

2023-1학기 DB Term Project 최종 보고서 및 소스코드

201902703 설윤환

시스템 구현 환경

운영체제: 윈도우10

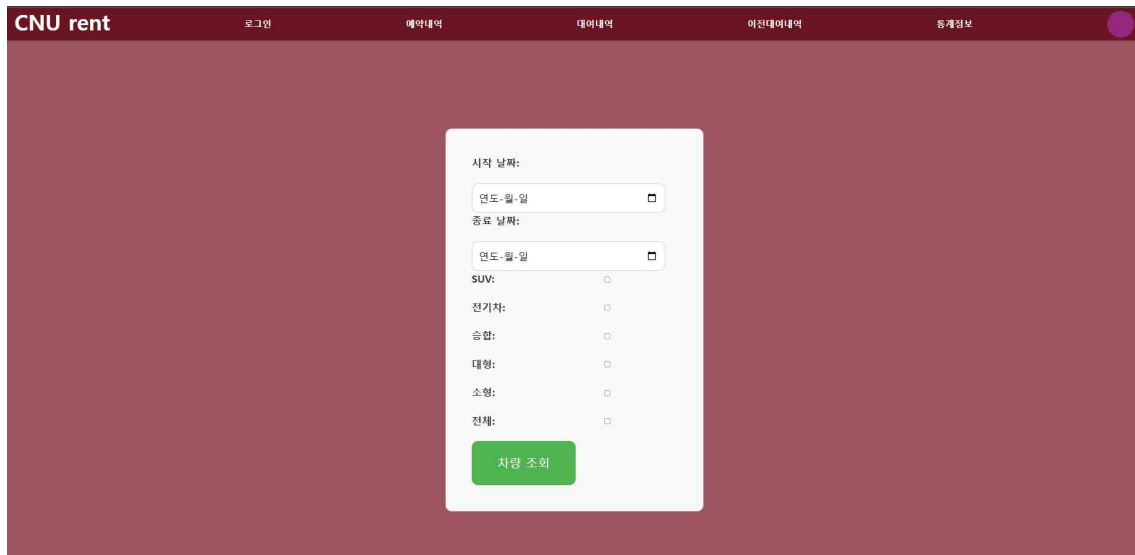
데이터베이스: mysql

구현 언어: html, php, javascript, css

전체 인터페이스

1. index.php (초기 화면)

가장 초기 화면이다.

The screenshot shows the 'CNU rent' website's initial search interface. The header is dark red with white text for '로그인' (Login), '예약내역' (Reservation History), '대여내역' (Rental History), '이전대여내역' (Previous Rental History), and '통계정보' (Statistics). The main content area is a light gray box with a white background, containing search filters. It includes '시작 날짜:' (Start Date) and '종료 날짜:' (End Date) with date pickers, and a list of vehicle types: 'SUV:', '전기차:' (Electric Car), '승합:' (Passenger), '태형:' (Type), '소형:' (Small), and '전체:' (All), each with a radio button. A green '차량 조회' (Search Vehicle) button is at the bottom.

우선 header에 cnurent(index.php), 로그인, 예약 내역, 대여 내역, 이전대여 내역, 통계정보를 표시하였다.

고객이 로그인을 한 후에 시작 날짜, 종료 날짜와 차종을 선택해 예약 가능한 렌터카를 검색할 수 있다. 차종을 선택하는 방법은 두 가지가 있는데 원하는 차종을 선택해 고르거나 전체를 선택해 차량을 조회하는 방법이다.

2. login.php (로그인 화면)

로그인을 할 수 있는 화면이다. customer테이블에 있는 회원 정보와 맞게 입력하면 로그인에 완료된다. 로그인에 session으로 구현하였다. 로그인에 성공하면 ‘환영합니다라’는 알림이 뜨고 index.php로 넘어간다. 이때 우측 상단에 로그인아이디가 프로필로 뜬다. 로그인 성공 후에는 로그인 header가 로그아웃으로 변경된다.

로그인에 실패할 경우 ‘존재하지 않는 아이디 또는 비밀번호입니다’가 alert로 뜬다.

