Imp

{% load static %}

<style>

header {

display: flex;

align-items: center;

padding: 20px 100px;

background-color: #ffffff;

}

.logo {

font-size: 24px;

font-weight: bold;

margin-right: 30px;

}

nav {

flex: 1;

}

nav ul {

display: flex;

list-style: none;

}

nav ul li {

margin-right: 15px;

}

nav ul li a {

color: #000000;

text-decoration: none;

text-transform: uppercase;

font-size: 14px;

cursor: pointer;

}

.header-right {

display: flex;

align-items: center;

}

.header-right i {

margin-left: 20px;

font-size: 18px;

cursor: pointer;

color: #000000;

}

/\* Hero section \*/

.hero {

display: flex;

justify-content: space-between;

align-items: center;

padding: 80px 100px;

}

.hero-content {

max-width: 500px;

}

.hero-content h1 {

font-size: 36px;

margin-bottom: 20px;

line-height: 1.2;

color: rgb(7, 108, 236);}

.hero-content p {

font-size: 16px;

color: #666666;

margin-bottom: 30px;

}

.hero-image img {

max-width: 300px;

}

/\* Search form \*/

.search-container {

margin-top: 20px;

display: flex;

}

.search-input {

flex: 1;

padding: 12px 15px;

border-radius: 10px;

border: 1px solid #cccccc;

background-color: #ffffff;

color: #000000;

margin-right: 10px;

}

.search-button {

padding: 12px 25px;

border-radius: 10px;

border: none;

background-color: #3a3a3a;

color: #ffffff;

cursor: pointer;

}

</style>

<header>

<div class="logo" id="na" onclick="na()">M</div>

<nav>

<ul>

<li><a href="/" class="fa">HOME</a></li>

<li><a href="/details/" class="fa">Product</a></li>

<li><a href="{% url 'home' %}#abt" class="fa">About Us</a></li>

<li><a href="#" class="fa">Map</a></li>

{% if user.is\_authenticated %}

<li ><a href="{% url 'logout' %} " class="fa">Log Out</a></li>

<li ><a href="{% url 'Dhasboard'} " class="fa">Dhasboard</a></li>

{% else %}

<li ><a href="{% url 'login' %} " class="fa">Log In</a></li>

<li ><a href="{% url 'register' %} " class="fa">Sign In</a></li>

{% endif %}

</ul>

</nav><hr>

<div class="header-right">

<i class="fa fa-search"></i>

<i class="fa fa-shopping-cart"></i>

<i class="fa fa-user"></i>

</div>

</header>

<hr>

from django.shortcuts import render, redirect

from django.http import HttpResponse

from .models import Post

from .forms import Login, RegisterForm

from django.core.paginator import Paginator # Corrected import

from django.contrib import messages

from .cart import Cart

from django.contrib.auth import authenticate, login as auth\_login,logout as auth\_logout

from django.http import JsonResponse

from django.shortcuts import get\_object\_or\_404

# Create your views here.

def home(request):

title = "Home" # Corrected variable

return render(request, "home.html", {"title": title})

def product(request):

all\_posts = Post.objects.all()

paginator = Paginator(all\_posts, 4) # Corrected import used

page\_no = request.GET.get('page')

page\_obj = paginator.get\_page(page\_no)

return render(request, "Product.html", {"title": "Product", "page\_obj": page\_obj})

def detail(request, id):

post = Post.objects.get(pk=id)

return render(request, 'detail.html', {"item": post})

def item(request):

posts = Post.objects.all()

return render(request, 'item.html', {"title": "Item", "item": posts})

def category\_view(request, id):

posts = Post.objects.filter(cate\_id=id) # Filter by cate\_id

return render(request, f"{id}.html", {"title": f"Category {id}", "item": posts})

def register(request):

form = RegisterForm()

if request.method == "POST":

form = RegisterForm(request.POST)

if form.is\_valid():

user = form.save(commit=False)

user.set\_password(form.cleaned\_data['password'])

user.save()

messages.success(request,"Succesfully Created ")

return redirect("login")

return render(request, "register.html", {"form": form})

def login(request):

form = Login()

if request.method =="POST":

form = Login(request.POST)

if form.is\_valid():

username = form.cleaned\_data['username']

password = form.cleaned\_data['password']

user = authenticate(username=username,password=password)

if user is not None:

auth\_login(request,user)

return redirect("dashboard")

return render(request,'login.html', {"form": form})

def cart\_add(request):

# Get the cart

cart = Cart(request)

# test for POST

if request.POST.get('action') == 'post':

# Get stuff

product\_id = int(request.POST.get('product\_id'))

# lookup product in DB

product = get\_object\_or\_404(Post, id=product\_id)

# Save to session

cart.add(product=product)

# Return response

response = JsonResponse({'Product Name: ': product.name})

return response

def dashboard (request):

blog\_title="My Orders"

return render(request,"dashboard.html",{"blog\_title":blog\_title})

def logout(request):

auth\_logout(request)

return render(request,"Home.html")

def cart\_add(request):

# Get the cart

cart = Cart(request)

# test for POST

if request.POST.get('action') == 'post':

# Get stuff

product\_id = int(request.POST.get('product\_id'))

# lookup product in DB

product = get\_object\_or\_404(Post, id=product\_id)

# Save to session

cart.add(product=product)

# Return response

response = JsonResponse({'Product Name: ': product.name})

return response

## **6. RESULT AND DISCUSSION**

### **6.1 Admin Portal**

The Admin Portal plays a crucial role in managing the overall functionality of the online medicine ordering platform. It allows the admin to monitor user activity, manage product listings, track orders, verify pharmacy partners, and ensure smooth operation of the system. The portal also provides analytics and insights regarding customer preferences, most-ordered medicines, and delivery performance. With secure access and role-based controls, the Admin Portal ensures efficient backend management for a seamless user experience.

