

블로그 사이트 구현 프로젝트

(Blog site implementation project)

본 보고서를 2022년도 Project [블로그 사이트 구현 프로젝트]의
최종보고서로 제출합니다.

학부(전공)/학과: 응용소프트웨어공학과

담당(지도)교수: 김남규 교수님

팀명: 구글링

제출일: 2022년 12월 13일

팀 장 : 신민주 20213067

팀원1 : 우가현 20213062

팀원2 : 성채은 20213014

동의대학교 응용소프트웨어공학과 학과장 귀하

〈 요약 문 〉

프로젝트 목표 (300자내외)	사용자가 회원가입 및 탈퇴가 가능하도록 하고 회원 등록 시 로그인 및 로그아웃이 가능하도록 한다. 사용자 본인의 프로필 변경을 가능하게 만들고 본인의 개인 블로그 주소를 생성한다. 로그인 후 개인 블로그에 게시글 작성이 가능하게 한다. 게시글 작성에는 제목과 내용, 파일을 업로드를 할 수 있도록 하고 업로드할 시 작성 날짜가 자동으로 저장된다. 작성된 게시글은 본인의 블로그에서 확인할 수 있다. 상대방 블로그에 접속 시 상대방의 게시글 목록과 게시글을 확인할 수 있게 한다. 상대방의 게시글에 조회수 기능도 추가하고, 게시글 하단에 댓글이 보이도록 한다. 로그인한 상태로 댓글을 입력하면 자동으로 작성자, 댓글 내용, 작성 날짜가 업로드되게 한다. 사용자가 자주 접속한 블로그의 다른 게시글들을 추천해주는 추천 블로그 기능과 검색 기능도 구현해서 사용자가 검색했던 내용이 들어간 게시글을 검색 결과로 나타나게 한다.		
내용 (500자내외)	<p>(1) 회원 관리 로그인 및 회원가입 프론트엔드 구축하고 정보 저장하는 DB 'minformation' 생성하여 회원가입 시 정보를 저장하고 만약, 비밀번호와 비밀번호 확인의 문자열이 동일하지 않으면 가입이 되지 않게 한다. 로그인 상태면 저장된 정보를 session_start()로 세션을 유지하고 로그아웃하면 세션이 종료되게 만든다. [프로필 변경]을 누르면 기본 프로필을 변경해 사용자의 프로필 사진을 변경한다.</p> <p>(2) 글쓰기 기능 구축 글쓰기, 내 블로그, 메인 블로그 프론트엔드 구축하고 정보 저장 DB 'post' 생성해 글 작성 시 정보를 저장하고 만약, 글의 제목, 내용 중 하나라도 적지 않으면 글을 생성하지 못하도록 만들기 해당 글 클릭할 때마다 조회수 증가하도록 만든다.</p> <p>(3) 댓글 기능 구축 댓글 프론트엔드 구축하고 정보 저장 DB 'coment' 생성해 댓글 작성 시 정보 저장하기 만약, 내용이 없다면 댓글을 생성하지 못하도록 만든다. 댓글을 작성한 게시글에만 댓글이 보이도록 설정한다.</p> <p>(4) 사용자 알고리즘 메인 블로그에 검색 기능 도입하고 이웃 블로그, 이웃 새글을 랜덤으로 게시글을 추천하고, 내 블로그는 현재 로그인 계정의 작성자가 쓴 글만 보이도록 설정한다.</p>		
기대효과 (200자내외)	기술적 측면은 블로그에서 사용되는 기능을 다른 SNS에서도 똑같이 사용할 수 있으며 다른 기능들을 지속해서 추가하여 더 나은 사이트로 구축해 낼 수 있다. 경제적으로 블로그 사이트에서 자신이 이익을 취할 수 있는 것들을 게시글에 업로드해 수익을 보거나 블로그 자체에서 광고를 띄어 광고비를 통해 수입을 창출할 수 있다. 그뿐만 아니라 사용자 유입 증가를 통해서도 수익이 창출된다. 마지막으로 블로그에 올린 정보로 여러 기관에 도움이 될 수 있으며 모금과 봉사를 위한 모집도 게시글을 통해 진행하여 사회적으로 도움이 필요한 곳에 함께 기부하거나 봉사활동을 진행하여 도움을 준다.		
Keywords	DBMS	로그인 기능	회원가입 기능
	댓글 작성	게시글 작성	사용자 알고리즘

목 차

I. 서론	1
1. 프로젝트의 배경 또는 필요성	1
2. 프로젝트의 목표	1
 II. 프로젝트 수행 내용	 3
1. 프로젝트의 추진전략 및 방법(체계)	3
2. 프로젝트 수행 과정	7
3. 프로젝트 수행 결과	8
 III. 결론	 14
1. 프로젝트 달성도	14
2. 프로젝트 결과 논의	14
3. 프로젝트 결과의 활용방안	14
4. 프로젝트 결과의 기대효과	14
 IV. 참고문헌	 15

I. 서론

1. 프로젝트의 배경 또는 필요성

현재 10대 후반에서 30대 초반은 개인 SNS에 자신의 일상이나 취미를 업로드하여 여러 사람과 정보를 공유하거나 댓글로 공감을 나누며 친목을 쌓아가고 있다. PC뿐만 아니라 개인 스마트폰으로 언제 어디서든 SNS에 접속하여 게시글을 업로드하거나 댓글과 공감을 표현할 수 있다.

