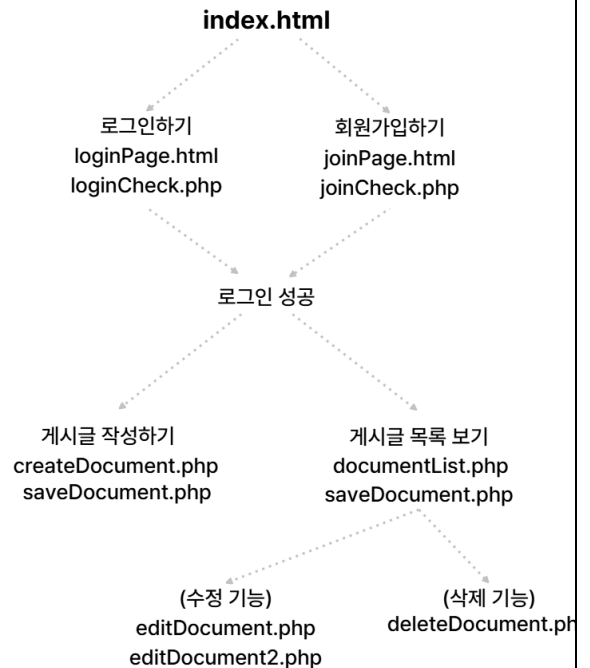


README

1) 구조도 및 전체 파일 기능 설명

```
src/
├── CreateDocument/
│   ├── createDocument.php
│   ├── deleteDocument.php
│   ├── documentList.php
│   ├── editDocument.php
│   ├── editDocument2.php
│   └── saveDocument.php
├── CssFiles/
│   ├── Preview.css
│   ├── index.html
│   ├── loginCheck.php
│   ├── loginPage.html
│   ├── joinPage.html
│   └── joinCheck.php
├── Dockerfile
├── docker-compose.yml
└── testdb.sql
```



*파일명 이름에 오타가 있습니다,
참고해주세요

loginPage, loginCheck, joinPage,
joinCheck

-> loginPage1, loginCheck1, joinPage1,
joinCheck1.

< testdb.sql (데이터 테이블 구조) >

users

"id"	"userId"	"userPassword"	"userName"
int(11)	varchar(50)	varchar(255)	varchar(50)
2	/ 123	/ 123	/ SeoYeon
3	/ 345	/ 345	/ cacao
4	/ uniCTJ	/ 1234	/ yunseo
5	/ 4321	/ 4321	/ 4321

< - "users" data table

createdocument

"id" "userId" "title" "document" "createDate"

int(50) varchar(50) text medium text date

15 / 345 / I love chocolate / dream cacao 56percent / 2025-05-18

18 / 4321 / 안녕하세요 / 제 이름은 4321입니다.!!! / 2025-05-20

19 / 123 / test edit / hi my name is testName / 2025-05-20

<- "createdocument" data table

2) 구조도 및 전체 파일 기능 설명

메인 페이지 설명

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Document</title>
7 </head>
8 <body>
9     <h2>Main Page</h2>
10    <br><br>
11    <h2>로그인 하러 가기</h2>
12    <a href="/loginPage1.html">Login!</a>
13    <br>
14    <h2>회원가입 하러 가기</h2>
15    <a href="/joinPage1.html">Join!</a>
16 </body>
17 </html>
```

<index.php>

Localhost에 접속 시 바로 뜨는 메인 페이지.

<a>태그로 로그인 페이지 링크와 회원가입 페이지 링크를 연결해줬다.

로그인 화면 작동 코드 설명

loginPage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/cssFiles/Preview.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <div class="center">
        <h2 style="margin-bottom: 30px;">로그인 폼</h2>
        <p></p>
        <form method="post" action="loginCheck1.php">
            <div>
                <p class="center">아이디: <input type="text" name="userId"></p>
                <p class="center">비밀번호: <input type="password" name="userPassword"></p>
                <br>
                <p>
                    <input class="button" style="text-align: center;"
                    type="submit" value="로그인"/>
                </p>
                <p>
                    <input
                    class="button" style="text-align: center;"
                    type="button" value="회원가입하러 가기" onclick="location.href='/joinPage1.html'"/>
                </p>
            </div>
        </form>
    </div>
</body>
</html>
```



