

JSP 웹 개발을 위한 환경 구성

- ❖ jdk1.8 설치 - <http://www.oracle.com>
- ❖ Oracle11g 설치 - <http://www.oracle.com>
 - sqldeveloper 압축해제 및 실행
- ❖ Tomcat8 다운로드(압축파일) - <http://www.apache.org>
 - 압축해제
- ❖ Eclipse EE 다운로드 및 압축해제 - <http://www.eclipse.org>
 - workspace 폴더 생성
 - eclipse 실행
 - Window - Preferences - Server - Runtime Environment - Tomcat추가

설문조사 애플리케이션

요구사항 확인

❖ 연봉 설문조사 애플리케이션

■ 설문 항목

- 연령대(10대, 20대, 30대, 40대, 50대, 60대, 기타)
- 성별(남, 여)
- 현재연봉(1000만원, 2000만원, 3000만원, 4000만원, 5000만원, 6000만원 이상)
- 희망연봉(1000만원, 2000만원, 3000만원, 4000만원, 5000만원, 6000만원 이상)

■ 설문 결과

- 연령대 별 현재연봉 평균
- 연령대 별 희망연봉 평균
- 성별 별 현재연봉 평균
- 성별 별 희망연봉 평균
- 전체 현재연봉 평균
- 전체 희망연봉 평균

요구사항 확인

❖ 연봉 설문조사 애플리케이션

■ 설문 항목 세부 사항

➤ 모든 항목에 다음과 같은 값을 지정하여 설문 값을 저장

(연령)AGE	(성별)GENDER	(현재연봉)SALARY	(희망연봉)SALARYDESIRED
10대 - 10	남자 - “남자”	1000만 - 1000	1000만 - 1000
20대 - 20	여자 - “여자”	2000만 - 2000	2000만 - 2000
30대 - 30		3000만 - 3000	3000만 - 3000
40대 - 40		4000만 - 4000	4000만 - 4000
50대 - 50		5000만 - 5000	5000만 - 5000
60대 - 60		6000만 이상 - 6000	6000만 이상 - 6000

DB설계

❖ 설문 저장 테이블

NUMBER(PK)	AGE	GENDER	SALARY	SALARYDESIRED
SURVEYDATA.nextval	10	“남자”	1000	2000
SURVEYDATA.nextval	30	“여자”	2000	3000

```
create table "SURVEYDATA"(  
  "NO" number(3,0) primary key not null,  
  "AGE" number(2,0) not null,  
  "GENDER" varchar2(10) not null,  
  "SALARY" number(4,0) not null,  
  "SALARYDESIRED" number(4,0) not null  
);
```

```
create sequence "SURVEYDATA_SEQ"  
  start with 1  
  increment by 1  
  maxvalue 99999  
  nocycle  
  noorder  
  nocache;
```

기능설계

❖ 설문조사 입력 폼

- 설문조사 입력 화면 출력

❖ 설문처리

- 입력 폼에서 사용자 입력 값 전송(DAO, DTO활용 데이터베이스에 저장)

❖ 설문결과

- 결과 요청에 응답(DAO, DTO활용 데이터베이스에서 값 조회)

