

Carmen Mosquera

CS 499 – 11429-M01

10 /16 /2024

Narrative : Artifact Three

Artifact Description

The Grocery Tracker is a web-based application designed to monitor and track the frequency of product purchases from a store, while also providing insights on sales trends.

The original version of this artifact was created in C++ and focused on reading and analyzing data from text files generated by a store. This enhanced version, built with Java Spring Boot for the backend, MySQL for the database, and Angular for the frontend, introduces a robust user interface, product management features, and visual representations of sales data, such as product frequency charts. The artifact was initiated as part of a project for the CS210 course in 2023.

Inclusion Justification

I decided to include this artifact in my ePortfolio because it highlights my strengths in database management, backend development, and data visualization, all important components of software engineering. The application showcases my ability to build scalable backend systems using Java Spring Boot, implement advanced SQL queries, and integrate frontend frameworks like Angular for dynamic user interfaces. The data

visualization aspect, which includes charts to display sales frequency, emphasizes my ability to work with complex data sets and present them in a user-friendly way. Also, transitioning the project from C++ to a full-stack web application with enhanced functionality demonstrates my growth in software development and my ability to adapt to different technologies.

Performed Enhancements List

1. Ported from C++ to Java.
2. Implemented a backend using Java and Spring Boot.
3. Integrated MySQL database.
4. Separated backend code into modules.
5. Implemented Data transfer objects (DTOs).
6. Designed and implemented RESTful APIs.
7. Ensured that API returns JSON objects.
8. Created a webSecurityConfig class to handle CORS issues.
9. Integrated Spring Security to manage authentication and protect API endpoints.
10. Built a responsive and user-friendly fronted interface using Angular.
11. Implemented the ability to sort and filter the product list by different criteria such as product name, category, location, or purchase frequency.
12. Added the ability to upload and display images for each product.
13. Created a dashboard to analyze sales data, such as purchase frequency, sales performance by region, and overall sales trends.

14. Added charts, graphs, and various visual elements to display sales trends.
15. Developed a detailed README file that outlines how to install and run the application.

Course Outcomes.

The development and enhancement of the grocery tracker helped me reach the following course outcomes:

Course Outcome 1: I Employed strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science by completing the following enhancements:

- I created a dashboard to facilitate collaboration and to provide decision-making support to various users, such as store managers, sales teams, and business analysts to easily access and analyze sales data, including top-selling products, sales by category, and store locations with the highest purchases.
- I designed the database schema and user interface to be user-friendly and accessible, ensuring that users can navigate the system and make data-driven decisions.
- The reports generated by the system help stakeholders track product performance and identify trends that impact purchasing strategies. This collaborative approach to sharing data across departments supports organizational goals and decision-making processes.

Course Outcome 2: I designed, developed, and delivered professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts by completing the following enhancements:

- I designed a “custom Angular form” to create a clear and effective ways for users to interact with the underlying data, this form allows users to add and manage grocery items, ensuring a smooth user experience. The form includes input validation, preventing users from entering incorrect data, and it connects seamlessly to the backend for product management.
- I provided users with a “dashboard” that visualizes key sales data, including top-selling products, sales by category, and store locations with the most purchases.
- I used charts and graphs to present this data, making it easy for users to understand sales trends and performance.
- Next to each chart, I added text boxes with chart summary and detailed explanation of the chart data
- The dashboard automatically updates based on the data returned from SQL queries, demonstrating how technical information can be visually communicated to support decision-making.
- I provided clear instructions and documentation in the README file to guide users and developers through installing and using the system.

Course Outcome 3: I designed and evaluated computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices by completing the following enhancements:

- I enhanced the original C++ artifact that reads data from a text file by redesigning it as a full-stack web application using Java Spring Boot and MySQL to track daily product purchases. This shift introduced the ability to handle larger datasets efficiently, with proper database management through MySQL and optimized SQL queries to ensure high performance when retrieving product sales data.
- I designed the Category and Location classes to allow users to track product sales by category (e.g., dairy, produce, baby products) and store locations, introducing an added layer of detail and usability. This solution addressed the need for real-time tracking of sales in multiple dimensions, optimizing how sales are displayed based on product and location.
- I implemented algorithms that aggregate and filter daily sales data to provide comprehensive sales reports, including top-selling products. In doing so, I balanced the trade-offs between query efficiency and data accuracy to ensure users received accurate insights without performance degradation.

- I designed a relational database that includes many-to-one relationships between products and categories, and products and store locations, which ensures efficient organization of the data.
- I used advanced SQL queries to retrieve and manipulate data for reporting. For example, I used queries such as “SUM()” to calculate the total sales of each product, “COUNT()” to determine the number of times each product was purchased, “GROUP BY()” to group the sales by product name, and “ORDER BY” to sort the sales in descending order. These queries formed the basis of the Top Selling Products chart on the dashboard, where I Used “JOIN()” to join the sales and products tables to display the product names alongside the sales figures. This merging of tables allowed me to present meaningful insights, helping users quickly identify high-performing products

Course Outcome 4: I used well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals by completing the following enhancements:

- I modularized the backend architecture by organizing it into small, reusable modules, such as the “ProductService”, “SalesService”, and DTOs for the frontend API. This approach followed industry best practices for maintainable and scalable code, allowing future developers or users to extend the application without risking breaking the codebase.

- The artifact frontend was built using Angular, which helped me create a dynamic and interactive user interface for users to track product sales visually. This choice demonstrated my ability to use modern front-end frameworks that meet industry standards for responsive and user-friendly interfaces.
- I used HTML and CSS, along with Angular material to create a sharedUI component to centralize all UI elements, such as buttons, tables, forms, and labels. This design allows future developers to modify the UI without disrupting the core functionality of the system
- I integrated modern web technologies and visualization tools such as the JavaScript library “Chart.js” with the Angular wrapper “ng2-charts” to create bar and pie type charts to visualize product sales data.

Course Outcome 5: I developed a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources by completing the following enhancements:

- I implemented Spring Security to enforce user authentication and protect access to sensitive areas of the system. Users must authenticate before being able to view sales data or add new products to the tracker, ensuring only authorized personnel can modify or access the data.

- To further enhance security, I created a `WebSecurityConfig` class that handles CORS (Cross-Origin Resource Sharing) issues, ensuring that the frontend can securely communicate with the backend while mitigating risks such as cross-site scripting (XSS). This configuration also ensures that unauthorized requests from untrusted origins are blocked.
- I employed parameterized SQL queries throughout the artifact to protect against SQL injection attacks. By using parameterized queries, user inputs are safely handled and passed to the database, preventing malicious actors from injecting harmful SQL code into the system.