SD Project Documentation

Project Title: Local Artisan Marketplace

Course Code: COMS3009A - Software Design

Team Members:

Harshil Vallabh: 2656158

Sunay Master: 2677874

Matthew Purkiss: 2697424

Yabsira Gebremichael: 2661262

Kamal Lalloo: 2652159

Tutor:

Tapiwa Mazarura: 2581366

Quick Links

GitHub Repository:
https://github.com/uno791/local-artisan-marketplace
The repository contains the complete source code, a structured folder layout, and a README with setup instructions.

Central Documentation:
All project planning, user stories, sprint records, and diagrams are available here:
Google Docs – Project Workspace

Website Link:
https://calm-meadow-0fbb07c03.6.azurestaticapps.net/

Video Demo Link:
https://drive.google.com/drive/folders/1IsnntK5I2Zu85vB_J0hiwcDLKhBQtiho

Admin Email Account (testing):
Email: h75766237@gmail.com
Password: SuperCoolPassword123#

Evaluation Instructions

To assist with marking and ensure a smooth experience reviewing the project, please follow the steps below:

Test Accounts

To test different user roles, you can use the following options::

- Buyer Test Account:
  Email: You can log in using any Gmail account via the "Sign in with Google" option. All new users start as buyers by default.
- Seller Test Account:
  Email: Use the same Gmail account you signed up with. After registering, submit a seller application from the buyer profile. You can then approve the seller application from the admin account to gain access to the seller dashboard.
- Admin Account (demo):
  Email: h75766237@gmail.com
  Password: SuperCoolPassword123#

  This account is flagged as admin in the database

Note: All accounts use Google OAuth, so simply click "Sign in with Google" and select the relevant email to log in.

Running Locally (Optional):

Clone the GitHub repo:
git clone https://github.com/uno791/local-artisan-marketplace.git

Client:
cd client
npm install
npm run dev

Server:
cd server
npm install
node app.js

Contents

Project Overview

1.1 Project Description

Localish is an online marketplace that helps local artisans connect with buyers in their communities. It's designed to make discovering and supporting original, handcrafted products easy and personal. Buyers get a tailored shopping experience based on their interests and browsing history, and if they're creators themselves, they can apply to become sellers and open their own shopfronts. At its core, Localish is about empowering individuals to share their creativity and giving others a more meaningful way to shop and support local talent.

1.2 Project Objectives

The main objectives of the Localish platform are to:

- Provide a digital space for local artisans to market and sell their products.
- Enable buyers to discover and purchase unique items based on their preferences.
- Facilitate secure transactions through the Yoco payment gateway.
- Support a layered user experience: buyer, seller, and admin roles.
- Ensure moderation and quality control through administrative oversight.
- Deliver insights to sellers for data-driven inventory and sales management.

1.3 Key Features

Buyer Features

- Google account sign-up and login (OAuth integration)
- Personalised "For You" page based on selected interests and search history
- Profile management (profile image, phone number, postal code)
- Order history and details
- Ability to report products (e.g., for offensive content, delivery issues)

Seller Features (after admin approval)

- Seller application and verification process
- Shopfront creation and product management (add/edit products, apply tags)
- Option to offer pickup and/or delivery
- Sales insights (monthly sales, top products, total revenue)
- Inventory status tracking
- Order management and shipping workflow

Admin Features

- Review and approve/reject seller applications
- View platform-wide aggregated sales analytics
- Investigate user-submitted product reports
- Moderate and remove products as needed

1.4 Technology Stack

Frontend: React, TypeScript, Vite
Backend: Node.js, Express
Database: Azure SQL
Authentication: Microsoft Entra ID (Azure AD)
CI/CD: GitHub Actions
Hosting: Azure App Service (API), Azure Static Web Apps (Frontend)
Testing: Jest

Team & Contributions

2.1 Team Members and Roles

The development of *Localish* was a fully collaborative effort undertaken by five team members:

- Harshil Vallabh (2656158)
- Sunay Master (2677874)
- Matthew Purkiss (2697424)
- Yabsira Gebremichael (2661262)
- Kamal Lalloo (2652159)

From the start, our primary goal was to ensure that every member of the team gained hands-on experience in every area of the project: backend, frontend, testing, deployment, and documentation. We made a conscious decision to avoid rigid role separation. Instead, we worked together to build a deep understanding of the full system and shared responsibility for its development.

We approached the project by dividing up initial research tasks, for example, one person would explore Azure, another would dive into payment integration, someone else would investigate testing or AI-based recommendation systems — and then we'd reconvene to share our findings, teach each other, and implement the work together. This allowed us all to become comfortable with the complete tech stack and development process. Whether it was deploying to Azure, writing tests, setting up CI/CD pipelines, or building pages in React, each team member contributed across all parts of the system.

2.2 Contribution Approach

Although everyone worked across the project, certain team members took initiative in leading specific areas:

- **Harshil** and **Sunay** took the lead in setting up the backend architecture using Azure, ensuring a smooth foundation for the rest of the team to integrate their work. Harshil also maintained our GitHub repository and handled deployment pipelines to keep the codebase clean and accessible for everyone.

- **Matthew** took initiative on testing. He researched best practices and testing tools, then helped the rest of the team implement test cases for their individual components. This ensured that everyone understood how to test their work and contribute to overall quality assurance.
- **Yabsira** focused on the AI and personalization component. He explored how to tailor product recommendations based on user interests and search history, and worked on the logic behind that functionality.
- **Kamal** took charge of documentation and diagramming. He was responsible for the UML diagrams, keeping our documentation clear and well-organized, and also researched and implemented the Yoco payment gateway.

However, these responsibilities were never siloed. Every person was involved in every aspect , whether it was writing user stories, building out UI components, designing how the application should flow, or testing backend endpoints. We consistently reviewed and revised each other's work, and any new tool or method was explained and taught to the group. By the end of the project, every member was capable of working independently across the full stack.

2.3 Communication and Tools

We kept all of our work transparent and accessible using a combination of tools that helped us collaborate efficiently throughout the project:

- **GitHub** served as our version control system and centralized codebase. We managed issues, pull requests, and code merges collaboratively, which helped keep everyone up to date and the project well-organized.
- **Google Docs** was used to write and revise user stories, document sprints, track decisions, and prepare our final documentation. It made it easy to collaborate on written work in real time. The link to the google doc is in the Appendix.
- **Discord** was our main communication platform. We held all our meetings, daily stand-ups, sprint reviews, and general discussions there. It also served as our space for sharing learning resources including tutorials, design links, and testing guides in structured channels that everyone could access.

Scrum Methodology

4.1 Overview of Scrum Approach

For this project, we followed the **Scrum methodology** to manage our development process. Our team worked in structured sprints, each lasting approximately one to two weeks. At the start of each sprint, we held planning sessions to decide which user stories to focus on, and created sprint backlogs with clearly defined tasks. Throughout the sprints, we held regular meetings (stand-ups) on discord or in person to track progress and address blockers. At the end of each sprint, we reviewed what we had completed and discussed what could be improved for the next one.

Scrum helped us stay organised, adapt to changes, and ensure that everyone was contributing consistently. We kept our backlog prioritised and flexible, which allowed us to continuously improve and evolve the platform with each sprint.

4.2 Sprint Review Documentation & Meeting Records

We documented all our Scrum ceremonies in a shared Google Doc, including daily stand-up summaries, sprint retrospectives, and sprint reviews. Stand-ups helped us track progress and blockers, while retrospectives allowed us to reflect on what went well and what needed improvement after each sprint. Sprint reviews were used to evaluate completed work and ensure alignment with our goals.

All meeting notes and records can be found here:

https://docs.google.com/document/d/1eT510AOd3YfFSMUHpNMAOoPj18lE-Y4twHv2UM0f9wA/edit?tab=t.gwzhs5eqrkyy

4.3 Product & Sprint Backlogs

Sprint 1:

Product Backlog for sprint 1:

User Story	Priority
I want to visually see and interact with the website, so that I can explore products and navigate the platform	4
As a Buyer I want to create my account and select my interests so that I can browse products	4
As a Buyer I want to log into my account and access the home page so that I can purchase art	4
As a user, I want to scroll through the homepage, so that I can easily browse featured products and discover new items	4
As Buyer I want to select a product and view the product details so that I can review the product details	4
As an admin I want to log in so that I can access the admin dashboard	3
As an admin I would like to see an admin dashboard so that I can navigate through admin activities	3
As an admin I want to be able to see seller applications so I can begin verification	3
As an admin I want to approve a seller so that they can become a seller	3
As an admin I want to deny a seller so that they can't become a seller	3
As a Buyer I want to add a product that I am interested in to my cart so that I can purchase it at a later stage	3
As a user I want to view my profile page so that I can see my profile details	3
As a user I want to sign up as a seller so that I can perform seller activities	3
As a seller I want to set my shop details so that I can have a represented shop	3
As a seller I want to go back to the buyer side so that I can be a buyer	3
As a user I would like to change my profile pic so that I can express myself	3
As a buyer I want to be able to search for a specific product so that I can find a product I am looking for	2
As a seller I want to go to seller dashboard so that I can see my shop front	2
As a seller I want to be able to add products so my products can be viewed by buyers	2
As a seller I want to be able to edit my products so that I can update their details	2
As a user I would like to report a product so that I can help keep the site safe	2
As a buyer I want to view a shop front so that I can see products from a specific seller	2

