Medicare.

Developed By: Pai Najranabanu Yakub

DESCRIPTION:

- Create a dynamic and responsive Java e-healthcare web application for ordering medicines of different categories.

Background of the problem statement:

- Medicare is a company that supplies medicines and a couple of other healthcare essentials at an affordable price. It was established in 2012 in Delhi, India. It had been serving fine all these years; however, the business analysts noticed a decline in sales since 2017. They found out that online ordering of medicines with companies, such as 100mg and mfine are gaining more profits by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop a healthcare web application with a rich and user-friendly interface.
- You are hired as the Full Stack Java developer and are asked to develop the web application. The management team has provided you with the requirements and their business model so that you can easily arrange different components of the application.

Features of the application:

- Registration.
- Login.
- Payment gateway.
- Searching.
- Filtering
- Sorting
- Dynamic data
- Responsive and compatible with different device.

Recommended technologies:

- Database management: MySQL and Oracle.
- Backend logic: Java programming, NodeJS.
- Frontend development: JSP, Angular, Bootstrap, HTML/CSS, and Javascript.
- Automation and testing technologies: Selenium, Jasmine (frontend testing), and TestNG.
- DevOps and production technologies: Git, GitHub, Jenkins, Docker, Kubernetes, and AWS.

Scope:

- The scope of the project is to build a backend and frontend web application for E-com store of a medicine retailer using spring boot and angular respectively by following REST architecture using MySQL as database and Hibernate as ORM tool. Apart from that it uses spring security robust solution along with JWT authentication token to authenticate and authorize users.

Steps to install:

- Download and install the java spring boot project from backend repository.
- Edit the application. properties in the backend with the database you are using.
- Uncomment all the commented code in the MedikareApplication.java (root file with @SpringBootApplication annotation) while running for the first time.
- After the first time comment the code again.
- install the Medicare frontend project made with angular.
- Edit the baseUrl in helper.ts in services folder to the url you are running your spring boot application on.
- Login as admin with credentials admin as username and password or create a new user with registration form.

Technologies used:

- Backend: Spring boot, spring data jpa, spring security, JWT Authentication.
- Frontend: Angular, Bootstrap, Material UI
- Database: MySQL (Hibernate as ORM).

SPRINT PLANNING:

- No.sprints: 2
- Duration : 1 week.

Product backlog:

- T101- Create spring boot Application and model for User, Roles, Product, Cart, Purchase and Category. Download and install the java spring boot project from backend repository.
- T102- Make database and map the model to it using Hibernate.
- T103- Make JWT authentication System with Spring security.
- T104- Make REST APIs for the products, users, category, cart and purchases.
- T105- Create frontend using angular and material UI.
- T106- Make pages for login, registration, shop, product, cart and admin dashboard.
- T107- Implement crud operations for Category and products in admin dashboard.
- T108- Add Cart and product validation and login and registration validation too.

Sprint Breakdown:

- T101:
  - Story points: 8
  - Description: Create spring boot Application and model for Users, Roles, Product, Cart, Purchase and Category.
  - Acceptance criteria: above mentioned work should be implemented properly.
- T102:
  - Story points: 8
  - Description: Make database and map the model to it using Hibernate.
  - Acceptance criteria: above mentioned work should be implemented properly.

- T103:
  - Story points: 8
  - Description: Make JWT authentication System with Spring Security.
  - Acceptance criteria: above mentioned work should be implemented properly.

- T104:
  - Story points: 8
  - Description: Make REST APIs for the products, users, category, cart and purchases.
  - Acceptance criteria: above mentioned work should be implemented properly.

- T105:
  - Story points: 8
  - Description: Create frontend using angular and material UI.
  - Acceptance criteria: above mentioned work should be implemented properly.

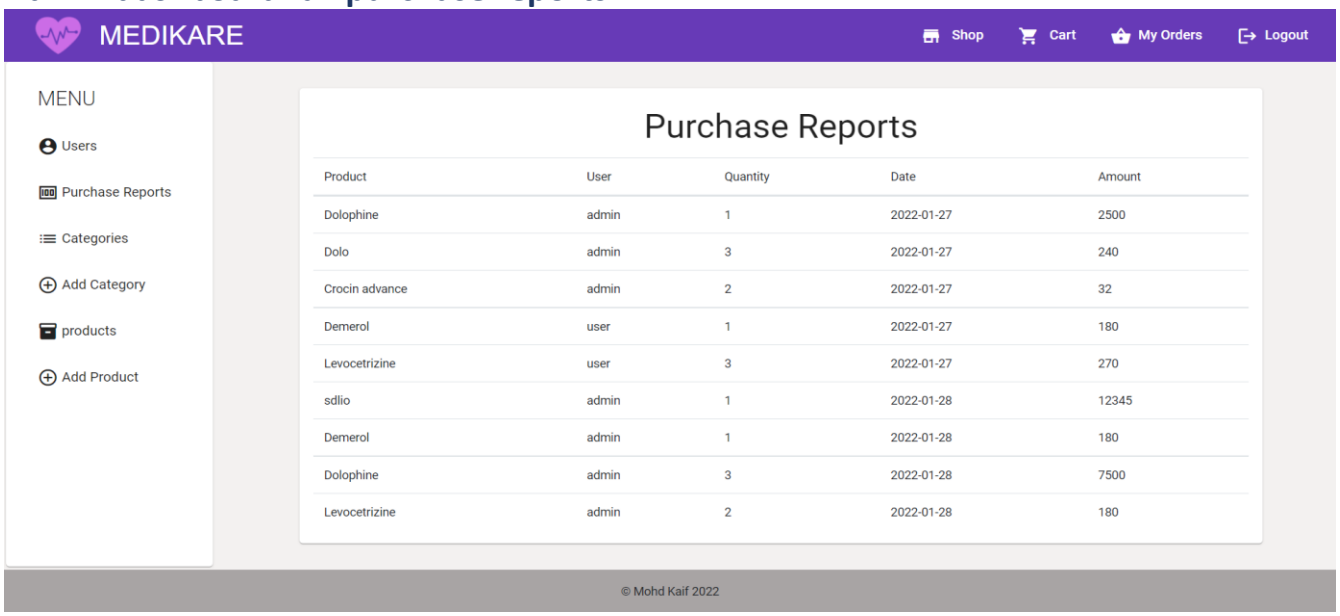
- T106:
  - Story points: 8
  - Description: Make pages for login, registration, shop, product, cart and admin dashboard.
  - Acceptance criteria: above mentioned work should be implemented properly.

- T107:
  - Story points: 8
  - Description: Implement crud operations for Category and products in admin dashboard.
  - Acceptance criteria: above mentioned work should be implemented properly.

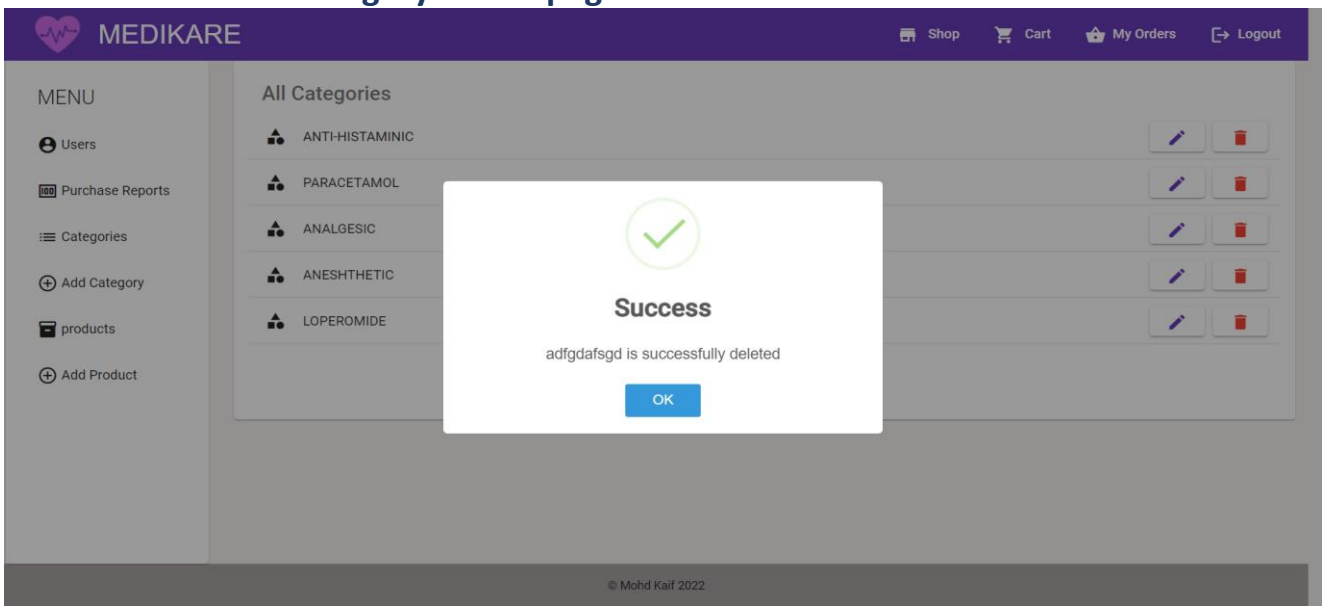
- T108:
  - Story points: 8
  - Description: Add Cart and product validation and login and registration validation too.
  - Acceptance criteria: above mentioned work should be implemented properly.

SCREENSHOTS:

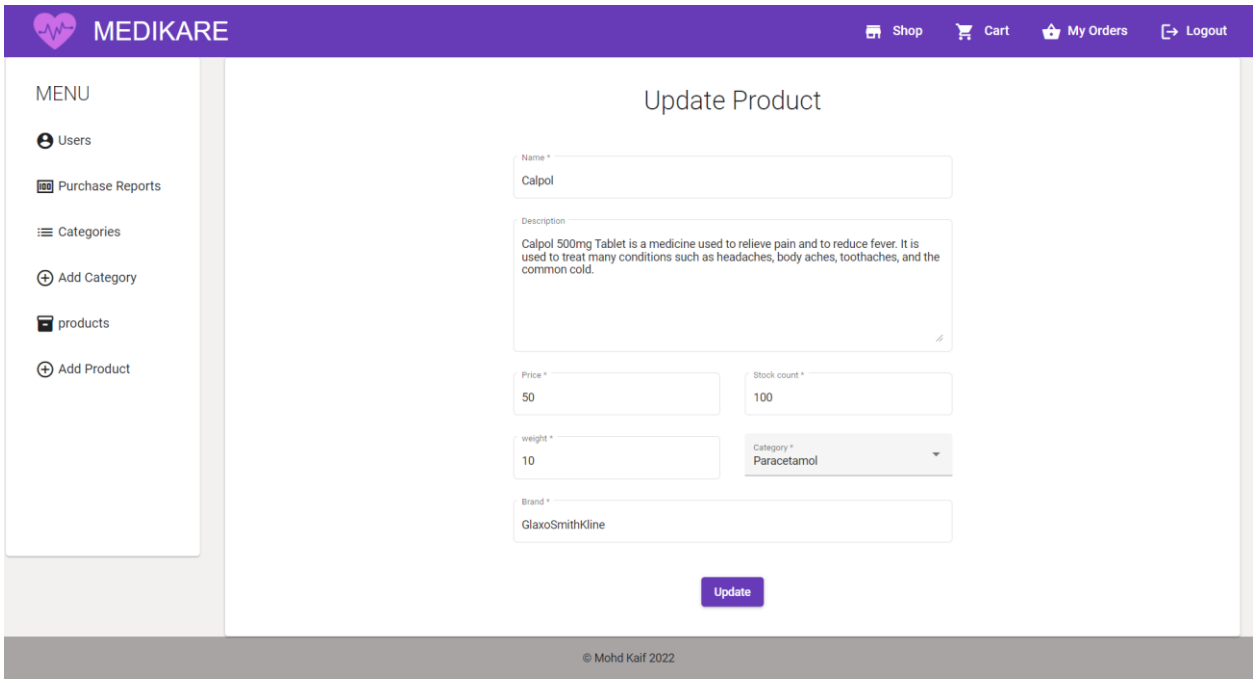
- **Admin dashboard- all purchase reports:**