loginCheck.php

```
1 <?php
2 session_start();
3 header("Content-Type: text/html; charset=utf-8");
4
5 $conn = new mysqli("db", "root", "root", "testdb");
6 $conn->set_charset("utf8mb4");
7 if ($conn->connect_error) {
8     die("DB connect Fail: " . $conn->connect_error);
9 }
10
11 $userId = $_POST['userId'];
12 $userPassword = $_POST['userPassword'];
13
14 if($userId && $userPassword) {
15     $sql = "SELECT * FROM users WHERE userID = ? AND userPassword = ?";
16     $stmt = $conn->prepare($sql);
17     $stmt->bind_param("ss", $userId, $userPassword);
18     $stmt->execute();
19     $result = $stmt->get_result();
20
21     if($result->num_rows == 1) {
22         $row = $result->fetch_assoc();
23         $userName = $row['userName'];
24         $userId = $row['userId'];
25         $_SESSION['userId'] = $row['userId'];
26         $_SESSION['userName'] = $row['userName'];
27
28         echo "Login Success!\nHello, " . htmlspecialchars($userName) . "!";
29         echo "<br><br>";
30         echo "<button type='button'
31         onclick='location.href='CreateDocument/createDocument.php'>Write Document</button>";
32         echo "<br><br>";
33         echo "<button type='button' onclick='location.href='CreateDocument/documentList.php'>Document List</button>";
34     } else {
35         echo "<script>
36         console.log('로그인 실패: 아이디 또는 비밀번호가 틀렸습니다!');
37         alert('로그인 실패: 다시 시도해주세요.');
38         window.location.href = '/loginPage1.html';
39         </script>";
40     }
41 }
42
43 $conn->close();
44 ?>
```

<loginPage.html>

Form 태그를 활용해 post 방식으로 loginCheck.php로 정보를 전달해주었다.

그 후 Input 태그로 아이디와 비밀번호를 받고, 추후 다른 파일에서 이 데이터를 저장할 수 있도록 name으로 각각의 이름을 userId, userPassword로 저장했다.

Input 태그를 이용해 로그인 버튼과 회원가입 버튼도 만들었다.

로그인은 submit 타입을 부여했고, 회원가입 버튼은 onclick을 이용해 회원가입 페이지로 넘어갈 수 있도록 href로 링크를 걸어주었다.

loginPage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="CssFiles/Preview.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="center">
    <h2 style="margin-bottom: 30px;">로그인 폼</h2>
    <p></p>
    <form method="post" action="loginCheck1.php">
      <div>
        <p class="center">아이디: <input type="text" name="userId"></p>
        <p class="center">비밀번호: <input type="password" name="userPassword"></p>
        <br>
        <p>
          <input class="button" style="text-align: center;"
            type="submit" value="로그인"/>
        </p>
        <p>
          <input class="button" style="text-align: center;"
            type="button" value="회원 가입하기"
            onclick="location.href='./joinPage1.html'"/>
        </p>
      </div>
    </form>
  </div>
</body>
</html>
```

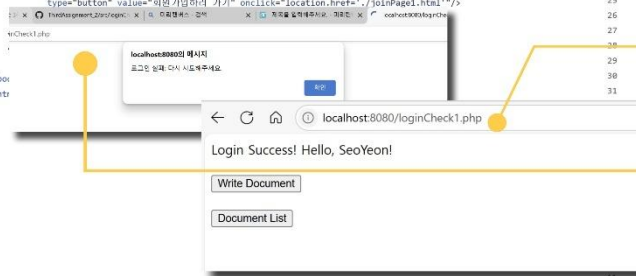
form 태그로 post 방식으로
입력 데이터 전달

loginCheck.php

