Portfolio Web Site



심현섭

https://github.com/SimHyunSub/PortfolioWebSite



- 개발 환경 - 구조

- 목적 - 개발기간

- 화면 구현 - 시행착오

2 / 13 Portfolio Web Site Project 2



Language









etc

















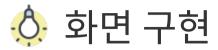






- 1. 개인의 포트폴리오에 올릴 아이디어들을 게시하고 구경할 수 있게 만든다.
- 2. 댓글을 작성할 수 있게 만들어 소통할 수 있는 Web Site를 만든다.
- 3. 웹사이트 운영자에게 회원가입을 하지 않아도 건의사항을 전달 할 수 있게 한다.

4 / 13 Portfolio Web Site Project 4





HashMap<String, Object> resultMap = new HashMap<String, Object>();

String boardNo = req.getParameter("boardNo");

resultMap.put("filesAndboard", filesAndboard);

return HttpdUtil.makeJsonView(resultMap);

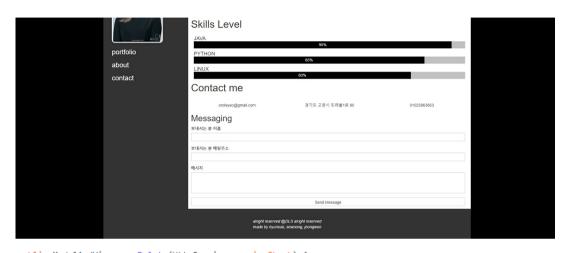
resultMap.put("boardNo", boardNo);

```
$(document).ready(function(){
                                                                                                      $.ajax({
                                                                                                              type:"post",
                                                                                                              url : "/shs/boardList"
                                                                                                       }).done(function(data){
                                                                                                              var d = JSON.parse(data)
                                                                                                              var filesAndboard = d.filesAndboard;
                                                                                                              console.log(filesAndboard);
                                                                                                              HTML(filesAndboard);
                                                                                                                     $(".arc-con").on("click", function(){
                                                                                                                     var index = $(".arc-con").index(this);
                                                                                                                     var boardNo = filesAndboard[index].boardNo;
                                                                                                                     console.log(boardNo);
                                                                                                                     location.href = "/shs/resources/ex/detail.html?boardNo=" + boardNo;
                                                                                                              });
                                                                                                      });
                                                                                               });
                                                                                                        <select id="filesAndboard" resultType="HashMap">
                                                                                                                 select b.boardNo, boardTitle, boardContents, fileURL
                                                                                                                          from board as b
List<HashMap<String, Object>> filesAndboard = session.selectList("board.filesAndboard");
                                                                                                                         left outer join files as f
                                                                                                                          on (b.boardNo = f.boardNo) where b.delYn = 'N'
                                                                                                        </select>
```

5 / 13

Main화면 익명 메세지 보내기

익명 메세지를 사용자들로부터 받을 수 있다.



```
public ModelAndView userDelete(HttpSession sessionCheck) {

    HashMap<String, Object> param = new HashMap<String, Object>();
    HashMap<String, Object> userMap = (HashMap<String, Object>) sessionCheck.getAttribute("user");
    HashMap<String, Object> resultMap = new HashMap<String, Object>();

    int userNo = (Integer) userMap.get("userNo");
    param.put("userNo", userNo);

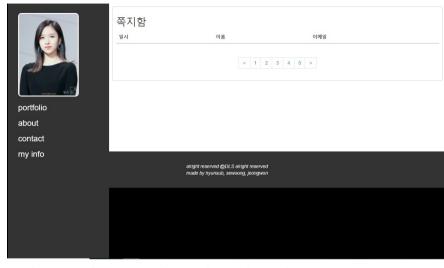
    int status = session.update("user.userDelete",param);

    resultMap.put("status", status);

    return HttpdUtil.makeJsonView(resultMap);
}
```

