PROJECT

SOFTWARE ENGINEERING

MMY STORE (E-commerce with AI bot)

Prepared for :

izdehar Jawabreh

Prepared by :

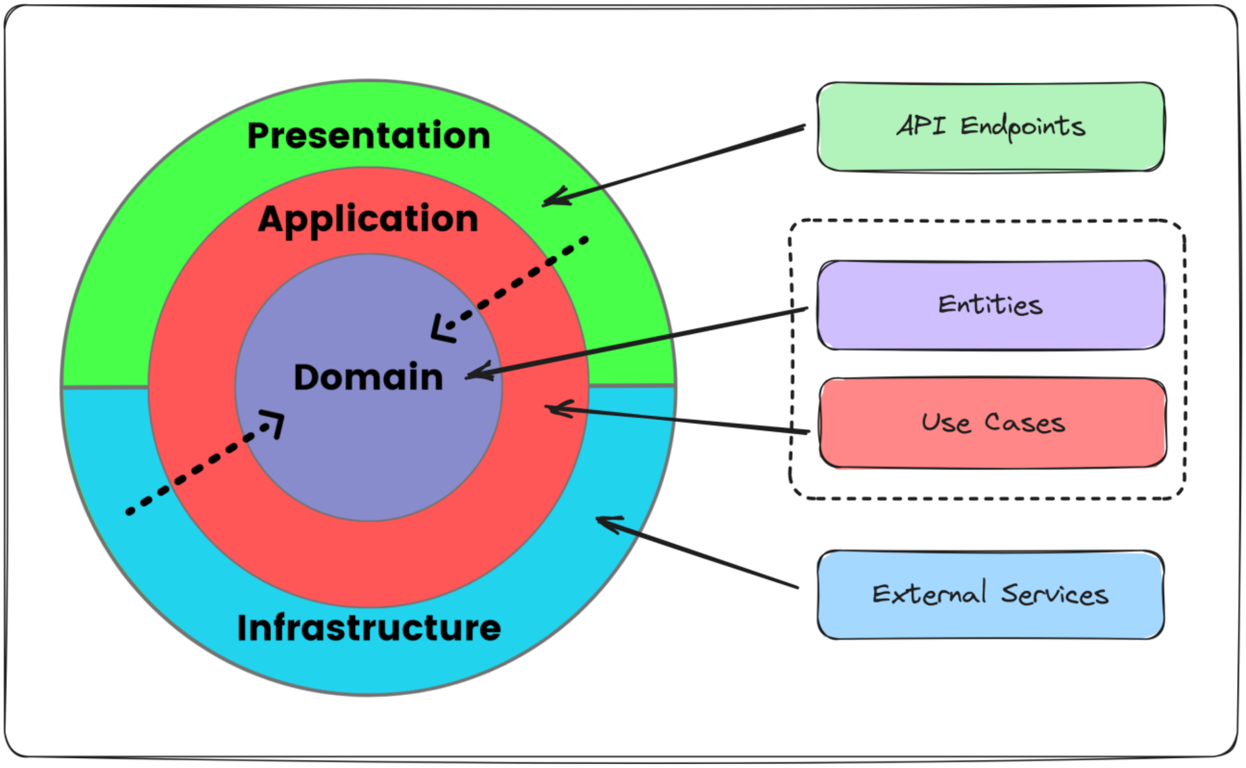
Monjed Tamimi (211069)

Mohammad Nayef (211026)

Yasser Sharabati (211140)

Contents

Architecture of the system:



Source: https://twitter.com/mjovanovictech/status/1684806975899525120/photo/1

The system will use an architectural pattern called Clean Architecture which is a domain-centric architecture where all other layers depend only on the Domain layer.

Clean Architecture mainly has 4 layers:

- **Core layers:**
  1. **Domain**: contains the high-level definition of the entities and abstractions that are shared across the system. This layer has no dependencies on other layers.
  2. **Application**: contains the implementation of the core business logic (use cases) that's defined in the Domain layer.
- **External layers:**
  3. **Infrastructure**: contains the implementation of the external services that are needed for the system which are defined in the Domain layer, like accessing the database or email service.
  4. **Presentation**: contains the implementation of the connection between the system and its clients, like web APIs that can be used to connect the system with a front-end application or a mobile application.

These layers transfer data between each other using Data Transfer Objects (DTOs) that are defined in the Domain layer, which are simple classes that contain fields to hold the data without logic (methods).

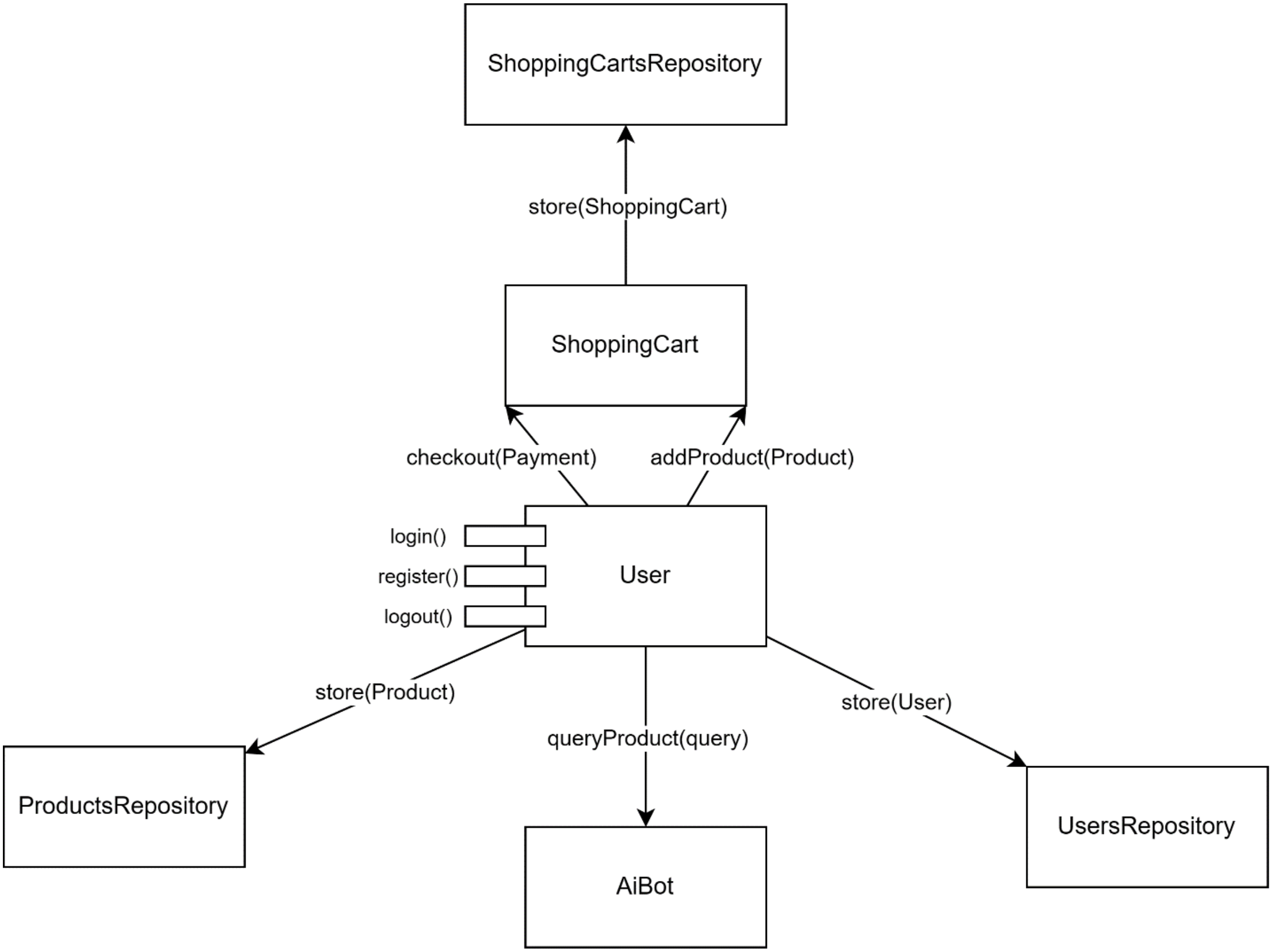
Motivation of using Clean Architecture:

Since the domain requirements as a whole don’t change unless the system needs to be rebuilt again from scratch, while the external services needs to be changed without rebuilding the system since they implement specific service, for example using a NoSQL database instead of SQL or changing the framework that’s used to connect the system with the clients, a domain-centric architecture can be used, so that external layers with low level implementation depend on the high level definitions in the domain layer.

