

# 애플리케이션 배포 포트폴리오

서태곤, 최진영

2020 년 3 월 27 일

# 목 차

1. 프로젝트 개요 .....	2
2. 프로젝트 설명 .....	2
2.1 개발환경 .....	2
3. 기능 설계 .....	3
3.1 DB Table 설계 .....	3
3.2 기능 설계 .....	4
4. 프로그램 구현 및 상세내용 .....	9
4.1 프로그램 구현 .....	5
4.2 형상관리 .....	14
4.3 PMD툴을 이용한 Code Check .....	16
4.4 WAR 생성 및 배포 후 프로그램 확인 .....	17
4.5 배포 결과 및 문제 발생과 문제 해결 .....	19
5. 결과 .....	21

## 1. 프로젝트 개요

회원관리를 위한 앱 프로그램을 구현하고 배포한다.  
아래의 요구기능을 구현한다.

- 회원가입/ 로그인/ 회원정보수정/ 회원탈퇴에 대한 기능을 웹에서 제공
- 추가 기능 제공하여도 무방함
- 상기 4개 기능 중 2개 이상 정상동작을 해야 함

## 2. 프로그램 배포 환경

### 2.1 개발 환경

개발환경은 아래와 같다.

- eclipses ee 202001
- jdk 1.8
- ApacheTomcat 8.5.49
- IDE : eclipses

### 3. 기능 설계

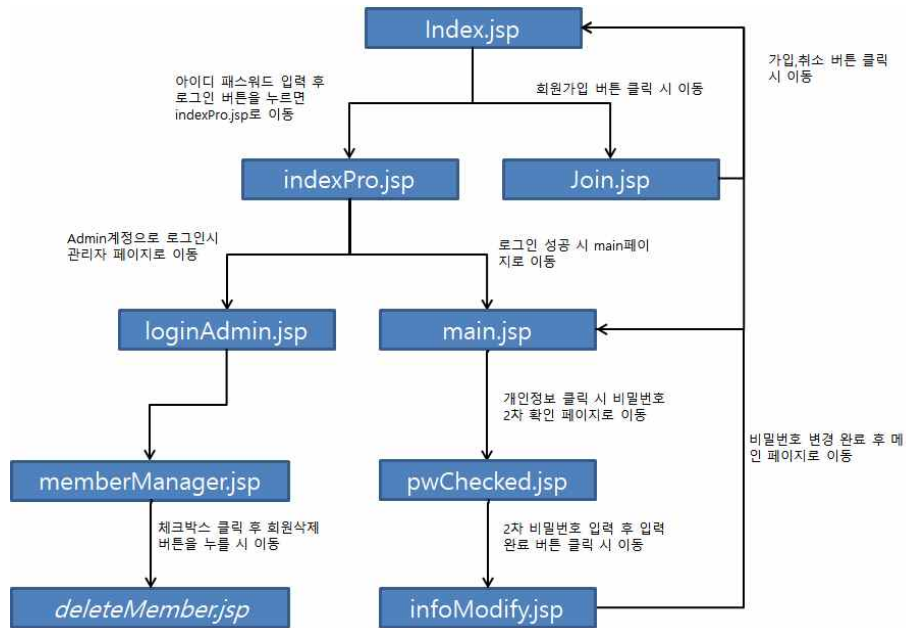
#### 3.1 DB Table 설계

Table을 다음과 같이 구성할 수 있도록

- 번호 : 회원 관리를 위한 회원 번호, 회원가입을 할 때마다 자동으로 증가
- 아이디 : 회원아이디
- 비밀번호 : 회원의 비밀번호
- 이름 : 회원의 이름 정보(한글 최대 5글자 까지 허용)
- 이메일 : 회원의 이메일 정보
- 회원가입 시간 : 회원 가입을 한 시간

이름	형식	추가 사항
NUM	최대 10자리 숫자	기본키, Null 비허용
ID	최대 50BYTE 크기 문자	Null 비허용
PASSWD	최대 16BYTE 크기 문자	Null 비허용
NAME	최대 15BYTE 크기 문자	Null 비허용
EM	최대 30BYTE 크기 문자	Null 비허용
REG_DATE	TIMESTAMP	Null 비허용

### 3.2 기능 설계



index는 첫 화면으로 ID, PW를 입력하는 부분과 로그인 버튼, 회원가입 버튼으로 설계 했다. ID, PW를 입력하고 로그인 버튼을 누르면 indexPro에서 관리자 계정과 일반 사용자 계정을 확인한 후 관리자 페이지와 사용자 페이지로 이동시켜 준다.

loginAdmin은 관리자 페이지로 회원 관리 버튼이 존재하고 회원 관리 버튼을 누르면 memberManager화면으로 넘어가고 DB에 저장된 데이터를 회원 번호를 기준으로 출력하게 설계하였다.