Portfolio Web Site Project Copy of 6

메세지함



```
public ModelAndView userDelete(HttpSession sessionCheck) {

    HashMap<String, Object> param = new HashMap<String, Object>();
    HashMap<String, Object> userMap = (HashMap<String, Object>) sessionCheck.getAttribute("user");
    HashMap<String, Object> resultMap = new HashMap<String, Object>();

    int userNo = (Integer) userMap.get("userNo");
    param.put("userNo", userNo);

    int status = session.update("user.userDelete",param);

    resultMap.put("status", status);

    return HttpdUtil.makeJsonView(resultMap);
}
```

```
$(document).ready(function(){
      $.ajax({
              url:"/shs/msgList"
      }).done(function(data){
              var d = JSON.parse(data);
              console.log(d);
              var list = d.msgList;
              $("#msgList").empty();
              var html =""
              for(var i=0; i<list.length; i++){</pre>
                     html += '';
                     html += ''+ list[i].regDate +'';
                     html += '' + list[i].msgName + '';
                     html += '' + list[i].msgEmail + '';
                     html += '';
              }
             $("#msgList").append(html);
```

7 / 13

글쓰기 page



에디터는 summernote를 사용하였습니다

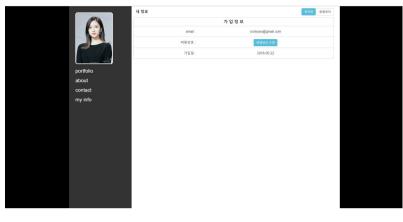
```
$("#write-btn").on("click", function(){
       // 데이터 가져오기
       var userNo = getUserNo();
       var category = $("#viewCategory").val();
       var title = $("#viewTitle").val();
       var content = CKEDITOR.instances['edit'].getData();
       // 데이터 초기화 하기
       $("#viewTitle").val("");
       CKEDITOR.instances['edit'].setData("");
       // 데이터 담기
       $("#category").val(category);
       $("#title").val(title);
       $("#content").val(content);
       $("#userNo").val(userNo);
       // 서버로 데이터 전송하기
       $.ajax({
               url : "/shs/boardInsert",
               data : new FormData($("#form")[0]),
               type : "POST",
               contentType:false,
               cache:false,
               processData:false
       }).done(function(data){
               console.log(data);
               if(data.status == 1){
                       alert("insert성공");
                       location.href="index.html"
               }else {
                       alert("실패");
       });
```

});

8 / 13

Portfolio Web Site Project 8





```
$.ajax({
        url: "/shs/userSession"
}).done(function(data){
       d = JSON.parse(data);
        console.log(d);
        $("#id").empty();
        $("#regDate").empty();
        var user = d.userSelectOne;
        var id = user.id;
        var passwd = user.password;
        var regDate = user.regDate;
        var html = id;
        var html2 = regDate;
        $("#id").append(html);
        $("#regDate").append(html2);
});
```

Java에서 session에

저장되어있는 user 정보 가져오는

부분의 소스코드 입니다.

비밀번호 수정하는 부분의 소스코드 입니다.

```
$("#submit_change").on("click", function(e){
       e.preventDefault();
       var password = $("#change_pw").val();
       $.ajax({
               url:"/shs/passwordUpdate",
               data : {"password" : password}
       }).done(function(data){
               var d = JSON.parse(data);
               console.log(d);
               if(d.status == 1){
                      alert("비밀번호 수정이 완료되었습니다. 다시 로그인 해 주세요.");
                      location.href = "/shs/logout";
               }else{
                      alert("뭔가 실패");
               }
       });
       layerOut();
});
```

```
public ModelAndView passwordUpdate(HttpServletRequest req,HttpSession sessonCheck) {

    HashMap<String, Object> userMap = (HashMap<String, Object>);
    HashMap<String, Object> param = new HashMap<String, Object>();

    HashMap<String, Object> resultMap = new HashMap<String, Object>();

    int userNo = (Integer) userMap.get("userNo");
    String password = req.getParameter("password");

    param.put("userNo", userNo);
    param.put("password", password);

    int status = session.update("user.passwordUpdate", param);
    resultMap.put("status", status);

    System.out.println(resultMap);
    return HttpdUtil.makeJsonView(resultMap);
}
```

