**인터넷프로그래밍 프로젝트 결과보고서**

|  |  |  |  |
| --- | --- | --- | --- |
| **기본 사항** | | | |
| **교과목명** | 인터넷 프로그래밍 | **담당교수** | 박유현 교수님 |
| **프로젝트명** | The Interior | | |
| **팀구성원**  **(학번/성명)** | 20153292 정현진  20163364 이자원 | | |
| **제출일자** | 2018-06-11 | | |

**목차**

[**1. 프로젝트의 필요성** 2](#_Toc516459619)

[**2. 프로젝트 내용** 3](#_Toc516459620)

[**3. 프로젝트 코드** 4](#_Toc516459621)

[**4. 프로젝트 결과** 36](#_Toc516459622)

[**5. 일정 대비 진행결과 분석** 48](#_Toc516459623)

[**6. 팀원 역할** 50](#_Toc516459624)

[**7. 총평** 51](#_Toc516459625)

# **1. 프로젝트의 필요성**

초기 생각했던 프로젝트와 동일하게 방 꾸미기와 인테리어 정보공유를 목적으로 한 사이트인 **‘The Interior’** 를 개발하였다. **‘The Interior’** 는 방 꾸미기와 인테리어에 관심이 있는 사람들과 관심은 있으나 어떤 방식으로 방을 꾸며야 할 지 모르는, 그리고 어려워하는 사람들을 위해 본 프로젝트가 꼭 필요하다 생각하고 개발을 진행하였다. ‘The Interior’ 를 이용하는 사람들이 자기들만의 다양한 인테리어 정보를 공유하고 또, 아이디어를 얻어갈 수 있도록 하는 것이 본 프로젝트의 목적이다.

본 프로젝트는 패션, 뷰티 정보를 공유하는 어플인 **‘스타일 쉐어’** 를 사용하다 영감을 얻었다. **‘스타일 쉐어’** 는 패션, 뷰티 정보공유 어플인 반면, 그 외에 필요 없는 정보나 돈을 받고 홍보하면서 마치 소비자인양 거짓된 정보를 제공한다. 이러한 정보를 제공하게 되면 패션, 뷰티에 대한 정보를 알고 싶어 하는 사람들은 불필요한 정보들을 제공받게 되어, 어플의 필요성을 느끼지 못하게 되어 사용을 하지 않게 된다. 이러한 문제점을 파악하여 **‘The Interior’** 에서는 오로지 사용자들이 필요로 하는 인테리어 정보만을 공유할 수 있도록 하였다.

1인 가구가 증가하는 시대인 만큼 오로지 자신만의 공간을 만들고, 그 공간을 꾸미고 싶어하는 사람들이 늘어나고 있다. 꼭 1인 가구 뿐만 아니라 트렌디한 젊은 부부들, 막 자기 방을 갖기 시작하는 사춘기 청소년들과 같이 자신만의 공간을 꾸미고 SNS에 자랑하고 싶어하는 사람들도 많다. **‘The Interior’** 는 자신만의 공간을 꾸미고 싶어하는 사람들을 위해 만들었다. 충분히 활성화만 잘 된다면 패션, 뷰티 정보공유 어플인 **‘스타일 쉐어’** 처럼 많은 사람들에게 영향을 줄 수 있을 거라고 자부한다.

# **2. 프로젝트 내용**

|  |  |
| --- | --- |
|  |  |
| **게시판 글쓰기 순서도** | **게시판 글 수정 순서도** |
|  |  |
| **게시판 글 삭제 순서도** | **게시판 댓글 작성 순서도** |

# **3. 프로젝트 코드**

1) cover.html

|  |
| --- |
| (1) body     * 100번 : div로 구역을 지정하고 배경사진을 집어넣는다. 클래스 이름을 img로 지정한다. * 101번 : 배경사진 위에 넣을 글과 start버튼을 넣는다. 이 전체를 content라는 클래스 이름을 가진 div로 묶는다. * 105번 : 화면 오른쪽 상단위에 나타낼 글을 content2라는 클래스 이름으로 따로 구역을 지정한다. * 108번 : start버튼을 누르면 Project.jsp로 페이지를 이동시키기 위해 a태그를 사용한다. 그리고 이 버튼의 이벤트를 수행하기 위해 클래스 이름을 button-1, eff-1로 지정한다. * 113번 : 배경사진 위에 나타나는 투명한 검은색 도형을 넣기 위해 img-cover라는 클래스 이름으로 div구역을 지정한다.   (2) css     * 12번 : body 전체 태그를 정의한다.      * 18번 : 배경사진의 속성을 정의한다. * 27번 : 배경사진 위에 덮을 불투명한 검정색 도형을 만들어 준다.      * 34번 : 배경사진 위에 올라가는 글의 속성을 정의한다. * 46번 : 화면 오른쪽 상단에 적혀지는 글의 속성을 정의한다.      * 56번 : start 겉에 둘러 싸여지는 도형을 만들어준다. * 70번 : start글자의 속성을 정의해 준다.      * 80번 : start부분에 마우스를 올렸을 때 start를 둘러싸고 있는 도형의 색을 바꿔준다. * 90번 : start부분에 마우스를 올렸을 때 eftt-1를 수행한다. * 93번 : start부분에 마우스를 올렸을 때 글자를 하얀색으로 바꿔준다. |