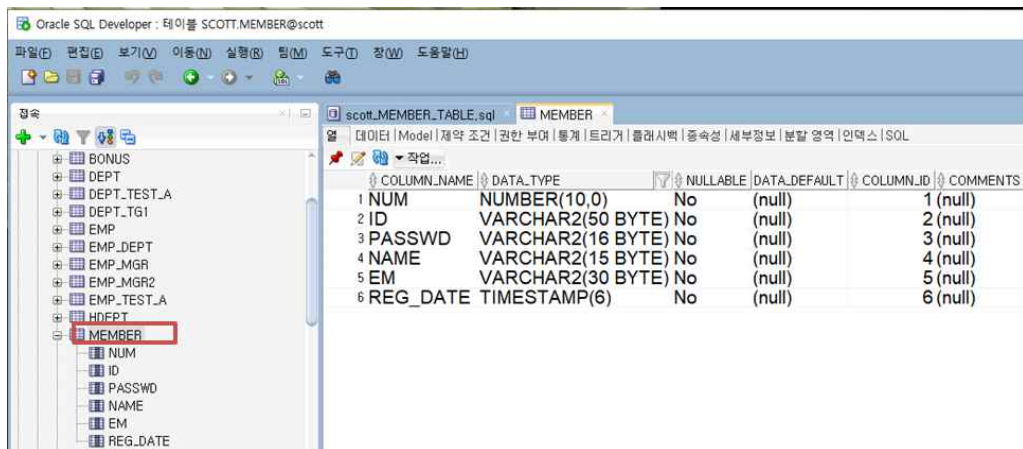
memberManager화면 에서는 표로 체크박스, 회원번호, 아이디, 비밀번호, 이름, 이메일 순으로 회원 정보를 표로 출력하고 체크박스 선택을 통해 회원 정보를 삭제하여 회원탈퇴 기능을 구현하도록 설계하였다. 체크박스를 클릭하고 회원탈퇴 버튼을 누르면 deleteMember에서 회원번호를 받아와 해당하는 회원정보를 삭제하는 것으로 회원탈퇴 기능이 되도록 설계하였다.

## 4. 프로그램 구현 및 상세내용

### 4.1 프로그램 구현

#### 4.1.1 DB Table 구현

```
CREATE TABLE MEMBER (  
  NUM NUMBER(10) PRIMARY KEY,  
  ID VARCHAR(50) NOT NULL,  
  PASSWD VARCHAR(16) NOT NULL,  
  NAME VARCHAR(15) NOT NULL,  
  EM VARCHAR(30) NOT NULL,  
  REG_DATE TIMESTAMP NOT NULL  
);
```



DB Table 설계에 맞게 SQL문을 작성하고 정상적으로 생성되는 것을 확인하였다.

#### 4.1.2 기능 구현



000사이트 입니다.

localhost:8090/PT\_1/index.jsp

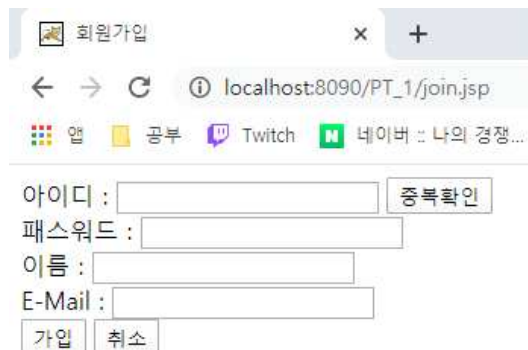
LOGIN

ID :

PW:

로그인 회원가입

가장 먼저 첫 화면인 index.jsp를 생성하고 ID, PW입력 하는 곳과 로그인, 회원가입 버튼을 구현하였다. ID와 달리 PW는 input type를 패스워드로 지정하여 비밀번호가 화면이 출력되지 않도록 구현하였으며 다른 모든 파일에도 동일하게 지정해 주었다.



회원가입

localhost:8090/PT\_1/join.jsp

아이디 :  중복확인

패스워드 :

이름 :

E-Mail :

가입 취소

회원가입 버튼을 누르면 회원가입 페이지가 나오는 join.jsp을 생성하고 ID, PW, 이름, 이메일을 입력받는 곳과 중복확인, 가입, 취소 버튼을 구현하였다. ID를 입력하고 중복확인 버튼을 클릭하면 DB안에 동일한 ID가 있는지 확인 후 중복 여부에 대해 알려줄 수 있도록 구현하였다.

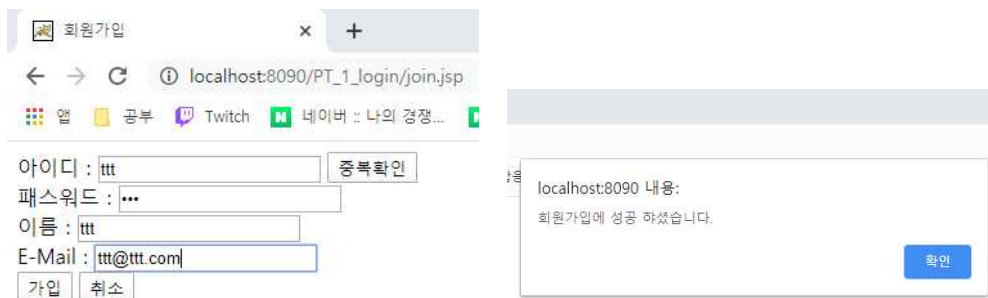


중복된 아이디를 입력하고 중복확인을 누르면 ‘중복된 아이디 입니다.’라는 경고창이 실행되고 확인을 누르면 다시 회원가입화면으로 이동하도록 구현하였습니다.



중복되지 않은 아이디를 입력하면 ‘사용 가능한 아이디 입니다.’라는 경고창이 실행되고 확인을 누르면 동일하게 다시 회원가입 화면으로 이동하도록 구현하였습니다.