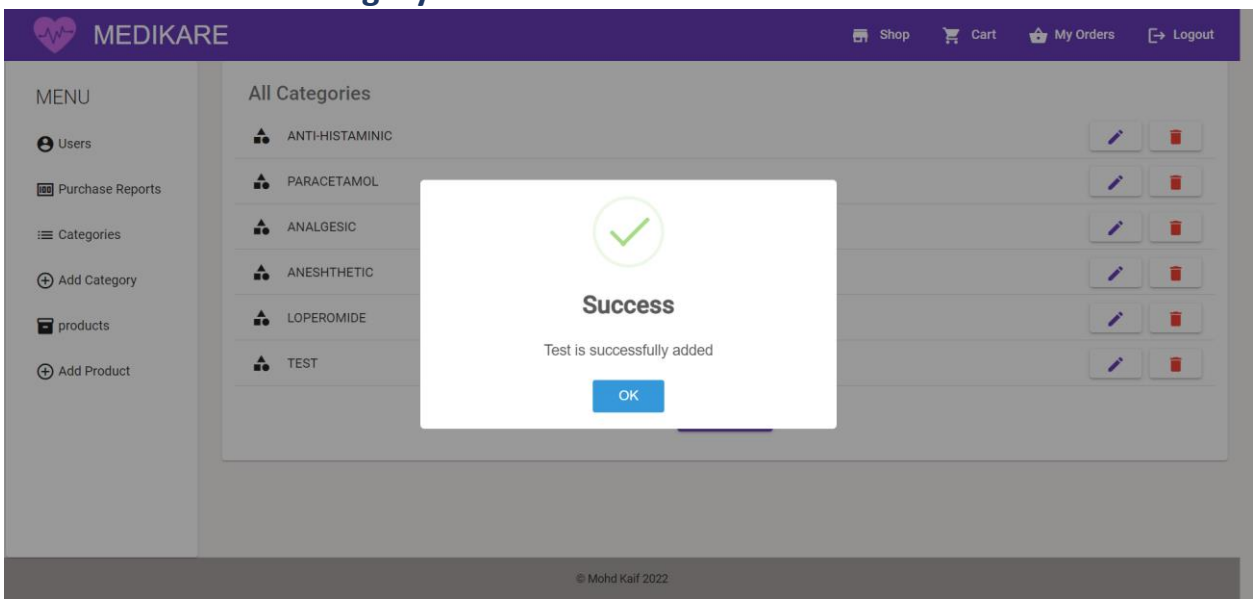
- **admin dashboard- category delete page:**



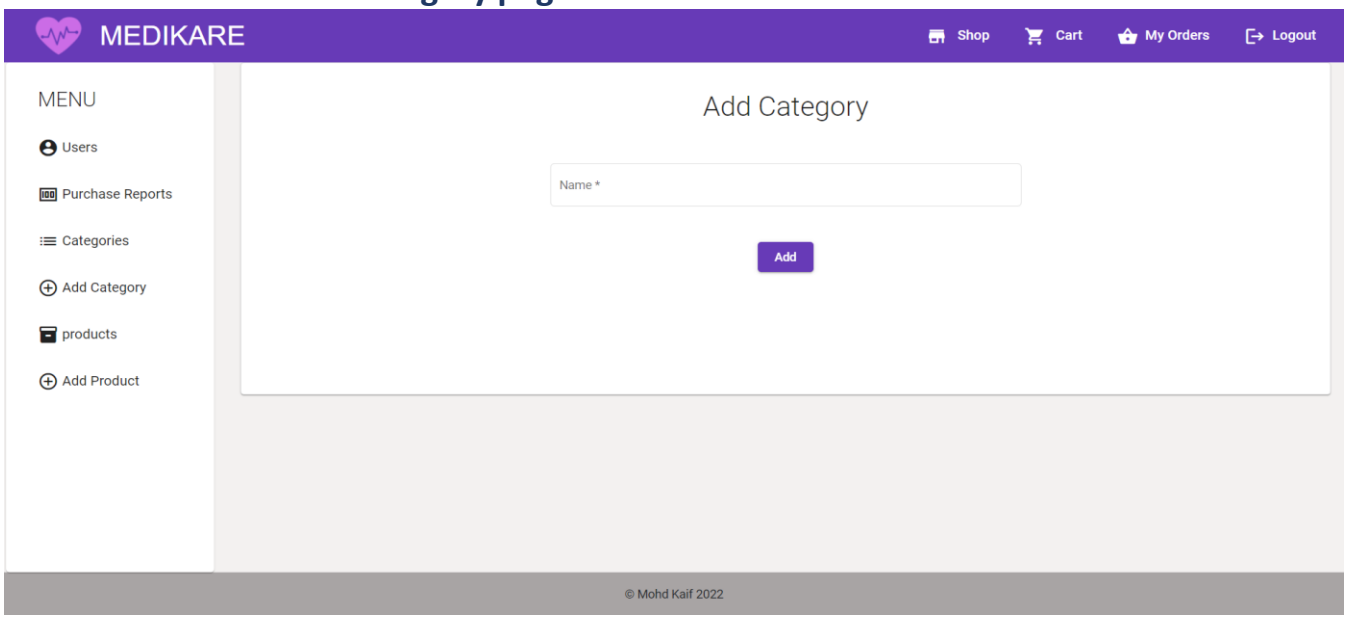
- **admin dashboard update product page:**



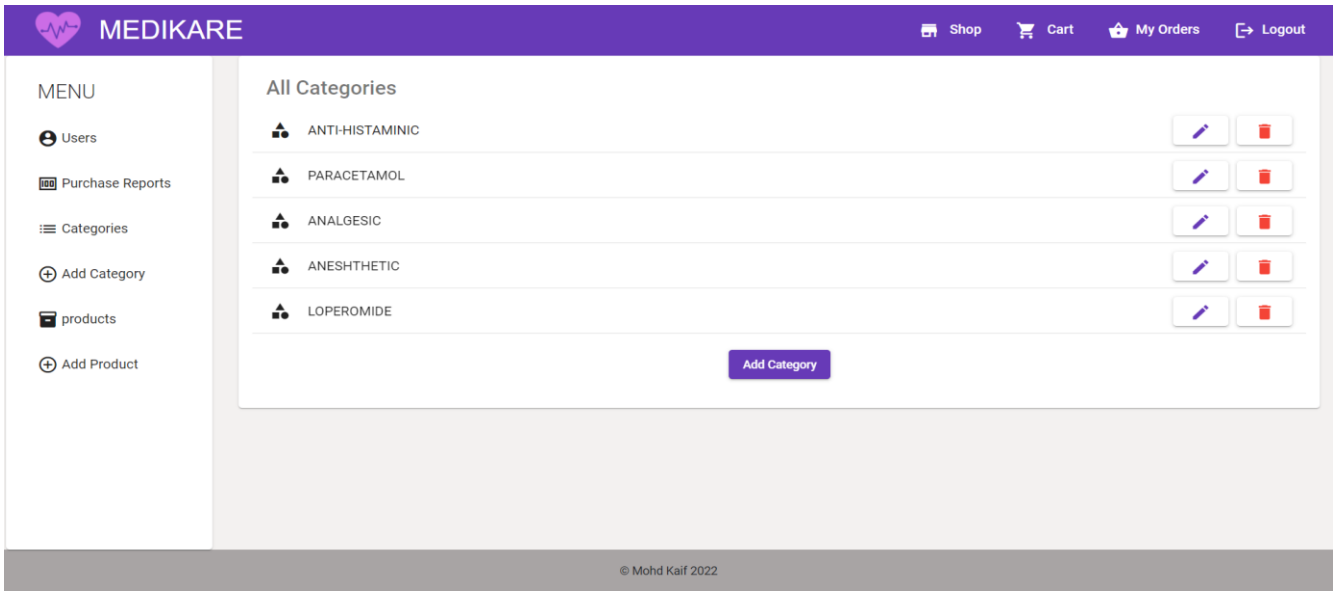
- **admin-dashboard category add validation:**



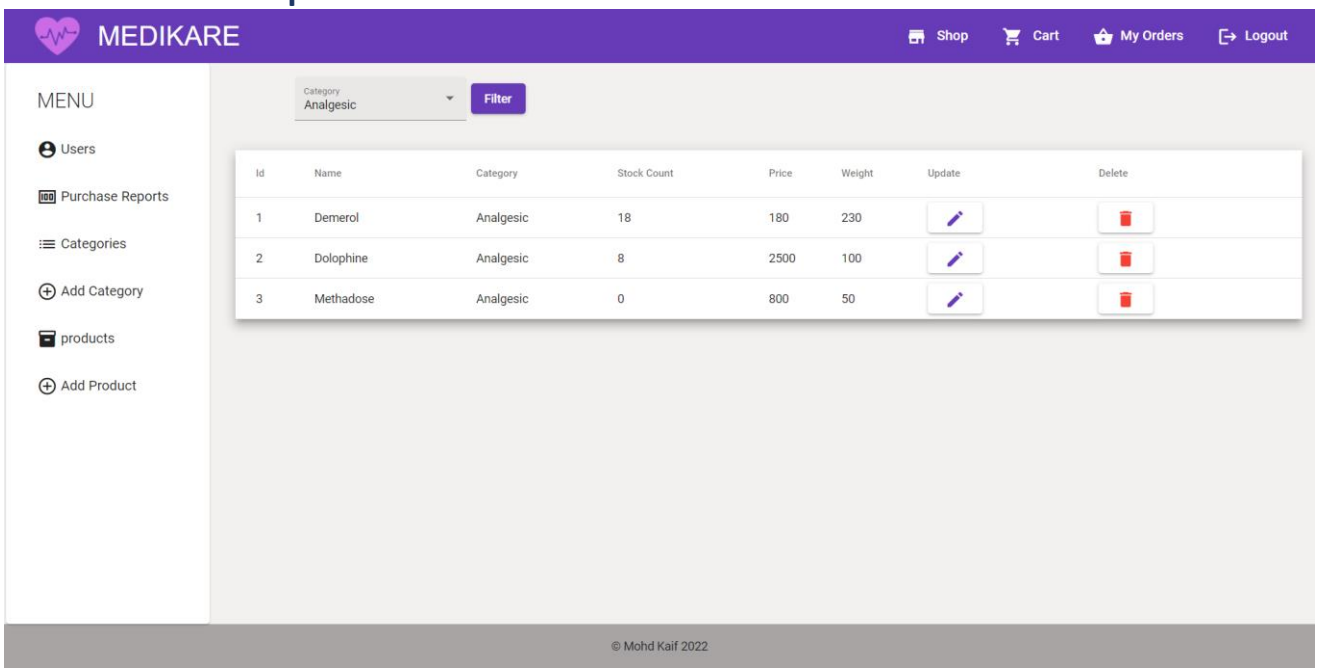
- **admin-dashboard add category page:**



