**REPORT**

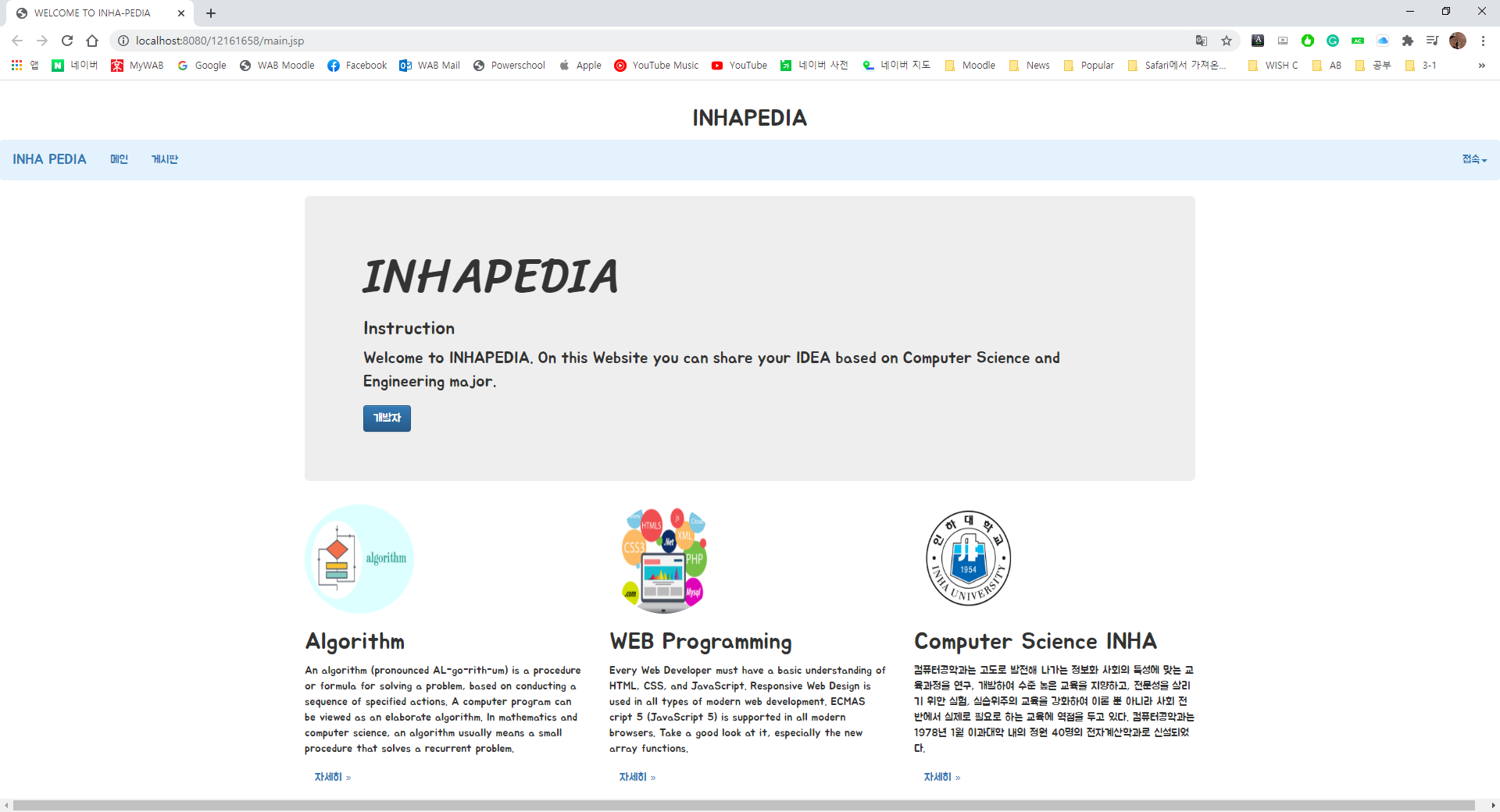


|  |  |
| --- | --- |
| **과목명** | 인터넷프로그래밍(001) |
| **학과** | 컴퓨터공학과 |
| **학번** | 12161658 |
| **이름** | 조재민 |
| **제출일자** | 20.07.01 |

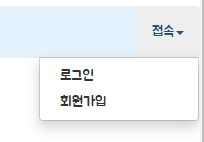
목차

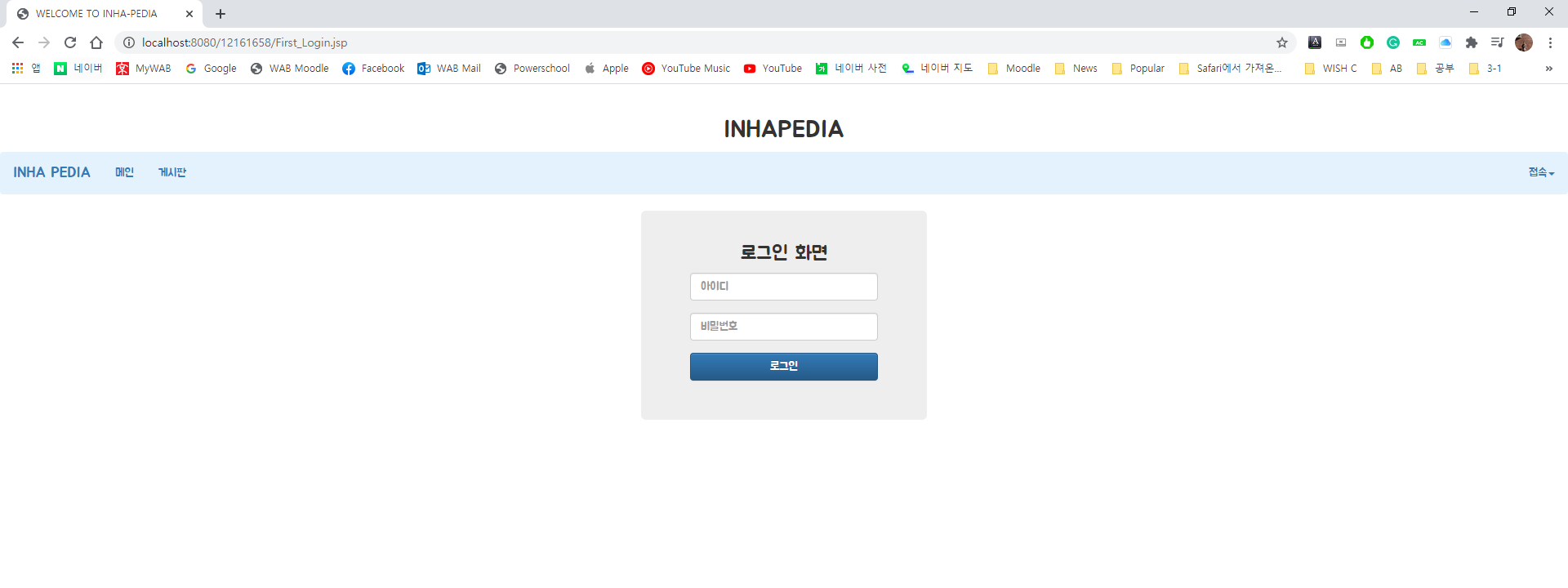
1. 개요
2. 기능/역할 설명
3. 문제점
4. 프로젝트 후기
5. 개요:

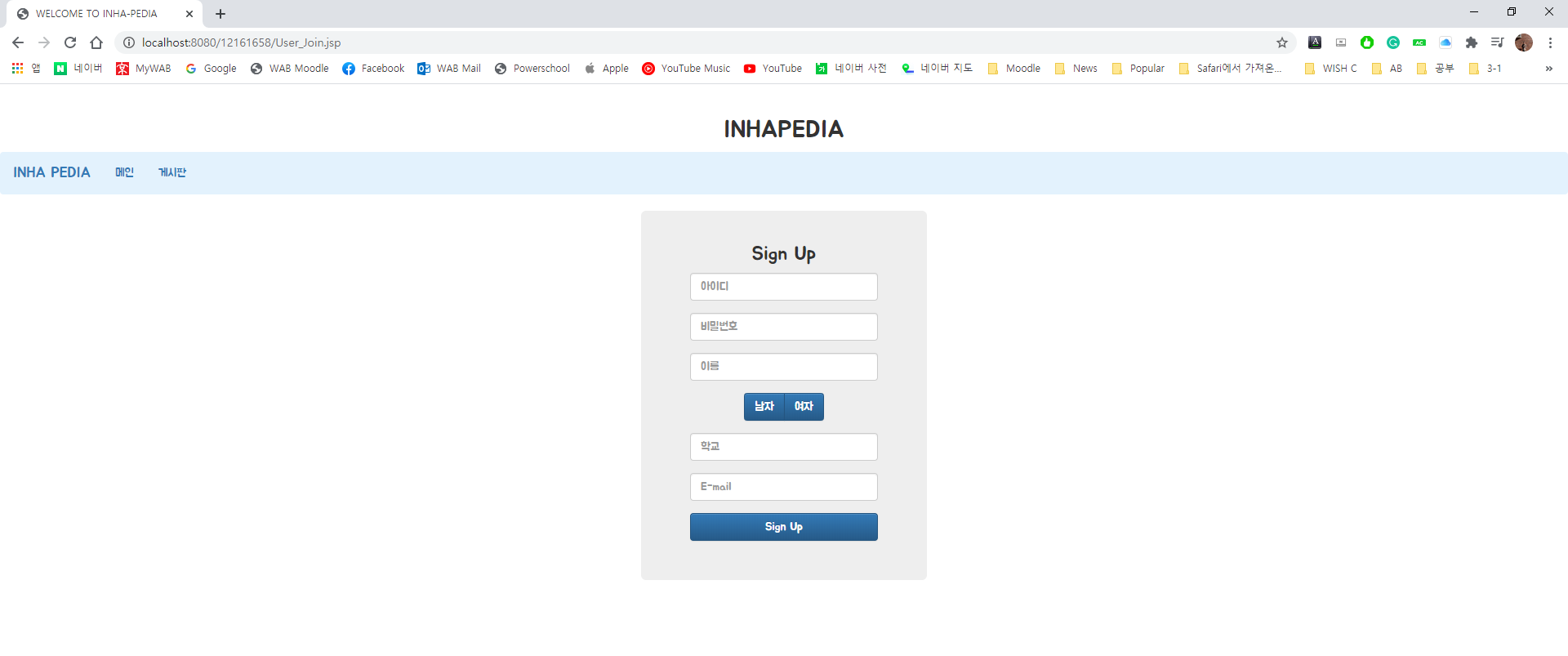
이번 2020년도 1학기 인터넷 프로그래밍 수업에서는, html5, JavaScript, CSS, 등을 활용하여 JSP를 만들고, MariaDB를 이용한 JDBC를 활용하여 ‘블로그형 게시판 만들기’ 프로젝트를 진행하였습니다. 이번에 제가 만든 게시판은 ‘INHAPEDIA’라는 이름의 컴퓨터 공학에 관련된 정보를 자유롭게 주고받는 블로그형 게시판입니다. 해당 웹 사이트에서는 누구든지 회원가입을 하여 자유롭게 글을 남기고, 글에 댓글을 남기며 정보를 공유할 수 있다는 취지를 바탕으로 만들었습니다. 또한 댓글 기능에 이모티콘 요소를 추가하여 구현하였습니다. 게시글 에는 파일을 자유롭게 업로드할 수 있으며, 해당 파일을 누구든지 확인하여 큰 화면에서 볼 수 있습니다.

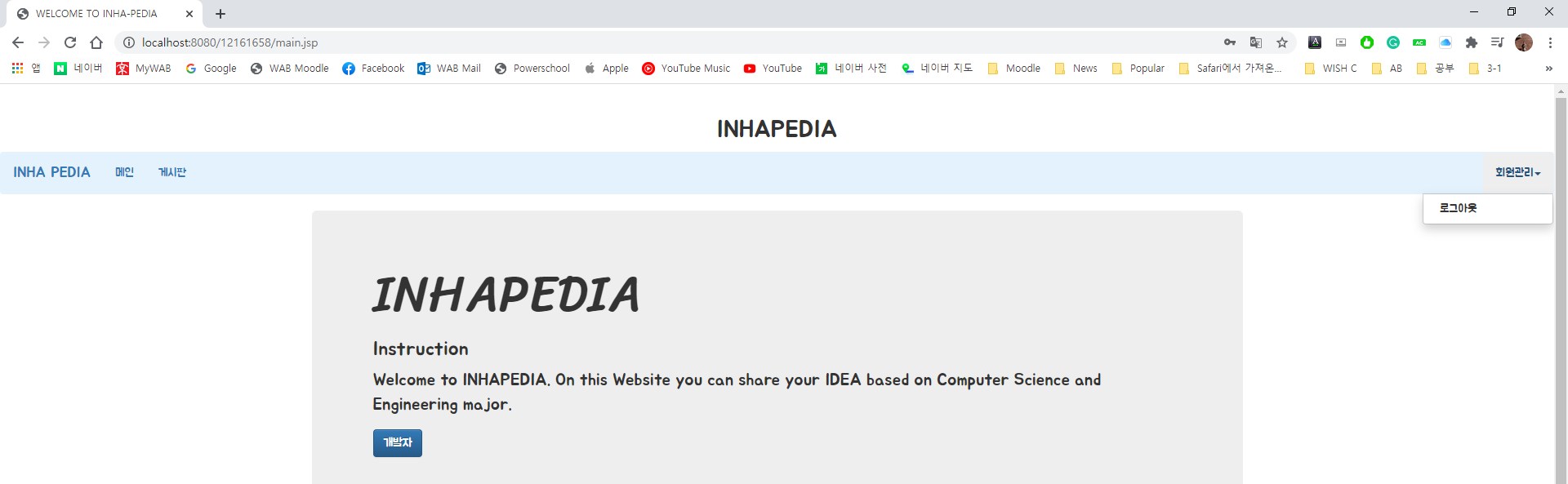
1. 기능/역할:

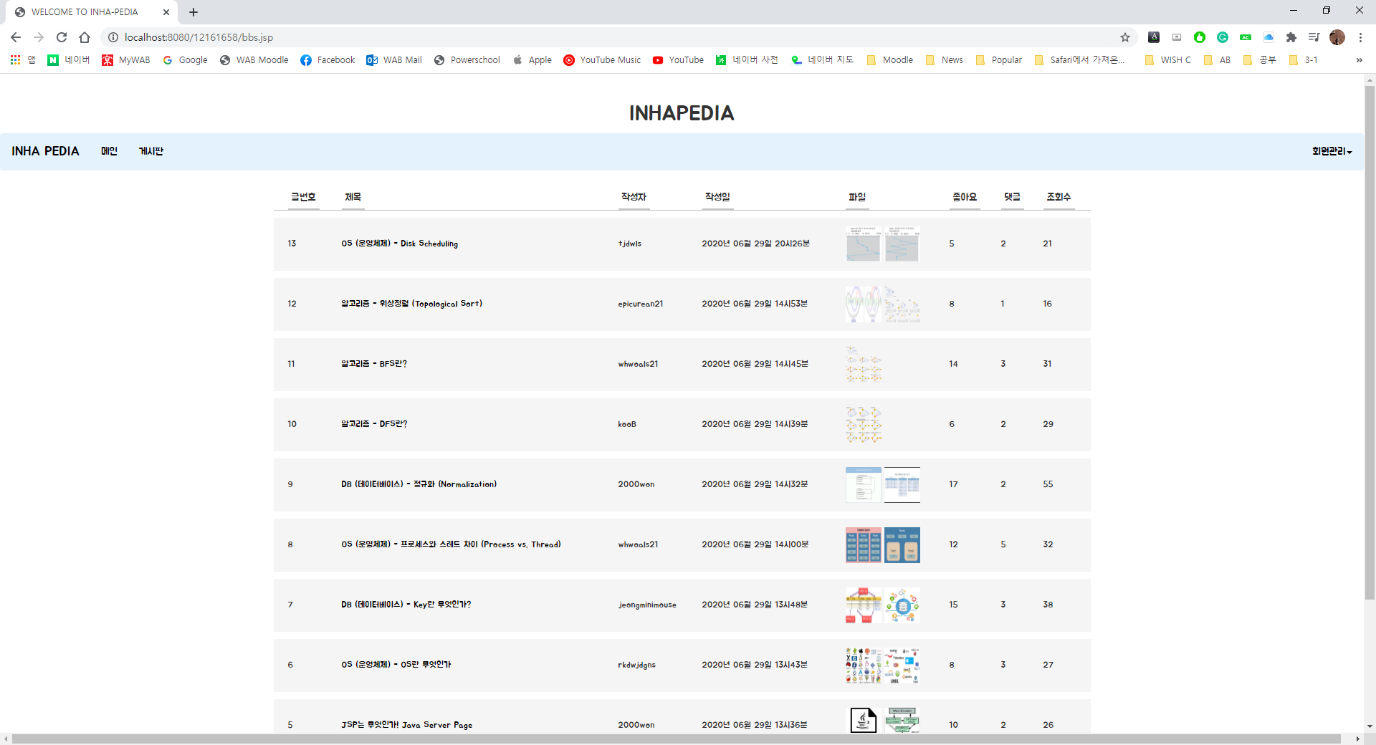
프로젝트의 첫 화면입니다. XML에 welcome-file로 index, default를 설정해 놓았기 때문에 프로젝트 **파일안에 있는 main.jsp, 또는 index.html파일로** 실행했을 시 나오는 첫 화면입니다. 해당 화면은 사용자가 로그인이 되어있지 않아도 볼 수 있습니다. 간략하게 ‘INHAPEDIA’의 설명과 각종 컴퓨터공학 내용이 담겨 있습니다. 설명란에 있는 개발자 버튼을 눌렀을 시 저의 기본적인 정보가 담긴 page로 넘어갑니다 (이부분은 프로젝트와 별개인 사이트입니다). Algorithm, WEB-programming, Computer Science INHA에도 자세히 버튼을 누르면 각각 연관된 page로 이동합니다. 이 부분은 main 화면을 꾸미기 위하여 적절한 연관성의 내용을 추가한 것입니다. Main은 첫 화면으로써 사용자가 선택을 할 수 있습니다. 게시판이 나열되어 있는 게시판화면으로 가기 위하여 위에 navigator 바에 있는 게시판을 클릭할 수 있고, 로그인 혹은 회원가입을 하기 위하여 navigator 바 제일 오른쪽에 있는 접속 버튼을 누르면 회원가입 혹은 로그인 선택권의 버튼이 내려옵니다.