```
1 <?php
2 session_start();
3 header("Content-Type: text/html; charset=utf-8");
4
5 $conn = new mysqli("db", "root", "root", "testdb");
6 $conn->set_charset("utf8mb4");
7 if ($conn->connect_error) {
8   die("DB connect Fail: " . $conn->connect_error);
9 }
10
11 $userId = $_POST['userId'];
12 $userPassword = $_POST['userPassword'];
13
14 if($userId && $userPassword) {
15   $sql = "SELECT * FROM users WHERE userId = ? AND userPassword = ?";
16   $stmt = $conn->prepare($sql);
17   $stmt->bind_param("ss", $userId, $userPassword);
18   $stmt->execute();
19   $result = $stmt->get_result();
20
21   if($result->num_rows == 1) {
22     $row = $result->fetch_assoc();
23     $userName = $row['userName'];
24     $userId = $row['userId'];
25     $_SESSION['userId'] = $row['userId'];
26     $_SESSION['userName'] = $row['userName'];
27
28     echo "Login Success!\nHello, " . htmlspecialchars($userName). "!";
29     echo "<br><br>";
30     <button type="button">
31       onclick="location.href='CreateDocument/createDocument.php'">Write Document</button>";
32     echo "<br><br>";
33     <button type="button">
34       onclick="location.href='CreateDocument/documentList.php'">Document List</button>";
35   } else {
36     echo "<br><br>";
37     console.log('로그인 실패: 아이디 또는 비밀번호가 틀렸습니다!');
38     alert('로그인 실패: 다시 시도해주세요!');
39     window.location.href = './loginPage1.html';
40   }
41 }
42 $conn->close();
```

입력받은 id, pass가
db에 있는지 체크,
그 결과를 \$result에 담기

있다는 가정 하에,
\$_SESSION에 저장할



loginCheck.php

loginPage에서 넘겨 받은 userId, userPassword를 해당 파일에서 변수로 따로 선언해서 저장하고,

sql문을 활용해 이 유저아이디와 비밀번호가 db에 있는지 체크했다.
그 결과를 \$result라는 변수를 선언해 저장했다.

\$result가 있는가를 가지고 if문을 돌려 해당 값을 \$_SESSION에 저장했고, 없을 시에 사용자가 다시 로그인을 시도할 수 있게끔 loginPage.html로 하이퍼링크를 걸었다.

(이 때, 경고 메시지를 띄우게 하고 싶었기에, 자바스크립트의 alert를 사용하기 위해서 echo로 자바스크립트 문법을 활용했다. 비슷하게, 하이퍼 링크를 걸어서 loginPage.html로 넘길 때에도, 자바스크립트 문법인 window.location.href 문법을 활용했다.)

\$result값이 제대로 들어갔다면, createDocument.php, documentList.php로 갈 수 있는 하이퍼 링크를 걸어서 글쓰기와 글 목록 보기 창으로 넘어갈 수 있도록 했다.

(이때 버튼으로 클릭해서 이동할 수 있게끔 만들기 위해서 button 태그를 활용했고, onclick을 사용해 클릭했을 시 페이지 이동이 일어날 수 있게끔 코드를 작성했다.)

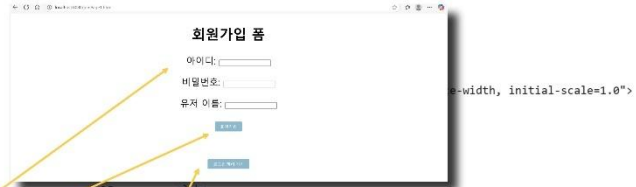
회원가입 화면 코드 설명

joinPage.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <link rel="stylesheet" href="CssFiles/Preview.css">
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Join Page</title>
8 </head>
9 <body>
10 <div class = "center">
11 <h2>회원가입 폼</h2>
12 <form method = "post" action = "joinCheck1.php">
13 <p>아이디: <input type="text" name="newId"></p>
14 <p>비밀번호: <input type="password" name="newPassword"></p>
15 <p>유저 이름: <input type="text" name = "newName"></p>
16 <button
17 style = "bottom: 20px; right: 20px;"
18 class = "button" type="submit">회원가입</button>
19 <br>
20 <br>
21 <br>
22 </form>
23 <button class="button" onclick="location.href='./loginPage1.html'">로그인 하러 가기</button>
24 </div>
25 </body>
26 </html>
```

joinCheck.php

```
1 <?php
2 $conn = new mysqli("db", "root", "root", "testdb");
3 if ($conn->connect_error) {
4 die("testDB connect Fail: ".$conn->connect_error);
5 }
6
7 $newId= $_POST['newId'] ?? '';
8 $newPassword= $_POST['newPassword'] ?? '';
9 $newName= $_POST['newName'] ?? '';
10
11 if ($newId && $newPassword && $newName) {
12 $sql = "INSERT INTO users (userId, userPassword, userName) VALUES (?, ?, ?)";
13 $stmt = $conn->prepare($sql);
14 $stmt->bind_param("sss", $newId, $newPassword, $newName);
15 $stmt->execute();
16
17 echo "Join Success!\n";
18 }else{
19 echo "Join failed";
20 }
21 $conn->close();
22 ?>
23 <a href="./loginPage1.html">Go to Login Page</a>
24 </body>
25 </html>
```



loginPage.html 처럼 form 태그를 통해 post로 정보를 joinCheck로 넘겨줬고, 넘긴 정보를 받아 joinCheck가 기존 db에 이 로그인 값이 존재하는지 체크하도록 설계함. Input 태그를 통해 아이디 비밀번호 유저 이름에 대한 정보를 받았고, button 태그를 눌러야 제출되게끔 설계했다.