2) Project.jsp

|  |
| --- |
| (1)header   * 36번 ~ 59번 : 홈페이지 왼쪽 위의 상단 바 부분 * 39번 ~ 44번 : 상단 바 메뉴를 클릭했을 시, 나타나는 각자 게시판 부분들 * 46번 ~ 58번 : 홈페이지 오른쪽 위의 로그인 메뉴   IF문으로 getAttribute의 id값이 null이면 Login 버튼을 출력하고, 버튼 클릭 시 Login.html을 출력 해 로그인을 진행할 수 있게 한다.  로그인에 성공해 getAttribute의 id 값이 null이 아니면 Logout 버튼을 출력하고, 버튼 클릭 시 Logout.jsp를 출력해 로그아웃을 진행할 수 있게 한다.  (2) main (Carousel)     * 62번 ~ 94번 : Main의 Carousel(부트스트랩 사용) 부분 * 64번 : Div 태그를 통해 공간을 나누고, data-ride를 통해 부트스트랩의 캐러셀을 받아온다. * 66번 ~ 70번 : 캐러셀의 지표를 설정해준다. Data-target 태그는 모달로 띄우고자 하는 id 값에 #을 붙여서 연결해주는 것을 의미한다. * 72번 ~ 84번 : 캐러셀의 각 슬라이드마다 들어갈 이미지를 설정해준다.   Div 태그 내에서 Img-responsive center-block으로 이미지가 화면의 중앙에 나타나게 설정해준다.  86번 ~ 93번 : 왼쪽, 오른쪽 화살표 아이콘을 넣어, 아이콘 클릭 시 사진이 이동하게 설정해준다.  (3)section     * 157번 ~ 192번 : mini-posts 구간으로 <section> 태그로 구조를 나눈다. * 159번 ~ 166번(189번) : <article> 태그로 첫번째 게시물 공간을 나눠주고, <header> 태그를 통해 첫번째 게시물의 헤더 부분에 제목과 작성날짜, 유저 이미지 사진을 출력해준다.   <header> 태그가 끝난 기점으로 인테리어 사진을 출력한다.  (4)script     * 268번 ~ 273번 : 사용되는 script들을 선언해준다. |

3) InteriorAccessories.jsp, furniture.jsp, interior.jsp, question.jsp

|  |
| --- |
| * 35번 ~ 55번 : 오라클 연결을 위한 try구문      * 91번 ~ 94번 : 위에서 rs에 저장한 레코드 값들을 while을 통해 꺼내 준다. * 93번 : 테이블을 제일 최신 글을 위해 값을 증가 시켜 값이 1이 되면 while을 끝낸다.      * 98번 ~ 120번 : 테이블에서 제일 최신 글을 작성해준다. * rs.getString(“테이블의 필드 이름”)을 이용해 필드에 들어있는 레코드 값을 꺼내 준다.      * 122번 ~ 132번 : 최신 글을 작성하면 try-catch 구문을 끝낸다. * 124번 : 최신 글이 표시가 되면 while을 벗어나고 다시 num값을 빼줘서 0으로 만들어 준다. 그로 인해 새로운 글이 작성이 되면 다시 최신 글을 업데이트 시켜준다.      * 135번 ~ 137번 : 글쓰기 버튼을 생성 해 준다. * 136번 : 글쓰기 버튼을 누르면 글을 쓸 수 있는 페이지로 넘어가게 하기 위해 a태그를 사용해 write.jsp를 href속성으로 연결 시켜준다.      * 177번 ~ 200번 : 선택한 메뉴에 해당하는 테이블에 가서 모든 레코드를 가져와 왼쪽 사이드 구역에 게시글 리스트를 만들어 준다. * 185번 : 게시글 리스트의 제목을 클릭하면 이 게시글을 자세히 볼 수 있도록 게시글 내용을 볼 수 있는 사이트 content.jsp로 연결시켜준다. 그리고 content.jsp에 선택한 게시글의 idx값을 넘겨준다. |

|  |
| --- |
| (2) content.jsp     * 40번 ~ 45번 : 게시글을 누를 때 마다 조회수를 1씩 늘려 주기 위한 구문이다. * 44번 : 넘겨받은 idx값에 해당하는 count레코드를 가져와 count변수에 넣어준다. * 45번 : count변수를 1씩 증가시켜준다.      * 79번 ~ 105번 : 게시글을 보여주는 구문이다. * 97번 : 게시글 삭제 버튼을 누르면 비밀번호를 적어서 확인하는 delete\_ok.jsp로 사이트를 이동시킨다. 이 때, 이 게시물의 idx값을 같이 그 사이트로 넘겨준다. * 99번 : 97번줄과 동일하게 게시글 수정을 누르면 게시글을 수정할 수 있는 modify\_write.jsp로 사이트를 이동시킨다. 이 때, 이 게시물의 idx값을 같이 넘겨준다. * 101번 : 게시글 목록으로의 버튼을 누르면 이 게시물이 들어있는 첫 게시판으로 넘어간다.      * 110번 ~ 122번 : 게시글의 조회수를 업데이트 하고 이 게시물을 다 표시하면 try-catch구문을 닫는다. * 110번 : 이 게시물의 idx값을 where로 조건을 걸어 이 게시물에 해당하는 레코드 값의 count 부분을 1 올려준 count값으로 갱신 시킨다.      * 124번 ~ 141번 : 게시물 밑에 댓글을 작성 할 수 있는 부분을 만들어 준다. * 125번 : 댓글을 작성하면 그 내용을 comment.jsp에 넘겨준다. 그리고 이 게시물에 해당하는 댓글임을 확인하기 위하여 이 게시물의 idx값도 함께 넘겨준다. * 124번 : session에 id값이 null이 아니면 즉, 로그인을 하면 이 부분을 나타내게 해주고 로그인을 한 상태가 아니면 이 댓글 작성 부분을 보여주지 않는다. * 132번 : 로그인을 해서 댓글을 쓸 때 작성자 부분에는 로그인을 한 회원의 이름을 기본값으로 준다.      * 163번 ~ 178번 : 그 게시물에 작성한 댓글을 보여준다. * 169번 : 댓글의 내용을 보여주기 위해 while문을 이용해서 이 게시물과 동일한 idx값을 가지는 댓글 테이블의 레코드 값들을 다 나타내어준다. |