로그인에 하지 않은 상태로 예약내역, 대여 내역, 이전 대여 내역으로 들어가려 하면 로그인이 필요하다는 문장이 나온다.

3. reserve.php (예약 화면)

CNU rent		로그아웃	예약내역	대여내역	이전대여내역	동계정보
모델 이름	차종	하루당 가격	연료	좌석 수	차량 번호	선택 옵션
기아EV6	전기차	120000	ev	5	123가4568	○ 네비게이션, 블루투스, 후방카메라
현대아반테	소형	100000	가솔린	5	123가4568	○ 블루투스
기아EV6	전기차	120000	ev	5	123가4567	○ 네비게이션
현대아반테	소형	100000	가솔린	5	123가4567	○ 블루투스, 후방카메라
기아EV6	전기차	120000	ev	5	141라1234	○ 네비게이션, 블루투스
현대소나타	소형	100000	가솔린	5	234라4568	○ 블루투스, 후방카메라
기아카니발	승합	170000	디젤	11	234라4568	○ 블루투스
쉐보레볼트EUV	전기차	120000	ev	5	345가3456	○ 네비게이션, 후방카메라
기아K5	대형	130000	가솔린	5	345라3425	○ 블루투스
기아카니발	승합	170000	디젤	11	566라4533	○ 네비게이션, 블루투스, 후방카메라
현대소나타	소형	100000	가솔린	5	999가4568	○ 네비게이션, 후방카메라
기아카니발	승합	170000	디젤	11	523라1341	○
현대산타페	suv	140000	가솔린	5	777라1234	○

시작 날짜: 2023-06-18
 종료 날짜: 2023-06-30
 예약하기

로그인에 성공한 후 시작 날짜, 종료 날짜를 선택하고 차량종류를 선택한 후 차량 조회를 누르면 reserve.php로 넘어간다. 선택한 옵션에 맞는 정보를 테이블로 보여준다.

각 정보는 rentcar, carmodel, optioj 테이블을 join하여 가져왔다. 이때 optioin은 다중값이기 때문에 left join으로 처리하고 option을 배열로 만들어 같은 차량일 경우에 배열에 추가

하는 방식으로 만든 후에 하나의 attribute로 만들어 주었다.

정보는 모델 이름, 차종, 하루당 가격, 연료, 좌석 수, 차량 번호, 옵션, 선택 버튼이 있다. 좌측 하단에 시작 날짜와 종료 날짜가 표시되고 예약하기 버튼이 있다.

선택 버튼은 radio로 구현하였고 선택에 표시한뒤 예약하기를 누르면 reserve_complete.php로 넘어가 예약이 성공한다. 여기서는 선택한 날짜에 같은 차종이 이미 예약이 되어있으면 안되게 하는 기능이 있다. select count(*)와 fetchColumn을 사용해 구현하였다.

할 수 있으면 insert문으로 reserve테이블에 값을 저장하였다.

예약한 것은 예약내역에 들어가 볼 수 있고 예약하기를 누르면 자동으로 예약내역(reserve_history.php)로 넘어간다.

4. reserve_history.php (예약내역 화면)

CNU rent						
로그아웃		예약내역	대여내역	이전대여내역	동계정보	
시작 날짜	차량 번호	예약 날짜	종료 날짜	고객 번호	모텔 이름	예약 취소
23/06/20	999714568	23/06/18	23/06/29	a1	현대소나타	○
23/06/20	141리1234	23/06/18	23/06/22	a1	기아EVS	○
예약 취소						

예약 내역 화면은 시작 날짜, 차량 번호, 예약 날짜, 종료 날짜, 고객 번호, 모델 이름, 예약 취소 버튼으로 되어있다. 앞서 했던 reserve의 form에서 radio와 hidden 타입으로 선택 차량의 licenseplatenumber, start_date, end_date 정보를 넘겨받았다.