아이디를 중복 여부 확인은 check\_id.jsp에서 DB와 통신을 통해서 확인하고 javascript의 alert문과 document문을 사용하여 경고창이 실행되고 확인 버튼을 누르면 join.jsp(회원가입 화면)으로 이동하도록 구현하였다.



정보를 모두 입력하고 회원가입 버튼을 누르면 ‘회원가입에 성공 하셨습니다.’라



는 경고창이 실행되고 확인하면 다시 회원가입 화면으로 이동하도록 구현하였습니다. 회원가입 기능은 join.jsp에서 입력된 정보를 joinPro.jsp에서 받아서 DB에 추가하도록 구현하였고 alert문과 document문을 사용하여 경고창이 실행되고 확인 버튼을 누르면 index.jsp(기본 화면)으로 이동하도록 구현하였다.

질의 결과 x

SQL | 인출된 모든 행: 6(0.002초)

NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com
2	1	test	test	test	test@aaa.com
3	2	aaa	aaa	aaa	aaa@aaa.com
4	3	bbb	bbb	bbb	bbb@bbb.com
5	4	ccc	ccc	ccc	ccc@ccc.com
6	5	ttt	ttt	ttt	ttt@ttt.com

회원 관리 x +

localhost:8090/PT\_1/memberManager.jsp

열 구분 Twitch 네이버 : 나의 계정... 야구 : 네이버 스포츠 Google YouTube 통합유통SW개발 자음

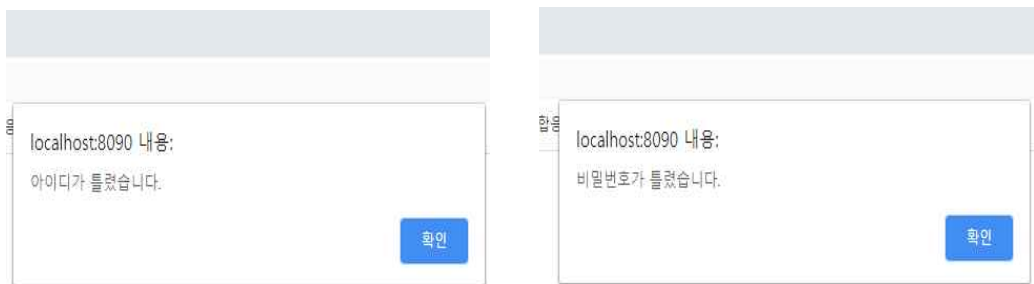
	회원 번호	아이디	패스워드	이름	이메일	가입일자
	1	test	test	test	test@aaa.com	2020-03-26 09:18:12.72
	2	aaa	aaa	aaa	aaa@aaa.com	2020-03-27 13:41:50.863
	3	bbb	bbb	bbb	bbb@bbb.com	2020-03-27 13:53:32.794
	4	ccc	ccc	ccc	ccc@ccc.com	2020-03-27 13:54:03.034
	5	ttt	ttt	ttt	ttt@ttt.com	2020-03-27 14:27:46.444

회원 삭제

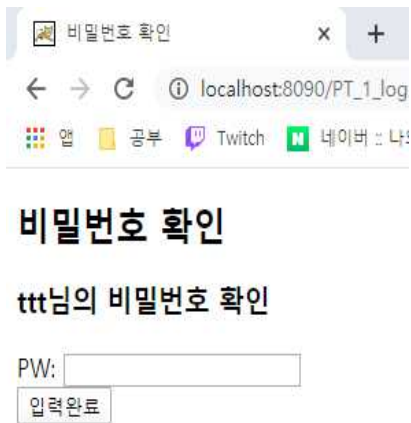
회원이입에 입력한 정보가 DB에 저장되어있는 것을 확인하였다.



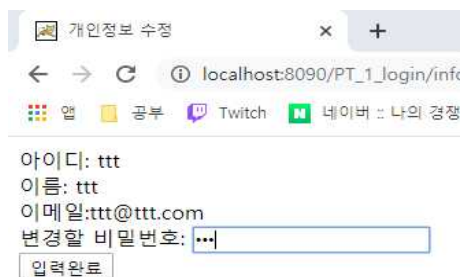
index.jsp(기본화면)에서 ID와 PW를 입력하고 로그인을 누르면 ID,PW가 모두 일치하면 회원정보수정을 할 수 있는 화면으로 이동할 수 있도록 구현하였다. 로그인을 할 때 index.jsp에서 DB와 통신하면서 ID와 PW가 모두 정확한지 확인하는 것을 구현하였다. 이 때 ID의 정보를 index.jsp에서 받아와 화면에 “ID’님이 로그인 하셨습니다.” 문구가 나타나도록 구현하였으며 이후 개인정보 수정에서도 정보를 받아 올 수 있도록 구현하였다. 로그아웃을 누르면 index.jsp로 돌아가고 그동안 저장했던 ID, PW, 이름, 이메일 같은 정보를 초기화 시켜주도록 구현하였다.



만약 ID나 PW가 일치하지 않으면 위 사진과 같이 경고를 띄우고 확인을 누르면 다시 기본화면으로 넘어오도록 구현하였다.



회원정보 수정 버튼을 누르면 다시 PW를 확인할 수 있는 페이지를 구현하였고 PW를 잘못 입력하면 경고창이 실행되고 다시 사진의 화면으로 돌아가도록 구현하였다. pwCheckedPro.jsp에서 DB와 통신을 통해 PW를 확인하고 맞으면 회원정보 수정 화면으로 넘어가고 잘못 입력했으면 경고창을 실행하고 다시 비밀번호 입력하는 화면으로 이동하도록 구현하였다. 올바른 비밀번호를 입력할 때 ID, 이름, 이메일 정보를 읽어와서 저장하였다가 다른 페이지로 전송할 수 있도록 구현하였다.