Portfolio Web Site Project 10 9 / 13

상세보기 page

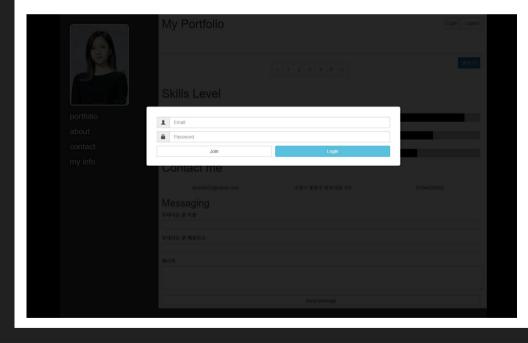


GET 방식으로 보낸 parameter받는 코드 입니다.

Portfolio Web Site Project 11 10 / 13

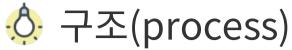
로그인/회원가입 page

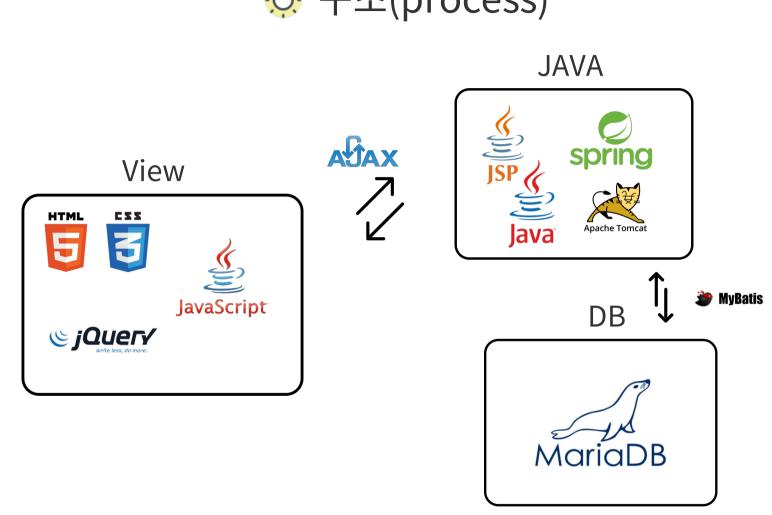




```
public ModelAndView userInsret(HttpServletRequest req) {
         HashMap<String, Object> resultMap = new HashMap<String, Object>();
         HashMap<String, Object> map = new HashMap<String, Object>();
         String id = req.getParameter("id");
         String userName = req.getParameter("userName");
         String password = req.getParameter("password");
         map.put("id", id);
         map.put("userName", userName);
         map.put("password", password);
         int status = session.insert("user.userInsert", map);
         resultMap.put("status", status);
         return HttpdUtil.makeJsonView(resultMap);
 ODequestManning("(userlegin")
public ModelAndView userLogin(HttpServletRequest req, HttpSession sessionC) {
       HashMap<String, Object> param = HttpdUtil.getParamMap(req);
       System.out.println(param);
       HashMap<String, Object> resultMap = session.selectOne("user.userLogin",param);
       if(resultMap == null) {
               resultMap = new HashMap<String, Object>();
               resultMap.put("status", 0);
               System.out.println("실패");
       }else {
               resultMap.put("status", 1);
               System.out.println("성공");
       sessionC.setAttribute("user", resultMap);
       return HttpdUtil.makeJsonView(resultMap);
```

Portfolio Web Site Project 12







⚠ 개발 기간

DB 설계 및 개발:96시간



♦ 시행착오

- 반응형 웹을 만들기 위하여 desktop css mobile css / tablet css / laptop css 들을 만들어서 화면이 변화할 때에 대한 적절한 코딩 이 쉽지 않았습니다.
- HTML구조를 잘 잡아놔야 반응형 웹을 만들 때 수 월하다는것을 깨닳았습니다.



♪ 감사합니다.