URL 설계

❖ survey 폴더

- serveyForm.jsp
- serveyProc.jsp
- serveyResult.jsp

❖ survey/error 폴더

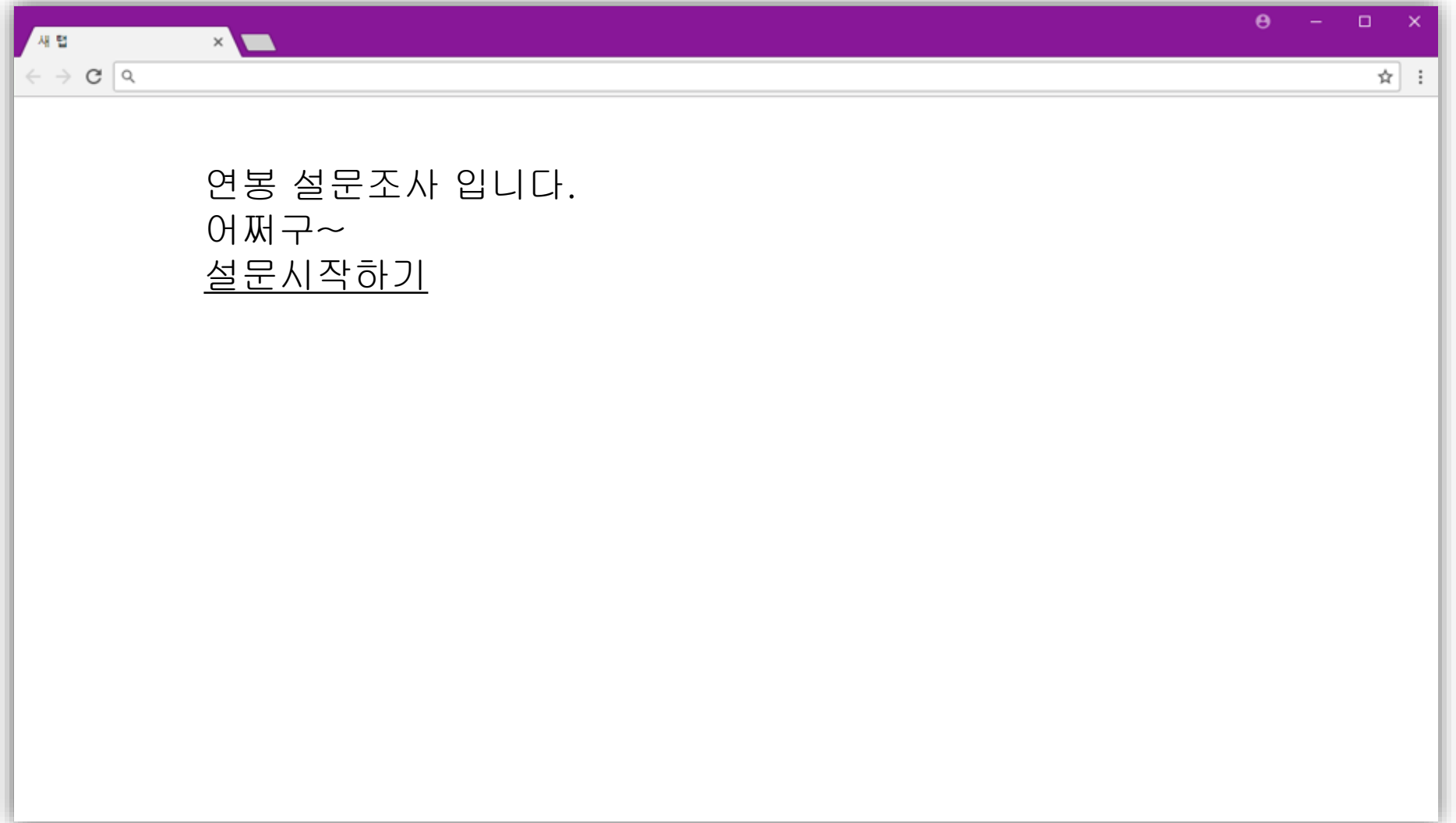
- error404.html
- error500.html

❖ survey/resource 폴더

- js파일
- css파일
- 이미지파일

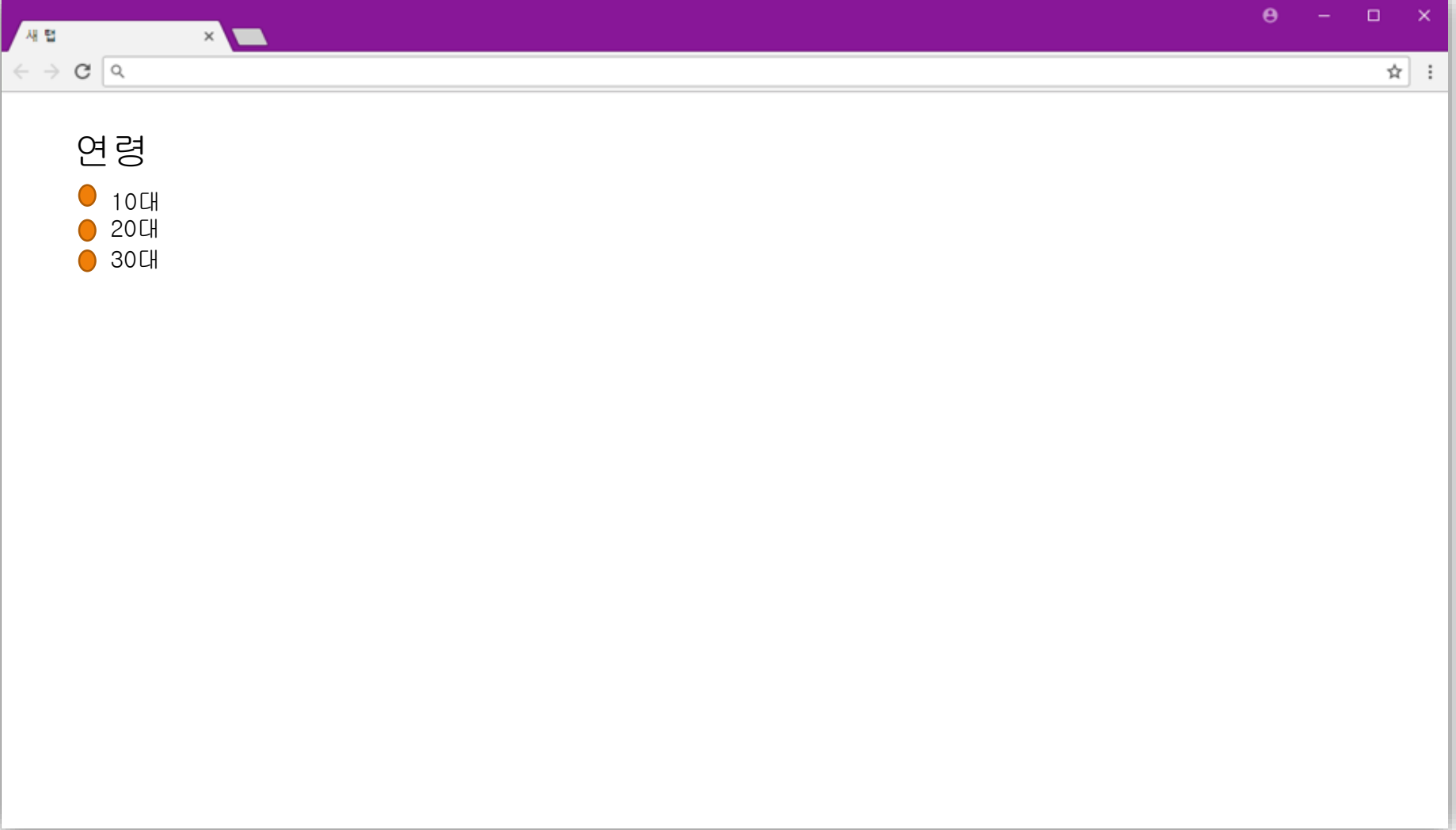
화면 UI설계

❖ 설문 메인



화면 UI설계

❖ 설문 입력



새 탭 x

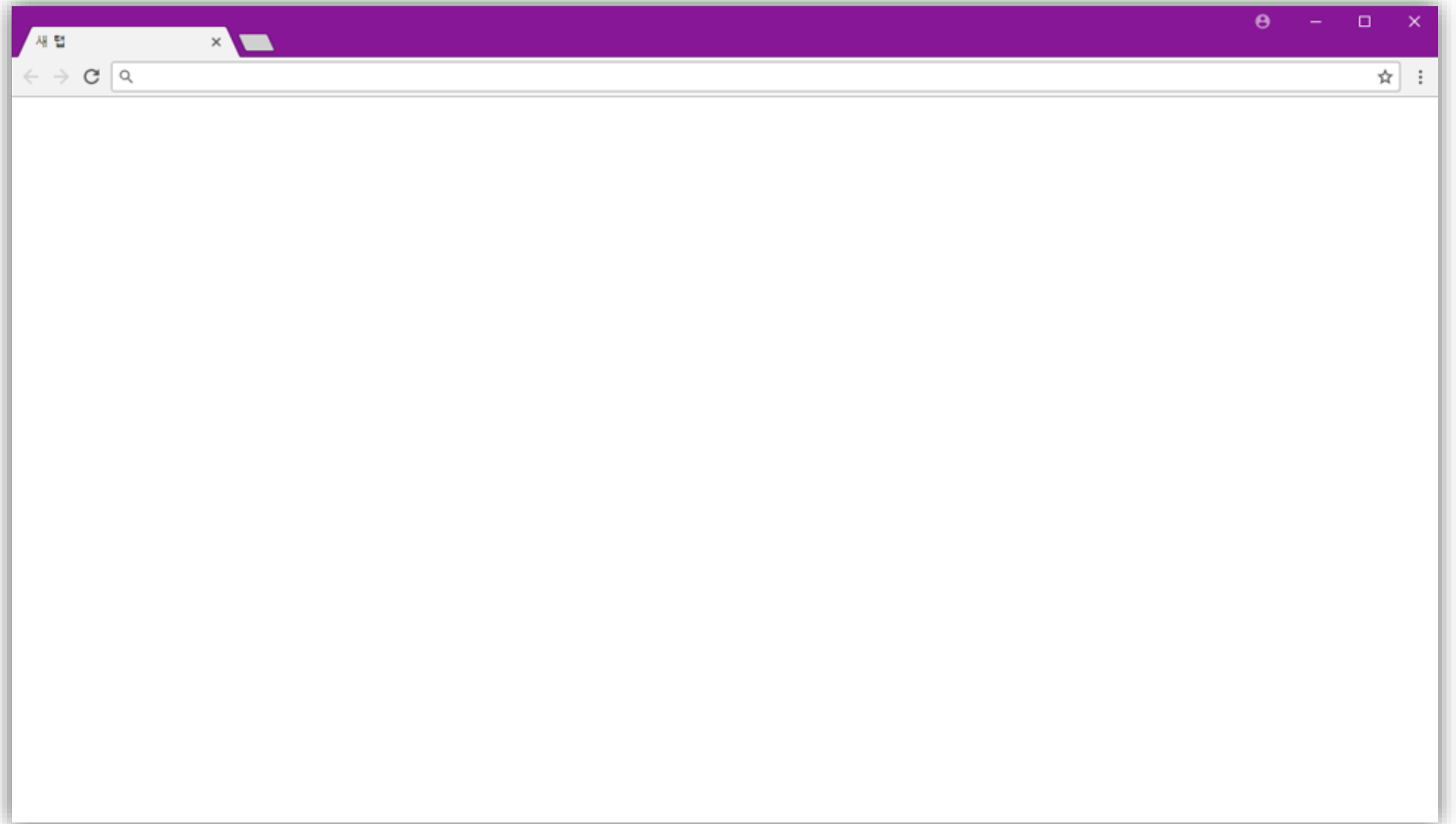