### **6.2 Student Portal**

The Student Portal serves as the front-facing platform for users, particularly targeting students and young adults who require easy and quick access to healthcare products. Inspired by the functionality and ease-of-use of platforms like Zomato, the portal enables users to browse and order medicines online with just a few clicks. Key features include:

* **User-Friendly Interface**: Simplified navigation and search functionality for finding medicines and wellness products.
* **Trusted Pharmacies**: Products are sourced only from verified and certified pharmacies.
* **24/7 Availability**: Users can place orders at any time, ensuring round-the-clock access to medicines.
* **Fast Delivery**: Orders are fulfilled quickly and efficiently, minimizing wait times.
* **Secure Transactions**: Ensures safe and encrypted payment processing for users.

This system is designed to make healthcare more accessible and convenient, especially for students and individuals who prefer online solutions for their daily needs. It bridges the gap between pharmacies and end-users, delivering essential medical products right to their doorstep with minimal effort.

Here's a well-written version of the **Conclusion** and **References** sections for your project report:

## **7. CONCLUSION**

The development of the online medicine ordering website represents a significant step toward digitizing healthcare access. By creating a platform that mirrors the convenience and efficiency of food delivery services like Zomato, we have simplified the process of purchasing essential healthcare and wellness products. The portal not only addresses the need for quick and reliable medicine delivery but also ensures safety, user-friendliness, and trust through verified pharmacies and secure operations.

With features like real-time order tracking, 24/7 accessibility, and fast delivery, the system offers an efficient and dependable solution for both urban and rural populations. The Admin and Student portals together establish a comprehensive framework for managing users, orders, and medical inventory. This platform has the potential to revolutionize how users interact with pharmacies, making medicine ordering seamless and more accessible than ever.

## **REFERENCES**

[1] World Health Organization. (2021). *Digital Health and Telemedicine Guidelines*. WHO Press.

[2] Pharmacy Times. (2020). *Growth of Online Pharmacy in India*. Retrieved from [https://www.pharmacytimes.com](https://www.pharmacytimes.com/)

[3] Ministry of Health and Family Welfare. (2020). *Guidelines for E-Pharmacies in India*. Government of India Publication.

[4] Zomato. (2021). *Case Study on Digital Delivery Systems*. Zomato Business Insights.

[5] Django Software Foundation. (2023). *Django Documentation*. Retrieved from [https://docs.djangoproject.com](https://docs.djangoproject.com/)

[6] Bootstrap. (2023). *Bootstrap Official Documentation*. Retrieved from [https://getbootstrap.com](https://getbootstrap.com/)

[7] W3Schools. (2023). *HTML, CSS, and JavaScript Tutorials*. Retrieved from [https://www.w3schools.com](https://www.w3schools.com/)

[8] Mozilla Developer Network. (2023). *Web Technologies Documentation*. Retrieved from [https://developer.mozilla.org](https://developer.mozilla.org/)

[9] Deshmukh, A., & Joshi, M. (2022). Predictive analytics in HR: A case study approach. *Journal of Data-Driven HR*, 7(2), 18–27.

