

**IS2108: Pair Project**

**Design of AuroraMart**

Project Pair 14

| Name | Student ID |
| --- | --- |
| Muhammad Hamzi Bin Roslan | A2103795X |
| Nathan Ng Kai | A0273550H |

[**1. Scope and Objectives 3**](#_thrvdfgrncvg)

[1.1 Project Scope 3](#_9xkbaadiqhi5)

[1.2 Objectives 3](#_kb3jzkeczxnh)

[**2. Overview of AuroraMart 3**](#_5fx8fbf9hpg)

[**3. Business Context 4**](#_gkbl8cmr5hmp)

[3.1 Market Environment 4](#_7p7jq8lfy8oc)

[3.2 Target Audience 4](#_wlcl4zdj4h9z)

[3.3 Key Stakeholders 4](#_s7hpfo5l86hw)

[**4. Key Features 5**](#_lwzm1c8cmz9l)

[**5. User Stories 6**](#_sj3eumdd6zm3)

[5.1 Storefront 6](#_vv6ykjhqsz0i)

[**6. Wireframes 8**](#_lxd81eltag7)

[**7. Relation Data Model 18**](#_mkuu905owh66)

[7.1 Product Table (updated for Stock Management) 18](#_82uk1jsel9v6)

[7.2 Customer Table 19](#_wggwfb34kq)

[**8. Use of AI Tools 20**](#_s28tgtarqtik)

[**9. Appendices 21**](#_eukupiudm2ap)

[9.1 Overview of Files 21](#_a25f5wywl0zm)

# 1. Scope and Objectives

## 1.1 Project Scope

AuroraMart is a full-stack e-commerce platform developed to improve how users shop online through personalisation and intelligent recommendations. Built using Django, the system includes both a consumer-facing storefront and an admin panel for product and inventory management. This project applies full-stack development, object-relational mapping, and AI model integration to deliver a functional B2C prototype.

## 1.2 Objectives

The project addresses two key pain points in e-commerce: users dropping off due to irrelevant products, and businesses over-relying on discounts to drive conversions. AuroraMart uses a decision tree model to personalise the first-time user experience and an association rules model to recommend relevant add-ons at checkout. These features aim to improve conversion rates, basket size, and retention through data-driven guidance rather than aggressive pricing.

# 2. Overview of AuroraMart

AuroraMart targets value-conscious and time-sensitive shoppers in Southeast Asia, including students and young professionals. These users often seek convenience, relevance, and a clean shopping experience without being overwhelmed by endless listings.

Our goal is to help users discover useful products quickly. First-time users are directed to personalised category pages, while returning users enjoy curated suggestions based on their shopping patterns. At checkout, relevant add-ons are surfaced through intelligent recommendations.

Most shoppers drop off within 90 seconds if they don’t find what they need. AuroraMart solves this by leading with relevance. We use data and behavioural patterns to immediately surface meaningful products. We also aim to retain users by creating a consistent and engaging experience powered by smart recommendations, not just sales.

We are not trying to outscale Shopee or Lazada. Instead, AuroraMart focuses on guided discovery and tailored shopping journeys. Every visit feels intentional, efficient, and personal.

# 3. Business Context

## 3.1 Market Environment

AuroraMart is entering a highly saturated e-commerce market that includes major players such as Shopee, Lazada, and Amazon. These platforms are known for their wide product variety, fast delivery, and low prices. While they offer convenience, users often face choice overload and struggle to find relevant products quickly.

AuroraMart positions itself differently by focusing on guided discovery. Instead of overwhelming users with too many options, the platform uses data to personalise the shopping experience from the very first visit. By showing users what matters to them within seconds, AuroraMart reduces friction and increases engagement in a way that large marketplaces often overlook.

## 3.2 Target Audience

AuroraMart is designed for modern online shoppers who value efficiency and relevance. The platform primarily serves two groups. The first group is consumers such as students, young professionals, and everyday shoppers who want an easier way to find useful products without excessive scrolling. The second group is merchants, particularly small to mid-sized sellers who want their products to be seen by the right audience without having to compete through heavy advertising.

## 3.3 Key Stakeholders

The key stakeholders involved in AuroraMart include three groups. The first group is consumers who use the platform to find and purchase products in a personalised and efficient manner. The second group is merchants who rely on the platform to display their products and reach the right customers. The third group is system administrators who manage the backend operations including the product catalogue, inventory levels, and customer data. Each group plays an important role in ensuring the platform operates smoothly and delivers a quality experience to all users.

# 4. Key Features

1. **Personalised Storefront (AI Integration)**

New users are shown a curated storefront based on their demographic input. A Decision Tree model predicts their preferred product category, enabling relevant discovery within the first 90 seconds of arrival.

1. **Category Browsing**

Users can browse products by category, such as Electronics, Fashion, and Beauty. This supports intuitive navigation for both new and returning users.

1. **Product Detail Page**

Each product has a dedicated page with its image, name, price, stock status, and description. This allows users to review product details before making a decision.

1. **Shopping Cart and Checkout**

Users can add products to a session-based cart, view total costs, and proceed to checkout. A confirmation page is displayed after a successful order, with no real payment logic implemented.

1. **Suggested Items at Checkout (AI Integration)**

At checkout, the system displays “Frequently Bought Together” suggestions based on an Association Rules model trained on past transaction data. This encourages relevant upselling in a non-intrusive way.

1. **Admin Product Management (Custom Panel)**

Admins can add, edit, or remove products through a custom-built admin panel. Product fields include name, description, price, category, stock, and reorder threshold. This panel was built using Django views, forms, and templates — no Django admin interface was used.

1. **Admin Stock Management (Custom Panel)**

Admins can track inventory levels via the stock tab. Products that fall below a predefined threshold are flagged for restocking. All management is done through the custom admin panel interface.

1. **AI Integration Summary**

Onboarding Personalisation (Decision Tree): Users input age, gender, employment status, and income → Decision Tree model predicts preferred product category → Redirects user to that category page.

Checkout Suggestions (Association Rules): When users check out, the system looks at their basket items → Uses Association Rules to suggest “Frequently Bought Together” products.