As a seller I want to view sales analytics so I can track how well my products are doing	1
As an admin I want to view sales analytics across the site so that I can monitor platform performance	1
As a seller I want to see my order history so that I can fulfill orders	1
As a buyer I want to pay for products so that I can order products I like	1
As a user I want to see a landing page when I visit the site so that I know what the platform offers	1
As a buyer I want the cart to reflect stock count so I know what's available	1
As a user I want a well-organized footer with quick links so I can navigate easily	1
As a user I want AI-based recommendations so I can discover new products	1

The user stories focused on for the first sprint were:

- I want to visually see and interact with the website, so that I can explore products and navigate the platform
- As a Buyer I want to create my account and select my interests so that I can browse products
- As a Buyer I want to log into my account and access the home page so that I can purchase art
- As a user, I want to scroll through the homepage, so that I can easily browse featured products and discover new items
- As Buyer i want to select a product and view the product details so that i can review the product details

Sprint 1 Backlog:

Task	Story Points
Create wireframes for key pages	1
Create mockups based on the wireframes	2
Create the database to support frontend display	9
Create the UI for the sign up page	3
Create the UI for interests/question page	3
Connect the sign up page to the database and add functionality	6
Connect the interests/question page to the database and add functionality	5
Set up third-party identity provider for authentication	5
Create UI for login page	3
Connect the log in page to the database and add functionality	5
Create UI for the homepage	4
Connect the homepage to the database and add functionality	6
Create product page UI	3
Connect the product page to the database and add functionality	5

Sprint 2:

Product Backlog for sprint 2:

User Story	Priority
As an admin I want to log in so that I can access the admin dashboard	3
As an admin I would like to see an admin dashboard so that I can navigate through admin activities	3
As an admin I want to be able to see seller applications so I can begin verification	3
As an admin I want to approve a seller so that they can become a seller	3
As an admin I want to deny a seller so that they can't become a seller	3
As a Buyer I want to add a product that I am interested in to my cart so that I can purchase it at a later stage	3
As a user I want to view my profile page so that I can see my profile details	3
As a user I want to sign up as a seller so that I can perform seller activities	3
As a seller I want to set my shop details so that I can have a represented shop	3
As a seller I want to go back to the buyer side so that I can be a buyer	3
As a user I would like to change my profile pic so that I can express myself	3
As a buyer I want to be able to search for a specific product so that I can find a product I am looking for	2
As a seller I want to go to seller dashboard so that I can see my shop front	2
As a seller I want to be able to add products so my products can be viewed by buyers	2
As a seller I want to be able to edit my products so that I can update their details	2
As a user I would like to report a product so that I can help keep the site safe	2
As a buyer I want to view a shop front so that I can see products from a specific seller	2
As a seller I want to view sales analytics so I can track how well my products are doing	1
As an admin I want to view sales analytics across the site so that I can monitor platform performance	1
As a seller I want to see my order history so that I can fulfill orders	1
As a buyer I want to pay for products so that I can order products I like	1
As a user I want to see a landing page when I visit the site so that I know what the platform offers	1
As a buyer I want the cart to reflect stock count so I know what's available	1
As a user I want a well-organized footer with quick links so I can navigate easily	1
As a user I want AI-based recommendations so I can discover new products	1

The user stories focused on for the second sprint were:

- As a Buyer I want to add a product that I am interested in to my cart so that I can purchase it at a later stage.
- As a user i want to view my profile page so that i can see my profile details
- As a user i would like to change my profile pic so that i can express myself
- As a user i want to sign up as a seller so that i can seller do activities
- As an admin i want to be able to see seller applications so i can begin verification
- As an admin I want to approve a seller so that they can become a seller
- As an admin I want to deny a seller so that they can't become a seller
- As an admin i want to log in so that i can access the admin dashboard
- As an admin i would like to see an admin dashboard so that i can navigate through admin activities
- As a seller I want to set my shop details so that i can have a represented shop
- As a seller i want to go back to the buyer side so that i can be a buyer

Sprint 2 Backlog:

Task	Story Points
Create the UI for the product page	3
Add functionality to the cart page	6
Integrate third-party payment system (Yoco) into the cart page	7
Create the UI page for the profile page	3
Connect the product page to the database and add functionality	5
Ensure profile picture can change	4
Ensure buyers can request to become sellers and upon acceptance from admin, become sellers	5
Create Admin Dashboard UI	4
Create seller verification UI	3
Ensure admin can start a review (Start Review button) and Accept or Reject a request (Accept and Reject button)	6
Create the UI for user reports page for the admin dashboard	3
Add functionality to the user reports page so the admins can investigate	7
Create the UI for the seller sign up page	3
Connect the seller sign up page to the domain and add functionality	5
Create the navbar for the buyer side	4
Create the navbar for the seller side and add a return option in the navbar to go back to being a buyer	5

Sprint 3:

Product Backlog for sprint 3:

User Story	Priority
As a buyer I want to be able to search for a specific product so that I can find a product I am looking for	2
As a seller I want to go to seller dashboard so that I can see my shop front	2
As a seller I want to be able to add products so my products can be viewed by buyers	2
As a seller I want to be able to edit my products so that I can update their details	2
As a user I would like to report a product so that I can help keep the site safe	2
As a buyer I want to view a shop front so that I can see products from a specific seller	2
As a seller I want to view sales analytics so I can track how well my products are doing	1
As an admin I want to view sales analytics across the site so that I can monitor platform performance	1
As a seller I want to see my order history so that I can fulfill orders	1
As a buyer I want to pay for products so that I can order products I like	1
As a user I want to see a landing page when I visit the site so that I know what the platform offers	1
As a buyer I want the cart to reflect stock count so I know what's available	1
As a user I want a well-organized footer with quick links so I can navigate easily	1
As a user I want AI-based recommendations so I can discover new products	1

The user stories focused on for the third sprint were:

- As a seller i want to be able to add products so my products can be viewed by buyers
- As a seller i want to be able to edit my products so that i can update their details
- As a buyer i want to be able to search for a specific product so that i can find a product i am looking for
- As a buyer i want to view a shop front so that i can see products from a specific seller
- As a user i would like to report a product so that i can help keep the site safe
- As a seller i want to go to seller dashboard so that i can see my my shop front

Sprint 3 Backlog:

Task	Story Points
Create Search Page UI	4
Connect search page to database and add search functionality	7
Add filtering by tags functionality	5
Create UI for Add Product page	4
Connect Add Product page to database and add functionality	6
Create Edit Product page	4
Connect Edit Product page to database and add functionality	6
Create UI for shop front page	3
Connect shop front page to database and add functionality	5
Create "Report Product" pop-up UI	3
Connect "Report Product" pop-up to database and add functionality	5
Create seller dashboard UI	4
Connect the seller dashboard to the database and add functionality	5
Add "All" button and category-specific buttons when a product is created by a seller, to their dashboard and add functionality	6

Sprint 4:

The user stories focused on for the fourth sprint were:

- As a seller, I want to view sales analytics so I can track how well my products are doing.
- As an admin, I want to view sales analytics across the site so that I can monitor platform performance.
- As a seller, I want to see my order history so that I can fulfill orders.
- As a buyer, I want to pay for products so that I can order products I like
- As a user, I want to see a landing page when I visit the site so that I know what the platform offers.
- As a buyer, I want the cart to reflect stock count so I know what's available.
- As a user, I want a well-organized footer with quick links so I can navigate easily.
- As a user, I want AI-based recommendations so I can discover new products.

Product Backlog for sprint 4:

User Story	Priority
As a seller I want to view sales analytics so I can track how well my products are doing	1
As an admin I want to view sales analytics across the site so that I can monitor platform performance	1
As a seller I want to see my order history so that I can fulfill orders	1
As a buyer I want to pay for products so that I can order products I like	1
As a user I want to see a landing page when I visit the site so that I know what the platform offers	1
As a buyer I want the cart to reflect stock count so I know what's available	1
As a user I want a well-organized footer with quick links so I can navigate easily	1
As a user I want AI-based recommendations so I can discover new products	1

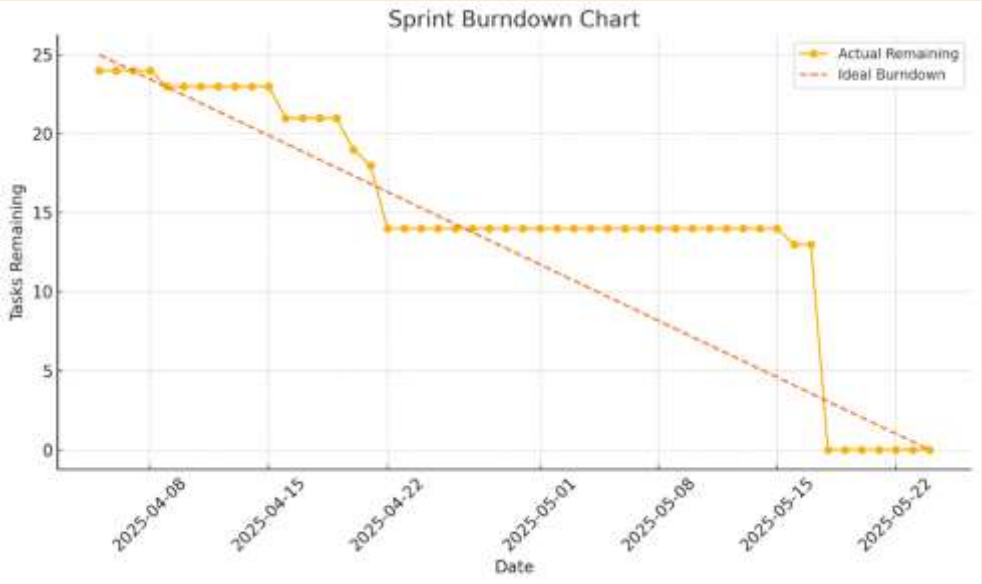
Sprint 4 Backlog:

Task	Story Points
Create Seller Stats page UI	4
Connect seller stats page to database and add functionality	8
Create stats page for Admin	4
Connect admin stats page to the database and add functionality	8
Create UI for seller orders page	4
Connect seller orders page to the database and add functionality	6
Create UI for Buyer orders page	3
Connect buyer orders page to the database and add functionality	5
Create UI for landing page	4
Connect the landing to the database and functionality	6
Add functionality that reflects current stock quantity in the cart page for a specific item	4
Create a footer with quick links inside it, for all pages that will use it	3
Create a rule-based AI system that is mainly coded in backend API calls	8

4.4 Sprint Summaries

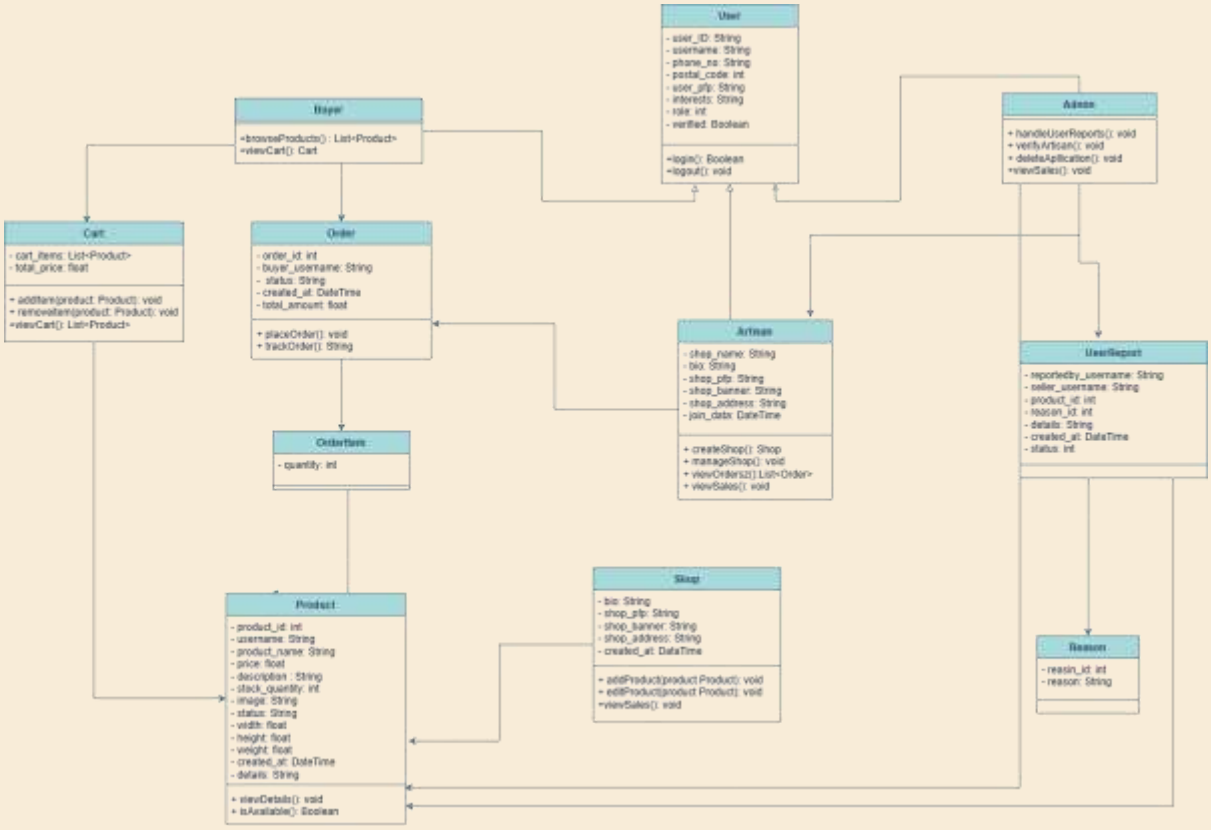
- **Sprint 1:** Focused on setting up the core buyer experience, including account creation, homepage navigation, and viewing product details.
- **Sprint 2:** Introduced seller registration, admin approval workflows, and profile customisation features for users.
- **Sprint 3:** Added seller dashboard functionality, allowing sellers to manage products and buyers to search and report items.
- **Sprint 4:** Finalised core functionality with payments, AI-based recommendations, sales analytics, and UI improvements like stock tracking and footer navigation.

4.5 Sprint Burndown Chart

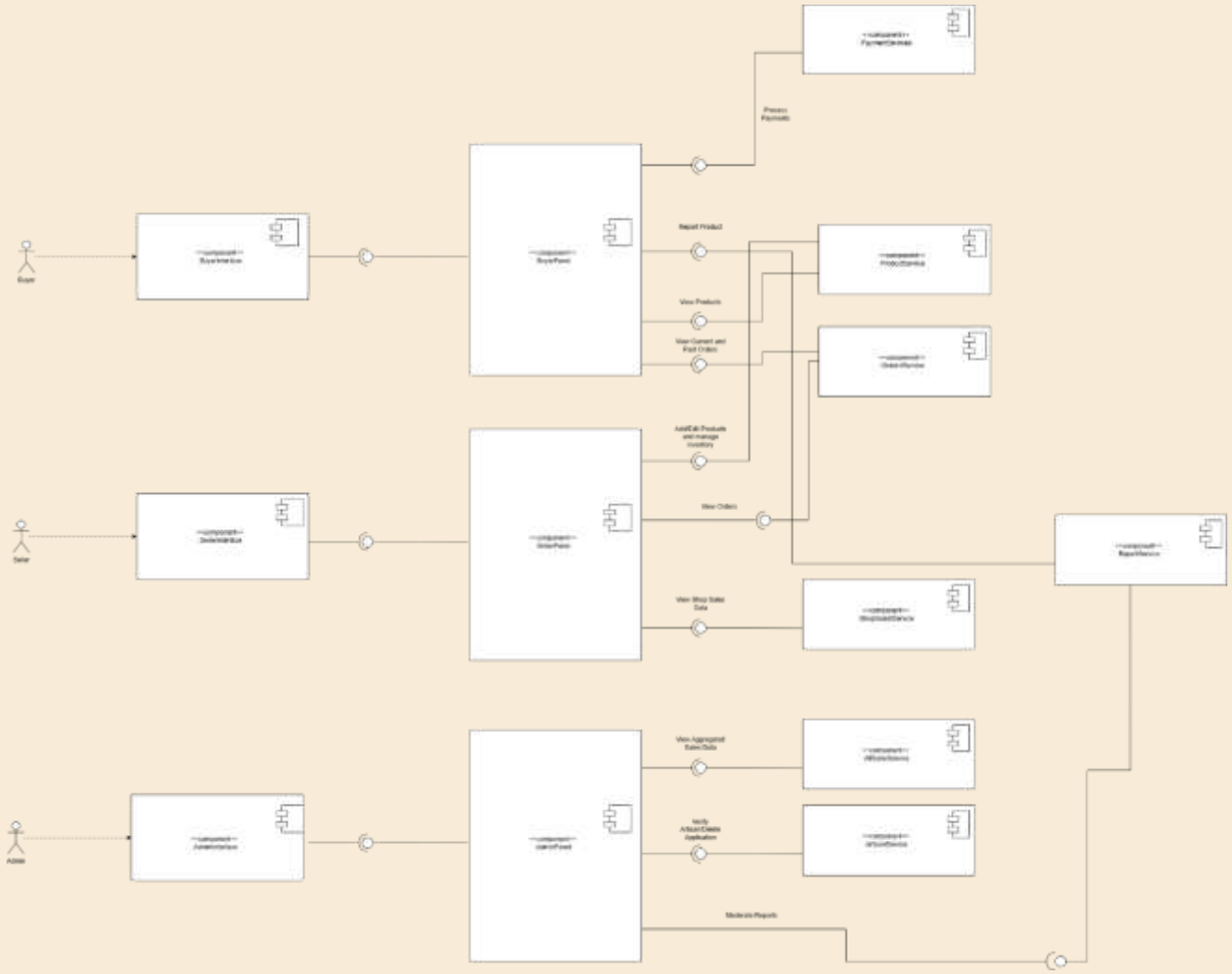


UML Diagrams(4+1)