그 이유로 우리는 MZ 세대들의 많은 관심을 가질 수 있는 블로그 사이트를 제작하기로 했다. 사이트를 만들어본 경험이 아직 없어 기존에 이미 존재하는 블로그 사이트와 SNS를 참고했다. 회원가입을 하면 로그인이 가능하고 로그인 후에는 블로그 메인 화면을 보거나 게시글 작성, 댓글 작성, 게시글 리스트 보기, 프로필 사진 변경, 사진 첨부, 게시글 검색, 로그아웃이 가능하다.

우리 조는 프로젝트 대상에서 기존 문제점을 찾기보단 실제로 구현할 수 있을지, 우리 조원들의 능력에 맞게 프로젝트를 구상해야 한다고 생각했기 때문에 사이트를 제작하면서 중간중간 아쉬운 부분이 있을 때마다 새로운 기능도 추가하기로 했다. 또, 조원들은 사이트를 제대로 만들어본 경험이 없었기 때문에, 정말로 실현 가능한 사이트인지 한참을 고민한 후, 대중적이며 쉽게 접근할 수 있으며 직접 만들기에 가장 적합한 블로그 사이트를 구현하기로 하였다.

2. 프로젝트의 목표

① 회원 관리

: 사용자가 회원가입 및 회원 탈퇴가 가능하게 하고 회원 등록 시 로그인 및 로그아웃을 할 수 있도록 한다. 사용자 본인의 프로필에 해당하는 닉네임, 사진 변경이 가능하게 하며 본인의 개인 블로그 주소를 생성한다.

회원 관리 DB를 생성하여 회원 등록 시에는 저장되면 회원 탈퇴 시에는 해당 DB에서 삭제되어 개인 정보를 보호해준다.

다른 사용자와 이웃을 맺을 수 있는 이웃 추가 기능도 만들어 전체적으로 공개하는 게시글과 이웃만 확인할 수 있는 게시글을 구별하며 이웃끼리 쪽지도 주고받을 수 있게 한다.

② 게시글 기능

: 로그인 후 개인 블로그에 게시글 작성이 가능하게 한다. 게시글 작성에는 제목과 내용, 파일을 업로드할 수 있도록 하고 업로드 시에는 게시글에 작성 날짜가 자동으로 저장한다. 작성된 게시글은 본인의 블로그에서 확인할 수 있고, 내용 수정 및 삭제를 할 수 있게 한다.

상대방의 블로그에 접속 시 그 사람의 게시글 목록과 게시글을 확인할 수 있게 한다. 상대방의 게시글에 '좋아요'를 누를 수 있는 기능과 조회 수 기능도 추가한다.

게시글 관리 DB를 생성하여 게시글 등록 시에는 전체적인 내용과 닉네임이 저장되며 게시글 고유 숫자와 '좋아요'가 자동으로 등록되게 한다.

③ 댓글 기능

: 게시물 하단에 댓글이 보이도록 한다. 로그인한 상태로 댓글을 입력하면 자동으로 작성자, 댓글 내용, 작성 날짜가 업로드되게 한다. 게시물과 마찬가지로 '좋아요' 기능을 추가하고 댓글 삭제 및 수정을 할 수 있게 한다. 댓글 밑에 댓글을 달 수 있는 대댓글 기능도 추가한다.

댓글 관리 DB를 만들어 게시물 관리 DB와 같은 기능을 구현해내게 한다.

④ 사용자 알고리즘

: 사용자가 자주 접속한 블로그와 관련된 내용이나 블로그 작성자의 다른 게시물들을 추천해주는 추천 블로그 기능과 랜덤으로 블로그를 방문하여 게시물을 볼 수 있는 기능을 만든다.

검색기능을 구현해서 사용자가 검색했던 내용과 유사한 게시물들을 추천해주는 추천 블로그 기능도 만든다.

II. 프로젝트 수행내용

1. 프로젝트 추진전략 및 방법(체계)

1) 프로젝트 추진 일정 및 방법

(1) 회원 관리

- 로그인 및 회원가입 프론트엔드 구축하기
- 정보 저장하는 DB 'minformation' 생성하여 회원가입 시 정보(아이디, 이름, 닉네임, 비밀번호, 전화번호, 이메일)를 저장하고 만약, 비밀번호와 비밀번호 확인의 문자열이 일치하지 않거나, 이미 DB에 등록되어있는 아이디와 중복되었을 경우는 가입이 되지 않게 한다.
- 로그인 상태면 저장된 정보를 session_start()로 세션을 유지하고 로그아웃하면 세션이 종료되게 만든다.
- [프로필 변경]을 누르면 기본 프로필을 사용자가 원하는 사진과 닉네임으로 변경하게 한다.

(2) 글쓰기 기능 구축

- 글쓰기, 내 블로그, 메인 블로그 프론트엔드 구축하기
- 정보 저장 DB 'post' 생성해 글 작성 시 정보(게시글 고유 번호, 제목, 내용, 닉네임, 등록 시간, 조회 수)를 저장한다. 만약, 글의 제목, 내용 중 하나라도 적지 않으면 글을 생성하지 못하도록 한다.
- 해당 게시글을 클릭할 때마다 조회 수가 자동으로 1씩 증가하게 한다.

(3) 댓글 기능 구축

- 댓글 프론트엔드 구축하기
- 정보 저장 DB 'coment' 생성해 댓글 작성 시 정보(고유 번호, 닉네임, 내용, 작성 날짜, 좋아요) 저장한다. 만약, 내용이 없다면 댓글을 작성하지 못하도록 한다.
- 해당 게시글에 해당 댓글만 보이도록 한다.

