IT 종합설계(1)

2012253077 지영오
2012253066 김기현

목차

주제 및 개요

웹사이트를 만들기 위한 기능을 전부 다루어보는 것을 목표로 프로젝트를 시작했습니다.
목표한 기능들은 반응형웹/웹크롤링/회원가입&로그인/게시판&댓글/장바구니/주식차트/날씨api/웹채팅/bot시스템입니다.
웹사이트를 직접 제작해봄으로써 다양한 웹언어(html/css/javascript/jquerry)등을 다뤄보고 jsp/node.js/php/파이썬등의 서버 단 언어중에서 jsp를 채택하였으며 웹크롤링과 그 데이터를 DB에 넣고 활용하기위해 JAVA를 활용했습니다. JSON파일과 XML파일을 다루고 만들어보았으며, 이를 활용하는 방법으로는 각종 라이브러리를 활요하였습니다.

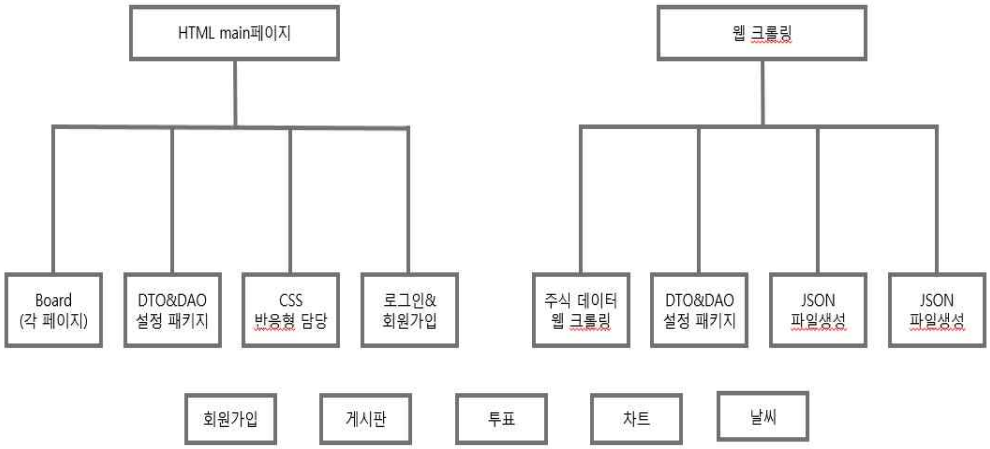
개발환경 및 과정

OS : window10 64bit
Program : Eclips Jee / Database : Oracle 11g / Toad for Oracle / Server : Tomcat 8.5
사용 언어 및 기능 : java, html5, jsp, javascript, Bootstrap, lombok, jstl, jquery, 기본 css, servlet
참고사이트 : Google / 네이버주식 / 폴로닉스 / 기상청

개발과정

내용	2018. 04				2018. 05				2018. 06			
	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주
웹 개념수립												
언어조사												
프론트엔드 언어공부												
DB선택&연동												
DB데이터 활용공부 (DTO,DAO)												
반응형 웹												
회원가입 기능구현												
웹 크롤링 기능구현												
날씨표시 XML활용												
게시판 기능구현												
JSON 파일생성 &라이브러리												
투표 기능구현												
코드 합치기 & 에러수정												

프로그램 구조



프로그램 코드해설

```
<nav class="navbar navbar-default" role="navigation">
    <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".nav-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>

    </div>

    <div class="collapse navbar-collapse nav-toggle">
        <ul class="nav navbar-nav">
          <li class="dropdown">
          <a href='#' class="dropdown-toggle" data-toggle="dropdown">소통의방<b class="caret"></b></a>
          <ul class="dropdown-menu">
            <li> <a href="#">글지</a></li>
            <li> <a href="../board/boardList.jsp?pg=1&checkFile=0">자유게시판</a></li>
            <li> <a href="#">게임게시판</a></li>
            <li> <a href="#">코인게시판</a></li>
            <li> <a href="#">소통게시판</a></li>
          </ul>
          </li>
          <li ><a href='../board/boardChard.jsp'>비트코인 차트</a></li>
          <li ><a href='../polling/pollList.jsp'>투표 프로그램</a></li>
          <li ><a href='../board/weather2.jsp'>오늘의 기온</a></li>
        </ul>
    <form class="navbar-form navbar-right" role="search">
    <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
    </div>
    <button type="submit" class="btn btn-default">검색</button>
    </form>
</div>
</nav>
```

bootstrap navbar class를 이용한 메인페이지. html 출력 자체를 깔끔하게 해주고 dropdown속성은 메뉴를 클릭 시 서브메뉴가 밑으로 표시되도록 할 수 있으며, navbar-toggle클래스로는 화면을 일정크기 이하로 줄일 시에 메뉴들이 사라지고 토글버튼이 나타나게 하여 토글버튼을 클릭 시에 메뉴들이 보이게 하는 동작을 한다.

회원가입 시에 어떤 조건이 맞지 않다면 경고문이 뜨는 코드이다.

```
<script type="text/javascript">

function checkWrite(){//javascript는 무조건 앞에 function
    if(document.writeForm.name.value=="") alert("이름을 입력해주세요....");
    else if(document.writeForm.id.value=="") alert("아이디를 입력해주세요...");
    else if(document.writeForm.pwd.value=="") alert("비밀번호를 입력해주세요...");
    else if(document.writeForm.pwd.value!=document.writeForm.repwd.value) alert("비밀번호가 같지않습니다");
    else document.writeForm.submit();
}
</script>
```

javascrip의 alert를 이용하여 경고문을 표시하였다.

```
if(name==null) {
    response.sendRedirect("loginFail.jsp");
}else {
session.setAttribute("memName", name);
session.setAttribute("memId", id);
response.sendRedirect("loginOk.jsp");
```

```
session.removeAttribute("memName");
session.removeAttribute("memId");

session.invalidate();//모든 세션 없애기
```

session을 이용한 로그인 세션을 등록하는 코드이다. 로그아웃을 하면 세션이 제거된다. 세션을 등록하여 다른 화면으로 넘어가더라도 로그인이 지속적으로 이루어질 수 있게 한다.

```
<c:if test="${sessionScope.memId!=null }">
    <a href='../member/logout.jsp'>로그아웃</a> 
    <a href='../board/boardWriteForm.jsp'>글쓰기</a>  
</c:if>
<c:if test="${memId==null }">
    <a href='../member/loginForm.jsp'>로그인</a> 
    <a href='/webProject/member/writeForm.jsp'>회원가입</a>  
```