As a result, layers aren’t tightly coupled to each other, and changes in any layer should not lead to changes in other layers, so technologies can be replaced with minimal changes. Also, since the implementation depend on the domain abstractions, it’s easier to test single units or a whole layer without taking care of other layers.

Component-level design:

1. **Classes that correspond to the problem domain**: Product, ShoppingCart, Payment, User.
2. **Classes that correspond to the infrastructure domain**: Payment, AiBot, ProductRepository, UserRepository, ShoppingCartRepository, PaymentRepository.
3. **Classes that are not reusable components**: Product, ShoppingCart, ProductsRepository, ShoppingCartRepository.
3. a. & b. **Collaboration diagram for message details of classes collaborations**:

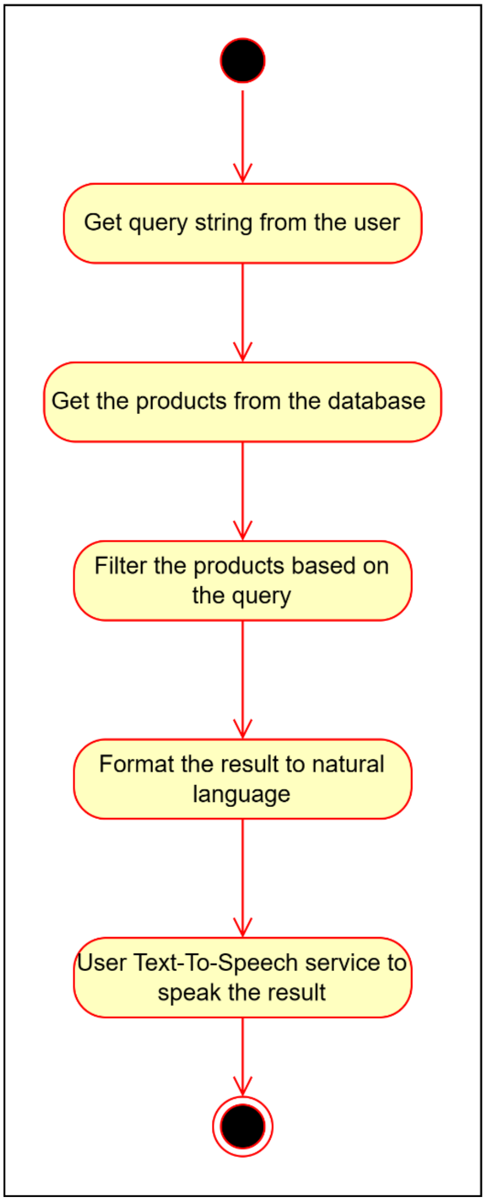


3. c. **Attributes and their datatypes**:

Attribute	Class	Data type
id	User	Int
name	User	String
username	User	String
password	User	String
userType	User	String
id	Product	Int
name	Product	String
quantity	Product	Int
price	Product	Float
warranty	Product	Date
description	Product	String
products	ShoppingCart	Array<Product>
cardId	Payment	Int
postalCode	Payment	String
location	Payment	String
phoneNumber	Payment	String
products	ProductRepository	Array<Product>
shoppingCarts	ShoppingCartRepository	Array<Array<ShoppingCart>>
payments	PaymentRepository	Array<Payment>
users	UserRepository	Array<User>

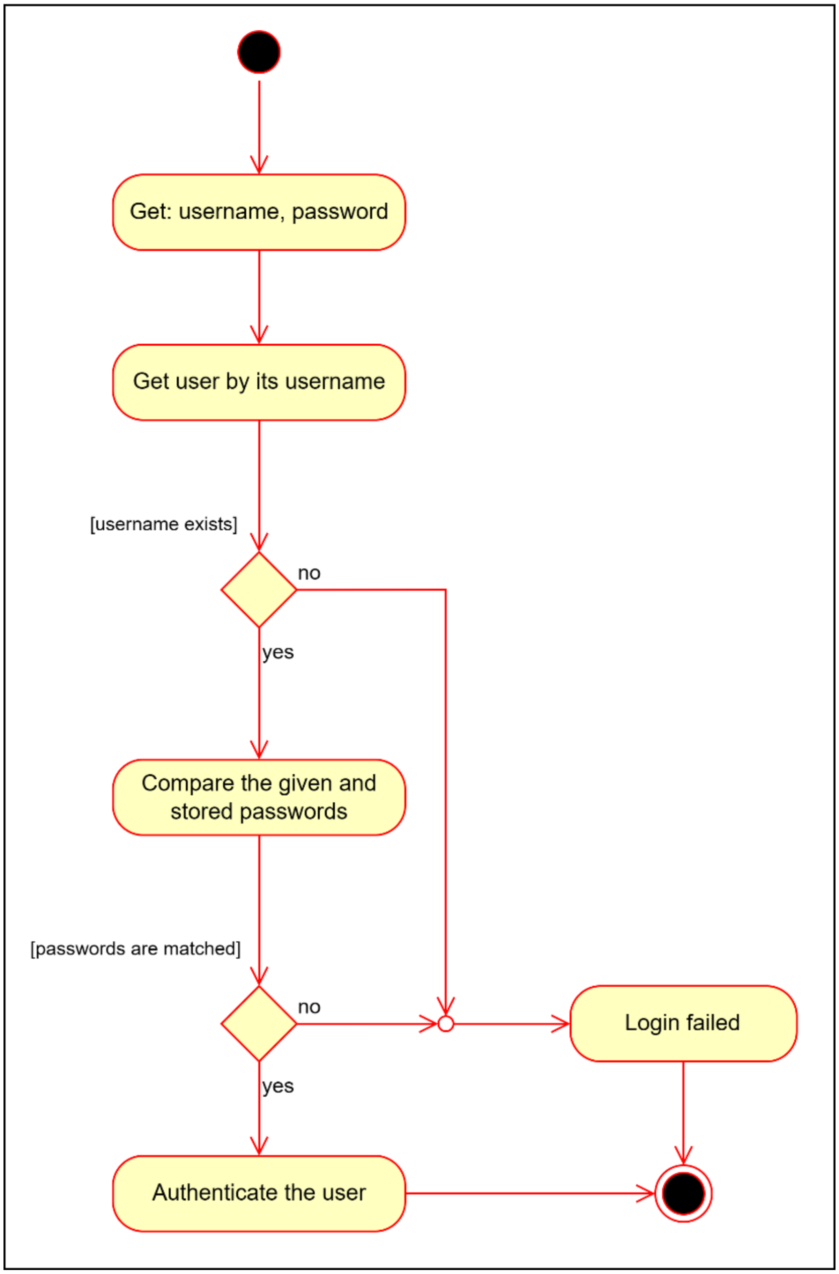
3. d. **Processing flow within each operation**:

AI Bot Query Process Flow

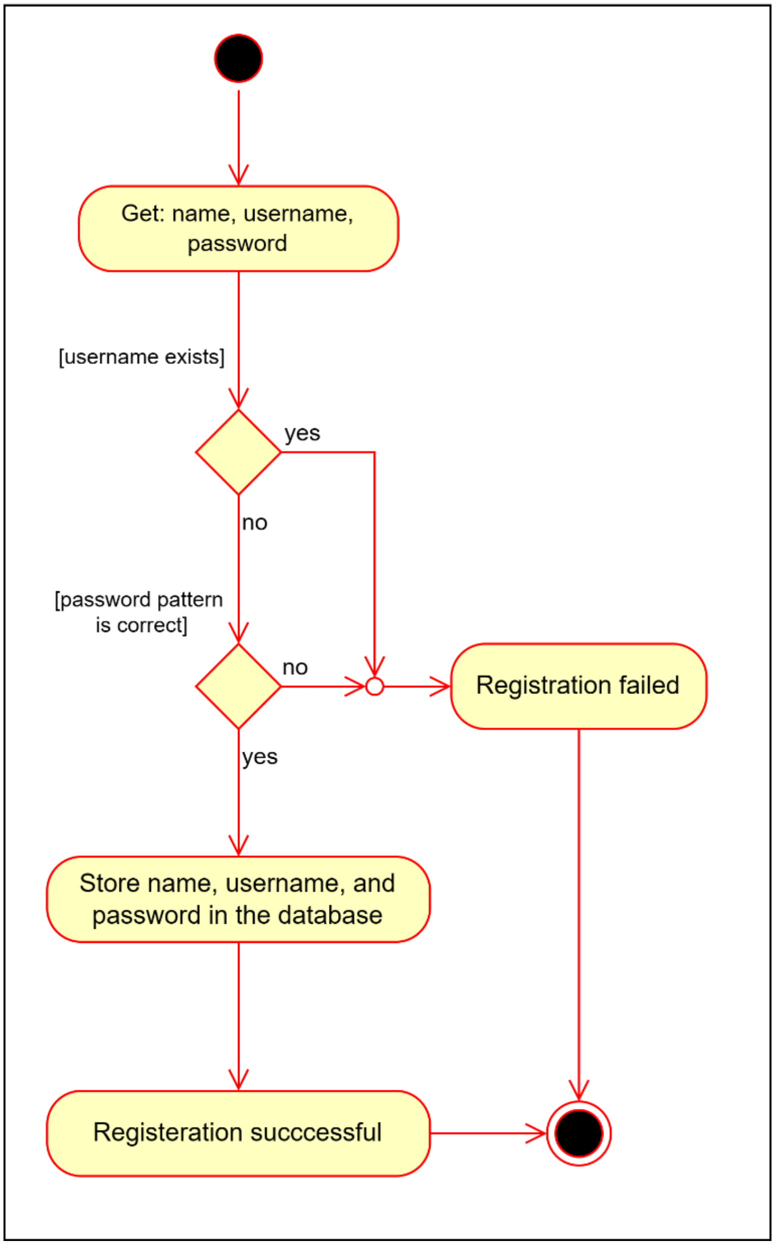


Get query string from the user
Get the products from the database
Filter the products based on the query
Format the result to natural language
User Text-To-Speech service to speak the result

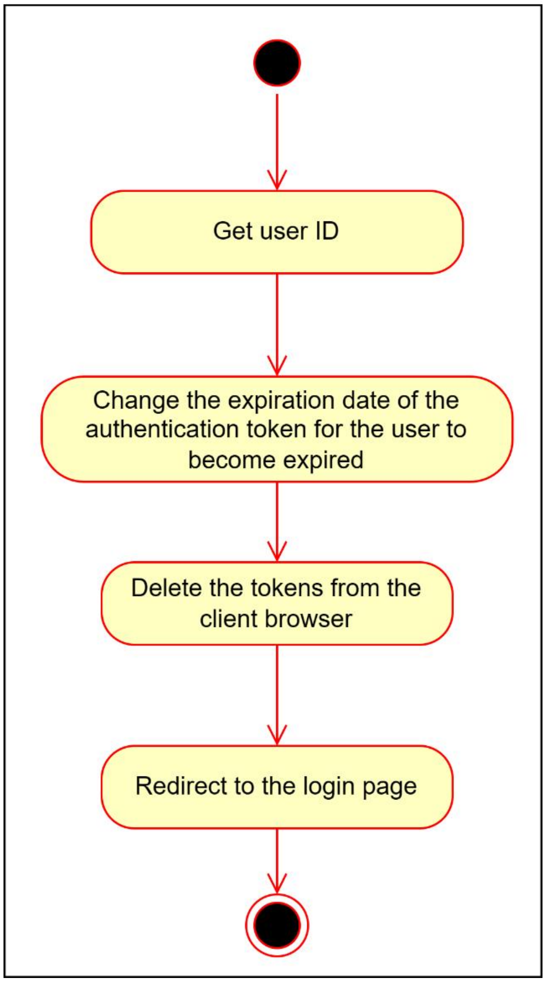
User Login Process Flow



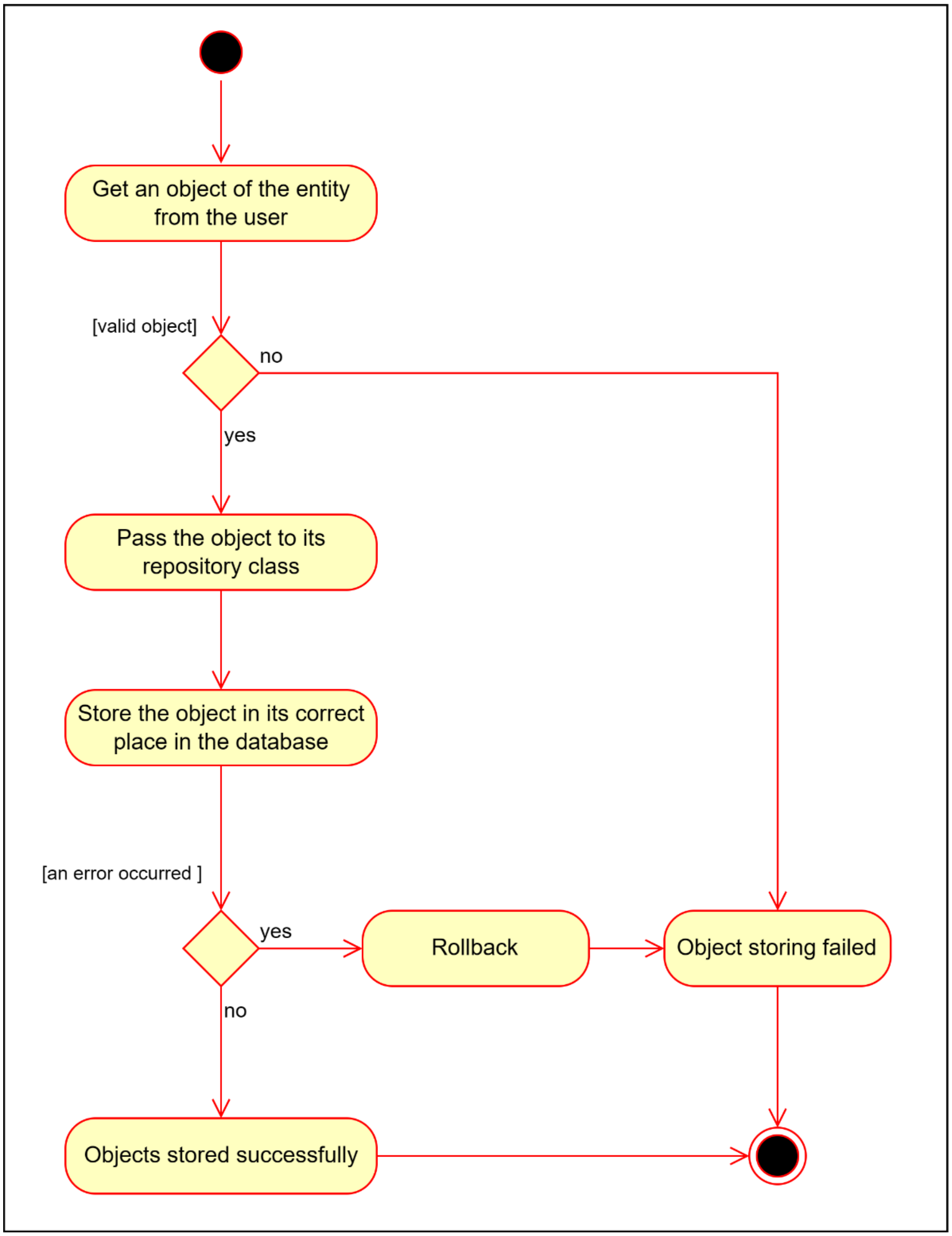
User Registration Process Flow


Get: name, username, password
[username exists]
yes
no
[password pattern is correct]
no
Registration failed
yes
Store name, username, and password in the database
Registeration succcessful

