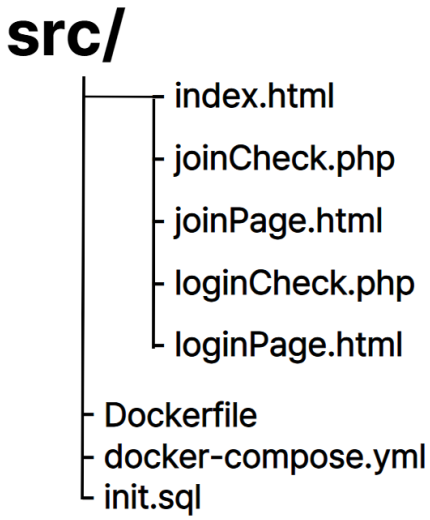
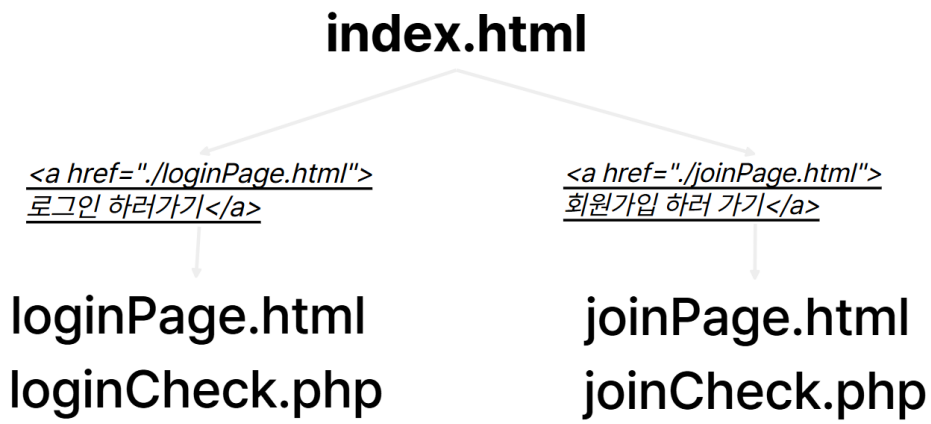
README

1. 구조도



Index.html로 페이지를 시작해서, loginPage.html과 joinPage.html로 하이퍼 링크 연결

각 페이지의 보여지는 부분은 Html 파일로,
정보 저장 등 동적인 부분은 Php 파일로 관리함.



About "Index.html"

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0"
6       <title>Main Page</title>
7   </head>
8   <body>
9       <a href="./loginPage.html">로그인 하러가기</a>
10      <br>
11      <a href="./joinPage.html">회원가입 하러 가기</a>
12  </body>
13  </html>
```

그냥 html의 <a> 태그로 하이퍼링크를 걸어서

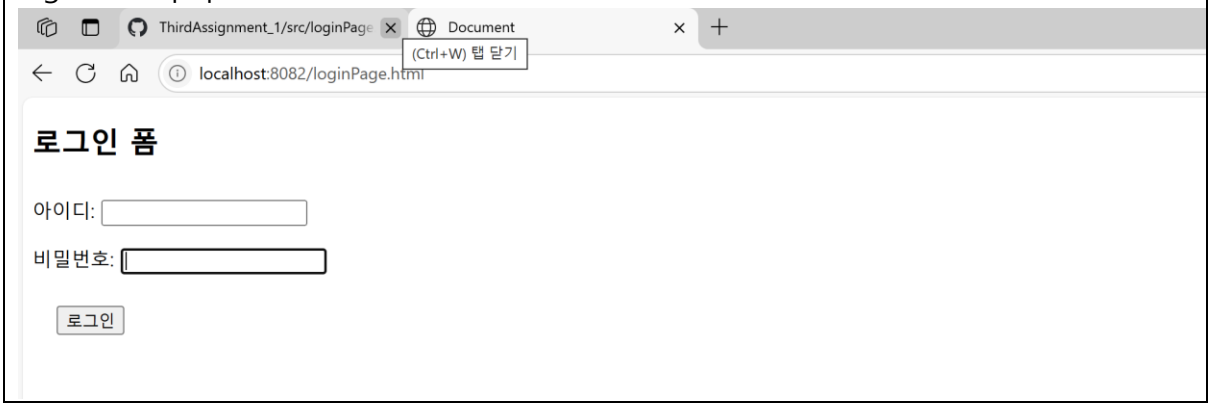
Login, join 페이지로 넘겨줌

About "loginPage.html", "loginCheck.php"

<loginPage.html>

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>LoginPage</title>
</head>
<body>
    <div>
    <h2 style="margin-bottom: 30px;"
    >로그인 폼</h2>
    <p></p>
    <form method="post" action="loginCheck.php">
        <div style="margin-bottom: 15px;">
            <p style="text-align:left; margin-bottom: 5px"
            >아이디: <input type="text" name="userId"></p>
            <p style="text-align: left; margin-bottom: 5px;"
            >비밀번호: <input type="password" name="userPassword"></p>
            <p style="padding: 10px 20px;"
            ><input type="submit" value="로그인"></p>
        </div>
    </form>
    </div>
</body>
</html>
```