# 5. User Stories

## 5.1 Storefront

| Feature | Role | Goal | Benefit | Idea |
| --- | --- | --- | --- | --- |
| Personalised Storefront | As a new customer | I want to enter basic info during onboarding | So I can be shown a personalised storefront | Uses the Decision Tree ML model to predict the preferred category |
| Category Browsing | As a returning customer | I want to browse products by category | So I can easily find what I’m looking for | Standard category browsing functionality |
| Product Detail Page | As a customer | I want to view product details | So I can make informed decisions before adding to the cart | Product page with image, price, description, and stock info |
| Shopping Cart and Checkout | As a customer | I want to add items to my shopping cart | So I can buy multiple items at once | Basic cart stored in session with total cost at checkout |
| AI-Powered Suggestions at Checkout | As a customer | I want to see suggested items at checkout | So I can consider adding frequently bought-together items | Uses the Association Rules ML model to show “Frequently Bought Together” items during checkout, improving basket size and product discovery. |
| Order Confirmation | As a customer | I want to receive confirmation after checkout | So I know that my order was placed successfully | Displays a basic confirmation screen (no payment processing) |

# 

# 

# 

# 

# 

# 

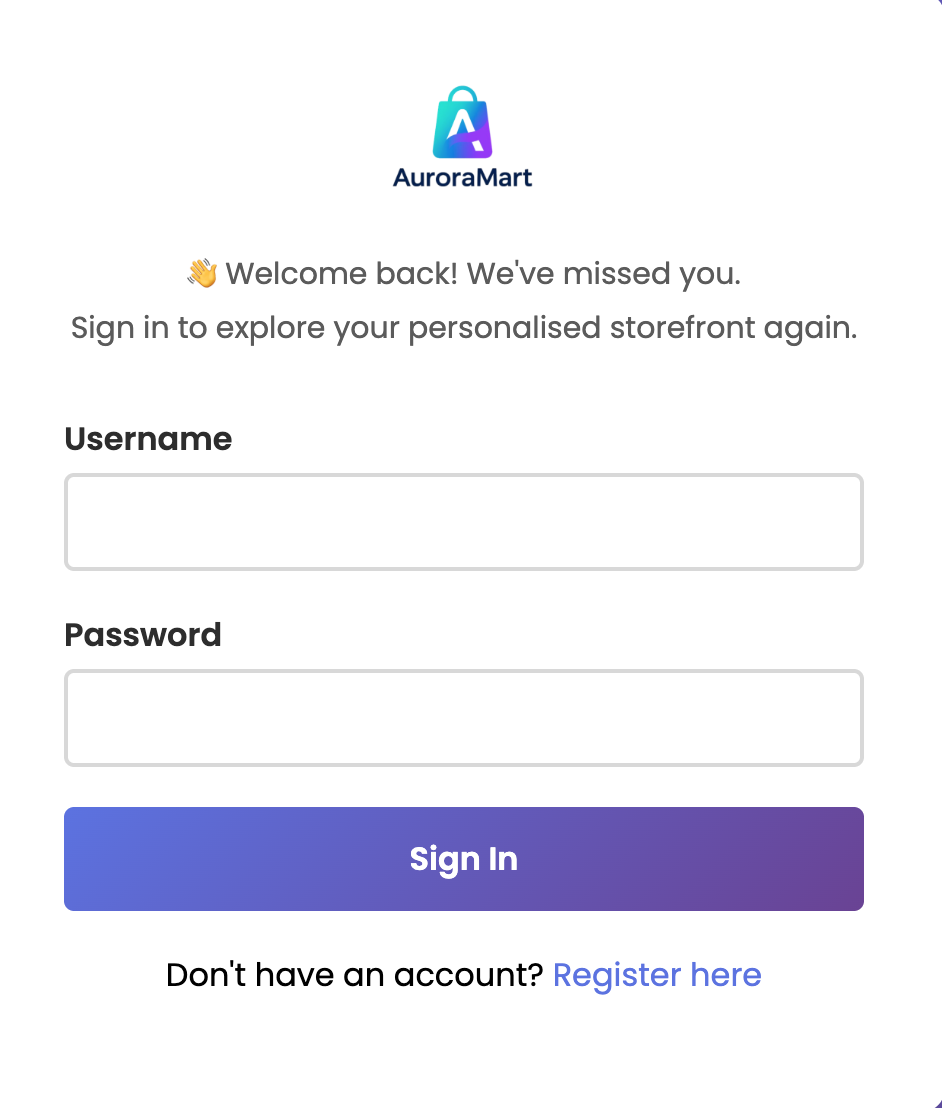
# 6. Wireframes

**Customer Registration**

## 

***Image 1:*** *Form for new users to register and input demographics for AI personalisation.*

**Customer Login**



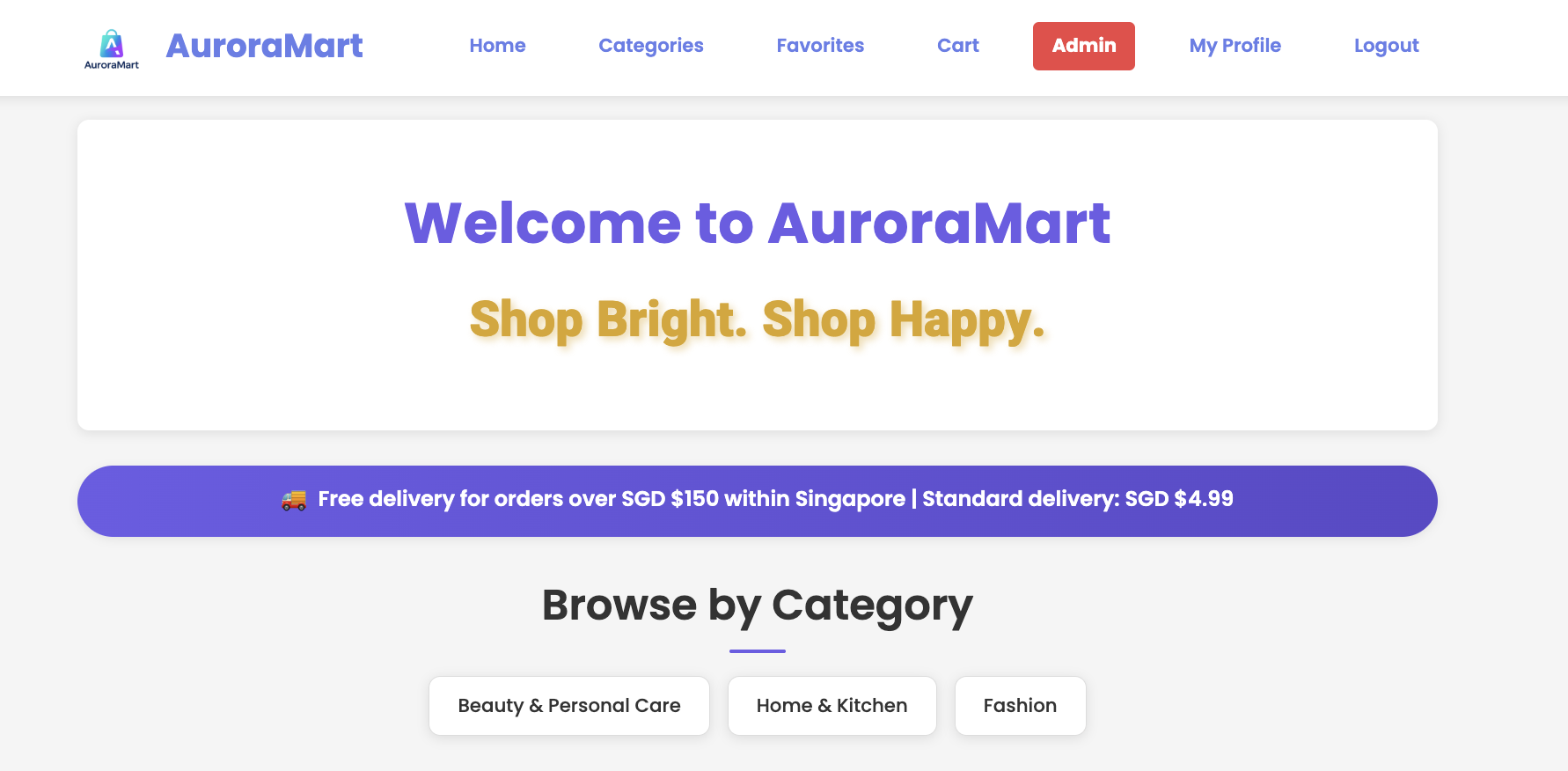
***Image 2:*** *Login page for returning customers to access their account.*