(4) 사용자 알고리즘

- 메인 블로그에 검색기능을 도입하고 이웃 블로그나 새 글을 랜덤으로 게시글을 추천해서 보여준다.
- 내 블로그는 현재 로그인 계정의 작성자가 쓴 글만 보이도록 설정한다.

[표 1] 프로젝트 추진 일정

No	프로젝트 활동 내용	추진 일정															기간 (주)	추진 방법
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	회원 관리																3	교수님 강의를 듣고 구현 프로젝트에 활용되는 DB 생성하기, 강의자료 참고
2	게시글 기능																4	인터넷 검색, 교재 내용 읽기, 교수님 강의자료 참고
3	댓글 기능																5	인터넷 검색
4	사용자 알고리즘																4	교재 참고, 인터넷 검색, 강의자료 참고

2) 프로젝트 추진 체계

(0) DB 및 서버 구축

- 본 프로젝트에선 'webblog'의 DB를 생성하고 DB 안에 3개의 테이블을 구축한다. 로그인 및 회원가입의 데이터를 저장하는 'minformation'과 글의 내용을 저장하는 데이터베이스인 'post', 마지막으로 댓글을 사용자에게 보여주기 위해 작성된 데이터를 'coment'에 저장한다. 교수님께서 올려주신 강의를 통해 하나의 DB 안에 많은 테이블을 생성하는 방법을 익혀 DB를 구축했다.

서버는 하나의 컴퓨터에서 프로젝트를 진행하기 때문에 한 컴퓨터의 IP주소를 얻어 서버를 구축하면 외부에서 접근 가능해 편리하게 접속할 수 있다.

(1) 회원 관리

- 회원가입에는 아이디, 비밀번호, 비밀번호 확인, 닉네임, 전화번호, 이메일의 DB가 필요하다. 회원가입 정보를 얻기 위해 사용자가 입력할 수 있게 눈에 보이는 [signup.html]을 만들고 사용자가 정보를 입력한 뒤 회원 등록 버튼을 누르면 입력한 정보를 [signup.php]을 통해 (1)의 'minformation'에 저장된다. 단, 비밀번호와 비밀번호 확인란에 입력한 내용이 서로 다르거나 이미 사용 중인 아이디를 입력한다면 회원가입이 불가하며 저장되지 않는다.

- 로그인은 회원 등록 후 저장된 database에 입력되어 있는지 없는지 POST 방식으로 값을 전달받아 'minformation'에 등록된 아이디와 비밀번호를 입력하면 메인 블로그 화면으로 이동하고 둘 중 하나라도 DB에 저장된 값이 아니라면 경고창을 띄우고 다시 입력하도록 로그인 화면을 다시 보여준다. 로그인 성공 시 세션을 유지하여 현재 로그인 중인 사용자의 정보를 저장한다.

- [프로필 편집]은 회원이 기본 사진이나 현재 저장된 프로필 사진을 변경할 수 있도록 만든 기능이다. 사진을 업로드 받으면 로그인 중인 사용자의 닉네임으로 사진이 img 폴더에 저장된다. 사진을 불러올 때는 세션 유지 중인 사용자의 닉네임에 따라 img를 불러온다.

(2) 게시글 기능

- 글 작성 시 제목과 내용, 닉네임, 작성 시간, 조회 수, 사진 업로드가 필요하다. 글을 작성할 수 있도록 구축하고 사용자가 '작성하기' 버튼을 누르면 [webpost.php]을 통해 입력한 값을 DB에 저장한다. 화면에 저장한 내용을 보여주기 위한 게시판을 작성하고 사용자가 입력한 순서에 따라 고유 게시글 번호가 부여되는데 이 고유 번호로 글의 제목을 클릭하면 고유 번호의 DB 내용이 [view.php]에 의해 사용자의 눈에 보이게 된다. 그리고 클릭하면 조회 수가 자동으로 1씩 증가하게 된다. 단, 제목과 내용을 반드시 입력해야 글을 작성할 수 있다. 그렇지 않다면 'post' 테이블에 값이 저장되지 않는다.
- 사진 업로드할 때 jpg, bmp, gif, png 파일로만 업로드 가능하다. 사진 업로드 시 {게시글 번호+사용자 닉네임} 이름으로 사진이 img 폴더에 저장된다. 그렇게 n번째 게시글을 클릭하면 (n+사용자닉네임)으로 이미지 파일이 불러와져 이미지가 게시글에 보이게 된다.




(3) 댓글 기능

- 댓글 작성에는 댓글 작성자, 댓글 내용, 작성 시간이 필요하다. 댓글을 작성하기 위한 프론트엔드를 구축하고 댓글을 작성할 때는 댓글 내용이 필수적으로 들어가야 댓글을 달 수 있다. 내용을 입력하고 '작성하기' 버튼을 누르면 현재 로그인된 회원의 닉네임이 댓글의 작성자가 되어 댓글이 사용자에게 보이게 된다.

(4) 사용자 알고리즘

- 사용자가 찾고 싶은 키워드를 검색하면 해당하는 글을 화면에 보여주는 검색기능을 도입하였다. 이때, 검색 필터에 따라 연관된 게시글을 보여준다.
- 게시글 작성 시 등록한 이미지 파일이 있다면 추천 블로그 및 이웃 새 글에서 해당 이미지가 보이고 등록한 이미지가 없다면 기본 구글링 로고를 보여준다.
- '추천 블로그'와 '이웃 새글'은 현재 로그인된 계정을 제외한 나머지 계정의 게시글을 랜덤으로 보여준다.