|  |
| --- |
| (3) write.jsp     * 52번 ~ 56번 : 로그인을 했을 때 저장하는 id값이 null이면 즉, 로그인을 하지 않았을 경우 글을 쓸 수 없게 로그인 후 이용해 주세요 라는 알림창을 뜨게 하고 이전 페이지로 돌아가게 한다. 그리고 로그인을 했을 때 else구문에 있는 게시글 작성 소스를 보이게 한다.      * 87번 ~ 112번 : 글쓰기를 할 수 있는 표를 만들어 준다. * 92번 : 여기에 적은 값들을 insert.jsp에 넘어가게 해준다. 그리고 값들을 넘어가게 할 때 formCheck()함수를 실행하게 해서 false가 넘어가게 되면 게시글 작성을 할 수 없게 오류 검사를 하게 한다. 그리고 사진 파일을 넘어가게 하기위해 entype을 설정한다. * 99번 : 로그인을 할 때 게시글을 작성할 수 있기 때문에 작성자에는 로그인을 한 사람의 이름을 기본값으로 넣어준다. * 110번 : 비밀번호 부분에도 로그인을 한 회원의 비밀번호를 기본값으로 설정해준다.      * 114번 ~ 122번 : 게시글을 작성하는 부분 밑에 제출, reset, 글 목록으로 돌아가는 버튼을 생성해주고 table을 닫아준다. * 117번 : 글 목록으로…라는 버튼을 누르면 다시 이 게시물을 쓰던 메뉴의 첫 화면으로 돌아간다.      * 119번 ~ 123번 : 파일 입력을 하게 하기 위한 php소스를 적어주고 게시글 쓰는 부분을 마무리 시킨다.      * 12번 ~ 40번 : 게시물을 썼을 때 null값을 전송하는지 오류검사를 하는 자바 스크립트 함수이다. 제목과 작성자 내용부분이 null값이면 알림창을 뜨게 하고 false을 리턴해서 insert.jsp 파일로 넘어갈 수 없게 한다. |

|  |
| --- |
| (4) insert.jsp     * 22번 ~ 30번 : 파일 저장을 위한 변수를 선언해 준다. * 22번 : 게시글 쓰기에서 첨부한 파일을 따로 저장해 줄 파일 경로를 설정해준다.   여기서는 ROOT파일 안에 uploadFile이라는 파일을 따로 만들어 주어서 거기에 데이터베이스 테이블에 들어오는 사진을 저장한다.   * 26번 : 같은 이름의 사진이 들어갈 수 있으니 중복 처리할 이름 변수를 선언한다.      * 32번 ~ 56번 : write.jsp파일에서 작성한 게시글의 사진 파일을 변수 안에 저장하기 위한 try-catch구문이다. * 46번 : 파일명의 중복을 막기 위해서 파일 이름 뒤에 숫자를 붙여서 저장하는 fileSystemName을 통해 fileName1 변수에 중복처리 된 파일 이름을 저장한다.      * 58번 ~ 64번 : write.jsp에서 작성한 값들을 데이터 베이스 테이블에 저장하기 위해서 write.jsp에서 넘겨온 값들을 변수에 저장한다.      * 67번 ~ 71번 : write.jsp에서 넘겨져 온 내용 중에서 첨부파일을 넣지 않은 경우 fileName1에 null값이 들어온다. 첨부파일을 넣지 않았을 경우 게시물을 작성할 수 없게 하도록 하기 위해서 fileName1이 null일 때 알림창을 띄우게 해서 이전 페이지로 넘어가도록 한다. 그리고 첨부파일이 들어왔을 경우 else 안에 있는 테이블에 값을 넣기 위한 try-catch구문을 실행한다.      * 72번 ~ 97번 : write.jsp에서 넘겨준 게시물 작성 값을 그 게시물에 해당하는 메뉴의 테이블에 INSERT로 넣는다. * 95번 : write.jsp에서 넘겨준 값을 테이블에 넣는 걸 완료하면 그 게시글에 해당하는 메뉴 첫 화면으로 돌아간다. |