위의 세션을 이용한 메인화면이다. 로그인이 되어있다면 위의 html, 되어있지않다면 밑의 html이 실행된다.

```
function isLogin(seq){

    if('${memId}'==''){
        alert("로그인을 해주세요!!");
        location.href="../member/loginForm.jsp";
    }
    else
        location.href='boardView.jsp?seq='+seq+'&pg=${pg}&count=0';
```

세션을 이용하여 로그인이 안 되어있을 시에 게시판 글을 보지 못하도록 하는 코드이다.

```
String filename="첨부파일없음";
String original="";
String savedPath = "D:/ITProject/webProject/WebContent/board/fileStorage";
String stringType = "utf-8";
String title = "";
String content ="";
boolean check = false;
int maxSize = 10*1024*1024; //10MB

try{
        MultipartRequest multi = null;

        multi = new MultipartRequest(request, savedPath, maxSize, stringType,
                new DefaultFileRenamePolicy());
        Enumeration params = multi.getParameterNames();

        while(params.hasMoreElements()){
            String dataname = (String)params.nextElement();
            String value = multi.getParameter(dataname);
            out.println(dataname + "=" +value+"<br>");
        }

        Enumeration files = multi.getFileNames();
        while(files.hasMoreElements()){
                String paramname = (String)files.nextElement();
                filename = multi.getFilesystemName(paramname);

                original = multi.getOriginalFileName(paramname);
                String type = multi.getContentType(paramname);

                File f = multi.getFile(paramname);
                out.println("파라미터 이름 : "+paramname+"<br>");
                out.println("실제 파일 이름 : "+original+"<br>");
                out.println("저장된 파일 이름 : "+filename+"<br>");
                out.println("파일 타입 : "+type+"<br>");
                if(f!=null){
                    out.println("크기:"+f.length()+"바이트");
                    out.println("<br>");
                    check=true;
                }
                title = multi.getParameter("title");
                content = multi.getParameter("content");


        }
}catch(IOException ioe){
    System.out.println(ioe);
}catch(Exception ex){
    System.out.println(ex);
}

if(check==false){
    filename="파일없음";
    original="파일없음";
}
```

게시판에서 파일 업로드를 할 수 있게 만든 코드이다. cos.jar파일과 servlet을 이용하였다. MultipartRequest를 이용해 업로드파일의 저장위치와 최대크기, 타입 등을 설정하고 그 조건에 맞춰 업로드 폴더에 저장한다. 그 이후에 업로드할 때 넘어온 파라미터를 받아 데이터이름과 값을 설정 후에 파라미터 이름, 실제 파일 이름, 저장된 파일 이름, 파일 타입 등을 받아 출력해준다. 업로드한 파일이 없을 시에는 check를 이용해 파일없음을 출력한다.

```
String fileName = request.getParameter("file_name");
String savedPath = "D:/ITProject/webProject/WebContent/board/fileStorage";
String filePath=savedPath+"\\"+fileName;
File oFile = new File(filePath);

byte[] b = new byte[10*1024*1024];

FileInputStream in = new FileInputStream(oFile);

String sMimeType = getServletContext().getMimeType(filePath);

if(sMimeType ==null){
    sMimeType = "application.octec-stream";
}

response.setContentType(sMimeType);

String A = new String(fileName.getBytes("euc-kr"),"8859_1");
String B = "utf-8";
String sEncoding = URLEncoder.encode(A,B);

String AA = "Content-Disposition";
String BB = "attachment; filename="+sEncoding;
response.setHeader(AA, BB);

ServletOutputStream out2 = response.getOutputStream();

int numRead = 0;

while((numRead=in.read(b,0,b.length))!=-1){
    out2.write(b,0,numRead);
}
out2.flush();
out2.close();
in.close();
```

파일 다운로드를 위한 코드이다. 업로드 파일이 저장된 위치와 파일 이름을 받아온 후에 붙여준 후 그 파일에 직접적으로 접근할 수 있는 파일 객체를 생성한다(oFile). 그리고 그 파일의 MimeType을 얻어 sMimeType에 저장한다. ContentType을 mimeType에 맞게 설정하고 파일에 맞는 헤더를 세팅해준다.fileInputStream in(파일을 1바이트단위로 읽어들이기위한 통로)과 ServletOutputStream out2(사용자에게 파일을 출력, 다운로드 하기위한 통로 준비). in.read메소드의 리턴값은 읽어들이기 성공한 바이트 수를 리턴하고 out2.write를 통해 출력 버퍼에 내보낸다. 그 후 flush로 출력버퍼의 데이터를 비워주고 output과 input을 닫아준다.

