

인터넷 프로그래밍

개인 프로젝트 2

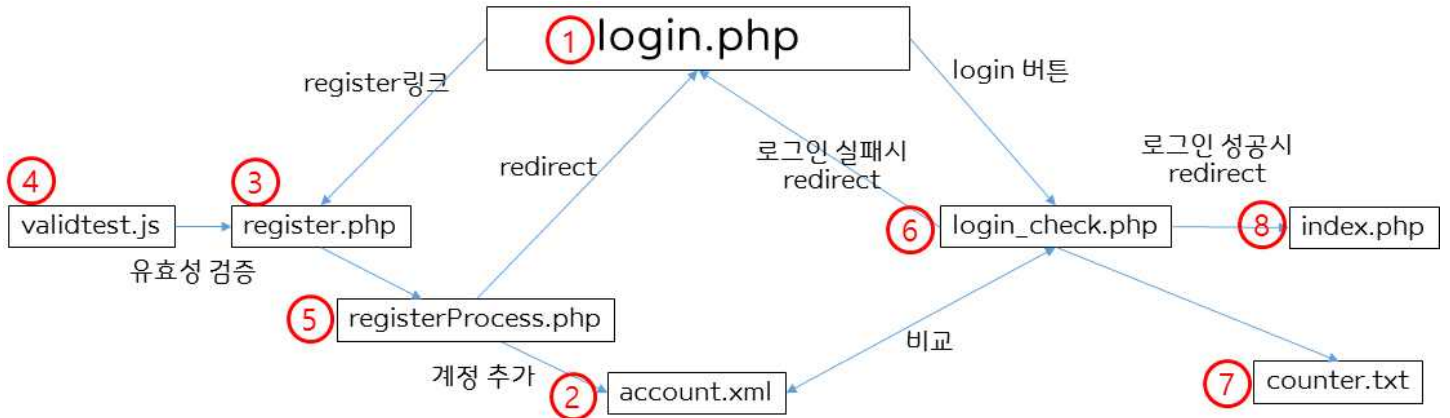
조영재

컴퓨터과학과 2013147513

페이지 동작설명 p2

스펙 요구사항 위치 p7

페이지 동작 설명



① login.php

로그인 페이지인 login.php는 두 개의 페이지와 연결되어 있다. register.php와 login_check.php이다. 이중 register.php는 register링크를 눌렀을 때 이동하는 회원가입 페이지이며, login_check.php는 login버튼을 눌렀을 때 로그인을 수행하는 페이지이다. 로그인 데이터는 데이터베이스 대신에 xml파일로 구현하였으며, account.xml이 그것이다.

② account.xml

```

▼ <database>
  ▼ <account>
    <id>admin</id>
    <password>1q2w</password>
    <visit>17</visit>
  </account>
  ▼ <account>
    <id>4nurl1205</id>
    <password>dk2121</password>
    <birthday>921205</birthday>
    <phone>010-3441-5131</phone>
    <email>4nurl1205@naver.com</email>
    <visit>35</visit>
  </account>
  ▶ <account>...</account>
  ▶ <account>...</account>
  ▶ <account>...</account>
  ▶ <account>...</account>
  ▶ <account>...</account>
</database>
  
```

◀ account.xml은 왼쪽과 같이 구성하였다.

첫 번째 account는 id가 admin인 관리자 권한의 계정이고, 그 이후부터의 account가 일반 사용자의 계정이다. 각각의 account 밑에는 6개의 children node가 존재하는데 순서대로 id, 비밀번호, 생일, 전화번호, 이메일, 방문횟수이며, 방문횟수를 제외한 데이터 들은 register.php에서 validtest.js를 통해 유효성 검증을 거친 데이터가 registerProcess.php를 통해 account.xml문서에 기록되게 된다. 방문횟수는 login_check.php를 통해 기록된다.

③, ④ register.php 및 validtest.js

register.php는 회원의 정보를 입력받아 데이터를 account.xml에 저장하는 회원가입 페이지이며, 유효성 검증을 하는 자바스크립트가 validtest.js이다.

기본적으로 text입력창에서 포커스가 벗어나면 실행되도록 input에 onblur이벤트를 등록하였으며, 이를 통해 validtest.js에 등록된 함수중 각각의 데이터에 맞는 유효성 검증 함수를 호출하게 된다. 그리고 모든 input들은 checkall()함수를 호출하게 되는데, 이를 통해 기본적으로 disabled되어 있는 제출 버튼을, 6가지의 유효성 검증을 모두 통과할 경우 disable="false"가 되어 누를 수 있도록 하였다. (**register.php line 98~135**)

회원으로부터 입력받는 정보는 아이디, 비밀번호, 비밀번호 확인, 생년월일, 휴대전화, 이메일이며 이 6가지의 유효성 검증은 다음과 같이 진행하였다.