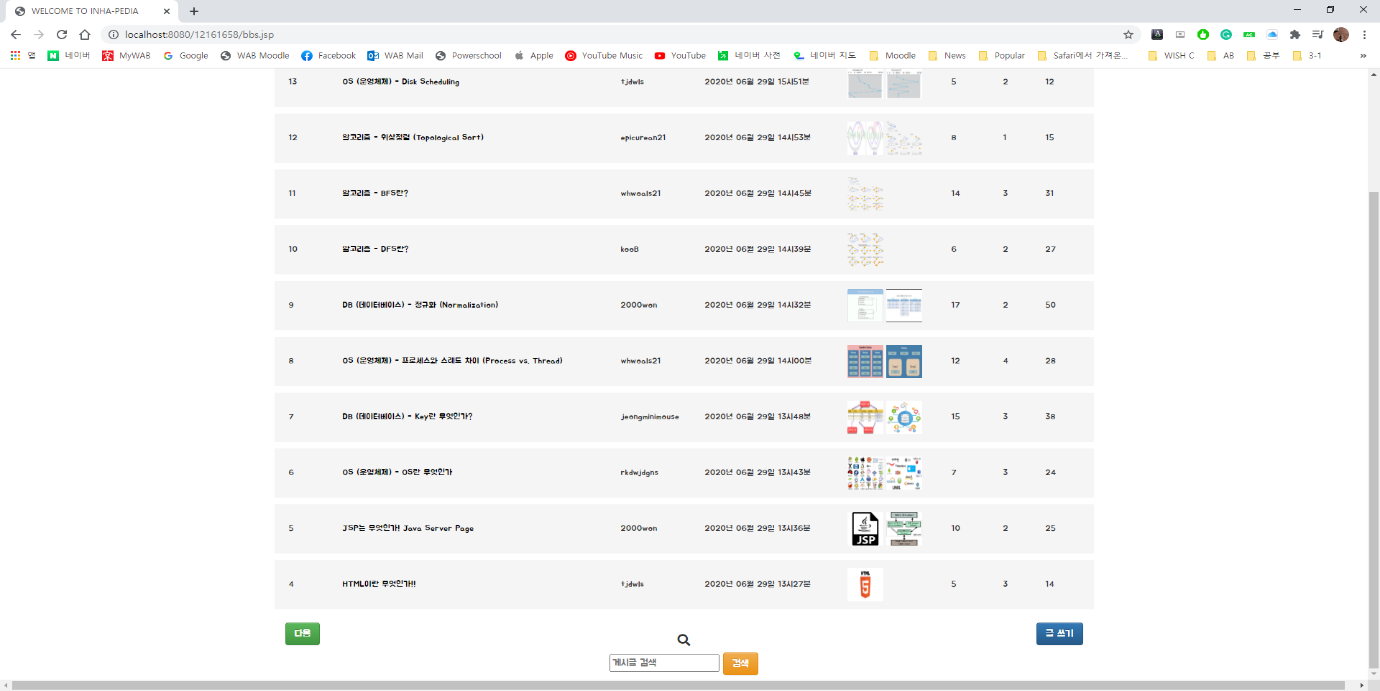


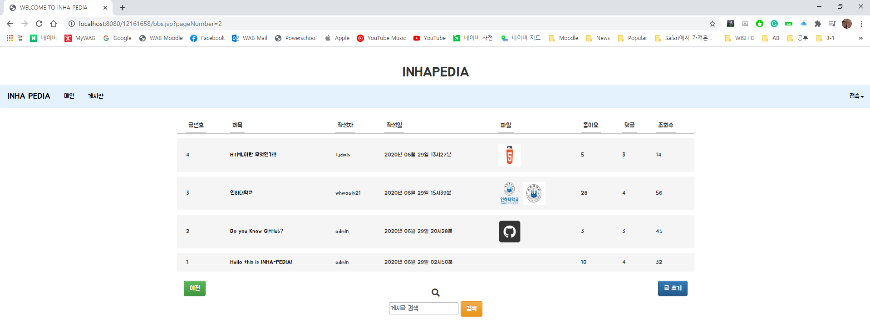
 로그인을 하기위해 로그인 버튼을 눌렀을 시 나오는 화면 (First\_Login.jsp) 입니다. 로그인을 위해서는 우선 사용자의 정보가 Database에 존재하여야 합니다. 입력되는 ID와 Password를 입력 받아 login\_try.jsp 파일로 이동합니다. 해당 파일에서는 미리 생성한 user의 정보를 확인하는 객체를 생성하여 입력된 정보가 옳은 지 확인하여 맞을 경우 session에 해당 사용자 정보를 입력하고, 틀렸을 경우 다시 입력하라는 경고와 함께 First\_Login.jsp 파일로 되돌아옵니다.

사용자가 회원가입을 하기위한 페이지 (User\_Join.jsp) 입니다. 아이디와 비밀번호, 이름, 성별을 선택하고, 학교와 메일을 기입하였을 때 정상적으로 회원가입이 진행됩니다. 빈 곳이 있다면 회원가입 진행 전 경고와 함께 빈 곳을 채우라는 경고가 출력됩니다. 또한, ID의 중복을 막기위해 Join.jsp 페이지로 넘어갔을 때 미리 만들어진 User\_check class의 join함수에서 데이터베이스에 insert sql문을 실행하였을 때 함수 리턴 결과값이 -1이면 primary key인 userID가 중복되었다는 뜻으로써 오류가 발생하여 사용자에게 이미 존재하는 아이디임을 알려주고 다시 회원가입 화면으로 돌아옵니다. 정상적으로 회원가입에 성공하면 입력한 정보로 ‘로그인한’ 상태로 메인화면 (main.jsp)로 돌아갑니다.

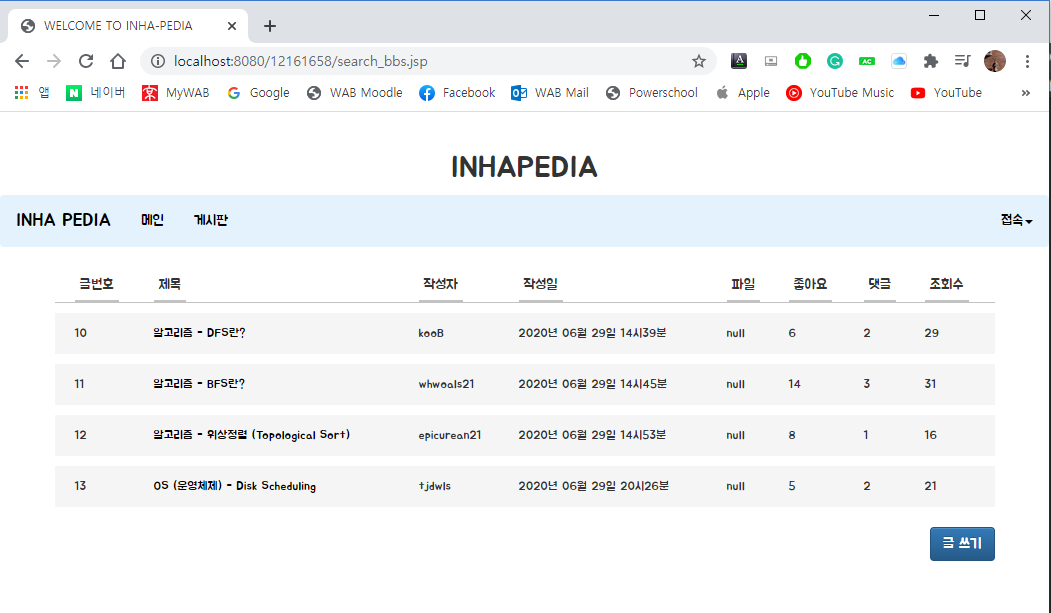
 회원가입 후 자동으로 해당 아이디로 로그인 된 뒤 main으로 이동된 상태의 화면입니다. 이제 navigator의 오른쪽 버튼이 접속이 아닌 회원관리, 로그아웃 버튼으로 변경되어 있습니다.

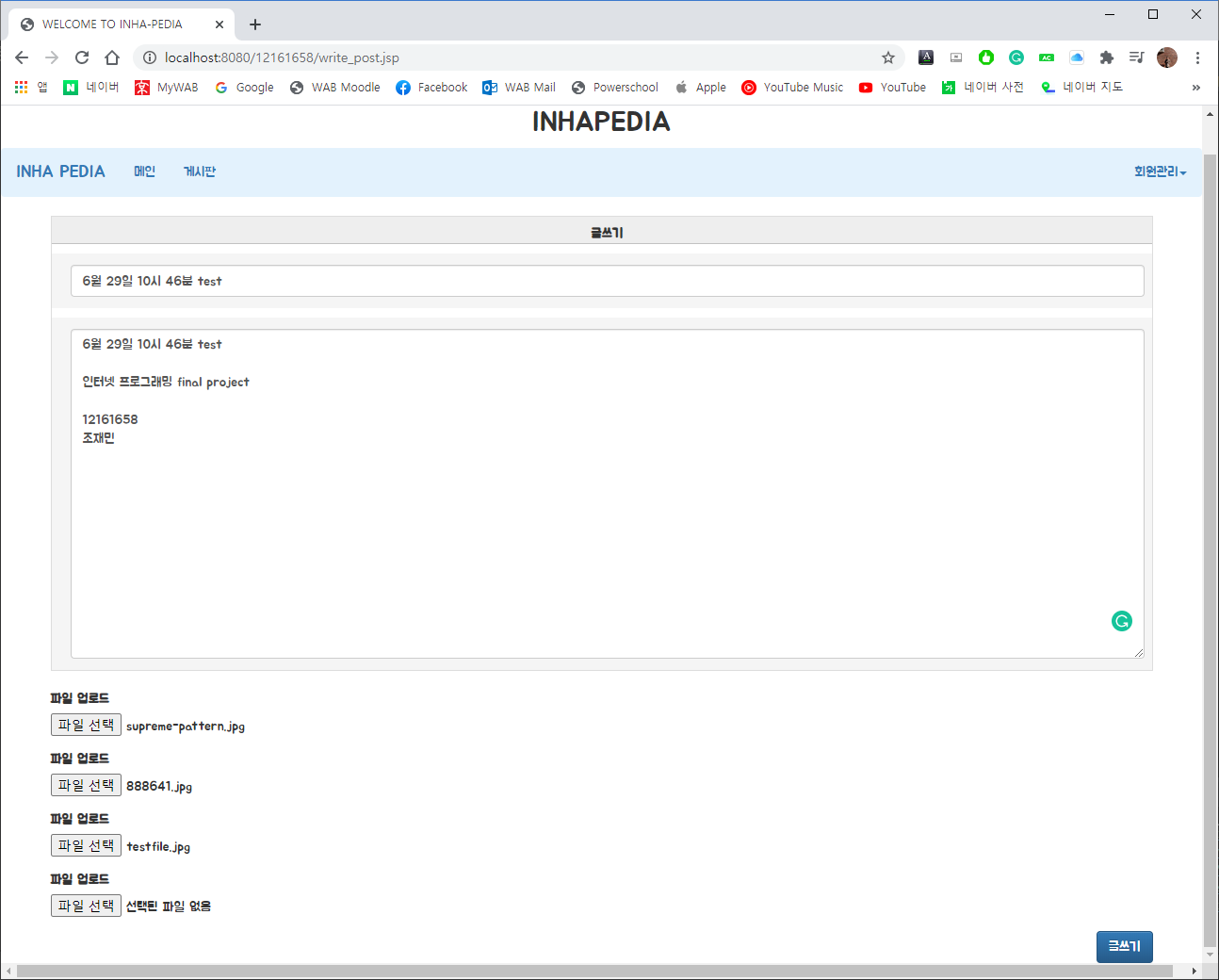
게시판 화면입니다. 게시판은 차례대로 글 번호, 제목, 작성자, 작성일 (수정일) 파일, 좋아요, 댓글, 조회수를 나타냅니다. 글 번호는 사용자가 글을 작성하였을 경우 차례대로 올라가며, 게시판 한 페이지에는 최대 10개의 게시글만이 보여집니다. 작성일은 최초 글이 작성되었을 때의 시간을 담고 있다가, 해당 작성 글의 작성자가 글을 수정하였을 때 최종적으로 수정된 시간이 작성일로써 남아있습니다. 파일은 게시글 목록 페이지에서 최대 두개만 출력됩니다. 해당 사진은 <a>태그로 감싸져 있으며 사진을 눌렀을 때 새 창에 해당 사진을 큰 화면으로 출력합니다. 원하는 게시글을 보는 방법은 제목을 눌렀을 때 해당 게시글 페이지 (view\_post.jsp)로 이동합니다. 조회수는 해당 게시글에 진입 하였을때마다 count를 하여서 1씩 증가합니다.