우선 예약내역 중에 현재 날짜가 예약시작날짜를 넘어서면 대여되기 때문에 reserve테이블에서 삭제하고 rentcar테이블에 해당 차량의 정보를 대여로 바꾸었다. merge문을 사용해 rentcar테이블에서 해당 정보를 가져온 값으로 바꾸고 delete문으로 reserve에서 삭제하였다. 그 후 로그인한 고객의 cno와 같은 값들을 reserve 테이블에서 가져왔다.

만약 예약 취소를 선택하고 예약 취소를 누르면 reserve_cancel.php로 넘어가 delete문으로 reserve테이블에서 삭제하였다.

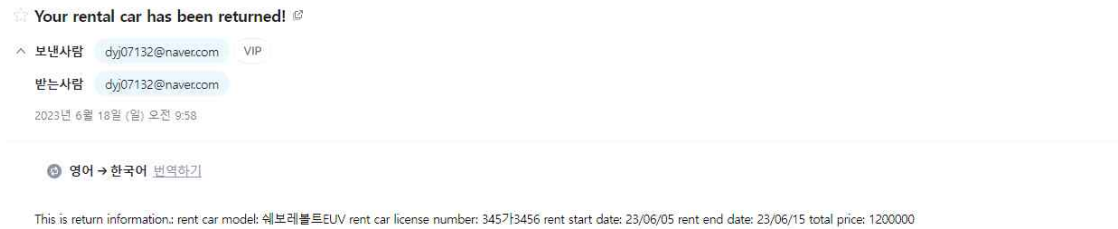
5. rent_history(대여 내역)

CNU rent						
로그아웃		예약내역	대여내역	이전대여내역	동계정보	
모텔 이름	차량 번호	시작 날짜	종료 날짜	총 가격	결제하기	
쉐보레볼트EUV	345713456	23/06/01	23/06/09	960000	○	
결제하기						

여기서도 reserve에서 날짜가 지나면 자동으로 대여가 되는 것을 하였고 현재 로그인한 사용자와 같은 cno를 갖는 값을 가지는 rentcar의 정보를 대여내역으로 만들었다.

결제하기 버튼이 있는데 이것을 하면 rent_complete.php로 넘어가 rentcar테이블에서 해당 차량의 정보를 전부 null값으로 바꾸고 반납된 차량의 정보를 previousrental에 저장한다.

반납이 완료되면 완료 메일을 보낸다.



6. pre_rent.php (이전대여내역)

대여 날짜	차량 번호	예약내역	반납 날짜	잔액 금액	고액 번호
23/06/19	523마1341		23/06/30	1870000	a1
23/06/19	999기4568		23/06/30	1870000	a1

이전대여내역은 previousrental 테이블에서 같은 cno를 갖는 정보를 가져와 테이블로 만들어 주었다. 없으면 이전 대여 기록이 없다는 문장이 뜬다.

7. statistic.php (통계 정보)

CNU rent					
로그아웃		예약내역	대여내역	이전대여내역	통계정보
매출순위					
랭킹	차량 번호	대여일	반납일	금액	총금액
1	234나4568	23/04/30	23/07/02	8190000	44650000
1	234나4568	23/06/01	23/06/09	960000	44650000
1	234나4568	22/06/18	23/06/08	35500000	44650000
4	141리1234	23/05/09	23/06/17	4680000	4680000
5	999기4568	23/06/19	23/06/30	1870000	1870000
5	523마1341	23/06/19	23/06/30	1870000	1870000
7	345기3456	23/06/05	23/06/15	1200000	1200000
8	566나4533	23/06/15	23/06/21	1020000	1020000
대여 횟수					
모형이름	횟수				
현대소나타	4				
기아카니온	2				
기아EV6	1				
쉐보레볼트EUV	1				
최근 반납된 차량					
차량 번호	대여일	반납일	금액		
234나4568	23/04/30	23/07/02	8190000		
999기4568	23/06/19	23/06/30	1870000		
523마1341	23/06/19	23/06/30	1870000		
566나4533	23/06/15	23/06/21	1020000		
141리1234	23/05/09	23/06/17	4680000		

통계 정보는 로그인 하지 않은 상태에서도 볼 수 있다. 우선 매출 순위는 전체 previousrental에서 반납금의 합계가 가장 높은 값을 기준으로 정렬하였다. 대여 횟수는 각 차량 별로 횟수를 표시하였다. 횟수를 기준으로 내림차순 하였다. 최근 반납된 차량은 반납일을 기준으로 내림차순하여 표시하였다.

