

2019/12/16

시큐어

수강신청의 정석

한학기에는 수강신청이 지배한다

1. 팀원소개

2. 프로젝트 개요

§1 역할 분담

§2 추진 일정

§3 목적

§4 연구내용 – 보안 적용 내용

3. 시나리오 및 실제 구현

§1 시나리오

§2 보안을 적용한 구현

4. 프로젝트 결과 및 활용방안

§1 프로젝트의 장점

§2 전망 – 예상 활용분야 및 활용방안

§3 기대성

§4 한계점

5. 참고문헌

1. 팀원소개

Dream High

Dream
정드림



High2
김연희



High1
권승주



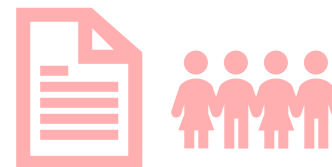
High3
김주언



2. 프로젝트 개요

§1. 역할 분담

- 주제발표 : 정드림
- 최종발표 : 김연희
- 자료조사, 시나리오 구상, PPT 제작, 구현 등 : 정드림, 권승주, 김연희, 김주언



2. 프로젝트 개요

§2. 추진일정

- 0916 : 주제 발표
- 0924 : 주제 변경
- 0924~1203 : 구현
- 1203~1208 : 최종 정리
- 1209&1216 : 최종 발표



2. 프로젝트 개요

§3. 목적

- 강의 별로 장바구니 담은 순위 노출해 수강신청에 도움
- 학생들이 보다 더 용이하게 수강신청을 할 수 있는 환경을 만듦



2. 프로젝트 개요

§4. 연구내용 – 보안 적용 내용

- 중요 개인정보를 암호화 하여 DB에 저장 – 중요정보 평문저장
- 중요 개인정보 저장 시 안전한 알고리즘 사용 – 취약한 암호화 알고리즘
- 학수번호 입력 시 입력 데이터 검증
- 세션 아이디 검사 및 세션 만료 시간 제한
- Prepared statement를 통한 SQL 인젝션 공격 방지

