**Master of Science (IT) SEM – 10**



LJ Institute of Computer Applications

**RetailIQ - Smart Retail Operations & Analytics Platform**

**Guided By:**

**Developed By:**

|  |  |  |
| --- | --- | --- |
| *Sr.* | *Enrollment* | *Name* |
| *1* | *21004500210065* | *Arnold Macwan* |
| *2* | *21004500210094* | *Manthan Parmar* |

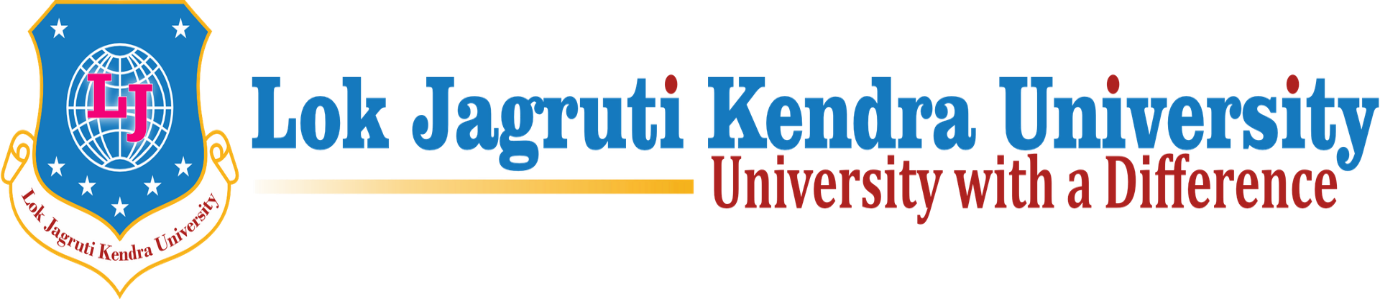
**Submitted To:**

**LJ Institute of Computer Applications**

LJ Campus, Nr. Sanand Cross Road, S.G Road,

Ahmedabad – 382110

Phone: +91 9099063417

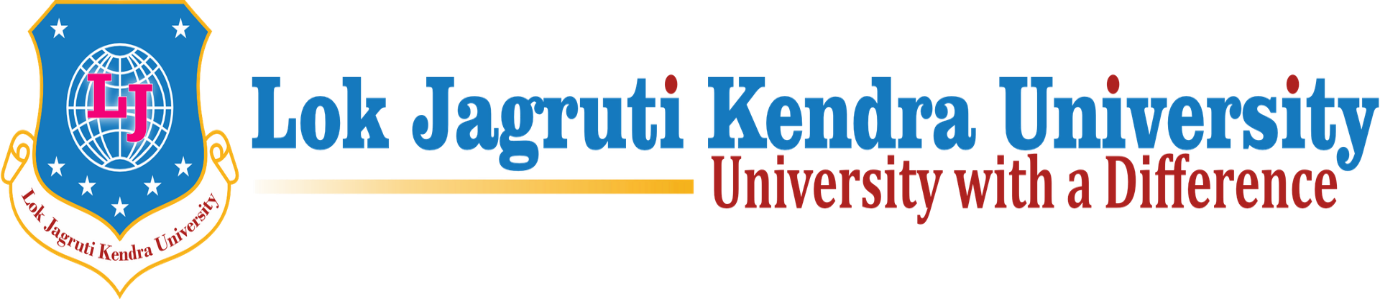


**Certificate**

Enrollment: 21004500210065

This is to certify that **Arnold Macwan** of MSc. IT, Semester 10, Roll no. B28 has satisfactorily completed his project with the title of **“RetailIQ - Smart Retail Operations & Analytics Platform”** under the supervision of guide.

Internal Guide: HOD:



**Certificate**

Enrollment: 21004500210094

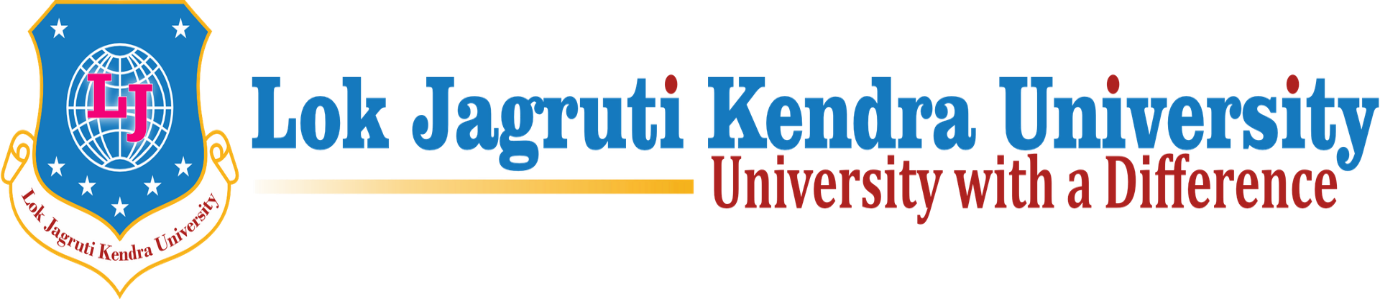
This is to certify that **Manthan Parmar** of MSc. IT, Semester 10, Roll no. B41 has satisfactorily completed his project with the title of **“RetailIQ - Smart Retail Operations & Analytics Platform”** under the supervision of guide.

Internal Guide: HOD:

**Plagiarism Declaration**

To whom so ever it may concern I/We, confirm that Project (document/PPT or Code) is my own work, is not copied from any other person's work (published or unpublished or generated using CHATGPT/AI, and has not previously submitted for assessment either at University or elsewhere. We confirm that we have read and understood the rules regulations on plagiarism in LJ University.

|  |  |  |
| --- | --- | --- |
| *Sr.* | *Enrollment* | *Name* |
| *1* | *21004500210065* | *Arnold Macwan* |
| *2* | *21004500210094* | *Manthan Parmar* |



**Commitment Form**

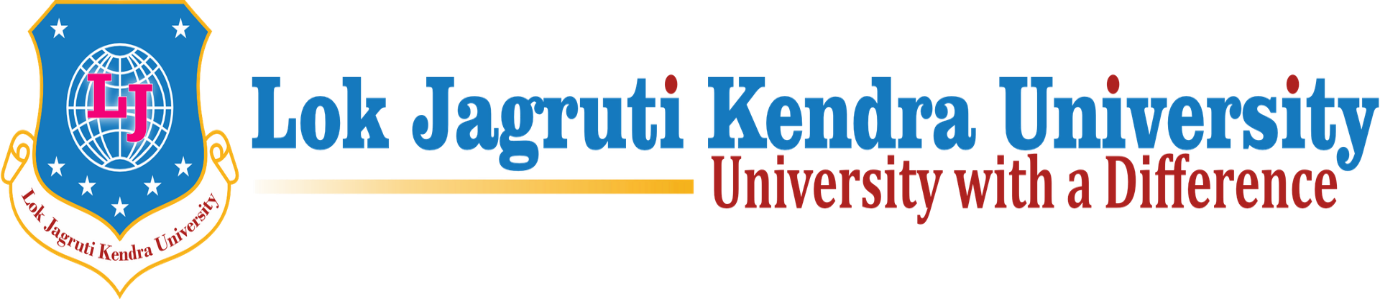
I/we assure that following are the components on which we worked in the said project titled **RetailIQ - Smart Retail Operations & Analytics Platform**. Further we confirm that, we have read and understood the rules and regulations of UFM in LJU.

Student 1.

Enrollment: 21004500210065 Div.: A Roll No.: B28

Name: Arnold Macwan

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | Component | Sr. | Component |
|  | Role based access control |  |  |
|  | Secured Routing & Session management |  |  |
|  | Analytical Dashboard |  |  |
|  | Manage stores |  |  |
|  | Inventory management |  |  |



**Commitment Form**

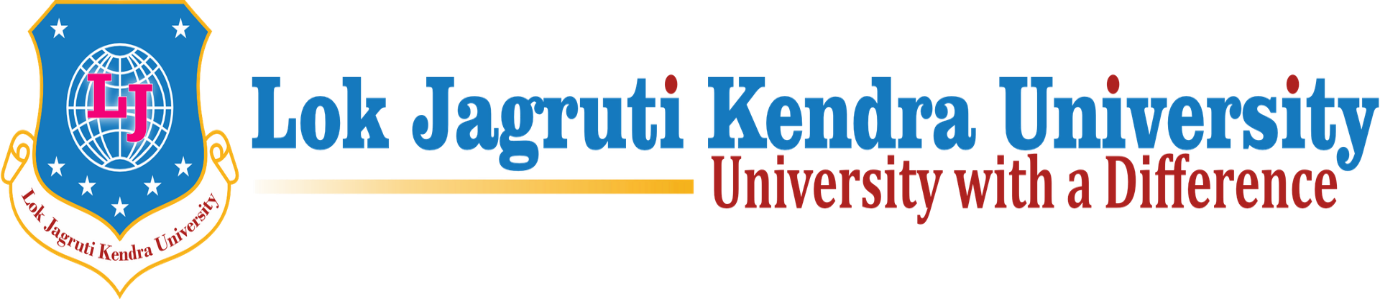
I/we assure that following are the components on which we worked in the said project titled **RetailIQ - Smart Retail Operations & Analytics Platform**. Further we confirm that, we have read and understood the rules and regulations of UFM in LJU.

Student 2.

Enrollment: 21004500210094 Div.: A Roll No.: B41

Name: Manthan Parmar

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | Component | Sr. | Component |
|  | Login - Signup |  |  |
|  | Feedbacks |  |  |
|  | Profile management |  |  |
|  | Supplier Dashboard |  |  |
|  | Product Category management |  |  |



**Acknowledgement**

I would like to take this opportunity to express my sincere gratitude and appreciation to all those who have supported me in the successful completion of my Semester 10 project. First and foremost, I would like to extend my heartfelt thanks to Dr. Jignesh Doshi, Head of the Department, M.Sc. (Information Technology), LJ School of Computer Applications, LJ University, for his continuous encouragement, expert guidance, and invaluable support throughout the duration of this project.

I am extremely thankful to my internal guide, Mr./Ms. Virendra Ingle, for providing insightful feedback, motivation, and guidance at every stage of the project. I would also like to acknowledge the contribution of my external guide, whose expertise and suggestions greatly enhanced the quality of my work.

I sincerely appreciate the efforts of all faculty members and lab instructors who provided assistance and support whenever needed. Lastly, I am grateful to my family and friends for their constant support, motivation, and encouragement throughout.

**Semester 10.**

**Project title:** **RetailIQ - Smart Retail Operations & Analytics Platform**

|  |  |  |
| --- | --- | --- |
| *Sr.* | *Enrollment* | *Name* |
| *1* | *21004500210065* | *Arnold Macwan* |
| *2* | *21004500210094* | *Manthan Parmar* |

|  |  |  |
| --- | --- | --- |
| Sr. | Table of Content | Page No. |
| 1 | Introduction | 09 |
|  | 1.1 Existing system | 09 |
|  | 1.2 Need for new system | 10 |
|  | 1.3 Objective | 10 |
|  | 1.4 Problem Definition | 11 |
|  | 1.5 Scope of project | 11-12 |
|  | 1.6 Project Profile | 12 |
|  | 1.7 Core components | 13 |
|  | 1.8 Assumptions & constraints | 13-14 |
|  | 1.9 Advantages & Limitations | 14 |
| 2 | Requirement determination & analysis | 15 |
|  | 2.1 Requirement determination | 15-16 |
|  | 2.2 Targeted users | 17 |
| 3 | System design | 18 |
|  | 3.1 Use case diagram | 18 |
|  | 3.2 Class diagram | 19 |
|  | 3.3 Activity diagram | 20 |
|  | 3.4 Sequence diagram | 21 |
|  | 3.5 Data dictionary | 22 |
| 4 | Development |  |
|  | 4.1 coding standard |  |
|  | 4.2 Screenshots |  |
|  | 4.3 Implementation of business logic |  |
|  | 4.4 Navigation flow |  |
|  | 4.5 Server-side validation (optional if needed) |  |
|  | 4.6 Client-side validation (optional if needed) |  |
| 5 | Agile documentations |  |
|  | 5.1 Project charter |  |
|  | 5.2 Roadmap/Schedule |  |
|  | 5.3 User story |  |
|  | 5.4 Release plan |  |
|  | 5.5 Sprint backlog |  |
|  | 5.6 Test plan |  |
| 6 | Proposed enhancement |  |
| 7 | Conclusion |  |

1. **Introduction**

The retail sector is undergoing in a massive **digital transformation** driven by the increasing need for efficiency, intelligent decision making and **data-driven operations**. Meanwhile the large organization leverage sophisticated use of AI-powered tools and automation techniques, but the small-medium size enterprises often rely manual processes and operations and traditional **POS – point of sale** systems.