SQL | 인출된 모든 행: 6(0.002초)

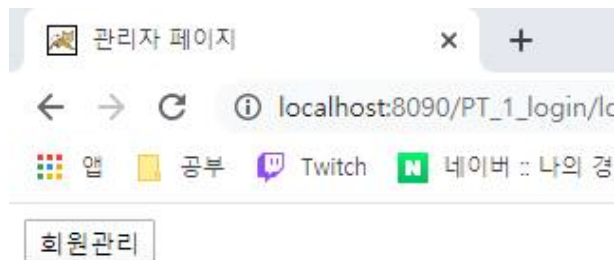
	NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com	20/03/26 09:18:03.803000000
2	1	test	test	test	test@aaa.com	20/03/26 09:18:12.720000000
3	2	aaa	aaa	aaa	aaa@aaa.com	20/03/27 13:41:50.863000000
4	3	bbb	bbb	bbb	bbb@bbb.com	20/03/27 13:53:32.794000000
5	4	ccc	ccc	ccc	ccc@ccc.com	20/03/27 13:54:03.034000000
6	5	ttt	ttt	ttt	ttt@ttt.com	20/03/27 14:27:46.444000000

SQL | 인출된 모든 행: 6(0.002초)

	NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com	20/03/26 09:18:03.803000000
2	1	test	test	test	test@aaa.com	20/03/26 09:18:12.720000000
3	2	aaa	aaa	aaa	aaa@aaa.com	20/03/27 13:41:50.863000000
4	3	bbb	bbb	bbb	bbb@bbb.com	20/03/27 13:53:32.794000000
5	4	ccc	ccc	ccc	ccc@ccc.com	20/03/27 13:54:03.034000000
6	5	ttt	aaa	ttt	ttt@ttt.com	20/03/27 14:27:46.444000000

회원정보 수정 화면에 들어오면 아이디, 이름, 이메일 정보를 읽어와서 화면에 출력하고 변경할 비밀번호를 입력하고 입력완료 버튼을 ‘누르면 비밀번호를 변경 하였습니다.’라는 경고창이 실행되고 확인을 누르면 다시 main.jsp(로그인된 상태) 화면으로 이동하도록 구현하였다. 기존의 비밀번호가 ‘ttt’에서 변경한 비밀번호인 ‘aaa’인 것을 데이터 베이스에서 확인하였다.



위 사진과 같이 기본 화면에서 관리자 계정으로 로그인 하게 된다면 관리자 페이지가 나오도록 구현하였다. 관리자는 회원관리를 통해 현재 가입되어 있는 회원정보와 회원을 탈퇴시킬수 있는 기능을 권한을 가질 수 있도록 구현하였다.

질의 결과 x

SQL | 인출된 모든 행: 6(0.002초)

	NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com	20/03/26 09:18:03.803000000
2	1	test	test	test	test@aaa.com	20/03/26 09:18:12.720000000
3	2	aaa	aaa	aaa	aaa@aaa.com	20/03/27 13:41:50.863000000
4	3	bbb	bbb	bbb	bbb@bbb.com	20/03/27 13:53:32.794000000
5	4	ccc	ccc	ccc	ccc@ccc.com	20/03/27 13:54:03.034000000
6	5	ttt	aaa	ttt	ttt@ttt.com	20/03/27 14:27:46.444000000

회원 관리 x +

localhost:8090/PT\_1\_login/memberManager.jsp

	회원 번호	아이디	패스워드	이름	이메일	가입일자
<input type="checkbox"/>	1	test	test	test	test@aaa.com	2020-03-26 09:18:12.72
<input type="checkbox"/>	2	aaa	aaa	aaa	aaa@aaa.com	2020-03-27 13:41:50.863
<input type="checkbox"/>	3	bbb	bbb	bbb	bbb@bbb.com	2020-03-27 13:53:32.794
<input type="checkbox"/>	4	ccc	ccc	ccc	ccc@ccc.com	2020-03-27 13:54:03.034
<input type="checkbox"/>	5	ttt	aaa	ttt	ttt@ttt.com	2020-03-27 14:27:46.444

회원 삭제

회원 관리 버튼을 클릭하면 위 사진과 같이 DB에 있는 회원 정보가 관리자를 제외한 상태에서 출력되도록 구현하였다. 회원 번호 앞의 체크박스를 체크된 상태에서 회원 삭제를 누르면 회원정보를 삭제하는 회원 탈퇴 기능을 구현하였다.