**Admin Login**



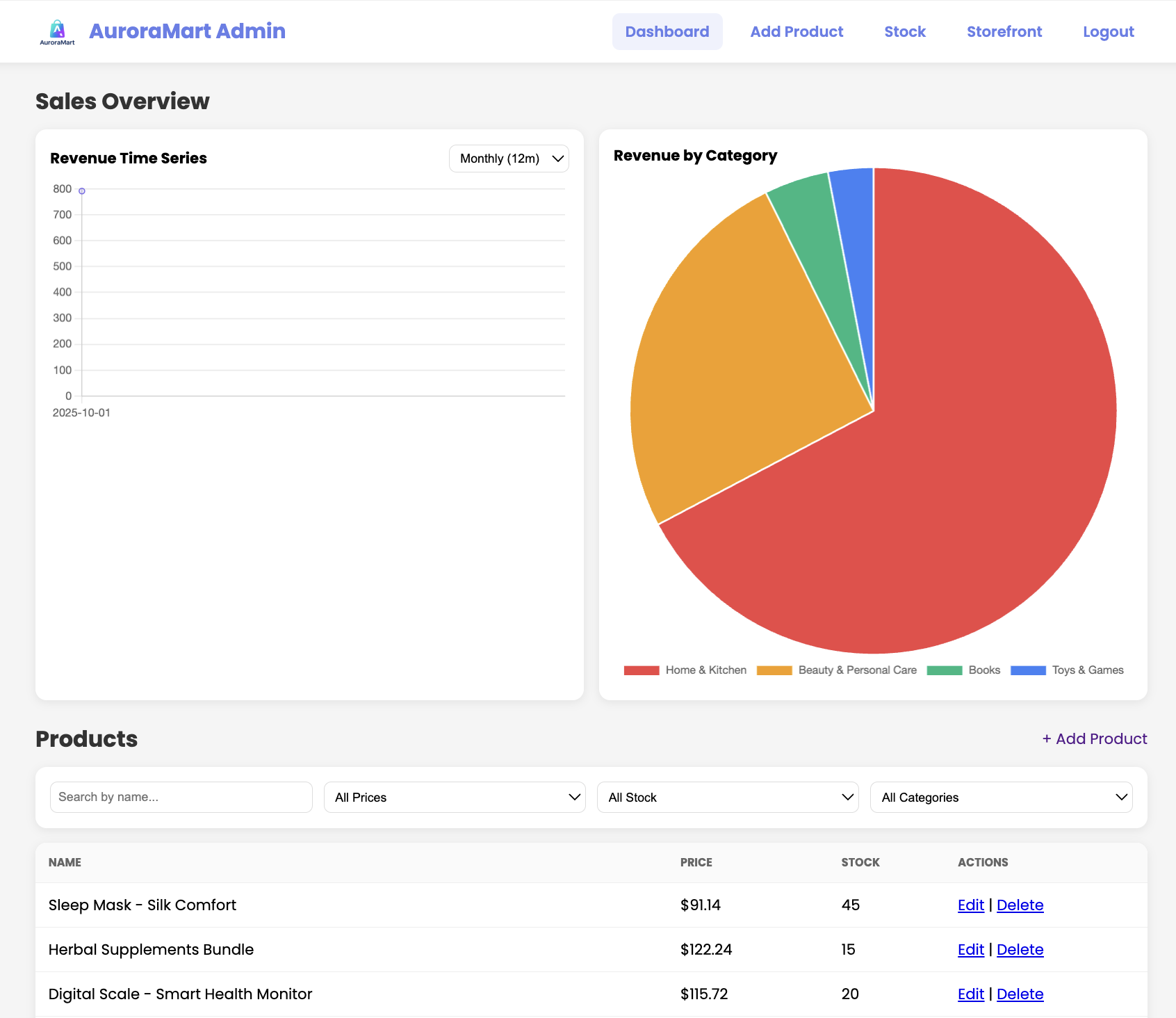
***Image 3:*** *Secure same login screen for administrators.*