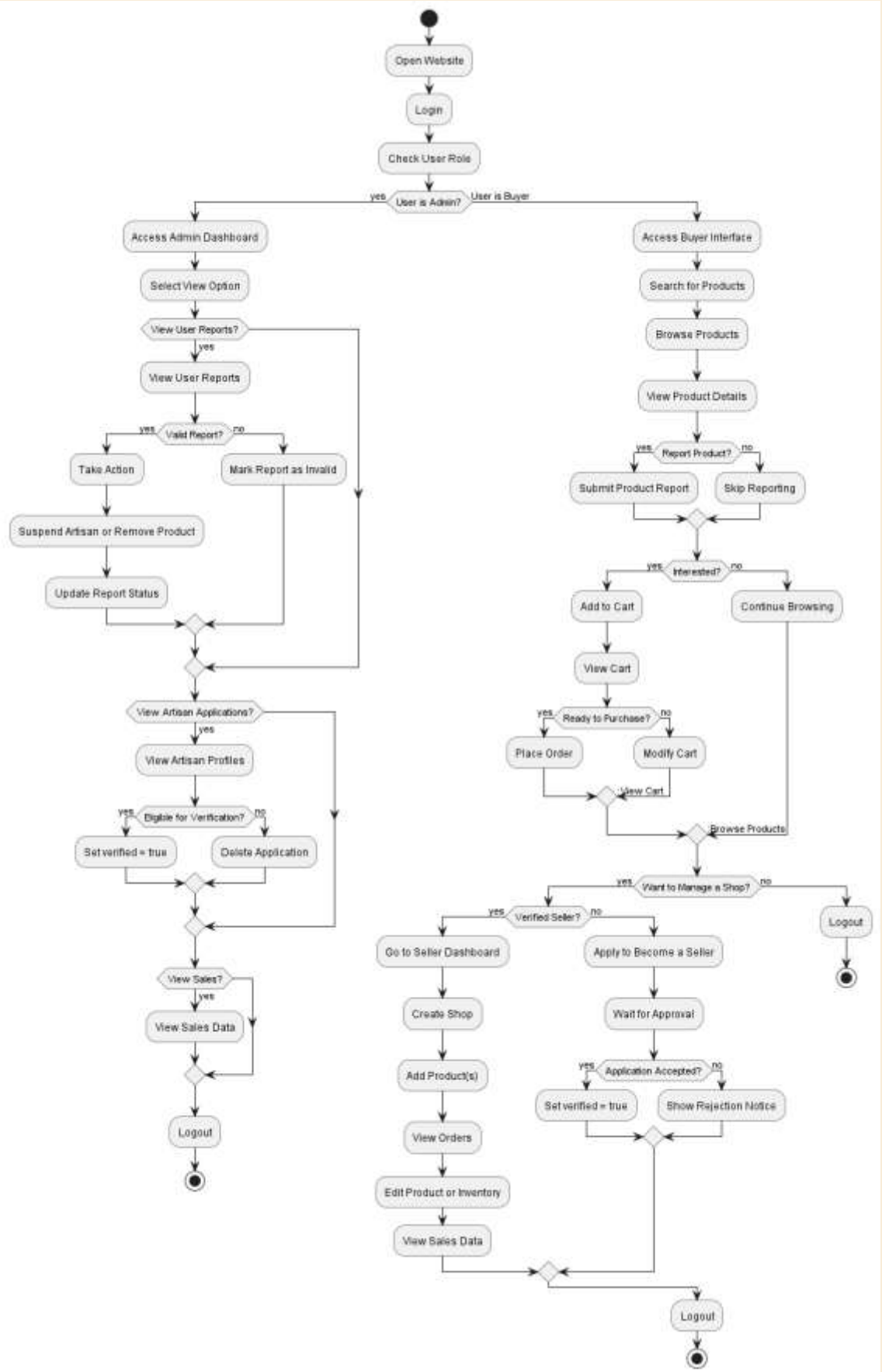
5.1 Class Diagram (Logical View)



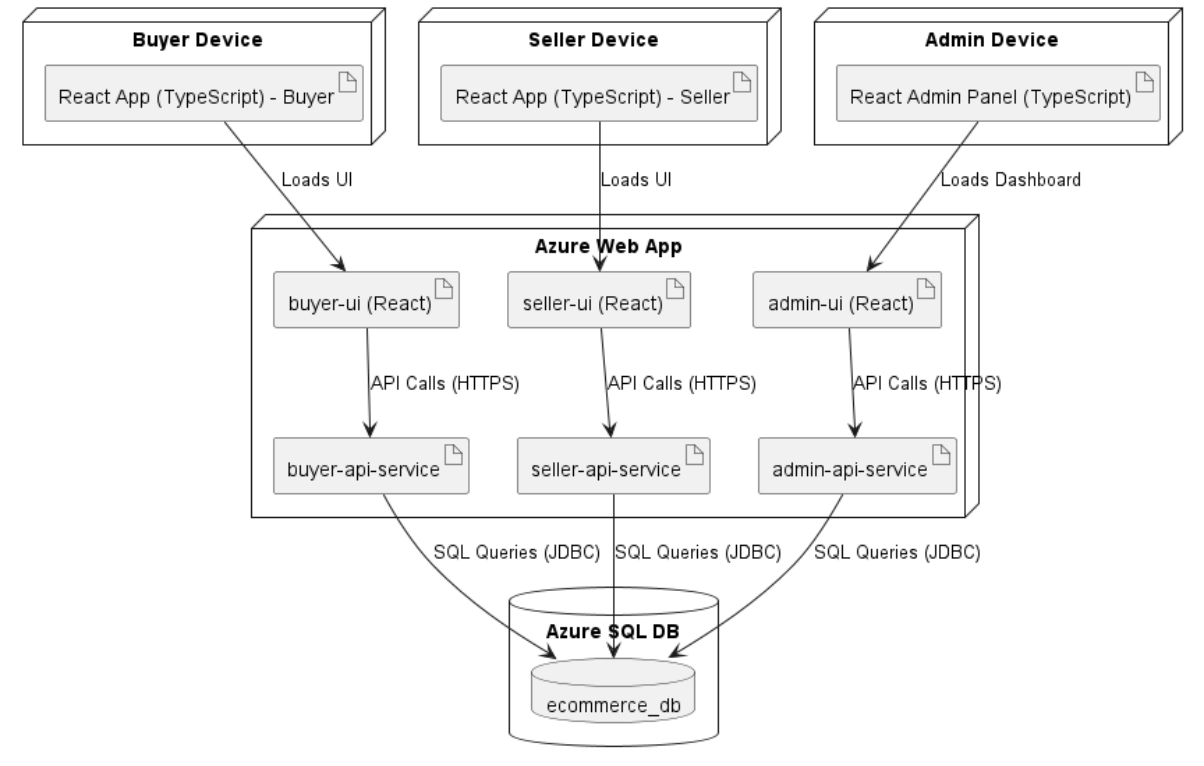
5.2 Component Diagram (Development View)



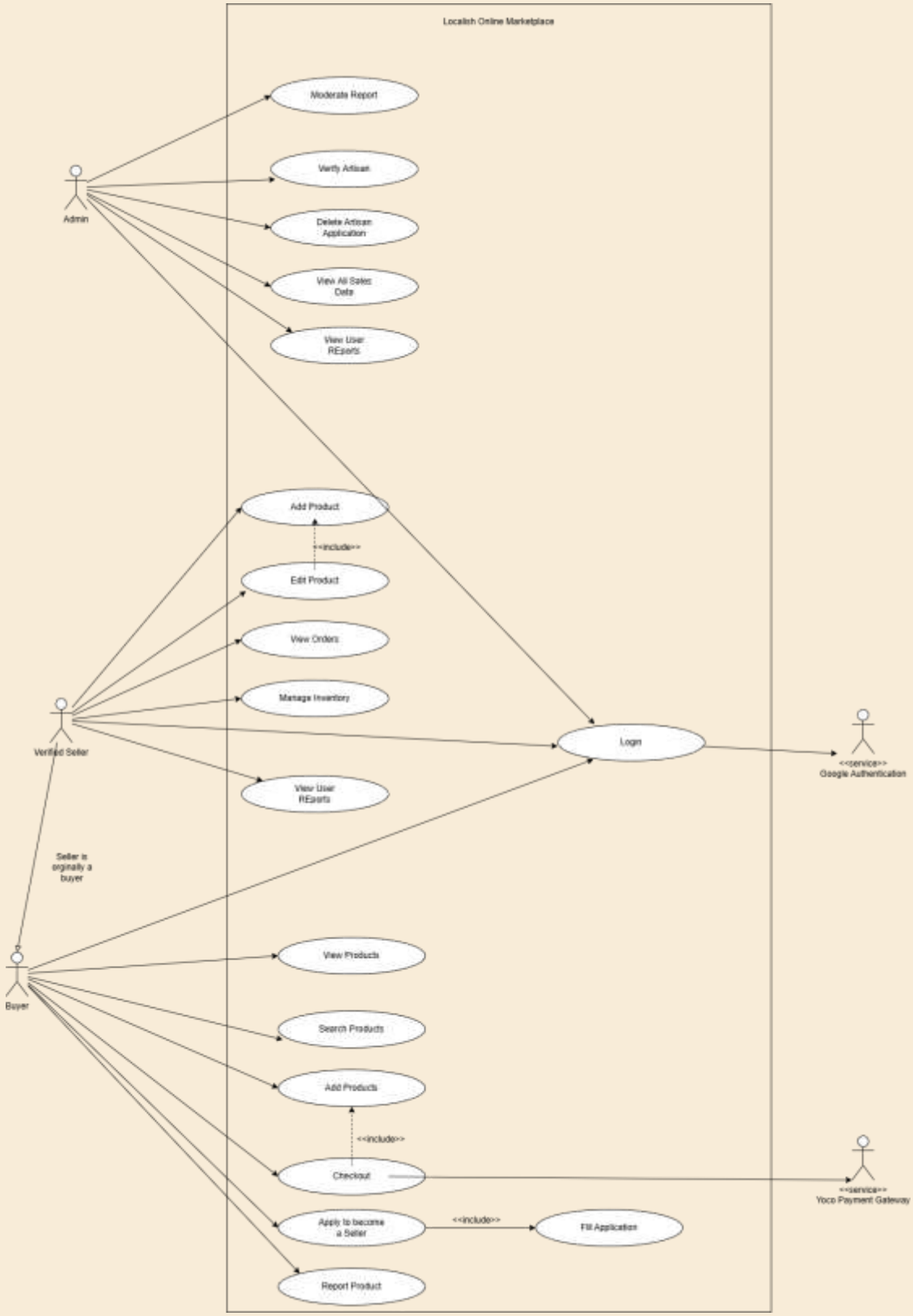
5.3 Activity Diagram (Process View)



