

Providing various information about financial products

KIM EUN BYEOL, 1401110 and CHOI CHAN MI, 1602119

SUMMARY: Various financial information will be given to the users.

Only registered users can use the board.

key words: financial product, bank, deposit, savings, information providing

1. Introduction

Seven out of 10 Korean workers think that it takes more than 10 years to purchase their own home, according to a survey result of Sejong Cyber University. Therefore, it seems that a financial strategy specialized for each person is needed. However, people usually get financial information from their acquaintance, for example friends and family members which can be wrong information. So, we suggests a web application that provides various financial products' information. With this web application, users can collect financial information related to savings and deposit and can decide which financial product will be appropriate for them.

If a user clicks one of the radio button at the below (deposit, savings), she or he can see the financial product classified by each bank or check on the financial product by multikey search. When the user select 'recommendation', web shows the financial product recommendation up to 5 classified by those 3 themes (for college students, for workers, for the elderly). There is a forum where all the users can share her/his financial know-how, experience or small questions. Our web application will provide products of seven banks (신한, 국민, 농협, 우리, KEB 하나, 케이뱅크, 카카오뱅크).

2. Comparison to the existing service

There are a lot financial application that provides financial information to users. For example, Bank Salad service provides the information that is related to the financial product, financing plan, accounting book service and so on. Bank Salad also provides the search service that is conducted by selecting several conditions. However, it

doesn't provide service that searches the financial products like savings and deposit classified by the bank name. If users can see the financial products classified by the bank name, she/he can easily get the information about the service from the certain bank that she is already using.

Also, Bank Salad service doesn't provide board service. Only managers of the company can write a post on the application. What if the 'real' users of the financial product can share their know-how, experience from the certain product? And what if the users can ask small questions in the forum? We think the main strength of our application comes from this part: Forum. We know the power of knowledge. People are well aware of the power that comes from accumulation of knowledge. By sharing their knowledge and experience about finance, users can check their financial plan and get the lively information in our application.

3. Tables and Figures

There are 7 tables, *product*, *product_detail*, *session_info*, *board*, *special_condition*, *recommand*, *user*. *product* and *product_detail* tables imply product lists and are especially used to conditional search. The *recommand* table is used to recommendation for 3 themes. The *special_condition* table is used to show special join condition of financial product.

Table board controls all the data related to the board page. It has 7 columns such as *board_pid* for post numbering, *user_number* for distinguishing a writer and so on.

Table session_info deals with the session of 1 user who has logged in. When a user logs in the system with proper data, session information is inserted into this *session_info* table.


```

<BODY>
<FORM ACTION="receivercommend.php" METHOD="POST">
<table align="center">
<tr><td><h1>테마별 추천</h1></td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="THEME" VALUE="1">20대를 위한 예금, 적금</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="THEME" VALUE="2">직장인을 위한 예금, 적금</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="THEME" VALUE="3">노후대비를 위한 예금, 적금</td></tr>
<tr><td><INPUT TYPE="SUBMIT" VALUE="추천"></td></tr>
</table>
</FORM>
<hr width=100% color="black" align="center" size=20 />
</BODY>

```

Figure 9: User can choose the recommendation with these radio buttons.

[donutzang님 어서오세요.](#)

logout

테마별 추천

- 20대를 위한 예금, 적금
- 직장인을 위한 예금, 적금
- 노후대비를 위한 예금, 적금

추천

Figure 10: Choosing the recommendation page.

```

$THEME=$_POST['THEME'];
$sql="SELECT p.product_name, p.bank_name, p.product_type, p.hits, d.period_month, d.min_interest_rate,
d.max_interest_rate, d.initial_deposit, d.min_monthly_payment, d.max_monthly_payment, r.keyword
FROM recommend AS r LEFT JOIN product AS p ON r.product_id=p.product_id
LEFT JOIN product_detail AS d ON d.product_id=p.product_id AND d.period_month=r.period_month
WHERE (r.keyword = '$THEME')";