← → ↻ 🔍 ☆ ⋮

연령

- ☐ 10대
- ☐ 20대
- ☐ 30대

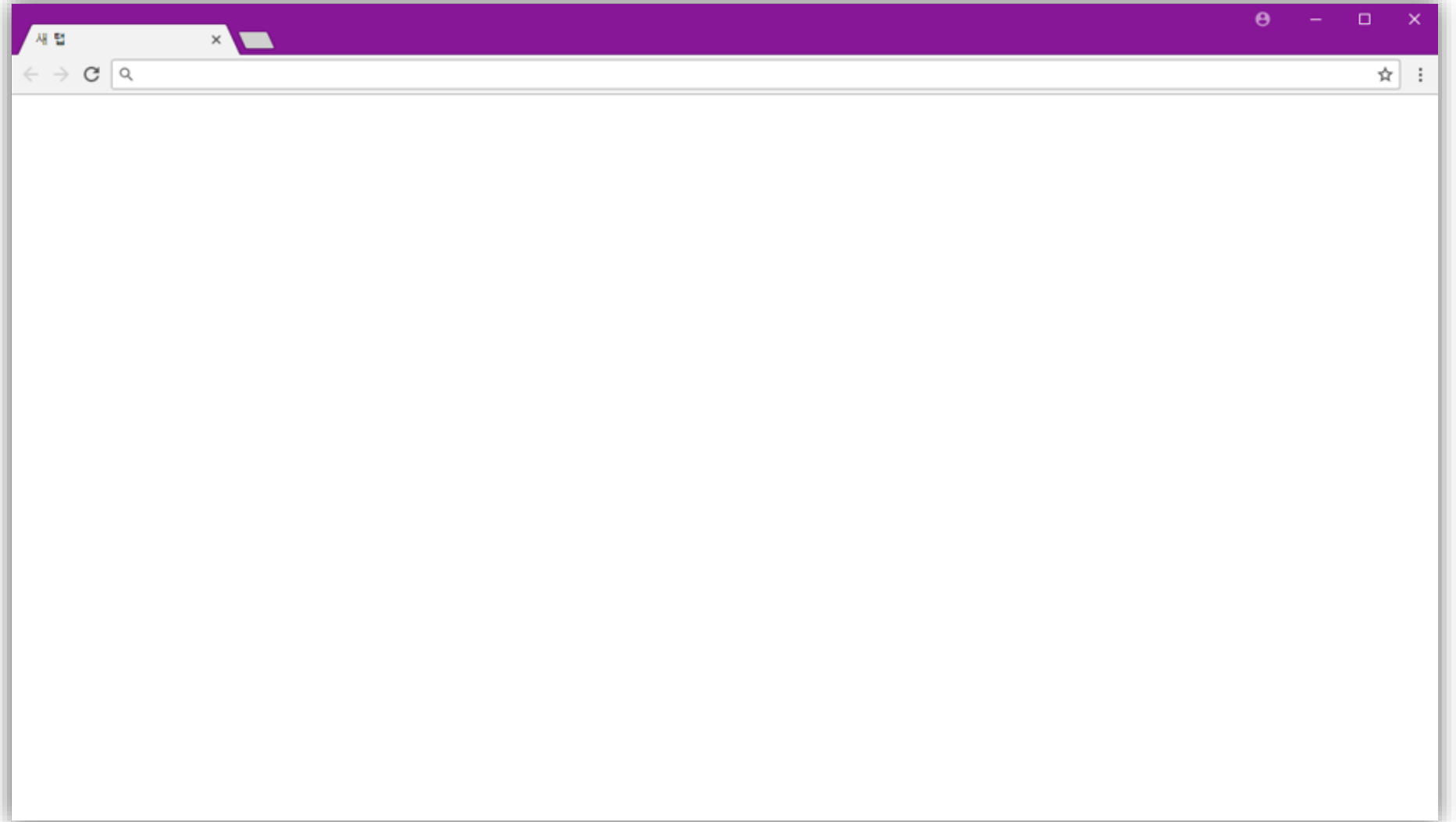
화면 UI설계

❖ 설문 결과



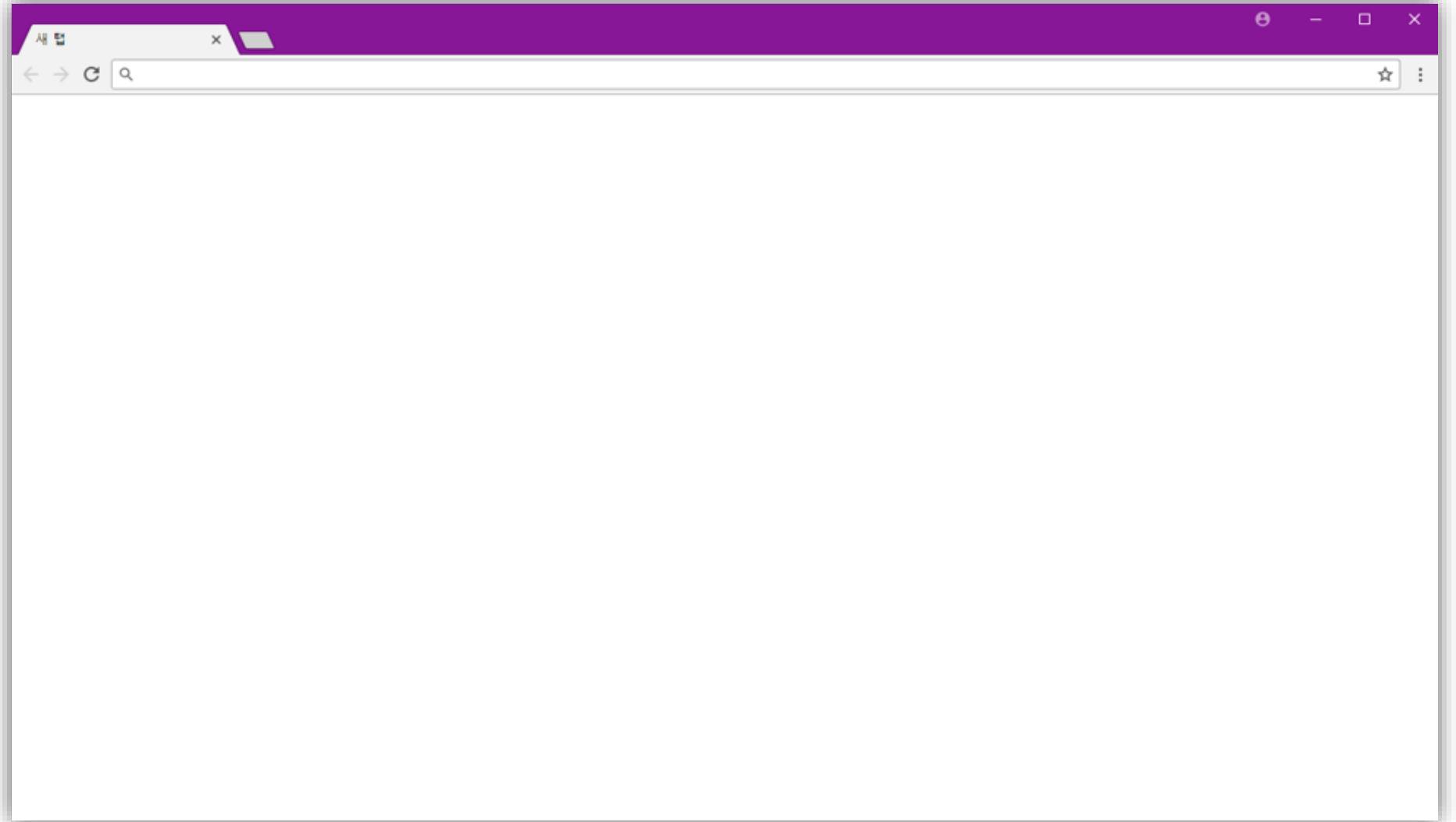
화면 시설계

- ❖ 에러 화면
 - code 404



화면 시설계

- ❖ 에러 화면
 - code 500



구현

❖ 단계별로 구현하여 여러 방법을 실습한다.

- ① JSP페이지만을 이용하여 구현 – simplesurvey
 - jsp페이지에서 DB연동 등 모든 기능 구현
- ② DTO, DAO를 이용하여 리팩토링(DB접속 기능 분리하기) – simplesurvey2
