



화면 구현

응용 프로그래밍 및 화면 구현

심규영

목차

- 1)게시판 목록 상단 헤더 회원정보 링크 추가
- 2)회원정보 비밀번호 인증
- 3)회원정보 설정
- 4)기능 동작 및 시연 화면

1) 게시판 목록 상단 헤더 회원정보 링크 추가

```
<header>
  <h3>Board System v1.0</h3>
  <p>
    <span>${ sessUser.nick }</span>님 반갑습니다.
    <a href="/Jboard2/user/info.do">[회원정보]</a>
    <a href="/Jboard2/user/logout.do">[로그아웃]</a>
  </p>
</header>
```

사람이름님 반갑습니다. [회원정보] [로그아웃]

2) 회원정보 비밀번호 인증

```
<div>
  <a href="/Jboard2/list.do" class="btn btnCancel">취소</a>
  <a href="/Jboard2/user/myInfo.do" class="btn btnNext">다음</a>
</div>
```

localhost:8081 내용:
비밀번호가 일치하지 않습니다.

확인

```
<script>
$(function(){
  $('.btnNext').click(function(){
    const pass = $('input[name=pass]').val();
    const uid = "${ sessUser.uid }";

    $.ajax({
      url: '/Jboard2/user/info.do',
      method: 'post',
      data: {"uid":uid,"pass":pass},
      dataType: 'json',
      success: (data)=>{
        if(data.result > 0) {
          location.href = "/Jboard2/user/myInfo.do";
        } else {
          alert('비밀번호가 일치하지 않습니다.');
```

```
@WebServlet("/user/info.do")
public class InfoController extends HttpServlet {

    private static final long serialVersionUID = 1L;
    private UserService service = UserService.INSTAN

    @Override
    protected void doGet(HttpServletRequest req, Http
        req.getRequestDispatcher("/user/info.jsp").fo
    }

    @Override
    protected void doPost(HttpServletRequest req, Ht
        String uid = req.getParameter("uid");
        String pass = req.getParameter("pass");

        int result = 0;

        result = service.selectUserCheck(uid, pass);

        JSONObject json = new JSONObject();
        json.addProperty("result", result);

        PrintWriter writer = resp.getWriter();
        writer.print(json.toString());
    }
}
```

- 1) 취소 클릭시 게시판 목록 이동
- 2) 비밀번호 입력 후 다음 버튼 클릭(AJAX) 이용하여 회원 인증
- 3) 비밀번호가 일치하지 않으면 '비밀번호가 일치하지 않습니다.' 대화상자(Alert) 출력
- 4) 비밀번호가 일치하면 회원정보 수정 페이지 이동

3) 회원정보 설정

Board System v2.0

- 1) 기본적인 사용자 정보 출력하기
- 2) 별명 중복체크, 이메일 인증, 우편번호 검색 기능 구현하기

회원정보 설정

아이디	a10002
비밀번호	<input type="text" value="비밀번호 입력"/> <small>비밀번호 입력시 비밀번호 변경</small>
비밀번호 확인	<input type="text" value="비밀번호 입력 확인"/>
회원가입일	2022-11-22 19:48:21

개인정보 수정

이름	<input type="text" value="사람이름"/>
별명	<small>공백없는 한글, 영문, 숫자 입력</small> <input type="text" value="사람이름"/> <small>중복확인</small>
이메일	<input type="text" value="me03454@gmail.com"/> <small>인증번호 받기</small>
휴대폰	<input type="text" value="010-2134-1111"/>
주소	<input type="text" value="46990"/> <small>우편번호찾기</small> <input type="text" value="부산 사상구 가야대로 1"/> <input type="text" value="무슨무슨동 무슨무슨호"/>
회원탈퇴	<small>탈퇴하기</small>

취소

회원수정

회원정보 설정

아이디	a10003
비밀번호	<input type="text" value="비밀번호 입력"/> <small>비밀번호 입력시 비밀번호 변경</small>
비밀번호 확인	<input type="text" value="비밀번호 입력 확인"/>
회원가입일	2022-12-05 17:54:04