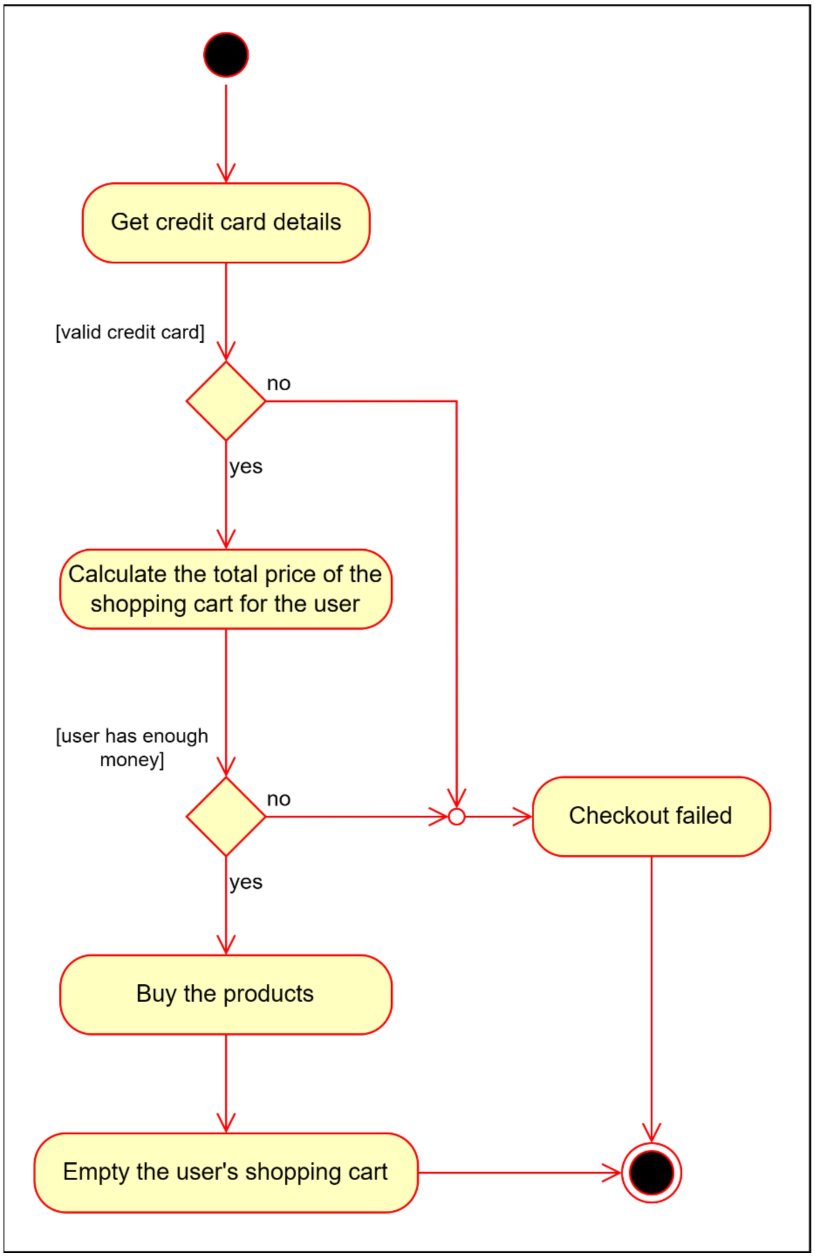
User Logout Process Flow


Get user ID
Change the expiration date of the authentication token for the user to become expired
Delete the tokens from the client browser
Redirect to the login page

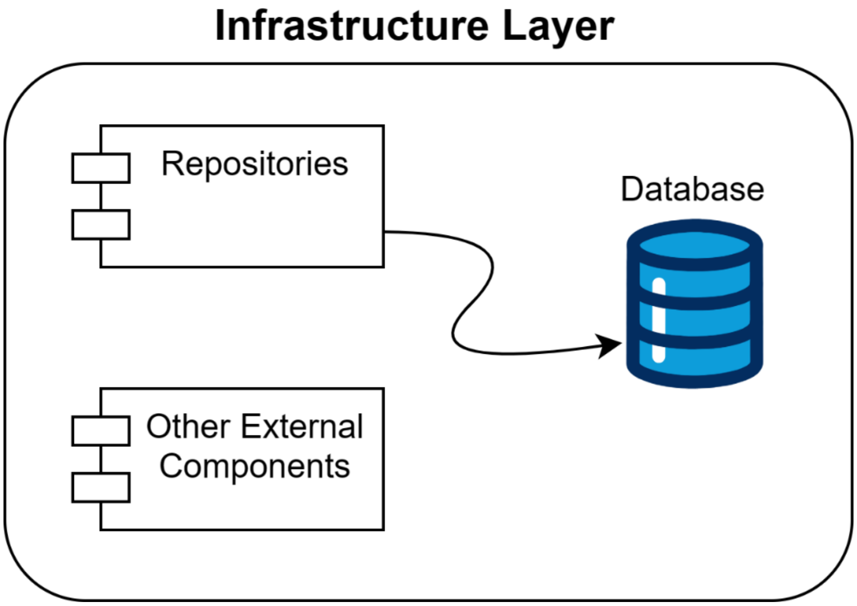
Entities Storing Process Flow


Get an object of the entity from the user
[valid object]
no
yes
Pass the object to its repository class
Store the object in its correct place in the database
[an error occurred]
yes
Rollback
Object storing failed
no
Objects stored successfully

Checkout Process Flow



4. **Persistent data sources**: the system shall should have a relational database to store and retrieve data. This layer exists in the infrastructure layer of the system, which contain repository classes (UserRepository, ProductRepository, ShoppingCartRepository, PaymentRepository) to abstract away the queries and use these repositories with database independent interfaces.



Database Design:

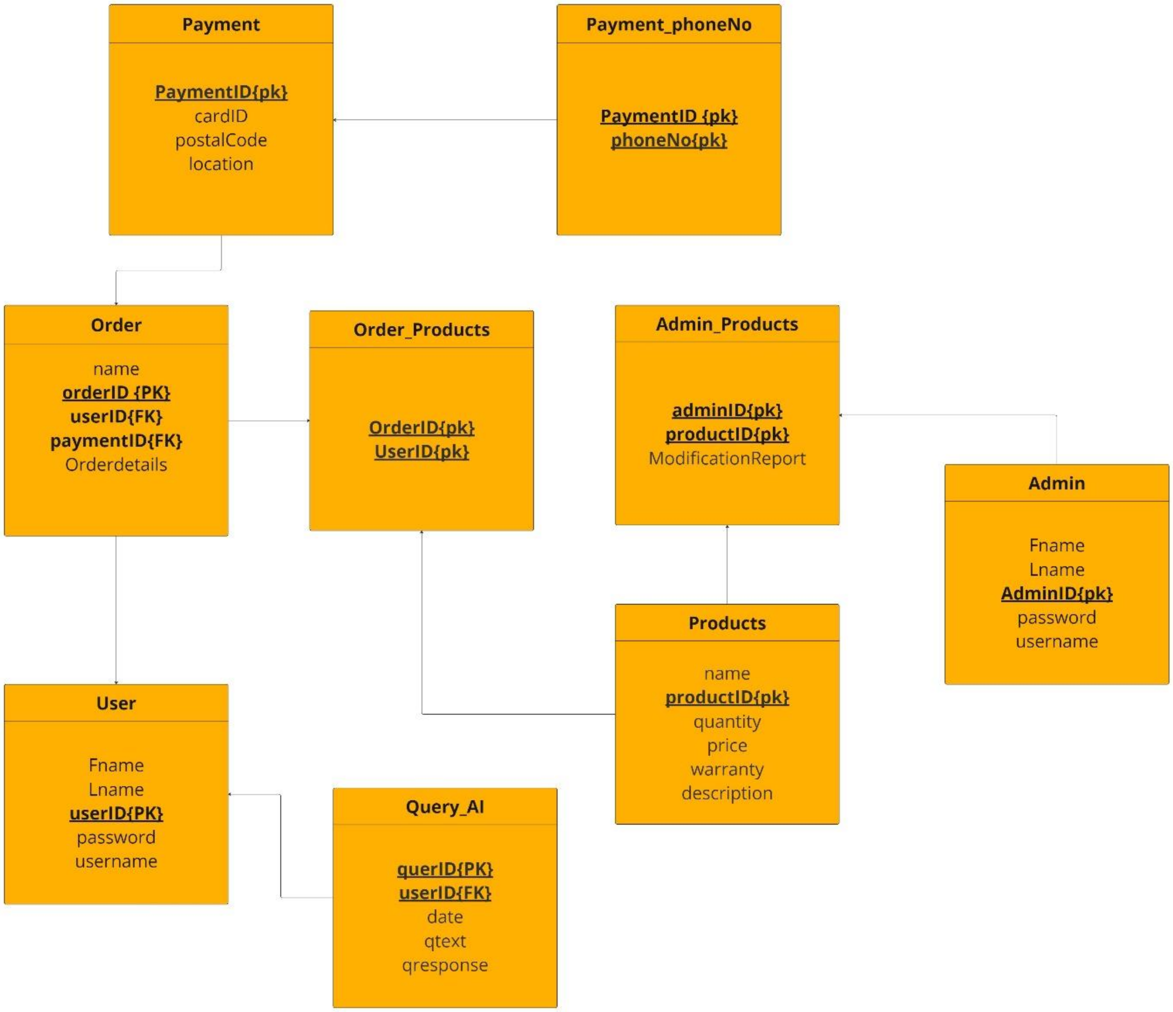
Database Description:

In our database design, we have six main classes. The first is the **User class**, which is used for user registration on our website. The second class is **Admin**, which is used for admin registration and grants access to the inventory. The **Products class** contains detailed information about each product in the inventory. The **Payment class** is used to store details about user payments, such as card numbers, postal code and any necessary info. The **Order class** enables users to place orders after selecting products. Lastly, we have the **Query AI class**, which represents the AI on our website that allows users to query product information. This setup covers user and admin registration, product management, order placement, payment processing, and user queries.

