**1조 1차 프로젝트 시나리오**

**조원: 황기창, 왕진우, 임희진, 박지영**

* **개요**

저희는 쿠버네티스 상에서 3티어 아키텍처를 설계하여 카페의 DB에 등록된 회원만 온라인 웹 페이지에서 메뉴를 구입할 수 있는 시스템을 구현하였습니다. 일반 카페에서 운영하는 것처럼 회원 여부에 따른 할인 적용은 개발에 큰 부담이 되어 제외하고, 단순한 시나리오로 구성하였습니다.

* **시스템 구성**

저희가 구축한 시스템은 다음과 같은 흐름을 따릅니다:

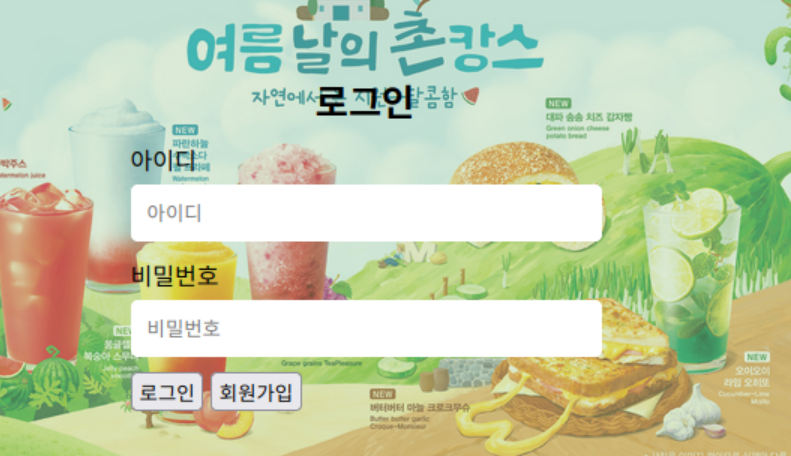
1. **접속 구조**: Tomcat 서버에 부하가 발생하지 않도록 사용자는 Nginx 서버를 반드시 거치도록 구성하였습니다.
2. **사용자 접근**: 사용자는 웹 페이지에 접근하기 위해 먼저 Nginx 서버를 통해 Tomcat 서버로 연결됩니다. 이는 Tomcat 서버의 부하를 분산하기 위함입니다.
3. **파일 저장소**: 저희 조는 웹 서비스와 관련된 html, jsp 파일들을 모두 Tomcat 서버에 저장하였습니다.



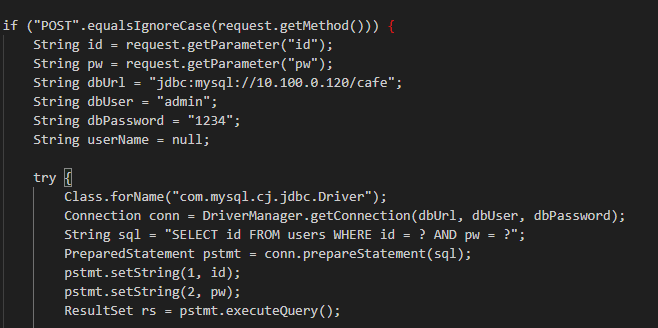
1. **메인 페이지 존재**: 사용자는 Nginx 서버를 통과하여 Tomcat 서버에 저장된 main.jsp 파일을 보게 됩니다. 여기서 로그인을 하거나 회원 가입 페이지로 이동할 수 있습니다.

* **시나리오 상세**

1. **로그인 시나리오**
2. 사용자가 main.jsp에서 바로 로그인을 시도합니다.



1. main.jsp 파일에서 Tomcat 서버에 설치된 Mysql J Connector를 통해 연결된 카페 DB의 users 테이블을 참조하여 회원 여부를 확인합니다.

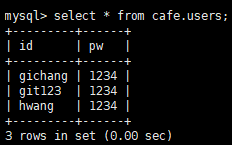


1. **회원 가입 시나리오**

* 회원 가입을 하지 않은 사용자는 main.jsp의 회원 가입 버튼을 클릭하여 signup.jsp 페이지로 이동합니다.

