##프로젝트 수행정보

프로젝트	쇼핑 플랫폼 사용자 인터페이스 구현	개발기간	2024.09 ~ 2024.10 (2개월)
사용기술	HTML5, CSS3	DBMS	
TOOL	vs Code	WAS	
프로젝트 상세내용	사용자가 쇼핑 플랫폼을 쉽게 탐색하고 이용할 수 있도록 다양한 기능을 제공하는 사용자 중심의 인터페이스를 설계 및 개발하였습니다. 본 프로젝트는 접근성, 직관성, 반응형 설계에 중점을 두었으며, 다양한 메뉴와 동적 이벤트를 통해 매끄럽고 효율적인 사용자 경험을 제공하였습니다. **※ 구현한 주요 기능 :** **1. 헤더 및 상단 네이게이션 설계** ▶ 검책창 - 헤더에 검색창을 고정 배치하여 언제든지 접근 가능하도록 설계. - 사용자가 입력한 키워드에 따라 특정 식당이나 음식 항목을 빠르게 찾을 수 있도록 구현. ▶ 로그인 및 회원가입 버튼 - 로그인 버튼 클릭 시 팝업 폼이 나타나도록 JavaScript를 활용하여 인터랙션 구현. - 회원가입 버튼을 별도로 배치하여 사용자 편의성을 고려. **2. 메인 메뉴 구성** - 상단에 "인기순," "지역별," "종류별" 메뉴를 배치하여 사용자가 원하는 카테고리를 빠르게 탐색할 수 있도록 설계. - 각 메뉴는 클릭 시 스크롤 애니메이션을 활용해 해당 섹션으로 부드럽게 이동하도록 구현. **3. 로그인/회원가입 시스템** ▶ 팝업 창을 이용한 로그인 폼 구현: - 사용자 ID와 비밀번호 입력 필드 제공. - 입력값 검증 및 로그인 성공/실패 메시지 표시 (e.g., "로그인 성공" 또는 "정보가 없습니다"). - 로그인 실패 시 사용자에게 명확한 피드백을 제공. ▶ 로그인 폼 닫기 버튼을 통해 폼을 종료할 수 있도록 구현. **4. 콘텐츠 카테고리화 및 시각적 구성** ▶ 인기순 섹션: - 인기 있는 음식점 및 아이템을 카드 형식으로 구성. - 각 카드에는 음식점 사진, 이름, 간략한 설명을 포함하며, 마우스 오버 시 추가 정보가 표시되도록 설계. - 클릭 시 새로운 창에서 상세 정보로 이동. ▶ 지역별 섹션: - 지역(예: 하대동, 평거동, 충무공동)별로 음식점을 분류하여 콘텐츠 구성. - 지역별 음식점을 슬라이드 및 카테고리 블록 형식으로 표현. ▶ 종류별 섹션: - 음식 종류(예: 고기, 국밥 등)별로 상품을 그룹화하여 사용자가 관심 있는 항목을 쉽게 탐		

색 가능하도록 구현.

5. 반응형 웹 디자인 적용

- CSS 미디어 쿼리를 활용하여 다양한 디바이스(PC, 태블릿, 모바일)에서 콘텐츠가 잘 보이도록 설계.

- 화면 크기에 따라 요소 크기와 배치를 유동적으로 조정.

6. 사용자 인터랙션 향상

▶ JavaScript 를 활용해 hover 효과와 클릭 이벤트를 적용:

- 상품 카드에 마우스를 올리면 설명이 나타나고 강조 효과 추가.

- 특정 요소 클릭 시 새로운 창이 열려 상세 정보 제공.

▶ 검색 초기화 기능:

- 검색창의 내용을 버튼 클릭으로 즉시 초기화할 수 있도록 설계.

▶ 설명란 토글 기능을 통해 사용자가 원하는 정보만 선택적으로 볼 수 있도록 구현.

※ 성과

디자인 및 사용자 경험 최적화

- 최신 웹 디자인 트렌드를 반영한 UI 로 사용성이 높은 쇼핑 플랫폼을 완성.

- 직관적이고 효율적인 네비게이션 구조로 사용자 만족도 향상.

기능성 강화

- 동적 이벤트와 반응형 설계를 통해 다양한 환경에서도 원활한 작동을 보장.

- 간단하고 직관적인 검색 및 로그인 시스템 구현으로 사용자 접근성 개선.

코드 품질 및 확장성

- 모듈화된 구조와 깔끔한 코드로 유지보수와 기능 확장이 용이.

- JavaScript 를 활용한 동적 기능 구현으로 재사용성과 유연성 확보.

이 프로젝트를 통해 사용자 중심의 인터페이스 설계와 다양한 기능 구현 경험을 쌓았으며, 웹 개발 전반에 걸친 기술적 이해도를 높였습니다. 특히, 반응형 웹 및 동적 콘텐츠 구현 능력을 크게 향상시킬 수 있었습니다.

