Work summary for Sprint-1

We identified 5 sprint levels to execute the whole project before December 16th.
The first sprint level includes the following functionality implementation:

1. Functional: Registration
2. Functional: Login
3. Functional: Item Listing for Auction
4. Functional: Browsing and Searching
5. Functional: Auction Win Notification

We planned on implementing the database part of the code from the second sprint as the second sprint involves core bidding and auction interaction. The user will be able to interact with live auctions, place bids, and receive real-time updates after the second sprint. So to check if the first sprint is working properly and is meeting all the requirements, we used dummy data in the service implementation component to verify its execution and goals. There was a lot of error in execution of the code along with MySQL without implementing the second sprint.

Summary:

Work Completed

1. Functional: Registration and 2. Functional: Login

Implemented login and registration features using Spring Boot, following the Controller-Service-Repository pattern in a separate folder.
Created the AuctionUser model to manage user information such as username and password.
Developed an in-memory repository (UserRepository) to store and retrieve user data without a database for the current sprint.
The AuthService was responsible for handling user registration and login logic.
Created an AuthController that exposed two endpoints:

POST /auth/register to handle user registration.
POST /auth/login to manage user login.
Implemented basic validation to check if a user already exists and to validate login credentials.
Security Configuration:
Set up basic security with WebSecurityConfigurerAdapter to allow all requests temporarily, enabling easy testing of the login and registration functionality.
Added the necessary Spring Security dependencies to facilitate future authentication and authorization requirements.

3. Functional: Item Listing for Auction

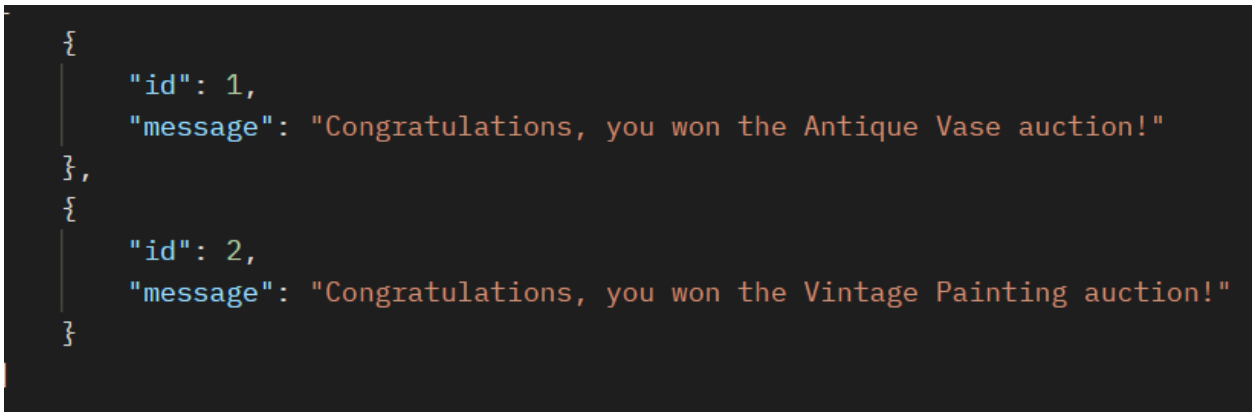
A separate package called com.acs560.AuctionEase.itemlisting is created. It has the model (entity), repository, service, serviceImpl, and controller layer for this respective functionality. The main goal of this functionality is to list items for auctions with detailed descriptions, multiple images, and starting bid prices. For this purpose, the serviceImpl layer has the sample data defined in it, from which the items for auction are listed along with the description, image and bidding rate. We can also retrieve all bidding items, or items specific to the ID.

4. Functional: Browsing and Searching

The `com.acs560.AuctionEase.browseandsearch package implements the browsing and searching functionality. This feature allows users to browse all available auction items or filter them by category (e.g., electronics, furniture) and condition (e.g., new, used). The serviceImpl layer includes sample data representing various items with detailed descriptions and starting bid prices. Users can retrieve all items or search for specific items based on their filters.

5. Functional: Auction Win Notification

The com.acs560.AuctionEase.winningnotification package is used to receive notifications when I win an auction so that I can proceed to payment. This provides the notification by ID number and also all the notifications. It has sample data that displays the message of winning the auction along with the ID number.



Work Not Completed

Database Integration: The planned integration of a MySQL database to store user data was not completed due to several unresolved errors that arose during the implementation process. As a result, user data is stored in memory for now, with the database integration postponed to a future sprint.

Challenges

The primary challenge faced during this sprint was the errors encountered while trying to integrate the database. These errors caused significant delays, as I was unable to resolve them within the sprint timeframe. This resulted in moving the database work to the next sprint.