아이디 입력 창을 <input type="text"> 태그로 만들었고,
비밀번호 입력 창을 <input type="password"> 태그로 만들었음.
Database에 저장하기 위해 name도 지정해줌.
가독성을 위해 각 태그에 직접 style로 간단하게 디자인함.
<form>태그를 활용해서 <input type="submit"> 버튼을 눌렀을 때,
loginCheck.php 파일에 정보를 보내게끔 설계함.



<loginCheck.php>

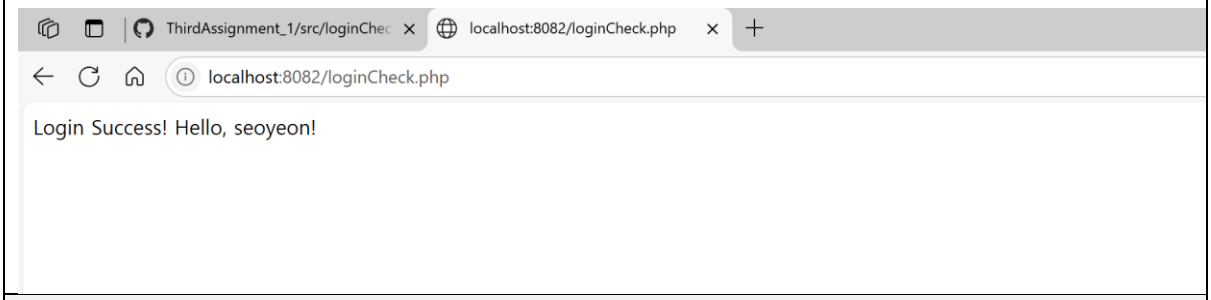
```php
<?php
$conn = new mysqli("db", "root", "root", "testdb");
if ($conn->connect_error) {die("DB connect Fail: " . $conn->connect_error);}

$userId = $_POST['userId'] ?? '';
$userPassword = $_POST['userPassword'] ?? '';

if($userId && $userPassword) {
    $sql = "SELECT * FROM users WHERE userId = ? AND userPassword = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ss", $userId, $userPassword);
    $stmt->execute();
    $result = $stmt->get_result();

    if($result->num_rows === 1) {
        $row = $result->fetch_assoc();
        $userName = $row['userName'];
        echo "Login Success!\nHello, " . htmlspecialchars($userName)."!";
    } else { echo "Failed to Login."; }
}
$conn -> close();
?>
```

loginPage에서 넘긴 정보를 관리하는 php파일

1. $conn라는 이름으로 데이터베이스를 불러옴.
2. $_POST 전역 리스트를 활용해, loginPage.html에서 넘어온 userId, userPassword를 이 파일 내에서 저장.
3. Sql 문법을 담은 문자열을 $sql 변수로 저장해서
4. $conn으로 불러온 데이터 베이스에 prepare로 실행시킴
5. bind_param() 함수를 사용하여 사용자로부터 입력받은 userId와 userPassword를 SQL 문에 바인딩함.
6. execute() 함수로 sql실행 후, 해당 조건에 맞는 결과를 데이터베이스에서 검색
7. get_result()로 결과값을 받아와 num_rows로 일치하는 사용자가 있는지 확인함.
8. 일치하는 사용자가 있으면 fetch_assoc()로 데이터를 배열 형태로 가져오고, 그 중 userName만을 추출하여 로그인 성공 메시지와 함께 출력함.
   (일치하는 사용자가 없을 시 로그인 실패 메시지를 출력)

모두 수행 후 close() 함수를 호출하여 데이터베이스 연결을 닫아줌

➔ 실행 결과



About "joinPage.html", "joinCheck.php"

