Use of AI

OrderService

Ai was used to help write the post/get functions, the professor showed how to write them in python, but to get them working in java, ai and online resources were used, to guide how to write it in java.

Used ai to learn how to send forward requests.

Helper functions to parse JSON found online and modified to fit needs.

Parse

Used some ai to guide in parsing and forming JSON. Received some guidance on sending requests in Python as well.

UserService

Used Generative AI to create the initial structure and endpoints. Also helped in writing the JSON parser because we thought it’s better to not import the Jar files to do it (more error prone). The code was then modified to some extent to satisfy the requirements such as handling JSON data and fitting with API, as well as internal data storage.

ProductService

Generative AI helped in starting the initial service. Like UserService JSON parser is written by AI. SHA-256 Hasher and its helper are by both AI and Stack Overflow posts, we adjusted it to work locally. We then adapted to include additional functionalities like API validation, data structure according to API and data storing.

Shortcuts Taken

Inter-service Communication Service: This service was not written as A1 was able to be completed without it. We have decided to leave it to A2 so that way the workload can be lighter.

In parse the method to deal with missing fields was to fill in missing fields with “placeholder” string but this won't be possible if the body parameters are changed often, since there are some hard coded values.

Lot of data structures were used for parsing info and sending JSON, which could slow down the product if enough commands are used.

Lack of database, we saved the data locally without using a database, this works for a small amount of data, but without a database larger amount will cause issues. This is to be added in A2, right now can get away with it but definitely a temporary solution.

Lack of error checking, currently the error checking is rather weak and as before, all data are stored in memory in A1 and causes some security vulnerabilities.

Code Structure

In terms of which file should be preserved in A2, first definitely all the endpoints for requests. Also, helper function such as JSON parser and password hasher should be preserved. Sample usage of User creation should also be preserved mostly.

Shell scripts, such as compiling and starting services, should be mainly unchanged unless further functionalities are to be added.

Command Parser, this Python file should be mostly unchanged, as it reads commands and parses them to Order Service, should follow the project structure and unless more functionalities are added or changed, this should preserve.