RetailIQ is designed to bridge this technological gap by providing a smart and unified platform which combines all the management and processes with advanced analytics. The system empowers local shops, supermarkets, and fashion stores to manage inventory, track sales, monitor supplier performance, and understand purchase patterns — all in a centralized ecosystem.

* 1. **Existing System**

In the current retail environment, most small and medium-sized stores rely on traditional **Point-of-Sale** (POS) software or manual methods to manage their operations. These existing systems primarily focus on **basic functionalities** such as billing, inventory updates, and transaction recording. While effective for routine tasks, they **lack intelligent decision-support** capabilities. Retailers must manually analyse sales data, estimate future demand, and track stock levels, which often leads to *inaccuracies* and *inefficiencies*.

Overall, the existing system is **transactional** rather than analytical. It records data but does not help retailers interpret it for **strategic decisions**. As a result, retail operations become reactive, inefficient, and less competitive in the modern digital market.

* 1. **Need for new system**

With increasing competition and rapidly changing consumer behavior, retailers require more than just POS software—they need an **intelligent platform** that can transform raw data into actionable insights. This necessity creates a strong demand for a system like RetailIQ, which integrates **conventional** retail management with advanced analytics and AI-driven predictions.

The system is needed to:

* Enhance decision making
* Prevent stock-out
* Understand customer behavior
* Streamline business management
* Reduce human error

The **need for the system** arises from the growing demand for smart, analytical retail operations that can support better decision-making, reduce costs, and increase competitiveness.

* 1. **Objective**

The main objective of RetailIQ is to develop an intelligent, integrated platform that enhances retail operations through advanced data analytics and AI-powered decision support. The specific objectives include:

* Automate & optimize inventory management
* Advanced sales & profit analysis
* Data-driven decision making
* Enhance customer relations
* Improve accuracy
* Scalable and unified platform

Build an AI-driven, analytics-focused system that enhances decisions, optimizes inventory, improves sales strategies, and empowers retailers with insights previously accessible only to large enterprises.

* 1. **Problem statement**

Retail businesses, especially small and mid-sized stores, generate large volumes of transactional data through daily sales, purchases, and inventory movements. However, most retailers lack the tools and expertise to analyze this data effectively. These limitations prevent retailers from operating efficiently, understanding their customers, or forecasting future business needs. Therefore, the problem is the absence of a unified, smart retail management system that not only handles operations but also provides predictive analytics, actionable insights, and intelligent decision support for modern retail environments.

* 1. **Scope of the project**

The scope of RetailIQ encompasses the design, development, implementation, and evaluation of an intelligent retail management and analytics platform aimed at empowering small and mid-sized retail businesses. The project covers functional, non-functional, analytical, and technological aspects as described below.

1. Operational scope
   1. Inventory management
   2. Sales & Billing integration
   3. Supplier & purchase management
2. Analytical scope
   1. Demand forecasting
   2. Market basket analysis
   3. Stock prediction
   4. Interactive dashboards
3. User Scope

The system is intended for:

1. Local shop owners
2. Vendors, mini-markets
3. Retails stores
4. Technical scope
   1. Frontend interface
   2. Backend services
   3. API services
   4. Database systems
   5. Machine learning pipelines
5. Out of scope
   1. Multi-store management
   2. Real-time or barcode hardware systems
   3. Integration with third part e-commerce systems

The project will deliver a functional, scalable platform capable of transforming traditional retail data into meaningful business intelligence.

* 1. **Project profile**

|  |  |
| --- | --- |
| Project Name | RetailIQ |
| Frontend | HTML, CSS, JavaScript, ReactJS |
| Backend | ExpressJS, NodeJS |
| Database | MongoDB, MySQL |
| ML Services | Python, FastAPI |
| Tools | VS Code, MS-Office |

* 1. **Core Components**

|  |  |  |  |
| --- | --- | --- | --- |
| Admin | Supplier | Customer | Store manager |
| * Login * Manage suppliers * Manage stores * Manage permissions * Analytics & monitoring * Pricing & Discounts * Reports * Logout | * Login * Manage purchases * Products & pricing * Communication * Profile management * Order analytics * Logout | * Login * Registration * Profile management * Past purchases * Feedbacks * Orders & Wishlist * Logout | * Login * Profile Management * Inventory management * Sales & Billing * Manage supplies * Sales dashboard * Store level Analytics * Logout |

* 1. **Assumptions & Constraints**
* **Assumptions**
* Uploaded datasets will not contain malicious or corrupted files that break the system.
* Users will provide timely feedback during each iteration for improvements.
* The system will operate in an environment with stable internet connectivity, as it relies on external services/APIs.
* Users will have basic digital literacy to operate the dashboard, upload files, and understand analytical insights.
* Retailers will provide accurate sales, inventory, and purchase data, either through POS exports or manual uploads.
* Other historic data will be used for training model for stock/sales analysis and prediction if retailer does not have sufficient data.
* **Constraints**
* Computational resources (RAM, CPU) may restrict training of large ML models.
* Retailers unfamiliar with technology may require training to fully utilize analytics features.
* Lack of sufficient historical data may reduce forecasting performance.
* System performance may depend on server capabilities (processing time for large datasets).
* Web browser compatibility may limit advanced visualizations.
* The ML model accuracy may vary based on dataset size, seasonal variations, and store type.
  1. **Advantages & Limitations**
* **Advantages**
  + Data-driven decision making
  + Improved inventory management
  + Enhanced profitability
  + Better customer segmentation and understanding
  + Enhanced stock management
* **Limitation** 
  + Accuracy depends on data quality
  + Can’t capture external market factors for model training
  + Limited real-time analytics
  + Lack of back-up service
  + Multilingual support
  + Low maintenance

1. **Requirement determination & Analysis**

Requirement determination and analysis is the process of gathering, studying, and validating the needs of users, stakeholders, and the system itself. This phase ensures that the project’s objectives align with real-world retail operations and that the final system is both technically feasible and user-oriented.

For RetailIQ, requirement analysis focuses on understanding how retailers currently manage operations, what limitations they face, and what features are essential to build an effective, intelligent retail analytics platform.

* 1. **Requirement determination**

Requirement determination and gathering has been done through several methods and techniques such as:

* Interviews
* Observations of existing system
* Case studies
* Surveys/Feedbacks
* Industry research
  1. **Requirement analysis**

After gathering information from different sources, the requirements are analyzed and broken into categories:

1. **Functional requirement**
   1. Inventory management

* Add, Read, Update or Delete products
* Track stock levels automatically
* Identifying slow moving stocks
* Low-High stock alerts
  1. Sales management
* Upload and handle sales data
* Generate reports
* Identify sales trends
* Billing and sales integration
  1. Supplier management
* Register suppliers
* Manage purchase history
* Track supplies
  1. AI-ML module
* Demand forecast
* Market basket analysis
* Stock/sales/profit prediction
  1. Dashboard & Reporting
* Interactive visuals & charts
* KPI insights
  1. User management
* Login, Register authentication
* Role based access control
* Purchases
* Interests

1. **Non-functional requirement**
   1. Performance requirement
   2. Reliability
   3. Usability
   4. Security
   5. Scalability
2. **User requirement** 
   1. Easy product/inventory management
   2. Sales analytics
   3. Recommendation
   4. Reports
   5. Accessible
3. **System requirement**
   1. Standard PC with windows 10+ and stable internet connection
   2. Python installed if needed in local PC
   3. Cloude storage or other storage devices to store sales data
   4. **Targeted Users**

The system primarily targets users who manage retail activities but lack access to advanced analytics or enterprise-level retail solutions.

* Retail shop owners
* Supermarket store managers
* Retail data analyst
* Store managers & operational staff
* Suppliers & vendors (passive users)

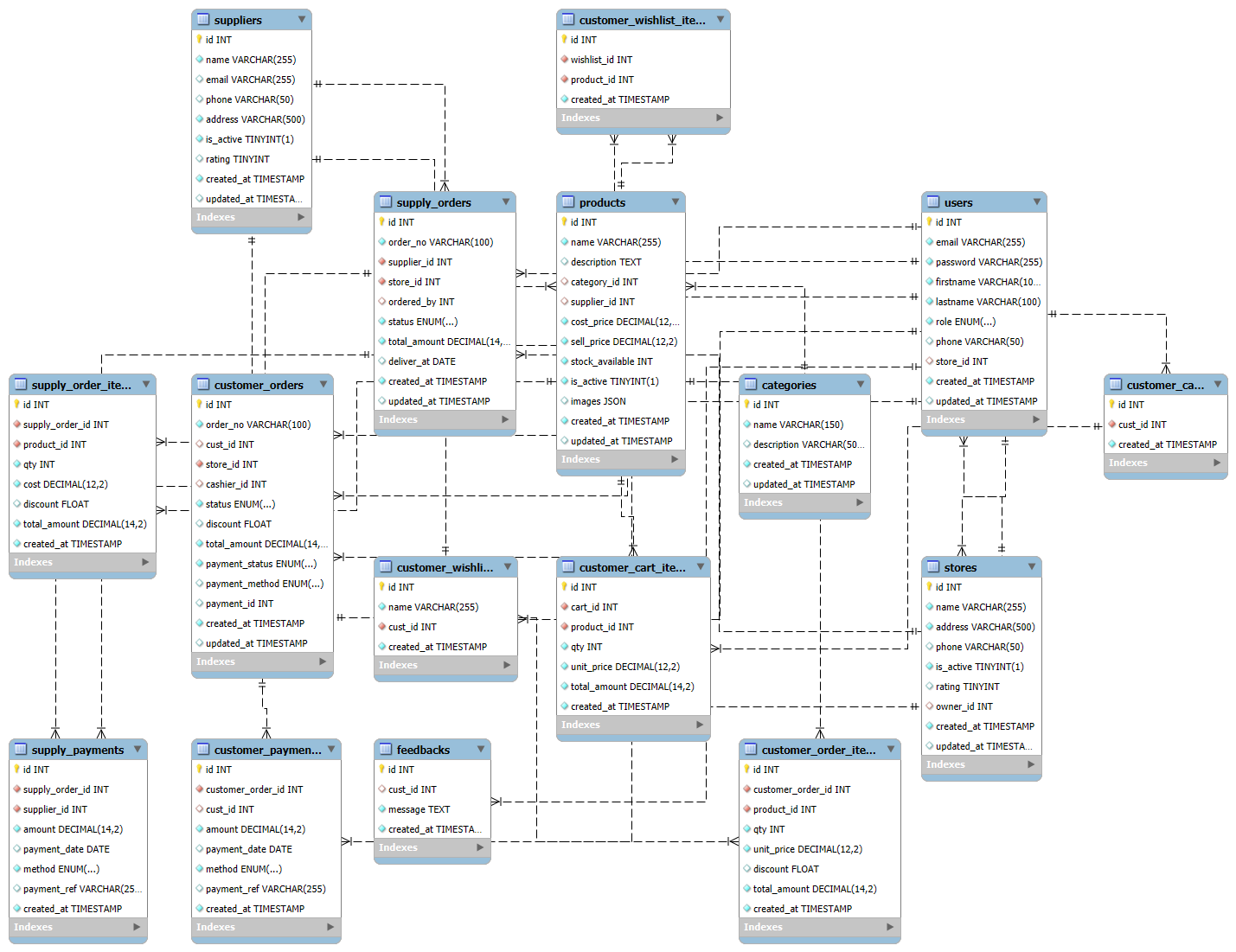
With context of login or role-based users the system allows following users:

