는데 이 때 error message를 같이 보내게 된다.

Login 사이트에서 회원가입을 선택하면 homepage/url.py의 path(‘SignUp/’ views.signup, name=”signup”) 이 호출된다. 이는 views.py 의 def signup이 호출되게 되고 ‘GET’이므로 Signup.html을 사용자에게 보여준다. 사용자가 일부 항목을 빼고 회원가입을 신청하면 if문에 의해 걸리게 된다. 다른 경우는 password가 서로 다를 경우 if문에 걸린다. 또한 사용자가 중복된 ID를 이용해 회원가입을 하면 이 또한 if문에 걸리고 이 후 Signup.html과 data, 각 case에 맞는 message를 보내게 된다. 이 모든 조건을 피해 if문에 안 걸리면 Query에 data를 입력 이 후 Signup.html과 data, 0번 message를 보내게 된다.

Login 사이트에서 관리자 로그인을 선택하게 되면 homepage/url.py의 path(‘AuthorLogin/’ views.authorlogin) 이 호출된다. 이는 views.py 의 def authorlogin이 호출되게 되고 ‘GET’이므로 AuthorLogin.html을 사용자에게 보여준다. 로그인 방식은 Login 사이트에서 실시한 것과 동일하나 조건이 하나 더 붙어 관리자 계정인지를 추가로 확인하게 된다. 이 4개의 조건이 일치하면 author 홈페이지를 호출하게 된다. 그렇지 않으면 error message를 호출한다.

def author가 호출되면 관리자 허가된 계정들의 data를 html측으로 보내게 된다. 만약 관리자가 조작을 통해 def message를 호출하면 관리자 허가가 안 된 계정들의 data와 Message.html사이트를 호출하게 된다. 여기서 체크된 항목이 있을 경우 서버에서는 각 항목의 query data의 check\_author 항목을 1로 수정 후 저장한다. 반대로 Author.html 에서 조작을 통해 def delete를 호출하면 관리자 허가된 계정들의 data를 Delete.html 사이트로 호출하게 된다. 여기서 체크된 항목이 있을 경우 서버에서는 각 항목의 query data를 삭제하는 과정을 거치게 된다.