**Storefront Page (Admin)**

****

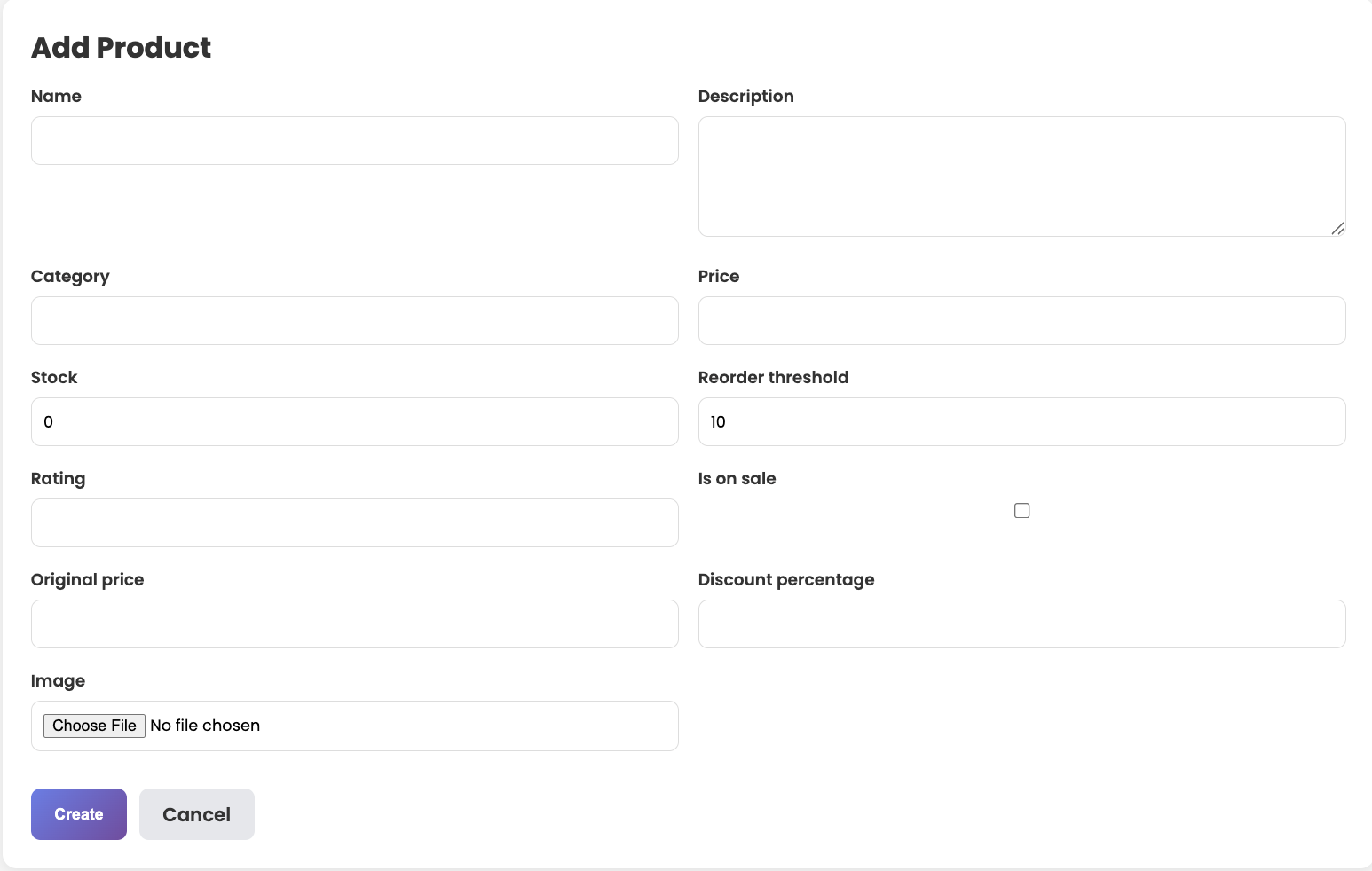
***Image 4:*** *Admin view of the storefront homepage, with access to backend controls****.***

**Admin Dashboard Page**

****

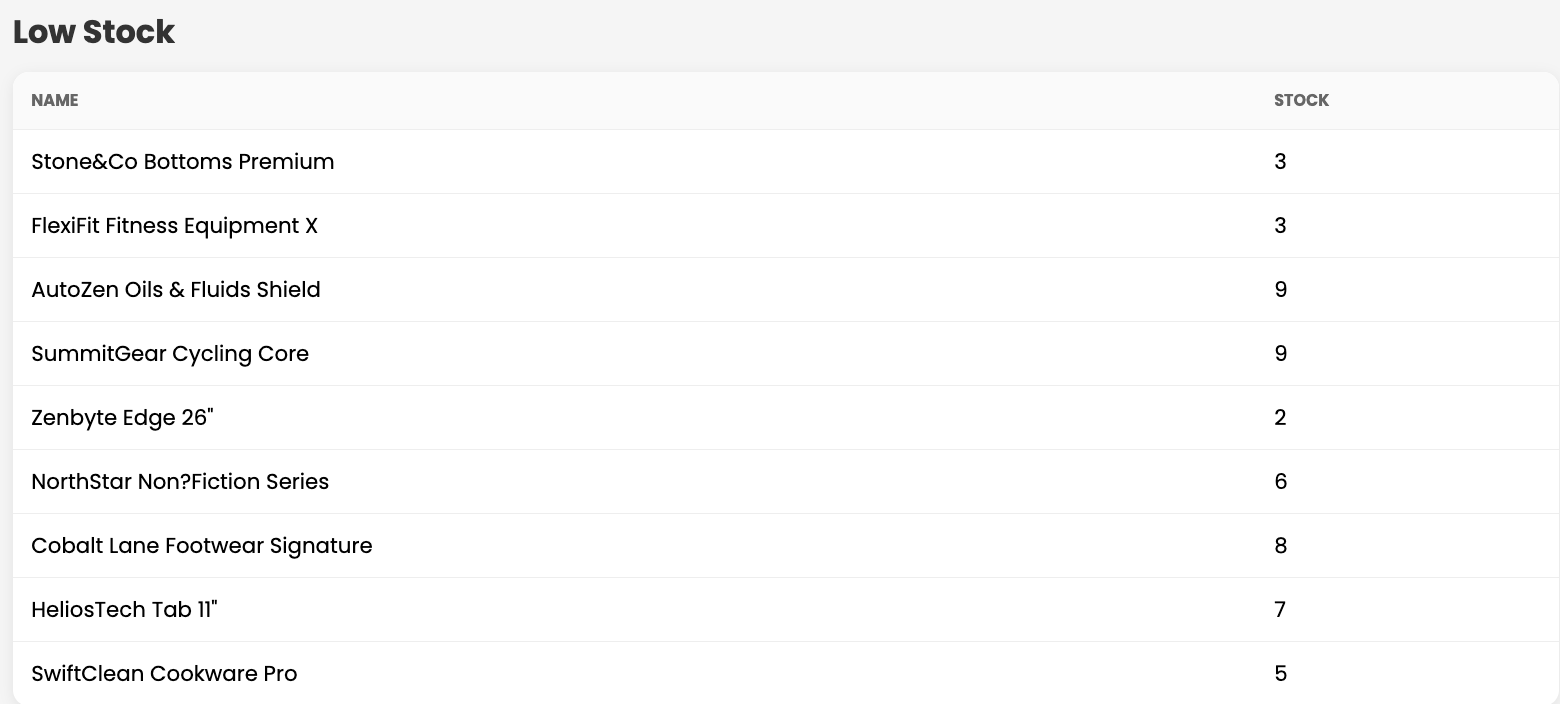
***Image 5:*** *Admin dashboard showing product category breakdown (pie chart), sales trends (line chart), and inventory status with edit/delete options.*