페이지는 최대 10개의 post가 보여지기 때문에 10개 이상의 게시글이 존재할 경우 다음 버튼이 보여집니다. 다음 버튼을 누르면 다음 page로 넘어가 더 오래전에 생성된 게시글 (더 낮은 게시글 번호)를 표시합니다.

첫 페이지가 아닌 경우 이전 버튼 역시 보이며 이전과 다음 버튼으로 게시글 목록을 이동할 수 있습니다.

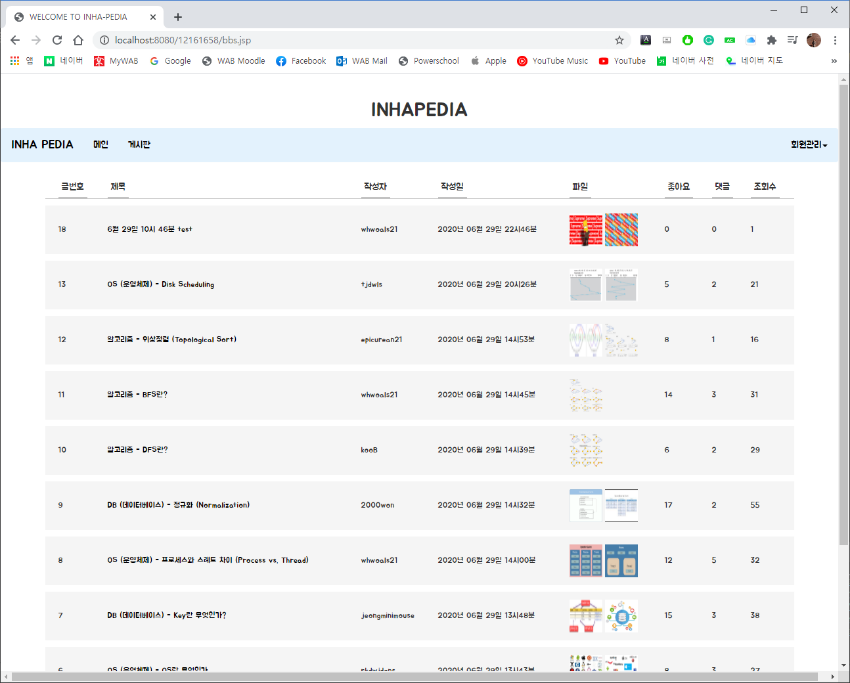
게시글 목록 하단에는 검색 버튼과 글쓰기 버튼이 존재하며, 게시글 검색은 제목과 내용을 토대로 검색 결과가 출력됩니다.

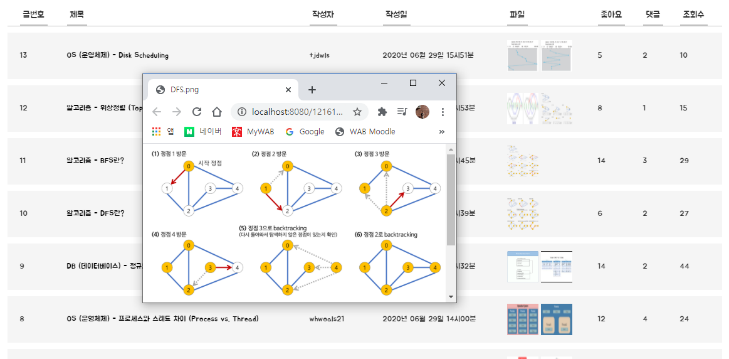
게시글 목록에서 ‘알고리즘’을 검색한 결과입니다. 제목에 알고리즘이 포함되어 있거나, 본문에 알고리즘이 포함되어 있으면 해당되는 페이지들을 search\_bbs.jsp 화면에 목록으로 출력합니다.