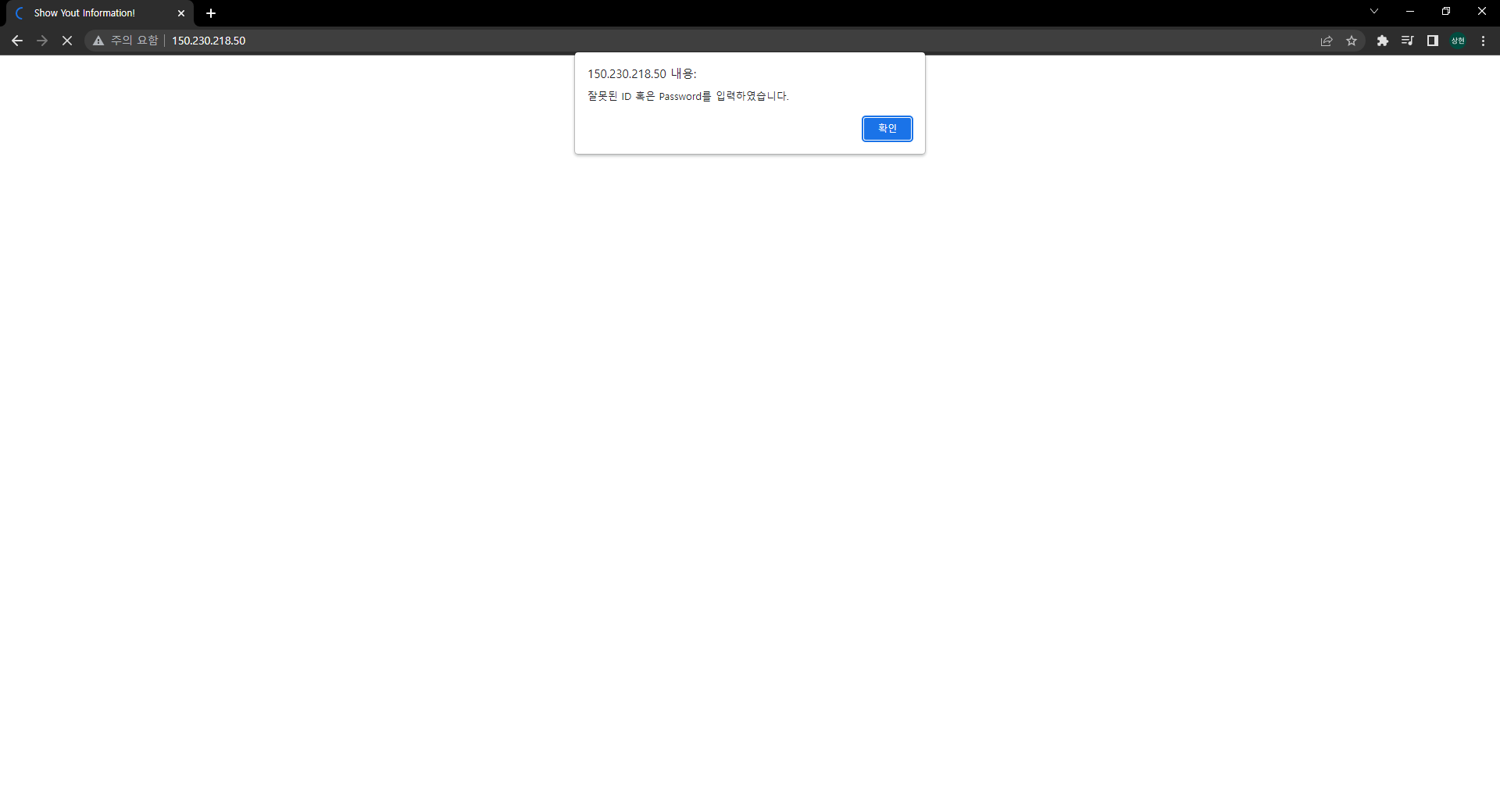
* **테스트 결과**

서버 연결 후 메인 페이지이다. 로그인 화면으로 아이디와 비밀번호를 입력하고 필요시 회원가입 페이지로 이동할 수 있는 버튼과 관리자들만 로그인 할 수 있는 관리자 로그인 버튼이 있다. 클릭 시 각 페이지로 이동할 수 있다. 잘못 입력했을 경우 다음과 같은 오류 메시지를 뜨게 했다.

<로그인 페이지>



<로그인 alert 메시지>



다음은 회원가입 페이지이다. 위의 설명처럼 이름, 아이디, 비밀번호 입력 및 확인, 학과, 학번을 입력하고 직책을 선택할 수 있다. 입력사항이 빠졌을 경우 오류 메시지를 뜨도록 설정했다.