프로젝트	사용자 관리 시스템 (CRUD 기능 구현)	개발기간	2024.11 ~ 2025.12 (2개월)
사용기술	Java (JSP, Servlets), HTML, CSS	DBMS	Oracle
TOOL	Eclipse IDE	WAS	Apache Tomcat
프로젝트 상세내용	사용자 데이터를 효율적으로 관리하기 위한 CRUD 웹 애플리케이션을 Eclipse IDE 와 JSP 를 사용해 개발했습니다. 로그인 및 회원 정보를 기반으로 한 사용자 등록, 조회, 수정, 삭제 기능을 제공하며, 사용자 친화적인 인터페이스와 안정적인 데이터 처리를 목표로 설계되었습니다. **※ 구현한 주요 기능 :** **1. 사용자 등록 (Create)** ▶ UI: - JSP 를 활용해 사용자 정보를 입력받는 등록 화면(mypage_top.jsp)을 구현. - 이름, 이메일, 비밀번호 등 필수 정보를 입력받도록 구성. - 입력값 누락 시 알림 메시지를 출력하여 데이터 검증. ▶ 백엔드: - 사용자 등록 요청을 처리하는 서블릿을 작성하여, 입력 데이터를 데이터베이스(Oracle)에 저장. - PreparedStatement 를 활용해 SQL 인젝션 방지 및 데이터 보안 강화. - 등록 성공 후 사용자에게 환영 메시지를 출력하고 로그인 페이지로 리다이렉트. **2. 사용자 조회 (Read)** ▶ UI: - login_page.jsp 를 통해 사용자가 로그인 시 자신의 정보를 확인할 수 있는 마이페이지로 이동. - 사용자 정보 조회 화면에 이름, 이메일, 가입일 등 주요 정보를 출력. ▶ 백엔드: - 로그인 시 입력한 이메일과 비밀번호를 기반으로 데이터베이스에서 사용자 정보를 조회. - 사용자가 로그인 상태에서만 접근 가능하도록 세션을 활용해 인증 처리. **3. 사용자 수정 (Update)** ▶ UI: - 사용자 정보를 수정할 수 있는 수정 화면(modify2.jsp)을 구현. - 기존 데이터를 불러와 사용자 편집이 용이하도록 구성. - 입력값 검증을 추가하여 잘못된 데이터 입력을 방지. ▶ 백엔드: - 수정 요청을 처리하는 서블릿 작성. - 데이터베이스에서 해당 사용자의 기존 데이터를 업데이트하는 SQL 쿼리를 실행. - 데이터 업데이트 후 성공 메시지를 출력하고 마이페이지로 리다이렉트. **4. 사용자 삭제 (Delete)** ▶ UI: - 마이페이지에서 "계정 삭제" 버튼을 클릭하면 확인 창을 출력하여 삭제를 진행하도록 구현.		

- 계정 삭제 후 로그아웃 화면(logout.jsp)으로 리다이렉트.

▶ 백엔드:

- 사용자가 계정 삭제 요청을 보내면 해당 사용자 데이터를 데이터베이스에서 삭제.
- 삭제 성공 시 세션을 무효화하고, 로그아웃 처리.

※ 추가 구현 기능 :

1. 로그인 및 로그아웃 기능

- 로그인 시 세션을 생성하여 사용자의 인증 상태를 유지.
- 잘못된 로그인 정보 입력 시 오류 메시지를 출력하며, 재로그인을 유도.
- 로그아웃 시 세션을 무효화하고 로그인 화면으로 이동.

2. 데이터 보안 강화

- 사용자 비밀번호는 해시 알고리즘을 사용하여 데이터베이스에 저장.
- SQL 쿼리 실행 시 PreparedStatement 를 활용해 보안 위협을 방지.

3. 사용자 경험 개선

- 알림 메시지와 화면 전환 애니메이션을 추가하여 사용자 친화적 UI 제공.
- 에러 발생 시 명확한 메시지를 출력하여 사용자가 문제를 인지하고 해결할 수 있도록 도움.

※ 성과

Eclipse IDE 와 JSP 활용 능력 향상

- JSP 와 서블릿을 활용한 웹 애플리케이션 설계 및 구현 능력을 배양.
- MVC 패턴을 적용하여 프론트엔드와 백엔드의 역할을 명확히 분리.

세션 관리와 사용자 인증

- 세션을 활용해 로그인 상태를 유지하고, 인증되지 않은 사용자의 접근을 제한하는 로직 구현.

보안 고려

- SQL 인젝션 방지 및 데이터 무결성 유지.
- 사용자 비밀번호 암호화를 통해 데이터 보안 강화.

디버깅 및 문제 해결

- Eclipse IDE 를 사용해 디버깅하며 SQL 쿼리 및 데이터베이스 관련 오류를 해결.
- 예외 처리 로직을 추가하여 예상치 못한 오류 상황에 대처.

