

[로컬 환경 실행 방법]

1. 압축 해제: 제출된 Team16-Phase4.zip 파일의 압축을 해제합니다.
2. Eclipse 실행 및 가져오기: Eclipse IDE를 실행합니다.
 - File → Import... 를 클릭합니다.
 - Maven 폴더에서 Existing Maven Projects를 선택하고 Next를 클릭합니다.
 - Root Directory에 압축을 해제한 Team16-Phase4 폴더 경로를 지정합니다.
 - pom.xml 파일이 인식되었는지 확인하고 Finish를 클릭하여 프로젝트를 가져옵니다.
3. Maven 업데이트: 가져온 프로젝트를 마우스 오른쪽 버튼으로 클릭합니다.
 - Maven → Update Project... 를 선택합니다.
 - 팝업창에서 해당 프로젝트가 선택되었는지 확인하고, OK를 클릭하여 필요한 라이브러리(Dependencies)를 다운로드하고 설정을 업데이트합니다.
4. 프로젝트 클린: Eclipse 상단 메뉴에서 Project→ Clean... 을 클릭합니다.
 - 팝업창에서 Team16-Phase4 프로젝트를 선택합니다.
 - 옵션 중 Start a build immediately에 체크된 상태를 확인하고 OK를 클릭하여 프로젝트를 정리 및 재빌드합니다.
5. Tomcat 런타임 지정:
 - Windows: Window → Preferences 메뉴를 선택합니다.
 - macOS: Eclipse → Settings...(또는 Preferences...) 메뉴를 선택합니다.
 - 이후, 설정 창에서 Server → Runtime Environments로 이동하여 Add를 클릭합니다.
 - Apache → Apache Tomcat v10.1 Server를 선택하고 Next를 클릭합니다.
 - Tomcat Installation Directory에 다운로드한 apache-tomcat-10.1.x 폴더의 경로를 지정합니다.
 - JRE (Java Runtime Environment)는 OpenJDK 11 또는 그 이상 버전을 선택합니다. Finish를 클릭합니다.
 - Apply and Close를 클릭합니다.
6. 서버 생성: Servers 뷰에서 마우스 오른쪽 버튼 클릭 → New → Server를 클릭합니다.
 - Server Type을 Apache → Tomcat v10.1 Server로 선택하고, Server name을 지정합니다. Next를 클릭합니다.
 - 프로젝트 추가: 왼쪽의 Available 목록에서 Team16-Phase4 프로젝트를 선택하여 오른쪽 Configured 목록으로 Add 합니다. Finish를 클릭합니다.
7. 서버 설정 변경: 생성된 서버 인스턴스를 더블 클릭하여 Overview 화면으로 진입합니다.
 - Ports: HTTP/1.1 포트가 8080으로 설정되어 있는지 확인합니다. (필요시 변경)
 - Modules: 화면 하단의 Modules 탭에서 Team16-Phase4를 선택하고 Edit를 클릭합니다.
 - Path를 애플리케이션의 루트 경로인 / 로 설정하고 OK를 클릭합니다.
 - 변경 사항 저장: File → Save(또는 Ctrl+S/Cmd+S)를 눌러 서버 설정 변경 내용을 저장합니다.
8. 실행: Servers 뷰에서 생성한 서버를 마우스 오른쪽 버튼으로 클릭한 후 Start를 클릭하여 애플리케이션을 실행합니다.
9. 실행 확인: 서버가 정상적으로 시작되면 웹 브라우저를 열어 <http://localhost:8080/>으로 접속하여 애플리케이션의 초기 화면을 확인합니다.

[온라인 접속 실행 방법]

1. 웹 브라우저에서 <http://gounchoice.iptime.org/>로 접속합니다.

[사용 방법]

1. 사이트 접속(<http://localhost:8080/> 또는 <http://gounchoice.iptime.org/>)
2. 우측 상단에 회원가입, 로그인을 한다. 로그인을 하지 않아도 상품에 대한 정보는 확인 가능하다.
 - 2-1. 회원가입 시 이메일은 중복되면 안 된다.
 - 2-2. 회원가입 시 비밀번호는 10자리 이상이어야 한다.
3. 메인페이지 상단에서 카테고리 선택, 검색어 입력, 필터 적용 등을 하여 원하는 상품을 검색할 수 있다.
 - 3-1. 주문 이력이 있는 사용자는 구매 패턴이 유사한 다른 사용자를 기반으로 한 추천 상품을 확인할 수 있다.
4. 원하는 상품을 클릭하여 수량을 선택해 장바구니에 담거나 바로 결제할 수 있다.
 - 4-1. 장바구니에 담기를 클릭하면 장바구니 페이지로 넘어갈 것인지 묻고 확인을 누르면 넘어간다.
5. 장바구니에서 구매할 상품들을 선택해 주문하기 버튼을 눌러 결제 페이지로 넘어간다.