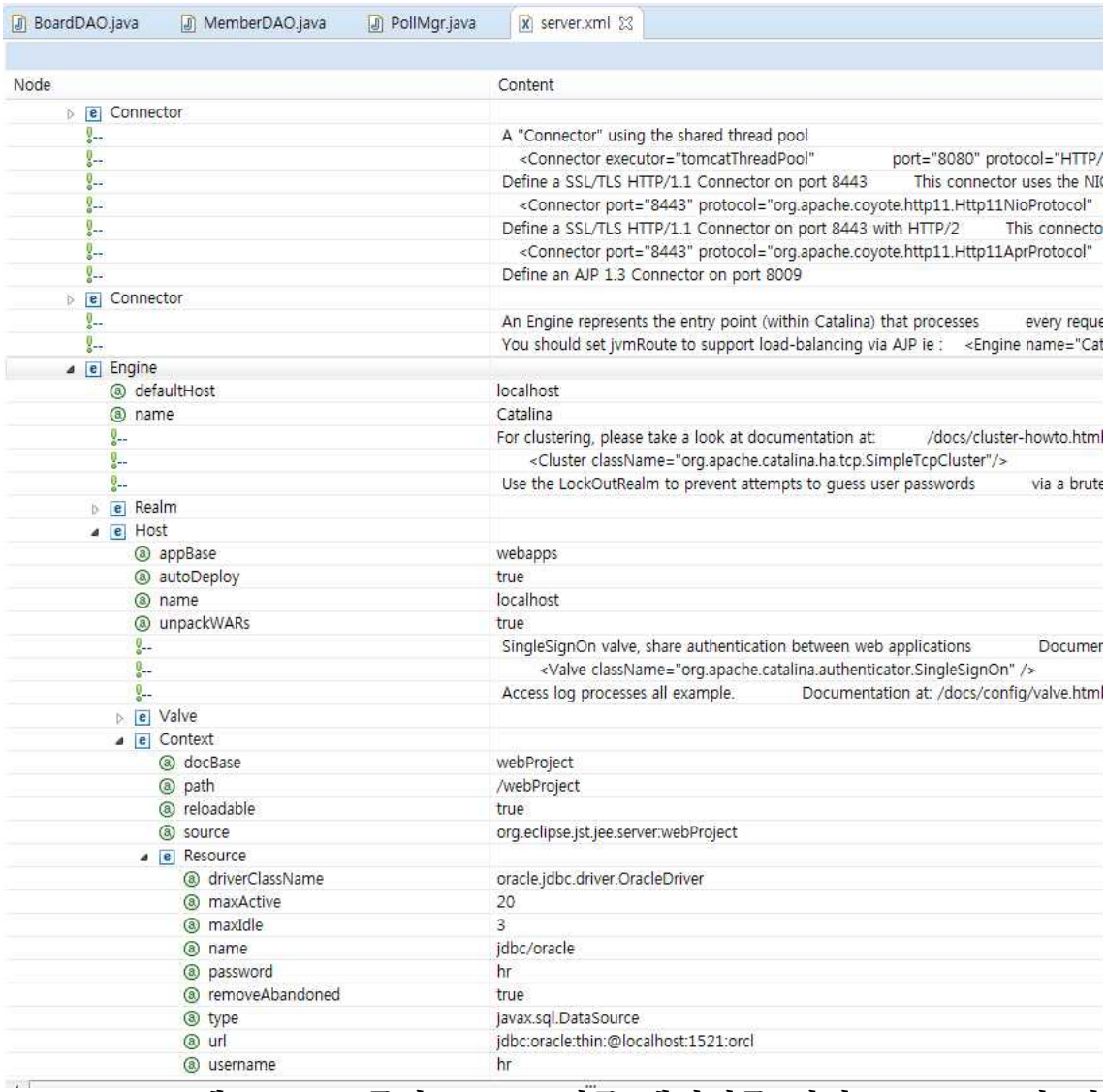
```
int pg = Integer.parseInt(request.getParameter("pg"));
int endNum = pg * 5;//끝 페이지
int startNum = endNum - 4;//시작 페이지

int totalAt = dao.countTotal();//total Article
int totalPg = 0;//total Page
if (totalAt % 5 == 0)
    totalPg = totalAt / 5;
else
    totalPg = (totalAt / 5) + 1;

//3개씩 페이지블록
int startPage = (pg-1)/3*3+1;
int endPage=startPage+2;
if(endPage>totalPg){

    endPage=totalPg;
}
   <c:if test="${startPage!=1 }">
       [<a href="boardList.jsp?pg=${startPage-1}" >이전</a>]
   </c:if>
   <c:forEach var="i" begin="${startPage}" end="${endPage}" step="1">

       <c:if test="${i==pg}">
           [<a href="boardList.jsp?pg=${i}" class="currentPaging">${i }</
       </c:if>
       <c:if test="${i!=pg}">
           [<a href="boardList.jsp?pg=${i}" id="paging">${i}</a>]
       </c:if>
   </c:forEach>

   <c:if test="${endPage<totalPg }">
       [<a href="boardList.jsp?pg=${endPage+1}" >다음</a>]
   </c:if>
```

게시판에서 글을 많이 작성할 시에 화면에 한번에 보여주면 너무 복잡하기 때문에 나눠서 보여주기 위하여 만든 코드이다. 게시글이 다섯 개가 채워지면 1 2 3 4 순서대로 페이지 번호가 등록되고 페이지 번호도 3 페이지 단위로 나뉘어 <이전>과 <다음> 페이지 이런 식으로 표시되게 하였다. pg라는 변수에 5개의 게시글 마다의 번호를 설정하여 넘겨주게 하였다. View로 페이지 번호를 출력하기 전에 jsp코드로 페이지의 초기 설정을 해주었다. startNum은 5개의 게시글 단위로 나뉘어졌을 때의 첫 번째 게시글 번호이고 endNum은 마지막 게시글의 번호가 된다(pg변수에 따라 변하게 된다). totalAt은 총 게시물의 개수를 받아오고 totalPg는 총 게시물을 5로 나누어 5개씩의 게시물을 보여주는 페이지를 숫자 순으로 페이지 번호를 매기기 위한 변수이다. startPage와 endPage는 페이지 번호를 매길 때 3개씩 나누기 위함이며 표시될 때 pg에 따라 변하게 수식을 정하였다.