회원 관리 x +

localhost:8090/PT\_1\_login/memberManager.jsp

	회원 번호	아이디	패스워드	이름	이메일	가입일자
<input type="checkbox"/>	1	test	test	test	test@aaa.com	2020-03-26 09:18:12.72
<input type="checkbox"/>	2	aaa	aaa	aaa	aaa@aaa.com	2020-03-27 13:41:50.863
<input checked="" type="checkbox"/>	3	bbb	bbb	bbb	bbb@bbb.com	2020-03-27 13:53:32.794
<input type="checkbox"/>	4	ccc	ccc	ccc	ccc@ccc.com	2020-03-27 13:54:03.034
<input type="checkbox"/>	5	ttt	aaa	ttt	ttt@ttt.com	2020-03-27 14:27:46.444

회원 삭제

질의 결과 x

SQL | 인출된 모든 행: 6(0.002초)

	NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com	20/03/26 09:18:03.803000000
2	1	test	test	test	test@aaa.com	20/03/26 09:18:12.720000000
3	2	aaa	aaa	aaa	aaa@aaa.com	20/03/27 13:41:50.863000000
4	3	bbb	bbb	bbb	bbb@bbb.com	20/03/27 13:53:32.794000000
5	4	ccc	ccc	ccc	ccc@ccc.com	20/03/27 13:54:03.034000000
6	5	ttt	ttt	ttt	ttt@ttt.com	20/03/27 14:27:46.444000000

회원 삭제 버튼을 클릭하기 전에 다시 DB를 확인하고 회원 삭제 버튼을 클릭하면 '탈퇴 처리 되었습니다.'라는 경고창이 실행되고 확인을 누르면 다시 회원 정보가 출력되는 화면으로 돌아가도록 구현하였습니다.



회원 관리

localhost:8090/PT\_1/memberManager.jsp

	회원 번호	아이디	패스워드	이름	이메일	가입일자
	1	test	test	test	test@aaa.com	2020-03-26 09:18:12.72
	2	aaa	aaa	aaa	aaa@aaa.com	2020-03-27 13:41:50.863
	4	ccc	ccc	ccc	ccc@ccc.com	2020-03-27 13:54:03.034
	5	ttt	aaa	ttt	ttt@ttt.com	2020-03-27 14:27:46.444

회원 삭제

---

질의 결과

SQL | 인출된 모든 행: 5(0.001초)

	NUM	ID	PASSWD	NAME	EM	REG_DATE
1	0	admin	admin12345	admin	aaa@aaa.com	20/03/26 09:18:03.803000000
2	1	test	test	test	test@aaa.com	20/03/26 09:18:12.720000000
3	2	aaa	aaa	aaa	aaa@aaa.com	20/03/27 13:41:50.863000000
4	4	ccc	ccc	ccc	ccc@ccc.com	20/03/27 13:54:03.034000000
5	5	ttt	aaa	ttt	ttt@ttt.com	20/03/27 14:27:46.444000000

정상적으로 선택된 회원 정보가 삭제되고 DB를 확인하여 DB에서도 선택된 회원 정보가 삭제되는 것을 확인하여 회원 탈퇴 기능을 구현한 것을 확인했다.

## 4.2 형상관리

협업을 통해 프로그램 구현을 하기 위해 형상관리 프로그램을 이용하여 아래와 같은 형상관리 프로그램 도구를 선택하였다.

- Git/GitHub
- TortoiseGit
- URL : [https://github.com/Seotaegon/PO\\_project.git](https://github.com/Seotaegon/PO_project.git)
- 참여자 아이디  
: 서태곤([xorhs1510@gmail.com](mailto:xorhs1510@gmail.com)), 최진영([asdfghjkl095@naver.com](mailto:asdfghjkl095@naver.com))

협업을 하기위해 서태곤의 GitHub를 베이스로 선택하고 Manage access 기능으로 최진영을 추가하여 2명이서 프로그램을 구현할 수 있도록 하였다. Repositories의 이름은 PO\_project로 지정하였다.

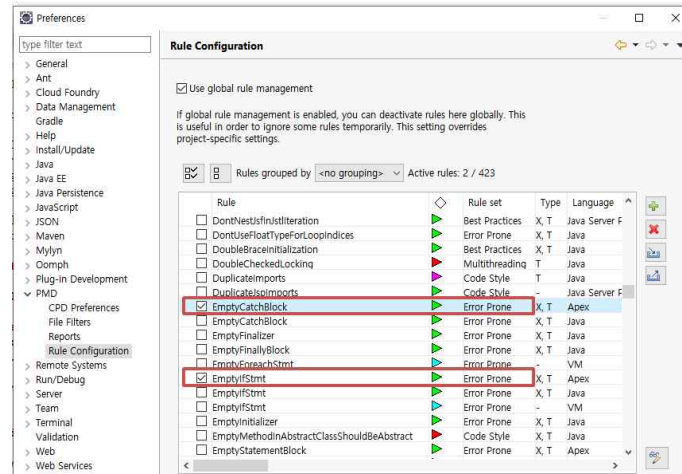


Commits on Mar 27, 2020		
ppt수정(구성도 재수정) choijinyoung-1 committed 5 minutes ago	48c78cb	<>
ppt수정(관계도 추가) choijinyoung-1 committed 36 minutes ago	b5196bd	<>
Merge branch 'master' of https://github.com/Seotaegon/PO_project tgseo committed 2 hours ago	99661c1	<>
War 배포사진 추가 tgseo committed 2 hours ago	72e2158	<>
history_포트폴리오_과정.xlsx 파일 수정 choijinyoung-1 committed 2 hours ago	f31fd1a	<>
sql문 업데이트,커밋구문 추가 choijinyoung-1 committed 3 hours ago	e5b71ae	<>
PPT 수정(Code Check 사진 추가) tgseo committed 3 hours ago	3e107cd	<>
Commits on Mar 26, 2020		
Merge branch 'master' of https://github.com/Seotaegon/PO_project choijinyoung-1 committed 21 hours ago	9dcecd4	<>
로그아웃, 비밀번호 2차확인, 비밀번호 변경기능 추가 choijinyoung-1 committed 21 hours ago	1ba232c	<>
SQL 코드 수정 tgseo committed 22 hours ago	c5c35cd	<>
회원 탈퇴 기능 추가 tgseo committed 22 hours ago	8a6d698	<>
sql 소스 코드 변경, 회원가입 기능, 관리자 기능 추가 tgseo committed yesterday	0f27605	<>
로그인, 개인정보 수정 기능추가 choijinyoung-1 committed yesterday	e3922c6	<>
최초생성 tgseo committed yesterday	5dd8cc3	<>