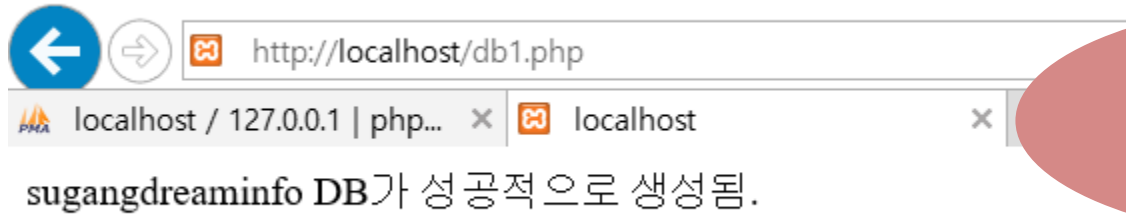


3. 시나리오 및 실제 구현

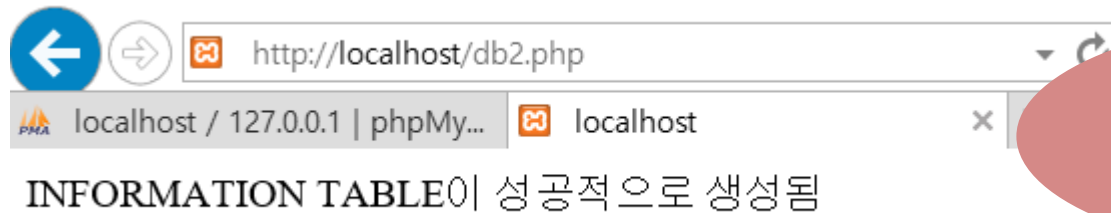
§1. 시나리오

시뮬레이션_DB 생성

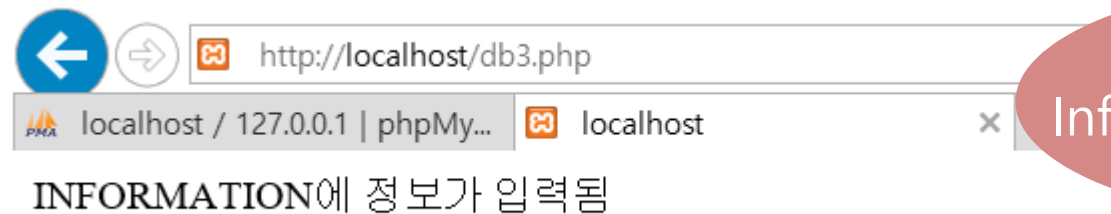
2



db1.php
Database 생성



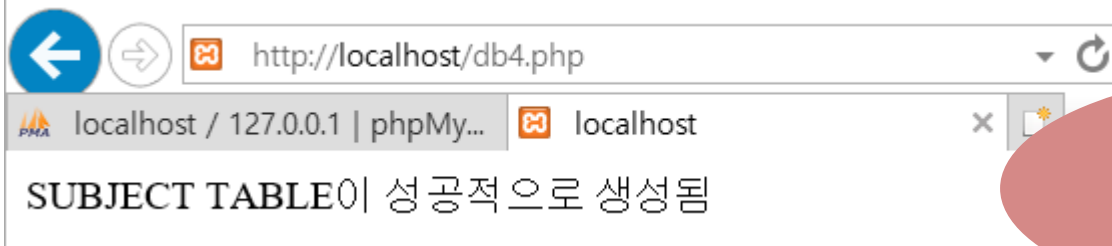
db2.php
학생정보 Table
(Information) 생성



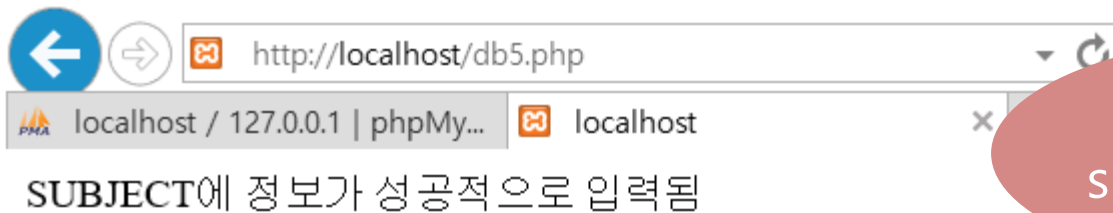
Db3.php
Information에 정보 입력

시뮬레이션_DB 입력

2



db4.php
과목정보 table
(subject) 생성



db5.php
subject에 정보 입력

시뮬레이션_홈 화면

종합정보시스템

: 10:00 ~ **16:00**

(전학년) 2.10(월) ~ 2.14(금)
 수강변경(전학년) 3.2(월) ~ 3.6(금)

4학년	02.10(월)
3학년	02.11(화)
2학년	02.12(수)
1학년	02.13(목)
전학년	02.14(금)

※ 학부수강신청 추가일정 ※
 전학년 **02.17(월)**
 [이용시간:10:00~16:00]

>
 </수강신청/변경 일정 >

이화여자대학교 2020-1학기 학부 수강신청

ID(학번)

PASSWORD(비밀번호)

[LOGIN](#)

강의 수강을 지양하기 바람.

[로그아웃(Logout)]을 클릭하십시오.
 (실습실/PC방 등 공용 PC 사용시 특히 주의요함!!)

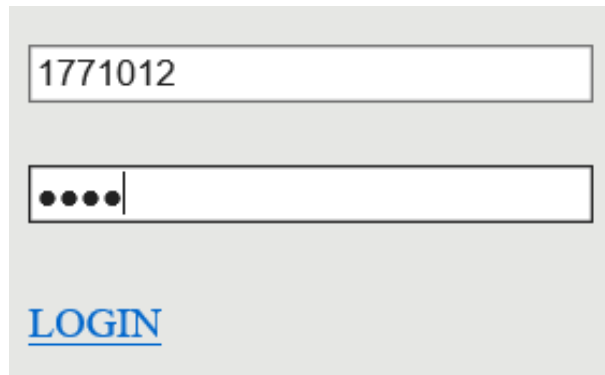
test2.html
홈 화면

ID(학번)

PASSWORD(비밀번호)

[LOGIN](#)

시뮬레이션_홈 화면



1771012

••••

[LOGIN](#)

ID, PASSWOR가
일치하지
않는 경우



로그인 실패
아이디 또는 비밀번호를 다시 확인하세요.
등록되지 않은 아이디이거나, 아이디 또는 비밀번호를 잘못 입력하셨습니다.
[로그인하기](#)

입력데이터 검증

The screenshot shows a web browser window with the address bar displaying `http://localhost/form3.php`. The browser tabs show `localhost / 127.0.0.1 / sugang...` and `localhost`. The page content includes a session timeout notice, user information, a login instruction, and input fields for student ID verification.