[표 2] 프로젝트 추진 체계

성명 / 학번	수행내용	역할
 <div> <div>신민주</div> <div>20213067</div> </div>	<ul style="list-style-type: none"> HeidiSQL을 통해 전체 DB 구축함 로그인 및 회원가입 프론트엔드 html, css, js 사용해 구축함 로그아웃 백엔드 구축함 글쓰기 백엔드 구축하여 DB 값을 저장하고 프론트엔드와 연결해 사용자 눈에 보이도록 웹 구축함 내 블로그에서 현재 로그인 계정의 사용자가 쓴 글만 나오도록 함 로그인 후 블로그 메인화면의 백엔드 구축 (추천 블로그, 이웃 새글 나타내기) 댓글 작성 백엔드 구축하여 댓글 작성 후 DB에 값을 저장시킴 프로젝트 보고서 요약본, 본문 작성 	팀장
 <div> <div>우가현</div> <div>20213062</div> </div>	<ul style="list-style-type: none"> 프로젝트에 사용되는 DB를 구축함 로그인 및 회원가입 백엔드 php를 구축함 회원 정보를 DB에 저장되게 구축함 게시글 작성 내에 파일을 첨부할 수 있게 함 블로그 메인화면 내 블로그에서 자신의 닉네임을 보이게 함 게시글 작성 백엔드 구축 프로젝트 보고서 결론 작성 	팀원
 <div> <div>성채은</div> <div>20213014</div> </div>	<ul style="list-style-type: none"> 블로그 메인화면, 게시글 작성, 게시글, 프로필 편집, 검색 프론트엔드 html, css 사용해 구축함 검색, 프로필 편집, 게시글 작성시 사진 첨부 백엔드 구축함 사용자마다 개인 블로그 화면 띄우기, 본인 블로그 접속시에만 글쓰기 버튼 띄우기, 회원가입시 아이디 중복이면 회원가입 불가하도록 함. 로그인시 세션 유지 프로젝트 보고서 서론 작성 	팀원

2. 프로젝트 수행 과정

하나의 웹사이트를 구현하기 위해 IP주소로 서버를 구축한다. 프로젝트에 활용되는 데이터 리스트는 크게 회원 정보, 게시글, 댓글 DB를 생성해야 한다. 회원 정보 DB는 아이디, 비밀번호(확인), 이름, 닉네임, 전화번호, 이메일로 구축하여 기본적인 회원 정보를 입력받는다. 게시글 DB는 현재 로그인한 작성자의 닉네임, 게시글 작성 시간, 제목, 내용, 조회 수, 식별코드로 구축하여 사용자가 게시글을 작성했을 시에 정보가 입력된다. 식별코드는 게시글 작성 순서로 부여된다. 마지막으로 댓글 DB는 댓글 내용, 현재 로그인된 작성자의 닉네임, 댓글 작성 시간, 댓글 식별코드를 구축하여 사용자가 댓글을 작성하면 식별코드를 통해 게시글마다 댓글을 작성할 수 있게 만든다. 그리고, 세부적으로 프로젝트 수행과정을 아래에서 살펴보려고 한다.

① 회원 관리

- 회원가입 시 비밀번호와 비밀번호 확인의 내용이 서로 달라도 회원가입이 되는 문제점이 일어나 if문을 사용하여 비밀번호와 비밀번호 확인에 입력한 문자가 다르면 회원 정보가 DB에 저장될 수 없도록 변경한다. 또한 중복된 아이디를 입력하거나 비어있는 내용이 있다면 DB에 저장될 수 없도록 변경한다.
- 회원 정보 DB에 저장된 아이디와 비밀번호가 아니라면 로그인 성공을 할 수 없게 만들고 로그인하면 세션을 부여해 모든 곳에서 사용할 수 있도록 만들고 로그아웃을 하게 되면 현재 세션을 종료시킨다.
- [프로필 편집]은 회원의 프로필 사진을 변경할 수 있도록 만든 기능이다. 사진을 업로드 받으면 로그인된 사용자의 닉네임으로 이미지를 저장시켜 불러온다.

② 게시글 기능

- 게시글마다 코드를 부여해 각자의 글이 링크를 통해 읽어올 수 있게 만들었다.
- 내 블로그에서는 게시글이 현재 로그인된 계정만 보이도록 만들기 위해 세션을 시작하고 게시글 DB에 저장된 닉네임 중 현재 로그인된 계정과 같다면 게시글이 보이도록 하며 게시글 작성 버튼도 보이도록 만들었다.
- 파일 업로드로 게시글 작성 시 해당 글에서 사진을 볼 수 있도록 만들었다. {해당 게시글의 번호 + 작성자 닉네임 이름}으로 이미지 파일 저장하고 불러온다.

③ 댓글 기능

- 게시글을 선택해 댓글을 작성하는 부분을 만들고 댓글마다 고유의 코드를 부여해 코드에 따라 댓글 작성한 게시글 밑 댓글 목록에 댓글이 보이도록 만들었다.
- 댓글 작성자는 현재 로그인된 계정의 닉네임을 DB에 저장시켜 만들었다.

④ 사용자 알고리즘

- 사용자가 찾고 싶은 키워드를 검색하면 해당하는 글을 화면에 보여주는 검색기능을 도입하였다. 이때, 검색 필터에 따라 게시글을 보여준다.
- 게시글 작성 시 등록된 이미지 파일이 있다면 추천 블로그 및 이웃 새 글에서 해당 이미지가 보이고 등록된 이미지가 없다면 기본 구글링 로고를 보여준다.
- '추천 블로그'와 '이웃 새 글'은 현재 로그인된 계정을 제외한 나머지 계정의 게시글을 랜덤으로 보여준다.