|  |
| --- |
| (5) delete\_ok.jsp     * 18번 ~ 21번 : delete\_ok.jsp 파일에서 idx변수를 선언해주고 이 변수에 content.jsp에서 넘겨준 삭제할 게시물의 idx값을 저장한다.      * 51번 ~ 70번 : 게시물을 삭제 할 때 작성자만 삭제 할 수 있도록 비밀번호를 확인하는 과정을 거친다. password라는 이름을 가진 text상자에 비밀번호를 입력하고 제출 버튼을 누르면 그 값이 post형식으로 delete.jsp파일에 넘어간다. 그리고 delete.jsp파일에 삭제할 게시물의 idx값도 같이 넘겨준다. |

|  |
| --- |
| (6) delete.jsp     * 15번 ~ 17번 : delete\_ok.jsp에서 넘겨받은 idx값과 password라는 이름을 가진 text박스에 입력한 패스워드를 변수를 선언해서 저장한다.      * 34번 ~ 50번 : delete\_ok.jsp 파일에서 입력한 패스워드가 맞으면 게시물을 삭제하는 소스이다. * 34번 : delete\_ok.jsp에서 넘겨받은 삭제할 게시물의 idx값을 where조건으로 두어 그 게시물의 패스워드를 sql1에 저장한다. * 39번 : sql1에서 찾은 패스워드 레코드를 위에서 선언해준 password변수에 저장시킨다. * 42번 : 그리고 delete\_ok.jsp에서 입력한 패스워드와 그 게시물의 패스워드가 일치하면 삭제할 게시물을 테이블에서 삭제 시킨다. 그러면 게시물이 삭제가 된다.      * 53번 ~ 65번 : 삭제가 잘 되었는지 안되었는지 확인하는 구문이다. * 54번, 55번 : delete\_ok.jsp에서 입력한 패스워드와 테이블에서 꺼내 온 패스워드가 일치하면 삭제가 되었다는 알림창과 함께 중복삭제를 막기 위해 redirect.jsp로 넘어가게 한다. * 63번, 64번 : delete\_ok.jsp에서 입력한 패스워드와 테이블에서 꺼내 온 패스워드가 일치하지 않으면 삭제가 되지 않았다는 알림창과 함께 중복 삭제를 막기 위해 redirect.jsp로 넘어가게 한다. |

|  |
| --- |
| (7) redirect.jsp     * 11번 ~ 14번 : delete.jsp에서 이 파일로 넘어 오면 삭제한 그 게시물에 해당하는 메뉴의 첫 화면으로 이동하게 한다. |

|  |
| --- |
| (8) modify\_write.jsp     * 12번 : content.jsp에서 게시글 수정을 누르면 넘겨준 그 게시글의 idx값을 idx변수에 저장한다.      * 46번 ~ 50번 : 로그인을 했을 경우와 했을 경우를 구분 지어 주는 소스이다. * 46번 : 로그인을 했을 때 session id값에 이름을 들어가게 했는데 그 값이 null이다. 즉, 로그인을 하지 않았을 때 로그인을 해달라는 알림창과 함께 이전 페이지로 돌아가게 해서 게시물을 수정할 수 없게 한다. * 50번 : session id값이 null이 아닐 때 즉, 로그인을 하였을 때 게시물을 수정 할 수 있는 코드를 나타내어 준다.      * 87번 ~ 116번 : 게시글을 수정하는 부분이다. 이 부분에 null값이 전송되지 않도록 formCheck() 함수를 이용해 오류검사를 시행한다. * 93번 : 게시글을 수정하고 제출 버튼을 누르면 수정 된 값이 modify.jsp에 post형식으로 넘어간다. 그리고 위에서 선언한 idx에 저장된 값도 같이 modify.jsp에 넘겨준다. * 102번 : 게시물 수정은 로그인을 했을 때만 이용가능 하다. 그러므로 로그인한 회원의 이름을 작성자 텍스트 박스에 기본값으로 넣어준다.      * 14번 ~ 43번 : 오류검사를 하는 자바스크립트 함수 부분이다. * 17번, 19번, 21번 : 제목, 작성자, 내용 부분을 각각 선언한 변수에다가 저장시켜준다. * 24번 ~ 40번 : 그 값들이 null이면 modify.jsp로 넘어갈 수 없게 false를 return시켜준다. |

|  |
| --- |
| (9) modify.jsp     * 14번 ~ 17번 : modify\_write.jsp에서 넘겨준 수정된 값을 선언한 각각의 변수에 넣어준다.      * 34번 ~ 39번 : modify\_write.jsp에서 넘겨받은 수정할 게시물의 idx를 where조건으로 넣어서 그 idx값에 해당하는 레코드 중 count값을 sql1에 저장시킨다. 그리고 그 sql1의 count값을 위에서 선언한 count변수에 저장시킨다.      * 41번 ~ 51번 : 수정할 내용의 게시물의 idx값을 where에 조건을 걸어서 그 게시물에 해당하는 내용의 레코드 값을 UPDATE문으로 새로운 내용을 덮어쓰게 한다. * 53번 : 수정할 내용의 게시물에 새로운 내용을 UPDATE하는 것을 완료하면 그 수정한 게시물로 다시 돌아가게 한다. |

|  |
| --- |
| (10) comment.jsp     * 21번 ~ 23번 : content.jsp파일에 있는 댓글을 작성한 내용들을 comment.jsp파일에서 선언한 각각의 변수에 저장한다.      * 38번 ~ 40번 : 댓글을 저장하는 테이블에 위에서 저장한 변수들을 집어 넣어준다. 여기서 댓글을 작성한 게시글의 idx값도 넣어준다. 그 이유는 그 게시물의 번호를 알아야 다른 게시물에 이 댓글이 표시되지 않기 때문이다.      * 46번 ~ 52번 : 댓글 테이블에 댓글 저장이 끝나면 그 댓글을 작성했던 게시물로 다시 돌아가게 한다. |