0:29:54 후 자동로그아웃

학과 : 사이버보안
학번 : 17
이름 : 김주연

30분이 넘어갈 동안
아무 활동도 하지 않으면
자동 로그아웃

1~3순위 학수번호를 입력해주세요.

입력데이터 검증

1~3순위 학수번호를 입력해주세요.

검색

웹 페이지 메시지

×



정확한 학수번호를 입력하세요.

확인



0:29:52 후 자동로그아웃

학과 : 사이버보안

학번 : 17

이름 : 김주연

1~3순위 학수번호를 입력해주세요.

검색

입력데이터 검증

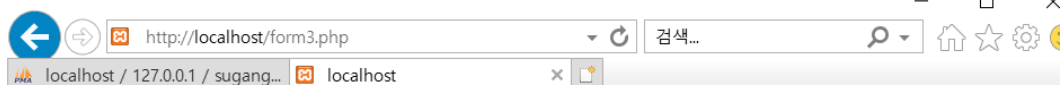
1~3순위 학수번호를 입력해주세요.

웹 페이지 메시지

×



서로 다른 학수번호를 입력해주세요.



0:29:52 후 자동로그아웃

학과 : 사이버보안

학번 : 17

이름 : 김주연

1~3순위 학수번호를 입력해주세요.

입력데이터 검증

38971 38977 38985 x



당신의 1순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 넣은 사람들의 수입니다.

1순위 : 1명

2순위 : 0명

3순위 : 0명

인원제한 : 20명

[수강신청](#)

당신의 2순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 넣은 사람들의 수입니다.

1순위 : 0명

2순위 : 1명

3순위 : 0명

인원제한 : 30명

[수강신청](#)

당신의 3순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 넣은 사람들의 수입니다.

1순위 : 0명

2순위 : 0명

3순위 : 1명

인원제한 : 30명

[수강신청](#)

세션 아이디어 검사

당신의 1순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 담은 학생의 수입니다.

1순위 : 17명

2순위 : 15명

3순위 : 26명

인원제한 : 20명

수강신청

당신의 2순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 담은 학생의 수입니다.

1순위 : 11명

2순위 : 15명

3순위 : 17명

인원제한 : 20명

수강신청

당신의 3순위 장바구니 과목을 각각 1순위, 2순위, 3순위에 담은 학생의 수입니다.

1순위 : 22명

2순위 : 15명

3순위 : 22명

인원제한 : 30명

수강신청

안정

위험

거의 불가능

세션 만료 시간 제한

2

웹 페이지 메시지 X



수강신청 성공!

확인

'에

웹 페이지 메시지 X



수강 인원 정원에 도달하였습니다.

확인

3. 시나리오 및 실제 구현

§2. 보안을 적용한 구현

1. 중요정보 평문저장

```

5 function AES_Encode($plain_text,$key)
6 {
7     return base64_encode(openssl_encrypt($plain_text, "aes-256-cbc", $key, true, str_repeat(chr(0), 16)));
8 }
9
10 $name1=AES_Encode('권승주',$key_name);$name2=AES_Encode('김연희',$key_name);
11 $name3=AES_Encode('김주연',$key_name);$name4=AES_Encode('정드림',$key_name);
12 $name5=AES_Encode('강민주',$key_name);$name6=AES_Encode('김현선',$key_name);
13 $name7=AES_Encode('김재원',$key_name);$name8=AES_Encode('강미현',$key_name);
14 $name9=AES_Encode('박정연',$key_name);$name10=AES_Encode('백주리',$key_name);
15 $name11=AES_Encode('백현정',$key_name);$name12=AES_Encode('조아라',$key_name);
16 $name13=AES_Encode('김주연',$key_name);$name14=AES_Encode('최지희',$key_name);
17 $name15=AES_Encode('손민주',$key_name);$name16=AES_Encode('이소은',$key_name);
18 $name17=AES_Encode('지수연',$key_name);$name18=AES_Encode('김소현',$key_name);
19 $name19=AES_Encode('안해민',$key_name);$name20=AES_Encode('최예진',$key_name);
20 $name21=AES_Encode('유현선',$key_name);$name22=AES_Encode('이은빈',$key_name);
21 $name23=AES_Encode('이재음',$key_name);$name24=AES_Encode('이주은',$key_name);
22 $name25=AES_Encode('임은지',$key_name);$name26=AES_Encode('정주현',$key_name);
23 $name27=AES_Encode('조희주',$key_name);$name28=AES_Encode('자혜진',$key_name);
24 $name29=AES_Encode('최슬기',$key_name);$name30=AES_Encode('최현지',$key_name);
25 $name31=AES_Encode('김효원',$key_name);$name32=AES_Encode('손지수',$key_name);
26 $name33=AES_Encode('도인실',$key_name);$name34=AES_Encode('윤아람',$key_name);
27 $name35=AES_Encode('채기훈',$key_name);$name36=AES_Encode('이상호',$key_name);