* Admin
* Customer
* Store Manager/staff
* Supplier

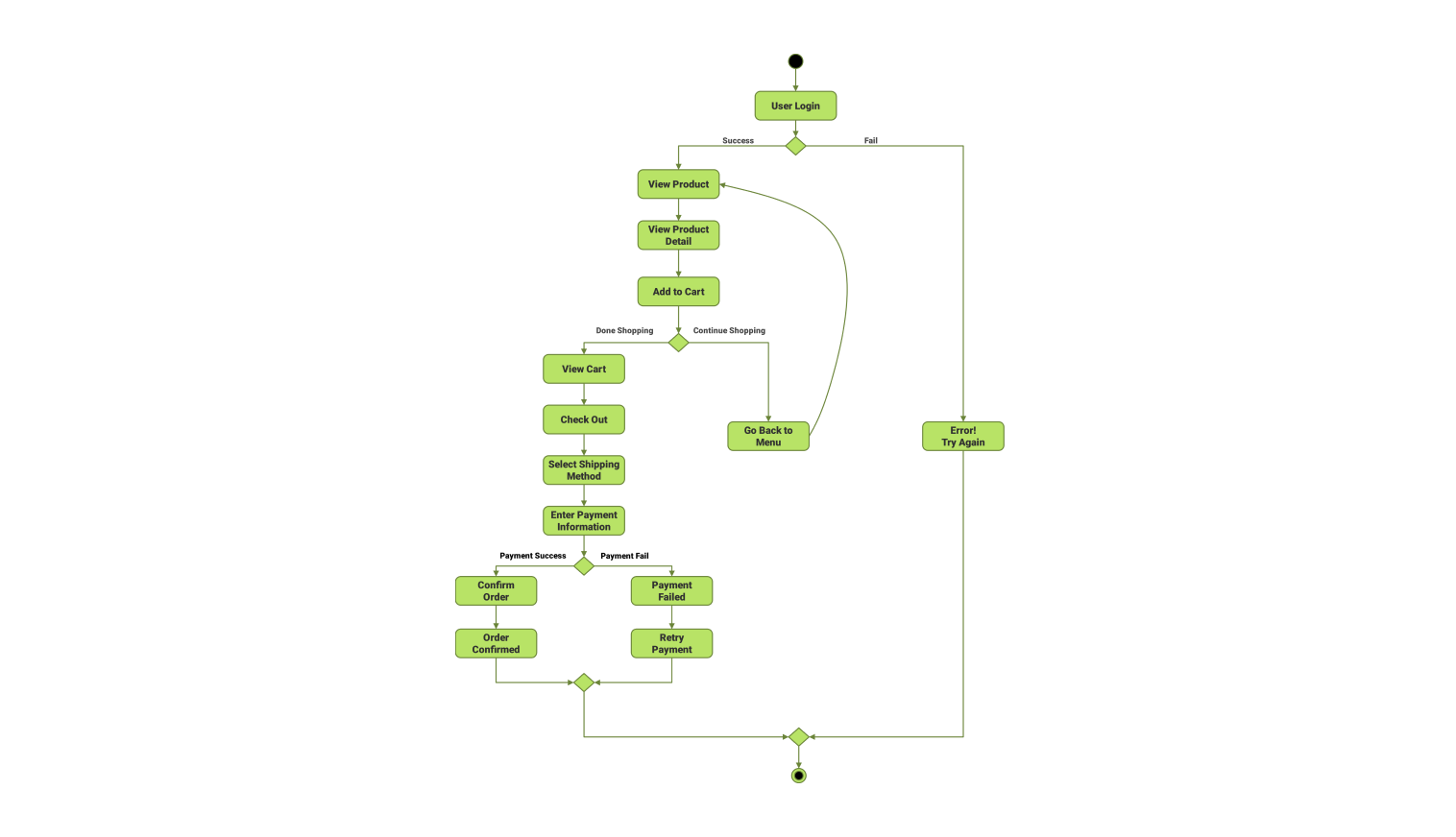
1. **System Design**
   1. **Use Case Diagram**



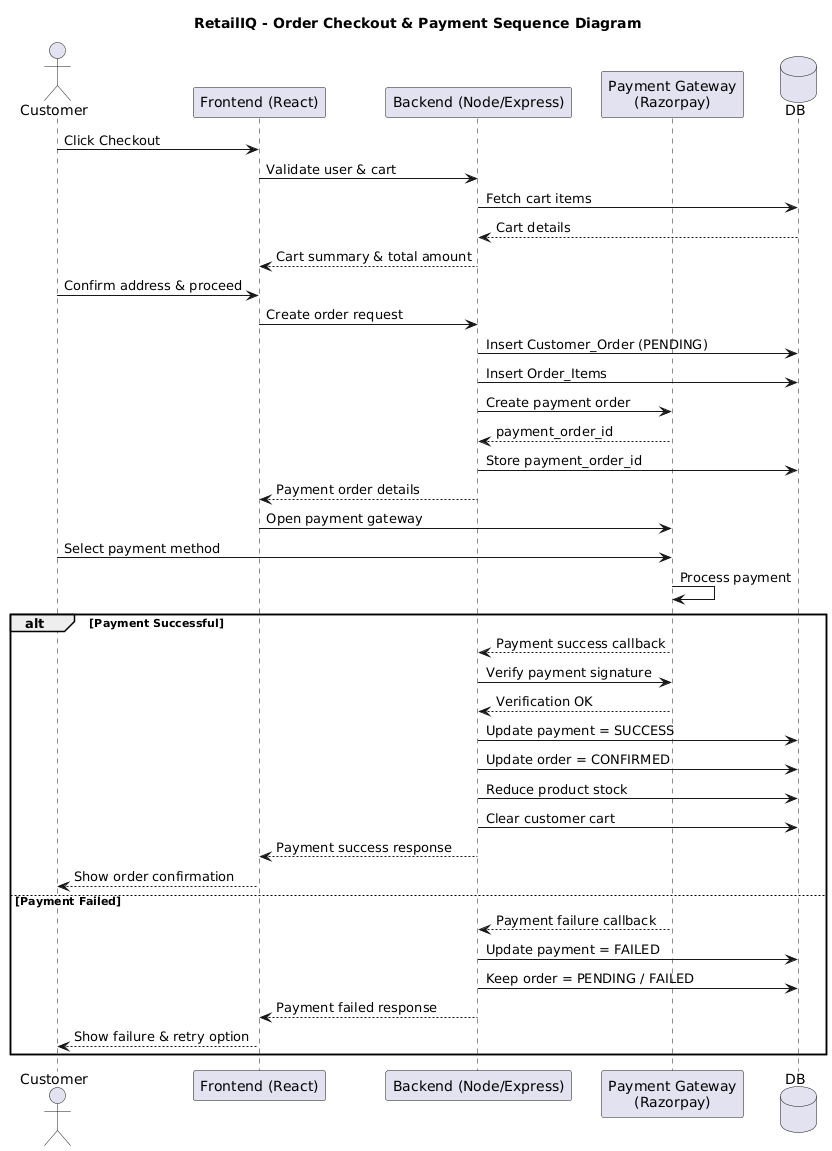
* 1. **Class Diagram**

****

* 1. **Activity Diagram**
* Order checkout and payment activity



* 1. **Sequence Diagram**
* Order checkout and payment activity



* 1. **Data Dictionary**

1. Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Email | Varchar | Unique | [abc@gmail.com](mailto:abc@gmail.com) | Email id of the user |
| Password | Varchar | Not null | 123456 | Hash password |
| Firstname | Varchar | Not null | John | Firstname of the user |
| Lastname | Varchar | Not null | Doe | Lastname of the user |
| Role | Enum (admin, customer, store manager) | Not null | Admin | Role of the user for access |
| Phone | Varchar | Unique | 82\*\*7\*\*2\*8 | Phone number |
| Store\_id | Int | Foreign key (refers store table) | 001 | Store id which belongs to a store manager |
| CreatedAt | Timestamp | Default | 20-02-2025 | Timestamp of record creation |

1. Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Name | Varchar | Unique | Men | Category for the products |
| Description | Varchar | - | All clothes for men | Summary of the category |
| CreatedAt | Timestamp | Default | 21-12-2025 | Date of creation |

1. Products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Name | Varchar | Unique | JoggersX | Name of the product |
| Description | Varchar | - | Skinny fit jogger | Summary of the product |
| CreatedAt | Timestamp | Default | 21-12-2025 | Date of creation |
| CategoryId | Int | Foreign key (refers category table) | 011 | A Category belongs to the product |
| SupplierId | Int | Foreign key (refers supplier table) | 121 | A supplier belongs to product |
| CostPrice | Decimal | Not null | $ 499 | Price of cost from user |
| SellPrice | Decimal | Not null | $ 599 | Price of sale |
| Stock\_available | Int | Not null | 50 | Quantity of stock available |
| isActive | Boolean | Not null | True | Status of product |
| Images | Array | - | abc.jpeg | Images for the products |

1. Suppliers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Name | Varchar | Unique | Torrent supplies Ltd. | Name of the supplier |
| Email | Varchar | Unique | [ts@gmail.com](mailto:ts@gmail.com) | Email of the supplier |
| CreatedAt | Timestamp | Default | 21-12-2025 | Date of creation |
| Phone | Varchar | Unique | 7820\*\*41\*6 | Phone/contact number |
| Address | Varchar | Not null | Avenue complex, NYC, USA | Address of the supplier |
| isActive | Boolean | Not null | False | Status of the supplier |
| Rating | Enum (1-5) | - | 4 | Rating for supplier |

1. Stores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Name | Varchar | Unique | Fista | Name of the store |
| Address | Varchar | Not null | Siteni complex, NYC, USA | Address of the store |
| CreatedAt | Timestamp | Default | 21-12-2025 | Date of creation |
| Phone | Varchar | Unique | 7820\*\*41\*6 | Phone/contact number |
| isActive | Boolean | Not null | False | Status of the supplier |
| Rating | Enum (1-5) | - | 4 | Rating for supplier |
| ownerId | Int | Foreign key (refers user table) | 0131 | A user/owner belongs to store |

1. Supply\_Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| OrderNo | String | Unique | PO-52045 | Unique identifier for order |
| supplierId | Int | Foreign key (refers suppliers table) | 125 | Supplier belongs to supply order |
| storeId | Int | Foreign key (refers store table) | 111 | Store from order belongs |
| orderedBy | Int | Foreign key (refers user table) | 101 | User belongs to order |
| Status | Enum | Not null | Sent | Status of the order |
| TotalAmount | Decimal | Not null | $ 4500 | Total amount for the order |
| deliverAt | Date | Not null | 12-05-26 | Date of delivery |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Supply\_Orders\_items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Supply\_orderId | Int | Foreign key (refers supply\_orders) | 500 | Supply Items belongs to supply |
| productid | Int | Foreign key (refers products) | 511 | Product belongs order |
| Qty | Int | Not null | 50 | Quantity of product |
| Cost | decimal | Not null | $ 50 | Price per unit of the product |
| Discount | Float | - | 5.5 | Discount if any |
| Total\_amount | Decimal | Not null | $ 2500 | Total amount for order |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Supply\_Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Supply\_orderId | Int | Foreign key (refers supply\_orders) | 500 | Supply Items belongs to supply |
| supplierId | Int | Foreign key (refers supply\_orders) | 111 | Supplier belongs to order |
| Amount | Decimal | Not null | $ 32500 | Amount of payment |
| Payment\_date | Date | Default | 12-05-26 | Date of payment |
| Method | Enum | Not null | IMPS | Transaction method |
| paymentRef | String | Not null | PayId: 56512 | Reference of payment |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| orderNo | String | Unique | CO-52146 | Unique no. of order |
| custId | Int | Foreign key (refers user table) | 111 | User belongs to order |
| storeId | Int | Foreign key (refers store table) | 233 | Store belongs order |
| Cashier | Int | Foreign key (refers user table) | 166 | Order created by |
| Status | Enum | Not null | Pending | Status of the order |
| Discount | Float | - | 5.56 | Discount if any |
| createdAt | Datetime | Default | 11-04-26 | Record creation |
| totalAmout | Decimal | Not null | $ 4500 | Total amount of the order |
| Payment\_status | Enum | Not null | Pending | Status of the payment |
| paymentMethod | Enum | Not null | Pending | Method of payment |
| paymentId | Int | Foreign key (refers customer\_Payment) | 52242 | Payment reference |