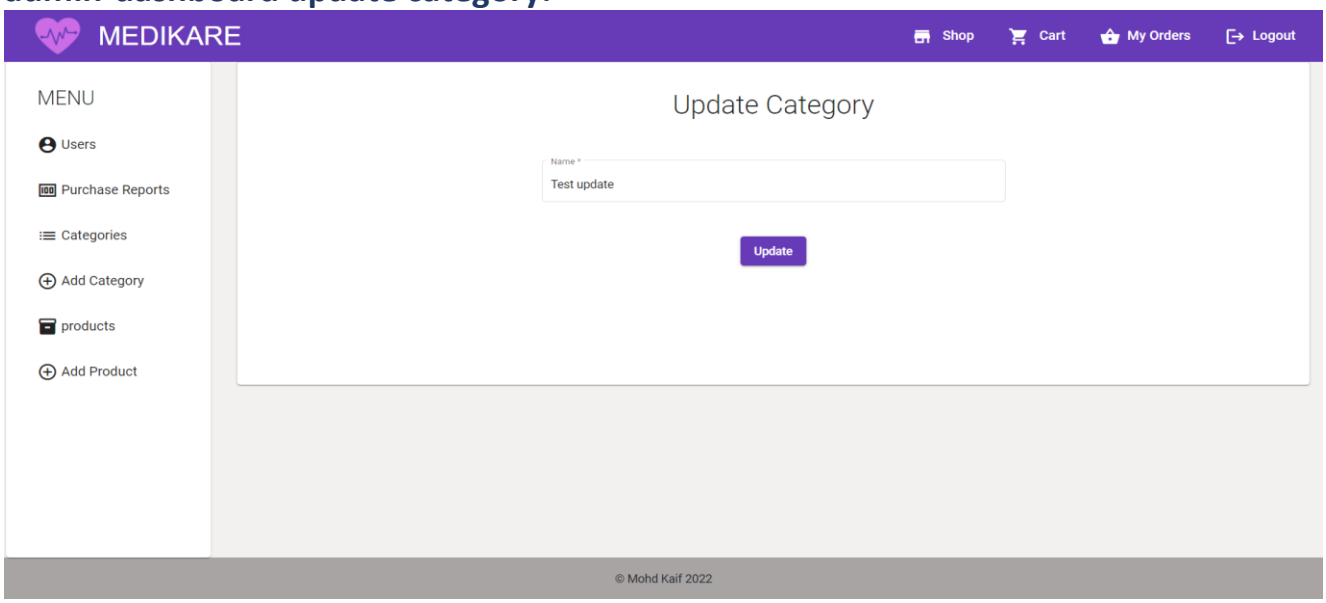
- **admin-dashboard categories list:**



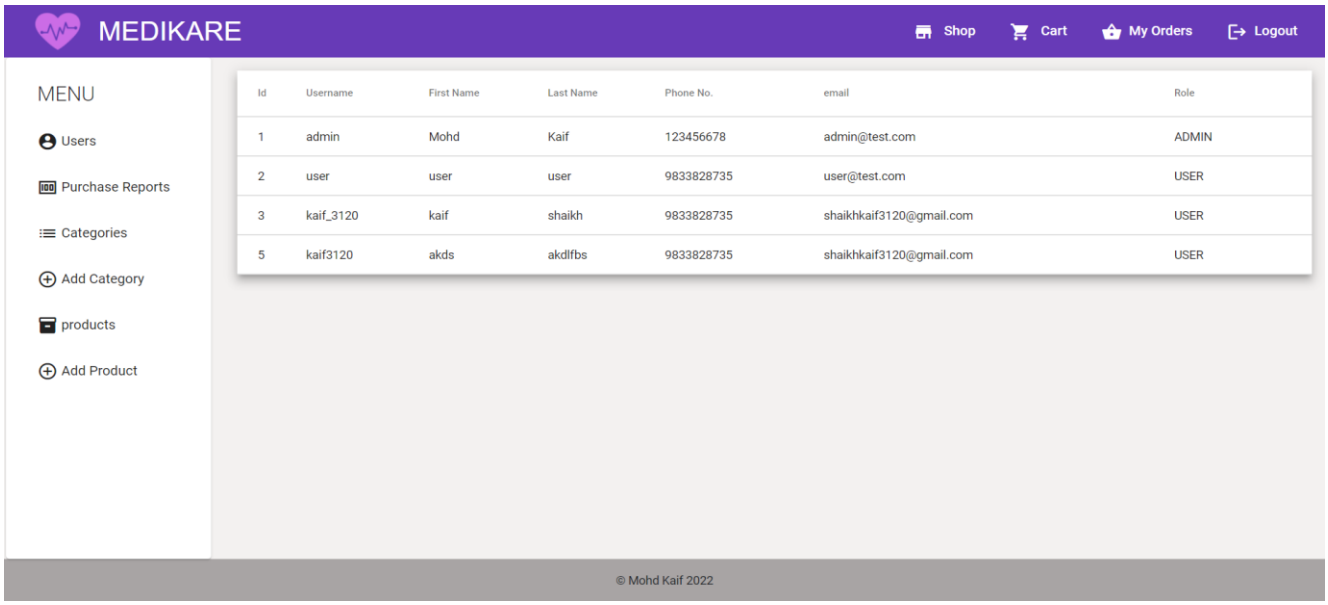
- **admin-dashboard products filter:**



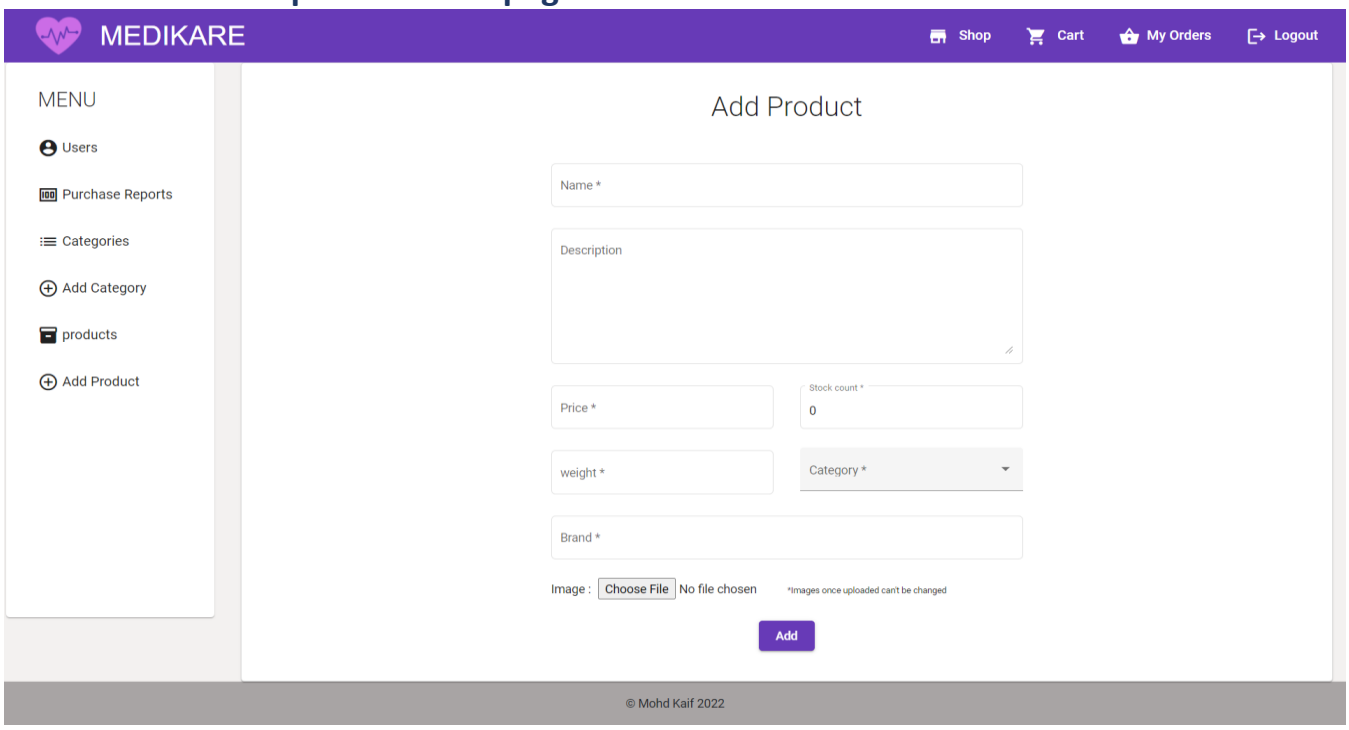
- **admin-dashboard update category:**



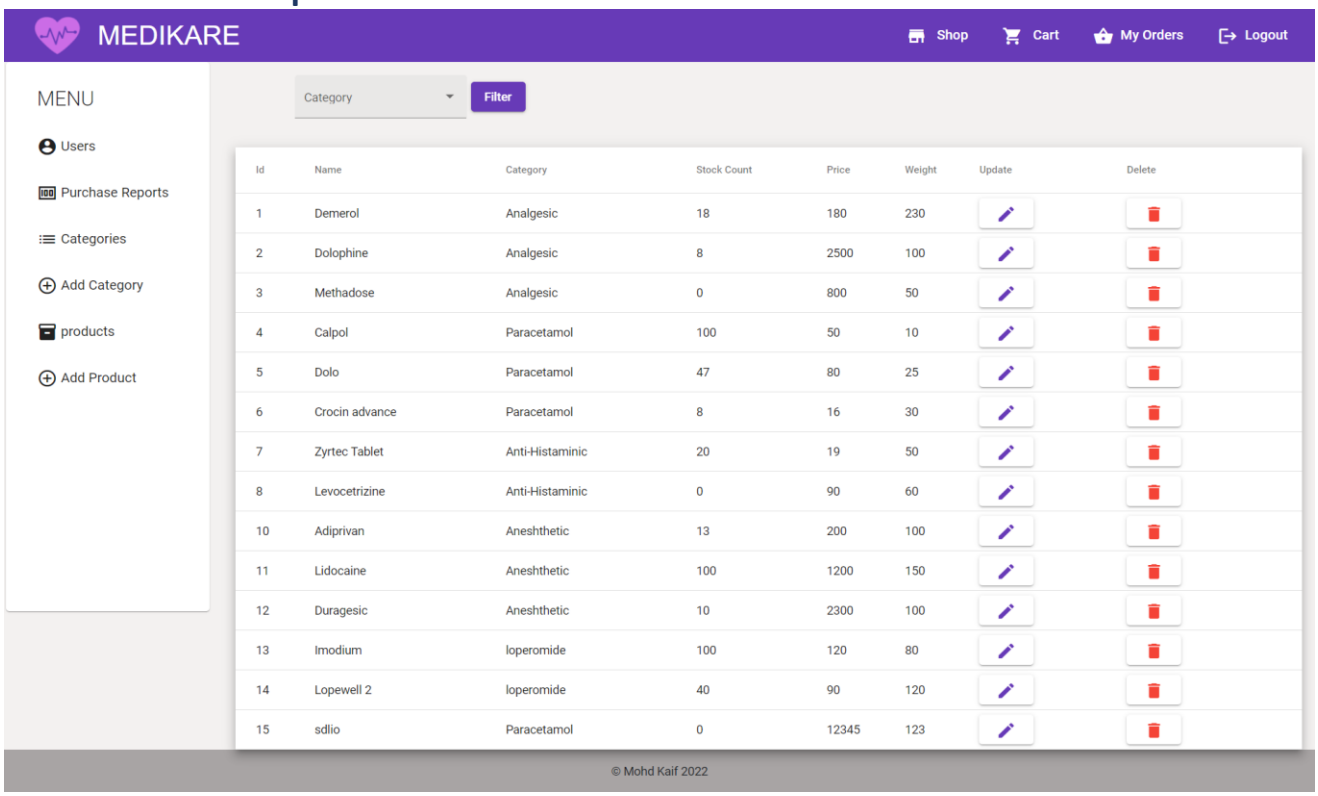
- **admin-dashboard users list:**



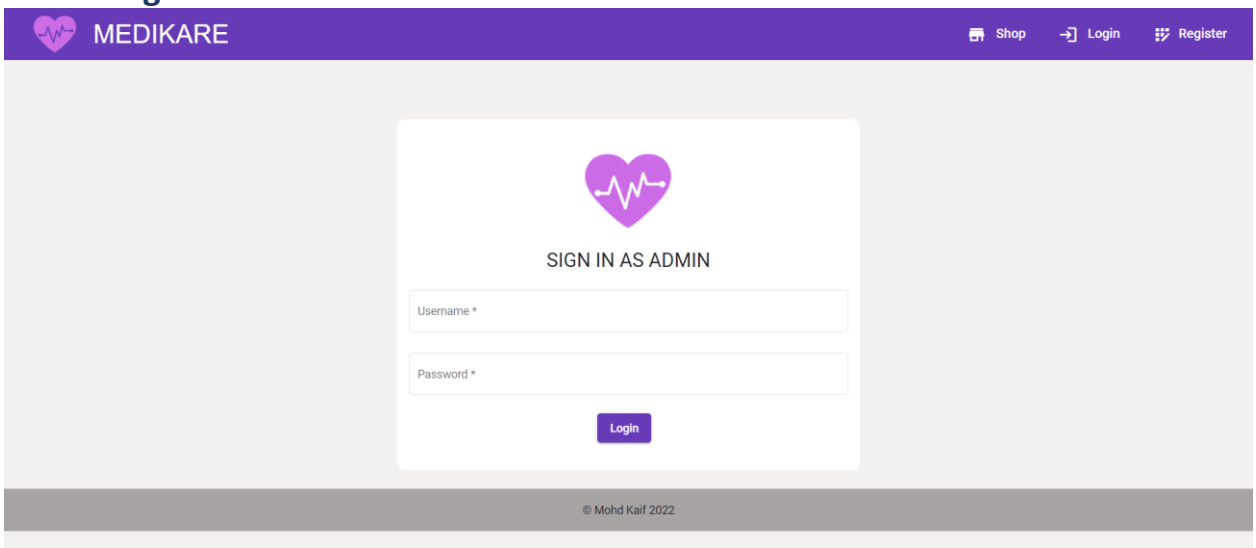
- **admin-dashboard-products add page:**