3. 프로젝트 수행 결과

- 프로젝트 DB

webblog DB				comentDB																																																																																				
<div><div><div></div><div></div><div></div></div><div>webblog</div><div>48.0 KiB</div></div>				<table><tr><th>#</th><th>이름</th><th>데이터 유형</th><th>길이/설정</th><th>부호 없...</th></tr><tr><td>1</td><td>NUM</td><td>INT</td><td>10</td><td><input type="checkbox"/></td></tr><tr><td>2</td><td>NICKNAME</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>3</td><td>CONTENT</td><td>TEXT</td><td></td><td><input type="checkbox"/></td></tr><tr><td>4</td><td>DATE</td><td>DATE</td><td></td><td><input type="checkbox"/></td></tr><tr><td>5</td><td>CHECKNUM</td><td>INT</td><td>11</td><td><input checked="" type="checkbox"/></td></tr></table>					#	이름	데이터 유형	길이/설정	부호 없...	1	NUM	INT	10	<input type="checkbox"/>	2	NICKNAME	VARCHAR	50	<input type="checkbox"/>	3	CONTENT	TEXT		<input type="checkbox"/>	4	DATE	DATE		<input type="checkbox"/>	5	CHECKNUM	INT	11	<input checked="" type="checkbox"/>																																																		
#	이름	데이터 유형	길이/설정	부호 없...																																																																																				
1	NUM	INT	10	<input type="checkbox"/>																																																																																				
2	NICKNAME	VARCHAR	50	<input type="checkbox"/>																																																																																				
3	CONTENT	TEXT		<input type="checkbox"/>																																																																																				
4	DATE	DATE		<input type="checkbox"/>																																																																																				
5	CHECKNUM	INT	11	<input checked="" type="checkbox"/>																																																																																				
<div><div><div></div><div></div><div></div></div><div>coment</div><div>16.0 KiB</div></div>																																																																																								
<div><div><div></div><div></div><div></div></div><div>mininformation</div><div>16.0 KiB</div></div>																																																																																								
<div><div><div></div><div></div><div></div></div><div>post</div><div>16.0 KiB</div></div>																																																																																								
mininformationDB				postDB																																																																																				
열: <div><div>+</div>추가</div> <div><div>×</div>제거</div> <div><div>▲</div>위로</div> <div><div>▼</div>아래로</div>																																																																																								
<table><tr><th>#</th><th>이름</th><th>데이터 유형</th><th>길이/설정</th><th>부호 없...</th></tr><tr><td>1</td><td>ID</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>2</td><td>NAME</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>3</td><td>NICKNAME</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>4</td><td>PW</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>5</td><td>PWh</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>6</td><td>PHONENUMB...</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>7</td><td>EMAIL</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr></table>				#	이름	데이터 유형	길이/설정	부호 없...	1	ID	VARCHAR	50	<input type="checkbox"/>	2	NAME	VARCHAR	50	<input type="checkbox"/>	3	NICKNAME	VARCHAR	50	<input type="checkbox"/>	4	PW	VARCHAR	50	<input type="checkbox"/>	5	PWh	VARCHAR	50	<input type="checkbox"/>	6	PHONENUMB...	VARCHAR	50	<input type="checkbox"/>	7	EMAIL	VARCHAR	50	<input type="checkbox"/>	<table><tr><th>#</th><th>이름</th><th>데이터 유형</th><th>길이/설정</th><th>부호 없...</th></tr><tr><td>1</td><td>NUM</td><td>INT</td><td>11</td><td><input type="checkbox"/></td></tr><tr><td>2</td><td>TITLE</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>3</td><td>CONTENT</td><td>TEXT</td><td></td><td><input type="checkbox"/></td></tr><tr><td>4</td><td>NICKNAME</td><td>VARCHAR</td><td>50</td><td><input type="checkbox"/></td></tr><tr><td>5</td><td>DATE</td><td>DATE</td><td></td><td><input type="checkbox"/></td></tr><tr><td>6</td><td>HIT</td><td>INT</td><td>11</td><td><input type="checkbox"/></td></tr><tr><td>7</td><td>FILE</td><td>TEXT</td><td></td><td><input type="checkbox"/></td></tr></table>					#	이름	데이터 유형	길이/설정	부호 없...	1	NUM	INT	11	<input type="checkbox"/>	2	TITLE	VARCHAR	50	<input type="checkbox"/>	3	CONTENT	TEXT		<input type="checkbox"/>	4	NICKNAME	VARCHAR	50	<input type="checkbox"/>	5	DATE	DATE		<input type="checkbox"/>	6	HIT	INT	11	<input type="checkbox"/>	7	FILE	TEXT		<input type="checkbox"/>
#	이름	데이터 유형	길이/설정	부호 없...																																																																																				
1	ID	VARCHAR	50	<input type="checkbox"/>																																																																																				
2	NAME	VARCHAR	50	<input type="checkbox"/>																																																																																				
3	NICKNAME	VARCHAR	50	<input type="checkbox"/>																																																																																				
4	PW	VARCHAR	50	<input type="checkbox"/>																																																																																				
5	PWh	VARCHAR	50	<input type="checkbox"/>																																																																																				
6	PHONENUMB...	VARCHAR	50	<input type="checkbox"/>																																																																																				
7	EMAIL	VARCHAR	50	<input type="checkbox"/>																																																																																				
#	이름	데이터 유형	길이/설정	부호 없...																																																																																				
1	NUM	INT	11	<input type="checkbox"/>																																																																																				
2	TITLE	VARCHAR	50	<input type="checkbox"/>																																																																																				
3	CONTENT	TEXT		<input type="checkbox"/>																																																																																				
4	NICKNAME	VARCHAR	50	<input type="checkbox"/>																																																																																				
5	DATE	DATE		<input type="checkbox"/>																																																																																				
6	HIT	INT	11	<input type="checkbox"/>																																																																																				
7	FILE	TEXT		<input type="checkbox"/>																																																																																				