```

강의를 담은 학생의 이름과
비밀번호 암호화하여 DB에 저장

```

49 $sql="
50 INSERT INTO INFORMATION VALUES
51 ($name1, '1771000', SHA2('skdud',512), '사이버보안', '00000', '00000', '00000'),
52 ($name2, '1771001', SHA2('alsgus99',512), '사이버보안', '00000', '00000', '00000'),
53 ($name3, '1771002', SHA2('ehdus87',512), '사이버보안', '00000', '00000', '00000'),
54 ($name4, '1771003', SHA2('codus123',512), '사이버보안', '00000', '00000', '00000'),
55 ($name5, '1771004', SHA2('wlsdud12',512), '사이버보안', '00000', '00000', '00000'),
56 ($name6, '1771005', SHA2('dnwls1',512), '사이버보안', '00000', '00000', '00000'),
57 ($name7, '1771006', SHA2('rufrud',512), '사이버보안', '00000', '00000', '00000'),
58 ($name8, '1771007', SHA2('tjddn55',512), '사이버보안', '00000', '00000', '00000'),
59 ($name9, '1771008', SHA2('sooo12',512), '사이버보안', '00000', '00000', '00000'),
60 ($name10, '1771009', SHA2('wltjd88',512), '사이버보안', '00000', '00000', '00000'),
61 ($name11, '1771010', SHA2('woghks12',512), '사이버보안', '00000', '00000', '00000'),
62 ($name12, '1771011', SHA2('rollcccc',512), '사이버보안', '00000', '00000', '00000'),
63 ($name13, '1771012', SHA2('choi',512), '사이버보안', '00000', '00000', '00000'),
64 ($name14, '1771013', SHA2('alsk2131',512), '사이버보안', '00000', '00000', '00000'),
65 ($name15, '1771014', SHA2('eognl',512), '사이버보안', '00000', '00000', '00000'),
66 ($name16, '1771015', SHA2('wlgn33',512), '사이버보안', '00000', '00000', '00000'),
67 ($name17, '1771016', SHA2('ksj',512), '사이버보안', '00000', '00000', '00000'),
68 ($name18, '1771017', SHA2('sooooom',512), '사이버보안', '00000', '00000', '00000'),
69 ($name19, '1771018', SHA2('ekslpdf53',512), '사이버보안', '00000', '00000', '00000'),
70 ($name20, '1771019', SHA2('tjddns7',512), '사이버보안', '00000', '00000', '00000'),
71 ($name21, '1771020', SHA2('parknr53',512), '사이버보안', '00000', '00000', '00000'),
72 ($name22, '1771021', SHA2('qwer1234',512), '사이버보안', '00000', '00000', '00000'),
73 ($name23, '1771022', SHA2('ming132',512), '사이버보안', '00000', '00000', '00000'),