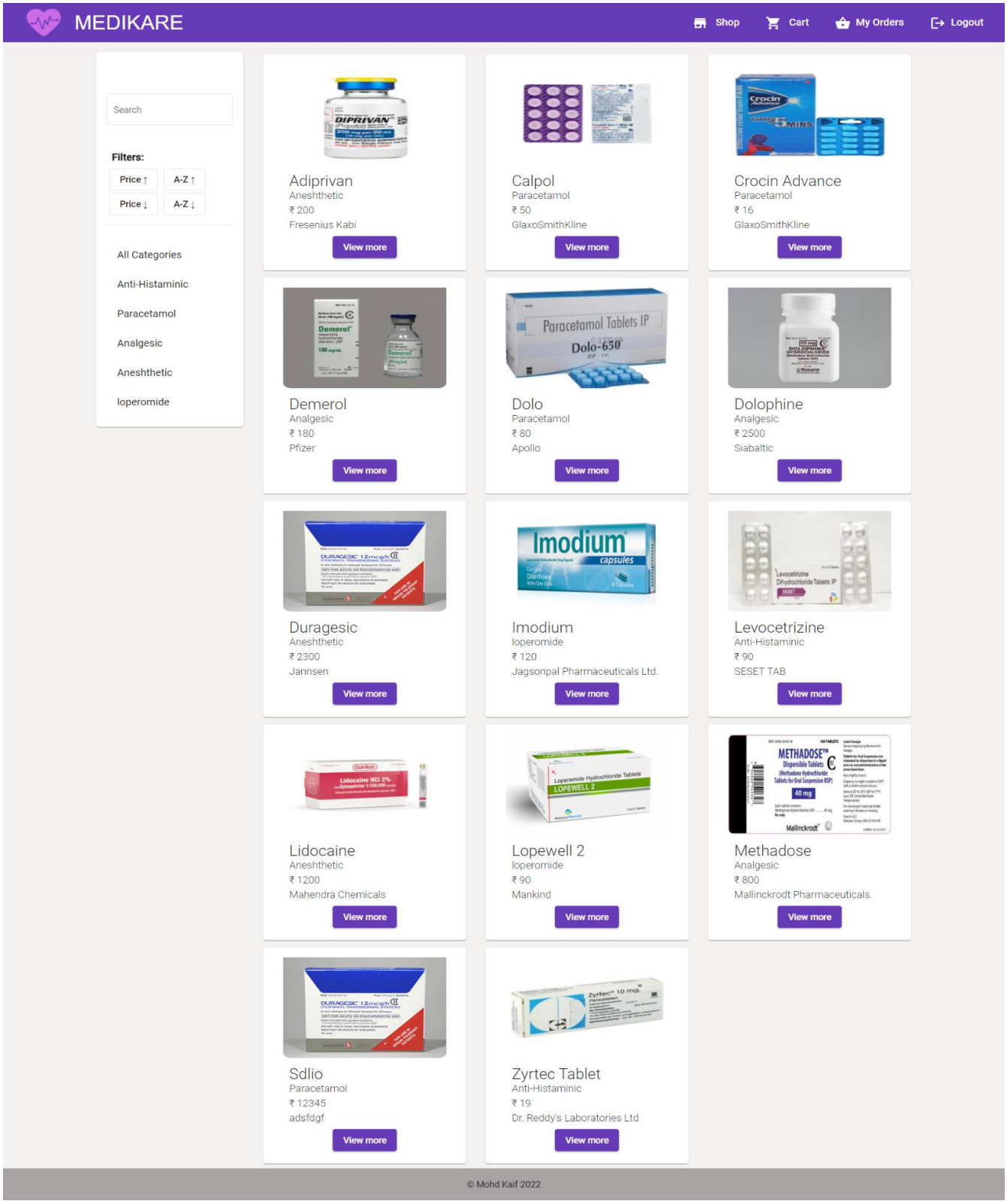
- **admin-dashboard-products list:**



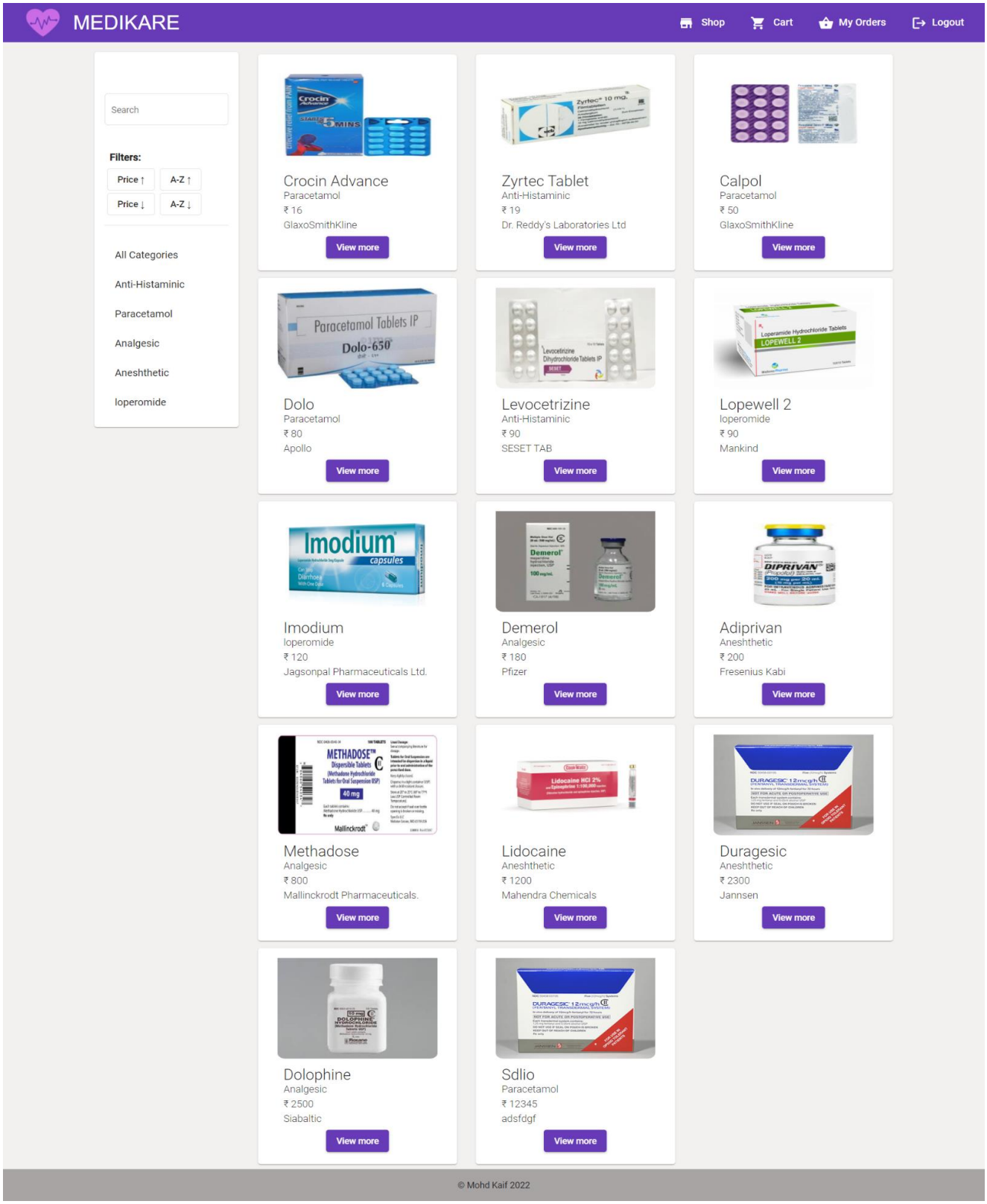
- **admin-login:**



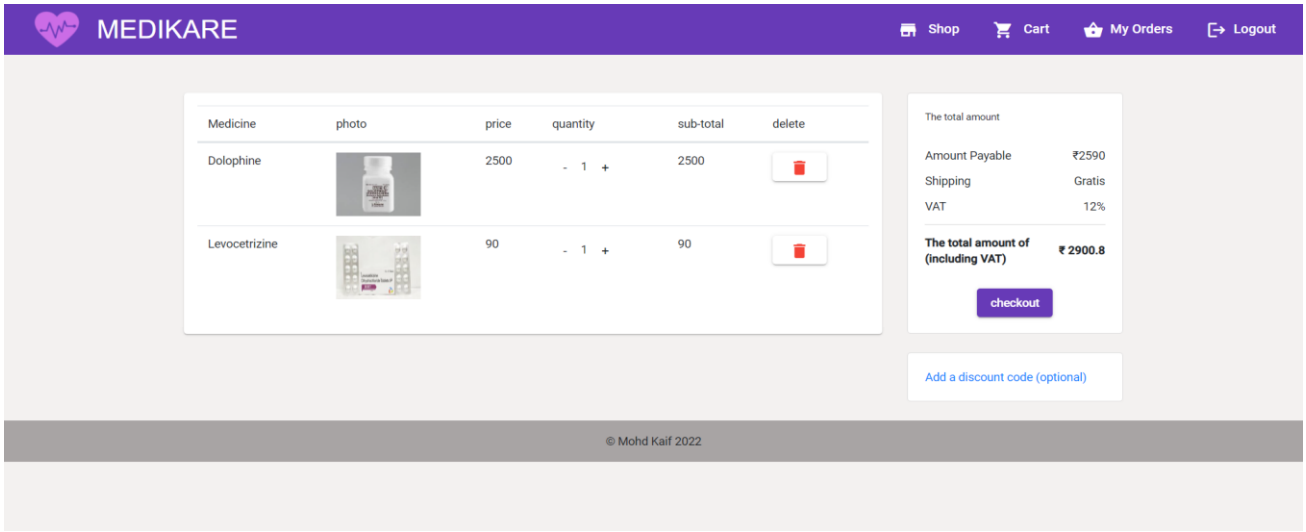
- **ascending alphabet sort:**



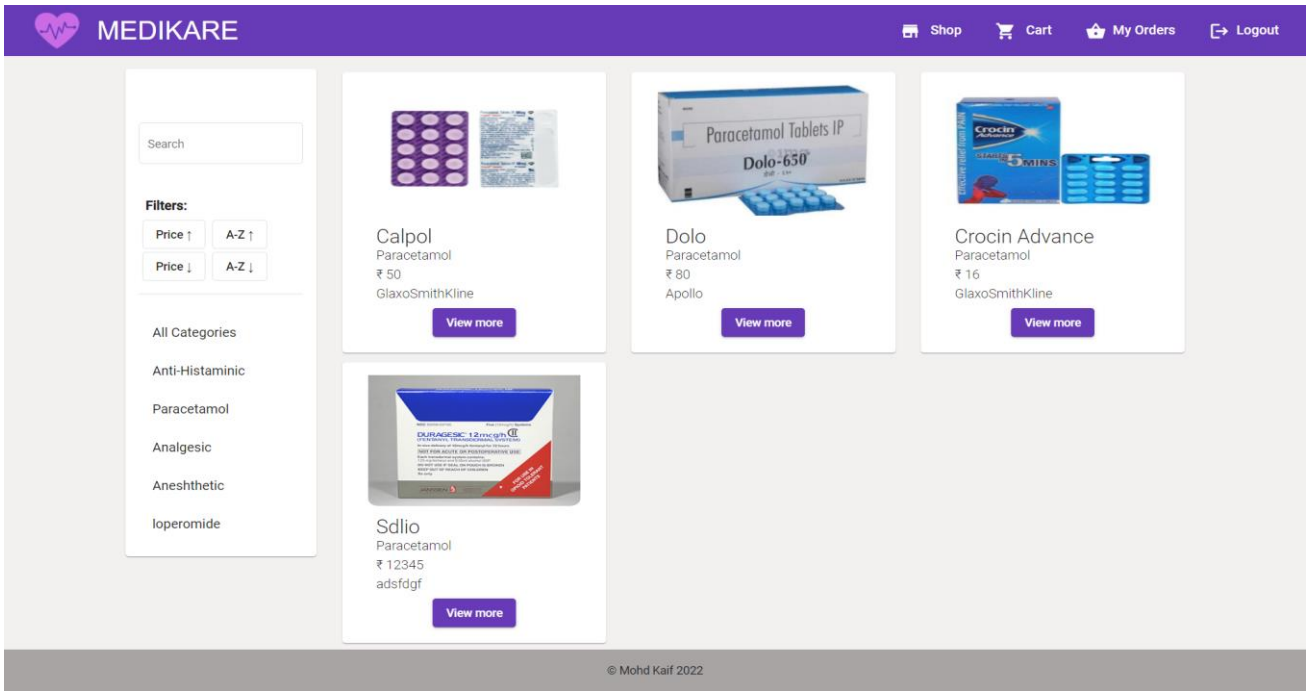
- **ascending prize sort:**



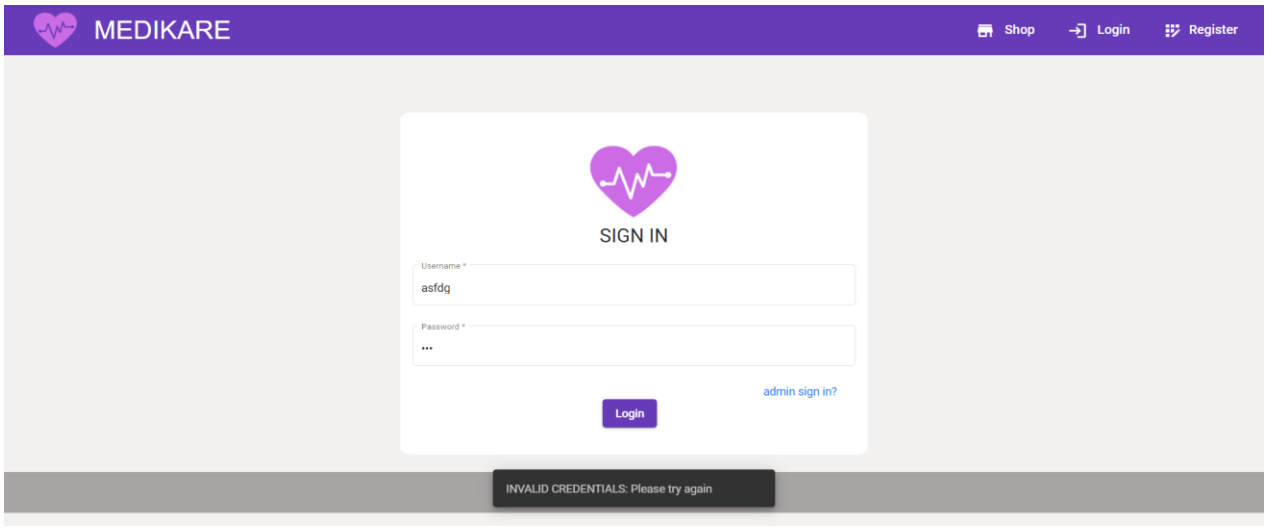