5) Login.html

|  |
| --- |
| (1)section     * 281번 ~ 320번 : container 구간으로 <section> 태그로 구조를 나눈다. * 282번 ~ 318번 : <article> 태그로 사진을 받아 줄 공간과 로그인, 회원가입을 진행할 공간을 반으로 나누어 준다. * 284번 ~ 287번 : 선언된 tabs class로 로그인과 회원가입 창을 자유롭게 전환할 수 있게 만든다. * 288번 ~ 317번 : content class를 통해 나뉘어진 두 공간에 본격적인 로그인과 회원가입 형식의 입력창을 만들어준다. * 290번 ~ 301번 : 만들어진 로그인 화면에 값을 출력, DB에 값이 있는지 확인해주는 LOGIN\_CHECK.jsp 파일을 불러준다.      * 304번 ~ 315번 : 만들어진 회원가입 화면에 값을 받아 DB에 저장, 회원가입을 원활하게 진행해주게 할 JOIN\_CHECK.jsp 파일을 불러준다. * 319번 : 나뉘어진 반의 공간에 사진을 불러와 넣어준다.   (2)script     * 326번 ~ 339번 : .tabs, .tab가 선언된 class들을 클릭했을 때, 수행해 줄 function을 입력한다. * 327번 ~ 332번 : Sign in을 눌렀을 때, Sign Up의 화면은 숨겨주고, Sign in 화면을 보여준다. * 333번 ~ 338번 : Sign UP을 눌렀을 때, Sign In의 화면은 숨겨주고, Sign Up 화면을 보여준다. * 340번 ~ 344번 : .contaier, .bg가 선언된 class들의 위에서 마우스를 움직였을 때, 수행해 줄 function을 입력한다. |

6) LOGIN\_CHECK.jsp

|  |
| --- |
| * 18번 ~ 58번 : 로그인을 원활하게 진행시켜줄 SQL문 * 20번 ~21번, 27번 : 오라클 드라이버 객체를 생성하여 DriverManger에 등록시킨다. * 23번 ~ 25번 : 사용하는 오라클 DB의 주소와 접속하기 위한 아이디, 비밀번호 * 29번 ~ 31번 : Login.html의 input에 입력한 값들을 각 변수에 담는다. * 35번 ~ 37번 : SQP구사, 내가 입력한 ID와 Password의 값이 있는지 확인한다. * 39번 ~ 43번 : isLogin은 로그인 확인 유무를 위한 변수, rs.next()가 true라면 정보가 존재한다는 말.      * 45번 ~ 49번 : DB에 내가 입력한 정보가 있다면, 지금 로그인 한 ID와 Password를 Session에 저장하고, InteriorAccessories/InteriorAccessories.jsp 페이지를 띄운다. * 50번 ~ 52번 : DB에 내가 입력한 정보가 없거나 입력한 정보가 틀리면, 경고창을 띄워준다. |

7) JOIN\_CHECK.jsp

|  |
| --- |
| * 17번 ~ 25번 : Login.html 회원가입의 input에서 입력된 필요한 값들을 변수에 담아준다. * 28번 ~ 29번, 35번 : 오라클 드라이버 객체를 생성하여 DirverManger에 등록시킨다. * 31번 ~ 33번 : 사용하는 오라클 DB의 주소와 접속하기 위한 아이디, 비밀번호      * 38번 ~ 42번 : 중복된 E-MAIL이 있는지 체크해준다. * 44번 ~ 49번 : 중복된 E-MAIL을 발견하면 경고창을 띄워주며 회원가입의 진행을 막는다. * 50번 ~ 58번 : 중복된 E-MAIL을 발견하지 못하면, SQL문을 진행하여 입력한 값들을 DB에 저장하고 원활한 로그인을 위해 Login화면으로 넘겨준다. |

8) LOGOUT.jsp

|  |
| --- |
| * 16번 : Session을 재시작 해 로그인 된 정보를 날려준다. * 17번 : 그리고 다시 InteriorAccessorise/InteriorAccessories.jsp 창을 띄워준다. |