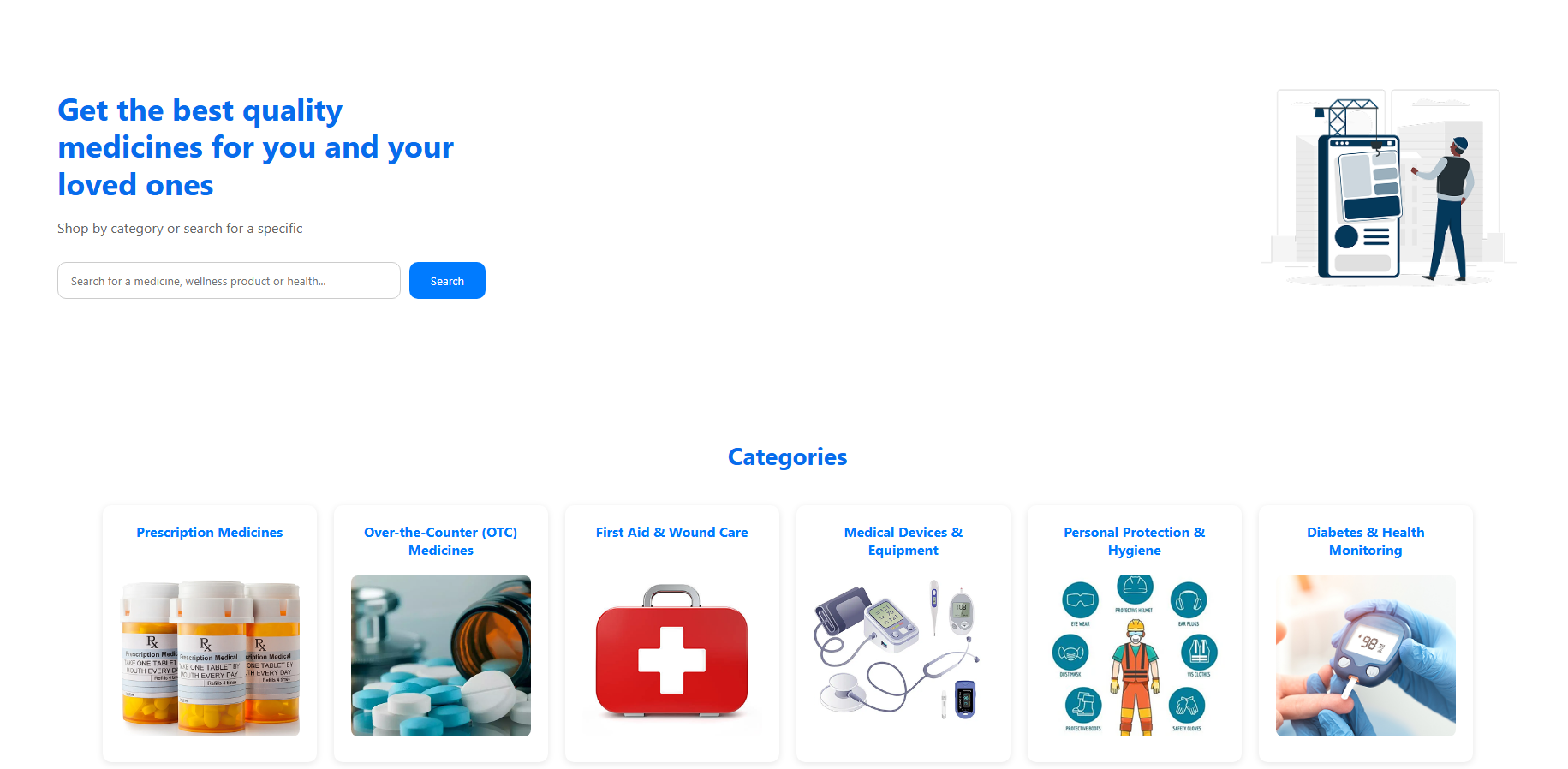
[10] Ahmad, F., & Malik, L. (2020). AI-based recruitment systems: Opportunities and ethical challenges. *Human Resource Review*, 30(3), 101–110.

[11] Zhao, W., & Li, X. (2021). Natural language processing for resume screening: A review. *International