**Admin Product Page**

****

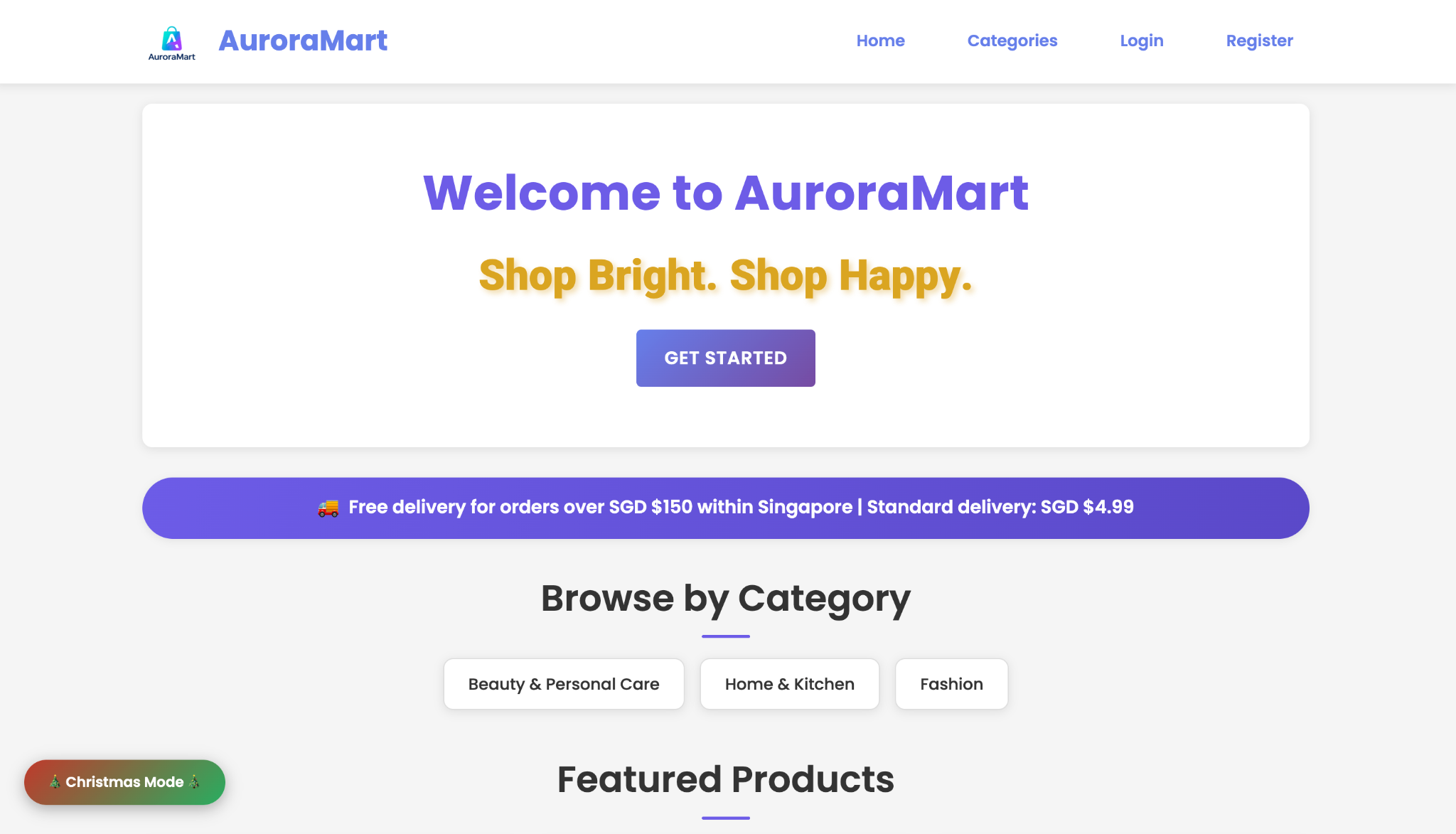
**Image 6:** *Product tab where admins can add new products or update stock levels.*

**Admin Stock Page**

****

**Image 7:** *Product tab where admins can add new products or update stock levels.*

**Storefront Homepage**



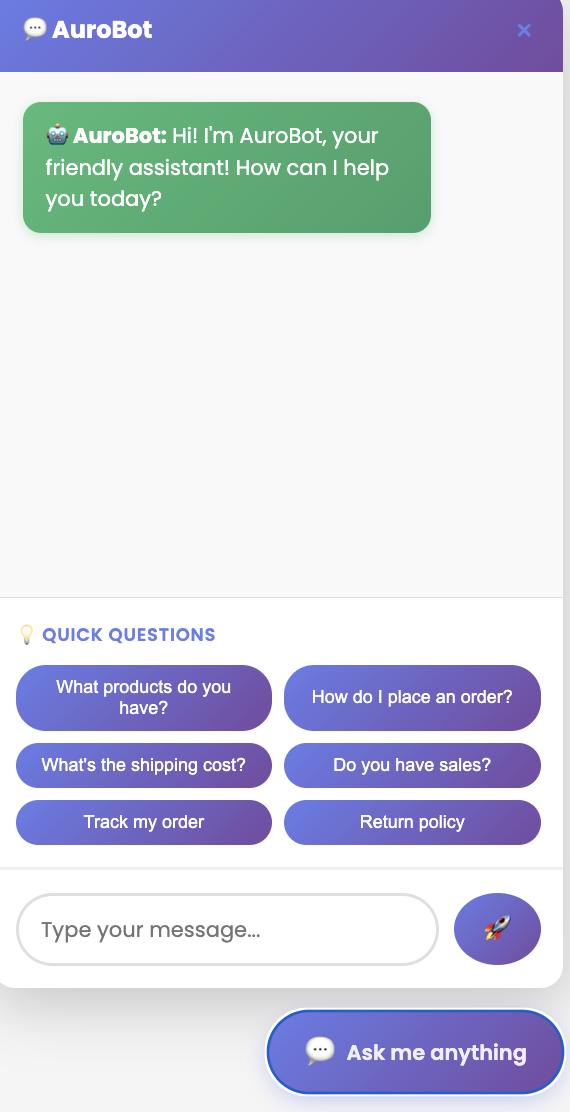
**Image 8:** *Landing page with featured or personalised products.*