Relations Description:

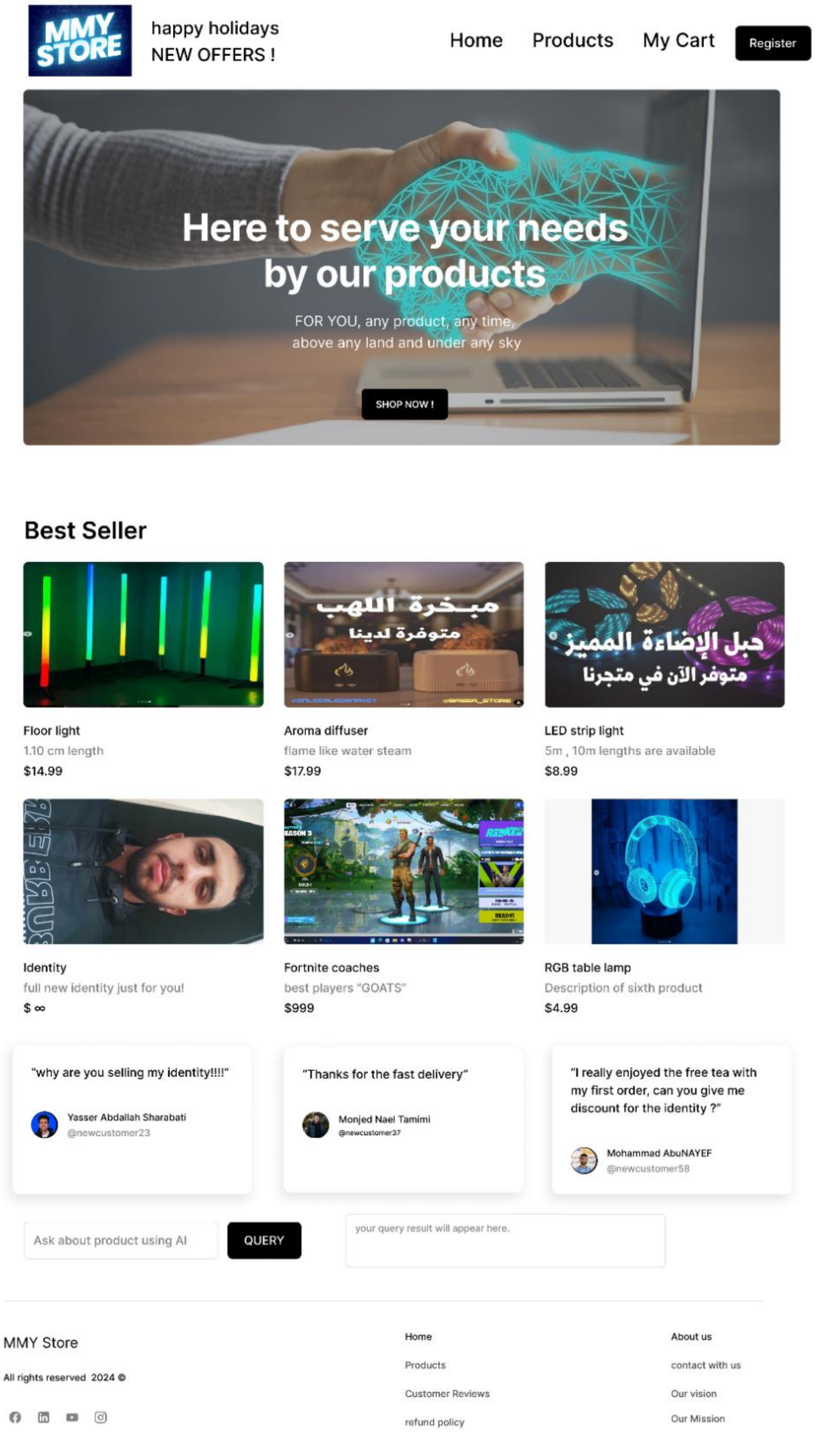
- User_Order --> 1 - ∞ (user can make multiple orders and order is being made by one user)
- User_QuerAI --> 1 - ∞ (user can make multiple queries for AI, and query is being made by one user
- Order_Payment --> 1 - 1(Order has one Payment method and Payment method is used for one Order)
- Order_Products --> ∞ - ∞ (Order has multiple products, and each product can be in multiple orders)
- Admin_Products --> ∞ - ∞ (Admin can CRUD multiple products and products is being CRUDed by multiple Admins) {CRUD : Create, Read, Write, Delete}

DataBase logical design after mapping and normalization



User Interface Design:

Screen 1: Home page



happy holidays
NEW OFFERS !

Here to serve your needs by our products

FOR YOU, any product, any time,
above any land and under any sky

Best Seller

Floor light
1.10 cm length
$14.99

Aroma diffuser
flame like water steam
$17.99

LED strip light
5m , 10m lengths are available
$8.99

Identity
full new identity just for you!
$ ∞

Fortnite coaches
best players "GOATS"
$999

RGB table lamp
Description of sixth product
$4.99

"why are you selling my identity!!!!"
Yasser Abdallah Sharabati
@newcustomer23

"Thanks for the fast delivery"
Monjed Nael Tamimi
@newcustomer37

"I really enjoyed the free tea with my first order, can you give me discount for the identity ?"
Mohammad AbuNAYEF
@newcustomer58

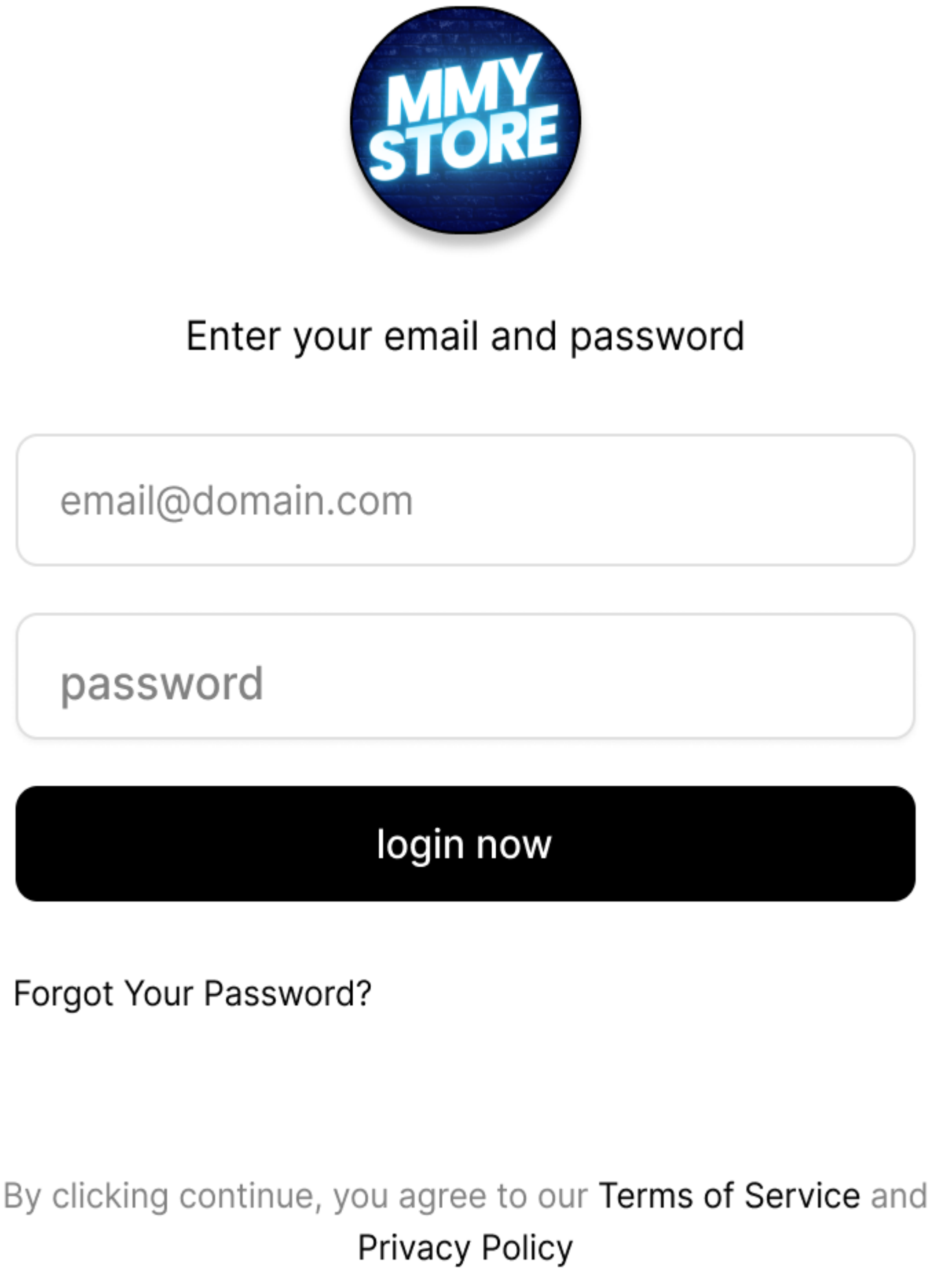
Ask about product using AI

your query result will appear here.