이전을 누르면 startPage보다 페이지 번호가 하나 적은 페이지로 이동하며 다음은 endPage의 다음 페이지(페이지 번호가 하나 큰)로 넘어간다.

```
for(int i=0;i< vlist.size();i++){
    PollItemBean piBean = vlist.get(i);
    String[] item = piBean.getItem();
    int rgb = r.nextInt(255*255*255);
    String rgbt = Integer.toHexString(rgb);
    String hRGB = "#" +rgbt;
    int count = piBean.getCount();
    int ratio = (new Double (Math.ceil((double) count/sum*100))).intValue();
%>
    <tr>
    <td width="20" align="center"><%=i+1%></td>
    <td width="120"><%=item[0] %></td>
            <td>
                <table width="<%=ratio %>" height="15">
                    <tr>
                        <td bgcolor="<%=hRGB %>"></td>
                    </tr>
                </table>
            </td>
            <td width="40"><%=count%></td>
    </tr>
<%
}
```

투표를 하였을 때 몇표가 나왔는지에 따라 그 표의 수를 비쥬얼적으로 보여주기 위하여 RGB를 이용해 표현한 코드이다. 먼저 투표지의 투표했던 아이템을 불러오고 rgb는 랜덤함수로 랜덤으로 정수를 받은 다음에 toHexString을 이용해 16진수로 바꿔준 후 #을 붙여 완성한다(색깔을 정해주는 변수). count에는 그 아이템에 투표한 투표자들의 수가 들어가며 ratio를 통해 전체 투표자들 중에 몇 할인지를 알아내어 bgcolor로 rgb표현을 할 때 width를 ratio로 줌으로써 상대적으로 표를 나타낼 수가 있다.



<server.xml에 context문장으로 DB연동 데이터를 넣어 Database와 연동할 때 데이터를 따로 입력하지 않도록 하였다)

```
public String encryptPwd(String pwd) throws Exception {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] mdResult = md.digest(pwd.getBytes("UTF-8"));
    sun.misc.BASE64Encoder encoder = new sun.misc.BASE64Encoder();
    String str = encoder.encode(mdResult);
    return str;
}
```

패스워드 암호화 코드이다. 방식은 SHA-256형식을 썼다. MesseageDigest함수를 이용하여 형식을 지정하고 pwd를 그 형식에 맞게 바꾼 뒤, encoder를 이용하여 부호화 시켜 리턴 시키는 동작을 한다.

```java
import lombok.Data;

@Data
public class MemberDTO {
    private String name;
    private String id;
    private String pwd;
    private String gender;
    private String email1;
    private String email2;
    private String tel1;
    private String tel2;
    private String tel3;
    private String zipcode;
    private String addr1;
    private String addr2;

}
```

lombok이라는 jar파일을 이용하여 getter와 setter를 따로 만들지 않아도 자동생성하게 하는 기능을 구현할 수 있다.

웹 크롤링 파트

```
Calendar now = Calendar.getInstance();
int year = now.get(Calendar.YEAR);
int month = now.get(Calendar.MONTH)+1;
int day = now.get(Calendar.DATE);
String samsung_date = year+"-"+month+"-"+day;
System.out.println(samsung_date);

String URL = "http://finance.naver.com/item/main.nhn?code=005930";
Document doc = Jsoup.connect(URL).get();

Element today_up = doc.select("em.no_up").get(0);
Elements upprice = today_up.select("span.blind");
System.out.println(upprice.text());
String samsung_high = upprice.text();

Element today_down = doc.select("em.no_down").get(0);
Elements downprice = today_down.select("span.blind");
System.out.println(downprice.text());
String samsung_low = downprice.text();

Element today_open = doc.select("em.no_open").get(0);
Elements openprice = today_down.select("span.blind");
System.out.println(downprice.text());
String samsung_open = openprice.text();

Element today_close = doc.select("em.no_close").get(0);
Elements closeprice = today_down.select("span.blind");
System.out.println(downprice.text());
String samsung_close = closeprice.text();
fInencedata dto = new fInencedata();
dto.setSamsung_date(samsung_date);
dto.setSamsung_high(samsung_high);
dto.setSamsung_low(samsung_low);
dto.setSamsung_open(samsung_open);
dto.setSamsung_close(samsung_close);
dto.setSamsung_volume(samsung_volume);

//DB
fInenceDAO dao = new fInenceDAO();
int su = dao.write(dto);//호출

makejson mj =new makejson();
mj.make();
```

자바함수인 Calendar함수를 통해서 크롤링을 시도하는 현재 날짜를 받아옵니다.
네이버 주식페이지의 데이터를 크롤링하기 위해 Jsoup라이브러리를 사용하였으며
Document객체에 크롤링하고자 하는 링크를 connect 시켜줍니다.
크롬의 개발자도구창을 이용해서 크롤링하고자 하는 부분에 마우스를 올려놓고 클릭하면
해당 html코드가 나옵니다. 그 부분의 class명을 따서 클래스명. 혹은 div명#을 통해서
해당부분의 코드를 가져올 수 있습니다. 주의할점으로는 네이버주식의 경우 개발자도구를 통해 보여지는 코드말고도 blind된 코드들이 나타나는데 이부분은 상위 코드를 전부 출력시켜보고 다시 내부 class명을 찾아서 파싱해야하는 문제점이 있었습니다.

이후 크롤링한 데이터들을 dto객체로 옮겨놓은후 dto객체를 dao객체로 넘겨주면
dao객체의 write함수가 해당 데이터를 DB로 옮겨 저장하게 됩니다.

DTO 파트

```
public class fInencedata {

    public String getSamsung_date() {
        return samsung_date;
    }
    public void setSamsung_date(String samsung_date) {
        this.samsung_date = samsung_date;
    }
    public String getSamsung_high() {
        return samsung_high;
    }
    public void setSamsung_high(String samsung_high) {
        this.samsung_high = samsung_high;
    }
    public String getSamsung_low() {
        return samsung_low;
    }
    public void setSamsung_low(String samsung_low) {
        this.samsung_low = samsung_low;
    }
    public String getSamsung_open() {
        return samsung_open;
    }
    public void setSamsung_open(String samsung_open) {
        this.samsung_open = samsung_open;
    }
    public String getSamsung_close() {
        return samsung_close;
    }
    private String samsung_date;
    private String samsung_high;
    private String samsung_low;
    private String samsung_open;
    private String samsung_close;
    private String samsung_volume;
}
```

DTO기능을 담당하는 finencedata.java 파일입니다.
DTO란 Data Trancefer Object 약어로써
각 계층간 데이터 교환을 위한 객체를 DTO 라고 합니다. 뷰 페이지(jsp, html)로부터 입력된 데이터를 DTO에서 일시적으로 저장하여 DAO로 넘겨주거나, 데이터베이스로부터 읽어온 데이터를 DAO로부터 넘겨받아 DTO에서 일시적으로 저장하여 뷰 페이지로 이동시켜 주는 역할을 합니다. 즉, 데이터 임시 저장 및 교환을 위한 객체(자바빈/Java Bean)입니다.
단순하게 DB로 넘길 데이터를 선언해두고 해당 변수에대한 get,set함수를 선언해주는 코드입니다.

