태그를 지정해주고 크기를 안잡아주면 잡히지 않음(높이 넓이 모두 지정해줘야 실체가 잡힘)

객체 탐색기로 엘리먼트가 어떻게 되어있는지 확인해가면서 보면 좋다.

설계할 때 경우의 수를 줄일 수 있는 방식으로 쉽게 짤 수 있는 방식을 생각해보기

기능별로 묶으면 공용으로 사용되는 파일의 구분이 어려워짐

차라리 html, css, js 별로 나누고 그 안에서 파일을 기능적으로 구분

프로젝트의 폴더 구조

1. 프로토콜
   1. 몰라도 개발하는 것 자체에는 문제가 없음(CS 지식)
   2. CS 공부하는 이유? (더 체계적인 개발을 위해)

정의

* 인터넷 통신 표준  
  표준이라는 것의 의미는 : (모든 사람이 동일한 방법으로 하게하는 수단  
  인터넷은? (뭔가 규칙이 없으면 쓰기 어려운 존재)

종류

* 인터넷 통신 방법 마다 1개의 종류가 정해져 있음
* 개수가 정해져 있지 않음 (새로운 통신 -> 새로운 프로토콜)  
  예시)VoLTE 프로토콜 : 4G 통신 프로토콜

HTTP: 단방향 통신 프로토콜(톰캣 사용 이유)

TCP: 양방향 통신 프로토콜

SSH: 원격 접속 통신 프로토콜

FTP: 파일 전송 통신 프로토콜

왜 알아야 하는가?

* 직접 구현하거나 개발하진 않음
* 다른 개발자가 먼저 해놨고, 우린 그 소프트웨어를 사용함
* 그래도 알고 있어야 하는 이유 – 더 체계적인 개발을 할 수 있음  
  예시) HTTP 대신 https를 써야하는 이유?   
  예시) 왜 웹은 TCP를 안쓰고 HTTP를 쓰는 가? 그럼 TCP는 못쓰는가?  
  (채팅이나 실시간 알림은 TCP를 사용해야만 가능함)

어떤 것을 공부해야하는가?

* 정의, 필요성, 장단점, 예시
* 추가적으로, 포트 번호 정도만 암기해두면 좋음 (프로토콜마다 배정된 포트 번호) (SSH: 22번 포트, http: 80번 포트, 1024번 밑의 포트는 사용이 불가(대부분의 프로토콜) 그 위는 특정 소프트웨어가 지정 )  
  만약 Tomcat을 안쓰고 직접 HTTP를 구현했다면 사용했을 포트는 80  
  - 지금은 Tomdat의 힘을 빌리고 있으므로 808을 사용하고 있는 것

SFTP

* SSH + FTP 합쳐진 프로토콜 (원격 접속 후 파일 전송) (남의 집 에어컨 리모콘 찾기) FTP면 파일질라 오른쪽 화면이 보이지 않는다.

활용

* 파일질라 써서 활용했었음
* 파일질라 불편한 점 (내가 코드를 수정하면, 매번 또 다시 올려줘야 함)
* 코딩 할때만은 또다른 SFTP 소프트웨어가 필요함

SFTP Extension

* VSD 내에 있는 Extension 기능
* SFTP 이기에 파일 전송을 하는 기능
* 다만, 코딩 전용

설명

* 내 컴퓨터에 있는 특정 폴더와, 서버에 있는 특정 폴더를 서로 공유 폴더로 만듦
* 내가 코드를 쓰고 저장을 하면, 실시간으로 해당 내용이 서버에 업로드
* 서버가 중심임 (서버에 있는 것을 따와서 내 컴퓨터에 공유폴더로 만드느 개념) (먼저 만든게 있다면 파일질라로 올려놓고 공유폴더로 가지고 와야함)

사용상 주의사항

* Sftp config을 통해서 설정해 주어야 함(json) 해야함
* Pem만 키로 설정이 가능
* 실시간 반영이 안되므로 터미널에서 작없하면 listall 해줘서 받와야함
* 저장해야만 서버에 반영이 됨(파일을 vscode에서 삭제해도 의미가 없음 -> 터미널로 직접 해야함)
* 파일의 이름,위치 등을 변경해도 반영이 되지 않음(이건 터미널에서 직접 해야함)
* 껏다켜도 해당 폴더 오픈하면 그대로 사용이 가능함(별도의 연결 불필요함)

\*\* 좀 깊게 알아보기 \*\*

1. 데이터베이스

정의

* 데이터 저장소
* 데이터를 보조메모리에 저장함(램은 휘발하므로) (반 영구적으로 저장)

데이터 베이스의 종류

1. Relational Database(RDB) (체계적임, 비유연 )
2. NoSQL Database (NoSQL) (비체계적임, 유연 )

얼마나 체계적인지 / 얼마나 유연한지

유연하다는 것 = 개발자한테 편리한 것 -> 체계성이 더 중요함 (고객한테 정확한 정보를 줘야하니까)

RDB의 종류

1. MySQL : 가장 많이 쓰임, 모든 지표의 기준
2. Oracle: MySQL보다 좋은 성능, 좋은 서비스 / 전면 유료
3. PosrtgreSQL : 주목받는 신흥 강자, MySQL과 비교했을 때 데이터가 많아 질수록 성능이 좋아짐
4. SQLite : 로컬 전용 DB(개인 PC용 DB) (기능이 적은 대신 매우 가벼움)
5. MariaDB : 오늘 쓸 거