1) 아이디 (**validtest.js line 23~74**)

- 유효성 검증을 통과할 경우 valid_ID값을 true로 설정하고 그렇지 않을 경우 false로 한다.
- regular expression을 이용하여 6~10자의 영어 대소문자, 숫자 특수기호(_)만 있는 경우 중, 영어 대문자가 없거나, 숫자가 없는 경우만 제외하도록 하였다.
- **(추가기능)** 이미 존재하는 아이디를 입력할 수도 있으므로, ajax를 통하여 account.xml문서를 가져온 뒤, <id>태그의 모든 값들과 비교하여, matching되는 동일한 문자열이 있을 경우 이미 존재하는 것으로 판단하여 valid_ID를 false로 하고, checkid함수를 실행하지 않는다.

아이디:	<input type="text" value="A1234"/>	유효하지않은 아이디입니다!	6자리 미만
아이디:	<input type="text" value="A1234_"/>	사용가능한 아이디입니다.	OK
아이디:	<input type="text" value="ABcdef"/>	유효하지않은 아이디입니다!	숫자 없음
아이디:	<input type="text" value="a12345"/>	유효하지않은 아이디입니다!	대문자 없음
아이디:	<input type="text" value="A12345@"/>	유효하지않은 아이디입니다!	_를 제외한 특수문자
아이디:	<input type="text" value="A1234567890"/>	유효하지않은 아이디입니다!	10자리 초과
아이디:	<input type="text" value="4nurl1205"/>	존재하는 아이디입니다!	이미 등록된 아이디

2) 비밀번호 및 비밀번호 확인 (**validtest.js line 77~147**)

- 유효성 검증을 통과할 경우 valid_PWD, valid_PWDcheck값을 true로 설정하고 그렇지 않을 경우 false로 한다.
- regular expression을 이용하여 동일한 3개의 문자가 반복될 수 없도록 하였다.
- ascii코드를 이용하여 연속한 3자리 숫자나 문자가 오지 못하도록 하였다.
- **(추가기능)** 비밀번호 확인을 통하여 두 번째 입력한 비밀번호와 첫 번째 입력한 비밀번호가 다를 경우 valid_PWDcheck를 false로 하여 가입하지 못하도록 하였다.

비밀번호:	<input type="text" value="ab2"/>	더 복잡한 비밀번호로 설정해 주세요!	4자리 미만
비밀번호:	<input type="text" value="12a@#zibhte2"/>	조금 짧은 비밀번호로 설정해 주세요!	10자리 초과
비밀번호:	<input type="text" value="1234"/>	더 복잡한 비밀번호로 설정해 주세요!	연속한 숫자
비밀번호:	<input type="text" value="cba133"/>	더 복잡한 비밀번호로 설정해 주세요!	연속한 문자
비밀번호:	<input type="text" value="ccc14ctf"/>	더 복잡한 비밀번호로 설정해 주세요!	연속한 동일문자
비밀번호:	<input type="text" value="banana"/>	더 복잡한 비밀번호로 설정해 주세요!	3개이상 동일문자
비밀번호:	<input type="text" value="4c@ttx"/>	사용가능한 비밀번호입니다.	OK

비밀번호:	<input type="text" value="4c@ttx"/>	사용가능한 비밀번호입니다.	일치하지 않는 비밀번호
비밀번호 확인:	<input type="text" value="4cdfeg"/>	비밀번호 가 일치하지 않습니다.	
비밀번호:	<input type="text" value="4c@ttx"/>	사용가능한 비밀번호입니다.	일치하는 비밀번호
비밀번호 확인:	<input type="text" value="4c@ttx"/>	비밀번호 확인.	