joinPage.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <link rel="stylesheet" href="css/joinPage.css">
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8 </head>
9 <body>
10 <div class="container">
11 <div class="row">
12 <div class="col">
13 <div class="card">
14 <div class="card-body">
15 <div class="form">
16 <div class="form-group">
17 <input type="text" name="newName">
18 </div>
19 <div class="form-group">
20 <input type="password" name="newPassword">
21 </div>
22 <div class="form-group">
23 <input type="text" name="newId">
24 </div>
25 <div class="form-group">
26 <input type="button" value="Join" />
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

기존 db 속 users 데이터 테이블을
joinCheck.php로 불러옴

joinCheck.php

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Document</title>
7 </head>
8 <body>
9 <div class="container">
10 <div class="row">
11 <div class="col">
12 <div class="card">
13 <div class="card-body">
14 <div class="form">
15 <div class="form-group">
16 <input type="text" name="newName">
17 </div>
18 <div class="form-group">
19 <input type="password" name="newPassword">
20 </div>
21 <div class="form-group">
22 <input type="text" name="newId">
23 </div>
24 <div class="form-group">
25 <input type="button" value="Join" />
26 </div>
27 </div>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>

```

post로 입력받은 값을 불러와
joinCheck.php 파일에서 변수 선언

우선 기존 db를 불러와야 하기 때문에, \$conn이라는 이름으로 변수를 선언해서 db를 불러왔고, (에러를 방지하기 위해 연결이 안됐을 시 띄울 창도 따로 마련함)

joinPage.html에서 post로 받아온 newId, newPassword, newName을 joinCheck에서도 쓸 수 있도록 변수로 선언해주는 과정을 거쳤다.

그리고 if 문을 선언해서 newId, newPassword, newName 세 정보가 모두 존재할 때, 이 값들을 sql문을 활용해서 데이터 테이블에 저장하도록 했다.

*데이터를 저장 시, id 때문에 에러가 나지 않도록, testdb 속 users 테이블에 id 속성을 다음과 같이 정해주었습니다.

#	이름	종류	문자정렬순서	보기	Null	기본값	설명	추가	실행
1	id	int(11)	utf8mb4_general_ci	아니오	없음	AUTO_INCREMENT		변경	삭제 더보기
2	userId	varchar(50)	utf8mb4_general_ci	아니오	없음			변경	삭제 더보기
3	userPassword	varchar(255)	utf8mb4_general_ci	아니오	없음			변경	삭제 더보기
4	userName	varchar(50)	utf8mb4_general_ci	예	NULL			변경	삭제 더보기

(id 속성: AUTO_INCREMENT)

CreateDocument(file)/createDocument