```

Figure 11: Join three tables(product table, product_detail table, special_condition table) and pick product informations according to received input (recommend theme keywords) using sql where.

```

echo "<tr><th>".$newArray['product_name']. "</th><th>".$newArray['bank_name']. "</th><th>".$newArray['product_type']. "</th><th>".$newArray['period_month']. "</th><th>".$newArray['min_interest_rate']. "</th><th>".$newArray['max_interest_rate']. "</th><th>".$newArray['initial_deposit']. "</th><th>".$newArray['min_monthly_payment']. "</th><th>".$newArray['max_monthly_payment']. "</th><th>".$newArray['hits']. "</th></tr>";

```

Figure 12: Print product informations (product_name, product_type, period, minimum and maximum of interest rate, initial deposit, minimum and maximum of monthly payment, special condition,hits) in the form of table.

PRODUCT NAME	BANK NAME	PRODUCT TYPE	PERIOD(MONTH)	MIN INTEREST RATE	MAX INTEREST RATE	INITIAL DEPOSIT	MIN MONTHLY PAYMENT	MAX MONTHLY PAYMENT
스프링 우리 자유저축	우리	savings	12	2.2	2.7	0	0	200000
신한 520(에스이클) 적금	신한	savings	6	1.05	1.65	1000	1000	10000000
카카오뱅크 자유저축	카카오뱅크	savings	12	1.5	1.7	1000	1000	3000000
카카오뱅크 정기예금	카카오뱅크	deposit	12	1.5	1.5	1000000	1000000	10000000
포토저 자유저축	케이뱅크	savings	12	1.8	1.8	10000	10000	300000

Figure 13: The result page that shows the recommendation of the financial products

```

<BODY>
<FORM ACTION="receiveviewbybank.php" METHOD="POST">
<table align="center">
<tr><td><h1>은행별 보기</h1></td></tr>
<tr><td><SELECT NAME="TYPE">
<OPTION VALUE="">상품종류선택</OPTION>
<OPTION VALUE="deposit">정기예금</OPTION>
<OPTION VALUE="savings">적금</OPTION>
</SELECT></td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="신한">신한</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="국민">국민</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="농협">농협</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="우리">우리</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="KEB하나">KEB하나</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="케이뱅크">케이뱅크</td></tr>
<tr><td><INPUT TYPE="RADIO" NAME="BANK" VALUE="카카오뱅크">카카오뱅크</td></tr>
<tr><td><INPUT TYPE="SUBMIT" VALUE="보기"></td></tr>
</table>
</FORM>
<hr width=100% color="black" align="center" size=20 />
</BODY>

```

Figure 14: viewbybank and receiveviewbybank php

Figure 15: The page that can choose the bank

```

$BANK=$_POST['BANK'];
$TYPE=$_POST['TYPE'];
$sql="SELECT p.product_name, p.bank_name, p.product_type, p.hits, d.period_month, d.min_interest_rate,
d.max_interest_rate, d.initial_deposit, d.min_monthly_payment, d.max_monthly_payment, s.special_condition
FROM product_detail AS d LEFT JOIN product AS p ON d.product_id=p.product_id
LEFT JOIN special_condition AS s ON d.no_special_condition=s.no_special_condition
WHERE (p.bank_name = '$BANK') AND (p.product_type = '$TYPE')";

```

Figure 16: Join three tables(product table, product_detail table, special_condition table) and pick product informations according to received inputs (bank type and product type) using sql where station

```
echo "<tr><th>".$newArray['product_name']. "</th><th>".$newArray['bank_name']. "</th><th>".$newArray['product_type']. "</th><th>".$newArray['period_month']. "</th><th>".$newArray['min_interest_rate']. "</th><th>".$newArray['max_interest_rate']. "</th><th>".$newArray['initial_deposit']. "</th><th>".$newArray['min_monthly_payment']. "</th><th>".$newArray['max_monthly_payment']. "</th><th>".$newArray['special_condition']. "</th><th>".$newArray['hits']. "</th></tr>";
```

Figure 17 Print product informations(product_name, product_type, period, minimum and maximum of interest rate, initial deposit, minimum and maximum of monthly payment, special condition,hits) in the form of table.