데이터베이스 통신 구조

클라 – 웹 – 서버 – {데이터 베이스}  
DB는 서버랑만 통신함 (서버를 통하지 않고는 DB 접속이 불가능) (JS에서 써도 의미없다 – 서버 없어서)

마리아DB

Table 구조

Table 이름, column(세로 축이자 소재목), row(가로 축이자 데이터:줄이 데이터)

SQL

RDB의 가장 중요한 특징

RDB 전용 언어

서버가 RDB에 요청을 보낼 때 SQL로 보냄

모든 RDB는 이 SQL로 동작하기 때문에, 한 번만 배워두면 다른 RDB도 쓰기 쉬움

SQL의 중요성

DB 통신이 소프트웨어에서 가장 동작이 오래걸리는 작업

SQL을 잘 쓰면 시간을 줄일 수 있음

백엔드 개발도 중요하지만, 가장 중요한 것은 SQL을 잘 쓰는 것

\*\* 깊게 공부할 것

설치하기

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

데이터베이스의 설치 구조

* 데이터베이스를 서버에 설치했음
* 데이터베이스 서버라고 부름(데이터베이스는 또다른 작은 서버의 느낌으로 설치가 됨)
* 그러다보니 서버 안에 설치가 되었지만, 마음ㅁ대로 접근할 수는 없고 서버대 서버 통신으로만 대화를 하게 됨
* 보안을 위해서 서버 대 서버로 통신함 (중요한 데이터베이스를 보호하기 위해서)

데이터베이스는 논리적으로 서버로써 설치가 됨

* 그래서 데이터베이스도 독립적인 계정 시스템을 가짐  
  예시) 내가 데이터베이스에 접속하고 싶다면, 데이터베이스의 아이디와 비밀 번호를 입력해야 함

서버 관리자 권한이 가능한 사람(ubuntu) 만 sudo mariadb가 가능함

1. Sudo 명령어는 원래 입력시, 서버의 관리자 계정 비밀번호를 물어본다.
2. Sudo mariadb는 mariadb의 관리자 계정으로 접속하겠다.
3. 고로 sudo mariadb는 서버의 관리자만이 사용할 수 있는 명령어다.(우리는 ubuntu로 동작하고 있기 때문에 뭘해도 상관이 없다)

마리아 DB 계정 만들기

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

(\*\*%는 그룹)

새로운 데이터베이스 만들기



계정에 해당 데이터베이스의 권한 주기



(특권(PRIVILGES ON)을 준다 (web .\* \*은 모든 것)에)

데이터베이스 보이기

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

중간 3개는 중요한 시스템 파일(들어갈 일 없음)

나가기



계정으로 접속하기

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

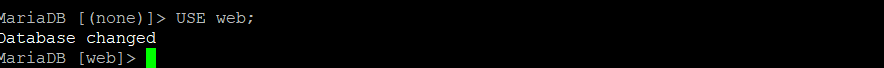
데이터베이스 보이기

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

(계정 삭제, 계정 이름 변경, 데이터베이스 이름 변경, 데이터베이스 삭제 등등… 찾아보라)

사용할 데베 접속



테이블 만들고 확인하기

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

테이블 내용 보기

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

태이블 이름 변경, 컬럼 추가 삭제 등등..

CRUD

* 데이터베이스가 기본적으로 해주는 4개의 기능을 부르는 말
* C : create (값의 생성)



(\*\* ; 안찍고 엔터치면 -> 나오는데 이어쓰기니까 그냥 쓸 것)

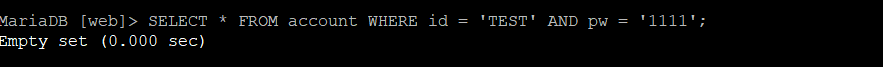
* R : Read (값의 읽기)

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명



* U : Update (값의 수정)

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

(where 사실상 필수)

* D : Delete (값의 삭제)

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

(where 사실상 필수)

각 row 마다 식별할 수 있는 고유값을 붙인 다면 where 문이 매우 간단해질 것이다. table에는 각 row의 고유 값을 붙이는 기능이 존재 -> primary key!

\*\* 테이블 설계가 중요

<이번 주 과제>

의뢰를 맡겼다고 가정

필요한 기능을 말할 것

기능을 보고 적절하게 테이블을 설계하는 것이 목표

기능:

1. 로그인
   1. 계정 만들기
   2. 계정 정보 보기 (마이 계정)
   3. 계정 정보 수정 (비밀번호 변경
   4. 계정 정보 삭제 (탈퇴)
   5. Id 찾기
   6. 비밀 번호 찾기
2. 게시글
3. 게시글 만들기
4. 게시글 보기
5. 게시글 수정
6. 게시글 삭제
7. 게시글 목록
8. + 카테고리
9. 댓글
10. 댓글 생성
11. 댓글 읽기
12. 댓글 수정
13. 댓글 삭제
14. 적절한 자료형 찾아서 적용해주기
15. erd 구조 (pk FK) 적용하기
16. 16개의 sql문 작성해오기
17. 데이터베이스 확인이 가능한 프론트 만들어 오기