MMY Store

All rights reserved 2024 ©

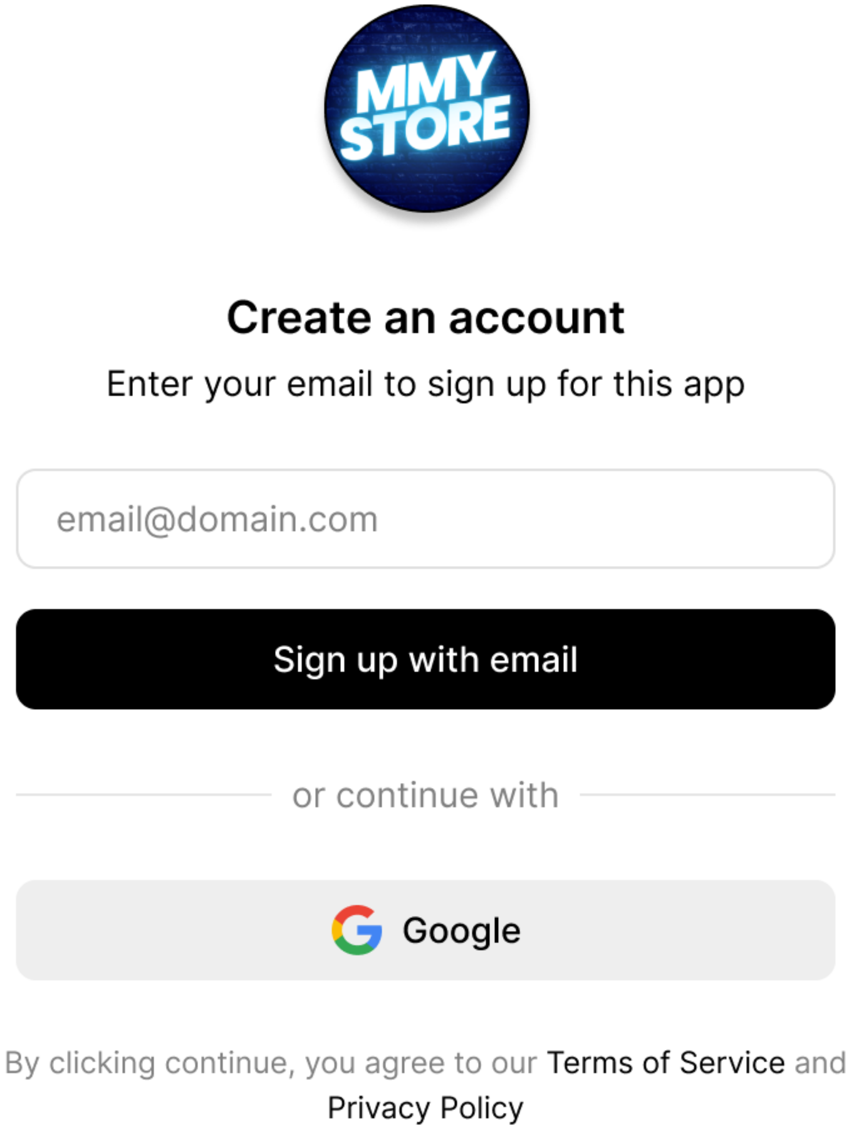
Screen 2: Log in



Enter your email and password

By clicking continue, you agree to our Terms of Service and Privacy Policy

Screen 3: Registration

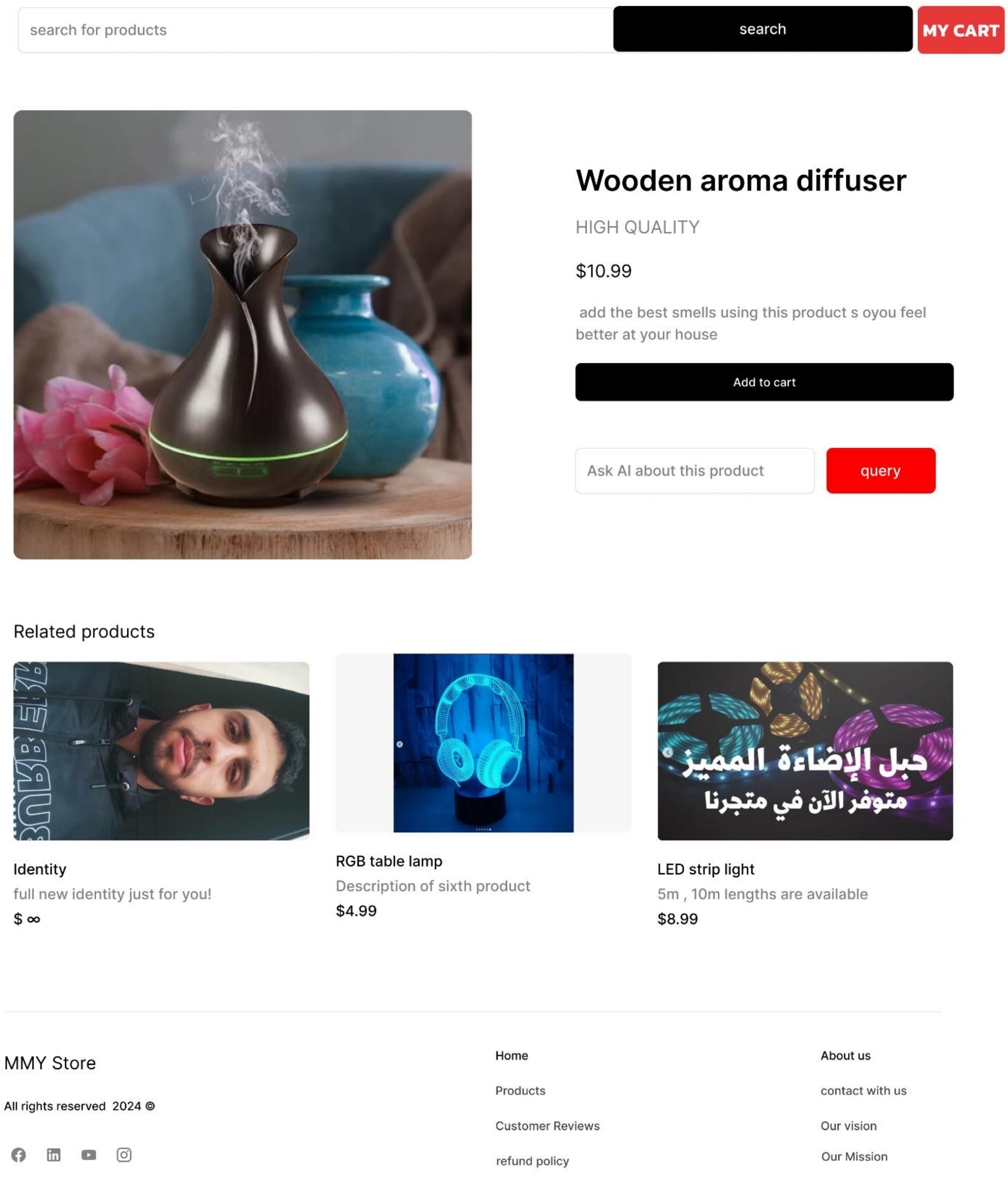


Create an account

Enter your email to sign up for this app

By clicking continue, you agree to our Terms of Service and Privacy Policy

Screen 4: Product details page



search for products

Wooden aroma diffuser

HIGH QUALITY

$10.99

add the best smells using this product s oyou feel better at your house

Ask AI about this product