```

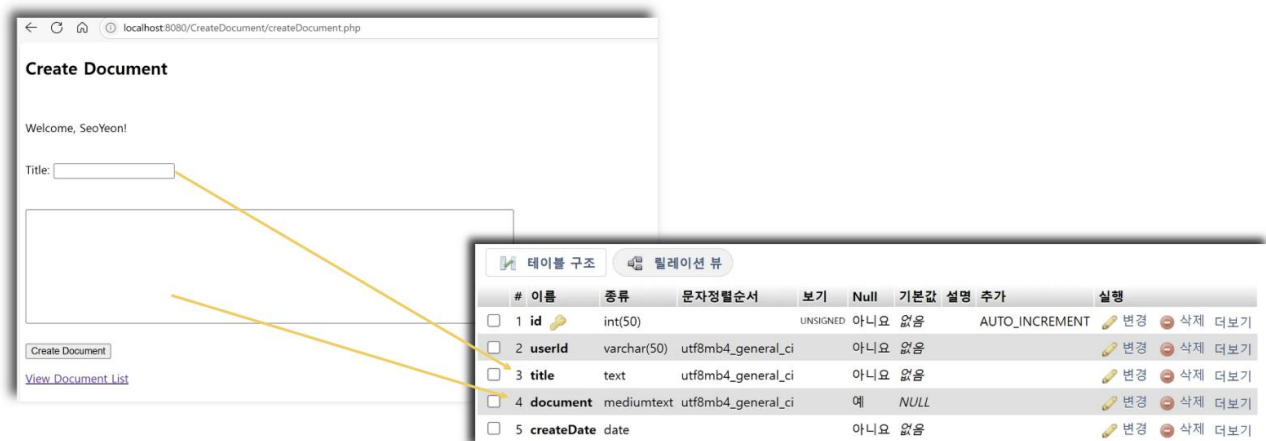
1  <?php
2  session_start();?>
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Create Document</title>
9  </head>
10 <body>
11     <div>
12         <h2>Create Document</h2>
13         <br>
14         <?php
15             $connection = new mysqli("db", "root", "root", "testdb");
16             if($connection->connect_error) {
17                 die("DB connect Fail: " . $connection->connect_error);
18             }
19
20             $sql = "SELECT * FROM users";
21             $result = $connection->query($sql);
22
23             echo "<p>Welcome, " . htmlspecialchars($_SESSION["userName"]) . "!</p>";
24         ?>
25         <br>
26         <form method="post" action="saveDocument.php">
27             <label for="title">Title:</label>
28             <input type="text" id="title" name="title" required><br><br>
29
30             <label for="document"></label><br>
31             <textarea id="document" name="document" rows="10" cols="100" required></textarea><br><br>
32
33             <input type="submit" value="Create Document">
34         </form>
35         <p></p>
36         <a href="documentList.php">View Document List</a>
37     </div>
38 </body>
39 </html>

```



우선 session_start() 함수로 세션을 시작한 후, 세션에 저장된 사용자 이름 (\$_SESSION["userName"])을 불러와 화면에 출력했다.
(사용자가 로그인했음을 나타냄)

제목, 내용을 입력받을 수 있게 했고, 제목은 input 태그를, 내용은 textarea 태그를 활용했다. Input(submit전용) 태그로 버튼을 만들었고, 이를 누르면 데이터가 saveDocument.php로 날아가게끔 설계함.



입력한대로 데이터테이블에 저장이 됨. id는 자동으로 설정되어 저장되고, createDate도 현재 (오늘) 날짜를 알아서 계산해서 저장하도록 코드를 짤. userId는 로그인되어있는 아이디를 session으로 가져와서 저장하게끔 코드를 짤.

CreateDocument(file)/saveDocument


```

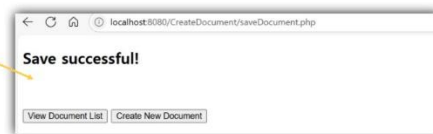
1  <?session_start();
2  >
3  <!DOCTYPE html>
4  <html lang="en">
5  <head>
6  <meta charset="UTF-8">
7  <meta name="viewport" content="width=device-width, initial-scale=1.0">
8  <title>Document</title>
9  </head>
10 <body>
11 <?php
12 $conn = new mysqli("db", "root", "root", "testdb");
13 if ($conn->connect_error) {
14     die("DB connect Fail: " . $conn->connect_error);
15 }
16
17 $title = $_POST['title'] ?? '';
18 $document = $_POST['document'] ?? '';
19 $todayDate = date("Y-m-d");
20 $userId = $_SESSION['userId'] ?? '';
21
22 if ($title || $document) {
23     $sql = "INSERT INTO createdocument (title, document, createDate, userId) VALUES (?, ?, ?, ?)";
24     $stmt = $conn->prepare($sql);
25     $stmt->bind_param("ssss", $title, $document, $todayDate, $userId);
26     $stmt->execute();
27     $stmt->close();
28 }
29
30 $conn->close();
31 >
32 <div>
33     <h2>Save successful</h2>
34     <br><br>
35     <button class="button" type="button" onclick="location.href='documentList.php'">View Document List</button>
36     <button class="button" type="button" onclick="location.href='createDocument.php'">Create New Document</button>
37 </div>
38 </body>
39 </html>

```



* DB
 ↳ testdb
 (아이디:root, 비번:root)

#	이름	종류	문자정렬순서	보기	Null	기본값	설명 추가	실행
1	id	int(50)		UNSIGNED	아니요	없음	AUTO_INCREMENT	변경 삭제 더보기
2	userId	varchar(50)	utf8mb4_general_ci		아니요	없음		변경 삭제 더보기
3	title	text	utf8mb4_general_ci		아니요	없음		변경 삭제 더보기
4	document	mediumtext	utf8mb4_general_ci		예	NULL		변경 삭제 더보기
5	createDate	date			아니요	없음		변경 삭제 더보기



\$_SESSION을 사용할 것이기 때문에 맨 앞줄에 session_start()선언.

\$conn으로 db와 연결하고, title과 document는 post로 받아오고, todadDate는 date라는 함수를 이용해서, userId는 session에서 가져오게끔 했다.

Id는 A I 옵션을 넣어서 알아서 자동으로 저장하게끔 만들.

(\$sql = "INSERT INTO createdocument (~, ~, ~, ~) VALUES (?, ?, ?, ?)")

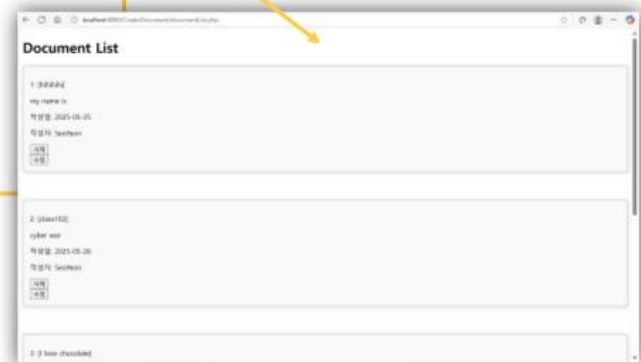
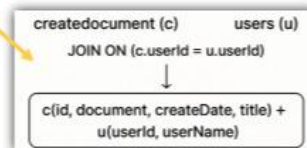
사용자가 보는 화면에는 <button> 태그와 onclick을 활용해서 documentList.php로 가는 버튼, createDocument.php로 가는 버튼을 추가해줌.

CreateDocument(file)/documentList.php