- 프로젝트 DB 내용

mininformationDB

webblog.mininformation: 3 행 (대략적)

ID	NAME	NICKNAME	PW	PWH	PHONENUMBER	EMAIL
20115	문상민	구글링	tkda0000	tkda0000	-5899-8754	tkda0@naver.com
2021	김두물	오징	emkf1234	emkf1234	-4864-1111	askdfo@naver.com
20211	김다원	곰돌이	xodhs5555	xodhs5555	-1235-1111	qwertr@google.com

```

26 SHOW ENGINES;
27 SHOW COLLATION;
28 SHOW CREATE TABLE `webblog`.`mininformation`;
29 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='webblog' AND TABLE_NAME='mininformation' LIMIT 1000;
30 SELECT * FROM `webblog`.`mininformation` LIMIT 1000;

```

posttDB

webblog.post: 8 행 (대략적)

NUM	TITLE	CONTENT	NICKNAME	DATE	HIT
2	오징	난 오징이야	오징	2022-12-11	6
3	오돌은 출항가?	곰부자	오징	2022-12-11	2
4	물자	마하고 물자?	구글링	2022-12-11	4
5	내 고양이 봐	귀염자?	구글링	2022-12-11	4
6	갯 저시절	안녕 반가워!	곰돌이	2022-12-11	1
7	오돌 뭐야?	취할지 추천해..	곰돌이	2022-12-11	0
8	오돌 뭐야?	나는 논다(www)	오징	2022-12-11	1
9	영복해!	석영도 영복해!	오징	2022-12-11	1

```

33 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='webblog' AND TABLE_NAME='post' AND REFERENCED_TABLE_NAME IS NOT NULL;
34 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='webblog' AND TABLE_NAME='post' AND REFERENCED_TABLE_NAME IS NOT NULL;
35 SHOW CREATE TABLE `webblog`.`post`;
36 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='webblog' AND TABLE_NAME='post' LIMIT 1000;
37 SELECT 'NUM', 'TITLE', LEFT('CONTENT', 250), 'NICKNAME', 'DATE', 'HIT' FROM `webblog`.`post` LIMIT 1000;

```

comentDB

webblog.coment: 5 행 (대략적)

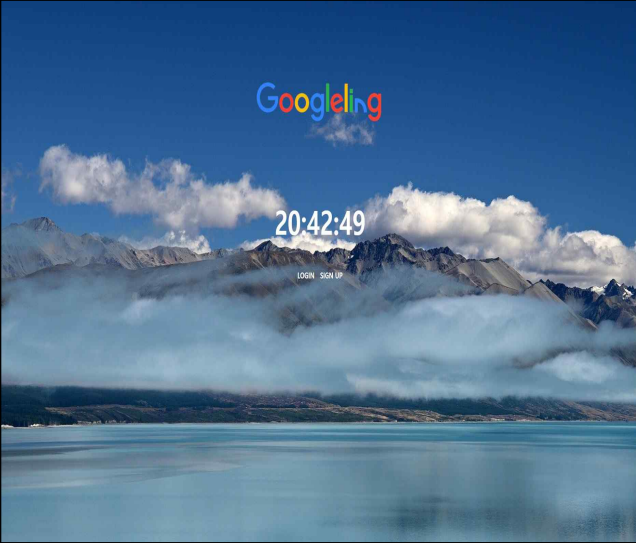
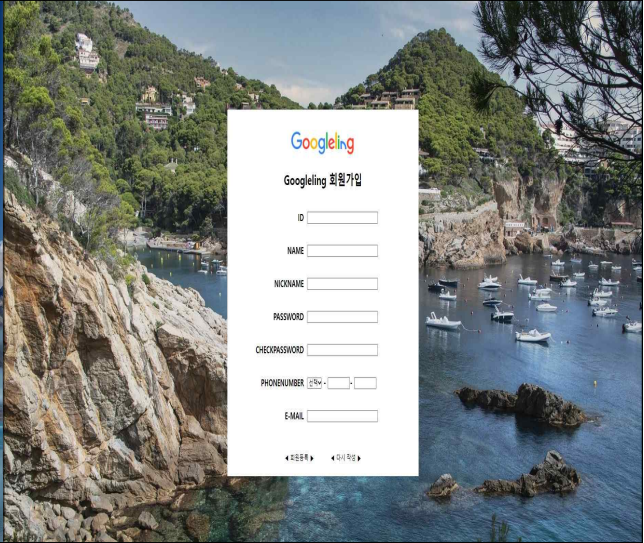
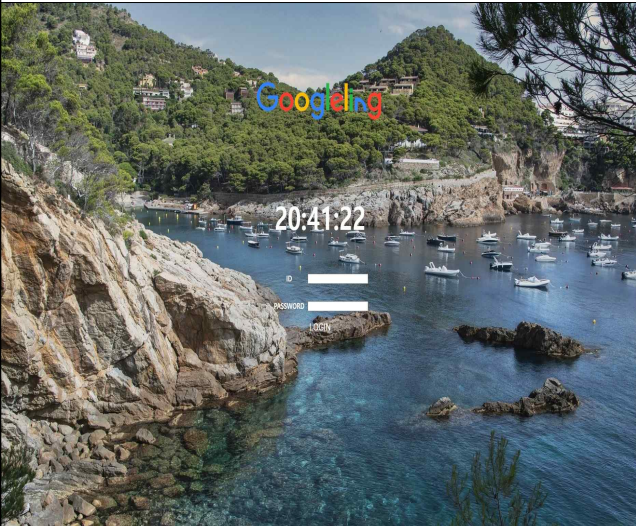
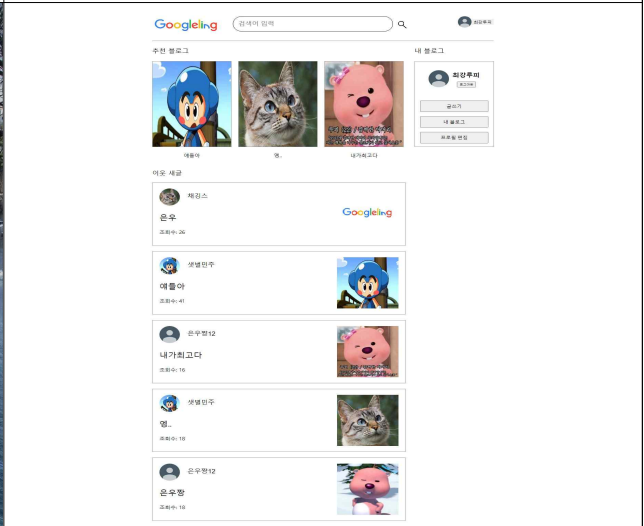
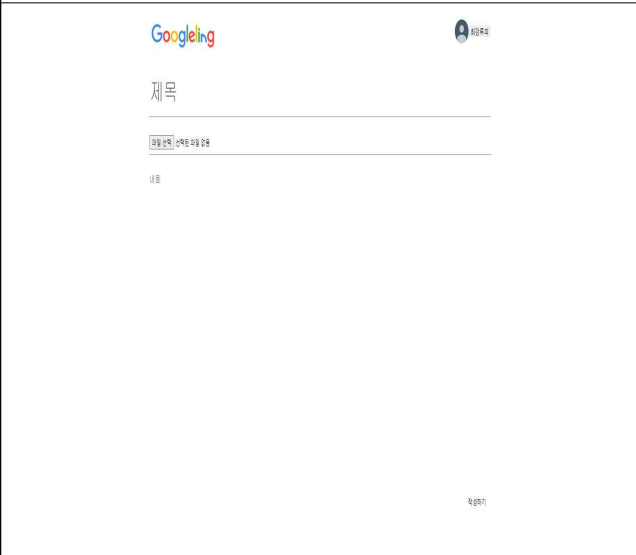