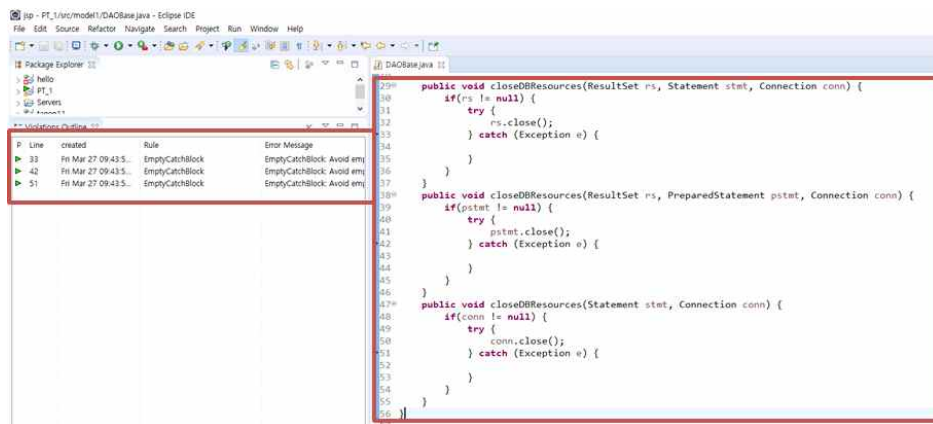
위 사진은 조원이 프로그램을 구현하기 위해 수행 후 수정사항을 올린 log 기록이다.



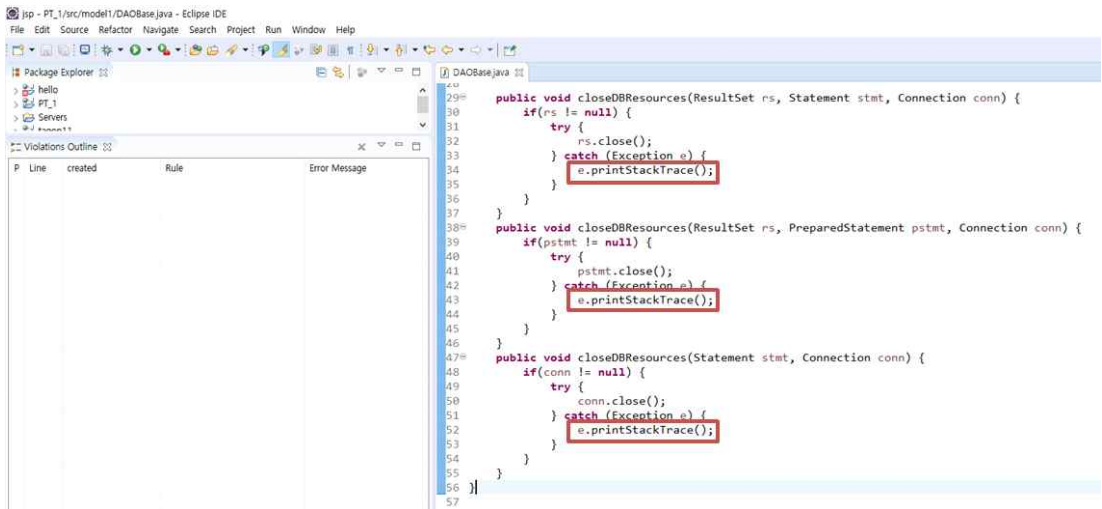
### 4.3 PMD툴을 이용한 Code Check



사진과 같이 EmptyCatchBlock(내용이 없는 Catch Block 존재), EmptyifStmt (빈 if 구문의 사용을 피하도록 함) 두 가지 룰을 선택하여 Code Check를 진행하였다.

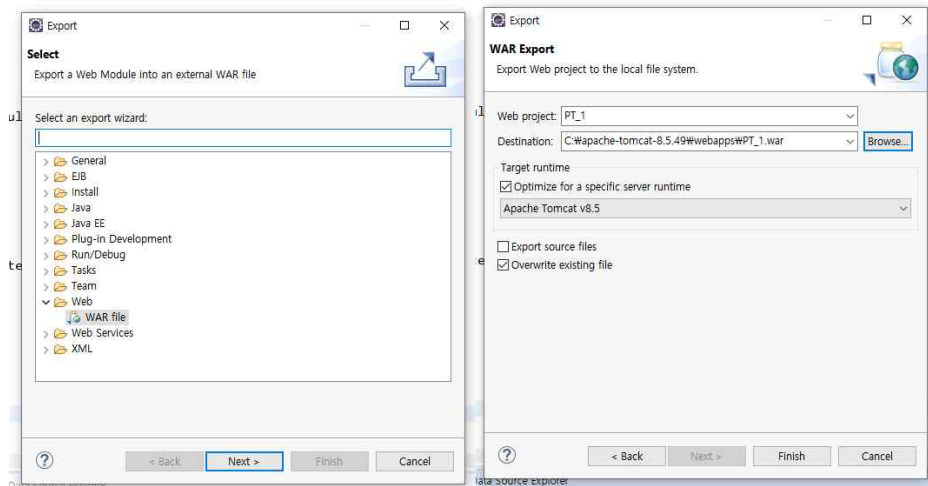


사진과 같이 32, 42, 51번 라인에 각각 EmptyCatchBlock의 룰에 걸리는 것을 확인할 수 있다. 실제 Code에서도 Catch구문 안에 내용이 없는 것을 확인하였다.