- **cart page:**



- **category based results:**

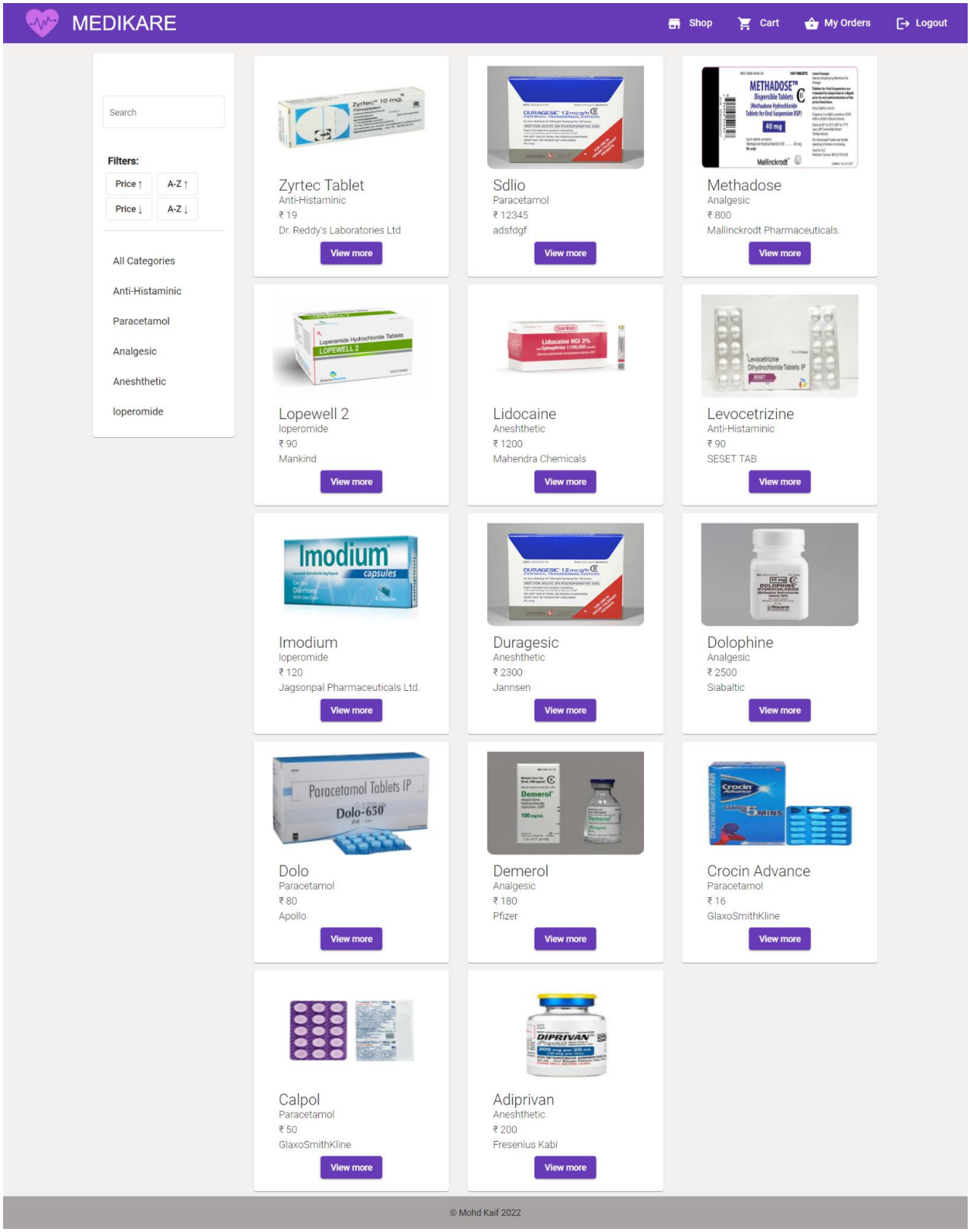


- **credentials validation:**



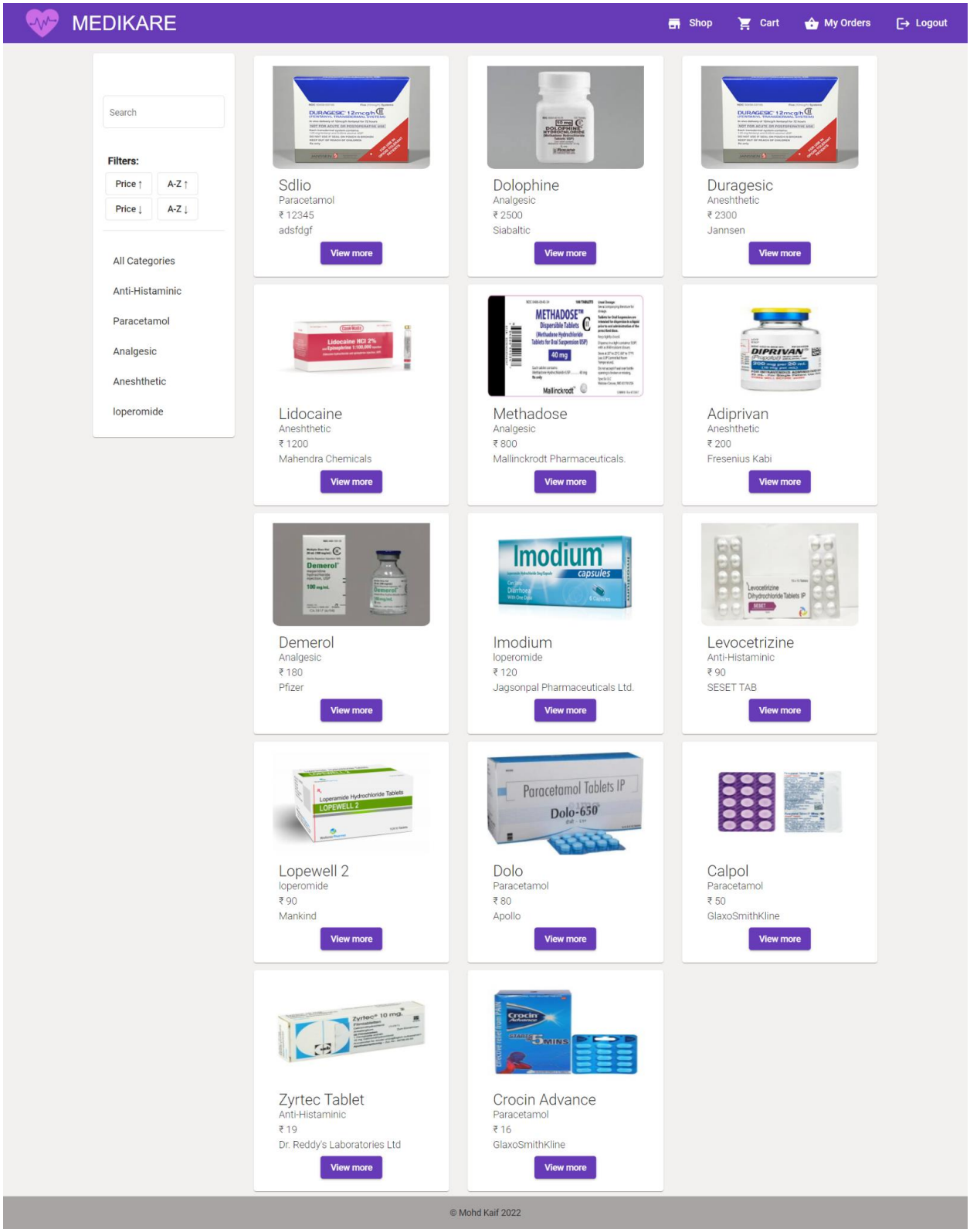
Project By: Pai Najranabanu

- **Descending alphabet sort:**

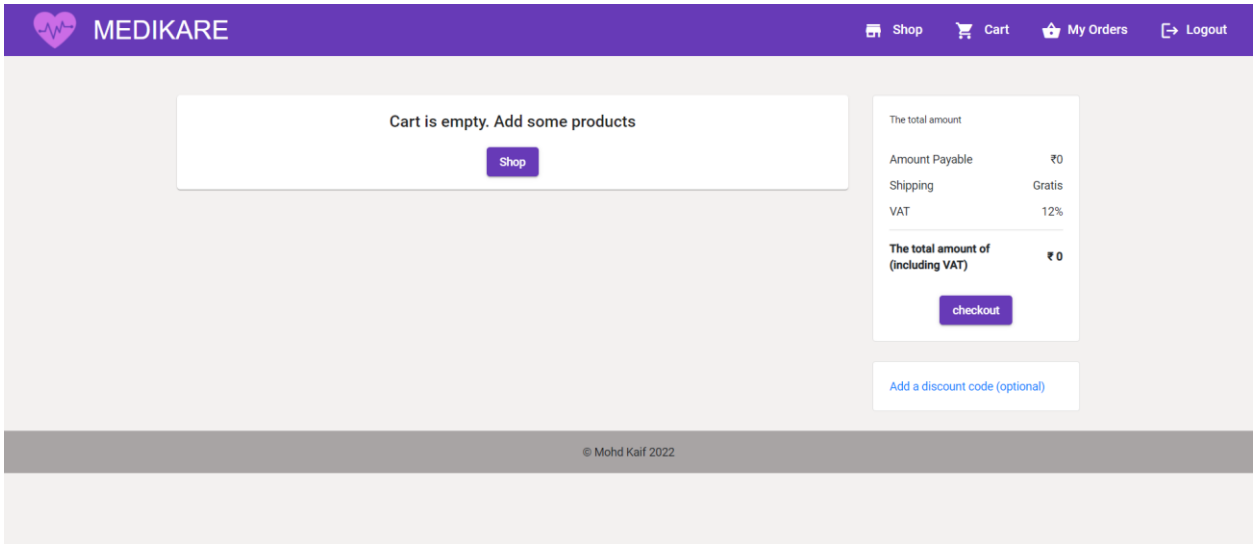


Project By: Pai Najranabanu

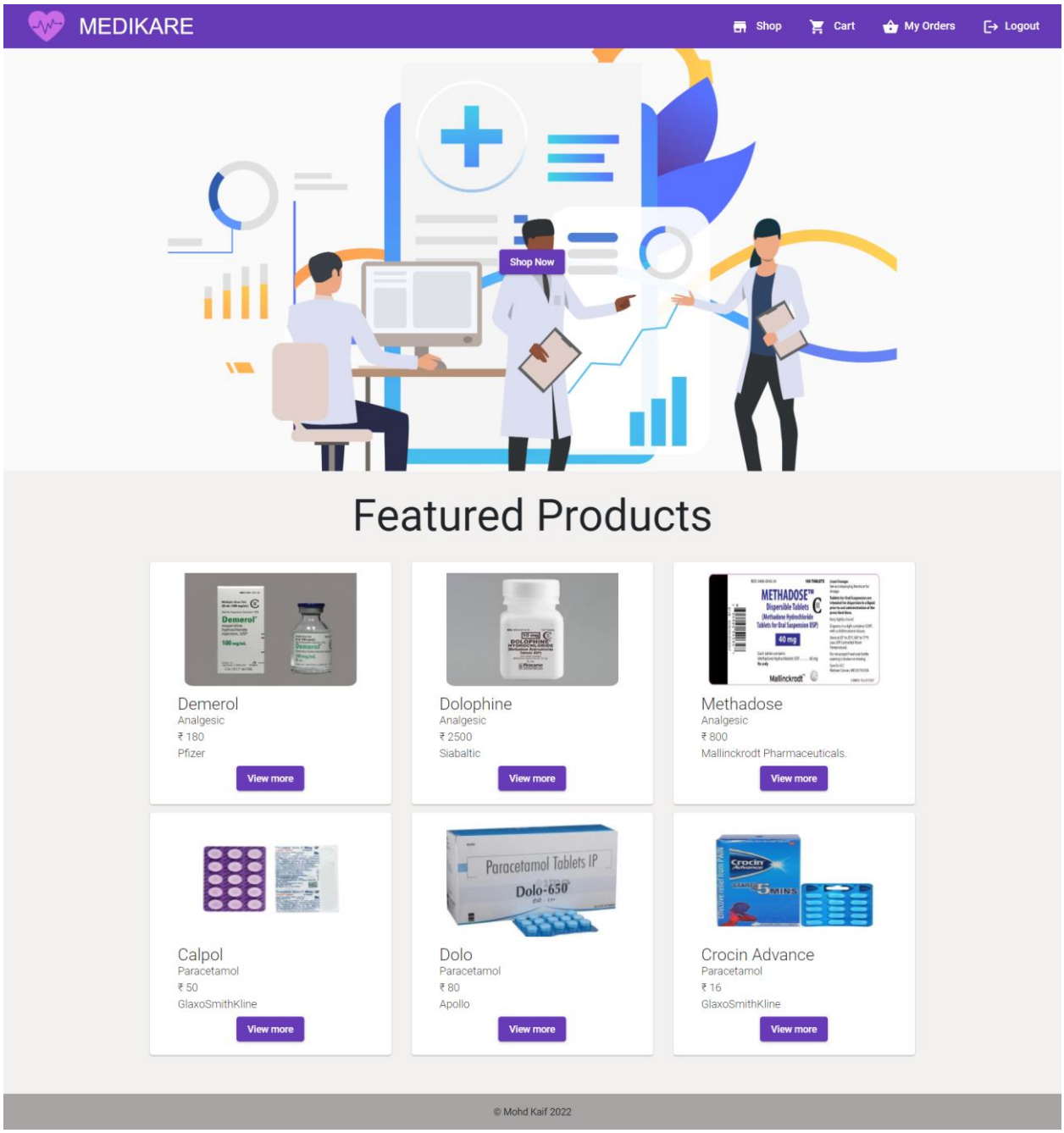