```

2. 취약한 암호화 알고리즘 사용²

```
$sql="
INSERT INTO INFORMATION VALUES
('$name1', '1771000', SHA2('skdud',512), '사이버보안', '00000', '00000', '00000' ),
('$name2', '1771001', SHA2('alsgus99',512), '사이버보안', '00000', '00000', '00000' ),
('$name3', '1771002', SHA2('ehdus87',512), '사이버보안', '00000', '00000', '00000' ),
('$name4', '1771003', SHA2('codus123',512), '사이버보안', '00000', '00000', '00000' ),
('$name5', '1771004', SHA2('wlsdud12',512), '사이버보안', '00000', '00000', '00000' ),
('$name6', '1771005', SHA2('dnwls1',512), '사이버보안', '00000', '00000', '00000' ),
('$name7', '1771006', SHA2('rufud',512), '사이버보안', '00000', '00000', '00000' ),
('$name8', '1771007', SHA2('tjddn55',512), '사이버보안', '00000', '00000', '00000' ),
('$name9', '1771008', SHA2('sooo12',512), '사이버보안', '00000', '00000', '00000' ),
('$name10', '1771009', SHA2('wltjd88',512), '사이버보안', '00000', '00000', '00000' ),
('$name11', '1771010', SHA2('woghks12',512), '사이버보안', '00000', '00000', '00000' ),
('$name12', '1771011', SHA2('rollccc',512), '사이버보안', '00000', '00000', '00000' ),
('$name13', '1771012', SHA2('choi',512), '사이버보안', '00000', '00000', '00000' ),
('$name14', '1771013', SHA2('alsk2131',512), '사이버보안', '00000', '00000', '00000' ),
('$name15', '1771014', SHA2('eogn1',512), '사이버보안', '00000', '00000', '00000' ),
('$name16', '1771015', SHA2('wlgn33',512), '사이버보안', '00000', '00000', '00000' ),
('$name17', '1771016', SHA2('ksj',512), '사이버보안', '00000', '00000', '00000' ),
('$name18', '1771017', SHA2('soooom',512), '사이버보안', '00000', '00000', '00000' ),
('$name19', '1771018', SHA2('eksldpf53',512), '사이버보안', '00000', '00000', '00000' ),
('$name20', '1771019', SHA2('tjddns7',512), '사이버보안', '00000', '00000', '00000' ),
('$name21', '1771020', SHA2('parknr53',512), '사이버보안', '00000', '00000', '00000' ),
('$name22', '1771021', SHA2('qwer1234',512), '사이버보안', '00000', '00000', '00000' ),
('$name23', '1771022', SHA2('ming132',512), '사이버보안', '00000', '00000', '00000' ),
('$name24', '1771023', SHA2('dodoll1',512), '사이버보안', '00000', '00000', '00000' ),
('$name25', '1771024', SHA2('ewhall1',512), '사이버보안', '00000', '00000', '00000' ),
-- ...

function AES_Encode($plain_text,$key)
{
    return base64_encode(openssl_encrypt($plain_text, "aes-256-cbc", $key, true, str_repeat(chr(0), 16)));
}

$name1=AES_Encode('권승주', $key_name); $name2=AES_Encode('김연희', $key_name);
$name3=AES_Encode('김주연', $key_name); $name4=AES_Encode('정드림', $key_name);
$name5=AES_Encode('강민준', $key_name); $name6=AES_Encode('김준선', $key_name);
$name7=AES_Encode('김재원', $key_name); $name8=AES_Encode('김미현', $key_name);
$name9=AES_Encode('박정원', $key_name); $name10=AES_Encode('박주리', $key_name);
$name11=AES_Encode('백현정', $key_name); $name12=AES_Encode('조아라', $key_name);
$name13=AES_Encode('김주연', $key_name); $name14=AES_Encode('최희원', $key_name);
$name15=AES_Encode('손민지', $key_name); $name16=AES_Encode('이소은', $key_name);
$name17=AES_Encode('지수연', $key_name); $name18=AES_Encode('김소영', $key_name);
$name19=AES_Encode('안혜원', $key_name); $name20=AES_Encode('최예진', $key_name);
$name21=AES_Encode('유현선', $key_name); $name22=AES_Encode('이은빈', $key_name);
$name23=AES_Encode('이재윤', $key_name); $name24=AES_Encode('이주은', $key_name);
$name25=AES_Encode('임은지', $key_name); $name26=AES_Encode('정주현', $key_name);
$name27=AES_Encode('조희주', $key_name); $name28=AES_Encode('차해린', $key_name);
$name29=AES_Encode('최슬기', $key_name); $name30=AES_Encode('최현지', $key_name);
$name31=AES_Encode('김도원', $key_name); $name32=AES_Encode('손지은', $key_name);
$name33=AES_Encode('도민실', $key_name); $name34=AES_Encode('윤아린', $key_name);
$name35=AES_Encode('재기은', $key_name); $name36=AES_Encode('이상보', $key_name);
```

SHA-512, AES
사용

비밀번호를 암호화할 때
취약한 (RC2, RC4, RC5, RC6,
MD4, MD5, SHA-1, DES) 대신
SHA-512와 AES를 사용하여 안전하게 암호화

3. 입력데이터 검증

1~3순위 학수번호를 입력해주세요.