1. Customer\_Orders\_items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Cust\_orderId | Int | Foreign key (refers customer\_orders) | 500 | order Items belongs to customer |
| productid | Int | Foreign key (refers products) | 511 | Product belongs order |
| Qty | Int | Not null | 50 | Quantity of product |
| Cost | decimal | Not null | $ 50 | Price per unit of the product |
| Discount | Float | - | 5.5 | Discount if any |
| Total\_amount | Decimal | Not null | $ 2500 | Total amount for order |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| customer\_orderId | Int | Foreign key (refers customer\_orders) | 500 | order Items belongs to customer order |
| custId | Int | Foreign key (refers user) | 111 | customerbelongs to order |
| Amount | Decimal | Not null | $ 32500 | Amount of payment |
| Payment\_date | Date | Default | 12-05-26 | Date of payment |
| Method | Enum | Not null | IMPS | Transaction method |
| paymentRef | String | Not null | PayId: 56512 | Reference of payment |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_wishlist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| Name | String | Not null | My wishlist | Wishlist name |
| custId | Int | Foreign key (refers user) | 111 | customerbelongs to order |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_wishlist\_items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| wishlistId | Int | Foreign key (refers wishlist) | 699 | Wishlist reference |
| productid | Int | Foreign key (refers product) | 155 | Product belongs wishlist |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_cart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| custId | Int | Foreign key (refers users) | 121 | User belongs cart |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. Customer\_cart\_items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| cartId | Int | Foreign key (refers customer cart) | 566 | Cart items belong to cart |
| productid | Int | Foreign key (refers product) | 155 | Product belongs cart |
| createdAt | Datetime | Default | 11-04-26 | Record creation |
| Qty | Int | Not null | 5 | Quantity of product |
| Unit\_price | Decimal | Not null | $ 25 | Price per product |
| Total\_amount | Decimal | Not null | $ 125 | Total amount to be paid |

1. Feedbacks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Constraints | Example | description |
| Id | Int | Primary key | 001 | Table identifier |
| custId | Int | Foreign key (refers user) | 121 | User belongs feedback |
| Message | Text | Not null | “Good” | Message for feedback |
| createdAt | Datetime | Default | 11-04-26 | Record creation |

1. **Development**
   1. **Coding standard**

* *Project structure*

Use clear and descriptive names for the whole directory of the project and also recommend the naming conventions rules for the variables and functions names.

* *Code version management*

Use a version control system (i.e., GitHub) for the project to collaborate effectively.

* *UI Design*

Design a user-friendly interface so that user can use the system and understand it easily

* *Error Handling*

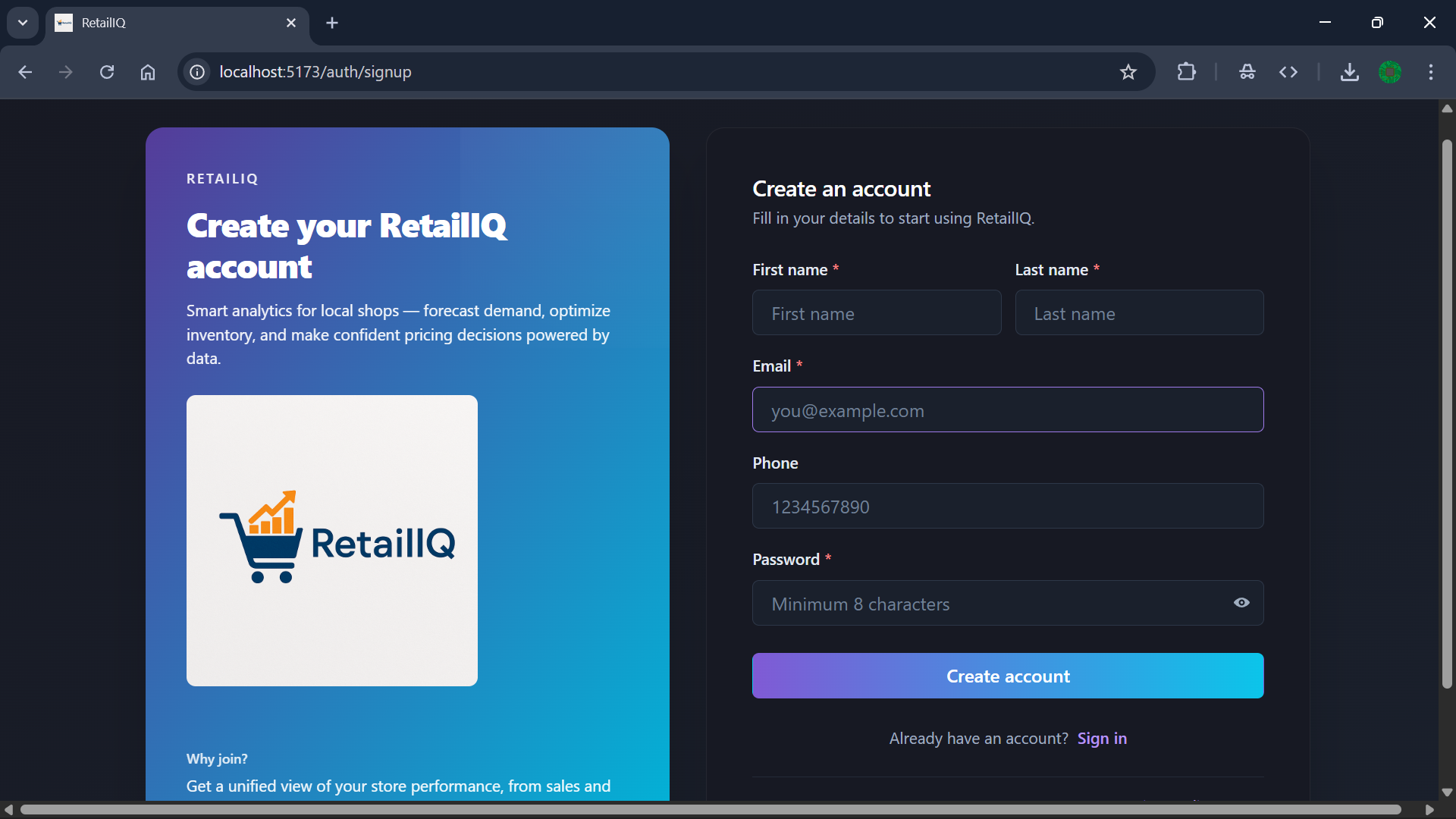
Implement an error free code and use error handling mechanism for user interactions

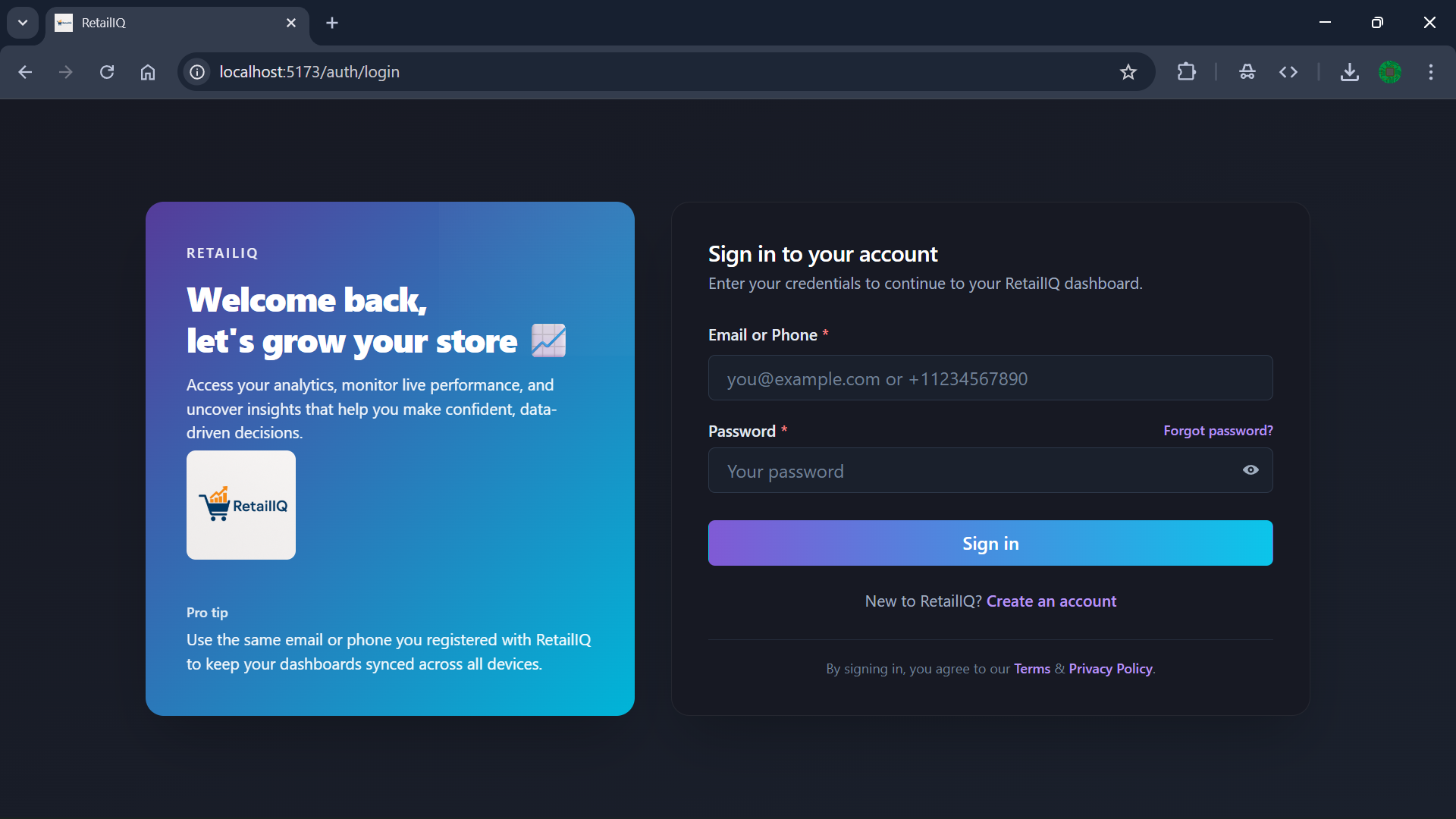
* *Further development*

Maintain the system as well conduct regular reviews that genuine feedback for the system.

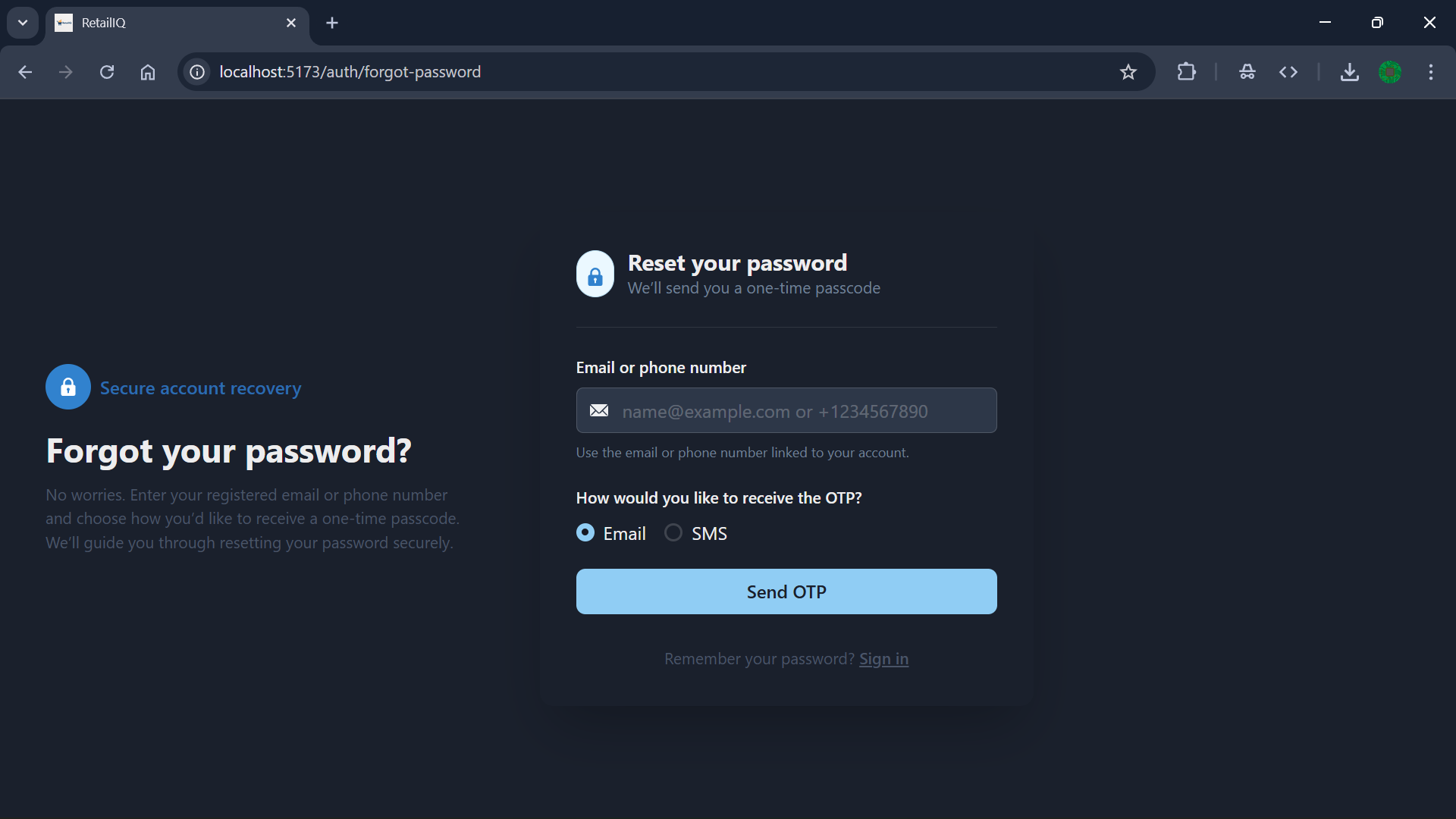
Always look for the further improvement of the system that can make the system work more accurate and reliable