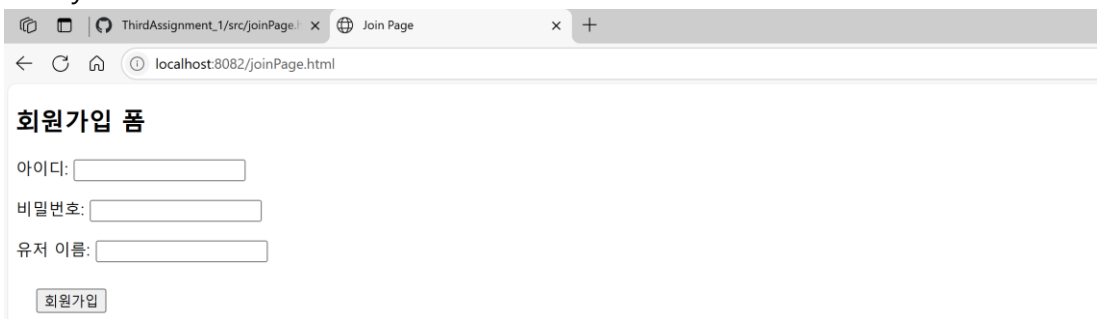
<joinPage.html>

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Join Page</title>
</head>
<body>
    <div class = "center">
        <h2>회원가입 폼</h2>
        <form method ="post" action = "joinCheck.php">
            <p>아이디: <input type="text" name="newId"></p>
            <p>비밀번호: <input type="password" name="newPassword"></p>
            <p>유저 이름: <input type="text" name = "newName"></p>
            <p style="padding: 10px 20px;"
            ><input type="submit" value="회원가입"></p>
        </form>
    </div>
</body>
</html>
```

회원가입 창을 띄우는 역할인 html 파일.

loginPage.html처럼, <form> 태그를 활용해서 post로 입력받고 joinCheck.php로 정보를 넘기는 형식이다.

아이디와 비밀번호, 회원가입버튼은 <input type="~"> 태그를 활용했고, 가독성을 위해 style로 간단하게 정리만 해줬다.



<joinCheck.php>

```
1     <!DOCTYPE html>
2     <html>
3     <head>
4         <meta charset="UTF-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1.0">
6         <title>JoinCheck</title>
7     </head>
8     <body>
9         <h1>Manage</h1>
10        <?php
11        $conn = new mysqli("db", "root", "root", "testdb");
12        if ($conn->connect_error) { die("testDB connect Fail: ".$conn->connect_error);}
13
14        $newId= $_POST['newId'] ?? '';
15        $newPassword= $_POST['newPassword'] ?? '';
16        $newName= $_POST['newName'] ?? '';
```

Html로 Manage라는 글귀 띄우고 (9)

Php를 바로 선언해서, $conn이라는 이름으로 docker의 sql database에 연결함.(10~12)

또한, joinPage로부터 post로 넘겨 받은 아이디 비번 이름을 받기위해 $_POST 리스트에서 찾아서 이 파일 내부에서 쓸 수 있도록 변수로 선언시켜줌.