```

1  <?php
2  header("Content-Type: text/html; charset=utf-8");
3  $conn = new mysqli("db", "root", "root", "testdb");
4  $conn->set_charset("utf8mb4");
5
6  if($conn->connect_error) {
7      die("DB connect Fail: " . $conn->connect_error);
8  }
9
10 $sql = "SELECT c.id, c.document, c.createDate, c.title, u.userId, u.userName
11        FROM createdocument c
12        JOIN users u ON c.userId = u.userId";
13
14
15 $result = $conn->query($sql);
16 ?>
17
18 <!DOCTYPE html>
19 <html>
20 <head>
21     <meta charset="UTF-8">
22     <meta name="viewport" content="width=device-width, initial-scale=1.0">
23     <link rel="stylesheet" href="../../CssFiles/Preview.css">
24     <title>Document List</title>
25 </head>
26 <body>
27     <h1>Document List</h1>
28
29     <?php
30     $indexNum = 1;
31     if($result->num_rows > 0) {
32         while($row = $result->fetch_assoc()) {
33             echo "<div class='list-card'>";
34             echo "<p>$indexNum: [" . htmlspecialchars($row['title']) . "</p>";
35             echo "<p>". htmlspecialchars($row['document']) . "</p>";
36             echo "<p>작성일: " . htmlspecialchars($row['createDate']) . "</p>";
37             echo "<p>작성자: " . htmlspecialchars($row['userName']) . "</p>";
38             echo "<p></p>";
39             echo "<form method='post' action='deleteDocument.php'>";
40             echo "<input type='hidden' name='id' value='". $row['id'] . ">";
41             echo "<button type='submit'>삭제</button>";
42             echo "</form>";
43             echo "<a href='editDocument.php?id=" . urlencode($row['id']) . ">";
44             echo "<button>수정</button>";
45             echo "</a>";
46             echo "</div>";
47             echo "<br><br>";
48             $indexNum++;
49         }
50     }
51     ?>
52     <p></p>
53     <a href='createDocument.php'>create Document</a>
54     <a href='../../index.php'>Logout</a>
55 </body>
56 </html>

```



Db속 저장된 데이터를 불러와야 하기 때문에, 먼저 \$conn으로 sql을 연결시켰음.

그 후, users table과 createdocument table을 JOIN ~ ON ~ 쿼리문을 이용해 합침.

Users (u) + createdocument (c)	
userId, userName	Id, document, createDate, title

사용자에게 보이는 화면은 php 코드 속 echo로 작성했다.

(sql에서 한줄씩 읽어와서 출력해야하기 때문)

\$result로 쿼리를 불러오고, "if (\$result->num_rows > 0)"로 쿼리 결과가 하나 이상 존재할 경우에만 출력을 수행하게끔 함

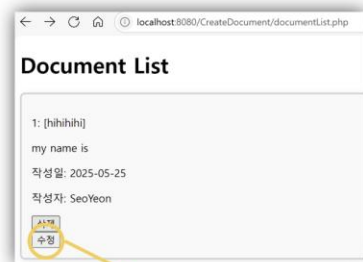
\$row로 \$result 쿼리를 한줄 씩 읽어오고, while로 돌려서 순차적으로 가져와 각 문서를 하나씩 출력할 수 있도록 함.

*htmlspecialchars() 함수를 사용하여 HTML 태그가 출력되지 않도록 이스케이프 처리.

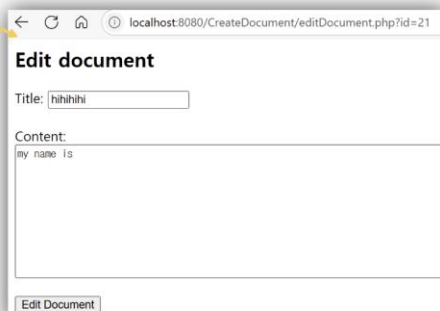
CreateDocument(file)/editDocument.php && editDocument2.php

1) editDocument.php