Journal of Artificial Intelligence in HR*, 5(1), 33–45.

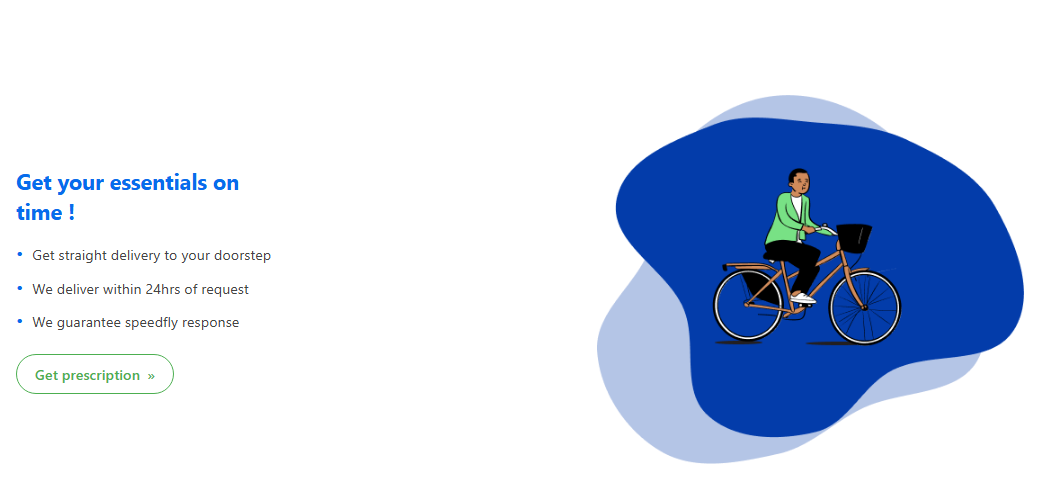
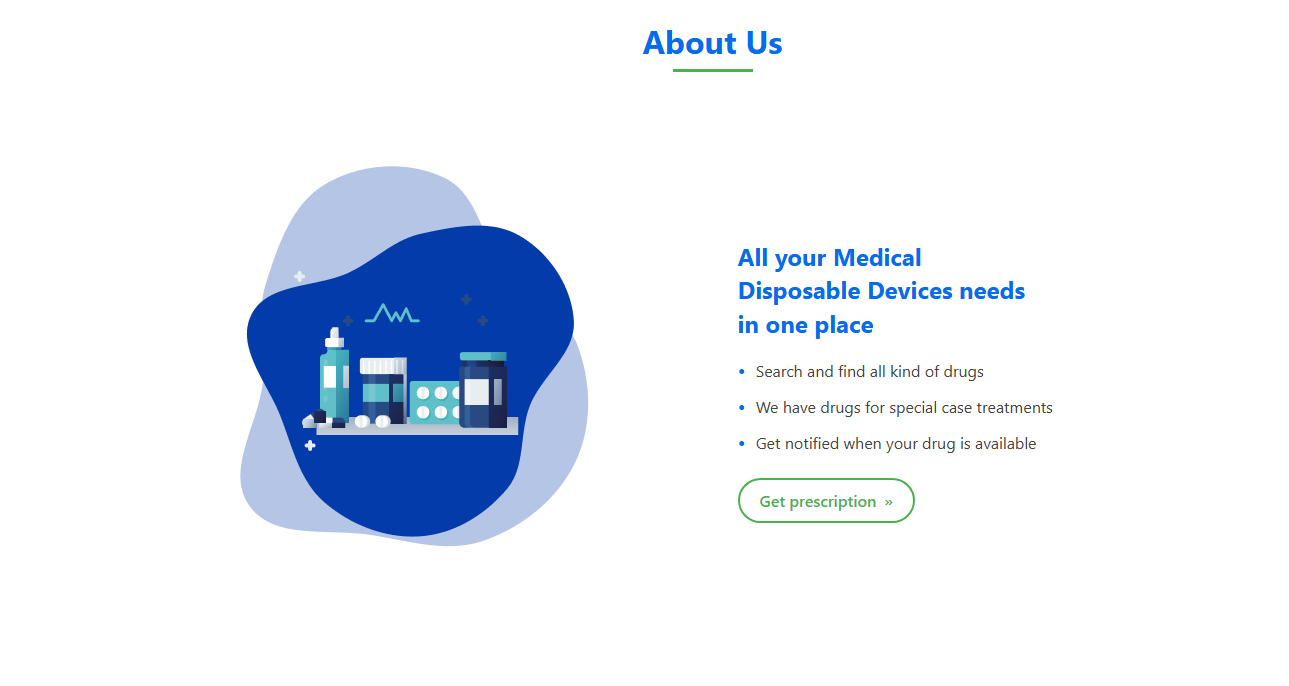
[12] Mehra, N., & Sinha, D. (2021). AI-powered performance management tools: A review. *International Journal of HR Analytics*, 3(3), 59–70.

