

종합 전자상거래 시스템

 생성 일	@2023년 3월 16일 → 2023년 6월 30일
 태그	<div><div>AWS</div><div>Amazon Elastic Beanstalk</div><div>Amazon S3</div><div>Apache Tomcat 8.5</div><div>Docker</div><div>Git</div><div>JAVA11</div><div>JavaScript</div><div>MYSQL</div><div>MariaDB</div><div>Maven</div><div>Mybatis</div><div>Spring MVC</div><div>Spring Security</div><div>Vue.js</div><div>jQuery</div></div>
 한 줄 설명	종합판매 동적 웹 사이트
 개발 인원(역 할)	1명 (풀스택)
 Github	https://github.com/rlarudqls/beans_mall.ver1
 URL	http://3.39.110.53:8080/

Project 소개



Make your fashion more beautiful

🤔 프로젝트 개요

😊 기능설계 및 화면구현

😬 데이터베이스 구축

😍 사용자 피드백 및 최종 배포

100 최종 완성본 보기

👤 역할

▼ 요구 사항 분석 및 설계:

- 고객 요구 사항을 분석하고 쇼핑몰 애플리케이션의 기능 및 시스템 요구 사항을 도출하였습니다.
- 데이터베이스 스키마 설계와 애플리케이션 아키텍처 설계를 수행하였습니다.

▼ 백엔드 개발:

- Java 11과 Spring MVC, Spring Security를 사용하여 백엔드 로직을 개발하였습니다.
- 상품 관리, 주문 처리, 사용자 인증과 권한 관리, 결제 처리 등 다양한 기능을 개발하였습니다.

- MyBatis를 활용하여 데이터베이스와의 상호작용을 구현하였습니다.

▼ 프론트엔드 개발:

- HTML, CSS, JavaScript를 사용하여 쇼핑몰의 사용자 인터페이스를 개발하였습니다.
- Vue.js와 jQuery를 활용하여 동적인 기능과 상품 목록, 장바구니, 주문 절차 등을 구현하였습니다.

▼ 데이터베이스 관리:

- MySQL 또는 MariaDB를 사용하여 쇼핑몰의 데이터베이스를 설계하고 관리하였습니다.
- 데이터베이스 테이블 설계, 쿼리 작성 및 최적화를 수행하였습니다.

▼ 테스트 및 디버깅:

- 백엔드와 프론트엔드의 유닛 테스트를 작성하고 실행하여 코드의 품질을 검증하였습니다.
- 버그를 발견하고 디버깅하여 안정성과 성능을 향상시켰습니다.

▼ 배포 및 운영:

- Maven을 사용하여 WAR 파일을 빌드하고, Apache Tomcat 8.5에 애플리케이션을 배포하였습니다.
- AWS EC2를 이용하여 애플리케이션을 배포하고 운영하였습니다.



새롭게 추가된 기능 및 작업

Docker 말기

Amazon Elastic Beanstalk에 배포

Amazon S3

- 컨트롤러 테스트 코드 작성
- NaverShopSearch 검색 기능 추가
- 비밀번호 해싱 추가 & 비밀번호 필드의 입력 타입을 "text"에서 "password"로 변경
- Spring Security XML 대신 자바 빈으로 구성

AppRunner로 배포[진행중]



AWS HTTP HTTPS 리다이렉트 공격 받은 기록 2023.10.27

▼ 빈스몰.ver2으로 리팩토링 진행중입니다.

빈스몰.ver2 제작시 변경사항

회원가입 페이지와 보안 관련

회원가입 페이지의 비밀번호 입력란의 input 타입을 password로 변경하여 별표(**)로 가려서 보이게끔 변경해야 합니다.

비밀번호를 안전하게 저장하기 위해 해시 함수를 사용하여 암호화해야 합니다.

코딩 스타일 및 구조

println 대신 로그 라이브러리를 사용하여 로깅하도록 변경해야 합니다.

autowire 대신 생성자 주입을 사용하여 의존성 주입을 해야 합니다.

DTO와 VO의 용도와 의미를 정확히 이해하고, 적절한 곳에 사용해야 합니다.

프로젝트 구조와 패키지 설계

MVC 패턴을 기반으로 프로젝트의 패키지 구조를 구성해야 합니다. 각 기능별로 패키지를 분리하고, 모듈화하여 개발해야 합니다.

도메인 별로 패키지를 나누어 관리하거나, 기능별로 패키지를 구분하는 방식 중 하나를 선택하여 구조를 설계해야 합니다.

프로젝트 성능 최적화

API 호출이 많은 경우, 부하를 모니터링하고 성능 최적화를 진행해야 합니다. 데이터베이스 쿼리 최적화, 캐싱, 비동기 처리 등을 고려해야 합니다.

성능 최적화에 대한 경험과 결과를 문서로 정리하고, 향후 프로젝트에 적용할 수 있도록 학습해야 합니다.

커뮤니케이션과 지식 공유

팀 내에서 지식을 나누고 의견을 교환하는 문화를 유지해야 합니다. 주기적인 미팅이나 코드 리뷰를 통해 팀원들 간의 소통을 촉진해야 합니다.

기술 블로그나 유튜브 채널을 활용하여 기술적인 경험과 지식을 공유하여 팀원들의 성장을 도모해야 합니다.

현업자분들의 피드백을 귀담아 더욱 나은 서비스의 품질을 제공하기 위해서 노력하겠습니다.



[링크] 빈스몰.ver2 리팩토링 Github

▼ 리팩토링 주요 내용

1. Autowire 대신 생성자 주입 활용
2. 클래스 이름은 대문자로 시작, 패키지 이름은 소문자로 작성.
3. Getter 및 Setter 대신 생성자 주입 사용
4. 도메인별로 패키지를 생성하고, 각 패키지 내에서 VO, Service, Controller 등을 구성합니다.
5. Branch 버전 관리 branch versioning