- **Descending prize sort:**



- **Empty cart page:**

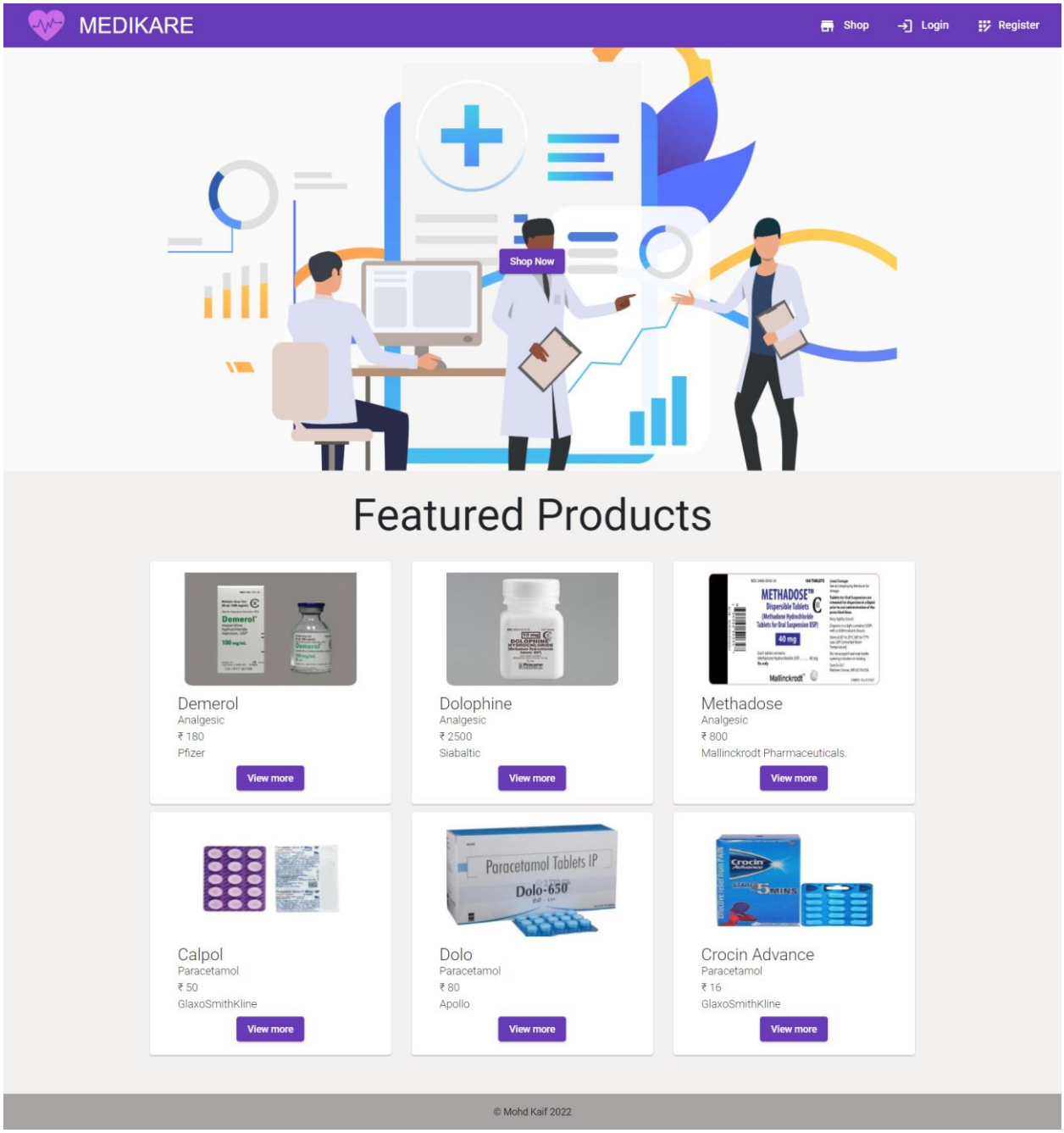


- **home- Page:**

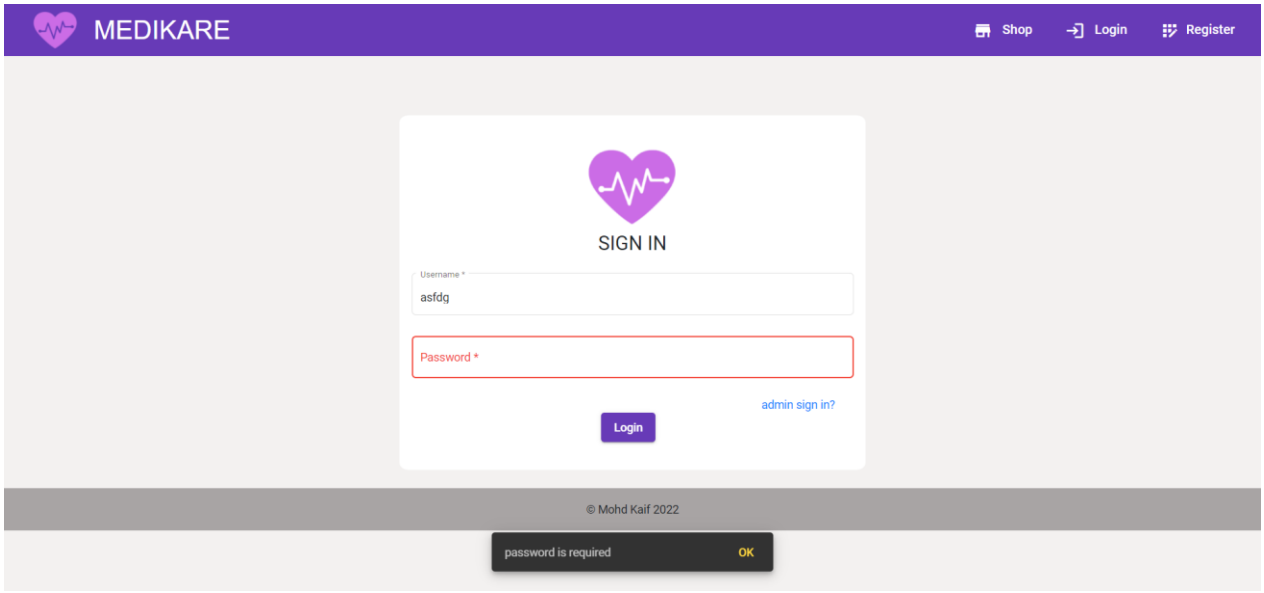


Project By: Pai Najranabanu

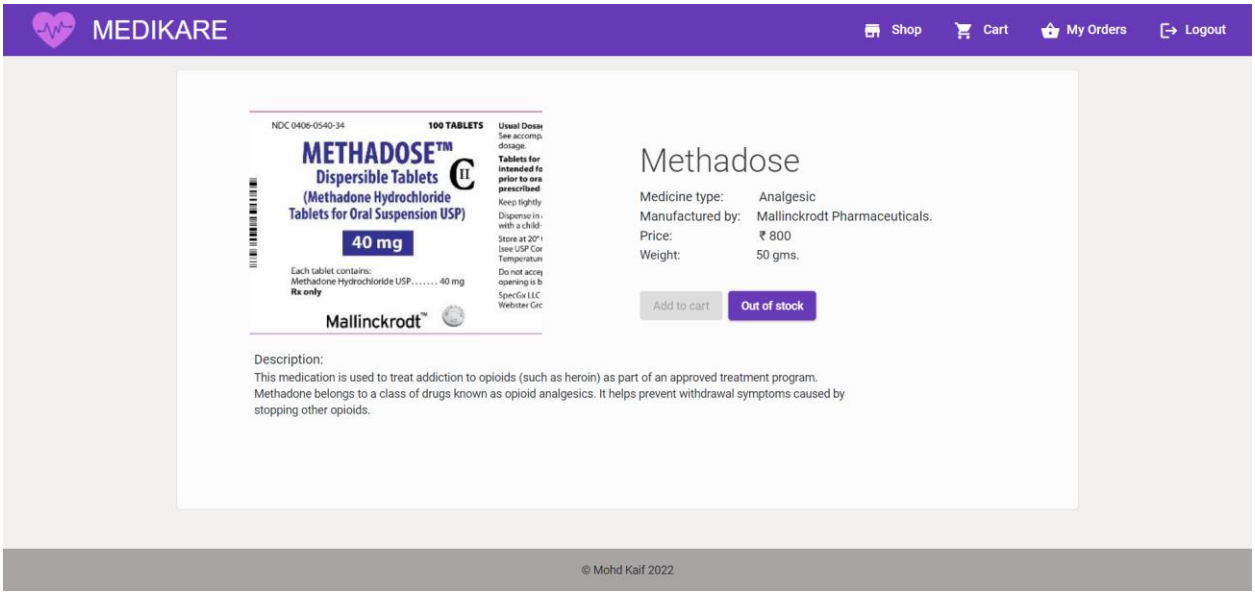
- **home-not logging:**



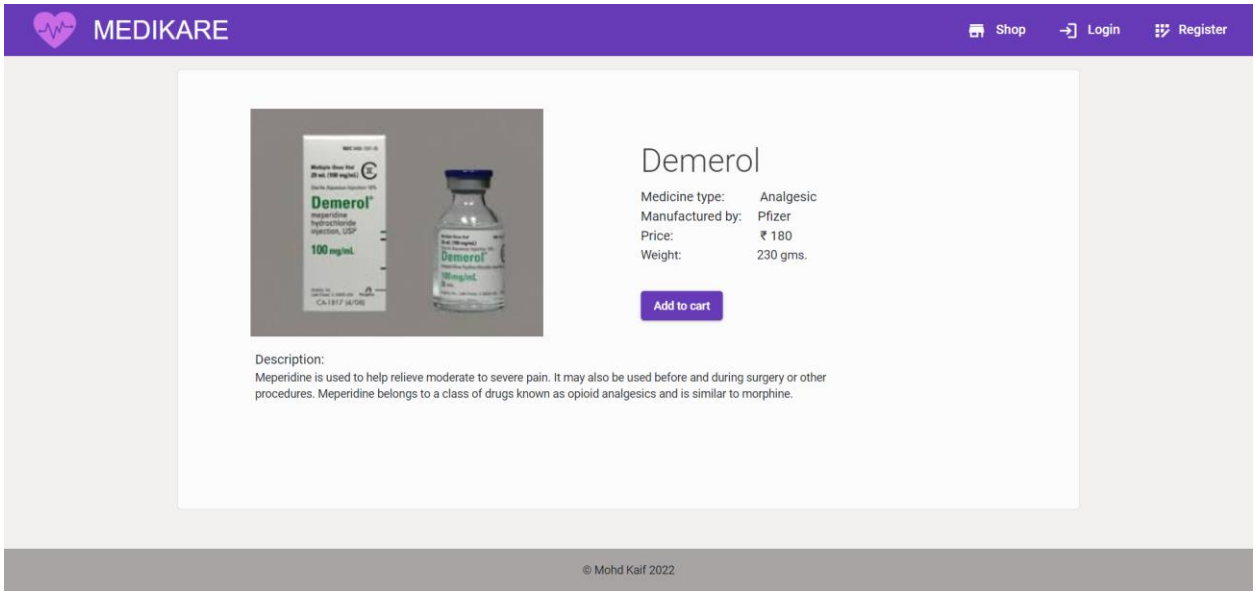
- **Login validation:**