NUM	NICKNAME	CONTENT	DATE	CHECKNUM
1	구글링	오징이 안녕	2022-12-11	2
2	구글링	곰부? 싫어....	2022-12-11	3
3	곰돌이	곰부해..	2022-12-11	4
4	곰돌이	우와 귀엽다!!!!	2022-12-11	5
5	곰돌이	안녕!!	2022-12-11	2

```

40 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='webblog' AND TABLE_NAME='coment' AND REFERENCED_TABLE_NAME IS NOT NULL;
41 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='webblog' AND TABLE_NAME='coment' AND REFERENCED_TABLE_NAME IS NOT NULL;
42 SHOW CREATE TABLE `webblog`.`coment`;
43 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='webblog' AND TABLE_NAME='coment' LIMIT 1000;
44 SELECT * FROM `webblog`.`coment` LIMIT 1000;

```


- 프로젝트

<p>메인 홈페이지</p> 	<p>회원가입</p> 
<p>로그인</p> 	<p>메인 블로그</p> 
<p>게시글 작성(파일 첨부 X) -> 작성된 게시글 보여주기</p>	
	

게시글 작성(파일 첨부 ○) -> 작성된 게시글 보여주기

내 블로그

댓글 작성

추천 블로그 제목 선택 -> 해당 게시글 보여주기

이웃 새글 제목 선택 -> 해당 게시글 보여주기

이웃 새글

셋별민주
엔
 조회수: 18

은우짱12
내가최고다
 조회수: 16

셋별민주
애들아 클릭시
 조회수: 42

Googleling 최강루피

셋별민주님의 블로그 목록

2022-12-11

애들아

나프사바뛰어
귀엽정ㅋㅋ

셋별민주 조회수: 42

댓글 목록

댓글을 입력하세요 (최대 100글자)

작성하기

이웃 새글 작성자 선택 -> 해당 글 작성자의 블로그 보여주기

이웃 새글

채깅스 클릭시
 은우
 조회수: 26

셋별민주
애들아
 조회수: 43

은우짱12
 내가최고다

Googleling 최강루피

채깅스님의 블로그

제목	작성자	날짜	조회수
은우	채깅스	2022-12-11	26

프로필 사진 변경

변경된 프로필 사진

Googleling 최강루피

로그아웃

프로필 수정

프로필 사진을 변경하고 새로그인을 누르세요.

파일 선택 선택된 파일 없음

이미지 권장 비율 1:1 변경하기

Googleling 최강루피

로그아웃

프로필 수정

프로필 사진을 변경하고 새로그인을 누르세요.