위에서는 비밀번호를 보여주기 위해 text를 사용하였으며 실제로는 password타입으로 지정하였음

3) 생년월일, 휴대전화, 이메일

- 위의 항목들의 유효성검증을 통과하면 각각 valid_BDAY, valid_PHONE, valid_EMAIL을 true로 만들고 그렇지 않다면 false로 한다.
- 각각의 형식은 text input에 placeholder로 표시해 놓았으며 다음과 같다
 - 생년월일 : 6자리 숫자, yymmdd. 월과 일의 유효성 또한 검증하였다. 31일이 있는 달과, 30일까지만 있는 달을 구분하여 놓았으며, 2월달은 무조건 29일까지 있는 것으로 하였다. (**validtest.js line 151~203**)
 - 휴대전화 : 000 - 0000 - 0000 의 형식을 갖도록 정해놓았다. (**validtest.js line 206~224**)
 - 이메일 : xyz@abc.def의 @와 @뒤의 .이 있는지 검사하도록 하였다. (**validtest.js line 227~244**)

4) checkall()함수 (**validtest.js line 10~17**)

- 이상에서 정의한 모든 valid_* 값이 true가 될 경우 제출버튼을 enable시키는 함수이다.

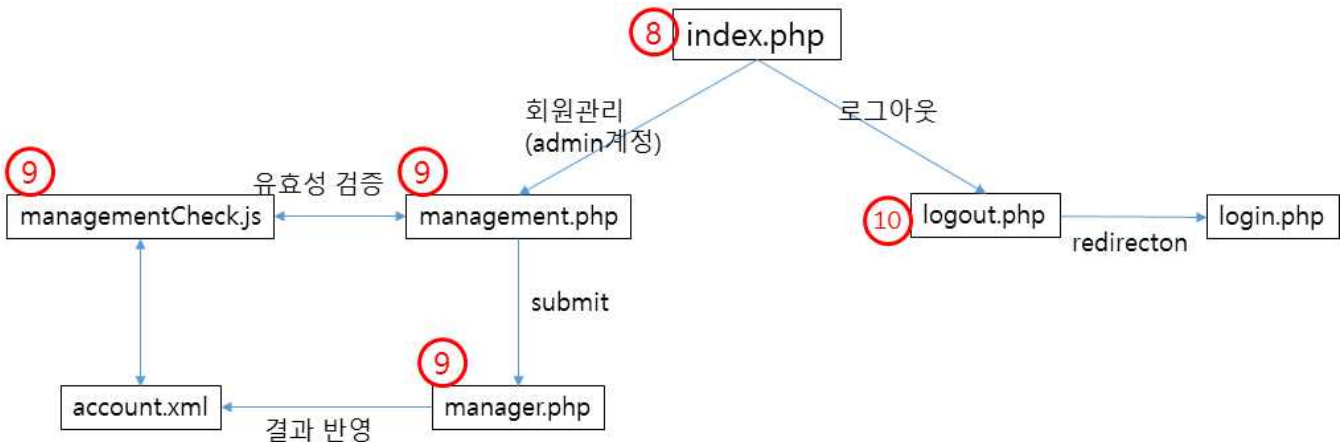
⑤ registerProcess.php

registerProcess.php는 register.php에서 유효성검증을 마친 데이터들을 받아, account.xml에 추가하는 코드이다. register.php에서 form을 POST방식으로 전송하였으므로 \$_POST superglobal 변수를 통하여 값을 받아온다. account.xml트리구조를 구현하는데 있어서, 단말노드인 text노드(데이터가 저장되는 노드)부터 생성한 후, 상위 노드에 붙여가는 식으로 구현한다.

⑥, ⑦ login_check.php 및 counter.txt

- login_check.php에서는 login.php에서 사용자가 입력한 아이디와 비밀번호를 받아서 account.xml파일에 일치하는 아이디와 패스워드가 존재하는지 검사한다. 만약 존재하지 않을 경우 다시 login.php로 redirection 시키고, 일치하는 것이 있을 경우 메인페이지인 index.php로 redirection 시킨다.
- 아이디와 비밀번호가 일치하여 로그인에 성공할 경우, account.xml의 visit를 1 증가시키고, alert를 통하여 몇 번째 방문인지 확인시켜 준다. 이후 index.php로 redirection시킨다. 아이디와 비밀번호가 틀려 로그인에 실패할 경우, 로그인 실패 alert를 알려준 후 다시 login.php로 redirection 시켜준다.
- 또한 로그인에 성공할 경우, 페이지 방문자 카운터를 증가시킨다. 이는 json을 통하여 구현하였다. 방문자 카운터에 대한 정보는 counter.txt파일에 저장하며 여기에는 세가지 변수가 존재하는데, visited는 총 누적 방문자 숫자이고, today는 오늘 방문자 숫자, date는 현재 로그인시의 시간을 저장하는 변수이다. 만약 로그인 시에 counter.txt에 저장된 date와 다르다면, 다음날에 방문한 것이므로, today카운터를 1로 초기화 시켜준다. (**login_check.php line 18~27**)

⑧ index.php



사이트의 메인페이지이다. SESSION을 확인하여, 로그인 되어있지 않을 경우 자동으로 로그인 페이지인 login.php로 이동하게 만든다. 로그인에 성공할 경우 이 페이지로 자동으로 redirection된다.

