Calypso

Iteration 3
Start date: 12th July 2023
End date: 21st July 2023

Process

Continuing from last week, this sprint will be focused on the backend. All members will continue to polish their features.

Changes from previous iteration

1. One change that we are implementing this sprint is decreasing the number of stand-up meetings we have during the sprint. Instead of having one every day we have cut that down to 5 meetings for the whole sprint. We are making this change because we saw that having a sprint meeting every day felt kind of redundant due to the fact that not a lot of progress was being made on a daily basis due to the members not being able to dedicate time every day to the project due to other commitments such as other classes and assignments. To measure the effectiveness of this change, we will monitor how the sprint meetings go and how much more value we get from each sprint meeting. If the meeting consists of less redundant and repetitive updates, then the process change will be considered a success.

Roles & responsibilities

Every team member will be doing both front end and back-end development for their assigned user stories. The different services and persons assigned as follows:

- Willam & Maaneth: Products & Media [Neo4j]
- Austin & Shadman: Transactions [Neo4j]
- Tasif: User Data [Neo4j]
- Cassandra: Messaging [MongoDB]
- Arielle: Users Authentication [MongoDB]

Events
Meetings will be held online on discord, to discuss our next steps and any difficulties. Regular stand-up meetings will occur 5 times during this sprint, all at 8pm via discord. Two code review

meetings will be held where we discuss what we did and explain the code we wrote to others / help other members if needed.

Artifacts

- We will use JIRA to keep track of our progress on stories and which stories need to get done
- Jira will also be used to list any new subtasks that must be completed in order to complete the user story
- The highest priority tasks would be the ones which are the foundation of the app. We prioritize the functionality of our app. We would like to always create the minimum viable project for each sprint.
- Each team member chooses which user stories they are assigned to. If two or more developers want the same task, we will randomize the choice.

Git / GitHub workflow

Since everyone is working on different parts of the project, there has not been any conflicts within our codebase. For this sprint, even though multiple persons were working with products or media, they were working on different user stories, so there wasn't conflict. We will continue to separate the work items in a similar way.

The workflow is as follows:

1. Pull the latest version of main from GitHub to local version and create a branch
2. Name the branch as the user story number or even subtask number, along with a short description if needed. Publish the branch.
3. Work on code. Commit as needed throughout your process, to this branch.
4. If someone pushes and merges new changes while working on this branch, ensure to pull those changes to branch
5. When all code is completed, ensure all commits are made, and create a pull request and request the review of a fellow team member.
6. Reviewers must check out the newly published branch, and review all changes made by peer.
7. After changes / suggestions are made (if any), the reviewer will approve of the pull request. The team member that created the pull request is responsible for merging their new code to main.
8. Notify the team that your changes are merged so that others can pull to their branch

Product

Goals and tasks

Goal:

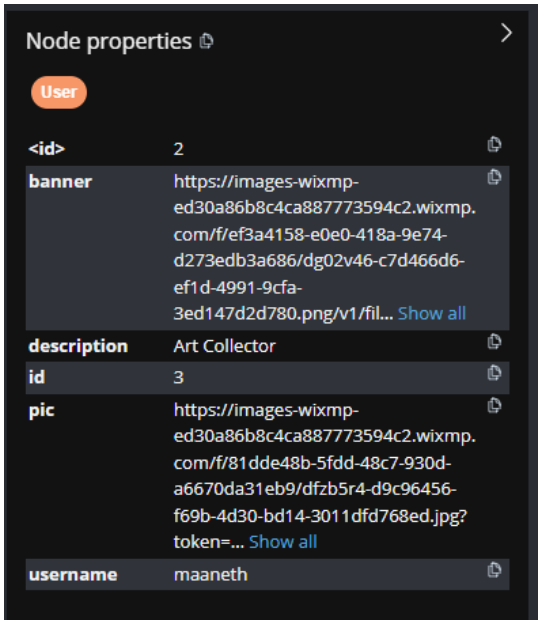
The main goal of this sprint is to ensure connection to databases. We would need to connect to MongoDB and Neo4j. This way, we can work on sending and retrieving data for sprint 3.

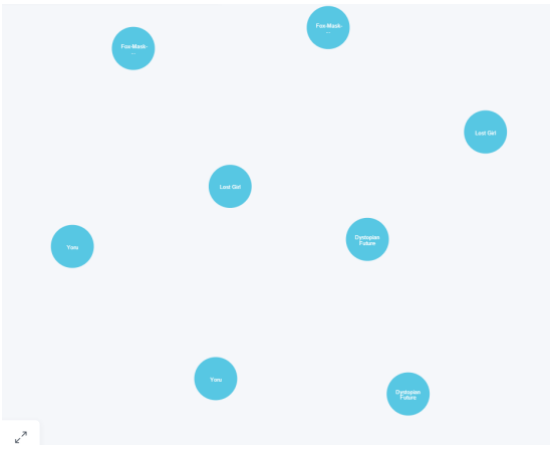
Tasks:

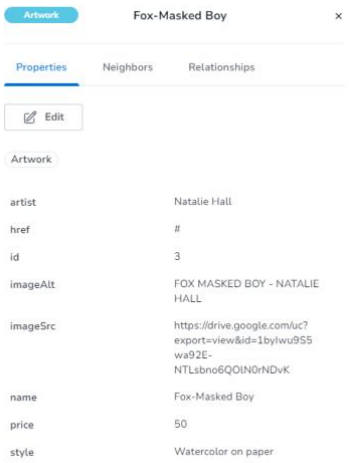
- Hash user passwords
- Be able to log in for users
- Edit user profile page
- Navigation to other users
- Implement messaging functionality
- Add Search/Sort/Filter options for the Product Listings page
- Implement a Product Recommendation feature
- Add bidding feature for products

Artifact #1: The Neo4j and MongoDB Databases

This is what the Databases look like currently. We have 3 databases, 2 for Neo4j and 1 in MongoDB. The Neo4j databases are used to store the user and product information, and the MongoDB is used to store messages and user authentication info. We plan to combine the Neo4j databases since they are very intertwined.







Artifact #2:

Add functionality in most pages

- Implement navigation to other users profile
- Implement Add/Remove product functionality
- Implement Search/Sort/Filter options for the Product Listings page
- Implement a Product Recommendation feature
- Implement bidding functionality, where a bid can be placed on a product, which can be accepted by the creator.
- Implement messaging functionality

Artifact #3:

Connect Stripe API so that transaction are functional

- Right now we just have the front end for the transactions ready
- Stripe API allows for payment processing and authentication using their API