DAO 파트

```
public class fInenceDAO {

    private Connection conn;
    private PreparedStatement pstmt;
    private Statement stmt;
    private ResultSet rs;


    public fInenceDAO() {// 생성자
        try {// 드라이버 연결
            Class.forName("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    private static fInenceDAO instance;

    public static fInenceDAO getInstance(){
        if(instance==null){
            synchronized(fInenceDAO.class){
                instance = new fInenceDAO();

            }
        }
        return instance;
    }
```

```
public void getConnection() {

    try {
        conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl", "hr", "hr");

    } catch (SQLException e) {
        e.printStackTrace();

    }

}
```

```
public int write(fInencedata dto){
    int su=0;
    getConnection();
    String sql = "insert into samsung values(?,?,?,?,?,?)";
    try{
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, dto.getSamsung_date());
        pstmt.setString(2, dto.getSamsung_high());
        pstmt.setString(3, dto.getSamsung_low());
        pstmt.setString(3, dto.getSamsung_open());
        pstmt.setString(3, dto.getSamsung_close());
        pstmt.setString(3, dto.getSamsung_volume());

        su=pstmt.executeUpdate();
```

```
public Vector getList() {
    Vector vectordata=new Vector();
    ResultSet rss = null;
    PreparedStatement ps = null;
    String pricedate, high, low, open, close, volume;
    getConnection();
    try {
        String sql = "select * from samsung order by CHART_DATE";
        ps = conn.prepareStatement(sql);
        rss = ps.executeQuery();

        while(rss.next()) {
            pricedate = rss.getString("CHART_DATE");
            high = rss.getString("HIGH");
            low = rss.getString("LOW");
            open = rss.getString("OPEN");
            close = rss.getString("CLOSE");
            volume = rss.getString("VOLUME");

            Vector innerdata=new Vector();
            innerdata.add(pricedate);
            innerdata.add(high);
            innerdata.add(low);
            innerdata.add(open);
            innerdata.add(close);
            innerdata.add(volume);

            vectordata.add(innerdata);
        }

        rss.close();
        ps.close();
        conn.close();
return vectordata;
```

DB연동과 getList함수와 write함수를 다루는 finenceDAO.java 파일입니다.
write함수는 DTO를 통해 받은 데이터값을 DB의 samsung테이블에 넣어줍니다.
getList함수는 DB로부터 데이터를 가져와 Vector형으로 저장하여 리턴하는 함수입니다.

JSON 생성 파트

```
public void make() throws IOException, ParseException{
    Vector data = new Vector();
    fInenceDAO dao = new fInenceDAO();
    data = dao.getList();
    JSONObject obj = new JSONObject();
    JSONObject ob2 = new JSONObject();
    JSONObject ob3 = new JSONObject();
    JSONArray chart_list = new JSONArray();

    String DB_Date;
    String[] split_Date;
    String rst_Date;
    for(int j = 0; j<data.size(); j++)
    {
        //2000-01-01값을 unix_timestamp값으로 변경하는 코드
        DB_Date = (String) ((java.util.List) data.get(j)).get(0);
        long unix_timestamp;
        SimpleDateFormat formatter = new SimpleDateFormat("yyyy-mm-dd");
        Date userDate = formatter.parse(DB_Date);
        unix_timestamp = userDate.getTime();


        ob2 = new JSONObject();
        ob2.put("date", unix_timestamp);
        ob2.put("high", (String) ((java.util.List) data.get(j)).get(1));
        ob2.put("low", (String) ((java.util.List) data.get(j)).get(2));
        ob2.put("open", (String) ((java.util.List) data.get(j)).get(3));
        ob2.put("close", (String) ((java.util.List) data.get(j)).get(4));
        ob2.put("volume", (String) ((java.util.List) data.get(j)).get(5));
        ob2.put("quoteVolume", (String) ((java.util.List) data.get(j)).get(5));
        ob2.put("weightedAverage", (String) ((java.util.List) data.get(j)).get(5));
        chart_list.add(ob2);
    }

    FileWriter file = new FileWriter("C:\\Users\\pc189\\Downloads\\ITProject\\webProject\\WebContent\\board\\test.json");
    System.out.println("json파일 생성");
     file.write(chart_list.toJSONString());
     file.flush();
     file.close();
}
```

DAO로부터 벡터형 데이터를 받아오기위한 data 객체를 선언합니다.
finenceDAO 클래스 객체도 선언하여
dao.getList를 통해 DB내 samsung테이블의 모든 값을 data객체로 불러옵니다.
JSON객체형 변수를 선언하고 이 json객체는 한줄만 읽어올수있기 때문에 이것을 이어주기 위한 JSONArray 형 변수를 선언해둡니다.

for문의 시작부분은 DB에 날짜값이 2000-01-01과 같은 DATE타입인 것을 JSON형인 unix_timestamp형식으로 변경하기 위한 코드입니다.
DB의 날짜값을 parse함수를 통해 '-'을 제거하고 getTime함수를 통해 1970년도를 기준으로 한 초단위시간으로 변경합니다.