11111 22222 33333 x

검색

1~3순위 학수번호를 입력해주세요.

aaaaa bbbbb ccccd x

검색

웹 페이지 메시지

×



정확한 학수번호를 입력하세요.

확인

```
<?php
$var1 = $_POST["ONE"];
$var2 = $_POST["TWO"];
$var3 = $_POST["THREE"];
$var1 = my_classnum($var1);
$var2 = my_classnum($var2);
$var3 = my_classnum($var3);
check($var1,$var2,$var3);
function my_classnum($x)
{
    if (($x>=38970) && ($x<=38994)) {
        return $x;
    }
    else
    {
        echo"<script>alert('정확한 학수번호를 입력하세요.')</script>";
        echo ("<meta http-equiv='refresh' content='0; URL=form3.php'>");
        exit;
    }
}
```

문자나 범위 밖의 학수번호를 입력하였을 때

→ 정확한 학수번호를 입력하라는 팝업창이 뜨고
다시 학수번호를 입력하는 페이지로 이동

입력값이 정수인지,
범위 내에 있는지 검사

4. 입력데이터 검증

1~3순위 학수번호를 입력해주세요.

웹 페이지 메시지



서로 다른 학수번호를 입력해주세요.

```
function check($x,$y,$z)
{
    if($x==$y||$y==$z||$z==$x)
    {
        echo"<script>alert('서로 다른 학수번호를 입력해주세요.')</script>";
        echo ("<meta http-equiv='refresh' content='0; URL=form3.php'>");
        exit;
    }
}
```

같은 학수번호를 입력하였을 때

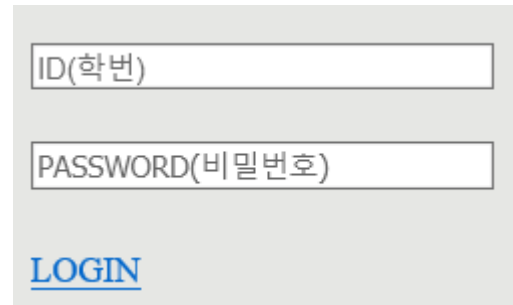
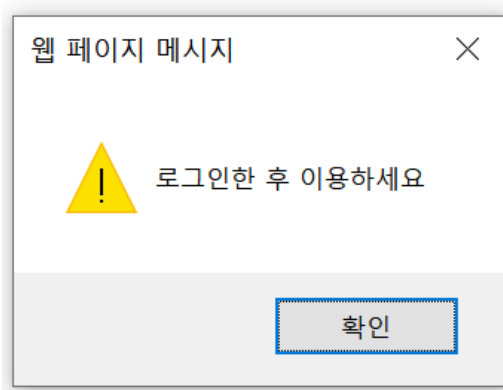
입력값이 같은지
검사

→ 서로 다른 학수번호를 입력하라는 팝업창이 뜨고
다시 학수번호를 입력하는 페이지로 이동

5. 세션 아이디 검사

2

```
<?php
session_start();
if(!$SESSION['user_id']){
    echo"<script>alert('로그인한 후 이용하세요')</script>";
    echo ("<meta http-equiv='refresh' content='0; URL=test2.html'>");
    exit;
}
?>
```

A screenshot of a login form. It has two input fields: "ID(학번)" (ID (Student Number)) and "PASSWORD(비밀번호)" (PASSWORD (Password)). Below the fields is a blue link labeled "LOGIN".

수강신청 페이지 코드 첫 부분에서 세션 아이디를 검사하여
로그인하지 않고 수강신청 페이지에 접근하지 못하게함

6. 세션 만료 시간 제한²

0:30:00 후 자동로그아웃

연장

학과 : 사이버보안

학번 : 17

이름 : 권승주

세션이 종료되었습니다. 다시 로그인해주세요.