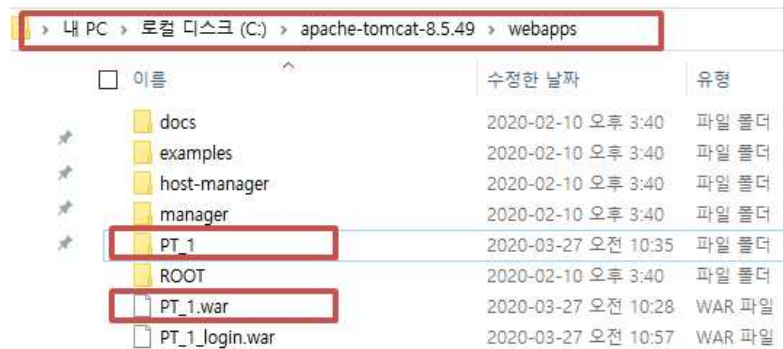
사진과 같이 Catch구문 안에 오류를 출력하는 Code를 추가하여서 PMD를 이용해 Code Check를 다시 실행했을 때 문제가 사라짐을 확인할 수 있다.

#### 4.4 WAR 생성 및 배포 후 프로그램 확인

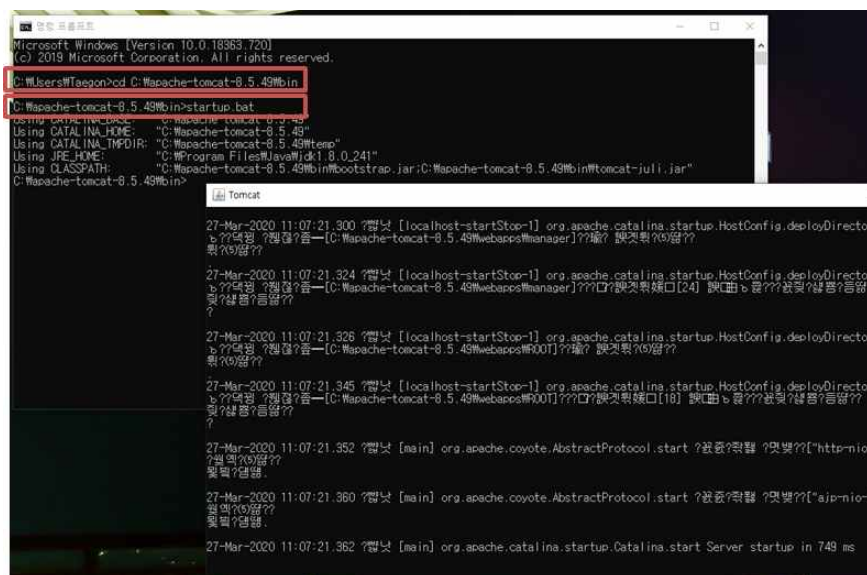


War파일을 생성하기 위하여 File-Export를 선택하고 Web-WAR file를 선택하

여 프로젝트의 이름과 저장 위치를 사진과 같이 설정한다. 그 후 형상관리 프로그램을 이용하여 서로 소스 파일을 공유하기 때문에 Export source files(소스를 확인할 수 있는 기능) 기능을 선택하지 않고 생성한다.

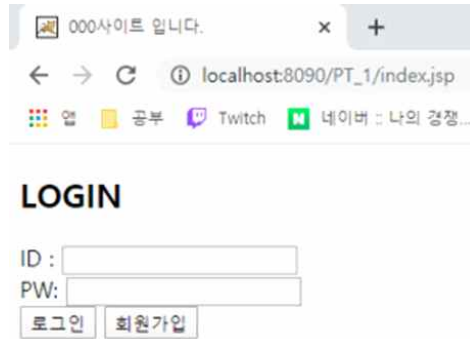


다음 사진과 같이 지정한 위치에 정상적으로 생성됨을 확인하였다.



사진과 같이 명령 프롬프트(CMD)를 실행하고 apache-tomcat-8.5.49 폴더안에 bin 폴더로 위치를 이동하여 startup.bat을 실행시킨다. 위 사진처럼 Tomcat Java

프로그램이 실행되고 서버에 정상적으로 접속된 것을 확인하였다.



localhost:8090(Server.xml 파일안에 지정한 포트번호)/PT\_1(WAR 파일이름)/index.jsp(첫화면 파일 명)을 웹주소에 입력하여 화면이 정상적으로 실행되는 것을 확인하였다.