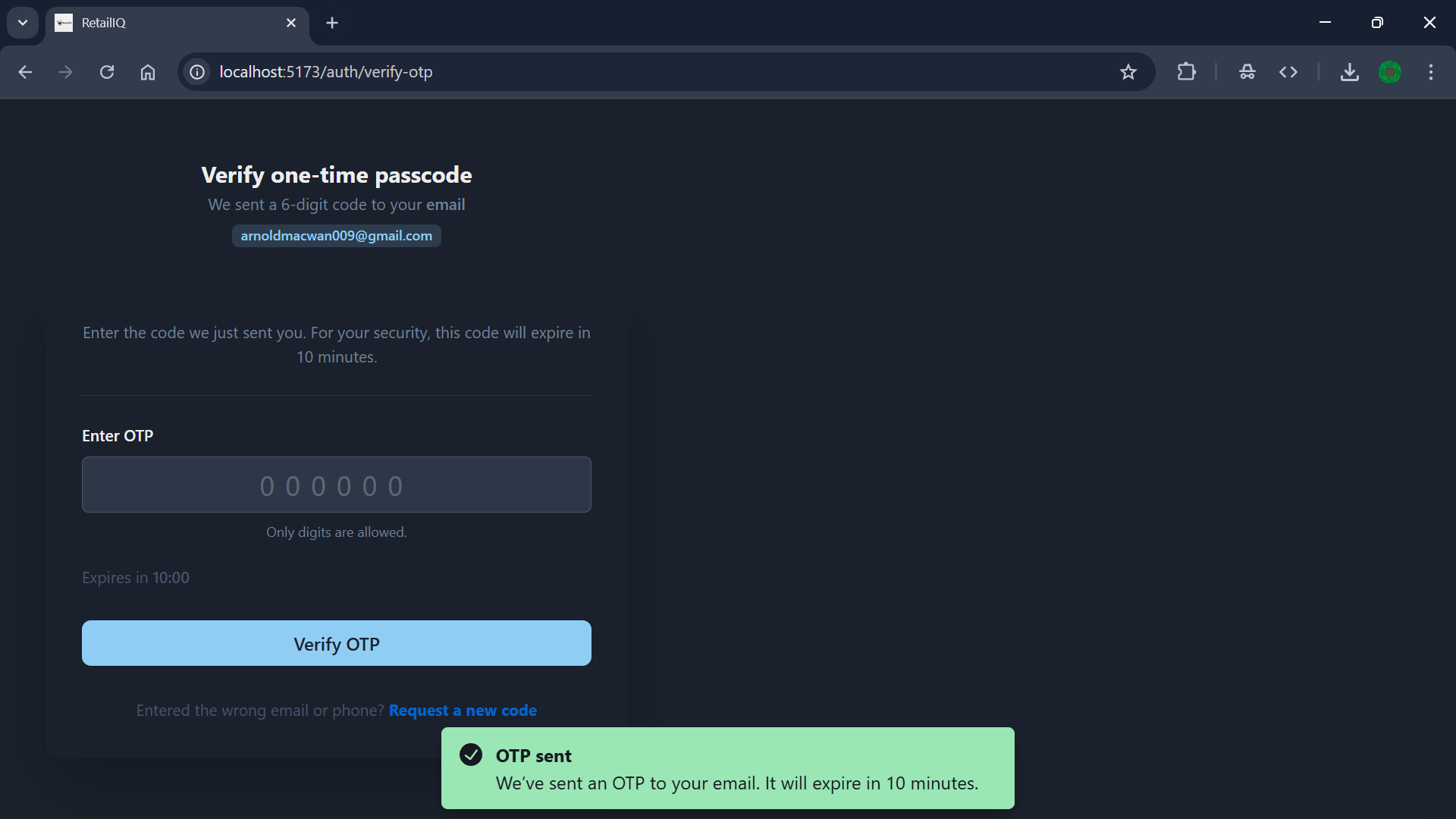
* 1. **Screenshots**
* Login and signup pages to access and join the system



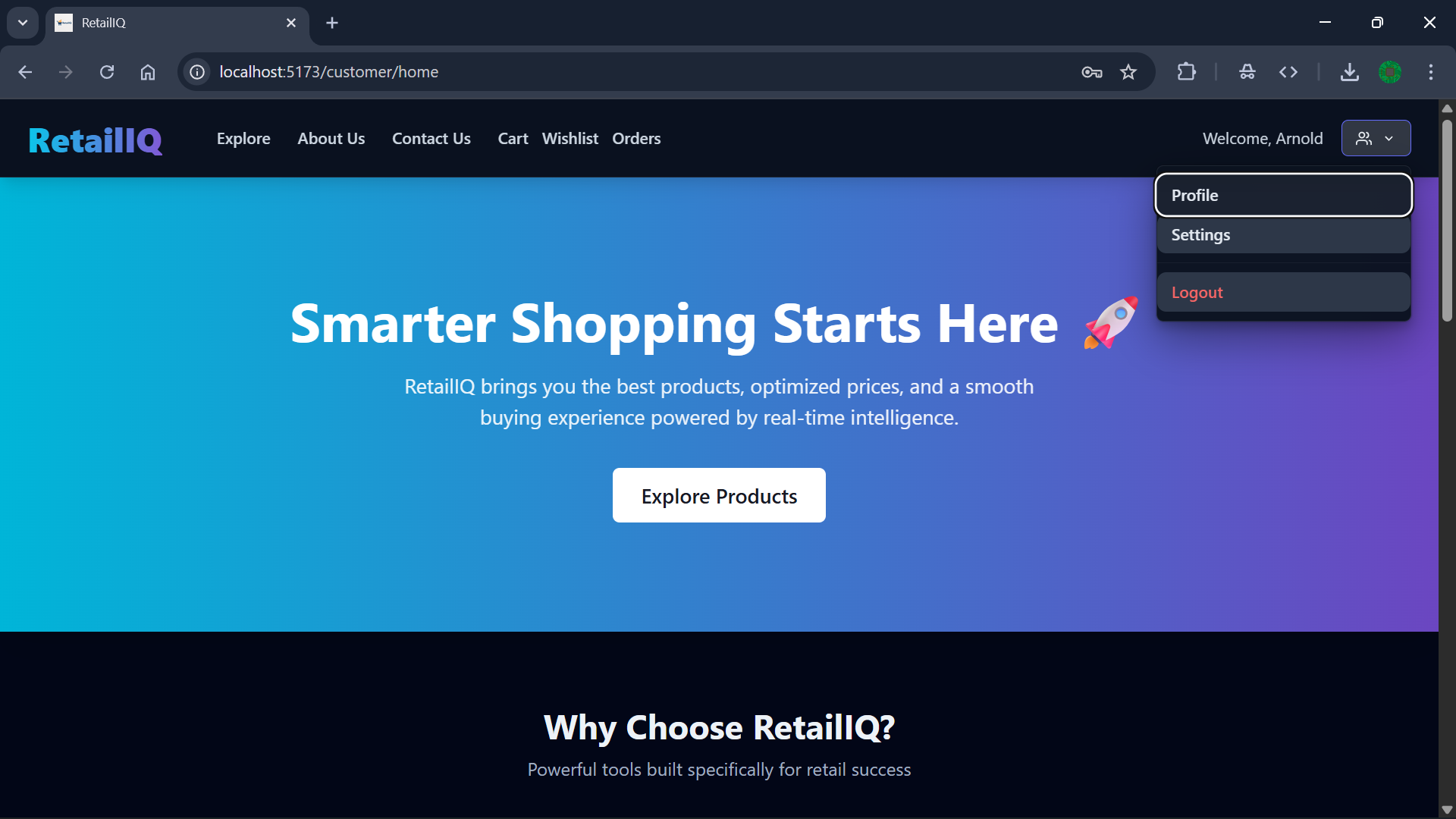


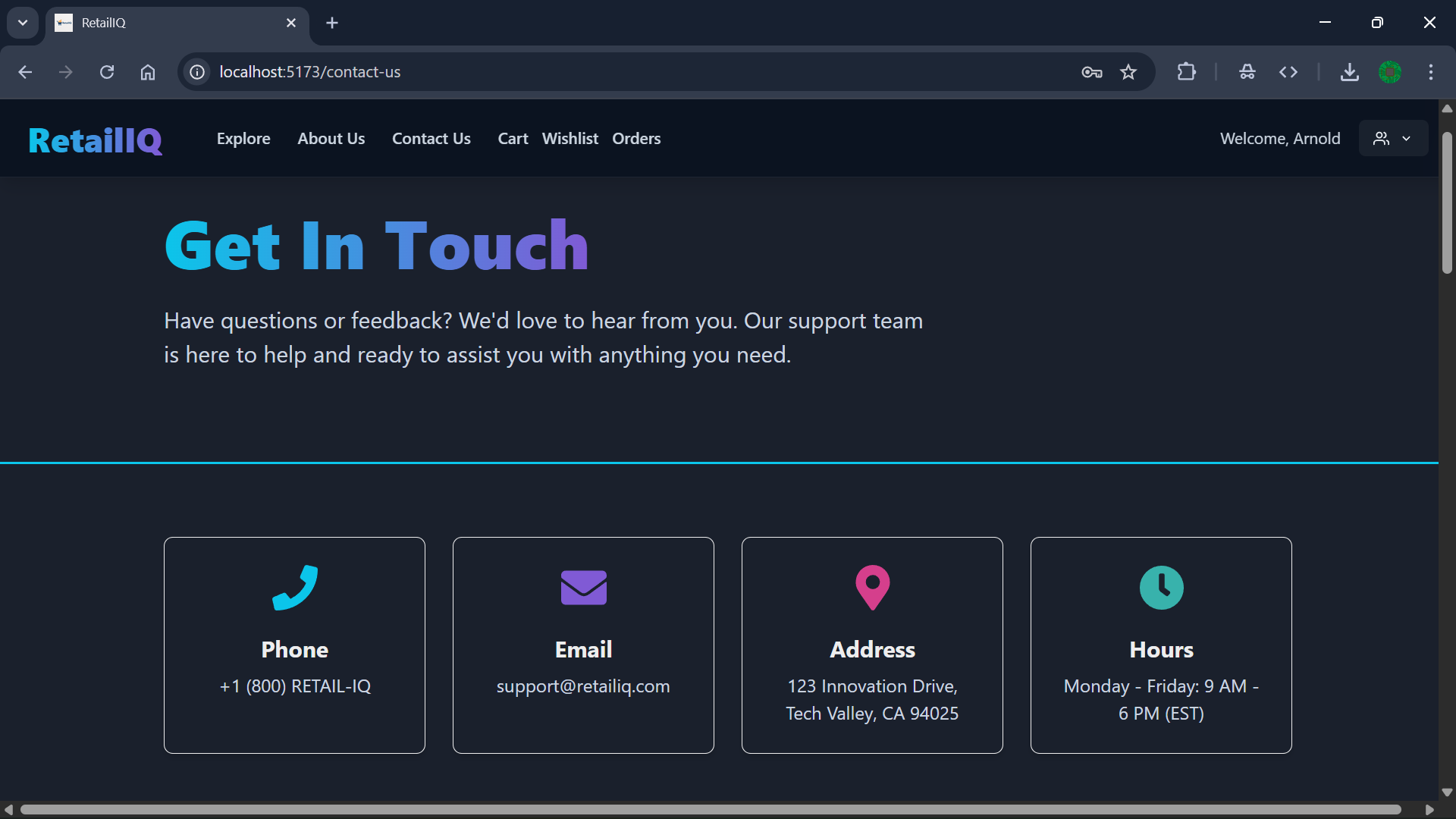
* Password reset with the third-party API/OTP verification