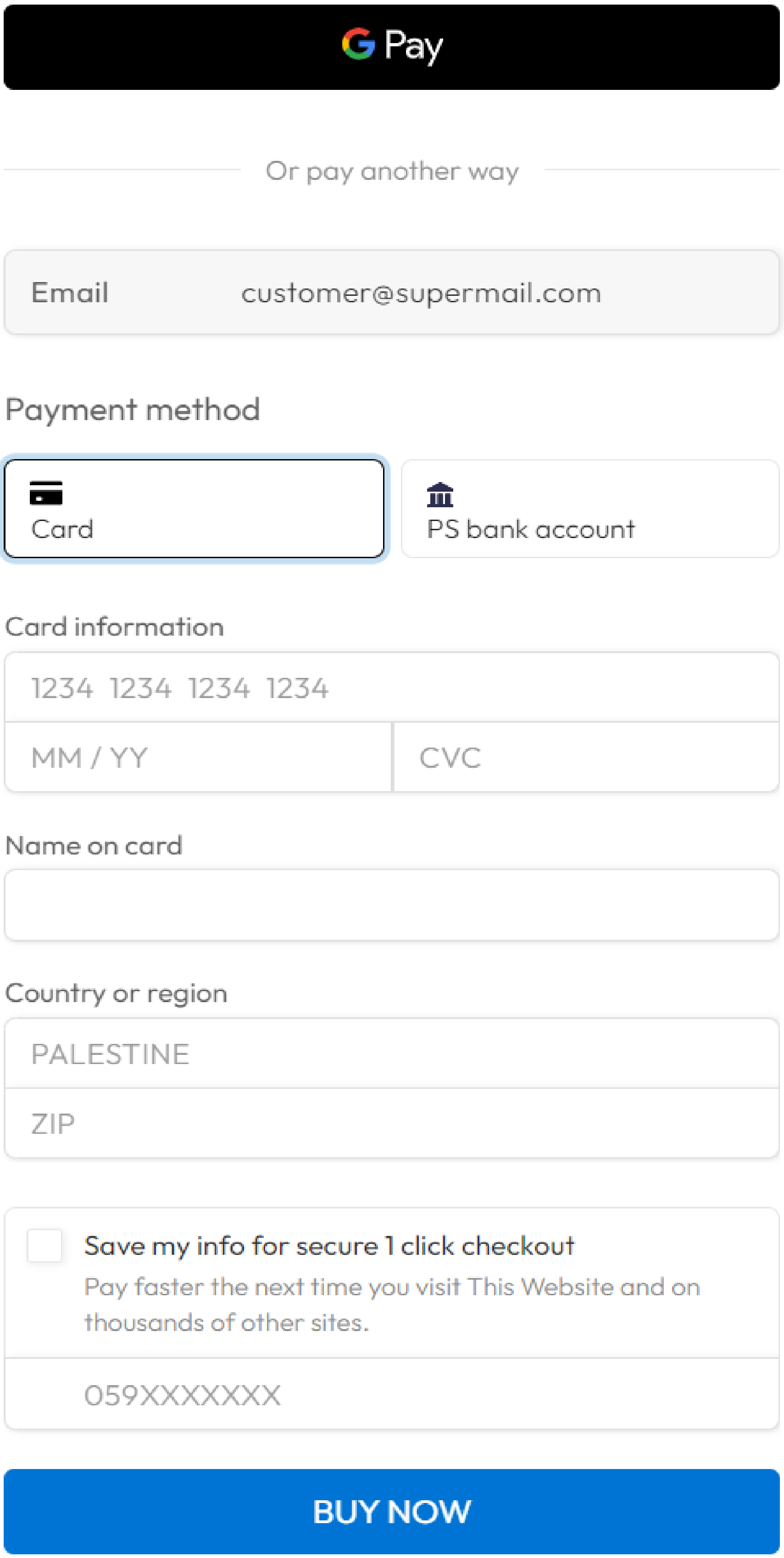
Related products

Identity	RGB table lamp	LED strip light
full new identity just for you!	Description of sixth product	5m , 10m lengths are available
$ ∞	$4.99	$8.99

MMY Store

All rights reserved 2024 ©

Screen 5: Payment checkout



G Pay

Or pay another way

Email customer@supermail.com

Payment method

Card

PS bank account

Card information

1234 1234 1234 1234

MM / YY

CVC

Name on card

Country or region

PALESTINE

ZIP

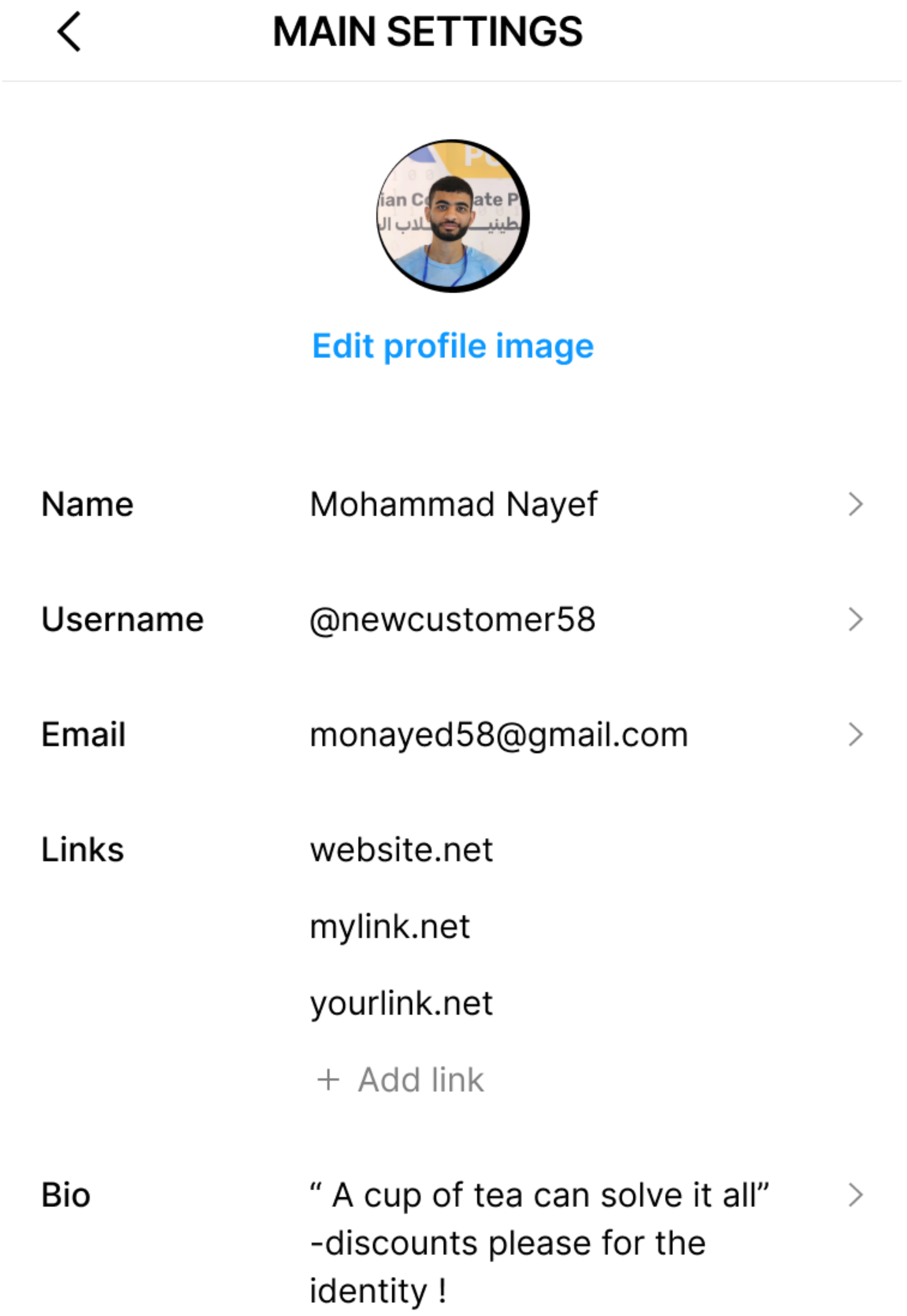
□ Save my info for secure 1 click checkout

Pay faster the next time you visit This Website and on thousands of other sites.