9) 파일 경로

|  |
| --- |
| (1) ROOT 파일 – 커버화면, 메인화면    ROOT파일에 커버화면과 로그인 기능의 화면과 jsp파일이 들어있다. |

|  |
| --- |
| (2) 인테리어 소품 메뉴    인테리어 소품 메뉴에 대한 jsp파일이 ROOT파일 안에 InterirorAccessories라는 파일을 따로 지정해서 그 안에 넣어주었다. |

|  |
| --- |
| (3) 가구메뉴    ROOT파일 안에 sofa라는 파일을 만들어서 그 안에 가구 메뉴에 대한 소스들을 넣었다. |

|  |
| --- |
| (4) 인테리어 메뉴    ROOT파일 안에 interior라는 파일을 만들어 그 안에 인테리어 메뉴에 대한 소스 파일을 넣었다. |

|  |
| --- |
| (5) Q & A 메뉴    ROOT파일 안에 question이라는 파일을 만들어 그 안에 Q&A에 대한 소스 파일을 넣었다. |

|  |
| --- |
| (6) css 파일, js파일, image파일    css파일을 ROOT안에 넣어줘서 그 안에 css에 대한 정보를 담는다.    js 파일을 ROOT안에 넣어줘서 그 안에 부트스트랩의 js에 대한 정보를 담는다.    메인 화면에서 사용할 이미지들을 ROOT파일 안에 images라는 폴더를 따로 만들어 그 안에 다 저장을 해준다. |

|  |
| --- |
| (7) 데이터베이스 사진 저장 경로    ROOT파일 안에 데이터베이스에 들어온 사진들을 저장할 수 있게 하는 uploadFile이라는 폴더를 따로 만들어 준다. |

# **4. 프로젝트 결과**

1. cover화면

– cover.html

|  |
| --- |
| 이 화면에서 start버튼을 누르면 Project.jsp파일로 이동하게 된다. |

1. 메인 화면

– Project.jsp

|  |
| --- |
| 이 사이트의 핵심이자 가장 처음으로 나타나는 화면이다.  상단바에 The Interior를 클릭하면 다시 이 화면으로 돌아오고 그 옆에 인테리어 소품, 가구, 인테리어, Q&A메뉴들을 클릭하게 되면 각자의 게시글이 들어간 화면으로 넘어간다.  그리고 오른쪽 상단에는 로그인을 할 수 있는 버튼이 있다. 상단바와 로그인 버튼은  어느 사이트에 이동을 하여도 기본으로 표시된다. |

1. 메뉴들 첫 화면

- InteriorAccessories.jsp, furniture.jsp, interior.jsp, question.jsp 파일

|  |
| --- |
| 상단바에 있는 각 메뉴들을 클릭했을 때 처음으로 나오는 화면이다 오른쪽 게시물은 이 메뉴의 가장 최신 글을 보여준다. 그리고 왼쪽에 작게 나열되어 있는 게시물들은 이 메뉴에 해당하는 전체 게시물을 보여준다. |

1. 게시물 보기

-content.jsp

|  |
| --- |
| 각 메뉴의 첫 화면에서 가장 최신 글에 밑에 있는 CONTINUE READING 버튼이나 왼쪽에 작게 나열되어 있는 전체 게시물의 제목을 클릭했을 때 그 클릭한 게시물만 보여주는 화면이다. 제목, 작성일자, 작성자, 조회수와 게시글을 작성했을 때 넣은 사진과 내용을 보여주고 글 하단에 게시글을 삭제하거나 수정을 할 수 있다. 그리고 게시글 목록으로 라는 버튼을 누르면 다시 메뉴들의 첫 화면으로 이동한다.  그리고 이 화면은 로그인을 하지 않은 상태여서 댓글을 적을 수 있는 공간이 없다. |

1. 게시글 쓰기

-write.jsp

|  |
| --- |
| (1) 로그인을 하지 않았을 경우    메뉴들의 첫 화면에서 하단에 있는 글쓰기 버튼을 눌렀을 때 로그인을 하지 않았다면 나타나는 화면이다 여기서 확인을 누르면 다시 이전페이지로 돌아간다.  (2) 로그인을 했을 경우      메뉴들의 첫 화면 하단에 있는 글쓰기 버튼을 눌렀을 때 로그인을 한 상태라면 이렇게 게시글을 쓸 수 있는 페이지로 이동한다 |

1. 게시글 삭제

-delete\_ok.jsp, delete.jsp

|  |
| --- |
| 여기서 비밀번호를 입력하고 삭제할 게시글에 저장된 비밀번호와 일치하면    비밀번호가 일치하지 않으면    이러한 화면이 나타난다. 이 두 화면 다 확인을 클릭하면 이전페이지로 돌아간다. |

1. 게시글 수정

-modify\_write.jsp, modify.jsp

|  |
| --- |
| (1) 로그인을 하지 않았을 경우    여기서 확인버튼을 누르면 이전페이지로 이동한다.  (2) 로그인을 했을 경우    게시글을 수정할 수 있는 페이지로 이동한다. |