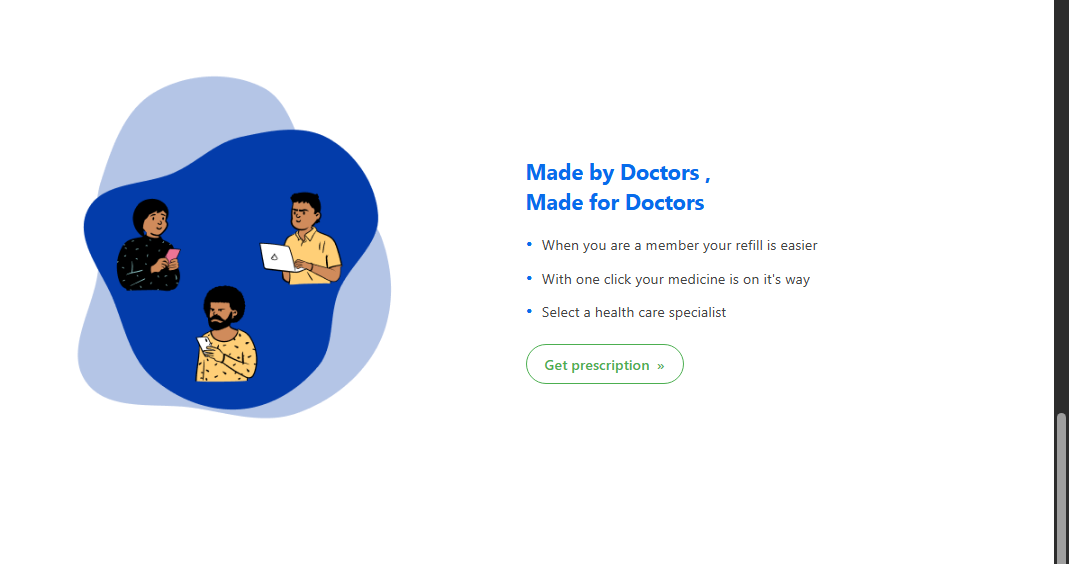
Image

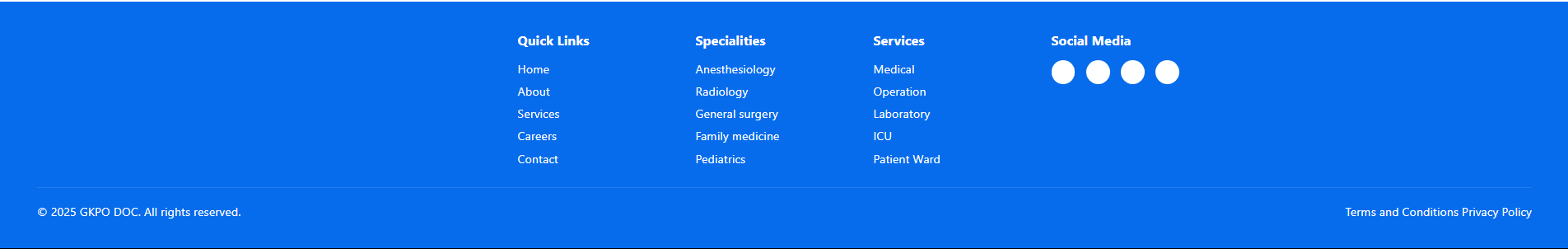
home



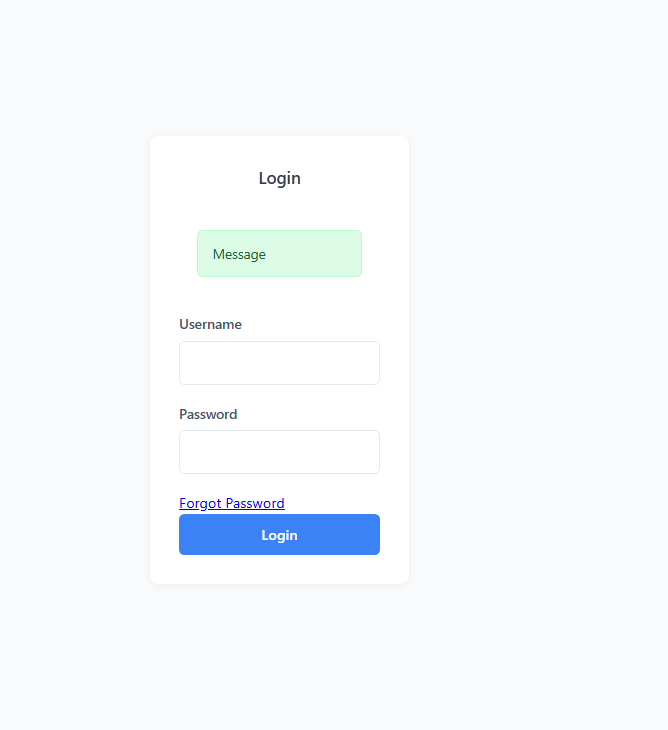
About us



