Progress Report: Sprint 2

Preston Knepper, Dalton Rogers, and Kevin McCall

CS496

Dr. Holliday

3/7/2025

Goals for the most recent sprint

- Implement sharding algorithm on the backend
    - refactor create operations
    - refactor read operations
    - refactor update operations
- Implement neo4j graph on frontend
- Add functionality to
    - The search bar
    - The home button

Activity Log

Name	Date	Time Spent	Description of Activities	Result of Activities
Kevin	2/21	3 hours	Investigated existing sharding solutions for Postgres, MongoDB, Neo4J and Redis	Gained a sense of direction of implementation and learned that a custom implementation is a valid solutions and would be better suited for our use case.
Kevin and Preston	2/26	2 hours	Met at the MTC and set up the Redis database. Laid out plans and code architecture for routing shards using Redis.	Created the Redis database and attached it to our project.
Kevin	3/3	2 hours	Started filling out the code skeleton established on 2/26.	Rewrote many asynchronous methods to return promises correctly. Functions query which database to connect to before performing queries.
Kevin	3/3	1 hour 30 mins	Rewrote the create.ts file to shard the postgres database.	The postgres database now considers all databases and routes new data to the least contended database.
Kevin	3/4	30 mins	Used Redis to handle synchronization between databases across networks.	Redis handles race conditions through INCR command.
Kevin	3/5	2 hours	Optimized the database connections to support hundreds of thousands of generated products.	Program no longer crashes due to running out of postgres connections.

Name	Date	Time Spent	Description of Activities	Result of Activities
Dalton	2/23	1 hour	Looked over the changes Kevin made to the frontend to familiarize myself.	Gained an idea of some components that I wanted to implement along with some things that could be changed.
Dalton	2/27	2 hours	Made it so that all the products are listed on the home page and added functionality to the search bar.	Now all the products can be seen and scrolled through, and specific products can be search by either name or product id.
Dalton	2/29	1 hour 30 minutes	I made the list of products scrollable fixed so that the list didnt extend past the right side component. I also added some more formatting to the way the products are formatted.	It is now cleaner on the frontend.
Dalton	3/2	2 hours	I made the home button functional.	When pressed it will reset whatever was searched for and relist all the products again.
Dalton	3/4	3 hours	Made the products clickable, so that when clicked they are also the search result. I also worked on getting the neo4j data from the backend to the frontend. Able to display a neo4j representation for a single node on the front end.	More functionality on the frontend, and able to view Neo4j products.
Dalton	3/5	30 mins	Made it so that when no product is selected, all of the nodes from neo4j are displayed.	Makes neo4j look better.
Dalton	3/6	1 hour	Began setting up routes on the backend to send the postgres data for either a specific product or for all of the data.	The api calls work, but did not get to implement any visual aspects on the frontend.
Preston	2/26	1 hour	created code skeleton for sharding	had a basic design for how sharding with redis would be implemented
Preston	3/5	3 hours	Worked on redis operations in relation to sharding, which was eventually scrapped	Had more ideas on how sharding should be implemented. considered multiple algorithms
Preston	3/6	3 hours	Added the functionality to get items from shards, involving the rewrite of our getters	We can now (theoretically) retrieve items from our sharded databases
Preston	3/6	1 hour	Added caching functionality to redis as a last minute decsision	There is a functional cache which stores retrieved products (and respective recommended products) for 10 minutes.
Preston	3/6	2 hour	Refactored sharding algorithm	Algorithm should work better in the long run. No longer shards Neo4j