## 4.5 배포 결과 및 문제 발생과 문제 해결



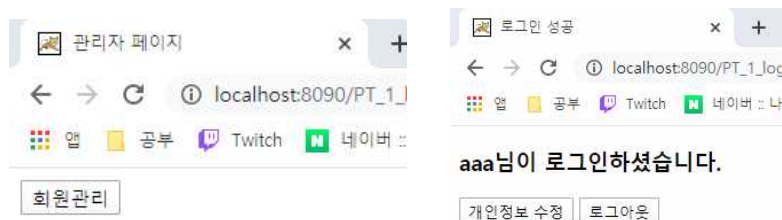
로그인 버튼을 누르면 로그인 화면이 아닌 오류가 발생하였다. 다른 페이지는 정상적으로 동작하는 것을 확인하였다. 먼저 WAR이 생성된 폴더로 이동하여 소스 파일들을 확인해 보았다.

내 PC > 로컬 디스크 (C:) > apache-tomcat-8.5.49 > webapps > PT_1			
이름	수정된 날짜	유형	크기
META-INF	2020-03-27 오전 10:35	파일 폴더	
WEB-INF	2020-03-27 오전 10:35	파일 폴더	
check_id.jsp	2020-03-26 오후 4:34	JSP 파일	2KB
checktest.html	2020-03-26 오후 12:12	Chrome HTML D...	2KB
checktest.jsp	2020-03-26 오전 11:41	JSP 파일	2KB
deleteMember.jsp	2020-03-26 오후 3:09	JSP 파일	2KB
index.jsp	2020-03-26 오전 10:24	JSP 파일	1KB
join.jsp	2020-03-26 오후 4:33	JSP 파일	2KB
joinPro.jsp	2020-03-26 오후 4:31	JSP 파일	2KB
loginAdmin.jsp	2020-03-26 오전 10:30	JSP 파일	1KB
memberManager.jsp	2020-03-26 오후 2:22	JSP 파일	3KB

위 사진과 같이 몇몇 파일들이 없는 것이 확인되었다. WAR파일을 만드는 과정에서 실수가 있었다는 것을 인지하고 WAR파일을 다시 생성하고 폴더 안에 소스파일들이 빠진 것이 없는지 확인했다.

내 PC > 로컬 디스크 (C:) > apache-tomcat-8.5.49 > webapps > PT_1_login			
이름	수정된 날짜	유형	크기
META-INF	2020-03-27 오전 11:07	파일 폴더	
WEB-INF	2020-03-27 오전 11:07	파일 폴더	
check_id.jsp	2020-03-26 오전 10:22	JSP 파일	2KB
checktest.jsp	2020-03-26 오전 11:41	JSP 파일	2KB
deleteMember.jsp	2020-03-26 오후 3:09	JSP 파일	2KB
index.jsp	2020-03-26 오후 4:22	JSP 파일	1KB
indexPro.jsp	2020-03-26 오후 12:51	JSP 파일	2KB
infoModify.jsp	2020-03-26 오후 4:22	JSP 파일	1KB
infoModifyPro.jsp	2020-03-26 오후 4:22	JSP 파일	2KB
join.jsp	2020-03-26 오전 9:31	JSP 파일	2KB
joinPro.jsp	2020-03-26 오전 10:19	JSP 파일	2KB
loginAdmin.jsp	2020-03-26 오전 10:30	JSP 파일	1KB
main.jsp	2020-03-26 오후 4:22	JSP 파일	1KB
memberManager.jsp	2020-03-26 오후 3:11	JSP 파일	3KB
pwChecked.jsp	2020-03-26 오후 4:22	JSP 파일	1KB
pwCheckedPro.jsp	2020-03-26 오후 4:22	JSP 파일	2KB

위 사진은 모든 소스 파일이 있는 것을 확인하고 다시 웹을 실행할 때 다음과 같이 로그인이 되지 않는 문제가 발생하지 않고 잘 되는 것을 확인하였다.



## 5. 결과

회원관리를 위한 웹 프로그램을 요구사항에 맞게 Git/GitHub를 이용하여 협업하고 로그를 통해 과정의 진행사항을 확인하였다. PMD툴을 이용한 Code Check를 하여서 문제 부분을 찾고 수정했고 완성된 프로그램을 WAR파일로 배포하였다. WAR파일로 배포하여 실행했을 때 로그인이 되지 않는 문제가 발생하고 이의 원인을 찾아 WAR을 새로 생성하고 재배포해서 해결하였다. 그 후 각 구현된 프로그램이 정상적으로 동작하는지 확인했으며 아래와 같은 기능들을 구현하였다.

- 메인화면에 ID, PW 입력을 통해 로그인 기능이 가능하게 함
- 회원가입 기능을 구현하고 동일한 ID의 여부를 확인할 수 있는 기능을 구현함
- 로그인 할 때 관리자와 일반 사용자를 확인하고 각각 맞는 화면으로 이동하는 기능을 구현함
- 일반 사용자가 로그인이 성공했을 때 회원 정보 수정하는 기능을 통해 비밀번호를 변경할 수 있도록 함
- 관리자 화면에서는 회원들의 정보를 확인하고 삭제를 통한 회원 탈퇴 기능을 구현함

향후 추가할 필요성이 있는 기능은 아래와 같다.

- 회원가입을 할 때 동일한 ID 판단후 입력한 문자가 사라지지 않고 유지되는 기능
- 일반 사용자가 로그인 성공했을 때 회원 탈퇴 기능
- 관리자가 회원 정보를 확인하는 화면에서 여러개의 정보를 한번에 삭제할수 있고 전체 선택, 전체 해제 기능