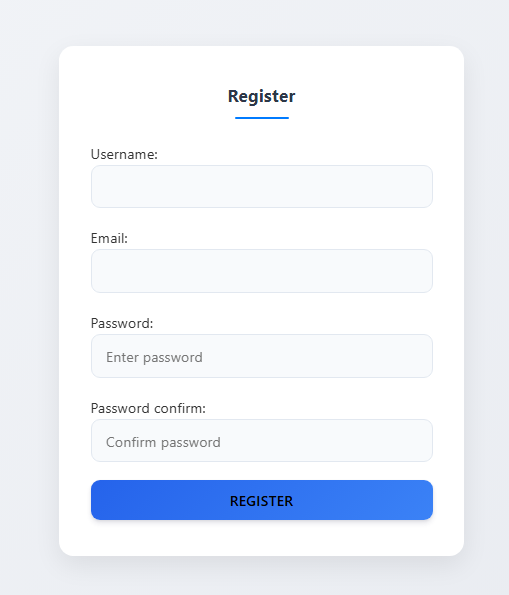




footer



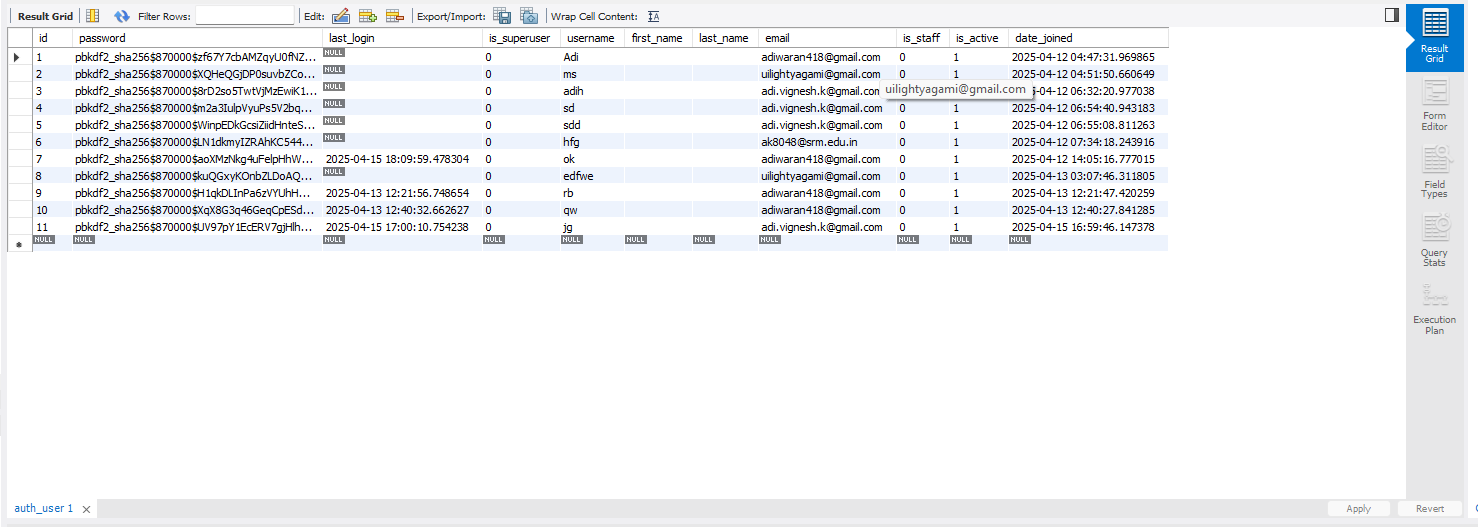
Login page



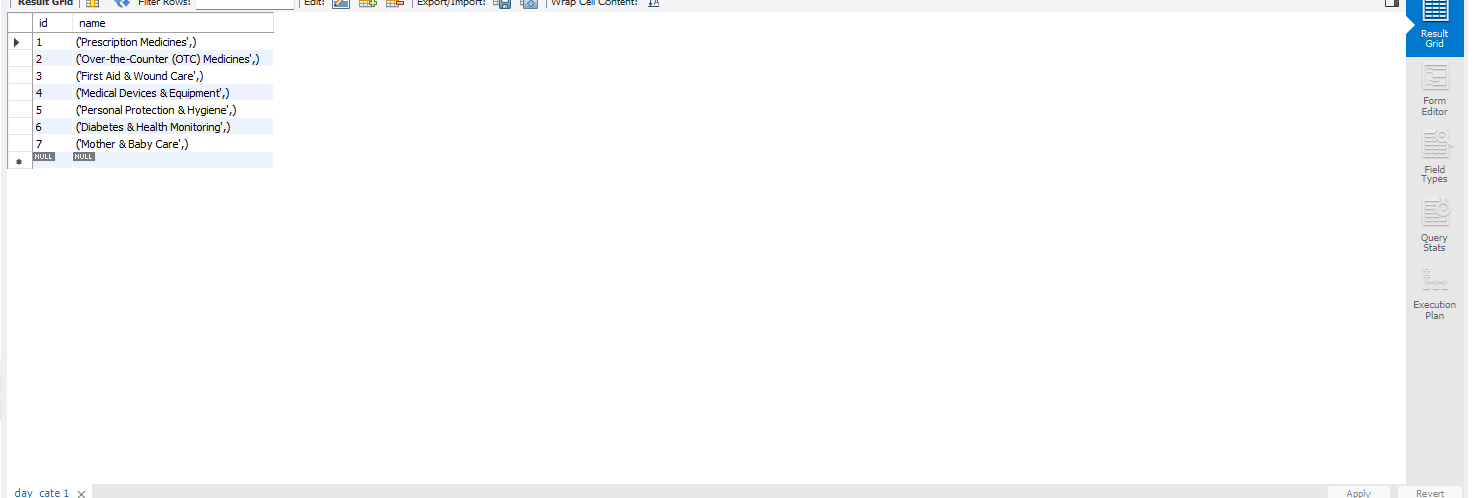