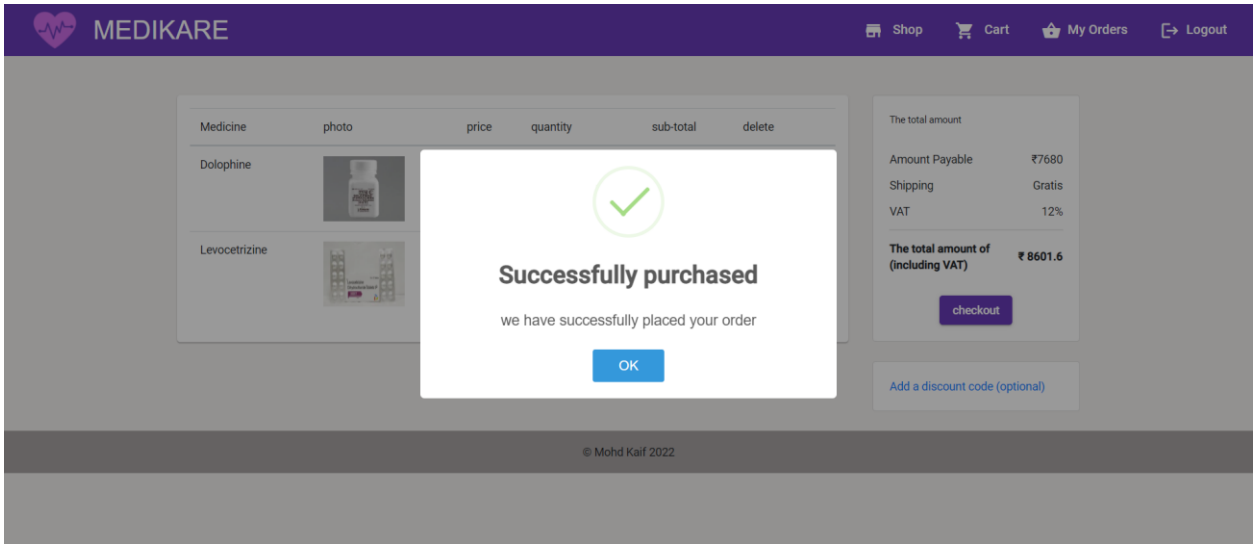
- **out of stock product:**



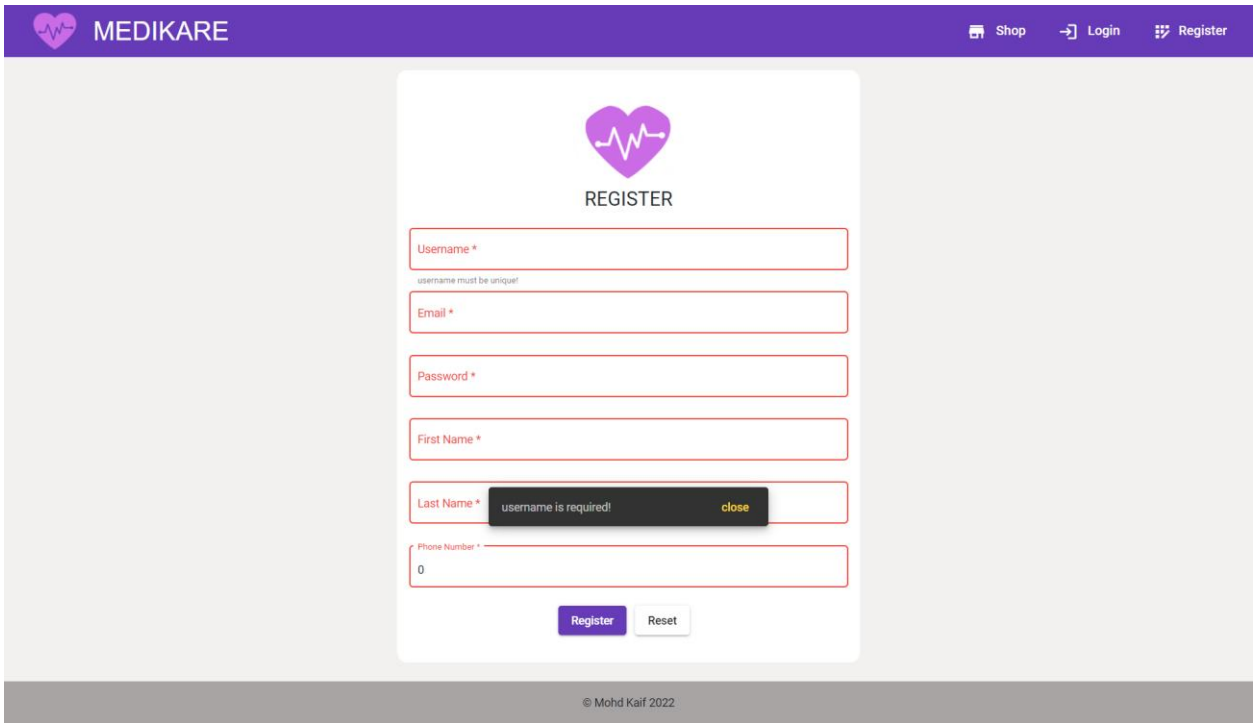
- **product page:**



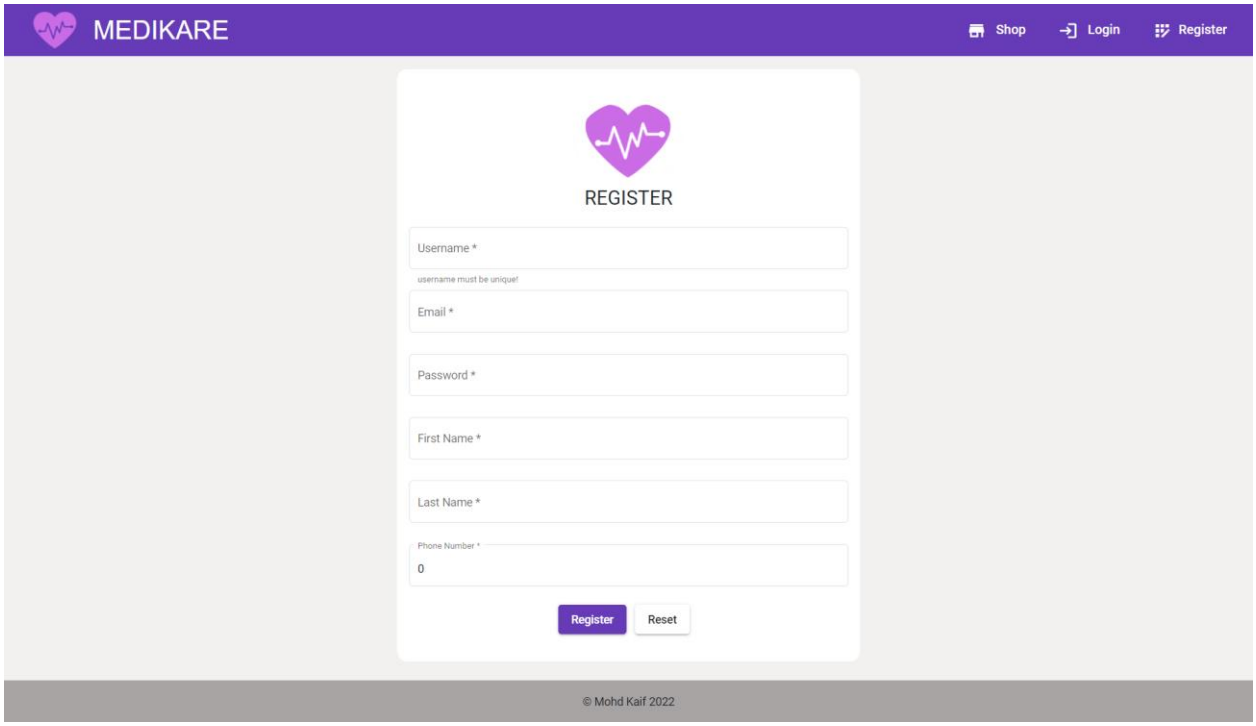
- **purchase validation:**



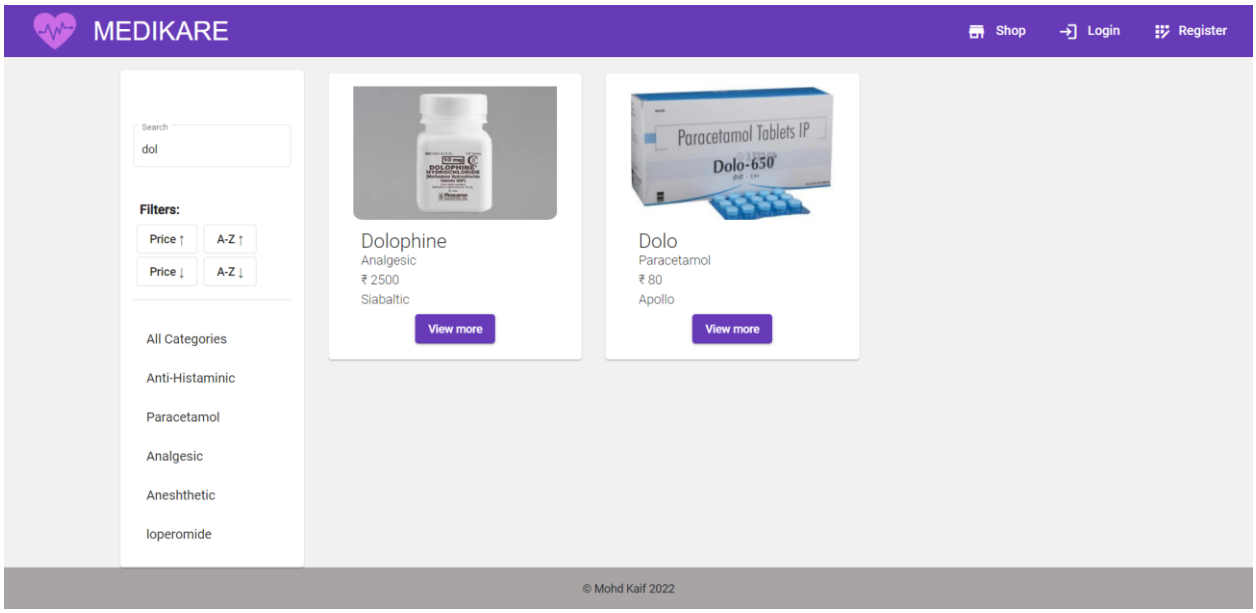
- **register validation:**



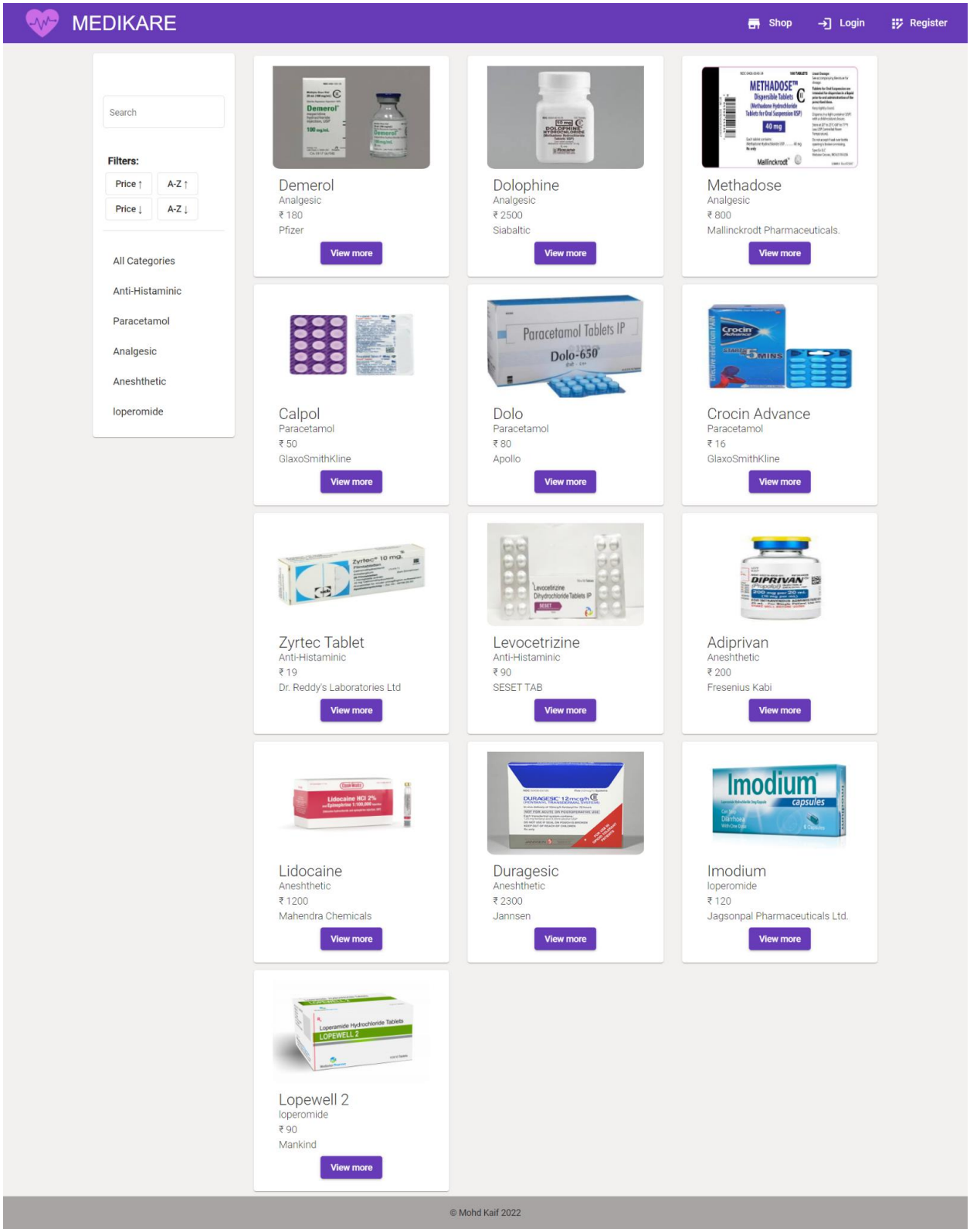
- **register:**