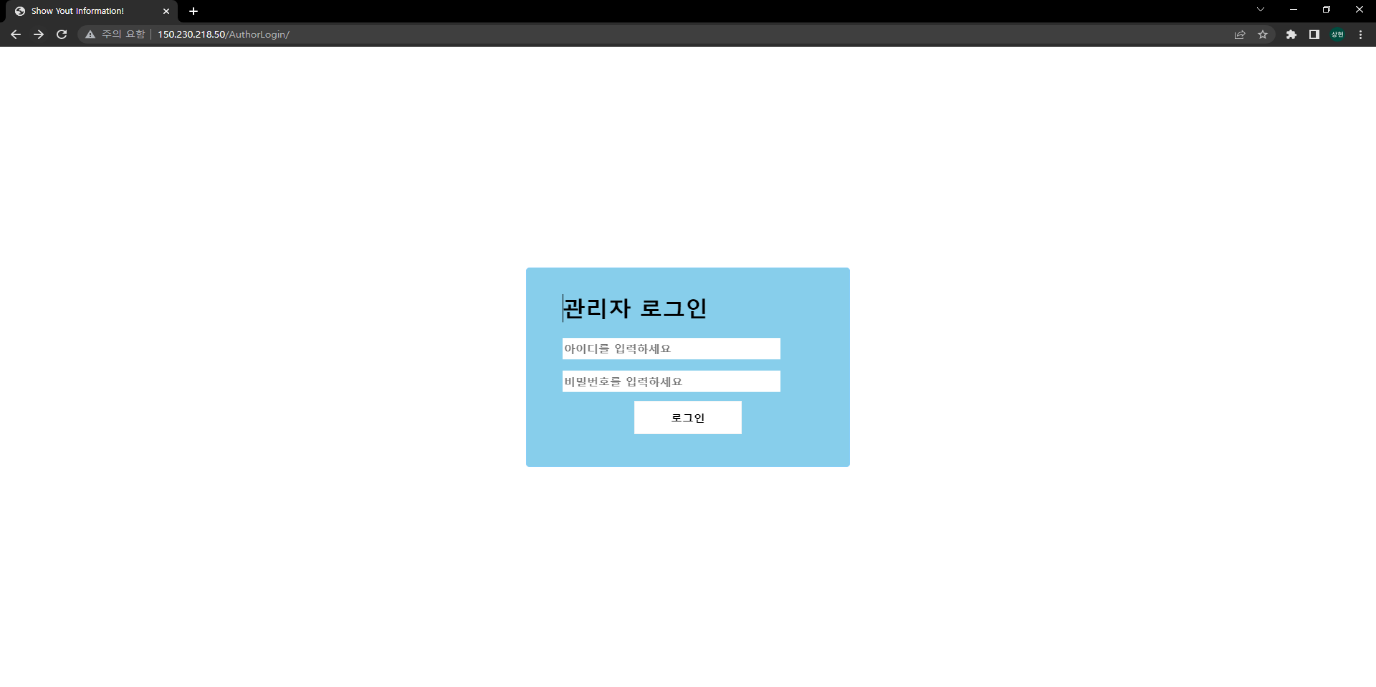
회원가입에서 입력한 인적사항들은 나중에 로그인하고 Info페이지에서 볼 수 있다.