**Overview of Featured Products and Aurabot**

### 

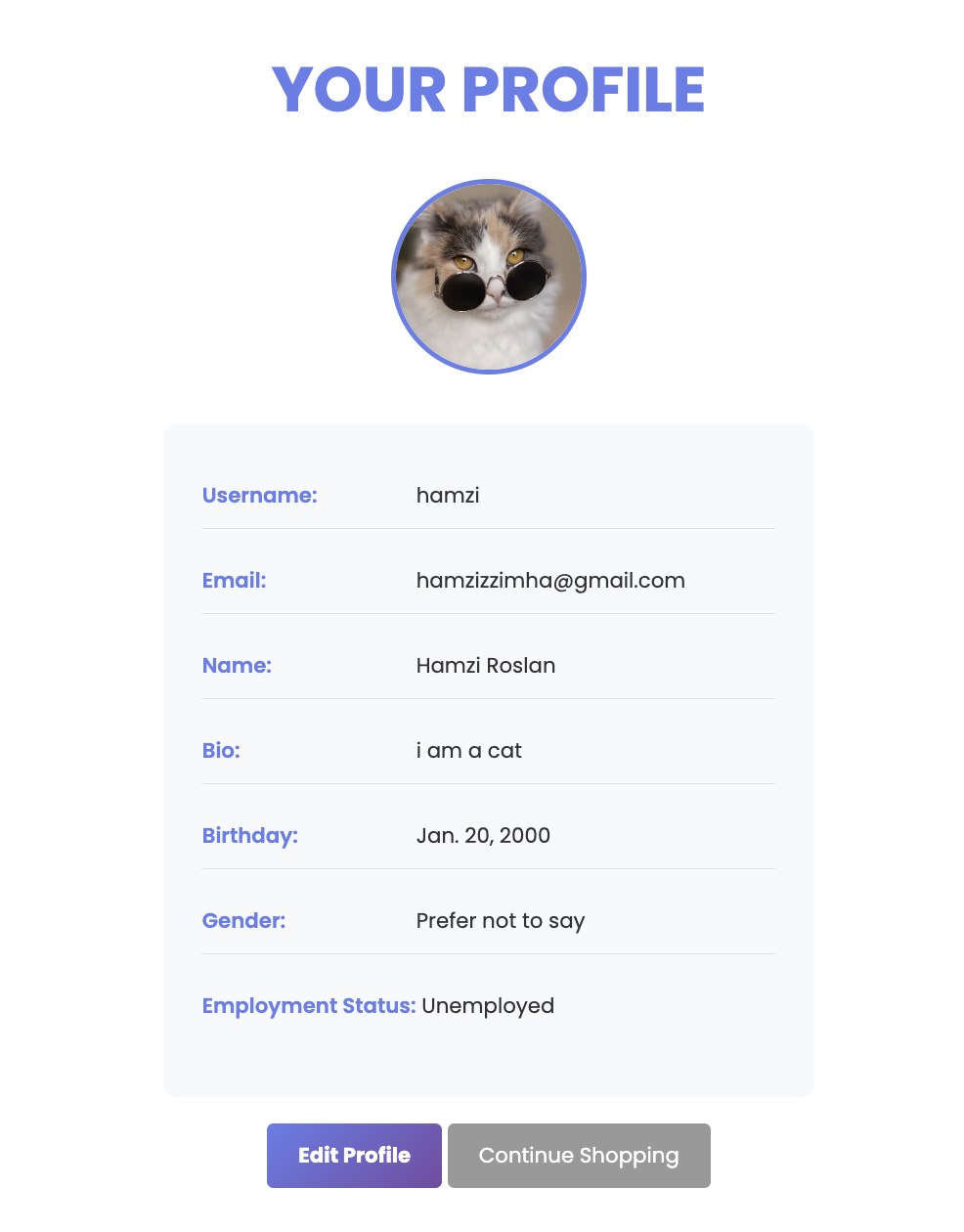
***Image 9:*** *Storefront homepage showing featured products & Aurabot chatbot for assistance.*

**Overview of Featured Products and Aurabot**



***Image 10:*** *Curated recommendations and category tiles for personalised browsing.*

**Profile Page**

****

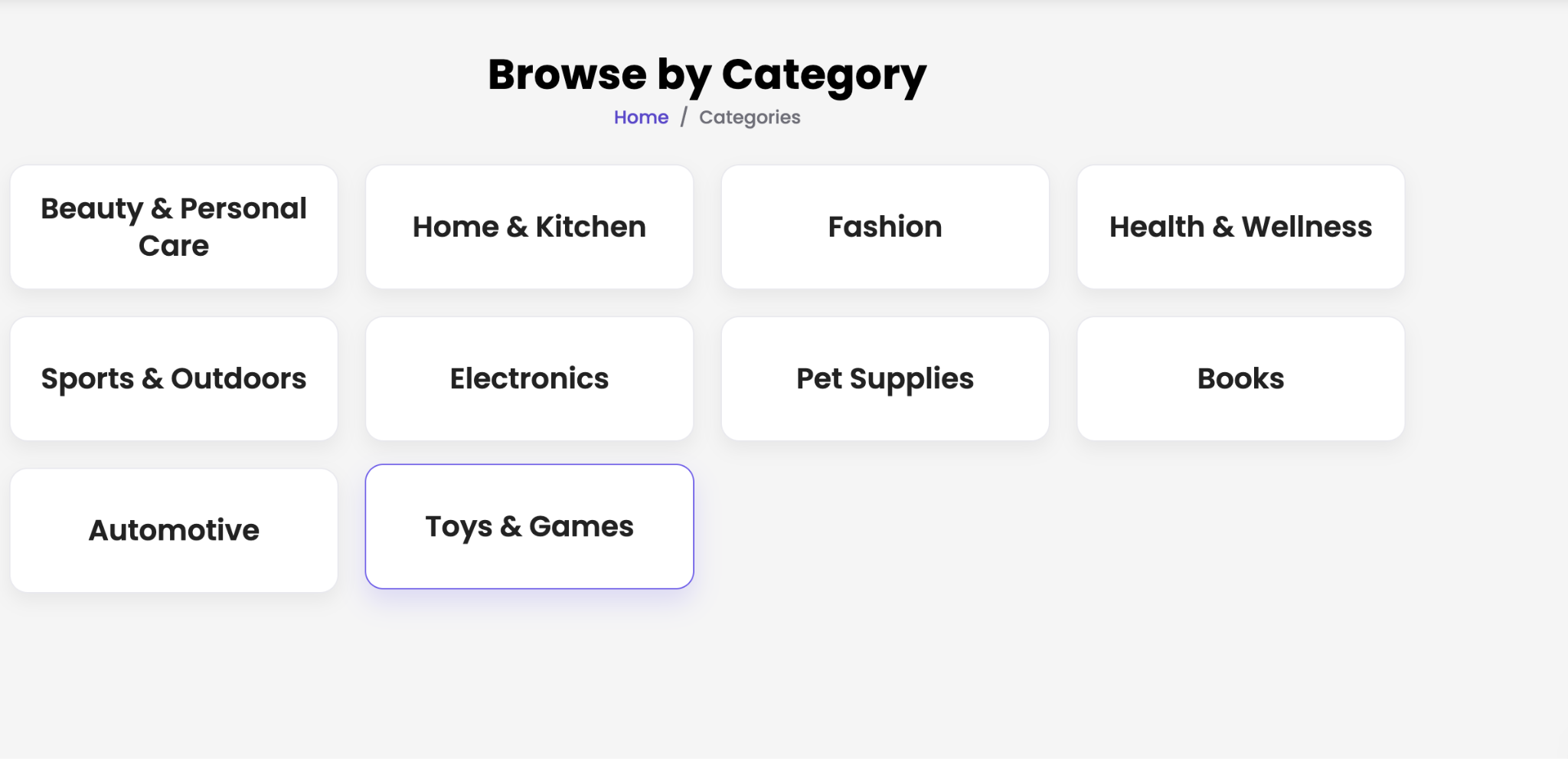
***Image 11:*** *Displays user details and saved preferences.*