주요 기능 코드

```
<!-- 체크박스 전부 선택하기 -->
<script>
    function toggleAllCheckboxes(checkbox) {
        var checkboxes = document.getElementsByName('car_type[]');
        for (var i = 0; i < checkboxes.length; i++) {
            checkboxes[i].checked = checkbox.checked;
        }
    }
</script>
```

모든 차종의 name을 car_type[]으로 만들어 js로 함수를 사용해 전부 체크하도록 만들었다.
그 후 전체버튼에 onclick으로 적용하였다.

```
// 로그인 폼에서 제출된 데이터 가져오기
if (isset($_POST['login_submit'])) {
    $loginid = $_POST['loginid'];
    $loginpw = $_POST['loginpw'];

    try {
        $conn = new PDO($dsn, $username, $password);
        $query = "SELECT * FROM customer WHERE cno = :cno AND passwd = :passwd";
        $stmt = $conn->prepare($query);
        $stmt->bindValue(':cno', $loginid);
        $stmt->bindValue(':passwd', $loginpw);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);

        if ($result) {
            // 로그인 성공
            $_SESSION['cno'] = $result['CNO'];
            $_SESSION['name'] = $result['NAME'];
            // 필요한 경우 세션에 추가할 데이터를 설정할 수 있다.

            echo "<script>alert('환영합니다!');
            location.href='./index.php'</script>";
        } else {
            // 로그인 실패
            echo "<script>alert('존재하지 않는 아이디 또는 비밀번호입니다.')</script>";
        }
    } catch (PDOException $e) {
        echo "에러 내용: " . $e->getMessage();
    }
}
```

session을 사용해 로그인 폼에서 제출된 데이터를 가져와 비교 후 맞으면 session에 저장하였다.

```

// 중복 제거를 위한 변수 초기화
$prev_license_plate = null;
$options = array();

foreach ($result as $row) {
    // 차량 번호가 이전과 같지 않은 경우에만 새로운 행 시작
    if ($row['LICENSEPLATENO'] != $prev_license_plate) {
        // 이전 차량 번호에 대한 옵션 출력
        if (!empty($options)) {
            echo "<td>" . implode(", ", $options) . "</td>";
        }

        echo "<tr>";
        echo "<td>" . $row['MODELNAME'] . "</td>";
        echo "<td>" . $row['VEHICLETYPE'] . "</td>";
        echo "<td>" . $row['RENTRATEPERDAY'] . "</td>";
        echo "<td>" . $row['FUEL'] . "</td>";
        echo "<td>" . $row['NUMBEROFSEATS'] . "</td>";
        echo "<td>" . $row['LICENSEPLATENO'] . "</td>";

        // 옵션 초기화
        $options = array($row['OPTIONNAME']);

        $prev_license_plate = $row['LICENSEPLATENO'];
        echo "<td><input type='radio' name='selected_car' value='" . $row['LICENSEPLATENO'] . "'></td>";
        echo "<input type='hidden' name='start_date' value='" . $_POST['start_date'] . "'>";
        echo "<input type='hidden' name='end_date' value='" . $_POST['end_date'] . "'></td>";
    } else {
        // 같은 차량 번호인 경우 옵션 추가
        $options[] = $row['OPTIONNAME'];
    }
}

// 마지막 행에 대한 옵션 출력
if (!empty($options)) {
    echo "<td>" . implode(", ", $options) . "</td>";
}

