JENSEN yrkeshögskola
TRÄNING FÖR VERKLIGHETEN

Kursnamn: *Systematisk programutveckling med Java*
Klass: *CAW22S*
Termin: *HT22/VT23*

BACKEND FOR THE SALT MERCH SITE

INLEDNING

Bakgrundsbeskrivning , frågeställning, avgränsning och mål	Bakgrund: In this course you will learn how to create a backend using Spring Boot, with its own database, continuously tested, and all of that running in Docker containers. Frågeställning: How can we create a backend for an existing frontend, so that we get a complete website for http://salt-merch.wickitly.com? Mål: You will learn what a backend consists of, how it works to store and serve data, and how to create and set it up so that a website can use it. You will also get more comfortable with using git for version control and make your projects available on GitHub. Avgränsning: You don't need to have an entire CI/CD pipeline (automatic continuous integration and deployment with daily builds), or to design the database and write the specific SQL queries for each operation, or to dockerize the backend; only a functioning backend that will respond to the requests made by the Salt Merch site.
Varför ska ni utföra detta arbete?	Syftet: This assignment will help you get more familiarized with the following:. ● how backends are created in the real world and what purpose they serve ● how to create databases and connect them to a controller ● how to make sure the application is properly tested ● how you can use Docker to compartmentalize the frontend and the database ● the entire build process
Vad ska ni leverera?	Följande ska levereras: 1. a GitHub repository containing the project 2. a simple UML diagram representing your model classes 3. a Spring Boot application running on Tomcat 4. a REST API for responding to all requests the frontend can make

	5. a database for storing all the merch items and their properties 6. proper handling of bad requests or server errors 7. a suite of unit tests for some of your methods and some integration tests for testing the endpoints

Er projektuppgift

Vad ska ni göra?	1. create a git repository and link it up with a GitHub repository 2. create a Spring Boot project with the right dependencies for running the app in a Tomcat server, connecting databases to it, and running tests 3. create the database tables and insert the data (using the scripts and data included in this assignment) 4. make a connection to the database 5. create the endpoints according to the specifications included in this assignment 6. write tests for the methods that your endpoints will use for reading or writing to the database 7. write the actual methods and make sure the tests pass Specification for the required endpoints and the database scripts can be found in other files attached to this assignment.
Hur ska ni lösa uppgiften?	You will do this assignment individually. You will be given the full frontend project, which you can run inside Docker in order to test your REST API. A version of the frontend is live at http://salt-merch.wickitly.com. The frontend project will contain clear instructions on how to start the frontend server. You will also be given the full database server, with tables and rows already created. In addition, you can also use Postman (which we will also use in classes) to test your endpoints. For your backend, start by downloading the skeleton Spring Boot project from the provided GitHub repository, then add the dependencies for Spring Web, JDBC, PostgreSQL driver, and the Spring Test starter pack. That gives you all the Java APIs necessary to set up a web server, expose endpoints, and write methods for querying the database and test functions. Before writing each method, you should start by writing a test method following the principles of test-driven development. First write down what the method's exact goal is, create the test by making it expect some result, and then implement the method itself, thus making the test pass. If you ever get stuck, you should refer to the practice assignments that we have had throughout this course, since they contain almost all steps that need to be taken for this project assignment.

	As always, Stackoverflow is the main place for specific programming questions.

Inlämning och redovisning

Inlämning	Inlämning sker via Learn Point ***22 januari senast kl. 23:59***
Redovisning	Muntlig redovisning sker den **24 januari kl. 13:00**, 5-10 min/individ

Bedömning och återkoppling

Bedömning sker mot följande betygskriterier:	Betygskriterierna för Godkänd respektive Väl godkänd är: **Godkänd** ● a GitHub repository for the project ● a REST API exposing the crucial endpoints ● Java classes that represent the merch and DTOs ● a working connection to the database ● at least some unit tests **Väl godkänd** ● clean code, including proper indentation and letter case ● separation of concerns/sensible grouping of classes into controllers, models, etc. ● UML diagram for the model classes ● all endpoints implemented and returning sensible data ● at least one integration test ● proper error handling
Återkoppling	I och med redovisningstillfället ges muntlig återkoppling till grupperna. Grupperna får skriftlig återkoppling via lärplattformen LearnPoint **senast den datum 03.02.2022**.