파일 선택 선택된 파일 없음

이미지 권장 비율 1:1 변경하기

메인 블로그 키워드 검색 -> 키워드에 따라 검색 결과 보여주기



전체
제목&내용
내용
닉네임
날짜

은우 의 검색결과

제목	작성자	날짜	조회수
은우짱	은우짱12	2022-12-11	19
내가최고다	은우짱12	2022-12-11	17
은우	재깁스	2022-12-11	26



전체
제목&내용
내용
닉네임
날짜

은우 의 검색결과

제목	작성자	날짜	조회수
은우짱	은우짱12	2022-12-11	19
내가최고다	은우짱12	2022-12-11	17


로그아웃 (버튼 클릭시) -> 로그인 화면으로 이동

게시글 이미지 첨부x, 기본 프로필 사진일 때

127.0.0.1 내용:


로그아웃 완료!


확인




은우짱12


추천 블로그


구글링조


과연


은우

내 블로그



은우짱12

로그아웃

글쓰기

내 블로그

프로필 편집

III. 결론

1. 프로젝트 달성도(공학계열만 해당)

[표 3] 설계 목표의 중요도 및 달성도

목표	중요도(%)	달성도(%)	수행내용
DB 및 서버 구축	25	25	회원가입 정보와 게시글 정보, 댓글 정보를 저장
로그인 및 회원가입 기능 구축	25	25	DB에 필요한 회원 정보들을 회원가입을 통해 저장하고 저장된 정보와 비교하여 로그인 수행
게시글 기능 구축	25	25	제목, 내용, 사진을 DB에 저장하며 사용자에게 보일 수 있게 함
댓글 기능 구축	25	25	다른 사용자가 올린 게시글에 댓글을 작성하여 DB에 저장
블로그 메인 화면	10	10	검색 기능, 추천 블로그, 이웃 새글을 보일 수 있게 함
합계	100	100	

2. 프로젝트 결과 논의

로그인 화면을 통해서 로그인과 회원가입을 진행하고 로그인 성공 시 블로그 메인 화면에 접속하게 한다. 블로그 메인 화면에는 검색기능과 DB에 저장된 블로그를 랜덤으로 3개의 게시글을 추천 블로그에 보이게 하며 이웃 새 글에는 자신이 작성한 게시글을 제외한 게시글들을 보이게 한다. 또한 내 블로그에서는 글쓰기와 내 블로그 버튼을 만들어 관련 기능을 수행하게 한다.

시간이 더 주어진다면 게시글과 댓글에 '좋아요' 기능을 구현하여 블로그 메인 화면에 '좋아요' 순으로 추천 블로그를 띄우고 댓글도 '좋아요' 순으로 띄웠을 것이다. 또한 방명록 페이지를 만들어 다른 사람의 블로그에 게시글을 작성할 수 있는 기능과 이웃 추가 기능을 만들어 자신의 블로그에서 이웃 블로그로 바로 접속할 수 있는 기능, 블로그 메인 화면의 이웃 새 글 목록에 본인이 추가한 이웃들만 뜨도록 했을 것이다.

3. 프로젝트 결과의 활용방안

구현한 프로젝트는 주된 사용자를 10대 ~ 30대로 목표하여 자신이 보낸 일과나 홍보하고 싶은 것들을 자신의 블로그에 게시글을 업로드하여 다른 사용자와 공유하게 한다. 사용자들은 상대방의 일상에 공감하거나 홍보 게시글을 통해 또 다른 정보를 얻을 수 있다. 지금 흔히 사용되고 있는 Instagram, 네이버 블로그와 같은 SNS로 활용될 수 있다.

4. 프로젝트 결과의 기대성과

HTML, SQL, PHP, JS, CSS 등을 이용하여 하나의 블로그 사이트를 구축해 내어 여러 사람과 일상을 공유하고 이익을 창출할 수 있는 홍보, 여러 정보를 얻을 수 있다.

기술적 측면	블로그에서 사용되는 여러 기능을 블로그 외에 다른 SNS에서도 똑같이 사용 가능하며 블로그 사용자가 불편함을 느끼는 부분이나 원하는 다른 기능들을 지속해서 추가하여 더 나은 사이트로 구축해 낼 수 있다.
경제적 측면	블로그 사이트에서 자신이 이익을 취할 수 있는 것들을 게시물에 업로드 하여 수익을 보거나 블로그 자체에서 광고를 띄어 광고비를 통해 수입을 창출할 수 있다. 그뿐만 아니라 사용자 유입 증가를 통해서도 수익이 창출된다.
사회적 측면	블로그에서 올린 정보로 여러 기관에 도움이 될 수 있으며 모금과 봉사를 위한 모집도 게시글을 통해 진행하여 사회적으로 도움이 필요한 곳에 함께 기부하거나 봉사활동을 진행하여 도움을 준다.

IV. 참고문헌

- tistory, 워누. "[PHP] 이미지 파일 업로드하기!", Internet: <https://seoneu.tistory.com/29> . 2021.02.14.
- 코딩팩토리, "PHP / Session / 로그인,로그아웃,로그인 상태에서만 보이는 콘텐츠 만들기", Internet: <https://www.codingfactory.net/11724>
- tistory, 이아임. "[PHP] textarea 엔터 값 그대로 가져오기", Internet: <https://hyoseung930.tistory.com/12> . 2017.11.27.
- HONG:PAGE, HONG. "[php] strpos(), strrpos() 특정 문자열 검색하기", Internet: <https://hongpage.kr/46>
- tistory, zxchr. "[PHP] MySQL 행 개수 구하기, mysqli_num_rows()" , Internet: <https://zxchr.tistory.com/215> . 2021.09.06.
- tistory, 염초코. "[웹개발]" 게시글 검색 기능 (기간 설정 포함), Internet: <https://choco4study.tistory.com/m/65> . 2021.11.15.