<회원가입 페이지>

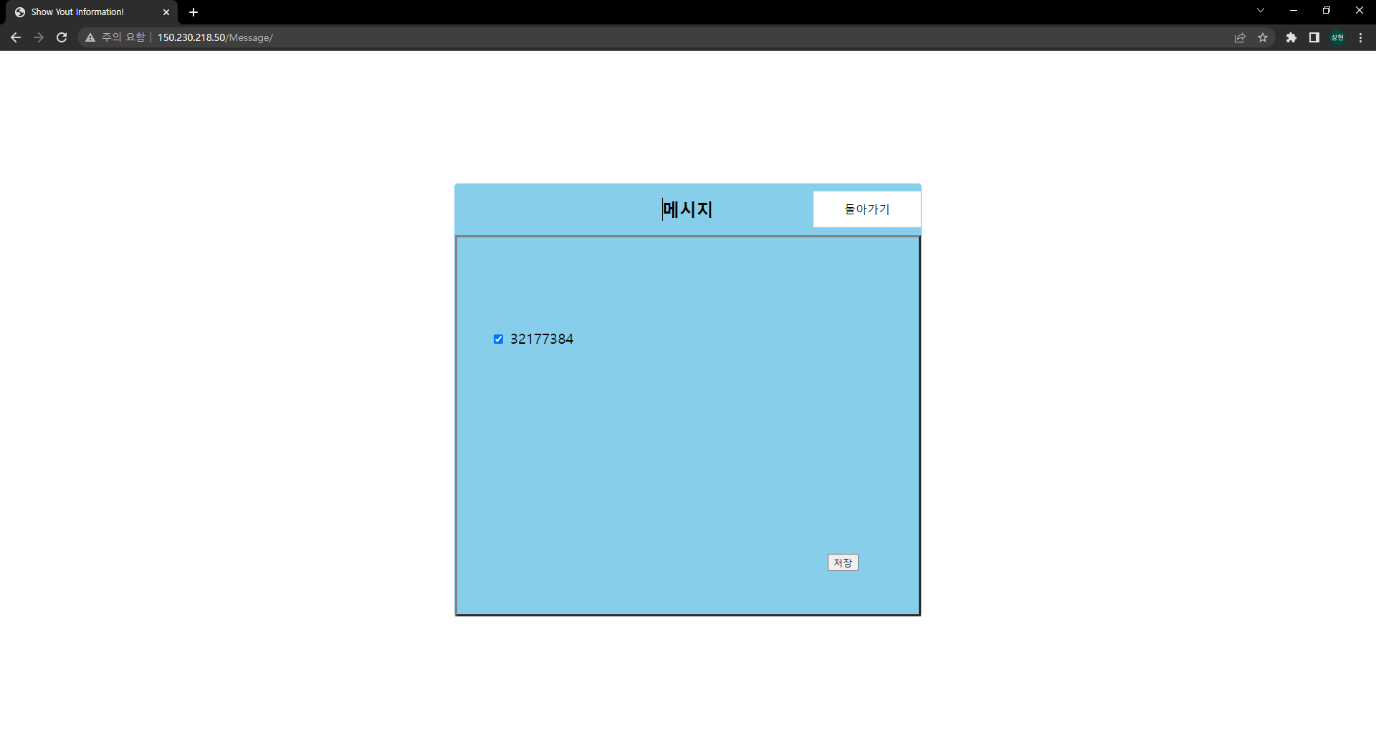


다음은 관리자 로그인 및 관리 페이지이다. 관리자 아이디와 비밀번호로 로그인을 할 수 있고 앞의 로그인 화면처럼 재입력 메시지도 뜬다. 로그인을 하면 회원가입 승인이 된 사람들의 정보가 뜬다. 오른쪽 위에 메시지 버튼을 클릭하면 회원가입을 요청한 사람의 정보가 뜨고 왼쪽의 checkbox를 체크한 후 저장을 하면 승인이 되면서 정보가 추가가 된 것을 볼 수 있다.