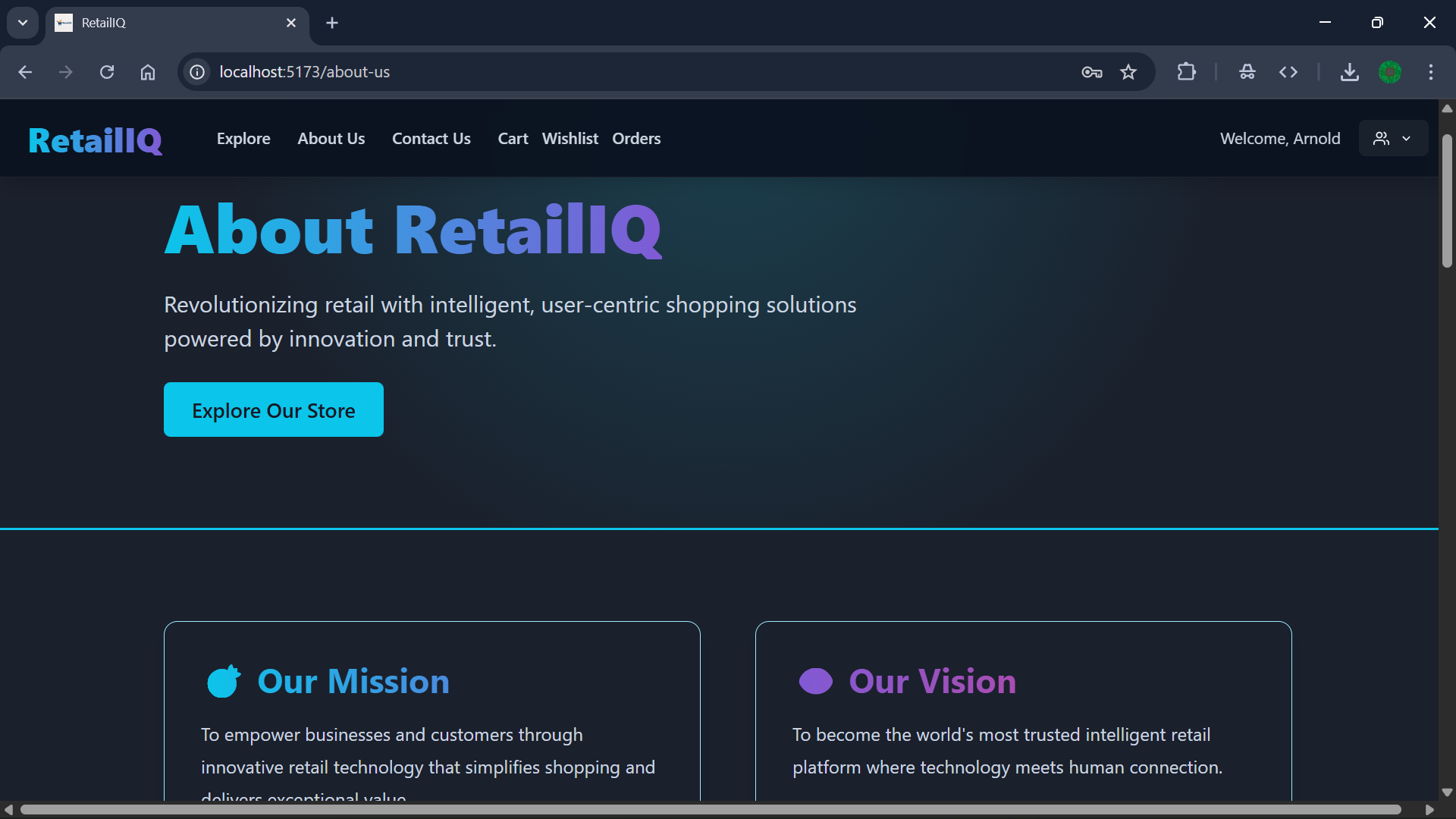




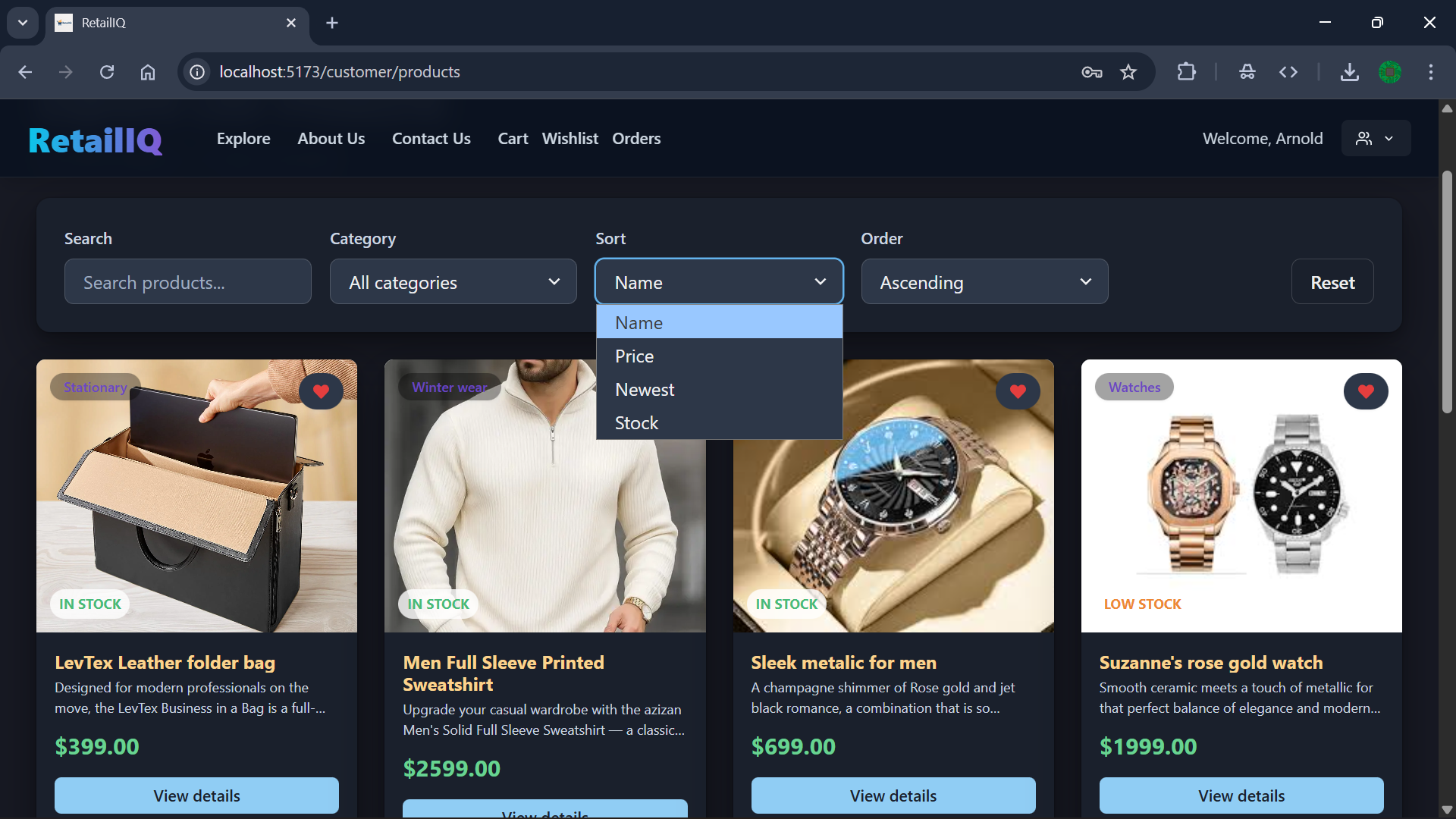
* Customer home pages and routes



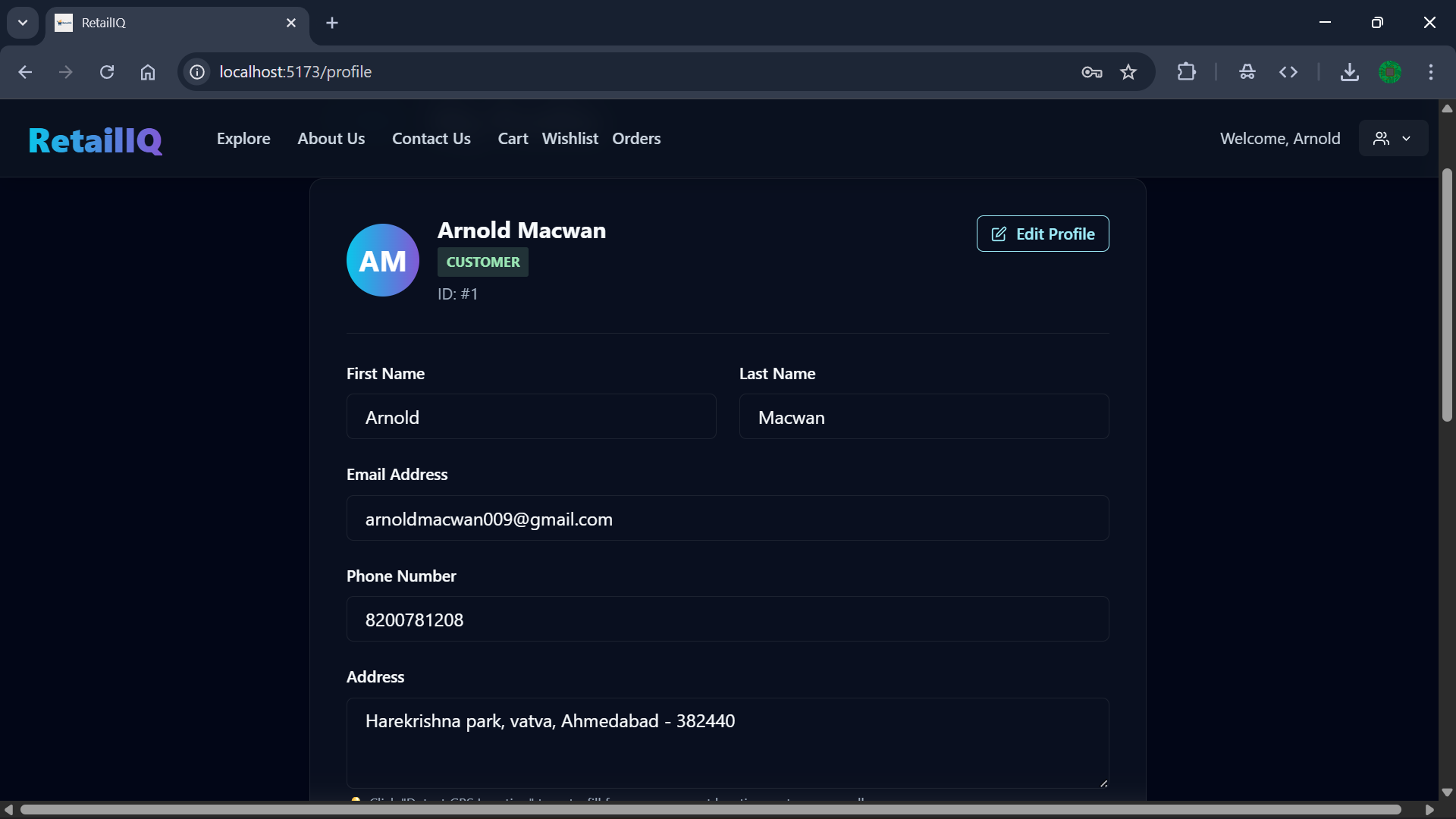




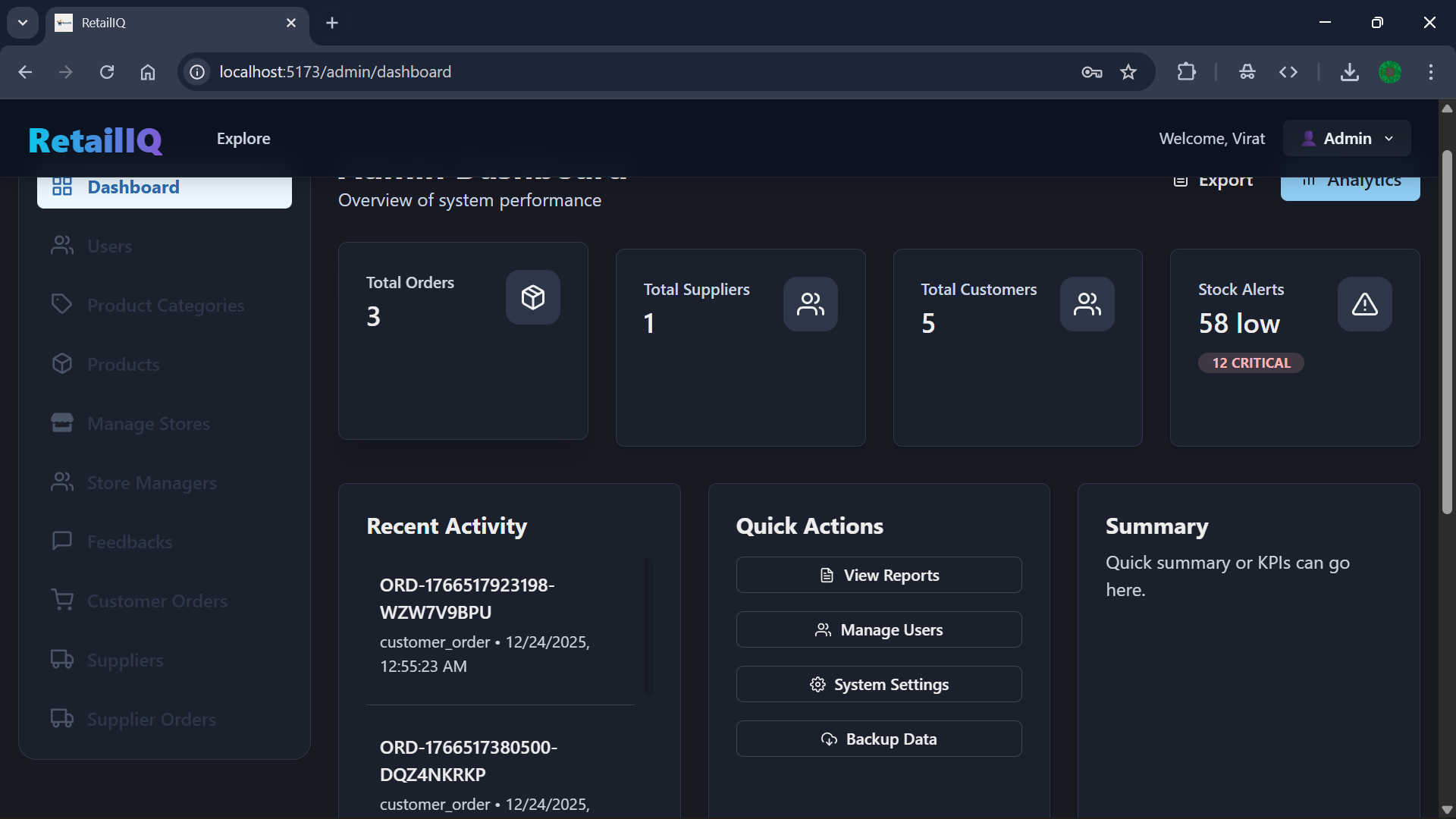