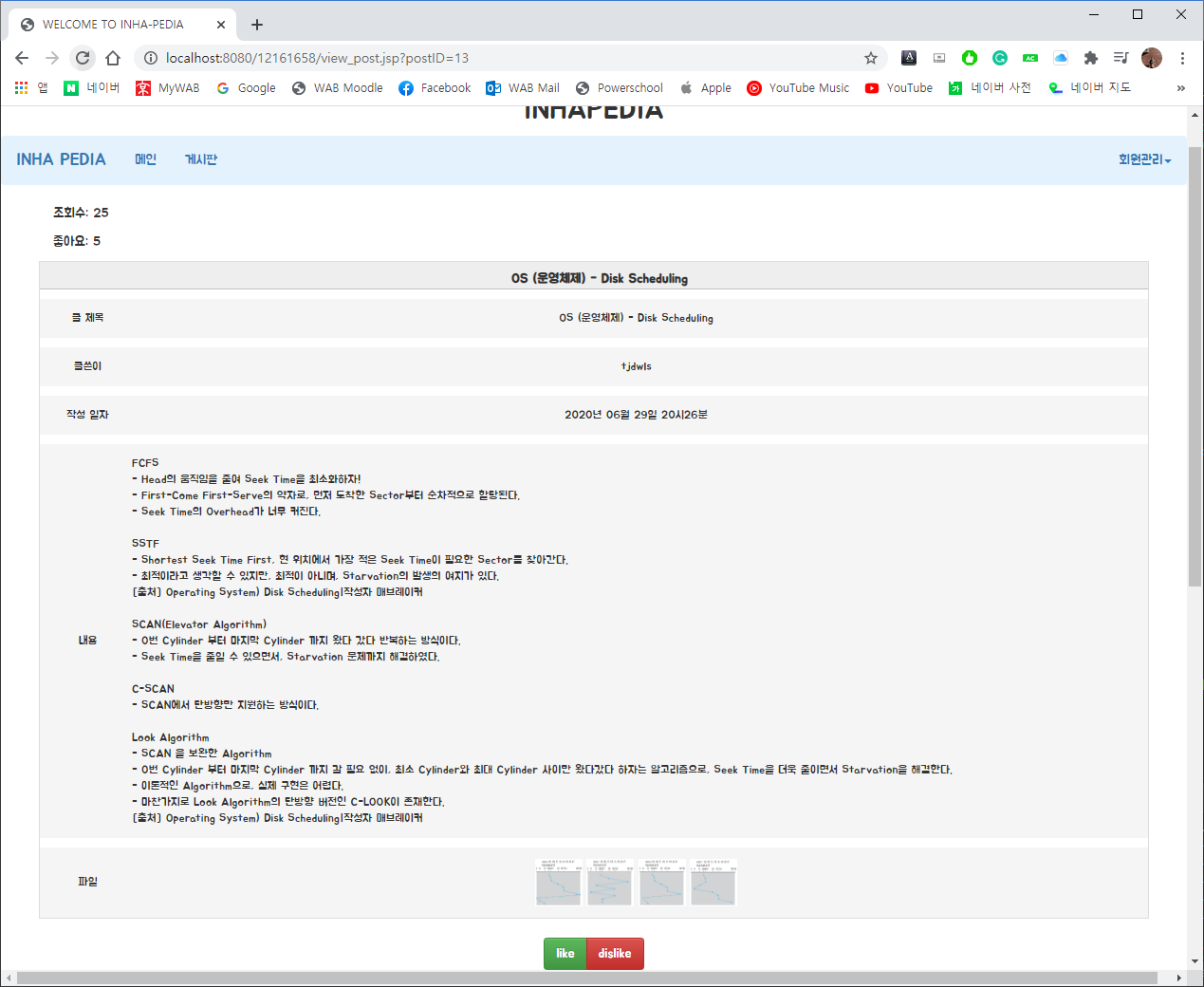
마지막으로 게시판 목록 페이지에는 글쓰기 버튼이 있습니다. 해당 버튼을 선택하면 write\_post.jsp 화면으로 넘어갑니다.

글쓰기 버튼을 선택했을 때 나오는 write\_post.jsp 화면입니다. 간단하게 글 제목과 글 내용을 입력하게 되어 있습니다. 또한 각 게시글 당 최대 4개의 file을 업로드 할 수 있도록 설정해 두었습니다. 제목 혹은 글 내용에 아무것도 입력하지 않았을 경우 글쓰기 버튼을 눌렀을 때 공백이 있다는 경고와 함께 입력이 필요하다고 알람을 줍니다. 또한 로그인이 되어있지 않은경우 글쓰기 버튼을 누르면 로그인이 필요하다는 경고창과 함께 로그인 화면 First\_Login.jsp 화면으로 넘어갑니다.

파일은 꼭 순서대로 입력할 필요가 없습니다. 4개의 선택 중 아무 곳에나 파일을 첨부하여 글쓰기에 성공하였을 때, write\_do.jsp 폴더로 이동하여 각 게시글 제목, 게시글 내용, 첨부된 파일을 BbsCheck class와 fileCheck class를 활용하여 database에 입력합니다. 성공적으로 입력이 완료되었을 경우 다시 게시글 목록 페이지 (bbs.jsp)로 돌아갑니다.

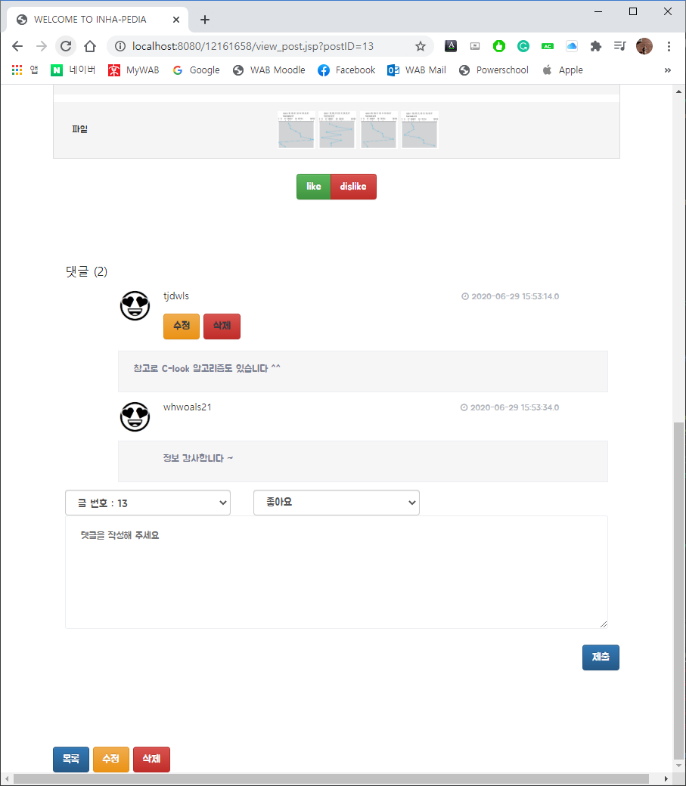
게시글 목록으로 돌아온 화면입니다. 가장 위에 정상적으로 글이 저장되어 작성일에 맞게 저장 되어있으며, 3개의 file을 첨부하였지만 두개만 썸네일로 출력되는 것을 확인할 수 있습니다.

원하는 사진을 게시판 목록에서 선택했을 때 나타나는 화면입니다. 각각의 사진은 view\_file.jsp로 이동하는 링크로 연결되어 있고, 해당 사진을 눌렀을 때 새로운 창에 누른 사진이 출력되도록 설계하였습니다.



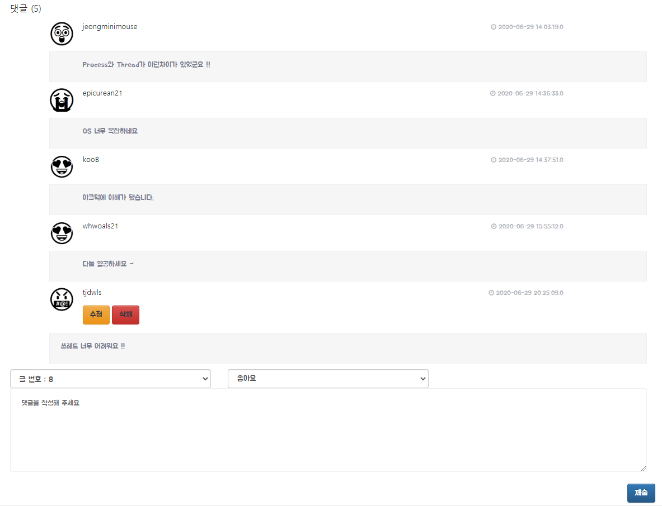
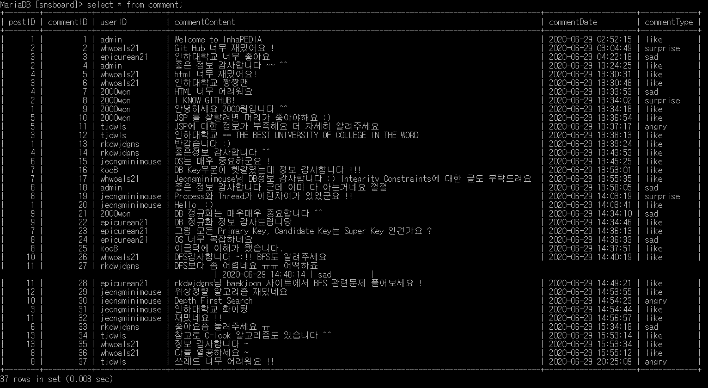
원하는 게시글의 제목을 선택하였을 때 진입하는 view\_post.jsp 화면입니다. 선택한 게시글의 postID에 맞추어서 해당 게시판에 들어갑니다.