<관리자 로그인 페이지>



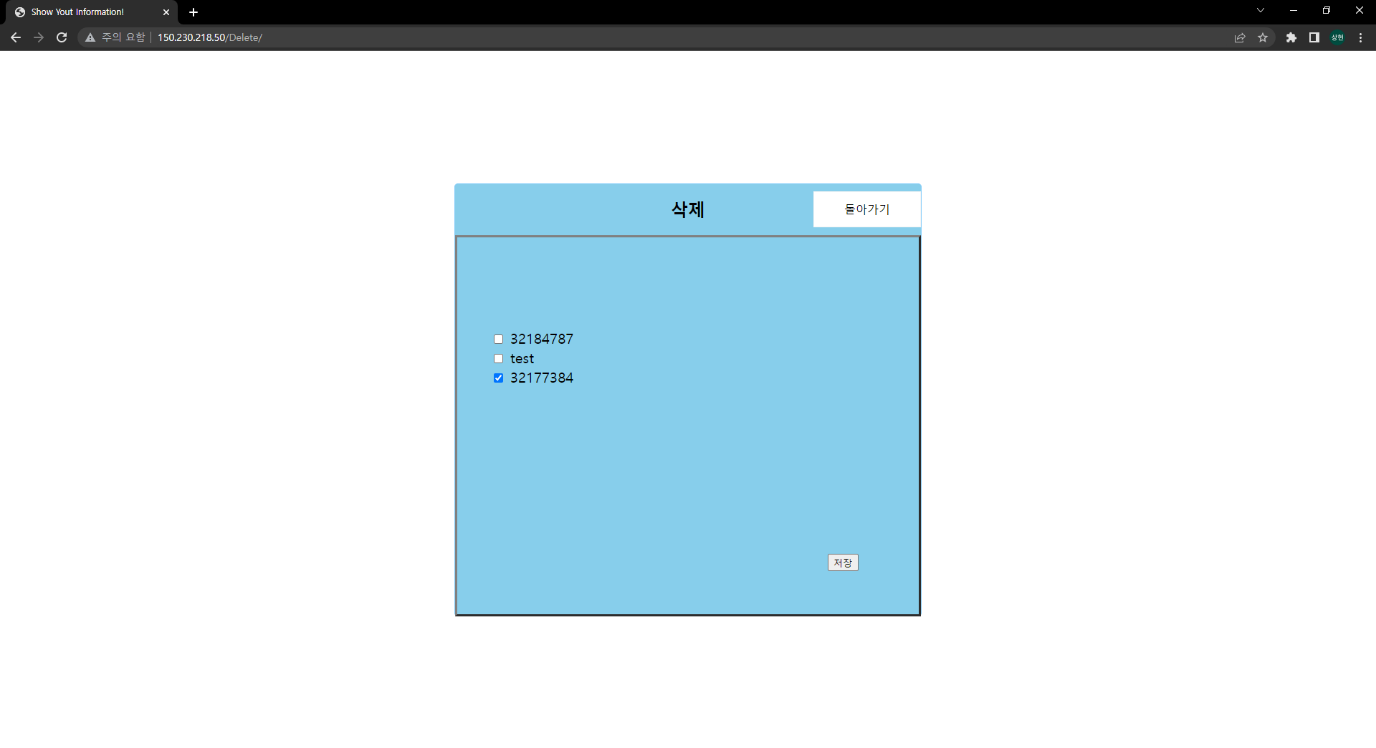
<관리자 페이지>



<승인 확인>



<회원 삭제>



<삭제 확인>



마지막으로 로그인 후 Info페이지이다. 회원가입 승인이 된 아이디와 비밀번호로 로그인을 하면 회원가입을 할 때 입력한 인적사항이 뜨는 것을 볼 수 있다.

<Info 페이지 사진>



* **역할 분담 및 개발 과정**
* **역할 분담**  
  32184787 표세훈(팀장): 리눅스(Ubuntu)를 이용한 실제 서버 구동, Django 서버 구축, html에서 들어오는 data 처리, html 간 연결 구축, 홈페이지 제작에 있어 서버와 연관성을 높이기 위한 일부 코드 수정

32177384 안상현(팀원): 홈페이지(html, css) 제작, 서버 테스트 및 코드 검토

* **과정 및 일자**

5/10 ~ 5/15: 온라인으로 프로젝트 계획 및 회의 진행. 각자 역할 분담과 의견 조합.

5/16: 오프라인으로 회의 진행. 구체적으로 서버 연결 및 홈페이지에 필요한 UI 논의.

5/17 ~ 5/23: Html을 이용해 홈페이지 제작.

5/31 ~ 6/7: 홈페이지와 서버 연결. 회의를 통해 수정사항 개선.

6/10 ~ 6/11: 온라인으로 회의 진행. PPT 제작 및 최종 보고서 작성.

6/12: 기능 추가 구현, PPT, 최종 보고서 보안

6/13: 오프라인으로 회의 진행. 보고서 제출용 동영상 촬영 및 수정사항 개선.

* **소스코드 링크**

표세훈: <https://github.com/kimpyo9357/Profile-Database>

안상현: <https://github.com/tedan96/Profile-Database>

* **발표 영상 링크**

오픈소스 1분반 프로젝트 발표 Team P.H.

<https://youtu.be/ibgHSKXlP0A>