```
1 <?php
2 session_start();
3 header("Content-Type: text/html; charset=utf-8");
4 $conn = new mysqli("db", "root", "root", "testdb");
5
6 if($conn->connect_error) {
7     die("DB connect fail: " . $conn->connect_error);
8 }
9
10 $doc = null;
11 $isCorrect = false;
12
13 if(isset($_GET["id"])) {
14     $documentId = $_GET["id"];
15     $loginId = $_SESSION["userId"] ?? '';
16
17     $ss = $conn->prepare("SELECT * FROM createdocument WHERE id = ?");
18     $ss->bind_param("s", $documentId);
19     $ss->execute();
20
21     $result = $ss->get_result();
22
23     if($result->num_rows > 0){
24         $doc = $result->fetch_assoc();
25         $documentUserId = $doc["userId"];
26
27         if($loginId == $documentUserId) {
28             $isCorrect = true;
29         }
30     }
31 }
32
33
34 <h2>Edit document</h2>
35 <?php if ($isCorrect): ?>
36     <form method="post" action="editDocument2.php">
37         <input type="hidden" name="id" value="{htmlspecialchars($doc['id'])}" />
38
39         <label form="title">Title:</label>
40         <input type="text" id="title" name = "title" value="{htmlspecialchars($doc['title'])}" required/><br><br>
41
42         <label form="document">Content:</label><br>
43         <textarea id="document" name="document" rows="18" cols="100" required/>{htmlspecialchars($doc['document'])} </textarea><br><br>
44
45         <input type="submit" value="Edit Document">
46     </form>
47 <?php else: ?>
48     <p>error</p>
49     <br>
50     <p>권한이 없거나, 게시물이 없음</p>
51 <?php endif; ?>
```



문서 내용 그대로
유지된 채
페이지 등장



editDocument.php 작동 로직

1. session에 저장되어있는 내 로그인 정보 속 userId를 불러온다
2. 내가 클릭한 게시글의 userId를 불러온다
3. 두 userId를 비교하고, 같을 시에만 isCorrect를 true로 바꿈
4. isCorrect가 true일 경우에만 화면이 출력되도록 함.

1: Get 방식으로 id를 가져와서 저장

```
$documentId <- id
```

```
$loginId <- userId
```

2: \$result 쿼리를 한줄씩 읽어오면서 (\$doc) 그 줄 속 userId를 \$documentUserId에 저장.

3: loginId == documentUserId 일 때, (로그인된 아이디와 문서의 아이디가 일치할 때)에만 html 코드를 실행할 수 있게 설계함.

form태그로 입력 값을 editDocument2로 보내게끔 코드를 짰고, input태그를 이용해서 title과 content를 수정했다.

*37line을 넣은 이유: 문서 식별을 위해 id가 필요하기 때문, 사용자에게는 굳이 보일 필요 없기에 hidden으로 숨겨서 보내게끔 코드를 짰

2) editDocument2.php

```
1 <?php
2 $conn = new mysqli("db", "root", "root", "testdb");
3 $conn->set_charset("utf8mb4");
4 header("Content-Type: text/html; charset=utf-8");
5
6 $id = $_POST['id'];
7 $title = $_POST['title'];
8 $document = $_POST['document'];
9
10 $stmt = $conn->prepare("UPDATE createdocument SET title = ?, document = ? WHERE id = ?");
11 $stmt->bind_param("sss", $title, $document, $id);
12 $stmt->execute();
13
14 echo "<script> alert('Success edit');
15     window.location.href = './documentList.php';
16     </script>";
17 ?>
```

editDocument.php

수정할 내용을 입력받는 역할

수정된 title
수정된 content

editDocument2.php

내용을 다시 저장하는 역할

\$conn으로 데이터베이스 연결하고,
editDocument로부터 받은 수정된 title과
document를 데이터베이스에 저장함.

Javascript 문법을 echo로 써서 alert 기능으로 성공됨을 알려주는 기능 까지.
또한, 이 코드 실행 후 바로 documentList로 이동할 수 있도록 15번 line도 추가했다.

CreateDocument(file)/deleteDocument.php