 - DAO와 DTO클래스 확인 및 이를 사용하는 surveyProc.jsp파일의 코드 분석
- ③ jsp의 Action Tag, Beans 사용하여 리팩토링 – simplesurvey3
 - surveyForm.jsp에서 넘긴 요청 파라미터를 받는 surveyProc.jsp에서 빈즈사용 코드 분석
- ④ DAO객체에 대해 싱글톤 패턴 적용하여 리팩토링 – simplesurvey4
 - DAO클래스의 싱글톤 적용 코드 분석 및 객체가 한번만 생성되는 것 확인, 싱글톤 클래스는 빈즈로 사용불가
- ⑤ Connection Pool 적용하여 리팩토링 – simplesurvey5
 - META-INF/context.xml, WEB-INF/web.xml, ConnectionUtil클래스 확인

❖ 최종목표

- 전체 기능을 MVC패턴을 적용하여 리팩토링

MVC 적용

- ❖ simplesurveyMVC 프로젝트 내용 분석
 - 기존의 방식과 MVC패턴을 적용한 방식의 차이
 - web.xml
 - servlet 설정 및 servlet-mapping 설정 이해
 - initParameter와 Properties사용 이해
 - 소스코드
 - Controller의 역할 및 기능 이해
 - CmdAction을 구현하는 Action클래스들의 동작 원리 이해
 - doGet, doPost메서드 이해
 - RequestDispatcher의 기능 이해
 - view에 해당하는 jsp파일을 /WEB-INF/하위에 위치
 - 클라이언트에서 jsp파일을 직접 요청하는 것을 막기 위함
 - WEB-INF폴더는 서버에서만 접근이 가능함

구현

❖ UML을 활용하여 애플리케이션 설계 및 동작을 표현

- Usecase 다이어그램
- Sequence 다이어그램
- Class 다이어그램

❖ GIT 등을 활용하여 버전관리

❖ JUnit과 같은 도구를 활용하여 동작에 대한 단위테스트 진행

개발자테스트

- ❖ 요구사항에 따른 전체 기능에 대한 동작 테스트
 - 설문조사 폼
 - 설문진행
 - 설문결과