PRODUCT NAME	BANK NAME	PRODUCT TYPE	PERIOD(MONTH)	MIN INTEREST RATE	MAX INTEREST RATE	INITIAL DEPOSIT	MIN MONTHLY PAYMENT	MAX MONTHLY PAYMENT	SPECIAL CONDITION
신한 소득통장 정기예금	신한	deposit	12	1.35	1.35	500000	0	0	없음
신한 보너스 정기예금	신한	deposit	12	1.7	1.8	500000	0	0	없음
농협안 정기예금	신한	deposit	12	1.75	1.75	100000	0	0	없음
농협안 정기예금	신한	deposit	24	1.8	1.8	100000	0	0	없음
농협안 정기예금	신한	deposit	36	1.8	1.8	100000	0	0	없음

Figure 18: The information page classified by bank

```
<?php
session_start(); //세션을 시작한다.
if(isset($_SESSION['is_login'])) //세션값을 설정을 했다면
{if($_SESSION['is_login']){
    // 로그인인 원 상태이다...
    $user_id=$_SESSION['user_id'];
    echo "<p><strong>$user_id</strong>님 어서오세요.";
    echo '<form action="..../logout.php" align="center" >
    <input type="submit" value="logout" />
    </form><br>';
}}
else{ //로그인이 되지 않았을 때..
    $url="..../login.php";
}

<script> alert("로그인이 필요합니다.");
location.replace("<?php echo $url ?>"); //사용자를 되돌려보낸다.
</script>
<?php
}>
```

Figure 19: session code in boardlist.php

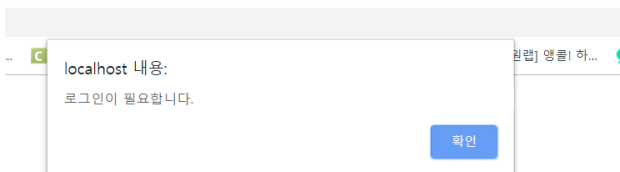


Figure 20: If a unlogged-in user tries to move on to the board.php, she will be alerted and forces to move back. All the php file includes this kind of job, so that the whole pages recognize who the user is.



login

은행별 보기 조건

Figure 21 If a user uses our service without login, a button for login will be activated.

당신을 위한 금융 정보

donutzzang님 어서오세요.

logout

게시판

글 쓰기

No.	제목	작성자	작성일
17	MySQLTest	donutzzang	2019-11-25
12	MySQL	jeung33	2019-11-24
11	인생	jeung33	2019-11-24
4	친구한테 빨리 졸업하세요	trumperpoppy	2019-09-01
3	최고의 상품 추천합니다	ninetapronounce	2019-09-01
2	Great IDEA	donutcowbird	2019-09-01
1	정말 멋진 상품 추천	rebelexitable	2019-09-01

Figure 22: board.php page shows a small forum.

```
$rowPerPage=500;
$sql="select board_pid, user_number, product_id, contents, hits,
date, contents_title from board order by board_pid desc limit ".$rowPerPageStart.", ".$rowPerPage.";";
$result=mysqli_query($db, $sql);
```

Figure 23: In **boardlist.php**, the sql query brings the whole information of the board database.

```
while($row=mysqli_fetch_array($result)){
    $user_id="select user_id from user where user_number=".$row["user_number"].";";
    $query=mysqli_query($db, $user_id);
    $result_id=mysqli_fetch_array($query);

    ?>
    <caption>포토 만듭니다.</caption>

    <tr>
    <td><?php echo $row["board_pid"] ?></td>
    <td><a href="board_detailed_see.php?id=<?php echo $row["board_pid"] ?>">
    <?php echo $row["contents_title"]; ?> </a></td>
    <td><?php echo $result_id["user_id"]; ?></td>
    <td><?php echo $row["date"]; ?></td>
```

Figure 24: And then the while loop will be repeatedly

prints the board_pid, contents_title, user_id and date which is from the board database. If user clicks the contents_title, she can read the post in detail by calling the **board_detailed_see.php**.