```
1 <?php
2 session_start();
3 $conn = new mysqli("db", "root", "root", "testdb");
4 header("Content-Type: text/html; charset=utf-8");
5 $conn->set_charset("utf8mb4");
6
7 if ($conn->connect_error) {
8     die("DB connect fail: " . $conn->connect_error);
9 }
10
11 if (isset($_POST['id'])) {
12     $documentId = intval($_POST['id']);
13     $loginId = $_SESSION['userId'] ?? '';
14
15     $ss = $conn->prepare("SELECT userId FROM createdocument WHERE id = ?");
16     $ss->bind_param("s", $documentId);
17     $ss->execute();
18     $result = $ss->get_result();
19
20     if ($result->num_rows == 1) {
21         $row = $result->fetch_assoc();
22         $documentUserId = $row['userId'];
23
24         if ($loginId == $documentUserId) {
25             $stmt = $conn->prepare("DELETE FROM createdocument WHERE id = ?");
26             $stmt->bind_param("s", $documentId);
27             if ($stmt->execute()) {
28                 echo "<script>alert('Document deleted successfully.');";
29             } else {
30                 echo "<script>alert('Failed to delete document.');";
31             }
32             $stmt->close();
33         } else {
34             echo "<script>alert('You do not delete this document.');";
35         }
36     } else {
37         echo "<script>alert('Document not found.');";
38     }
39 }
40 ?>
```

주요 흐름은 다음과 같습니다.

1. 세션(session_start)에서 로그인한 사용자 ID를 확인
2. POST 방식으로 전달된 id 값을 정수로 변환
3. 데이터베이스(testdb)에서 해당 문서의 소유자(userId)를 조회
4. 세션 사용자 ID와 소유자 ID가 일치할 경우 삭제 쿼리 실행
5. 결과에 따라 알림(alert) 출력

```
41
42 <!DOCTYPE html>
43 <html lang="en">
44 <head>
45     <meta charset="UTF-8">
46     <meta name="viewport" content="width=device-width, initial-scale=1.0">
47     <title>Document</title>
48 </head>
49 <body>
50     <a href = "documentList.php">Document List</a>
51     <br>
52     <a href = "../loginPage.html">Logout</a>
53     <br>
54     <a href = "createDocument.php">Create Document</a>
55 </body>
56 </html>
```

게시글 삭제 버튼을 누른 후에 나오게 될 페이지.
<a> 태그를 이용해 하이퍼링크를 걸어서,
documentList(게시글 목록), loginPage(로그인페이지), createDocument.php(게시글 작성하기) 로 바로 이동할 수 있게 만들,

Dockerfile docker-compose.yml 에 대한 설명

```
1  version: '3.8'
2
3  services:
4    web:
5      build: .
6      container_name: my_php_web
7      ports:
8        - "8080:80"
9      volumes:
10       - ./src:/var/www/html
11      depends_on:
12        - db
13
14    db:
15      image: mysql:5.7
16      container_name: my_mysql_db
17      environment:
18        MYSQL_ROOT_PASSWORD: root
19        MYSQL_DATABASE: testdb
20        MYSQL_USER: user
21        MYSQL_PASSWORD: password
22      volumes:
23        - db_data:/var/lib/mysql
24
25    phpmyadmin:
26      image: phpmyadmin/phpmyadmin
27      container_name: my_phpmyadmin
28      ports:
29        - "8081:80"
30      environment:
31        PMA_HOST: db
32        MYSQL_ROOT_PASSWORD: root
33
34    volumes:
35      db_data:
```

현재 디렉터리(.)에 있는 Dockerfile을 이용해 이미지를 빌드.

컨테이너 이름: my_php_web로 고정 지정.
호스트의 8080 포트를 컨테이너 내부 80 포트(웹 서버)로 포워딩

MySQL 5.7 공식 이미지를 사용,
컨테이너 이름을 my_mysql_db로 고정 지정

→ environment

- MYSQL_ROOT_PASSWORD: root
- MYSQL_DATABASE: testdb
- MYSQL_USER: user
- MYSQL_PASSWORD: password

컨테이너 이름을 my_phpmyadmin로 고정
호스트 8081 포트는 컨테이너 80 포트로 매핑.