```

예약화면에서 옵션의 경우는 for문을 사용하면 각자 다른 테이블에 나와서 배열을 만들어 같은 차량 번호인 경우에 옵션을 배열에 넣어 한번에 나오도록 하였다. radio를 사용해 차량번호를 넘겨받도록 했고 hidden input을 사용해 start_date와 end_date를 넘겨주었다.

```

// 선택한 날짜에 이미 예약이 있으면 예약안됨
$checkQuery = "SELECT COUNT(*) as count
FROM reserve
WHERE cno = :cno and ((:start_date between startdate and enddate) or (:end_date between startdate and enddate)) ";
$checkStmt = $conn->prepare($checkQuery);
$checkStmt->bindParam(':cno', $cno);
$checkStmt->bindParam(':start_date', $_POST['start_date']);
$checkStmt->bindParam(':end_date', $_POST['end_date']);
$checkStmt->execute();
$count = $checkStmt->fetchColumn();

// 같은 차종을 다른 고객이 같은 날짜에 선택했다면 예약안됨
$checkQuery = "SELECT COUNT(*) as count
FROM reserve
WHERE licenseplateno = :licenseplateno and ((:start_date between startdate and enddate) or (:end_date between startdate and enddate)) ";
$checkStmt = $conn->prepare($checkQuery);
$checkStmt->bindParam(':cno', $cno);
$checkStmt->bindParam(':start_date', $_POST['start_date']);
$checkStmt->bindParam(':end_date', $_POST['end_date']);
$checkStmt->execute();
$count += $checkStmt->fetchColumn();

if ($count > 0) {
    echo "<script>alert('해당 날짜에 이미 예약이 있습니다.');

```

reserve_complete에 경우에 reserve테이블에 선택한 start_date와 end_date가 이미 startdate와 enddate 사이에 있을때를 count로 해서 fetchcolumn으로 count값을 받아서 0 보다 크면 알람을 띄우고 다시 돌아가도록 만들었다. 같은 차종이면 다른 고객이어도 예약을

못하게 하였다.

```
else{ //예약 할 수 있을때
    // 차량의 정보를 reserve테이블에 넣음
    $query = "INSERT INTO reserve (startdate, licenseplatenno, reservedate, enddate, cno)
              VALUES (:start_date, :licenseplatenno, to_date(sysdate,'yyyy.mm.dd hh24:mi'), :end_date, :cno)";

    $stmt = $conn->prepare($query);

    $stmt->bindParam(':start_date', $_POST['start_date']);
    $stmt->bindParam(':licenseplatenno', $licenseplatenno);
    $stmt->bindParam(':end_date', $_POST['end_date']);
    $stmt->bindParam(':cno', $cno);
    $stmt->execute();

    // 돌아오기
    header("Location: ../reserve_history.php");
} catch (PDOException $e) {
    echo("에러 내용: ".$e->getMessage());
}
```

만약 예약을 할 수 있으면 insert문으로 reserve테이블에 값을 저장하고 돌아오도록 하였다.

```
try {
    $conn = new PDO($dsn, $username, $password);
    session_start();
    $licenseplatenno = $_POST['selected_car'];

    // 예약내역에서 해당 값을 삭제함. licenseplatenno가 같으면 reserve테이블에서 삭제
    $deleteQuery = "DELETE FROM reserve WHERE licenseplatenno = :licenseplatenno ";

    $deleteStmt = $conn->prepare($deleteQuery);
    $deleteStmt->bindParam(':licenseplatenno', $licenseplatenno);

    $deleteStmt->execute();

} catch (PDOException $e) {
    echo("에러 내용: " . $e->getMessage());
}
```

예약 취소는 licenseplatenno가 같으면 delete문으로 삭제해 주었다.

```
//예약내역 중에 예약시작날짜가 현재 날짜 이하가 되면 예약테이블에서 삭제하고 rentcar에서 값을 수정한다.
$updateQuery = "MERGE INTO rentcar
                USING reserve
                ON (rentcar.licenseplatenno = reserve.licenseplatenno)
                WHEN MATCHED THEN
                UPDATE SET rentcar.daterented = reserve.startdate,
                           rentcar.datedue = reserve.enddate,
                           rentcar.cno = :cno
                WHERE reserve.startdate <= SYSDATE";

$deleteQuery = "DELETE FROM reserve WHERE startdate <= SYSDATE";

$updateStmt = $conn->prepare($updateQuery);
$updateStmt->bindParam(':cno', $_SESSION['cno']);

$deleteStmt = $conn->prepare($deleteQuery);

$updateStmt->execute();
$deleteStmt->execute();
```

예약내역 중에 시간이 지나면 자동으로 대여되도록 해 주었다. merge문으로 rentcar와 reserve를 조인한 후 update로 rentcar테이블을 reserve에 있는 값으로 바꿔주었다.