1. 댓글 달기

-content.jsp, comment.jsp

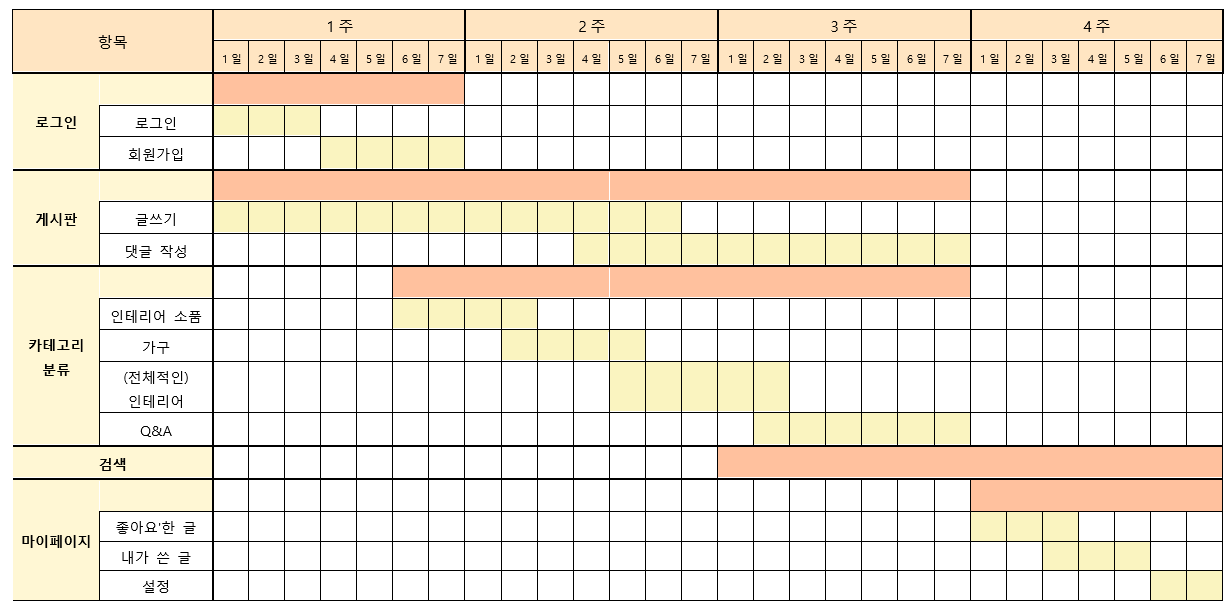
|  |
| --- |
| (1) 로그인을 하지 않았을 경우 댓글을 적을 수 있는 공간이 없다.    위와 같이 오른쪽 상단에 로그인을 하라는 버튼이 있는 것으로 보아 로그인을 하지 않은 상태이고 로그인을 하지 않았을 때 게시물을 들어가 보면 댓글을 달 수 있는 공간이 제공되지 않는다.  (2) 로그인을 했을 경우    오른쪽 상단에 로그아웃이라는 버튼이 있는 것으로 보아 로그인을 했을 때의 상태이고 위와 같이 댓글을 달 수 있는 공간이 생긴다 그리고 댓글을 작성했을 때 댓글이 달린 것을 볼 수 있다. |

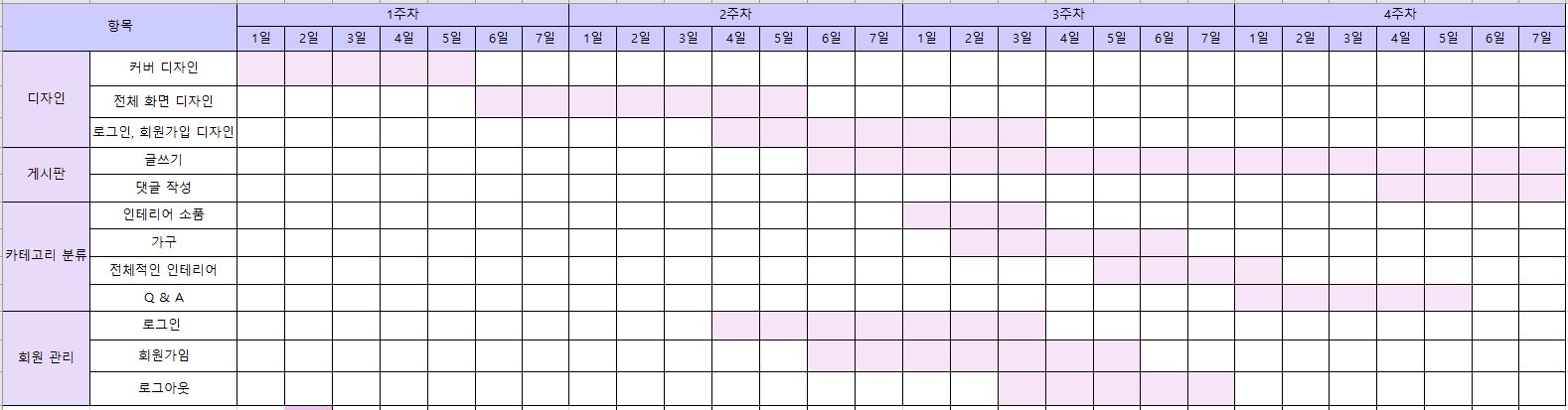
1. 로그인, 회원가입 화면

-Login.html, LOGIN\_CHECK.jsp, JOIN\_CHECK.jsp, LOGOUT.jsp

|  |
| --- |
| (1) 로그인 화면    (2) 회원가입 화면    로그인을 했을 때 화면의 오른쪽 상단에 LOGOUT이라는 버튼이 생성되고, 로그아웃을 했을 때는 오른쪽 상단에 LOGIN이라는 버튼이 생성된다. |