059XXXXXXX

BUY NOW

Screen 6: Profile settings

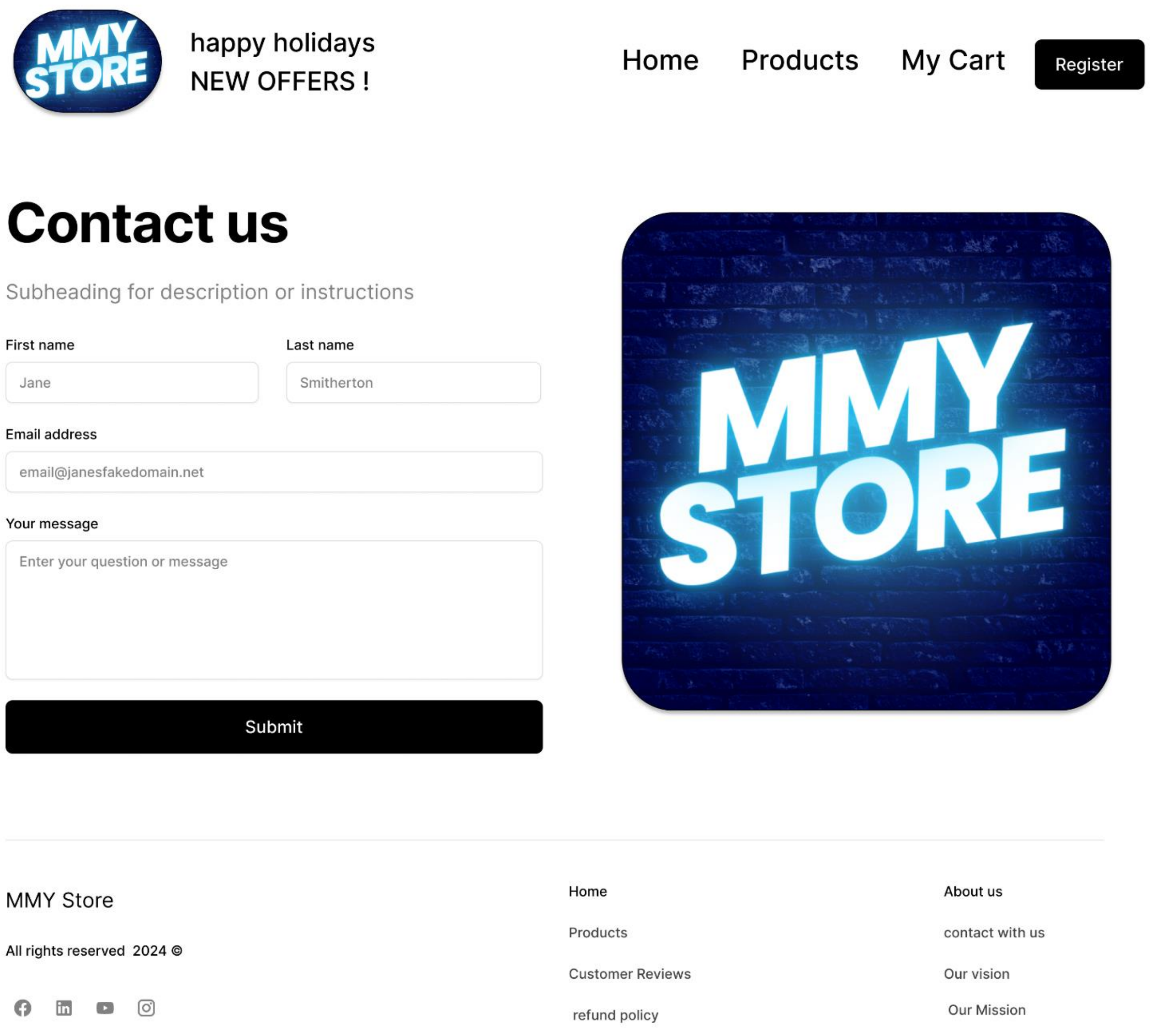


MAIN SETTINGS

Edit profile image

Name	Mohammad Nayef
Username	@newcustomer58
Email	monayed58@gmail.com
Links	website.net mylink.net yourlink.net + Add link
Bio	“ A cup of tea can solve it all” -discounts please for the identity !

Screen 7: Contact us



happy holidays
NEW OFFERS !

Contact us

Subheading for description or instructions

First name
Jane

Last name
Smitherton

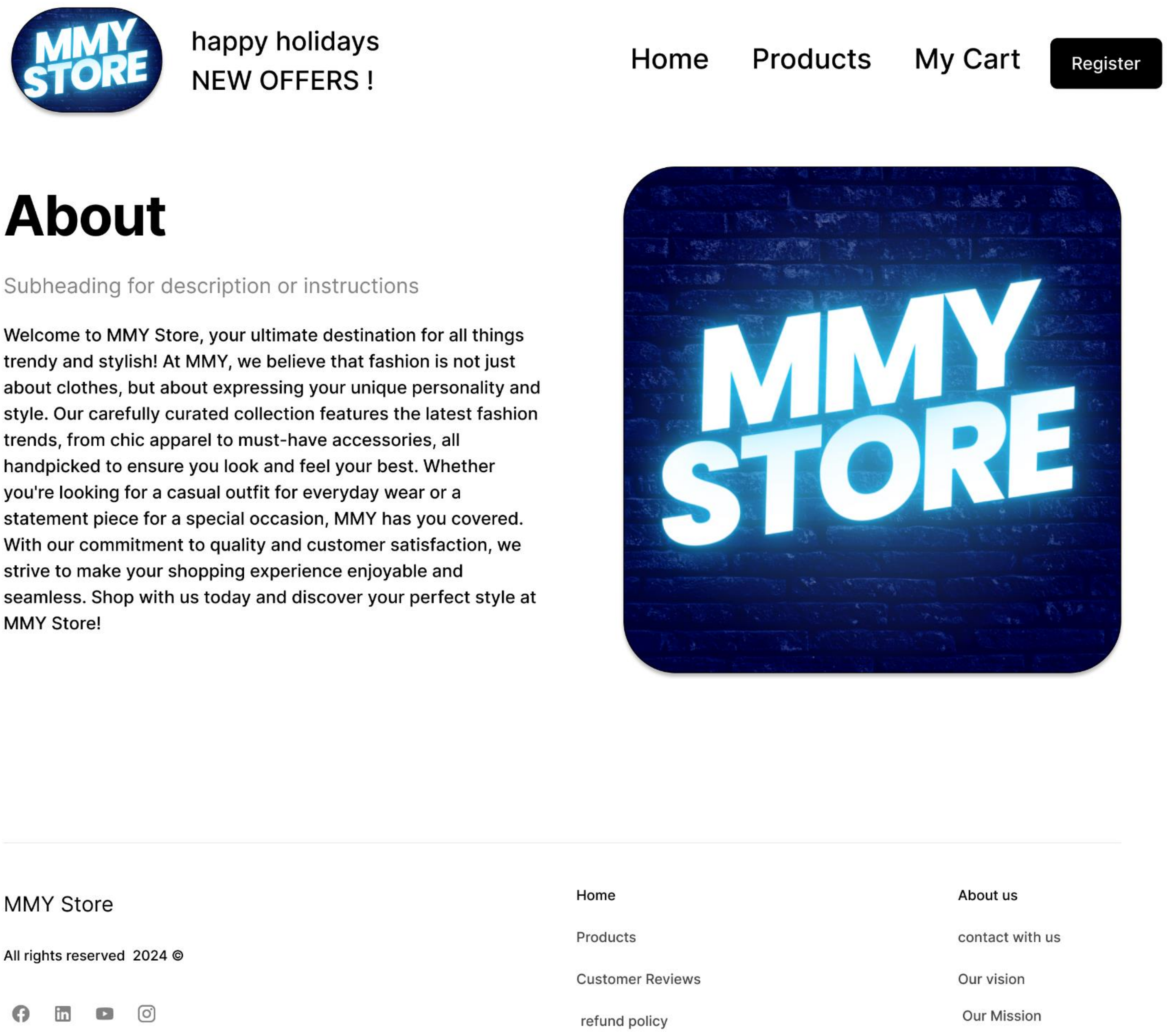
Email address
email@janesfakedomain.net

Your message
Enter your question or message

MMY Store

All rights reserved 2024 ©

Screen 8: About us



About

Subheading for description or instructions

Welcome to MMY Store, your ultimate destination for all things trendy and stylish! At MMY, we believe that fashion is not just about clothes, but about expressing your unique personality and style. Our carefully curated collection features the latest fashion trends, from chic apparel to must-have accessories, all handpicked to ensure you look and feel your best. Whether you're looking for a casual outfit for everyday wear or a statement piece for a special occasion, MMY has you covered. With our commitment to quality and customer satisfaction, we strive to make your shopping experience enjoyable and seamless. Shop with us today and discover your perfect style at MMY Store!

MMY Store

All rights reserved 2024 ©