6. 결제 페이지에서 배송지, 결제 정보를 입력하고 결제를 누를 시 주문 내역으로 넘어가 정보를 확인할 수 있다.
7. 주문 내역에서 보고 싶은 주문 내역의 "상세 보기"를 클릭 시 주문에 대한 상세 정보를 확인할 수 있다.
8. 배송 완료된 주문들 중 도착일 기준 7일이 지난 주문에 대해서 상품에 "리뷰 작성" 버튼을 클릭해 리뷰를 작성할 수 있고, 수정/삭제 가능하다.
 - 8-1. 리뷰 작성 관련 기능을 확인하기 위해서는 배송 완료 후 7일이 지나야 하기 때문에 테스트 아이디로 로그인하면 기능을 확인할 수 있다.
이메일: jackellen82373@outlook.com, 비밀번호: 190dc0f9b1
9. 어느 페이지에서든 좌측 상단에 로고를 클릭할 시 메인 페이지로 돌아올 수 있고, 우측 상단에 있는 버튼(마이페이지, 주문 목록, 장바구니, 로그인, 회원가입, 흄 버튼 등)을 통해 그 페이지로 넘어갈 수 있다.

[기능 설명]

1. 회원 및 사용자 관리

- 회원 가입: 이메일, 비밀번호, 이름, 연락처, 기본 배송 주소 정보를 입력하여 회원으로 가입할 수 있다.
- 로그인/로그아웃: 등록된 이메일과 비밀번호를 통해 서비스에 접속할 수 있다.
- 마이페이지: 사용자의 개인 정보(이메일, 비밀번호 등)를 조회하고 수정할 수 있는 기능을 제공한다.

2. 상품 탐색 및 검색

- 상품 검색: 상품 이름, 가격 범위, 평점 등 다양한 조건을 적용하여 원하는 상품을 검색할 수 있다.
- 상품 정렬: 상품 목록을 가격, 평점, 리뷰 수 등 주요 지표를 기준으로 오름차순 또는 내림차순으로 정렬할 수 있다. (단, 리뷰 수는 내림차순만 제공)
- 상품 추천: 사용자의 과거 구매 기록 분석을 통한 구매 패턴 유사 기반 맞춤 상품 추천 기능을 제공한다.
- 상품 상세 정보: 선택한 상품의 이미지, 주요 정보, 가격, 리뷰, 종합 평점 등 상품에 대한 모든 정보를 상세하게 조회할 수 있다.

3. 주문 및 결제 관리

- 장바구니 담기: 구매하고자 하는 상품과 수량을 선택하여 장바구니에 저장할 수 있다.
- 장바구니 관리: 장바구니에 담긴 상품들을 한눈에 확인하고, 각 상품의 수량을 자유롭게 조절하거나 목록에서 제거할 수 있다. 주문을 원하는 상품들을 선택해 결제 단계로 넘어갈 수 있다.
- 상품 구매 (바로 구매): 장바구니 단계를 거치지 않고, 상품의 수량을 선택하여 즉시 주문 및 결제 단계로 넘어갈 수 있다.
- 결제: 배송지(기본 배송지 = 사용자 주소), 최종 결제 금액 확인 후, 신용카드 정보 등을 입력하여 결제를 완료한다.
- 주문 목록 조회: 사용자가 결제한 모든 주문 내역을 조회. 주문 날짜, 최종 결제 금액, 현재 배송 상태 및 배송 예정일/도착일 정보를 제공. 배송이 완료되지 않은 주문에 대해 주문 취소도 가능하다.
- 주문 상세 목록: 특정 주문 건에 포함된 상품들의 상세 정보 및 수량을 개별적으로 확인 가능하다.

4. 리뷰 및 사용자 피드백

- 리뷰 작성: 상품에 대한 만족도를 5개의 개별 평가 항목에 대해 별점(0.5점 ~ 5.0점)으로 부여할 수 있다.
- 리뷰 작성 기간: 상품의 도착일 기준으로 7일 후부터 리뷰 작성이 가능하다.
- 리뷰 수정/삭제: 작성된 리뷰는 언제든지 내용을 수정하거나 삭제할 수 있다.

[제작 환경]

운영 체제 (OS): Windows 10 / macOS

개발 언어 (Language): Java 11

웹 기술 (Web): JSP (JavaServer Pages), Servlet, HTML5, CSS3, JavaScript (ES6)

데이터베이스 (DB): Oracle Database (19c)

웹 서버 (WAS): Apache Tomcat 10.1.x

API/규격: Jakarta Servlet 6.0

개발 도구 (IDE): Eclipse IDE for Enterprise Java and Web Developers

버전 관리 (VCS): Git, GitHub

1) 도착일 다음 날부터 7일이 경과한 시점

디자인: Figma

협업 도구: Google Sheets

[YouTube Link]

<https://www.youtube.com/watch?v=DNc5siJq5FU>

[변경사항]

OVERALL_RATING 뷰 추가.

- 모든 상품에 대한 평균 별점과 총 리뷰 수를 집계하여 조회.

UPDATE_DELIVERY_STATUS_PROC 프로시저 추가.

- 주문 날짜(order_date)를 기준으로 일정 시간이 경과하면 배송 상태(delivery_status)를 자동 변경

- 매일 자정(0시 0분 0초)에 한 번씩 자동으로 실행하도록 예약.