* Product catalog with filter and sort features



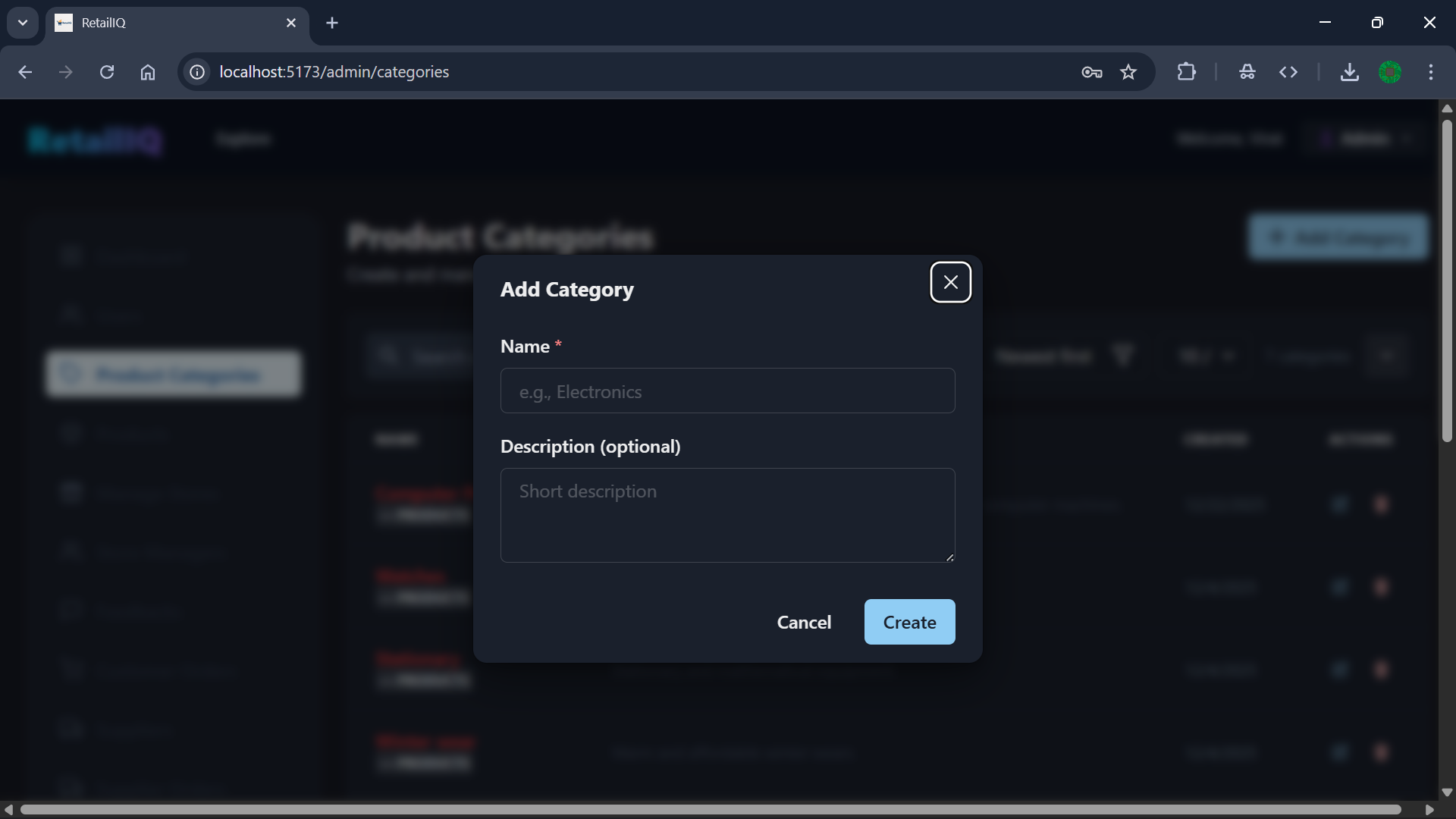
* Profile management (almost same for all) where user can edit their profile