HOME

30분이 지나면 logout.php 페이지로 넘어간 후
session_destroy() 시켜서 자동으로 로그아웃

자동 세션 종료

```
var tid;
var cnt = parseInt(1800); //초기값(초단위)
function counter_init() {
    tid = setInterval("counter_run()", 1000);
}

function counter_reset() {
    clearInterval(tid);
    cnt = parseInt(1800);
    counter_init();
}

function counter_run() {
    document.all.counter.innerText = time_format(cnt);
    cnt--;
    if(cnt < 0) {
        clearInterval(tid);
        self.location = "logout.php";
    }
}

function time_format(s) {
    var nHour=0;
    var nMin=0;
    var nSec=0;
    if(s>0) {
        nMin = parseInt(s/60);
        nSec = s%60;

        if(nMin>60) {
            nHour = parseInt(nMin/60);
            nMin = nMin%60;
        }
    }
    if(nSec<10) nSec = "0"+nSec;
    if(nMin<10) nMin = "0"+nMin;

    return ""+nHour+":"+nMin+":"+nSec;
}
```

```
<html>
<body>
<?php
    session_start();
    echo "세션이 종료되었습니다. 다시 로그인해주세요.";
    session_destroy();
    ?>

<form method="post" action="test2.html"><br>
<br><br>
<input type="submit" value="HOME">
</form>
</body>
</html>
```

7. SQL injection

```
<?php
if($_SERVER["REQUEST_METHOD"]=="POST"){
    $conn=mysqli_connect("localhost","root","","sugangdream");
    if(!$conn){
        die("Connection failed : ".mysqli_connect_error());
    }
    $password=$_POST["password"];
    $stmt=$conn->stmt_init();
    $stmt=$conn->prepare("SELECT password FROM INFORMATION WHERE user_id=?");
    $stmt->bind_param("s",$_POST["user_id"]);
    $stmt->execute();
    $result=$stmt->get_result();
```

```
<?php
$conn=mysqli_connect("localhost","root","","sugangdream")
or die("접속 실패");

$stmt1=$conn->stmt_init();
$stmt1=$conn->prepare("SELECT MAJOR FROM INFORMATION WHERE user_id=?");
$stmt1->bind_param("s",$_SESSION['user_id']);
$stmt1->execute();
$result1=$stmt1->get_result();

$stmt2=$conn->stmt_init();
$stmt2=$conn->prepare("SELECT left(user_id,2) user_id FROM INFORMATION WHERE user_id=?");
$stmt2->bind_param("s",$_SESSION['user_id']);
$stmt2->execute();
$result2=$stmt2->get_result();

$stmt3=$conn->stmt_init();
$stmt3=$conn->prepare("SELECT NAME FROM INFORMATION WHERE user_id=?");
$stmt3->bind_param("s",$_SESSION['user_id']);
$stmt3->execute();
$result3=$stmt3->get_result();
```

```
<?php
$conn=mysqli_connect("localhost","root","","sugangdream")
or die("접속 실패");

$stmt1=$conn->stmt_init();
$stmt1=$conn->prepare("SELECT PEOPLENUM1 FROM SUBJECT WHERE NUMBER = ");
$stmt1->bind_param("s",$var1);
$stmt1->execute();
$result4=$stmt1->get_result();

$stmt2=$conn->stmt_init();
$stmt2=$conn->prepare("SELECT PEOPLENUM2 FROM SUBJECT WHERE NUMBER = ");
$stmt2->bind_param("s",$var2);
$stmt2->execute();
$result5=$stmt2->get_result();

$stmt3=$conn->stmt_init();
$stmt3=$conn->prepare("SELECT PEOPLENUM3 FROM SUBJECT WHERE NUMBER = ");
$stmt3->bind_param("s",$var3);
$stmt3->execute();
$result6=$stmt3->get_result();

$stmt4=$conn->stmt_init();
$stmt4=$conn->prepare("SELECT NUM FROM SUBJECT WHERE NUMBER =?");
$stmt4->bind_param("s",$var1);
$stmt4->execute();
$show1=$stmt4->get_result();

$stmt5=$conn->stmt_init();
$stmt5=$conn->prepare("SELECT NUM FROM SUBJECT WHERE NUMBER =?");
$stmt5->bind_param("s",$var2);
$stmt5->execute();
$show2=$stmt5->get_result();

$stmt6=$conn->stmt_init();
$stmt6=$conn->prepare("SELECT NUM FROM SUBJECT WHERE NUMBER =?");
$stmt6->bind_param("s",$var3);
$stmt6->execute();
$show3=$stmt6->get_result();
```

statement 대신
preparedstatement 사용

SQL injection 공
격
방지

4. 프로젝트 결과 및 활용 방안

§1. 프로젝트의 장점

- 장바구니에 순위당 몇 명이 담았는지 볼 수 있어서 수강신청 계획 짤 때 효과적
- 자기가 들을 강의만 장바구니에 담아 정확한 인원파악 가능
- 보안 취약점을 분석해 안전한 암호화 코드를 삽입함으로써 안전성 보장 (세션 만료 시 자동 로그아웃, 로그인하지 않고 바로 장바구니 페이지로 넘어가지 못함 등)
- DB에 입력된 개인정보를 암호화해 높은 보안성 유지