각 게시판에는 조회수와 좋아요 수가 좌측 상단에 위치하고, 이후로 Table 형태로 제목, 작성자, 작성일자와 내용, 그리고 파일이 보여집니다. 이곳에서도 파일을 눌렀을 때 view\_file.jsp 파일로 옮겨지며 해당 파일을 새로운 창에서 보다 크게 볼 수 있습니다. Post 밑에는 해당 게시글에 좋아요와 싫어요를 남길 수 있는 like와 dislike 버튼이 있습니다. 해당 버튼들을 눌렀을 때 각각 like\_try.jsp 혹은 dislike\_try.jsp로 보내져 해당 게시글의 좋아요 정보를 불러와 변경시킨 뒤 다시 게시글 화면으로 돌려 보내줍니다. 이는 빠르게 처리하여 한 화면에서 좋아요/싫어요를 눌렀을시 그 화면 그대로 좌측 상단의 좋아요 개수가 변경되도록 처리하였습니다.

게시글의 아래 부분입니다.

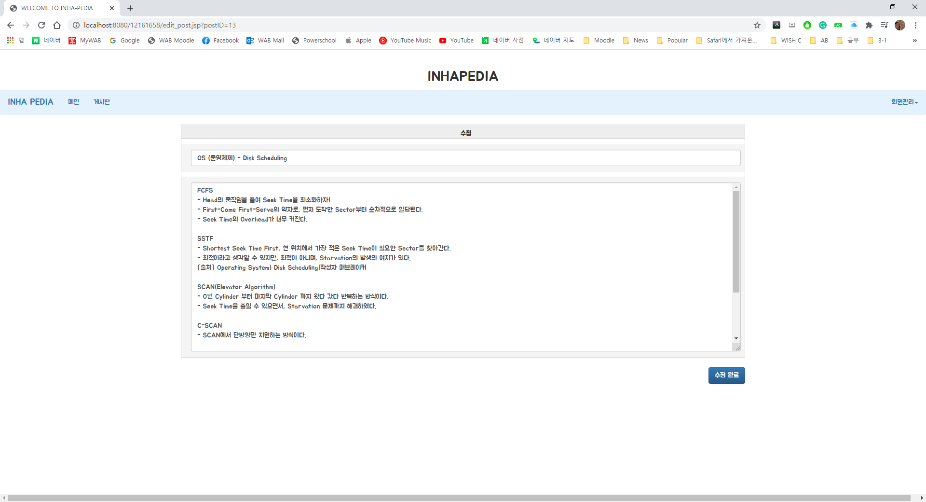
각 게시글에는 별도의 댓글이 있습니다. 로그인이 되어있다면 댓글을 남길수 있고, 자신이 적은 댓글은 수정 및 삭제가 가능합니다.

또한 단순 글로써 남기는게 아닌, 이모티콘 형식의 기분을 글에 표현 할 수 있게끔 하였습니다.

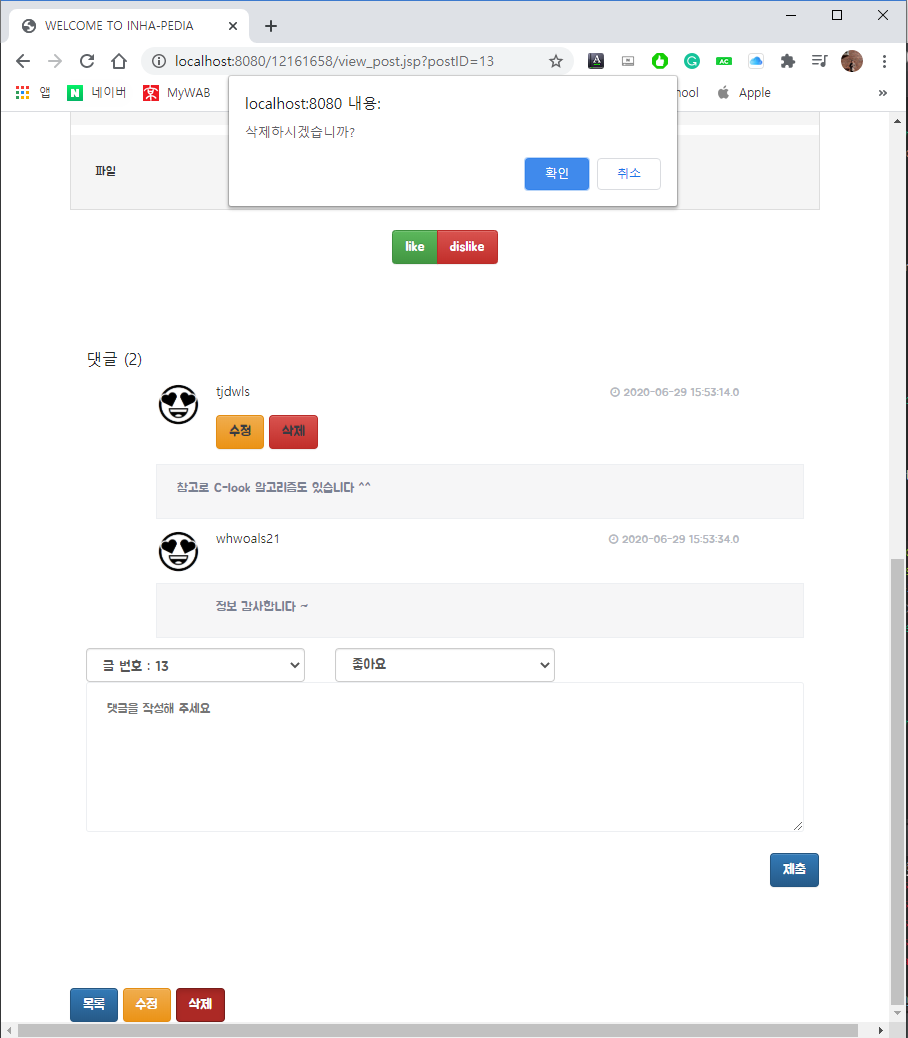
댓글을 작성할 때 총 4가지의 기분을 선택할 수 있습니다. 좋아요, 화나요, 놀라워요, 슬퍼요, 4가지로 이모티콘 이 표현됩니다.

각 댓글역시 별도의 database table을 갖고 있습니다. 해당 Post에 맞는 댓글을 출력하기 위하여 post에는 comment의 개수를 갖고 있고, view\_post.jsp 화면에서 해당 postID에 존재하는 모든 comment를 보이도록 설계하였습니다.