이후 선언해둔 Json객체형 변수에 put을 통해서 json형식으로 데이터들을 모두 집어넣습니다.for문이 다시 돌기전에 JsonArray형 변수에 ob2객체를 넣어둠으로써 for문이 종료된후 chart_list객체를 통해 DB의 모든내용이 담긴 JSON파일을 만들어낼 수 있습니다.

여기서 중요한점은 FOR문의 초입부에 다시한번 ob2를 초기화해줘야만 모든값이 들어간다는 점입니다. 만약 초기화하지않을시 json파일은 2000~2018년의 데이터를 저장해두는 것이 아닌 똑같은 날짜의 데이터를 4500개 보유하게 됩니다. 이는 초기화를 하지 않을시 ob2객체가 메모리 할당에서 오류가 생기는 것이 아닐까 생각됩니다.

JSON 파일활용 (차트 게시판)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
    <title>Insert title here</title>
    <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
    <script src="https://code.highcharts.com/stock/highstock.js"></script>
    <script src="https://code.highcharts.com/stock/modules/exporting.js"></script>
</head>
<body>
    <div id="container" style="height: 400px; min-width: 310px"></div>
        <script>
            function draw3(){
                var chartdata = [];
                $.getJSON('test.json', function (data) {
                    $.each(data, function(i, item){
                        chartdata.push([item.date*1000, item.open, item.high, item.low, item.close]);
                    });
                }).done(function(){
                    Highcharts.stockChart('container',{
                        title: {
                            text: 'samsung'
                        },
                        rangeSelector: {
                            buttons: [
                                {type: 'hour',count: 1,text: '1h'},
                                {type: 'day',count: 1,text: '1d'},
                                {type: 'all',count: 1,text: 'All'}
                            ],
                            selected: 2,
                            inputEnabled: true
                        },
```

```
                        plotOptions: {
                            candlestick: {
                                downColor: 'blue',
                                upColor: 'red'
                            }
                        },
                        series: [{
                            name: 'samsung',
                            type: 'candlestick',
                            data: chartdata,
                            tooltip: {
                                valueDecimals: 8
                            }
                        }]
                    });
                });
            }
            draw3();
        </script>
    </body>
</html>
```

생성된 JSON파일을 불러들여 차트를 생성하는 코드입니다.

직접 차트를 제작할 경우 JSON파일로 만들지않고 DB내 데이터를 활용하여 제 나름대로 꾸밀 수 있다는 장점도 있으나, 이미 만들어진 라이브러리의 기능이 워낙 강력하여 highcharts의 라이브러리를 활용했습니다.

단점으로는 이미 만들어진 라이브러리의 사용법을 그대로 따라서 사용해야 한다는 점입니다. 제가 원하는 차트를 띄울 수는 없고 제공되는 차트만 봐야 한다는 점인데 이 부분은 라이브러리를 직접 뜯어보고 이를 튜닝하는 것을 통해서 해결해볼 수 있다고 판단이 되었지만, 이번 프로젝트 기간내에는 무리라고 판단하여 다음에 시도해볼 예정입니다.

XML 파일 활용 파트

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript" src="../script/jquery.min.js"></script>
<link rel="stylesheet" href="../css/board.css">
<link rel="stylesheet" href="../css/bootstrap.min.css">
<script type="text/javascript" src="../script/bootstrap.min.js"></script>
<script type="text/javascript" src="../script/angular.min.js"></script>
<script type="text/javascript">
    window.onload = function() {
        document.getElementById("btnOk").onclick = function() {
            startXHR(); //function startXHR로 이동
        }
    }

    var xhr;
    function createXHR() {
        if (window.ActiveXObject) {
            xhr = new ActiveXObject("Msxml2.XMLHTTP");
        } else {
            xhr = new XMLHttpRequest(); // xhr을 얻음
        }
    }

    function startXHR() {
        createXHR();

        xhr.open("get", "weather2.xml", true); //weather.xml 호출
        xhr.onreadystatechange = function() {

            if (xhr.readyState == 4) {
                if (xhr.status == 200) {
                    process();
                }
            }
        }
        xhr.send(null);
    }

    function process() { // xml 업데이트 시점 불러오기
        var data = xhr.responseXML;

        var weatherNode = data.getElementsByTagName("weather")[0];
        var strDate = weatherNode.getAttribute("year") + "년 "
                + weatherNode.getAttribute("month") + "월 "
                + weatherNode.getAttribute("day") + "일 "
                + weatherNode.getAttribute("hour") + "시";

        document.getElementById("disp").innerHTML = strDate; // disp 레이어에 집어넣음

        var localNode = data.getElementsByTagName("local");
        //날씨 출력방법1 - HTML DOM을 쓰는 방식
        for (var i = 0; i < localNode.length; i++) {
            var row = document.createElement("tr"); // tr 생성
            var col1 = document.createElement("td"); // td 생성
            var col2 = document.createElement("td"); // td 생성
            var loc = localNode[i].childNodes[0].nodeValue;
            var ta = localNode[i].getAttribute("ta");
            col1.appendChild(document.createTextNode(loc)); //appendchild로 넣음
            col2.appendChild(document.createTextNode(ta)); //appendchild로 넣음
            row.appendChild(col1);
            row.appendChild(col2);
            document.getElementById("myTbody").appendChild(row); // 테이블을 만들어서 row에 appendchild로 넣음
        }

        //날씨 출력방법1 - innerHTML을 쓴 방식
        var ss = "<table>";
        for (var i = 0; i < localNode.length; i++) {
            ss += "<tr>";
            ss += "<td>" + localNode[i].childNodes[0].nodeValue + "</td>";
            ss += "<td>" + localNode[i].getAttribute("ta") + "</td>";
            ss += "</tr>";
        }
        ss += "</table>";
        document.getElementById("disp2").innerHTML = ss;
    }
</script>
</head>
<body>
```

자바스크립트로 XML파일을 활용하는 페이지를 구현해봤습니다.
버튼클릭시 start 함수로 이동하며 xml파일을 불러옵니다. start함수내에서 다시 process함수를 호출하고 프로세스함수는 loc변수에 지역을 ta변수에 온도값을 넣어주며 이를 document에 쓰고 출력하는 방식입니다. innerHTML방식은 html코드를 직접 쓰는것과 같은 코드입니다. 직관적이여서 더 이해하기 쉬웠습니다.