그 후 delete문으로 reserve문에 있는 해당 정보를 삭제하였다.

```
// previousrental과 메일에 사용할 정보 가져오기
$selectQuery = "SELECT r.daterented, r.modelname, r.datedue, (r.datedue-r.daterented)*c.rentrateperday as pay
FROM rentcar r join carmodel c on r.modelname = c.modelname
WHERE r.licenseplateno = :licenseplateno";
$stmt = $conn->prepare($selectQuery);
$stmt->bindParam(':licenseplateno', $selectedLicensePlateNo);
$stmt->execute();
$result = $stmt->fetch(PDO::FETCH_ASSOC);

// 반납된 차량의 정보를 previousrental에 저장한다
$insertQuery = "INSERT INTO previousrental (daterented, licenseplateno, datereturned, payment, cno)
VALUES (:daterented, :licenseplateno, :datedue, :pay, :cno)";

$insertStmt = $conn->prepare($insertQuery);
$insertStmt->bindParam(':daterented', $result['DATERENTED']);
$insertStmt->bindParam(':licenseplateno', $selectedLicensePlateNo);
$insertStmt->bindParam(':datedue', $result['DATEDUE']);
$insertStmt->bindParam(':pay', $result['PAY']);
$insertStmt->bindParam(':cno', $cno);
$insertStmt->execute();

// rentcar에서 반납된 차량의 정보를 null값으로 바꿈
$updateQuery = "UPDATE rentcar SET daterented = NULL, datedue = NULL, cno = NULL
WHERE licenseplateno = :licenseplateno";
$updateStmt = $conn->prepare($updateQuery);
$updateStmt->bindParam(':licenseplateno', $selectedLicensePlateNo);
$updateStmt->execute();
```

대여에서 결제를 하면 rent_complete로 넘어간다.

select문으로 previoureental에 저장할때와 메일에 사용할 정보인 daterented, modelname, datedue, 총 결제금액을 구해 result에 fetch하였다.

그 후 반납된 차량의 정보를 previousrental에 저장하는데 licenseplateno는 radio에서 받아 온 정보로 하였고 cno는 session에서 가져왔다.

나머지는 아까 result에 저장한 값으로 하였다.

그 후 rentcar에서 반납된 차량의 정보를 null값으로 update해 주었다.

```
// 완료 메일 보내기
$mail = new PHPMailer(true);

$message = "This is return information.\n\n";
$message .= "rent car model: " . $result['MODELNAME'] . "\n";
$message .= "rent car license number: " . $_POST['selected_car'] . "\n";
$message .= "rent start date: " . $result['DATERENTED'] . "\n";
$message .= "rent end date: " . $result['DATEDUE'] . "\n";
$message .= "total price: " . $result['PAY'] . "\n";

$mail->IsSMTP();
$mail->SMTPAuth = true;
$mail->Port = 465;
$mail->SMTPSecure = "ssl";
$mail->CharSet = "EUC-KR";
$mail->Encoding = "base64";

$mail->Host = "smtp.naver.com";
$mail->Username = ""; // 계정
$mail->Password = ""; // 패스워드

$mail->SetFrom('dyj07132@naver.com'); // 보내는 사람 email 주소와 표시될 이름
$mail->AddAddress('dyj07132@naver.com'); // 받을 사람 email 주소와 표시될 이름
$mail->Subject = "Your rental car has been returned!"; // 메일 제목
$mail->MsgHTML($message); // 메일 내용
if($mail->Send()){
    echo "완료";
}
else{
    echo $mail->ErrorInfo;
}
$mail->smtpClose();
```


phpMailer를 사용해 완료 메일을 보냈다.