5.4 Deployment Diagram (Physical View)



5.5 Use Case Diagram (Scenario View)



Platform Overview

1. Buyer Experience

Buyers can sign up using Google OAuth, which provides a seamless and secure login experience. During onboarding, users select their art interests, which are used along with their search history to generate a personalised “For You” page. This recommendation system is driven by a basic AI model that learns from user preferences to suggest relevant products.

Buyers can:

- Browse and search for products
- View product details
- Add items to their cart and checkout using Yoco, our integrated payment gateway
- Report products for issues like offensive content or late delivery
- View their order history
- Manage their profile, including profile image, phone number, and postal code

2. Seller Functionality

Any buyer can choose to apply to become a seller. Once approved by an admin, they gain access to the seller dashboard, where they can:

- Create their own shopfront
- Add and edit products, with the ability to include tags for better visibility
- Choose between pickup or delivery options
- Manage and track orders
- View key sales metrics, including:
    - Total and monthly sales

    - Inventory levels
    - Top-performing products

Sellers can switch between buyer and seller modes as needed.

3. Admin Capabilities

Admins are responsible for maintaining the integrity and quality of the marketplace. They can:

- View and approve or reject seller applications
- Monitor and investigate user-reported products
- Take action by suspending a seller or removing a product
- View aggregated sales data across the platform to analyse trends and ensure healthy performance

4. User Flow Summary

A user begins by logging in and is routed based on their role:

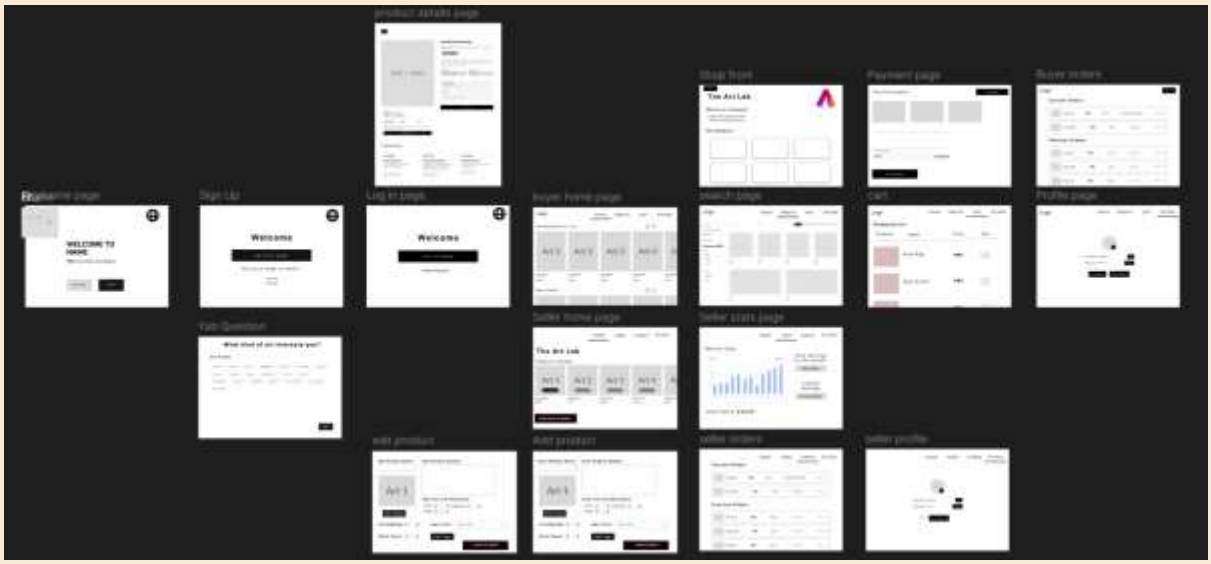
- Admins can moderate reports, review seller applications, and access analytics.
- Buyers can browse, purchase, and interact with the marketplace, or choose to apply as sellers.
- Verified sellers gain full access to their dashboard, where they manage their shop and products.

Wireframe and Mockup

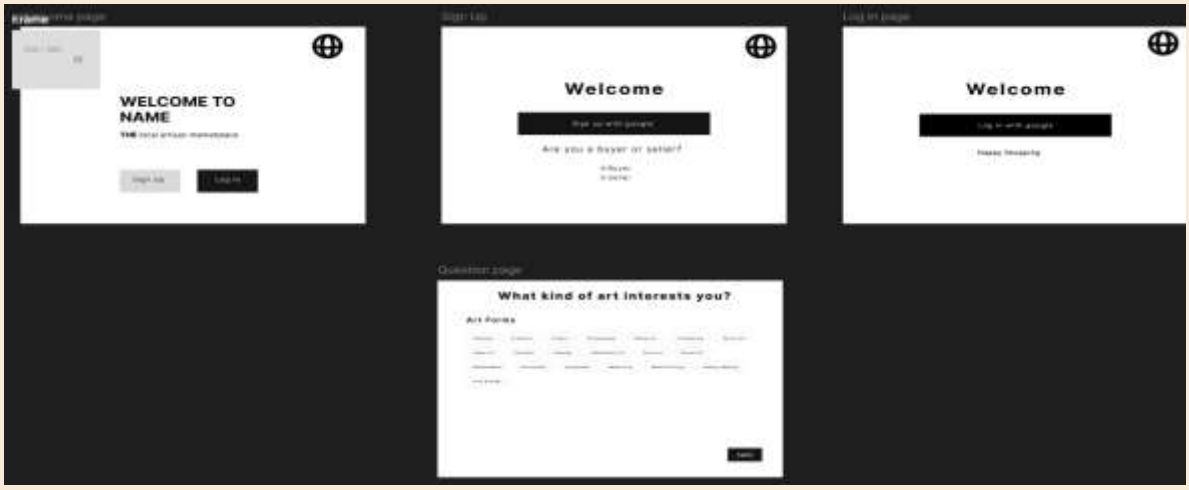
7.1 Wireframe

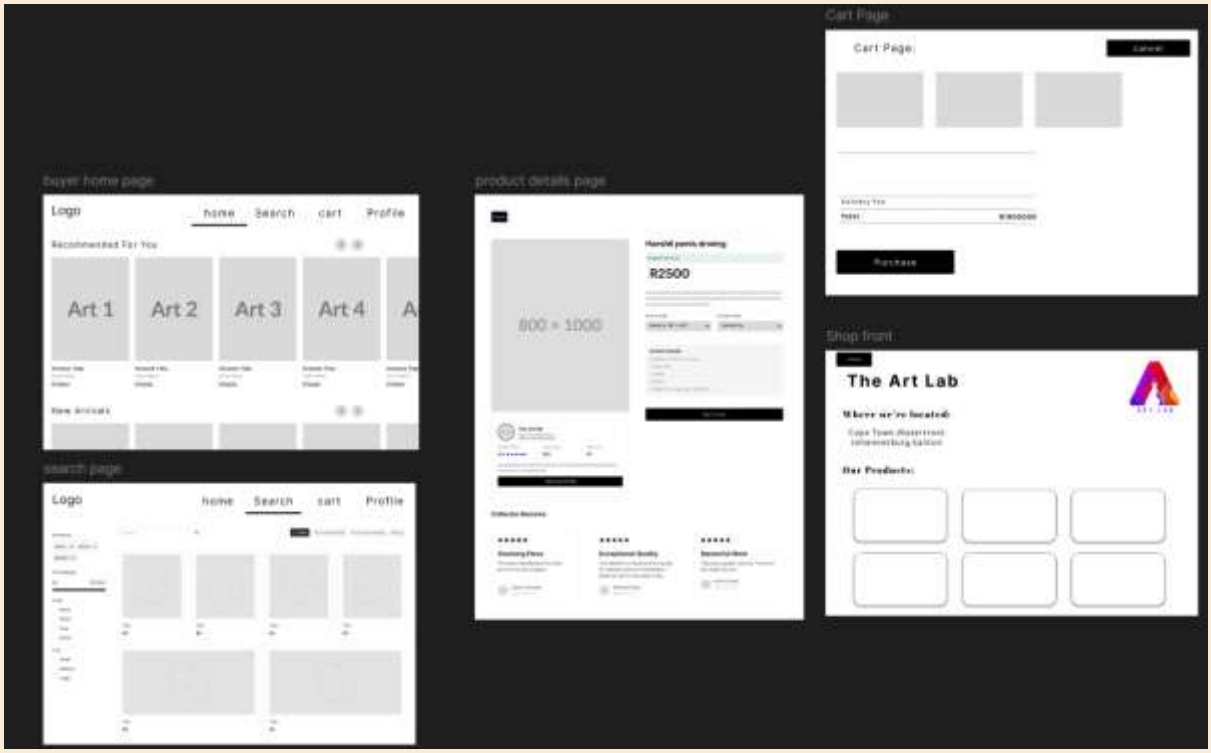
This is the basic visual design representing the skeletal framework of localish's user interface. It shows the initial layout and structure without detailed design or functionality, the initial blueprint of localish