개인정보 수정

이름	<input type="text" value="사람이름"/>
별명	<small>공백없는 한글, 영문, 숫자 입력</small> <input type="text" value="사람별명1"/> <small>중복확인</small> <small>사용 가능한 별명입니다.</small>
이메일	<input type="text" value="me03454@gmail.com"/> <small>이메일이 인증 되었습니다.</small> <small>인증번호 받기</small> <input type="text" value="226992"/> <small>확인</small>
휴대폰	<input type="text" value="010-3234-1002"/>
주소	<input type="text" value="우편번호"/> <small>우편번호찾기</small> <input type="text" value="주소 검색"/> <input type="text" value="상세주소 입력"/>
회원탈퇴	<small>탈퇴하기</small>

Daum Postcode Service - Chrome

about:blank

tip
아래와 같은 조합으로 검색을 하시면 더욱 정확한 결과가 검색됩니다.

도로명 + 건물번호
예) 판교역로 235, 제주 첨단로 242

지역명(동/리) + 번지
예) 삼성동 681, 제주 영평동 2181

지역명(동/리) + 건물명(아파트명)
예) 분당 주공, 연수동 주공3차

사서함명 + 번호
예) 분당우체국사서함 1~100

Powered by kakao | 우편번호 서비스 안내

3) 회원정보 설정

```
public static final String DELETE_USER =
    "UPDATE `board_user` SET "
    + " `pass` = NULL,"
    + " `name` = NULL,"
    + " `nick` = NULL,"
    + " `email` = NULL,"
    + " `hp` = NULL,"
    + " `grade` = 0,"
    + " `zip` = NULL,"
    + " `addr1` = NULL,"
    + " `addr2` = NULL,"
    + " `regip` = '',"
    + " `rdate` = '1999-01-01 00:00:10',"
    + " `sessId` = NULL,"
    + " `sessLimitDate` = NULL,"
    + " `wdate` = NOW()"
    + "WHERE `uid`=?";

public int deleteUser(String uid) {
    int result = 0;
    try {
        logger.info("deleteUser...");
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.DELETE_USER);
        pstmt.setString(1, uid);
        result = pstmt.executeUpdate();
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    return result;
}

// 회원 탈퇴 기능
$('button[name=UserDelete]').click(()=>{
    const check = confirm('정말 삭제 하시겠습니까?');

    const uid = $('input[name=uid]').val();

    if(check == true) {
        $.ajax({
            url: "/Jboard2/user/userDelete.do",
            method: "get",
            data: {"uid":uid},
            dataType: 'json',
            success: (data)=>{
                if(data.result > 0) {
                    alert('회원 탈퇴가 완료 되었습니다.');
```

3) 탈퇴하기 기능 구현

- 탈퇴시 회원 grade = 0 으로 수정
- 탈퇴날짜 입력받기(탈퇴날짜 컬럼 : wdate 추가)
- 탈퇴 완료 후 로그인으로 이동

```
protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
    String uid = req.getParameter("uid");
    HttpSession sess = req.getSession();

    int result = service.deleteUser(uid);

    // 세션 해제
    sess.removeAttribute("sessUser");
    sess.invalidate();

    // 쿠키 삭제
    Cookie cookie = new Cookie("SESSID", null);
    cookie.setPath("/");
    cookie.setMaxAge(0);
    resp.addCookie(cookie);

    JsonObject json = new JsonObject();
    json.addProperty("result", result);

    resp.getWriter().print(json.toString());
}
```

grade	wdate
0	2022-12-05 17:35:44
0	2022-12-05 17:52:37
2	(NULL)
2	(NULL)
2	(NULL)
0	2022-12-05 17:12:56
0	2022-12-05 17:22:14

3) 회원정보 설정

```
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
    IOException {
    userVO vo = new userVO();
    vo.setUid(req.getParameter("uid"));
    vo.setPass(req.getParameter("pass"));
    vo.setName(req.getParameter("name"));
    vo.setNick(req.getParameter("nick"));
    vo.setEmail(req.getParameter("email"));
    vo.setHp(req.getParameter("hp"));
    vo.setZip(req.getParameter("zip"));
    vo.setAddr1(req.getParameter("addr1"));
    vo.setAddr2(req.getParameter("addr2"));

    int result = 0;

    if(vo.getPass().equals("") || vo.getPass() == null ) result = service.updateUserNotPass(vo);
    else result = service.updateUser(vo);

    JsonObject json = new JsonObject();
    json.addProperty("result", result);

    resp.getWriter().print(json.toString());
}
```