# **5. 일정 대비 진행결과 분석**





|  |  |  |
| --- | --- | --- |
| 기능 | 계획서 | 최종 보고서 |
| 디자인 | 일정 계획에 넣지 않음 | 예상보다 디자인에 시간을 많이 지체에서 다른 구현에 집중을 못하였던 것 같다. |
| 게시판 | 1~3주차에 글쓰기와 댓글 작성을 끝낸다. | 디자인 구현이 2주차까지 이어져서 기능 구현을 2주차 중간부터 시작하였다. 생각했던 기간보단 훨씬 일찍 구현하였다. |
| 카테고리 분류 | 2,3 주차에 끝낸다. | 카테고리 분류는 게시판과 동시에 진행해야 한다. 게시판 구현이 4주차 까지 이어져서 카테고리 분류도 4주차 까지 이어졌다. 구현 예상 기간은 비슷했다. |
| 회원관리 | 1주차에 끝낸다. | 디자인을 끝내고 난 뒤에 구현을 시작해서 2주차 중간부터 3주차까지 이어 졌다. 예상 기간과는 거의 비슷했다. |
| 검색 | 3,4주차에 끝낸다. | 생각보다 기능이 어렵고, 생각지도 못했던 디자인에 시간을 지체하게 되어, 구현을 하지 못하였다. |
| 마이페이지 | 4주차에 끝낸다. | 계획서에는 기간을 7일동안 잡았지만, 생각보다 기능이 어려워 구현하다 포기하였다. |

# **6. 팀원 역할**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 기능 | 정현진 | 이자원 |
| 1 | 디자인 | 10 | 40 |
| 2 | 로그인, 로그아웃, 회원가입 기능 구현 | 5 | 40 |
| 3 | 전체적인 기능 구현 | 15 | 10 |
| 4 | 게시판 기능 구현 | 50 | 0 |
| 5 | 최종 테스트 | 20 | 10 |
|  | 합계 | 100 | 100 |

# **7. 총평**

**<정현진>**

예상치 못하게 주요 기능인 게시판을 구현하는 데에 있어서 시간이 너무 오래 걸렸고 전체적인 화면을 디자인하는 데에 시간을 너무 많이 지체되었다. 이로 인해서 계획서에 적은 내용을 다 구현은 못하였지만 구현을 한 기능은 거의 완벽하게 지금 시중에 있는 홈페이지의 게시판처럼 구현을 하였다. 이러한 점에서 완성도의 점수는 높게 주고 싶다. 프로젝트를 진행하면서 인터넷 한 페이지를 구현하는 데에 사소한 거 하나까지 다 신경을 써야 하고 문장 하나하나 구역 하나하나에 신중해야 한다는 것을 느꼈다. 지정한 한 구역을 잘못 설정하게 되면 다른 구역들의 위치가 다 변경되는 오류를 접하면서 느낀 점이다.

시간이 된다면 여기서 마무리 된 프로젝트를 가지고 계획서에 다 구현하지 못하였던 검색, 마이페이지 기능을 개인적으로 구현해 보고싶다. 그리고 부가적으로 사람들이 많이 쓰는 SNS처럼 게시글 좋아요 라던지, 인스타그램처럼 사진의 좌표값을 이용해 정보를 알려줄 수 있는 기능도 구현해 보고싶다.

이 때까지 해왔던 프로젝트 중에서 제일 흥미롭고 재미있게 진행했던 프로젝트였던 것 같고 같이 힘내서 잘 따라와준 팀원인 자원이에게도 고마움을 전한다.

**<이자원>**

이번 프로젝트를 하며 데이터베이스 수업 말고, 프로젝트에 데이터베이스를 처음 적용해봤는데, 생각보다 너무 어려워 애를 먹었다. 그런 체로 프로젝트를 진행하다 보니 계획서에서 로그인에 할당해주었던 7일이라는 시간보다 조금 더 지나서야 구현을 제대로 하게 되었다. 사소한 태그 하나에도 데이터베이스에 값이 제대로 들어가게 되거나, NULL로 들어가게 되거나 하는 것을 보고, 코드를 짤 때 조금 더 신중하게 생각하고 짜야 한다는 생각이 들었다. 그리고 디자인을 해보며 느낀 점인데, 아무리 똑같이 사용하는 태그라도 css를 섬세하게 사용해줄수록 정교해지고, 이쁘게 탈바꿈하는 홈페이지를 보며 재미를 느꼈지만, 사용하기 조금 어렵다는 생각이 들었다.

프로젝트 완성도는 75%라고 생각한다. 로그인과 디자인에 시간이 조금씩 더 투자되다 보니 구현되야 할 것들이 조금씩 밀려버렸고, 결국 검색과 마이페이지는 구현을 못하게 되었다. 하지만 검색과 마이페이지를 구현하지 못한 대신, 로그인과 디자인 그리고 게시판을 보다 완벽하게 구현했다고 생각하기 때문에 완성도를 높게 평가하였다. 로그인을 하는 홈페이지인 만큼 마이페이지도 완벽하게 구현했다면 나무랄 것 없는 정말 완벽한 홈페이지가 되었을 텐데 라는 아쉬움이 있고, 나 때문에 구현이 밀려버려서 제대로 구현을 하지못해 같이 프로젝트를 진행한 현진이 언니에게 미안한 마음도 든다.

3학년 1학기 수업 중 제일 흥미 있었던 수업인 만큼 재미있고, 조금 더 공들여서 프로젝트에 참여 했다고 생각한다.