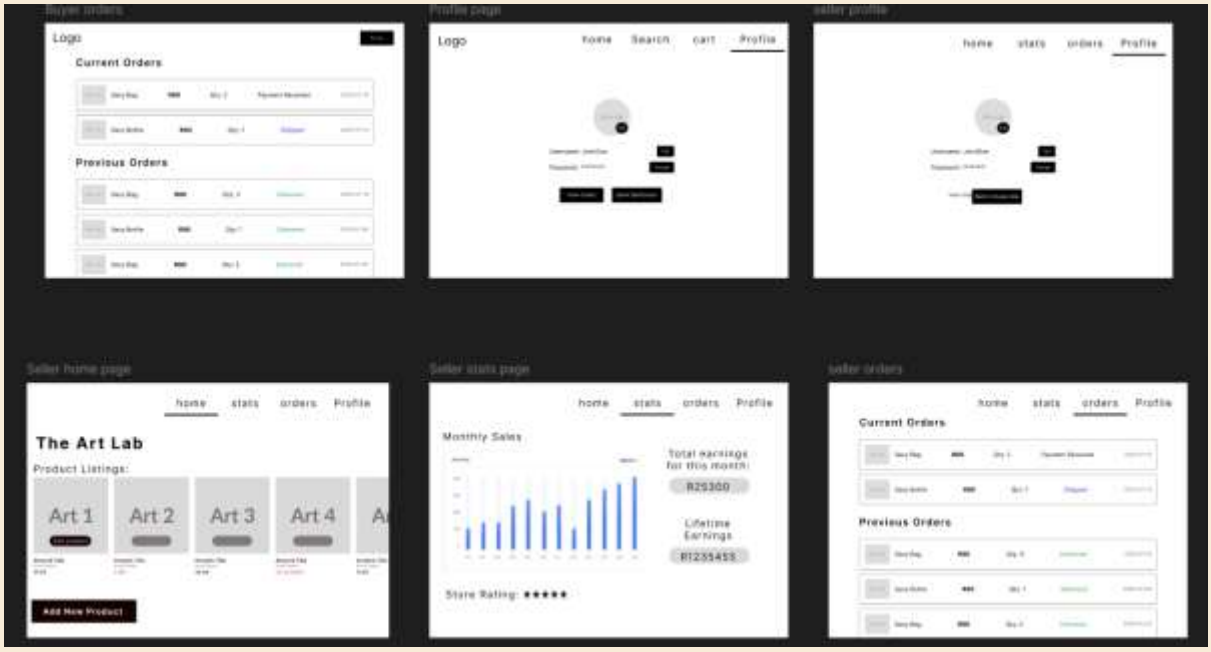
Overview:

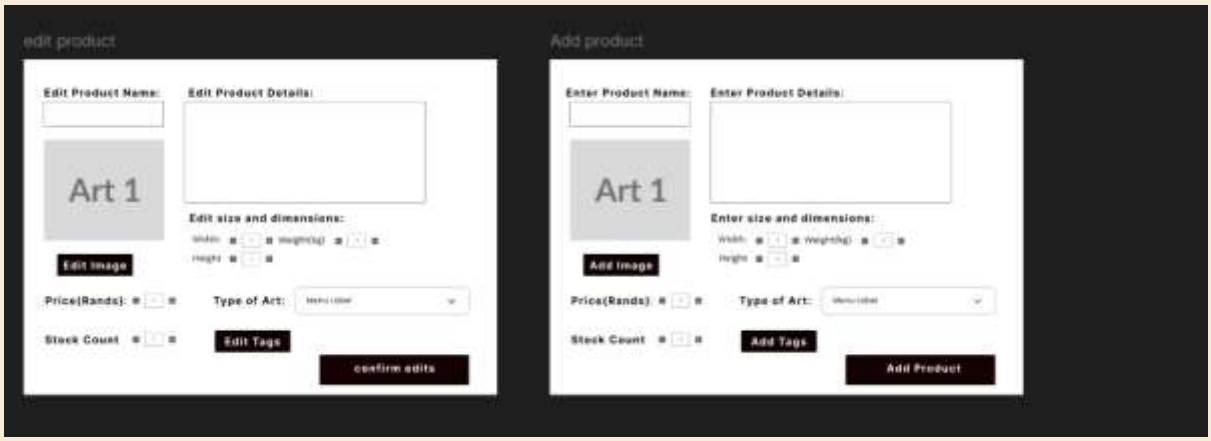


Clear view:


WELCOME TO NAME
Welcome
Are you a buyer or seller?
Welcome
What kind of art interests you?


buyer home page
Logo
home
Search
cart
Profile
Recommended For You
Art 1
Art 2
Art 3
Art 4
New Arrivals
search page
Logo
home
Search
cart
Profile
product details page
800 x 1000
R2500
Cart Page
Cart Page:
Purchase
Shop front
The Art Lab
Where we're located:
Our Products:


Logo
Current Orders
Previous Orders
Logo
home
Search
cart
Profile
home
stats
orders
Profile
home
stats
orders
Profile
The Art Lab
Product Listings:
Art 1
Art 2
Art 3
Art 4
Add New Product
home
stats
orders
Profile
Monthly Sales
Total earnings for this month:
R25300
Lifetime Earnings
R1235455
Store Rating: ★★★★★
home
stats
orders
Profile
Current Orders
Previous Orders

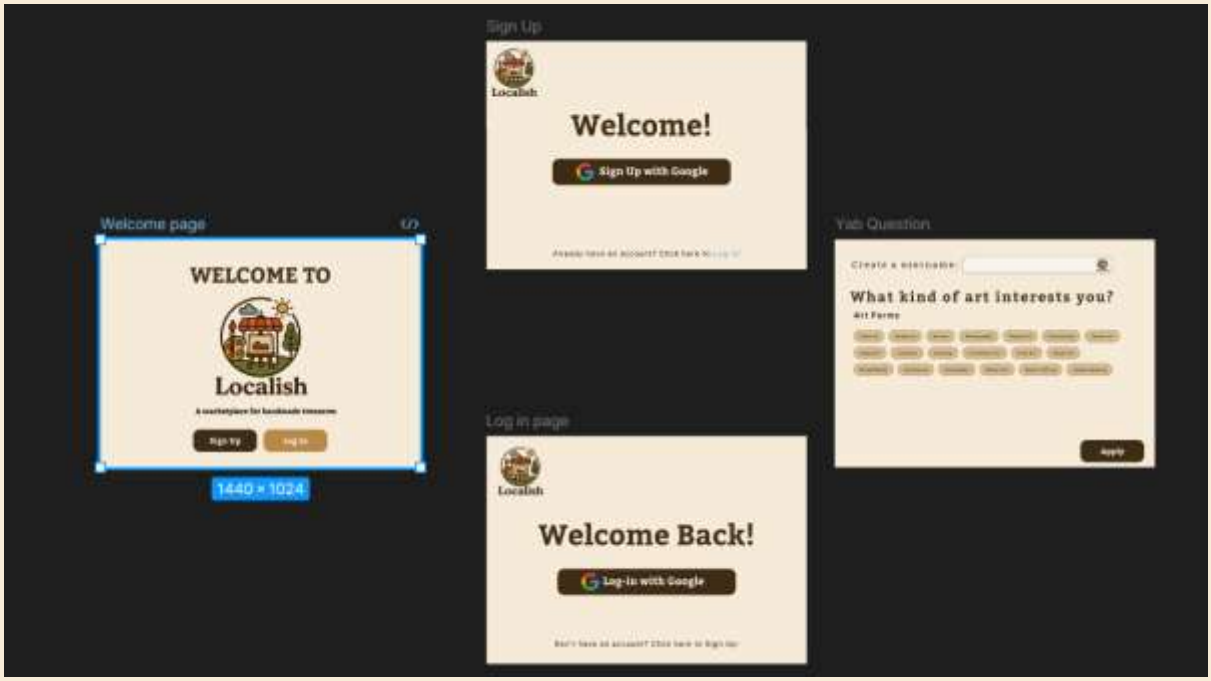

edit product
Edit Product Name:
Edit Product Details:
Art 1
Edit size and dimensions:
Edit Image
Price(Rands):
Type of Art:
Stock Count
Edit Tags
confirm edits
Add product
Enter Product Name:
Enter Product Details:
Art 1
Enter size and dimensions:
Add Image
Price(Rands):
Type of Art:
Stock Count
Add Tags
Add Product