<결과 화면>

게시판

Y&K Explore Experience Site

로그인 회원가입

소통의방 주식 차트 투표 프로그램 Search 검색

글번호	제목	파일첨부	작성자	조회수	작성일
32	123	파일없음	123	1	2018.06.05
31	모두다	다운로드.jfif	minyoung	0	2018.06.05
30	그래	파일없음	minyoung	0	2018.06.05
29	코딩연습하는데	index.jsp	rlgus03	0	2018.06.05
28	나도 고양이 키우지롱	432400_341075_268_20171226192715130.jpg	rlgus03	0	2018.06.05

[1] [2] [3] [다음]

주식 차트

Y&K Explore Experience Site

로그인 회원

소통의방 주식 차트 투표 프로그램 Search 검색

samsung

Zoom 1h 1d All From Jan 28, 2017 To Jun 11, 201

투표 프로그램

0

Q : 제일 예쁜 연예인

- 수지
- 전지현
- 김태희
- 나연
- 설리
- 이선화
- 현아
- 고소영

투표 결과

번호	제목	시작일~종료일
4	제일 예쁜 연예인	2018-01-01 00:00:00.0 ~ 2019-01-01 00:00:00.0
3	제일 맛잇는 야식2	2018-05-01 00:00:00.0 ~ 2019-01-07 00:00:00.0
2	제일 맛있는 야식	2018-01-01 00:00:00.0 ~ 2019-01-01 00:00:00.0
1	제일 잘생긴 연예인	2018-08-08 00:00:00.0 ~ 2019-04-13 00:00:00.0

설문 작성하기

날씨

Y&K Explore Experience Site

소통의방 주식 차트 투표 프로그램 오늘의 날씨

온도 확인

2018년 06월 05일 16시

속초	25.7
북춘천	25.1
철원	26.4
동두천	26.4
파주	26.5
대관령	20.7
춘천	26.5
백령도	26.0
북강릉	26.0
강릉	27.4
동해	26.9
서울	27.0

디스커션

지영오 : 웹에 대해 전반적으로 개념을 잡고 이해하는 계기가 되는 프로젝트였습니다. 말 다양한 언어와 기능을 공부해야 해서 깊게 공부해야한다고 생각이 들었고 기본적인 예제만 만들어보는데도 불구하고 상당히 힘들었습니다.

저는 주로 java쪽에서 웹에서 쓸 데이터를 만들고 하는 부분과 자바스크립트를 통해 XML파일을 다루는 부분을 코딩했습니다. 이번 기회에 dto, dao에 대한 개념을 확실히 잡았고 이를 통해 웹크롤링/데이터파일 등 여러 파일을 DB를 통해 다룰 수 있게 되었습니다.

웹 크롤링또한 그냥 막연히 그런 것이 있다는것만 알고 있었는데 이번기회를 통해 java를 이용한 웹 크롤링 방법을 기초적으로 익혔으며, 한 방법을 배우고나니 다른방법도 있다는게 눈에 들어와서 다음번에는 파이썬을 통해 크롤링하여 그 데이터를 기반으로한 프로젝트를 진행해 볼 수도 있겠다는 생각이 들었습니다.

가장 어려웠던 파트는 JSON파일을 만드는 파트였습니다. 구글링을 통해서 제대로 자료를 찾기가 어려웠기에 그랬습니다. 보통 JAVA만을 활용해서 JSON객체만 만들고 이를 출력하는 선에서 그치는 예제들이였기에 제가 구현하고자했던 DB의 데이터를 끌어와 JSON객체가 아닌 JSON파일로 만드는 부분을 직접 구현하다보니 어려움이 있었습니다. 가장 큰 문제는 unix_timestamp에 대한 지식이 아예 없다보니 이 부분에서 시간을 많이 허비하였습니다.
결국 검색어를 계속 늘려보고, 대략 비슷한 예제가 나오면 해당 함수를 찾아서 함수사용법을 보고 직접 바꿔가면서 코드를 테스트했습니다.

이번 프로젝트에서 얻은 경험과 지식 덕분에 다음 프로젝트를 해나가는데 기반을 마련할 수 있었다고 생각합니다. 다음 프로젝트로는 node.js를 기반으로 서버를 꾸리고 기존의 구성방식인 모놀리식 구성이 아닌 마이크로 서비스식 웹을 구현해보고 남의 라이브러리를 튜닝해볼 예정이며, 쇼핑몰기능들을 구현하면서 이번에는 제대로 하지못했던 css기능을 잘 활용하여 예쁘게 꾸며보는 실력도 기르고자 합니다. 마지막으로 가능하다면 서버와 데이터를 처리하는 방법에 대해서도 공부하여 다음 프로젝트를 진행하고자 합니다.

김기현: 처음 자료를 검색하고 새로운 기능들을 검색해보았을 때는 웹 개발이라는게 HTML, 서버, DB 이런 단순한 구조이면서도 정말 많은 보조적인 언어들을 가지고 있다는 것에 놀랐다. 예전에는 HTML로 기본 베이스를 짜고 css나 javascript로 꾸민 뒤, 데이터베이스와 jsp 같은 서버용 언어를 서버와 연동시키면 끝이라는 이미지를 가지고 있었는데 말과는 다르게 프론트 엔드서든지 백 엔드에서든지 정말 많은 종류의 언어 또는 프로그램이 있었고 또한 그에 따른 장점과 단점, 그리고 많은 기능들이 있었다. 아직 기초적인 것 밖에 알지 못했던 때이기 때문에 간단하고 작게만 보였던 웹 개발에 대한 내 시야가 더 넓어졌다.