**Category Browsing**

## 

***Image 12:*** *Category view for users to explore product types.*

**Categories Page**

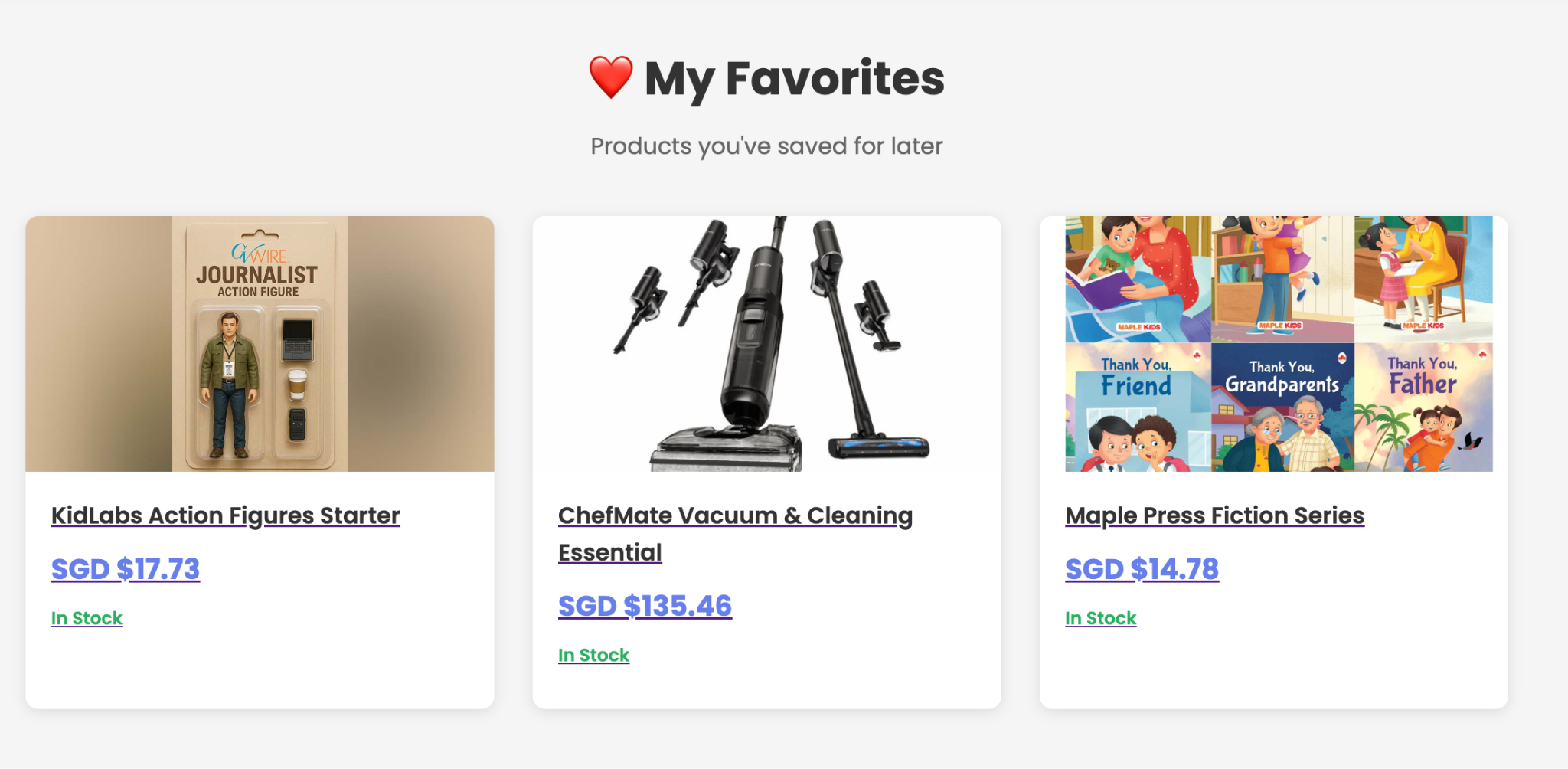


***Image 13:*** *Grid of products under a selected category.*

**Product Page**

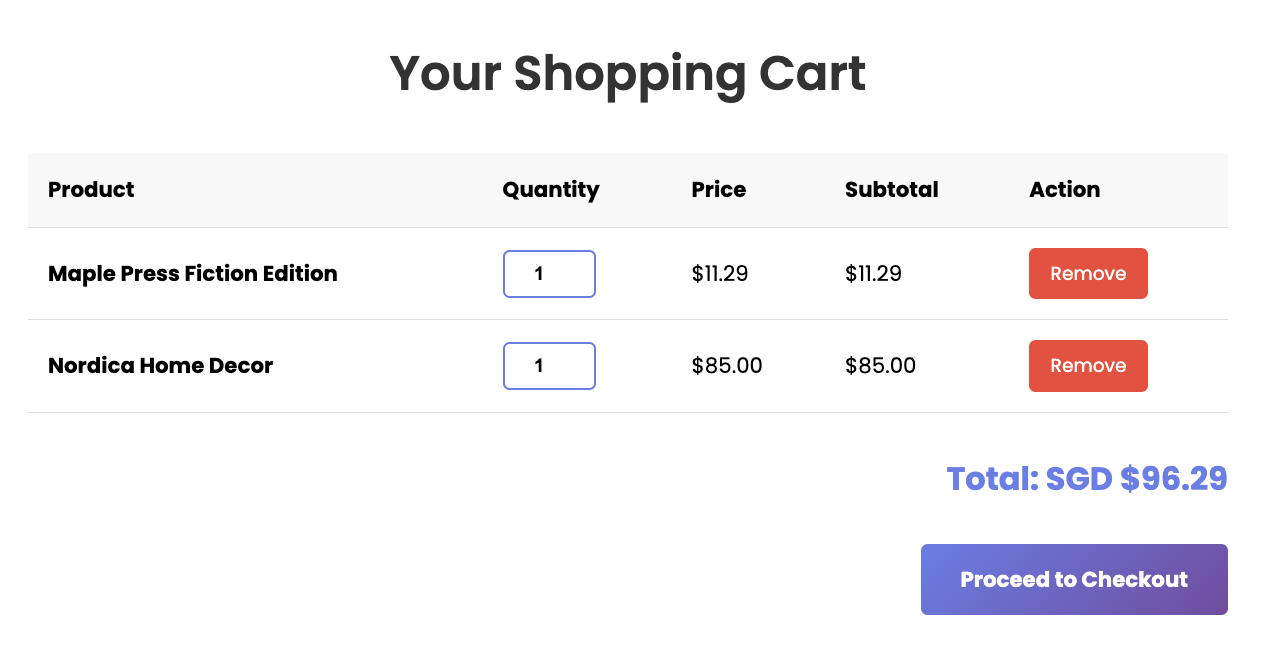
## 

***Image 14:*** *Detailed product info with image, price, and stock.*

**Favourites Page**

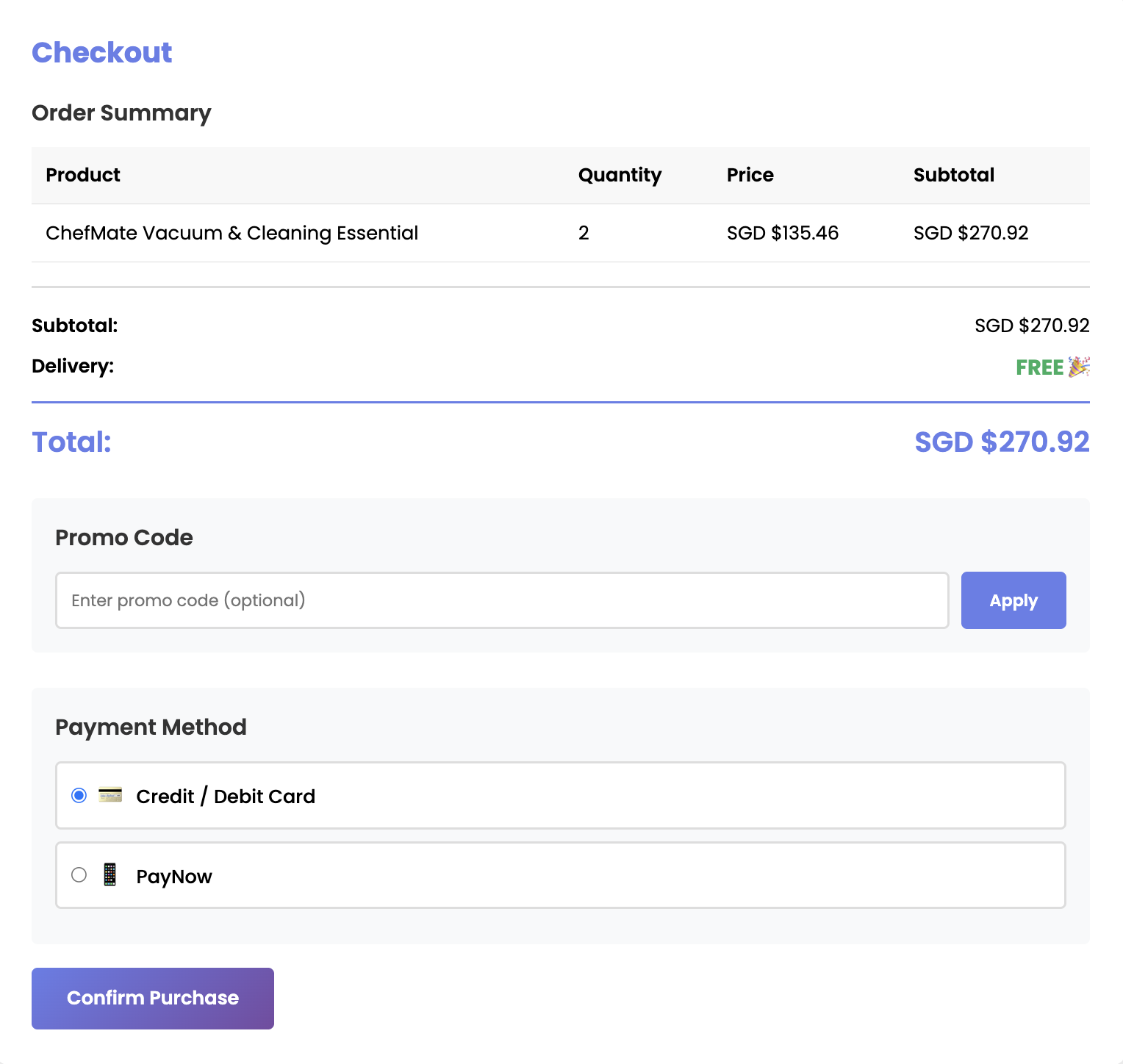
***Image 15:*** *List of saved products for future purchase.*

**Shopping Cart Page**

****

***Image 16:*** *Cart summary with items, quantity, and total.*

**Checkout Page**

****

***Image 17:*** *Checkout form with suggested items (AI-based).*

**Confirmation Page**

## 

***Image 18:*** *Final order summary after successful checkout.*

# 7. Relation Data Model

## 7.1 Product Table (updated for Stock Management)

| **Field Name** | **Data Type** | **Description** |
| --- | --- | --- |
| product\_id | Integer (PK) | Unique identifier for each product |
| name | Text | Name of the product |
| description | Text | Description of the product |
| category | Text | The category the product belongs to |
| price | Decimal | Selling price |
| stock | Integer | **Current quantity available** *(used for Feature 7)* |
| reorder\_threshold | Integer | **Low-stock threshold** to trigger warnings *(Feature 7)* |
| rating | Decimal (optional) | Average rating |
| image\_url | Text (optional) | Link to product image |

### 

### 7.2 Customer Table

| **Field** | **Type** |
| --- | --- |
| customer\_id | Integer (PK) |
| name | Text |
| email | Text |
| age | Integer |
| gender | Text |
| employment\_status | Text |
| income\_range | Text |
| preferred\_category | Text (ML predicted) |

**7.3 Cart (session-based)**

| **Field** | **Type** |
| --- | --- |
| cart\_id | Integer (PK) |
| customer\_id | FK to Customer |
| product\_id | FK to Product |
| quantity | Integer |

# 8. Use of AI Tools

In this project, we used several generative AI tools to support the design and development of AuroraMart. These tools were used as assistants in our workflow and not as direct code generators for the entire system. Human judgment, manual refinement, and debugging were involved at every stage.