- **search based products :**

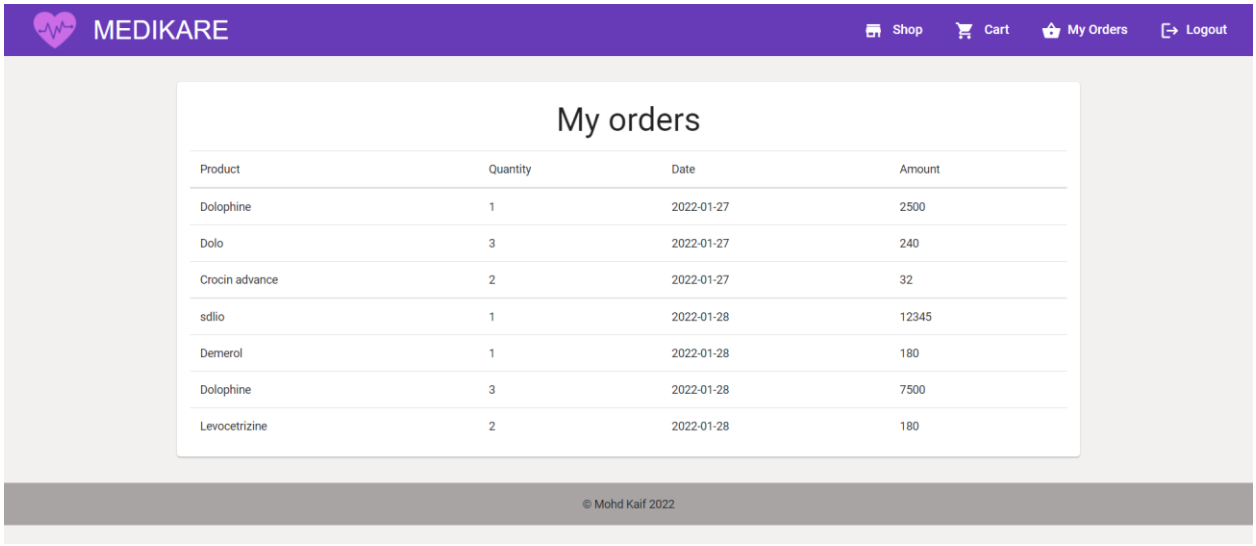


- **shop page:**

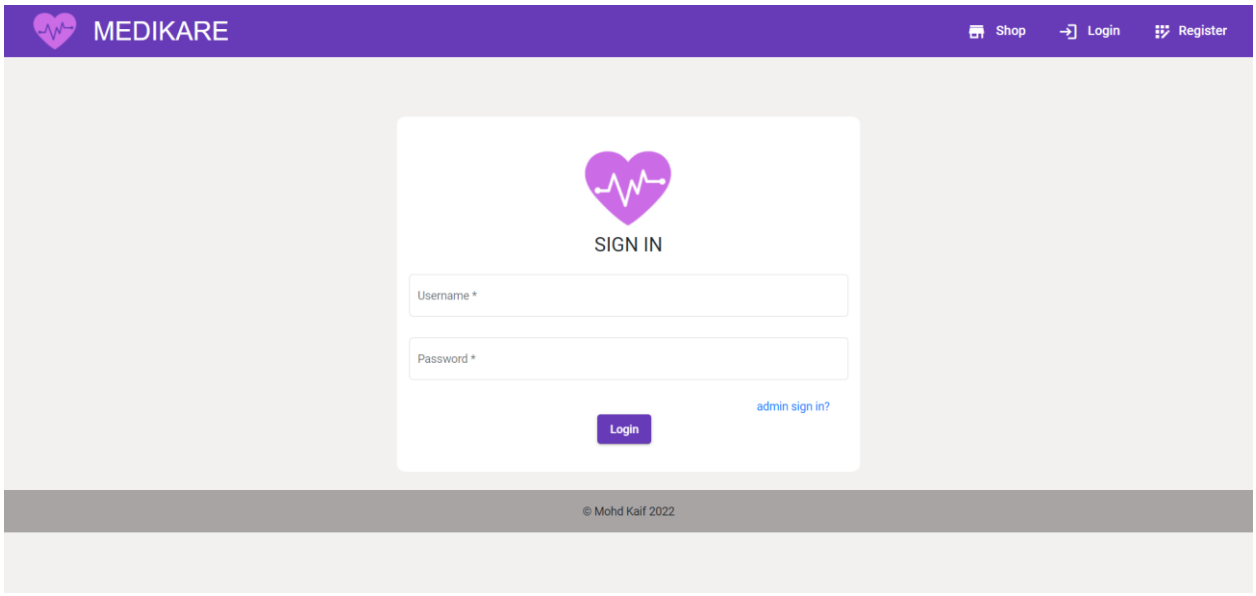


Project By: Pai Najranabanu

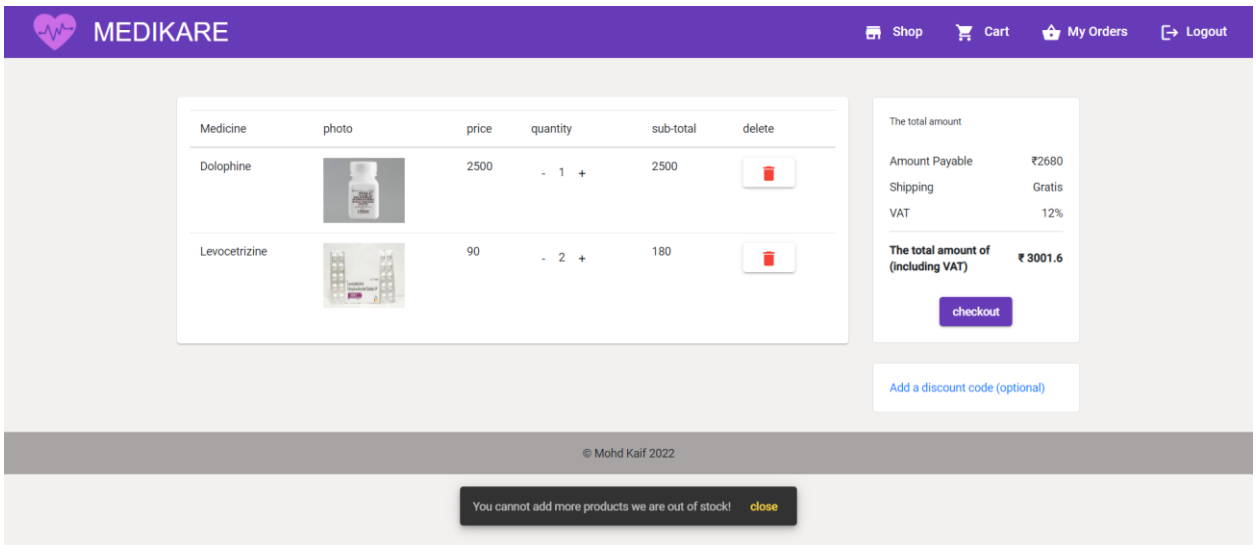
- **user specific purchase details:**



- **user-login:**



- **validation with stock count:**



- **variable quantity cart page:**