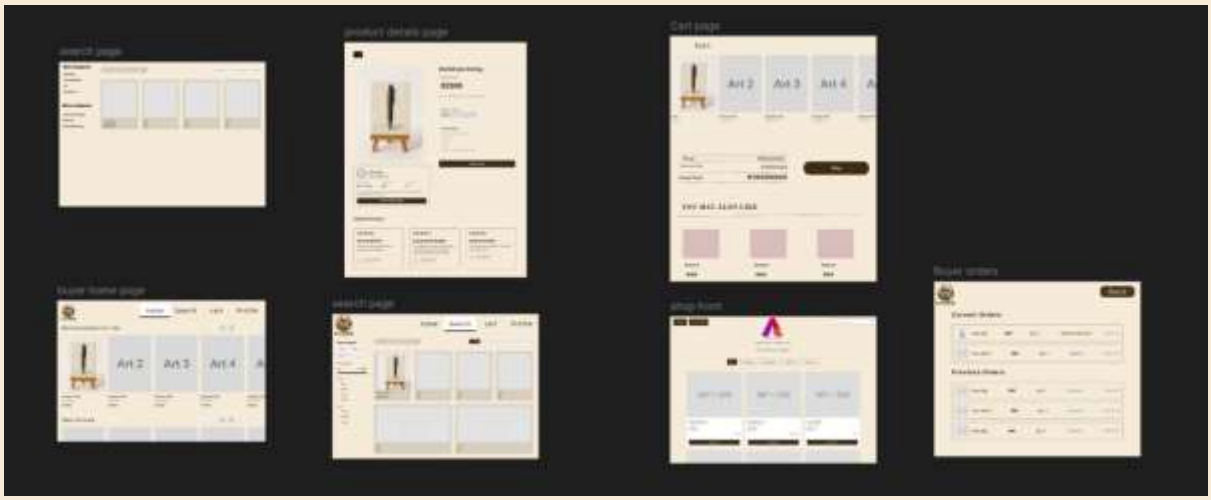
7.2 Mockup

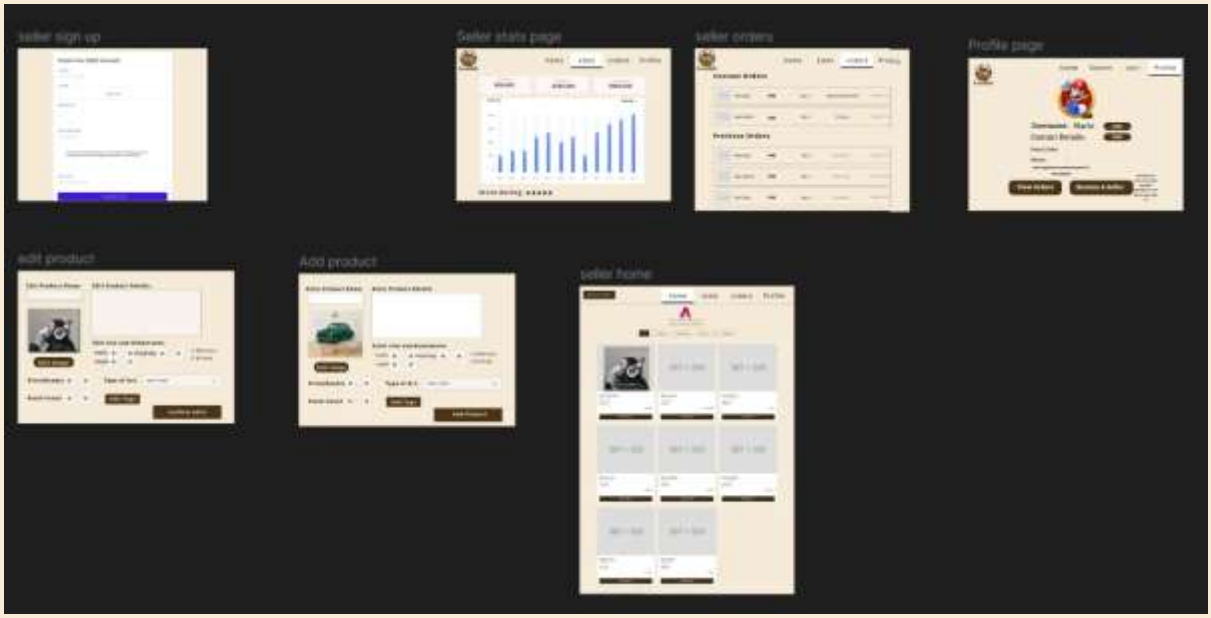
This is the higher fidelity visual design that almost fully shows what the final product will look like.

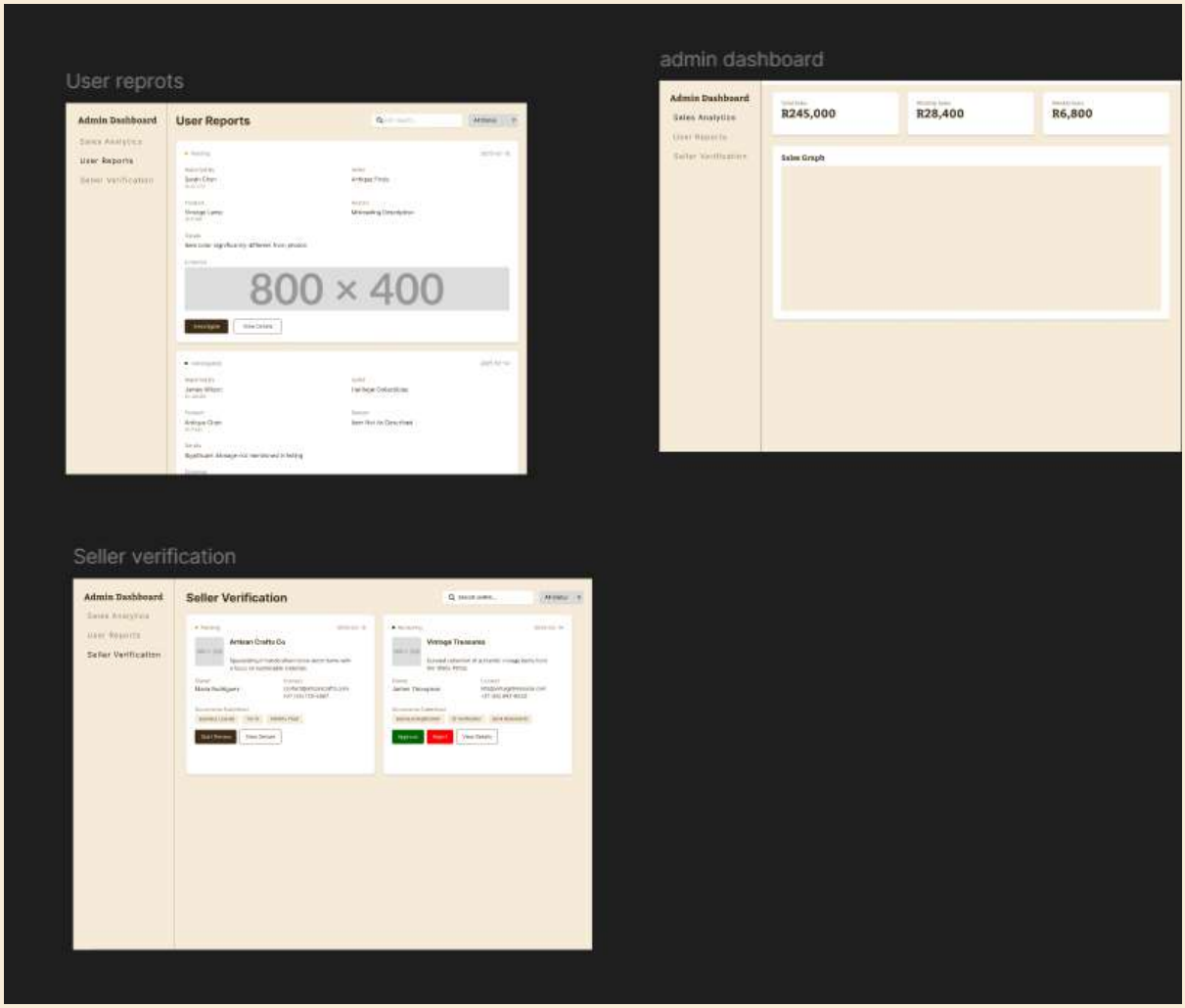
Overview:

Clear view:


Sign Up
Welcome!
Sign Up with Google
Welcome page
WELCOME TO
Localish
1440 × 1024
Yab Question
What kind of art interests you?
Log in page
Welcome Back!
Log-in with Google




seller sign up
Seller stats page
seller orders
Profile page
edit product
Add product
seller home


User reprots
Admin Dashboard
User Reports
800 × 400
admin dashboard
Admin Dashboard
R245,000
R28,400
R6,800
Sales Graph
Seller verification
Admin Dashboard
Seller Verification

Testing

8.1 Testing Strategy

We followed a test-driven development (TDD)-inspired approach, specifically the Red-Green-Refactor cycle:

1. Red (Write Failing Tests): We first outlined each desired behaviour as a .todo in the test files before writing the actual page/component logic.
2. Green (Implement Logic): After listing test intentions, we implemented the corresponding UI and logic until the tests passed.
3. Refactor: Once tests were green, we refactored the codebase while ensuring that functionality remained intact.