이 프로젝트를 통해 JSP 와 서블릿을 활용한 CRUD 기능 구현과 데이터베이스 연동 기술을 심화할 수 있었습니다.
특히, 사용자 인증과 데이터 보안을 고려한 설계를 통해 안정적인 시스템 구축의 중요성을 배웠습니다.
이 경험은 향후 더 복잡한 웹 애플리케이션을 설계하고 개발하는 데 큰 도움이 될 것입니다.

<table>
<tr><td>프로젝트</td><td>학생 성적 관리 시스템 (CRUD 기능 구현)</td><td>개발기간</td><td>2024.12 ~ 2025.01 (2개월)</td></tr>
<tr><td>사용기술</td><td>Java, 스프링부트3, HTML, CSS, JavaScript
Rest API, Json, Ajax</td><td>DBMS</td><td>Oracle</td></tr>
<tr><td>TOOL</td><td>IntelliJ IDEA, Junit, Apache Jmeter</td><td>WAS</td><td>Apache Tomcat</td></tr>
<tr><td>프로젝트 상세내용</td><td colspan="3">이 프로젝트는 학생의 성적 데이터를 효율적으로 관리하기 위한 CRUD 웹 애플리케이션을 개발하는 것을 목표로 했습니다. Spring Boot 와 JPA 를 사용해 백엔드 로직을 구현하고, Mustache 템플릿 엔진을 활용해 사용자 친화적인 UI 를 제작했습니다.
주요 기능으로는 학생 등록, 조회, 수정, 삭제가 포함되며, 성적의 합계 및 평균을 자동으로 계산하고 출력하는 기능을 추가하여 데이터 처리의 편리성을 높였습니다.

※ 구현한 주요 기능 :
1. 학생 등록 (Create)
▶ UI:
- Mustache 템플릿 엔진을 사용해 학생 등록 화면(insert.mustache) 제작.
- 이름, 국어, 영어, 수학 점수를 입력받아 등록할 수 있는 폼 구현.
▶ 백엔드:
- /students/insert 엔드포인트에서 POST 요청을 처리하여 새로운 학생 데이터를 등록.
- StudentService 를 통해 데이터베이스에 저장.
- 성공적으로 등록된 후 리스트 페이지로 리다이렉션 처리
2. 학생 조회 (Read)
▶ UI:
- 학생 목록 화면(studentlist.mustache)에 등록된 모든 학생의 데이터와 성적 합계 및 평균을 표 형식으로 출력.
- 각 학생의 이름을 클릭 시 해당 학생의 상세 정보 조회 및 수정 페이지로 이동.
▶ 백엔드:
- /students GET 요청으로 전체 학생 목록 데이터를 가져옴.
- StudentDto 를 통해 엔티티 데이터를 DTO 로 변환하여 응답.
3. 학생 수정 (Update)
▶ UI:
- 학생 정보를 수정할 수 있는 화면(edit.mustache) 구현.
-기존 데이터를 불러와 편집 가능하며, 수정 완료 후 저장 버튼 클릭 시 반영.
▶ 백엔드:
- /api/students/edit/{id} POST 요청으로 특정 학생의 데이터를 수정.
- patchCheck() 메서드를 통해 변경된 데이터만 업데이트.
4. 학생 삭제 (Delete)</td></tr>
</table>

▶ UI:

- 학생 목록에서 특정 학생의 ID 를 클릭하면 삭제 확인 창 표시.
- 삭제 후 목록이 자동으로 갱신되도록 구현.

▶ 백엔드:

- /api/students/{id} DELETE 요청을 처리해 해당 학생 데이터를 삭제.
- 삭제 실패 시 오류 메시지를 반환하도록 설계.

※ 성과

성적 합계 및 평균 계산 자동화

- 학생 데이터를 DTO(StudentDto)로 변환하며 성적 합계와 평균을 자동으로 계산.
- 이를 통해 프론트엔드와 백엔드 간 데이터의 일관성을 유지.

RESTful API 설계

- CRUD 기능을 RESTful 하게 설계하여 확장성과 유지보수성을 확보.
- 각 요청의 상태 코드와 메시지를 명확히 정의하여 오류 발생 시 사용자 피드백을 제공.

동적 데이터 처리

- JavaScript 와 AJAX 를 활용해 학생 목록 삭제 및 갱신 시 페이지 새로고침 없이 동작하도록 구현.
- 성적 합계 및 평균 처리를 위한 동적 테이블 업데이트 기능 개발.

유지보수성을 고려한 설계

- 서비스 계층(StudentService)을 통해 데이터 처리 로직을 분리하여 재사용성을 높임.
- patchCheck 메서드를 활용해 불필요한 데이터 업데이트를 최소화.

이 프로젝트는 Spring Boot 를 활용한 웹 애플리케이션 개발 능력을 심화할 수 있는 좋은 경험이었습니다.
특히, RESTful API 설계, DTO 와 엔티티 변환, 데이터 검증 및 예외 처리, 그리고 사용자 경험을 향상시키는 동적 UI 설계에 대한 자신감을 얻게 되었습니다.
향후 프로젝트에서도 유사한 기술을 활용해 확장성과 유지보수성을 고려한 시스템을 구축할 수 있을 것입니다.