비밀번호, 이름, 별명, 메일, 휴대폰의 유효성 검사는
재활용 함

4) 회원 수정하기 기능 구현

- 비밀번호를 입력하지 않으면 기존 비밀번호 유지
- 이름, 별명, 이메일, 휴대폰, 주소 수정하기
- 회원수정 후 게시판 목록으로 이동

```
const uid = $('input[name=uid]').val();
const pass = $('input[name=pass2]').val();
const name = $('input[name=name]').val();
const nick = $('input[name=nick]').val();
const email = $('input[name=email]').val();
const hp = $('input[name=hp]').val();
const zip = $('input[name=zip]').val();
const addr1 = $('input[name=addr1]').val();
const addr2 = $('input[name=addr2]').val();

$.ajax({
    url: '/Jboard2/user/myInfo.do',
    method: 'post',
    data: { "uid":uid,"pass":pass,"name":name,"nick":nick,"email":email,
    "hp":hp,"zip":zip,"addr1":addr1,"addr2":addr2},
    dataType: 'json',
    success: (data)=>{
        if(data.result > 0) {
            alert('회원정보를 수정 했습니다.');
```

3) 회원정보 설정

```
let isPassOk      = true;
let isNameOk      = true;
let isNickOk      = true;
let isEmailOk     = true;
let isEmailAuthOk = true;
let isHpOk        = true;

let isEmailCheckSend = false; // 이메일 인증코드를 보냈는가?
let emailChecking    = false; // 이메일 검사중 체크
let emailCode        = 0; // 이메일 인증 코드

// 비밀번호 검사하기
$('input[name=pass2]').change(function(){
    let pass1 = $('input[name=pass1']").val();
    let pass2 = $('input[name=pass2']").val();

    if(pass2.match(rePass)) {
        if(pass1 == pass2) {
            isPassOk = true;
            $('.passResult').css('color', 'green').text('사용하실 수 있는 비밀번호 입니다. ');
        } else {
            isPassOk = false;
            $('.passResult').css('color', 'red').text('비밀번호가 일치하지 않습니다. ');
        }
    } else {
        isPassOk = false;
        $('.passResult').css('color', 'red').text('비밀번호는 숫자,영문,특수문자 포함 8자리 이상 이어야 합니다. ');
    }
});
```

1. UserInfo.js 를 만들어서 유효성 검사 재활용
2. Focusout을 전부 change로 변경

3) 회원정보 설정

```
public int updateUserNotPass(userVO vo) {
    int result = 0;
    try {
        logger.info("updateUserNotPass...");
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.UPDATE_USER_NOT_PASS);
        pstmt.setString(1, vo.getName());
        pstmt.setString(2, vo.getNick());
        pstmt.setString(3, vo.getEmail());
        pstmt.setString(4, vo.getHp());
        pstmt.setString(5, vo.getZip());
        pstmt.setString(6, vo.getAddr1());
        pstmt.setString(7, vo.getAddr2());
        pstmt.setString(8, vo.getUid());
        result = pstmt.executeUpdate();
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("updateUserNotPass result : "+result);
    return result;
}
```

```
public int updateUser(userVO vo) {
    int result = 0;
    try {
        logger.info("updateUser...");
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.UPDATE_USER);
        pstmt.setString(1, vo.getPass());
        pstmt.setString(2, vo.getName());
        pstmt.setString(3, vo.getNick());
        pstmt.setString(4, vo.getEmail());
        pstmt.setString(5, vo.getHp());
        pstmt.setString(6, vo.getZip());
        pstmt.setString(7, vo.getAddr1());
        pstmt.setString(8, vo.getAddr2());
        pstmt.setString(9, vo.getUid());
        result = pstmt.executeUpdate();
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("updateUser result : "+result);
    return result;
}
```