Figure 25: This is the page after clicking a certain post title. If a user is the writer of this post, the delete button is activated.

```
//이제 조회수를 하나 추가해준다
$hit="update board set hits=hits+1 where board_pid=".$number;
$result_hits=mysqli_query($db, $hit);
```

Figure 26: Also, the hits for the post will be updated.

```
if($user_id['user_id']==$_SESSION['user_id']){
    //지금 접속해있는 사람이랑 글쓴 사람이랑 일치하면 delete버튼 활성화
    echo "<br><form action='delete.php' align='center'>
    <input type='submit' name='delete' id='delete' value='Delete This Post' /> <br>";
}
```

Figure 27: In **board_detailed_see.php**, use can read the post in detail and writer's identification is compared with session's id.



Figure 28: Post delete event

```
if(isset($_SESSION['user_id'])){
    $sql1="select user_number from user where user_id='".$_SESSION['user_id']."'";
    $query=mysqli_query($db, $sql1);
    $fet=mysqli_fetch_assoc($query);
    $sql="delete from board where user_number=".$fet['user_number'];
    $query=mysqli_query($db, $sql);
    echo "<script>window.alert('글이 삭제 되었습니다.');";
}
```

Figure 29: When clicking the delete button, board

database will delete the this post's data. Alert will be popped up, and then a user will be sent to the **boardlist.php**.

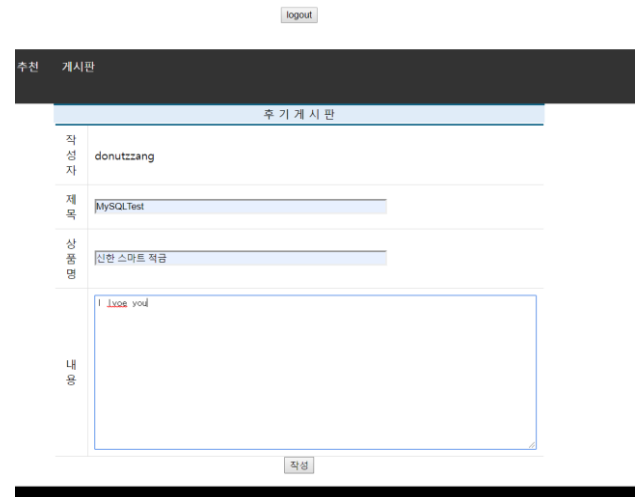


Figure 30: If registered user wants to post a new one in the board, she can easily insert data.

```
$user_id=$_SESSION['user_id'];
$title=$_GET['title'];
$product_name=$_GET['product_name'];
$content=$_GET['content'];
$sql="select user_number from user where user_id='".$_user_id."'";
$result=mysqli_query($db, $sql);
$fet=mysqli_fetch_row($result);

$today=date("Y-m-d");
```

Figure 31: In **write.php**, all the writing materials from a post form will be delivered. Also, date information is to be added.

```
$sql="insert into board(user_number, product_id,contents_title, contents, hits, date)
values('".$fet[0]."',".$fet[2[0]."', '".$title."', '".$content."',0, '".$today."')";
$result=mysqli_query($db, $sql);
if(!$result) echo "입력 실패:".$mysqli_error($db);
else{
    echo "<script> alert('UploadOK.');";
}
```

Figure 32: Insert new user to the database

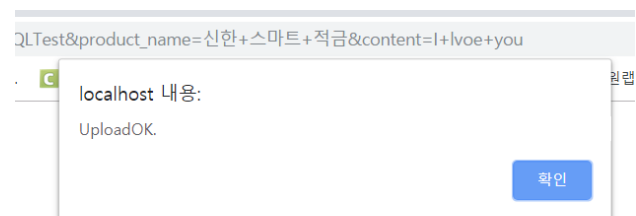


Figure 33: If everything's okay and no error, the data will be inserted by \$result and alert message that says Upload ok will be popped up.