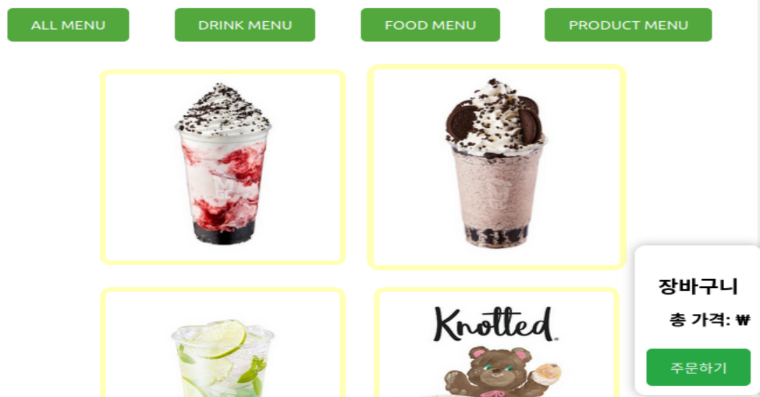


* 여기서 아이디와 비밀번호를 입력하면, 해당 정보가 DB에 저장됩니다.



1. **주문 시나리오**

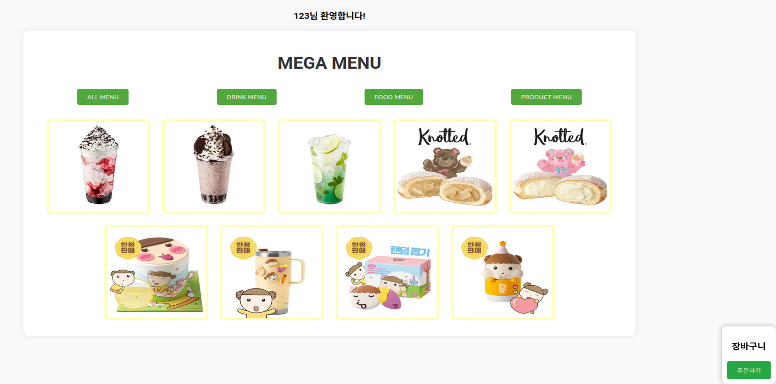
* 이후 사용자는 order.jsp 페이지에서 메뉴를 장바구니에 담을 수 있습니다.



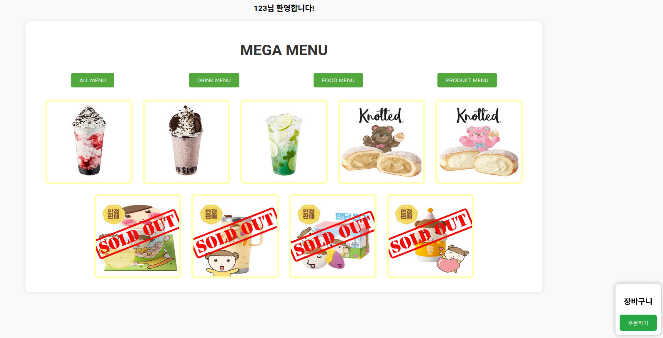
1. **롤링 업데이트를 활용한 재고가 매진되었을 때의 시나리오**

* 특정 메뉴가 매진되었을 때, “SOLD OUT”이라는 이미지로 변경하여 client에게 혼선을 주지 않게끔 하였습니다.

**<Before>**

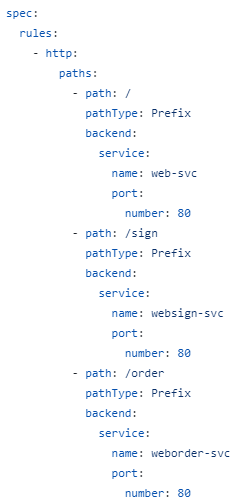
****

**<After>**

****

1. **Ingress 컨트롤러에 시나리오를 반영한 설정 내용**

* 저희는 nginx 디플로이먼트 파일을 구성할 때 main, sign, order 부분으로 구분하였습니다.
* 또한, 로그인에 성공하지 않은 client가 주문하는 경우를 방지하기 위해 main 부분은 (web-svc) 상위 디렉토리인 ‘/’로 설정하였고, 회원 가입을 담당하는 sign과 order 부분은 각각 ‘/’ 밑에 sign 디렉토리와 order 디렉토리로 경로를 설정하였습니다.



* **결론**

k8s 환경에서 운영되는 3-tier 웹 서비스 구축에 성공했고, 개발 내용이 보다 더 구체화되었을 때 일반 카페에서 시행하는 이벤트들도 구현이 가능했습니다. 이상으로 저희 1조의 1차 프로젝트 시나리오에 대한 발표를 마치겠습니다. 감사합니다!