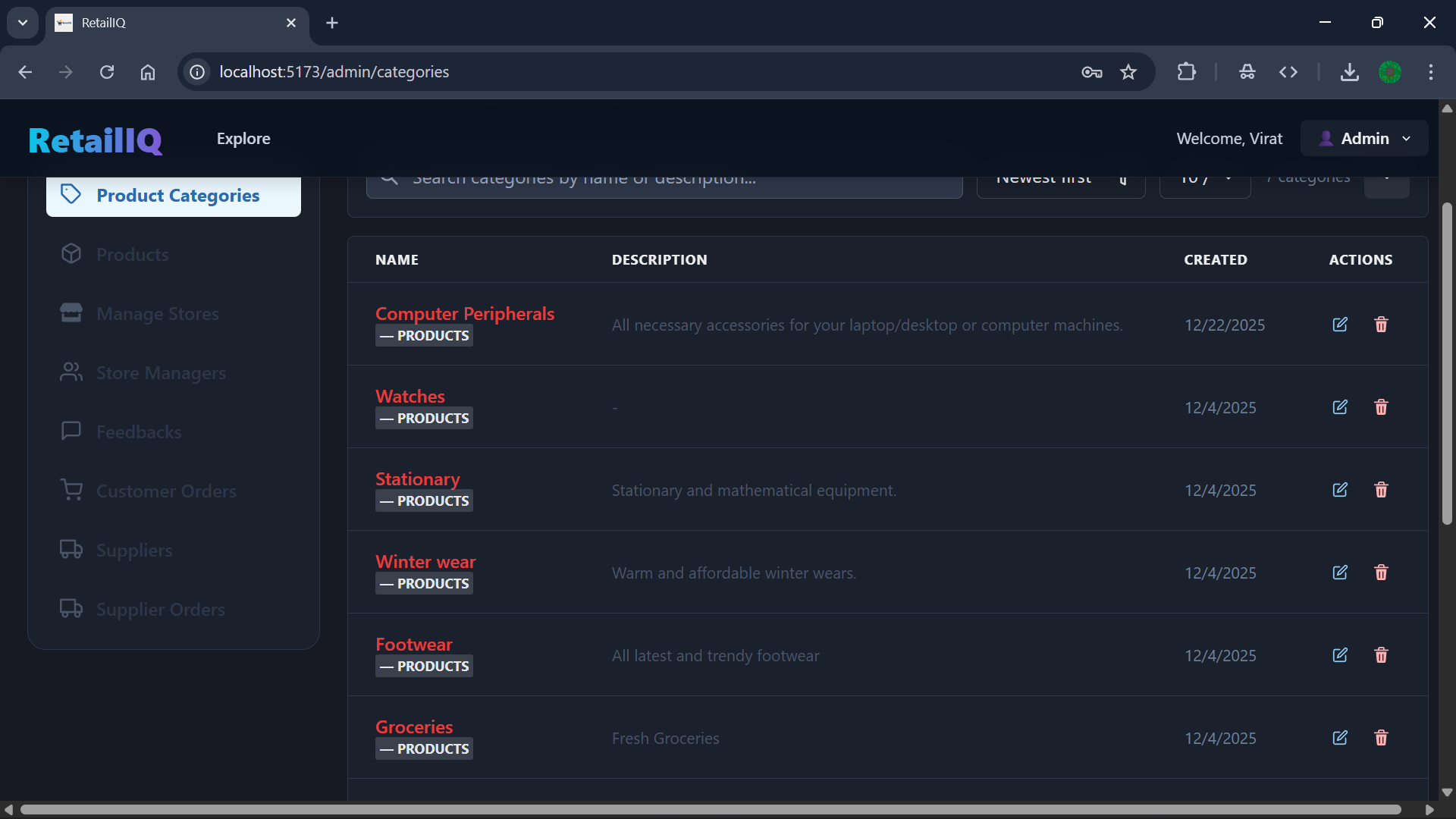


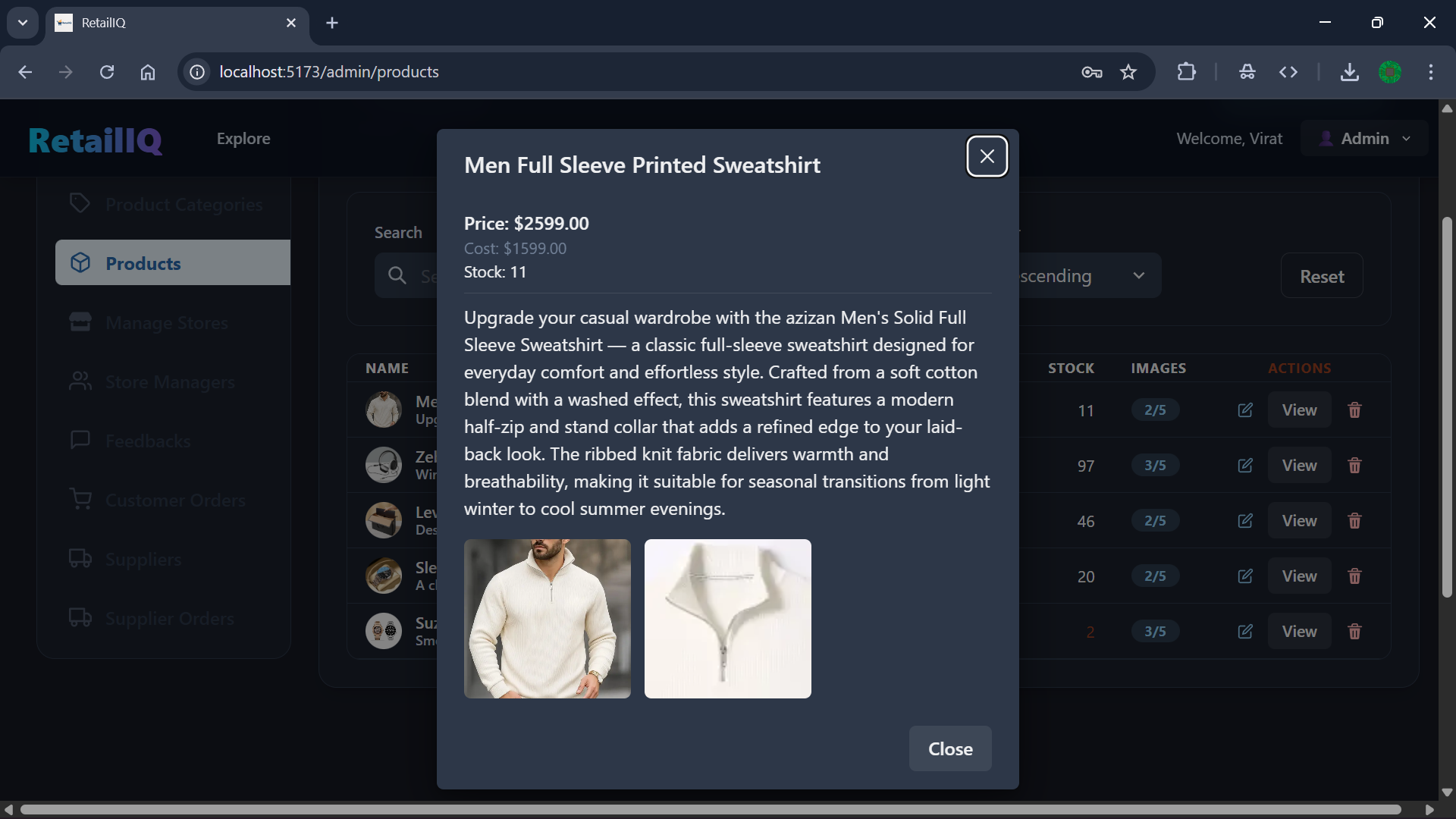
* Admin analytical dashboard

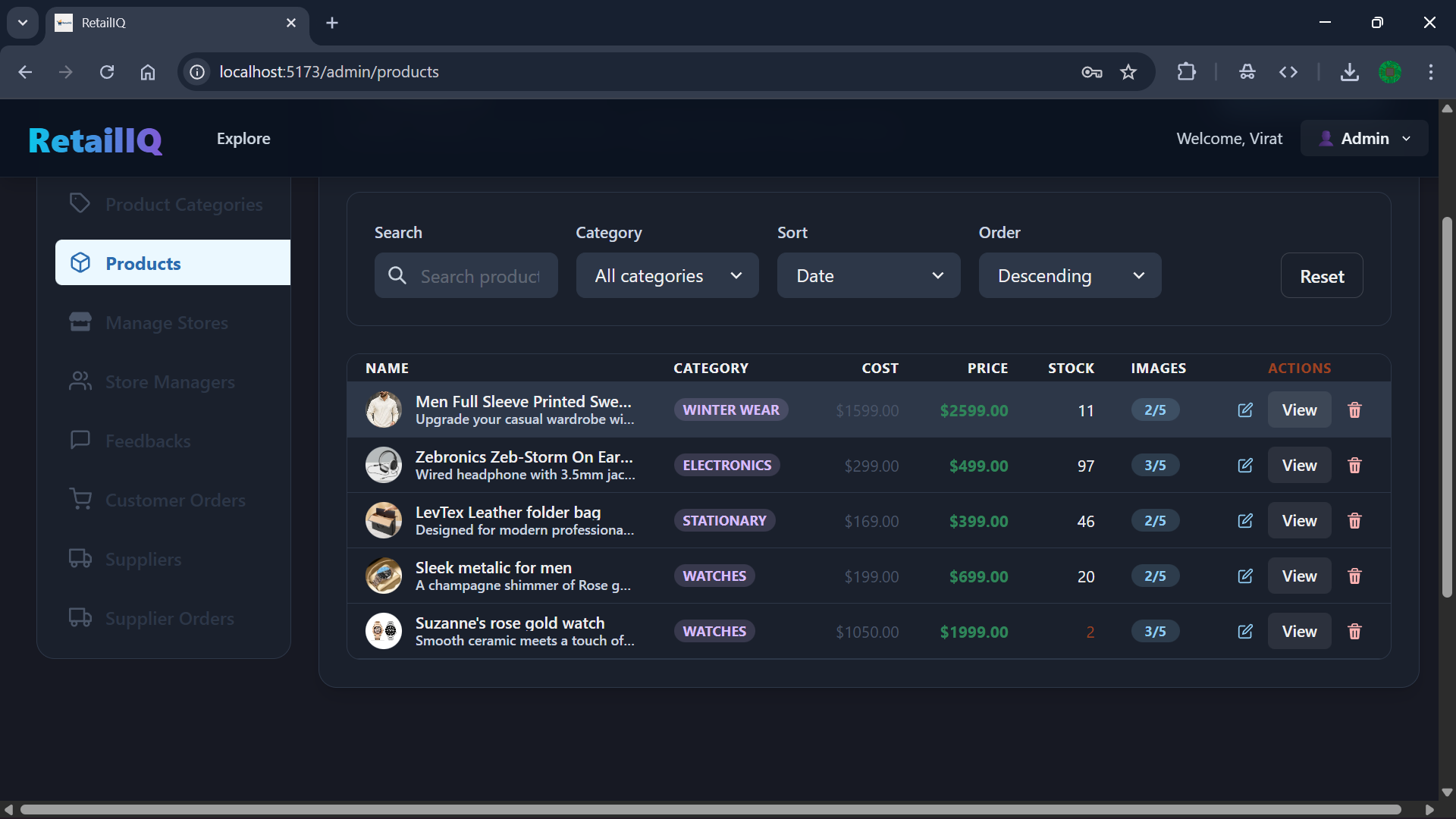


* Manage category and inventory by admin

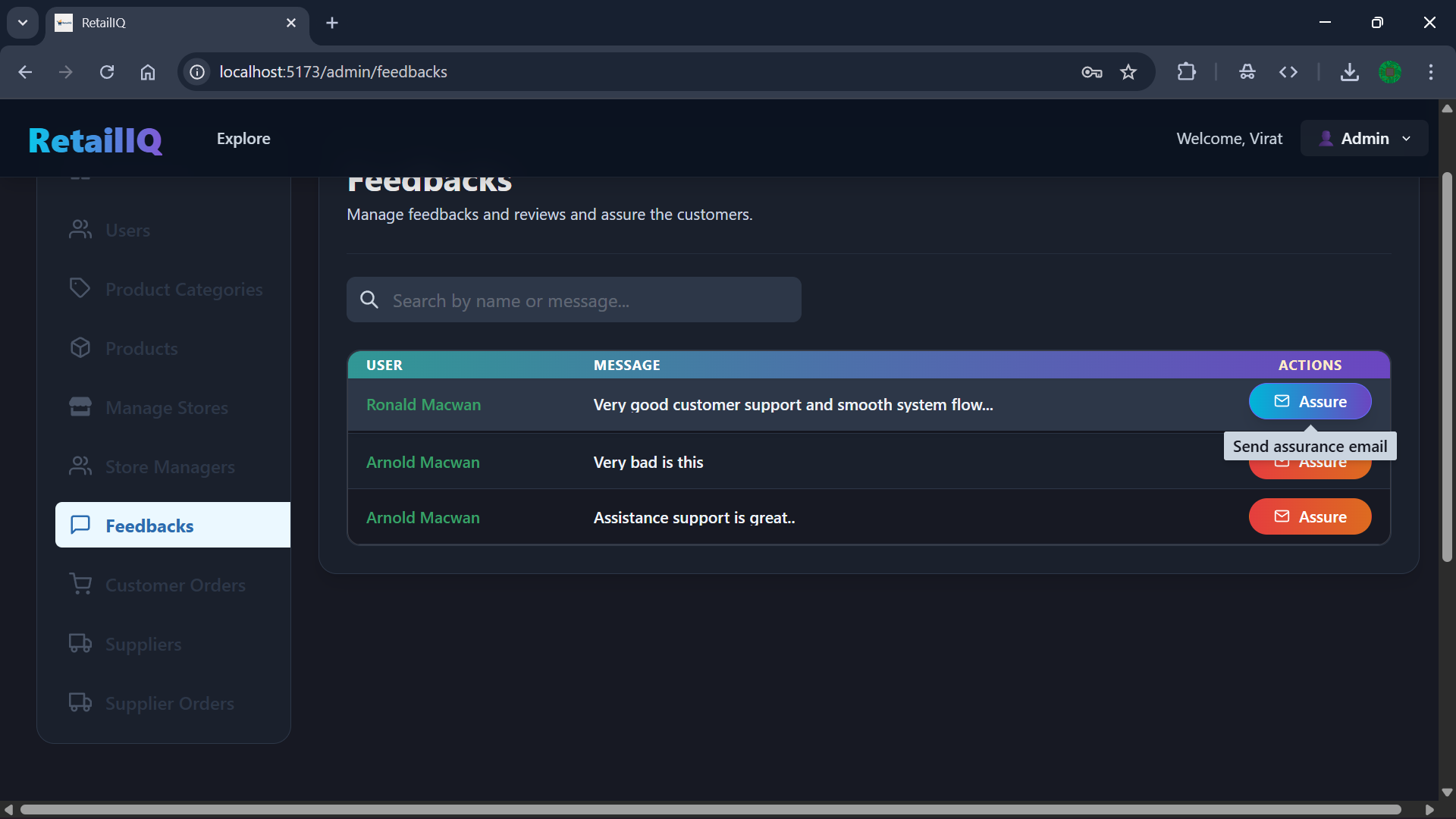








* View feedbacks by admin



1. **Agile Documentation**
   1. **Project Charter**

|  |  |
| --- | --- |
| Project Name | RetailIQ |
| Guide |  |
| Start Date | 01-10-2025 |
| End Date | 30-04-2025 |
| Project Scope | RetailIQ is a smart retail analytics and operations platform built to unify and automate key retail workflows. The project covers inventory management, sales tracking, supplier coordination, and AI-driven analytics for demand forecasting, market-basket insights, dynamic pricing, and predictive stock management. |
| Project Mission | To empower local retailers, supermarkets, and small businesses with intelligent, data-driven tools that simplify operations, optimize inventory, enhance profitability, and enable smarter decision-making through actionable insights and modern AI/ML technologies. |
| Project Vision | To become the leading smart retail analytics platform that transforms traditional retail into efficient, tech-enabled businesses—bridging the gap between small stores and enterprise-level retail intelligence through affordable, scalable, and automated solutions. |

* 1. **Project Roadmap**
  2. **User Story**
  3. **Release Plan**
  4. **Sprint Backlog**
  5. **Test Plan**

1. **Proposed Enhancement**
2. **Conclusion**
3. **Bibliography**