```
17
18        if ($newId && $newPassword && $newName) {
19            $sql = "INSERT INTO users (userId, userPassword, userName) VALUES (?, ?, ?)";
20            $stmt = $conn->prepare($sql);
21            $stmt->bind_param("sss", $newId, $newPassword, $newName);
22            $stmt->execute();
23
24            echo "Join Success!\n";
25        } else { echo "Join failed." }
26
27        $conn->close();
28        ?>
29        <a href="loginPage.html">Go to Login Page</a>
30    </body>
31    </html>
```

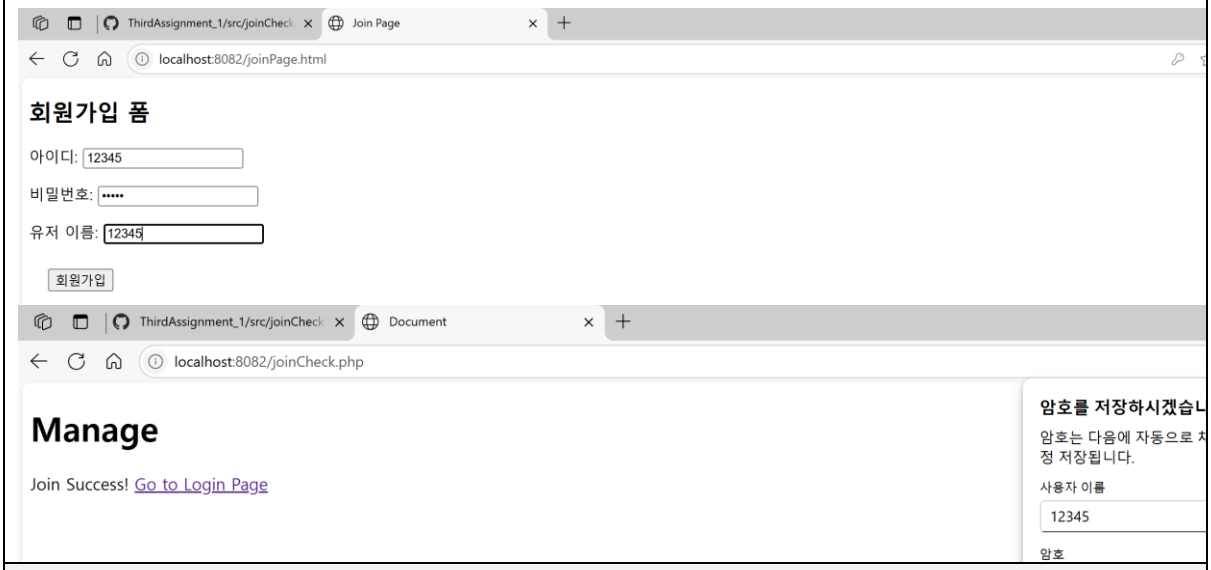
모두 제대로 된 값이 들어가 있다면(17line~) sql을 실행할 준비를 함

1. 실행하고자 하는 sql 명령어를 $sql에 문자열로 저장하고,
2. sql문을 미리 준비해서 $stmt라는 이름으로 저장.
3. 21line에서 $sql 속 ?들에 각 변수들을 binding 해주고,
4. Execute를 선언하며 실행시킴

이렇게 database에 저장시키고, echo를 활용해서 화면에 성공 실패 여부를 출력함.

바로 로그인 페이지로 갈 수 있는 하이퍼링크도 걸어놓았음. (29line)

➔ 실행 화면



About "Dockerfile"

Code | Blame | 10 lines (6 loc) · 173 Bytes

```
1   FROM php:8.1-apache
2
3   RUN docker-php-ext-install mysqli
4
5   COPY src/ /var/www/html/
6
7   RUN chown -R www-data:www-data /var/www/html \
8       && chmod -R 755 /var/www/html
9
10  EXPOSE 80
```

Mysql로 데이터베이스를 다뤄야 하기 때문에 확장프로그램을 깔아주는 코드인 (3 line)을 작성했고,
웹페이지에 접속 시 src/ 속 파일들이 메인 루트가 되게끔 하기 위해,
src/ 에 있는 PHP 파일들을 웹 서버 루트에 복사해줌 (5 line)

7-8번 라인은 권한 설정을 위한 부분인데,
웹서버가 파일을 읽고 쓰게 만들었고,
~ & & chmod -R "7": 소유자에게 읽기, 쓰기, 실행 권한을 주었고,
5: 기타 사용자에게는 읽기, 실행 권한만을 줌.
→ 즉, 서버가 파일을 실행, 브라우저는 파일을 읽을 수 있게 설정함.

About "docker-compose.yml"

```yaml
version: '3.8'
services:
  web:
    build: .
    ports:
      - "8082:80"
    volumes:
      - ./src:/var/www/html
    depends_on:
      - db

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: testdb
    ports:
      - "3306:3306"
    volumes:
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

파일로, PHP 웹 서버와 MySQL 데이터베이스를 동시에 실행시키기 위해 작성함.

8082포트를 사용했고,

로컬 src/ 폴더를 /var/www/html에 연결해서, 코드를 바꾸면 실시간으로 반영되게끔 함.
Db 실행 후 파일들이 실행될 수 있게끔 depends_on 코드 추가.

12 line부터 시작하는 코드는...

MySQL 5.7 버전의 데이터베이스를 자동으로 실행,
Root의 비밀번호를 root로 설정,
처음 시작할 때 testdb라는 데이터베이스 자동 빌드,
init.sql 파일에 적힌 명령어들을 실행해서 db를 초기화.

하는 역할을 담당함

About "Init.sql"

```sql
CREATE DATABASE IF NOT EXISTS testdb;
USE testdb;

CREATE TABLE IF NOT EXISTS users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  userId VARCHAR(50) NOT NULL,
  userPassword VARCHAR(50) NOT NULL,
  userName VARCHAR(50) NOT NULL
);

INSERT INTO users (userId, userPassword, userName)
VALUES ('123', '123', 'seoyeon');
```

userId	userPassword	userName
123	123	seoyeon
...	...	...
...	...	...

userId, userPassword, username 총 3개의 table 존재