This process helped us focus on user intent and ensured each feature was coded with test coverage from the start.

1. Tools & Frameworks Used

- Jest: JavaScript testing framework used for unit and integration tests.
- React Testing Library: Simulates user interactions and tests rendered output.
- jsdom: Provides a virtual DOM environment.
- Axios mocking: Replaces actual API calls with controlled responses.
- Google OAuth mocking: Simulates Google login flows using mocks.

2. Testing Approach

Our testing strategy is structured around ensuring:

- Robust user authentication flows are correctly handled.
- Critical user journeys, including shopping, selling, browsing, and admin operations, behave as expected.
- All components and pages render correctly and respond to user inputs.
- Error boundaries and edge cases are managed without crashing.
- Navigation and redirects are verified by mocking route transitions.

3. Test Design

Tests were structured as:

- Unit tests for isolated components
- Integration tests for pages covering multiple component interactions and state changes.
- Async flow tests including fetch calls, product CRUD, and cart operations.

8.2 Test Results

Summary

```
Test Suites: 22 passed, 22 total
Tests:       124 passed, 124 total
Snapshots:   0 total
Time:        12.746 s
Ran all test suites.
```

Confirmed PASS:

- LoginPage.test.tsx

```
PASS src/Tests/LoginPage.test.tsx
```

- AddProductPage.test.tsx

```
PASS src/Tests/AddProductPage.test.tsx
```

- Cart.test.tsx

```
PASS src/Tests/Cart.test.tsx
```

- ProductPage.test.tsx

```
PASS src/Tests/ProductPage.test.tsx
```

- SearchPage.test.tsx

```
PASS src/Tests/SearchPage.test.tsx
```

- Home.test.tsx

```
PASS src/Tests/Home.test.tsx
```

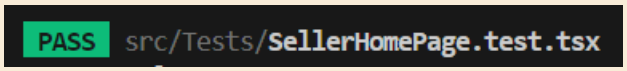
- AdminDashboard.test.tsx

```
PASS src/Tests/AdminDashboard.test.tsx
```

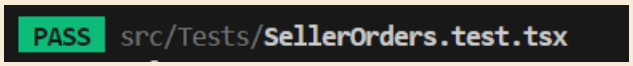
- UserReports.test.tsx

```
PASS src/Tests/UserReports.test.tsx
```

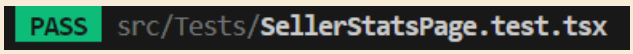
- SellerHomePage.test.tsx



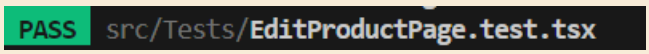
- SellerOrders.test.tsx



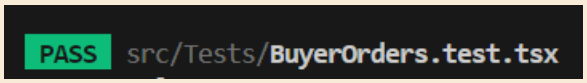
- SellerStatsPage.test.tsx



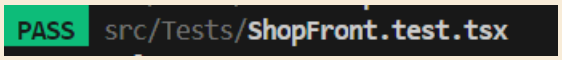
- EditProductPage.test.tsx



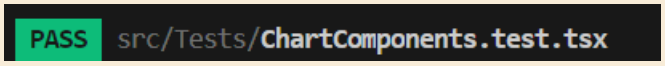
- BuyerOrders.test.tsx



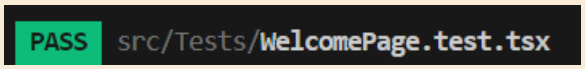
- ShopFront.test.tsx



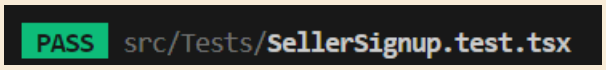
- ChartComponents.test.tsx



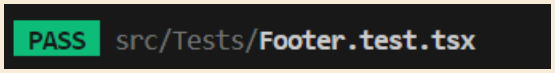
- WelcomePage.test.tsx



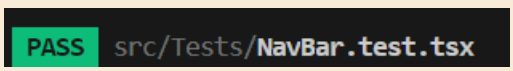
- SellerSignup.test.tsx



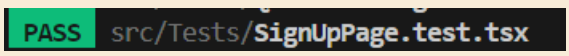
- Footer.test.tsx



- NavBar.test.tsx



- SignUpPage.test.tsx



- QuestionsPage.test.tsx

- Profile.test.tsx

Our Jest-based testing framework achieves comprehensive coverage of all major features and user flows in the app. While some network mocks caused intentional errors to test error boundaries, no unhandled exceptions or crashes occurred.

All test suites passed successfully, demonstrating stable functionality across the platform.

BONUS - AI Recommendation System

Localish employs an intuitive, behavior-driven AI recommendation system designed to personalize the homepage experience. This system dynamically adjusts product suggestions based on two primary user behaviors:

- Real-time Click Behavior
- Explicit Preferences selected once at signup

We have two api calls that track the main categories and/or minor categories whenever a product is clicked, and records these categories in the respective tables:

- **Main Tag Scores:** Tracks both click activity and explicit user preferences (Page asking for interests upon sign up) for each main category ("Painting", "Woodwork").
- **Minor Tag Scores:** Tracks click behavior only for more specific sub/minor categories ("Leather goods").

Each category linked to a product that a user interacts with gets updated in real-time. These tables store counts and timestamps for how recently and frequently a category was interacted with, which makes the localish's dynamic personalization possible.

From these tables we then take the entries and make the necessary score calculations, which will be used to provide a score to each category. The scoring calculations are as follows:

- **Click Score:** Used for both main and minor categories, the click score represents how often and how recently a user has interacted with a product containing a given category.

$$ClickScore(category) = ClickCount \times 0.8^{dslc}$$

$$dslc = days\ since\ last\ clicked$$

The exponent introduces decay over time. The further back a click occurred,

the less influence it has. recent and repeated clicks carry more weight,

keeping the recommendations fresh.

- **Explicit Score** (Main categories only): Used only for main categories that a user explicitly selected during signup.

$$ExplicitScore(Category) = SelectionScore \times 0.9^{dss}$$

$$dss = days\ since\ signup$$

It Captures user-declared preferences. Slight decay gives them lasting but gradually reducing influence, allowing actual behavior (clicks) to take priority over time.

- **Final Scores**: These are the scores used to rank and sort categories recommendations:

1. Minor categories: $FinalScore = ClickScore$
2. Main categories: $FinalScore = (0.9 \times ClickScore) + (0.1 \times ExplicitScore)$

This hybrid score ensures that a user's explicit interests influence recommendations without outweighing their real-time behavior.

When fetching the recommended items for the homepage (For you page) It alternates between products related to minor and main categories:

First we fetch all categories with non-zero final scores, we then sort categories separately:

- Minor categories will be sorted by **click score**
- Main categories will be sorted by **final score**

After that we simply display the product batches by using an alternating pattern:

First we fetch items for the top 2 Minor categories, then fetch items related to the top Main categories, then the next 2 Minor categories and then the next top Main category etc…

- For each category: It shows up to 10 products associated with it, it ensures to randomly select if more than 10 exist to create a fresh feel and ensure that there are no repeated products across categories, shown on the For you page.

Cart Page:

The cart page also makes use of a recommendation system, but it is much more simple in nature. It simply looks at the items in a users cart, takes note of the categories associated with those products and recommends other products that share those tags in the “You may also like” section.

Appendix

9.1 Centralised Project Documentation

To stay organised and maintain transparency throughout the project, we kept a shared Google Doc that served as our single source of truth. This document includes:

- Sprint-by-sprint meeting minutes
- To-do lists and task allocations
- User stories (grouped by sprint)
- UML diagrams created during design and planning
- Retrospective notes
- Contribution tracking and planning discussions

Link to project documentation:
https://docs.google.com/document/d/1eT510AOd3YfFSMUHpNMAOoPj18lE-Y4twHv2UM0f9wA/edit?usp=sharing

9.2 UI Design & Prototyping

All our user interface designs, mockups, and the clickable prototype were created collaboratively using **Figma**. These mockups were the foundation for our frontend development and were regularly reviewed and updated based on feedback from the team.

Figma allowed us to visualise our design ideas early on, test user flows, and ensure a consistent look and feel across the platform.

Figma Design File:
https://www.figma.com/design/PBgCosURN3MxwWmN574Cbv/Ui-designing?node-id=0-1&p=f&t=Bzc4NgWLS3dwj3yj-0

Interactive Prototype:

https://www.figma.com/proto/PBgCosURN3MxwWmN574Cbv/Ui-designing?node-id=5-2&starting-point-node-id=5%3A2&scaling=scale-down&content-scaling=fixed

9.3 GitHub Repository

Our full source code is available in a public GitHub repository. It contains:

- The complete backend and frontend codebases
- README files with setup instructions
- Well-maintained commit history
- Branches showing collaborative development

GitHub Repository:
https://github.com/uno791/local-artisan-marketplace

9.4 Live Demo

We deployed both the frontend and backend to Azure to make the application publicly accessible for demonstration and evaluation.

- **Frontend:**
  https://calm-meadow-0fbb07c03.6.azurestaticapps.net
- **Backend API:**
  https://artisan-api-server1-fzd8fbfwgkc2cqe2.southafricanorth-01.azurewebsites.net