서버 연동은 TOMCAT 서버를 쓰기로 하였는데 처음 이클립스와 연동하여 실행하였는데 작동이 안 되어서 인터넷어 물어봤더니 이클립스가 자바 9버전 이상부터는 연동이 잘 안된다 말을 들었다. 따라서 버전을 8버전으로 낮췄더니 서버와 연동이 되었다. 또한 데이터 베이스는 ORACLE DB와 연동하였는데 이건 설치할 때 조금 어려운 부분이 있었다. 패스워드를 잘못 입력했는지 db에 계정 로그인이 안되어서 삭제 후 다시 설치하려니 설정해야하는 부분이 많았다. 결국 다시 설치하여 hr이라는 계정을 만들어서 연동하게 되었다.

개발에 들어가서는 HTML과 JSP를 이용한 기능 개발, 서버와 데이터 베이스 연동, 웹 디자인 부분을 맡았습니다. 먼저 로그인과 회원가입에 관련하여 프로그래밍을 하게 되었는데 처음은 다른 웹을 참고하면서 간단하게 만들어보았는데 보기에는 그럴싸하지만 보안에 너무 문제가 있을 것 같아서 패스워드 암호화, 자바 스크립트를 이용한 일정 란 공란에 관한 오류출력, 세션 설정이라는 기능을 넣었습니다. 쿠키라는 기능도 넣으려고 하였으나 세션과 겹

치는 부분이 많아서 빼는 것이 좋다고 생각하였습니다. 또한 세션에 관한 조건문을 쓸 때는 jstl이라는 태그 라이브러리를 썼습니다(C언어와 비슷하여 사용하기 가장 좋다고 판단하였습니다).

다음은 게시판을 만들어보았는데 일단 간단하게 형식만 만든 후에 게시글 5개씩 나누어서 페이지 나누는 기능을 만들었는데 이건 좀 고난이었다. pg에 따라 어떻게 나누어야하는지 어려워서 인터넷에서 참고도하고 책도 보면서 만들게 되었다. SQL문도 짜는게 고생이었는데 게시글 삭제, 수정, 작성 3기능을 다 구현하려다보니 조금 복잡하고 실수도 많이 일어났다. 특히 SEQUENCE를 만들어 게시판 번호를 매길려다보니까 중간에 게시글 작성 중 어떤 오류를 발견하게 되면 SEQUENCE는 증가한 채로 남아버려 게시글을 모두 지우고 다시 만들어야했다. 마지막으로 파일 업로드기능과 다운로드 기능을 추가하였는데 책과 인터넷 모두 참고하여 만들었다. 파일 업로드 기능은 무난히 구현하였는데 다운로드 기능에서 만약 업로드 파일이 없었을 시에 계속 NULL POINTER EXCEPTION이 일어나 고생이었다. 디버깅을 해서 찾아보니 original name과 filename이 NULL로 나오게 되어 예외가 일어났는데 이건 파일이 없었을 때라는 조건문을 주어서 바꿔보려했더니 잘 되지않았는데 CHECK라는 변수를 하나 더 만들어 file!=null일 경우 true로 만들어 조건식을 생성해 내 original과 filename에 따로 파일없음이라는 스트링 값을 주어 해결하였다.

세 번째로는 투표기능을 구현했는데 이건 게시판과 기본 틀은 비슷하여 많이 어렵지는 않았다. 여기서 투표자 수를 수로 보여주기 보다는 비쥬얼로 보여주는 것이 더 깔끔하고 보기 편하다고 생각하여셔 rgb를 이용한 bar 구현을 하였다. 투표 기능을 만들면서 났던 에러는 작성 부분이었는데 8칸의 입력칸을 만약 순서대로 쓰지않고 띄어서 쓸 경우에 띈 부분 이후로는 입력이 안되는 오류가 발생하였다. 이 부분은 조건문을 수정해줌으로써 해결하였다.

마지막으로 대개 HTML만으로만 웹 출력을 하게 되다보니 웹 자체가 너무 허술하고 싸(?)보였다. 그래서 요번에 배워본 BootStrap과 Angular를 이용하여 디자인을 시도해보았다. BootStrap은 클래스와 속성 값을 이용한 디자인이라 많이 어렵지는 않았는데 Angular는 1과 2로 나누어져 1은 javascript언어를 쓰는 대신 연동이 어렵고 2는 typescript라는 언어를 써서 많이 어려웠다. 결국 라우터를 이용하여 웹 페이지들을 쉽게 관리하려했지만 기능을 완전히 구현하지 못하여 쓰지 않는 것으로 하였다. 게시판 꾸미기는 게시판 기능을 라이브러리로 불러와 추가하려하였는데 의외로 검색에서는 그렇게 자료가 많지않았다. 남이 올려놓은 라이브러리가 있었으나 그것은 게시판 자체의 라이브러리고 기능은 조금만 달려있었고 ciboard라는 게시판 라이브러리를 사용하려했으나 어떻게 연동시켜야하는지만 내내 하다가 결국 라이브러리 기능은 추가하지 못했다.

요번 IT종합설계를 하면서 웹에 대한 편견이 깨지고 보는 시야가 넓어졌다고 생각한다. 무언가 계속 결과가 보이다 보니까 더욱 재미를 느꼈고 특히 디자인을 하면서 깨끗하고 예쁘게 바뀌다보니 그것에 대한 뿌듯함을 느꼈다. 간단하다고 느꼈던 웹 사이트들의 한 페이지, 한 페이지에는 정말 내가 알지 못했던 많은 기능들이 들어있다는 것을 새삼 느꼈고 나도 언젠가는 그런 완벽한 사이트를 만들어보고 싶다는 생각이 들었다. 요번에 쓰지 못했던 node.js라던가 angular도 한번 공부해볼 예정이다. 또한 이번 개발에서는 pc용 웹을 만들었으니 다음번에는 앱 웹을 한번 구상해보고 싶다. 딱 웹 개발에 종사 하고 싶다 라고 마음이 든 것은 아니지만 웹 개발도 재밌고 배우는데 실증이 나지 않을 만큼 여러 분야와 기능이 있다는 것을 깨달았다.