Sign up page

**Front end**

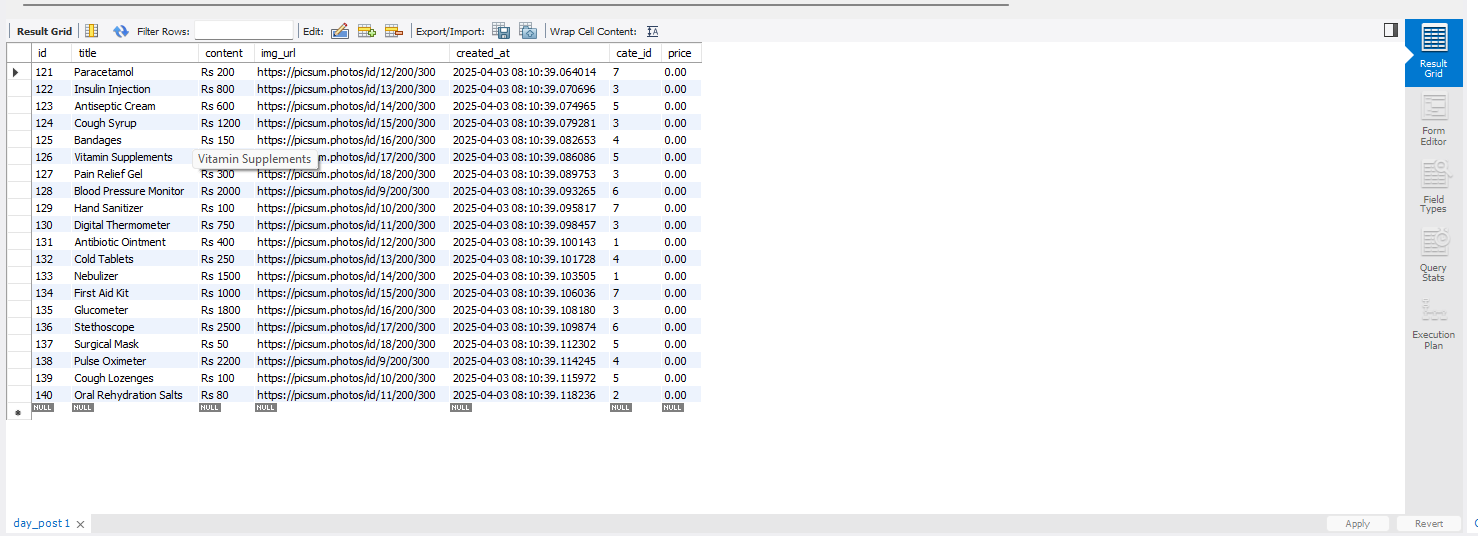
t



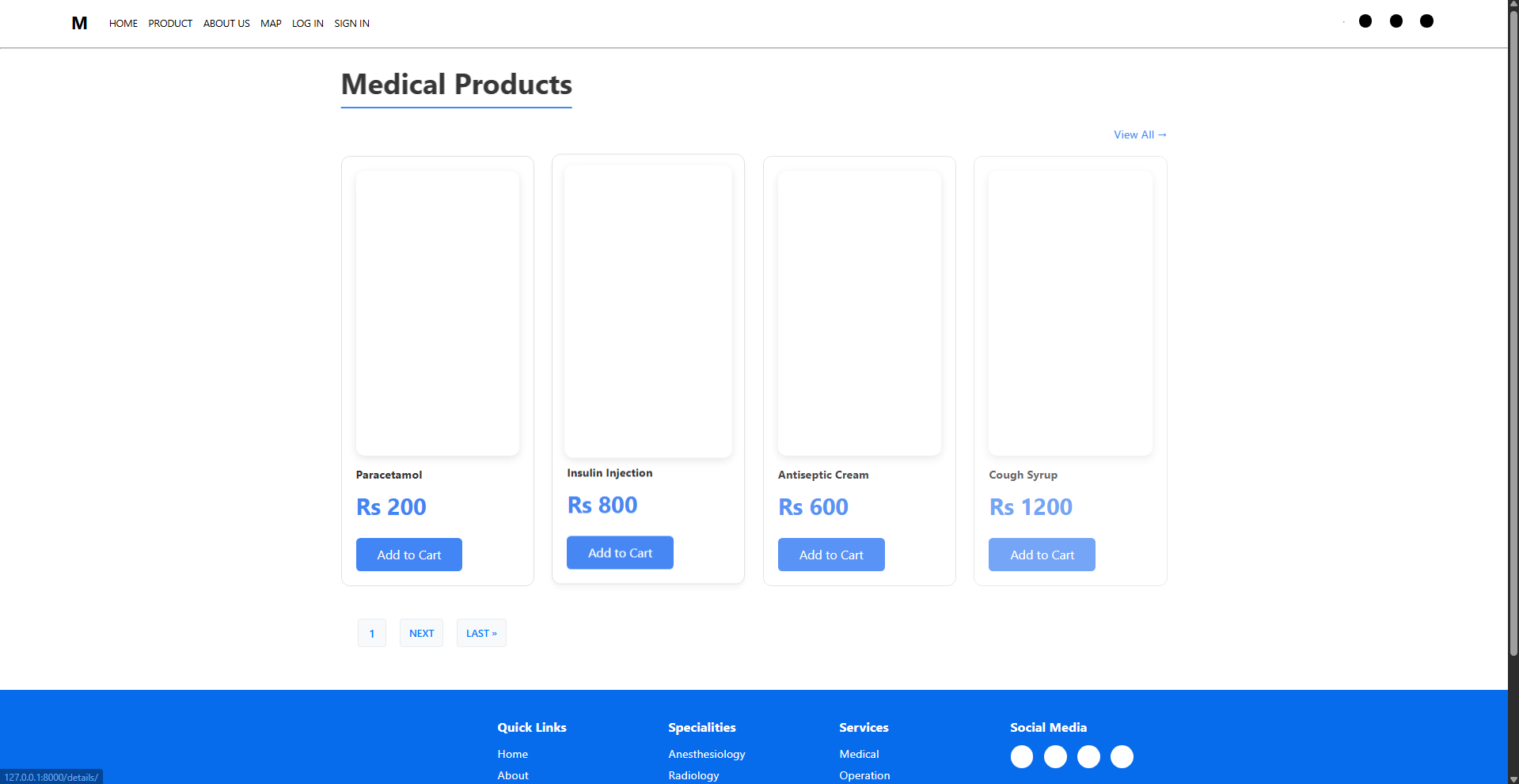
Login details