Figure 34: login Page in main

Figure 35: If a unregistered user tries to log in the system, alert will be popped up.

```
session_start();
$id=$_POST['id'];
$pw=$_POST['password'];

$sql="select * from user where user_id='".$id."' and user_passwd='".$pw."'";
$result=mysqli_query($db,$sql);
$num=mysqli_num_rows($result); //몇개인지 인트형 숫자 넘어 저장

$fetch_assoc=mysqli_fetch_assoc($result); //배열로 반환함.
```

Figure 36: In **logincheck.php**, id and password are delivered. With these parameters, the sql query will conduct the check if the user's input data(id and password) corresponds to the user database.

```
else{
    echo "<script> alert('아이디 또는 패스워드를 확인해주세요'); </script>";
}
```

Figure 37 : If either data is not correct, the alert will be popped up. If there is no problem, log-in will be well along.

Figure 38: Register Page

```
$id=$_POST['joinid'];
$password=( $_POST['joinpw']);
$password2=$_POST['joinpw2'];
$sex=$_POST['sex'];
$birthday=$_POST['birthday'];
```

Figure 39: In **joinokcheck.php**, parameters is delivered for checking the validity.

```
$check_id="select * from user where user_id='".$id."'";
$result=mysqli_query($db,$check_id);
$num=mysqli_num_rows($result);
if($num>0){ //회원 데이터베이스에서 같은 아이디를 발견했을 때
    echo "<script> window.alert('이미 있는 아이디 입니다.');"history.back(-2); </script> ";
}
```

Figure 40: If a user tries to register with her id that is already registered, the alert will be popped up and the user has to re-try the register.

```
else{//새로운 아이디일 때
    if($password!=$password2){ //확인 비밀번호가 일치하지 않을 때
        echo "<script> window.alert('확인 비밀번호는 같아야 합니다');" history.back(-2); </script>";
    }
    else{ //확인 비밀번호에 문제가 없어 같게 입력했을 때 회원 데이터에 추가해준다.
        $sql="INSERT INTO user(user_id, user_passwd, user_birth, user_gender)
        VALUES('".$id."','".$password."','".$birthday."','".$sex."','".$sex."'";
        $insert=mysqli_query($db,$sql);
        echo "<script> window.alert('회원가입 완료. 로그인 해주세요.');"</script>";
    }
}
```

Figure 41: User's id input is okay and the both of the password&passwordcheck parameters are the same, new user's data will be inserted to the user database .

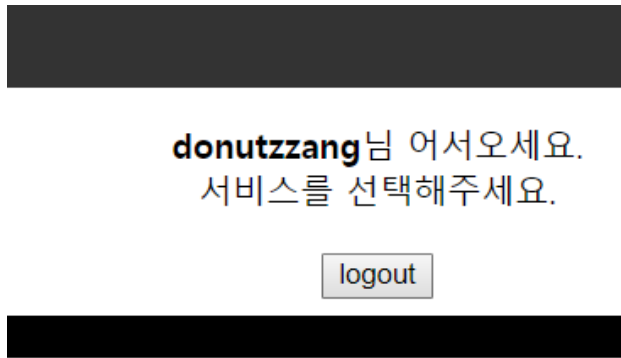


Figure 42: Logout function is in the system. If a user clicks the logout button, she can log out.

```

session_start();
$user_id=$_SESSION['user_id'];
echo "<script> alert('로그아웃됩니다.');";
$sql="delete from session_info where user_id='".$user_id."'";
$result=mysqli_query($db, $sql);
setcookie(session_name(), "", time()-99999999);
session_destroy();

header('Location: ./login.php');

```

Figure 43 : Session is started and the whole data about this user who wants to log out will be deleted from the session_info database. Also, the cookie is initialized. A user will be sent to the main page.

5. Conclusion

Until now, we looked at how the system works with PHP and HTML. With this application, users can easily get the financial information by choosing the specific conditions and share their knowledge in the forum. Forum is only for the registered users.

However, there is small problem with dealing with session. When a user doesn't log out properly like just shut down her/his window, the session information cannot be updated properly. We are going to solve this problem

6. References

- [1] 『생활코딩! HTML+CSS+자바스크립트』, 이고잉, 위키북스
- [2] 『생활코딩! PHP+MySQL』, 이고잉, 위키북스
- [3] 『PHP 프로그래밍 입문』, 황재호, 한빛아카데미