message에 아까 저장한 반납차량 정보들을 넣었고

smtp를 설정하고 한글도 지원되도록 EUC-KR을 설정했다. 호스트는 네이버를 사용하였고 계정과 비밀번호를 작성한 후 아까 작성한 message를 내용으로 전달하였다.

```
// 이전대여내역에서 정보를 가져온다. 대여 시작일을 기준으로 내림차순으로 한다.
$selectQuery = "SELECT daterented, licenseplateno, datereturned, payment, cno
FROM previousrental
WHERE cno = :cno
ORDER BY daterented desc
";

$selectStmt = $conn->prepare($selectQuery);
$selectStmt->bindParam(':cno', $cno);
$selectStmt->execute();
$result = $selectStmt->fetchAll(PDO::FETCH_ASSOC);

//결과 테이블
if (count($result) > 0) {
    echo "<table class='table table-light'>";
    echo "<tr class='table-dark'><th>대여 날짜</th><th>차량 번호</th><th>반납 날짜</th><th>결제 금액</th><th>고객 번호</th></tr>";

    foreach ($result as $row) {
        echo "<tr>";
        echo "<td>" . $row['DATERENTED'] . "</td>";
        echo "<td>" . $row['LICENSEPLATENO'] . "</td>";
        echo "<td>" . $row['DATEReturned'] . "</td>";
        echo "<td>" . $row['PAYMENT'] . "</td>";
        echo "<td>" . $row['CNO'] . "</td>";
        echo "</tr>";
    }

    echo "</table>";
} else {
    echo "이전 대여 기록이 없습니다.";
}
```

이전대여 내역은 previousrental에서 select로 가져와 foreach문으로 table에 저장하였다.

```
// 매출 순위 쿼리
$query = "SELECT licenseplateno, daterented, datereturned, payment ,
sum(payment) over (partition by licenseplateno) paysum, rank() over (order by paysum desc) rank
FROM (
    SELECT licenseplateno, daterented, datereturned, payment,
    SUM(payment) OVER (PARTITION BY licenseplateno) AS paysum
    FROM previousrental
)";
```

statistic에는 우선 매출 순위를 출력했다.

licenseplateno로 partition해서 총합계를 구한 후 그 총 합계를 rank로 해서 합계가 높은 순으로 rank를 매겼다.

```
// 최근 반납된 차량
$query = "SELECT licenseplateno, daterented, datereturned, payment
FROM (
    SELECT licenseplateno, daterented, datereturned, payment,
    ROW_NUMBER() OVER (ORDER BY datereturned DESC) AS rn
    FROM previousrental
)
WHERE rn <= 5";
```

최근 반납된 차량은 row_number로 datereturned를 내림차순으로 출력했다.

```
//대여 횟수 쿼리
$query = "SELECT rc.modelname, count(*) as modelcount
FROM previousrental pr, rentcar rc
WHERE pr.licenseplateno = rc.licenseplateno
GROUP BY rc.modelname
ORDER BY modelcount desc"
;
```

이때 5개만 출력되도록 하였다.

대여 횟수는 model별로 count해서 출력했고 많이 대여된게 먼저 나오도록 order했다.

```
header__logout = document.getElementById("header__logout");
header__login = document.getElementById("header__login");
header__logout.style.display = "inline";
header__login.style.display = "none";
// 로그인시 로그인창은 사라지고 로그아웃이 나타남
```

로그인 시 로그인창은 사라지고 로그아웃이 나타나도록 js를 만들었다.

```
const logout_submit = document.getElementById("logout_submit");

const logOut = () => {
  if (confirm("로그아웃 하시겠습니까?")) {
    logout_submit.type = "submit"; //true일 때 submit으로 넘김
  }
};

logout_submit.addEventListener("click", logOut);
```

로그아웃을 하면 submit되도록 js를 만들었다.

테이블은 전부 부트스트랩을 사용하였다.