일반 사용자일 경우에는 표시되지 않지만, 아이디가 admin인 아이디로 로그인 할 경우, 관리자 권한이 생기며, management라는 항목이 메뉴에 추가된다. 이를 통해 회원의 비밀번호를 변경할 수 있다.

일반 사용자 메뉴 모습



admin아이디 메뉴 모습



또한, 로그인 시에 업데이트된 방문횟수를 표시해 주며(index.php line 93~98), LOGIN글씨를 SESSION을 통해 로그인이 되어있음을 확인하면 LOGOUT으로 바뀌서 표시해 준다.(index.php line 101~107)LOGOUT버튼은 logout.php로 연결되며, 생성되었던 모든 세션을 제거하며, 이후에 login.php로 redirection시킨다.

⑨ management.php, managementCheck.js, manager.php

※관리자 id: admin, 비밀번호: 1q2w

management.php는 index.php에서 admin으로 로그인하였을 경우만 접근할 수 있다. 이 페이지의 주 목적은 회원아이디의 비밀번호를 변경해주는 관리자의 기능이다. managementCheck.js는 입력값의 유효성 검증을 하는 자바스크립트이며, manager.php는 account.xml에 변경내용을 바꾸어 주는 파일이다.

관리자가 입력해야 하는 정보는 비밀번호를 바꾸려는 아이디와 새로 바꿀 비밀번호, 그리고 비밀번호 확인을 위해 한번 더 비밀번호를 입력해야 한다. 이전의 register.php와 validtest.js와 비슷한 과정으로 동작한다. onblur이벤트를 통해서 입력창에서 포커스가 벗어날 때 유효성 검증을 하며, 그 조건은 다음과 같다.

1) 아이디

- validtest.js에서는 account.xml에 아이디가 존재하면 안되었었지만, 반대로 이 경우에는 account.xml에 비밀번호를 바꾸고자 하는 아이디가 존재해야 valid_ID = true가 되도록 설정하였다.

2) 비밀번호, 비밀번호 확인

- 바꾸고자 하는 비밀번호 또한 과제 스펙상의 조건을 만족해야 하므로 valdtest.js와 동일하게 설정하였다.

3) checkall()함수

- 마찬가지로 아이디, 비밀번호, 비밀번호 확인 3가지 조건이 모두 만족할때만 변경버튼을 활성화 시키도록 하였다.

변경버튼을 누르면 입력한 데이터는 manager.php파일로 전송되고, 그에 따라서 account.xml파일을 알맞게 변경하도록 한다.

⑩ logout.php

로그아웃 버튼은 현재 생성되어있는 모든 SESSION을 삭제하며, login.php로 redirection된다.

스펙 요구사항 위치

- 유효성 검증

유효성 검증		해당 파일	줄수
아이디	존재하는 아이디	validtest.js	40~47
	길이제한		60
	1대문자, 1숫자 제한		61
비밀번호	길이제한		88~89
	연속한 동일문자 제한		92~93
	동일문자 3개 제한		98~105
	연속한 문자 제한		108~116
비밀번호 확인			134~147
생일			151~203
전화번호			206~224
이메일			227~244

- PHP 기능구현

종류	해당 파일	줄수
비 로그인시 로그인페이지로 이동	index.php	171~178
로그인시 ○○번째 방문을 환영합니다	login_check.php	10~15, 34~40
회원관리를 위한 관리자 추가기능	management.php	118~131(form양식)
	managementCheck.js	전체(유효성 검증)
	manager.php	11~25(xml문서 수정)

- 기타 요구사항 구현

종류	해당파일	줄수
HTML DOM API	managementCheck.js	8, 34~35 등
XML DOM API	validtest.js	37
JavaScript event	register.php	102, 107 등등..(onblur event)
AJAX	validtest.js	26~48
XML	account.xml	XML파일
	registerProcess.php	XML노드 생성 파일
JSON	counter.txt	전부(JSON파일)
	index.php	93~98