4) 회원 수정하기 기능 구현

- 비밀번호를 입력하지 않으면 기존 비밀번호 유지

- 이름, 별명, 이메일, 휴대폰, 주소 수정하기

- 회원수정 후 게시판 목록으로 이동

```
public static final String UPDATE_USER =
    "UPDATE `board_user` SET "
    + "`pass` = SHA2(?, 256), "
    + "`name` = ?, "
    + "`nick` = ?, "
    + "`email` = ?, "
    + "`hp` = ?, "
    + "`zip` = ?, "
    + "`addr1` = ?, "
    + "`addr2` = ? "
    + "WHERE `uid` = ?";

public static final String UPDATE_USER_NOT_PASS =
    "UPDATE `board_user` SET "
    + "`name` = ?, "
    + "`nick` = ?, "
    + "`email` = ?, "
    + "`hp` = ?, "
    + "`zip` = ?, "
    + "`addr1` = ?, "
    + "`addr2` = ? "
    + "WHERE `uid` = ?";
```

4)기능 동작 및 시연 화면

Board System v1.0

사람별명님 반갑습니다. [회원정보] [로그아웃]

글목록

제목 키워드, 글쓴이 검색

검색

번호	제목	글쓴이	날짜	조회
1	sdfadf [0]		22-12-05	2

1

글쓰기

비밀번호 확인

비밀번호	<input type="password" value="비밀번호 입력"/>
------	--

회원님의 정보를 보호하기 위해 비밀번호를 다시 확인합니다.

취소

다음

v2.0

localhost:8081 내용:
비밀번호가 일치하지 않습니다.

확인

비밀번호 확인

비밀번호

.....

회원님의 정보를 보호하기 위해 비밀번호를 다시 확인합니다.

취소

다음

4)기능 동작 및 시연 화면

localhost:8081 내용:
회원정보를 수정 했습니다.

확인

회원정보 열람

아이디	a10003	
비밀번호	비밀번호 입력	비밀번호 입력시 비밀번호 변경
비밀번호 확인	비밀번호 입력 확인	
회원가입일	2022-12-05 17:54:04	

개인정보 수정

이름	사람이름	
별명	공백없는 한글, 영문, 숫자 입력	
	사람별명	중복확인
이메일	me03454@naver.com	인증번호 받기
휴대폰	010-3234-1002	
주소	우편번호	우편번호찾기
	주소 검색	
	상세 주소 입력	
회원탈퇴	탈퇴하기	

취소

회원수정

a10003	1421167377d01272e96c2c0b4c9d6c3e3288997ab267ca...	사람이름	사람별명	me03454@naver.com	010-3234-1002
a10003	b2b94f7661a97d56312223170ba2900e4645447f98a800...	인간이름	인간별명	me03454@gmail.com	010-9876-5432

2

46729

회원정보 설정

아이디	a10003	
비밀번호	비밀번호 입력시 비밀번호 변경
비밀번호 확인	사용하실 수 있는 비밀번호 입니다.
회원가입일	2022-12-05 17:54:04	

개인정보 수정

이름	인간이름	
별명	공백없는 한글, 영문, 숫자 입력	
	인간별명	중복확인 사용 가능한 별명 입니다.
이메일	me03454@gmail.com	이메일이 인증 되었습니다. 인증번호 받기
	706051	확인
휴대폰	010-9876-5432	
주소	46729	우편번호찾기
	부산 강서구 가달1로 7	
	무슨무슨동 무슨무슨호	
회원탈퇴	탈퇴하기	

취소

회원수정

4)기능 동작 및 시연 화면

a10002	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
a10003	b2b94f7661a97d56312223170ba2900e4645447f98a800...	인간이름	인간별명	me03454@gmail.com	010-9876-5432
a102	edee29f882543b956620b26d0ee0e7e950399b1c4222f5...	사람이름2	사람별명2	human2@naver.com	010-1234-2222

localhost:8081 내용:

정말 삭제 하시겠습니까?

확인

취소

localhost:8081 내용:

회원 탈퇴가 완료 되었습니다.

확인

a10002	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
a10003	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
a102	edee29f882543b956620b26d0ee0e7e950399b1c4222f5...	사람이름2	사람별명2	human2@naver.com	010-1234-2222