1. **ChatGPT**

We used ChatGPT primarily to guide our technical planning and decision-making. It helped us draft user stories, structure our data model, and break down complex features such as AI model integration and session-based cart logic. We also used it to review code snippets for Django and ensure our explanations in the report were clear and professional.

1. **VSCode AI (Copilot)**

GitHub Copilot was used within Visual Studio Code to assist in writing boilerplate code, especially for repetitive components such as forms, views, and template loops. We used it to suggest helper functions and identify potential syntax errors. Final implementations were manually reviewed, tested, and modified to ensure correctness and alignment with project requirements.

1. **Figma AI**

Figma AI was used to accelerate the design of wireframes and UI components. We used AI-generated layout suggestions to experiment with page structures for the storefront and admin panel. These suggestions were then adjusted to match our intended user experience, focusing on usability and consistency.

Overall, these AI tools enhanced our productivity and helped us focus more on the creative and logical aspects of software engineering. All AI-generated content was carefully reviewed and refined to ensure it aligned with the learning objectives of this module.

# 9. Appendices

## 9.1 Overview of Files

IS2108Project/

├── .gitignore ← Git configuration to exclude virtual environments and unnecessary files

├── readme.txt ← Project and group member information, demo video link

└── source/

├── accounts/ ← Manages customer authentication and registration

├── adminpanel/ ← Custom-built admin panel for managing products and stock

├── auroramart/ ← Root Django project configuration (settings, URLs, WSGI)

├── storefront/ ← Main storefront logic and customer shopping flow

├── ml\_models/ ← Pretrained machine learning models (Decision Tree, Association Rules)

├── media/ ← Folder for storing uploaded product images

├── static/ ← Static assets (CSS, JavaScript, images)

└── templates/ ← HTML templates for all views and pages