각 게시글에 진입하였을 때 해당 게시글의 작성자와 현재 로그인된 session 사용자가 일치한다면, 게시글 수정과 삭제 버튼이 목록버튼 옆에 출력되도록 설계하였습니다.

게시글 수정 page입니다. edit\_post.jsp 화면으로 들어가며, 이 곳에서 해당 게시글의 제목과 내용을 수정할 수 있습니다.

불가능한 접근 (예를 들어 edit\_post.jsp 로 바로 접근)등을 막기위하여 첫번째로 해당 edit\_post의 userID와 session에 등록된 아이디가 일치하는지 확인하고, 불가능한 접근으로 일치하지 않는다면 다시 해당 게시글 화면으로 되돌아갑니다.

또한 게시글을 변경하였을 때 변경한 시간을 새롭게 해당 post의 postDate에 넣어주어 수정된 날짜가 새롭게 출력되도록 설정하였습니다.

사용자가 자신의 게시글을 삭제하려 한다면 추가적으로 삭제를 하겠냐는 경고글과 함께 재확인을 합니다. 여기서 확인을 눌렀을 경우 정상적으로 게시글이 삭제됩니다. 이때 해당 게시글에 존재하던 comment 역시 함께 삭제됩니다.

게시글을 삭제하는 경우에는, 실제 database에서 삭제하는 것이 아닌, bbs 테이블에 존재하는 postAvailable 변수를 0으로 설정함으로써 게시판 리스트에 노출이 안되게 합니다. 그렇기 때문에 postID 즉 게시글 번호는 유지되면서 올라갑니다. postAvailable attribute의 역할이 바로 해당 게시글을 유저들에게 보이도록 할 것인가 아닌가를 결정하는 것입니다.

이러한 기능들을 담고 있는 아주 가벼운 느낌의 블로그형 게시판입니다. 사용자들이 서로 ‘Wikipedia’와 같이 글을 쓰고 사진을 올릴 수 있습니다. 테마는 컴퓨터 공학에 관련된 글들이며, 글들에 각자 기분을 표시하는 댓글을 달 수 있고, 좋아요 혹은 싫어요를 눌러 변경시킬 수 있습니다.

1. 문제점

이번 프로젝트를 진행하며 총 4개의 문제점이 있었습니다. 파일 입출력 부분, 함수를 만들면서 생기는 개발 문제 등, 대다수의 문제들은 스스로 해결하였지만 끝내 해결하지 못한 부분이 있었습니다.

첫번째로, 저는 데이터베이스에 bbs라는 table과 file이라는 table을 postID와 fileName으로 연결하였습니다. bbs table에는 file table에서 fileName을 foreign key로 받아오고, file에서는 postID를 foreign key로 가져왔습니다. 게시글 목록을 출력하는 화면에서 원했던 설계는 bbs table의 정보를 갖고 있는 객체를 생성하여 모든 부분을 처리하고 싶었으나, 이유는 정확히 알 수 없는 database 오류로 결국 file들은 file table의 별도의 객체를 만들어서 해결하였습니다. 또한 그 과정에서 랜덤 적인 오류가 발생하기도 했습니다. NullPointerException이 발생하기도 하였는데 지속적인 오류가 아닌 ‘특별한’, 알 수 없는 상황에서만 발견되고 또한 그 상황이 자주 일어나지 않아 해결하지 못한 오류로 남아있습니다.

두번째로, 게시글을 좋아요, 싫어요를 적용하는 과정에서 최초로 원했던 건 네이버/다음 등에서 볼 수 있는 좋아요와 싫어요를 동기화 시켜서 해당 유저당 한번씩만 누를 수 있게 만들고 싶었습니다. 하지만 저는 로그인이 되어있지 않더라도 모든 게시글을 볼 수 있고, 좋아요/싫어요를 누를 수 있습니다. 그렇기 때문에 따로 session으로 처리하지 못하여 좋아요/싫어요를 무한대로 누를 수 있습니다. 이 부분은 만약 로그인 되어있을 때만 좋아요 싫어요를 누를 수 있도록 변경하면 매우 쉽게 해결할 수 있는 문제지만 저는 누구든지 누를 수 있다는 최초의 디자인적 아이디어를 유지하다 보니 해결 못한 문제로 남아있습니다.

세번째 역시 게시글 설계관련 문제입니다. 누구든지 게시글을 볼 수 있기 때문에 조회수 역시 한정적이지 않고 무한대로 늘어날 수 있습니다. 심지어 해당 게시글 에서 refresh (f5)를 했을 때도 조회수는 올라갑니다.

마지막은 게시글 화면 문제입니다. 초기 조건 중 하나인 최대 100자만을 노출시키고 더 보기 기능을 추가하여야 하지만, 외부 CSS를 적용해 놓은 table에서 원하는 대로 CSS를 변경하는 것에 어려움이 생겼습니다. Max-length등을 설정하고 다른 CSS로 추가적인 화면이 노출되도록 설계하고 싶었으나 실패하였습니다. 그렇기 때문에 저는 작성할 때 우선 최대 글자수를 512자로 제한해 두고 모든 걸 한번에 보이도록 설계하였습니다.

디자인적 문제 혹은 기능적 문제등 여러 문제가 있지만 제 개인적인 기준에서는 위 네가지 문제가 제 게시판형 블로그의 대표적인 문제점이라 생각합니다.

1. 프로젝트 후기

컴퓨터 공학과 3학년 재학생이지만 평소 해오던 전공 공부는 이론에 그쳤었습니다. 서버와 데이터베이스, 즉 Back-end 쪽 개발자에 관심이 있었기 때문에 주변에서 Android나 IOS, 혹은 Web쪽 개발 공부를 해올 때 단 한번도 해본 경험이 없었습니다. 2학년때 수강신청에 실패하여 3학년 1학기에 인터넷 프로그래밍을 듣게 되었습니다. 인터넷 프로그래밍 수업을 2학년때 들었다면 얼마나 좋았을까 라는 생각을 할 정도로 매우 중요한 수업이라 생각합니다. 사실상 처음으로 대학교에서 컴퓨터공학과 학생으로써 ‘개발’을 해볼 수 있는 수업이었습니다. Java, 객체지향 수업 때 만드는 계산기, Tetris 게임 등 개발에 관련된 코딩은 많이 해보았지만, 이번 인터넷 프로그래밍 프로젝트를 진행하며 Front부터 데이터 베이스 Back-End까지 full-stack으로 개발을 해보며 많은 걸 배웠습니다. 또한 교수님께서 말씀하신 대로 인터넷 프로그래밍은 (혹은 모든 개발은…) 코드 작성과 동시에 결과물이 눈에 보였기 때문에 매우 흥미롭게 프로젝트를 진행하였습니다. 심지어 시험기간에 다른 과목 공부가 하기 싫으면 도피성으로 프로젝트 개발을 하곤 했습니다. HTML, CSS, JavaScript, JSP, MySQL 등 정말 많은 부분을 배운 것 같습니다. 비록 부족한 결과물을 만들어 냈지만 (최초 원했던 거와 비교하면..) 스스로 많이 배우고 재밌었던 것에 만족합니다. 기회가 된다면 스스로 좀더 공부하여 더 나은 나만의 블로그를 만들어 보고 싶습니다.