4. 프로젝트 결과 및 활용 방안

§2. 전망 – 예상 활용분야 및 활용방안

Enrollment Capacity

학수번호(Course No): 22250 분반(Class): 01 ~ 05 조회(Search)

최대 5개 분반까지만 표시되며 화면 내용은 조회 시점의 수강인원이므로 실제 수강신청 입력 때는 차이가 있을 수 있습니다.
Up to 5 classes are displayed, page displays class size at time of query and may change when actual course registration takes place

순번 No	교과목명 Subject Title	학수번호 Course no	분반 Class	실습분반 Lab/Practice Class	총정원 Total Capacity	학년 Curr
1	재무관리 Financial Management	22250	01		1순위 : 1명	
2			02			
3			03			
4			04		2순위 : 0명	
5			05		3순위 : 0명	

총정원 (Total Capacity) : 해당 교과목의 총 수강가능정원

인원제한 : 20명

학년별 수강정원(Current Capacity)*

1) 우선수강신청 기간(2월/8월)

해당 교과목의 학년별 우선수강 배정인원

- ① 우선수강신청대상 교과목인 경우 : 해당 교과목의 본인 해당 학년 우선수강 배정인원
- ② 우선수강신청대상 교과목이 아닌 경우 : 해당 교과목의 총 수강 가능 정원

2) 학년별 수강신청 기간(2월/8월)

학년별 수강신청기간의 해당 학년별 배정인원에 대한 누적정원

- 4학년 수강신청기간 - 4학년 수강가능정원
- 3학년 수강신청기간 - 3, 4학년 수강가능정원
- 2학년 수강신청기간 - 2, 3, 4학년 수강가능정원
- 1학년 수강신청기간 - 총정원 (단, 학년별 계산식에 의해 정원오차가 있을 수 있으며, 해당 TO는 '3'의 전체학년 수강신청 기간에 오

3) 전체학년 수강신청 기간 및 수강신청 확인 및 변경 기간(2월/3월초, 8월/9월초)
해당 교과목의 총 수강가능정원 (=총정원)



4. 프로젝트 결과 및 활용 방안

§3. 기대성

- 수강신청 할 강의의 실질경쟁률 파악 가능
- 학교측에서 실질경쟁률을 확인하고 인원수에 맞게 분반 개설 가능
- 자동 로그아웃 기능으로 공공장소에서도 안전하게 이용 가능
- 개인정보 노출이나 도용과 같은 피해 없음
- 현재 개설된 강의만 장바구니에 담을 수 있음



4. 프로젝트 결과 및 활 용방안

§4. 한계점

- 수강신청 할 강의만 장바구니에 담을 수 있기 때문에 실패 시 차선책을 세우기가 어려움
- 장바구니 담은 인원을 전체학년으로만 보여줘서 학년별 인원 파악은 불가능



5. 참고문헌

<http://au2.php.net/>

<http://php.net/manual/kr/tutorial.forms.php>

<https://www.codingfactory.net/>

<http://php.net/manual/kr/funcref.php>

<http://php.net/manual/kr/language.types.intro.php>

<http://php.net/manual/kr/ref.session.php>

<http://php.net/manual/kr/function.session-destroy.php>

성공적인 웹 프로그래밍과 PHP와 MySQL, 루크 웰링, 로라 톰슨, 정보문화사

이것이 MySQL이다(MySQL 설치부터 PHP 연동까지 한번에!), 우재남, 한빛미디어

암호화 데이터베이스에서 영역 질의를 위한 기술, 김천식, 김형중, 홍유식, 2008, 전자공학회논문지, 대한전자공학회



2019/12/16

시큐어

수강신청의 정석

한학기는 수강신청이 지배한다

감사합니다