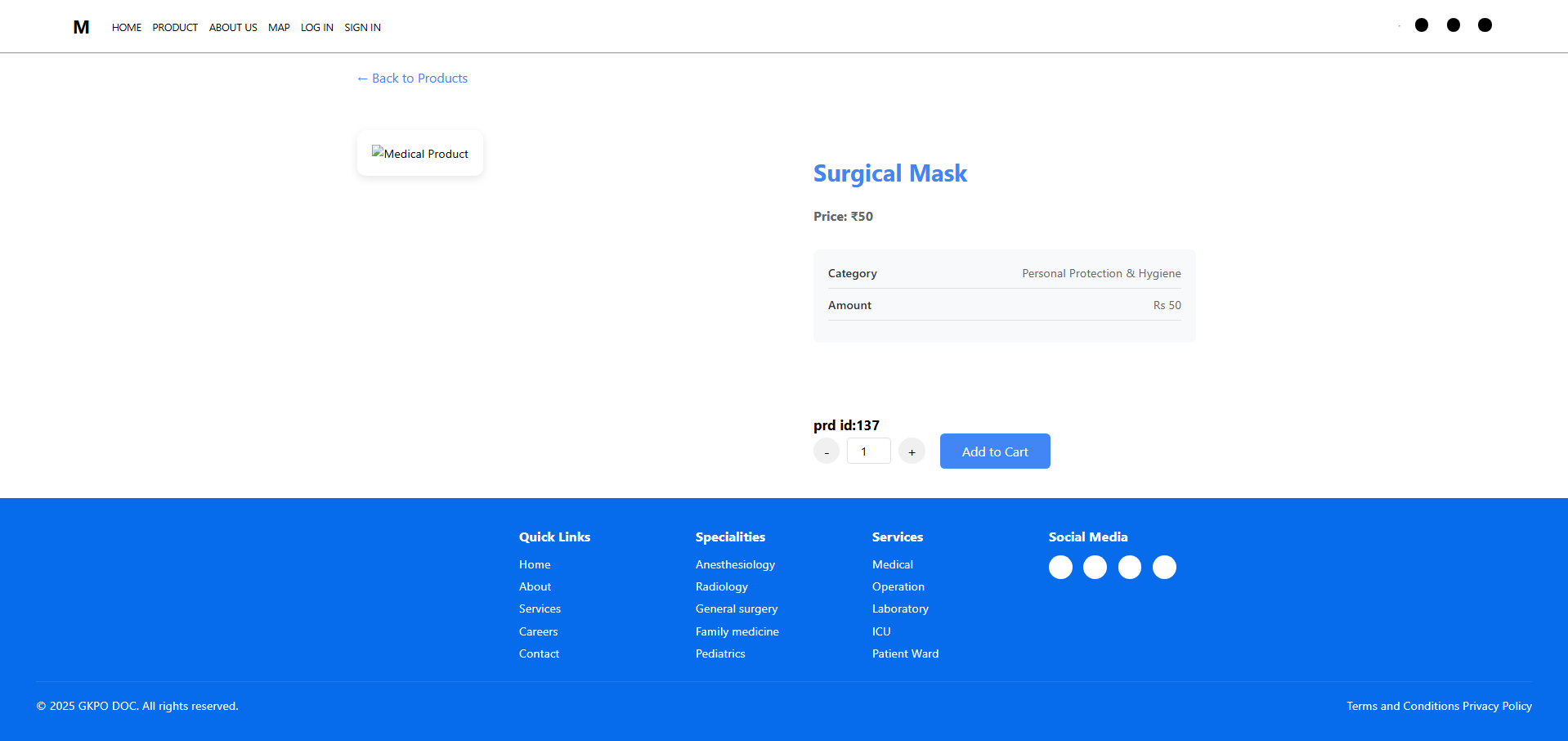
Categories



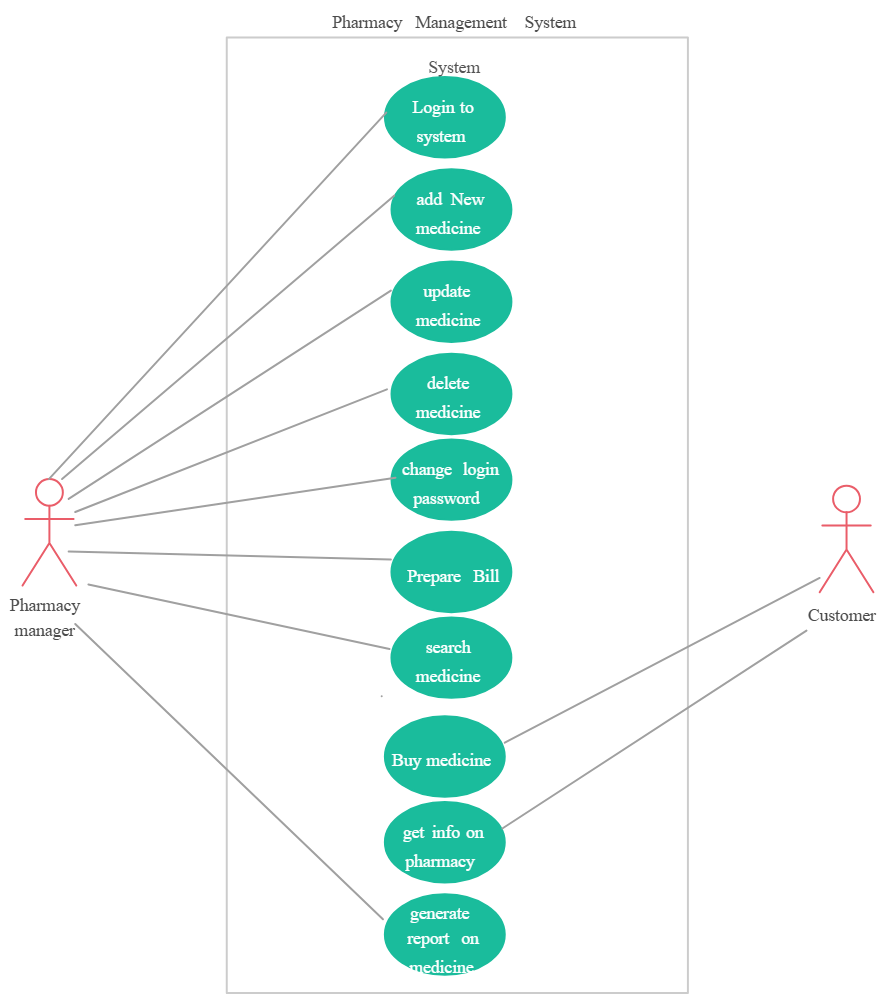
Data set



Product page



Order page

use case diagram