

# **1. INTRODUCTION**

## 1. INTRODUCTION

### *1.1 PROBLEM STATEMENT*

In the rapidly evolving e-commerce industry, purchasing office furniture and accessories remains a niche market underserved by existing platforms. Customers often face challenges such as limited product variety, insufficient details about items, unclear availability, and lack of a streamlined order management system. Furthermore, businesses struggle with inventory management and order processing due to inadequate tools and inefficient systems.

The absence of a specialized platform focused on ergonomic, functional, and aesthetic office furniture exacerbates these issues, leaving a gap in the market for a solution tailored to the needs of both customers and administrators. A comprehensive, user-friendly, and secure platform is required to address these challenges, providing customers with a seamless shopping experience and enabling administrators to efficiently manage products, inventory, and orders.

**Desk Delight** aims to solve these problems by offering a dedicated platform designed to meet the specific needs of the office furniture market, ensuring convenience, reliability, and satisfaction for all stakeholders.

### *1.2 PROPOSED SYSTEM*

The **Desk Delight** platform is designed to address the challenges faced in purchasing and managing office furniture by providing a seamless and efficient e-commerce solution. It aims to create a specialized marketplace that caters to customers seeking quality office furniture and administrators managing products and orders.

The proposed system simplifies the shopping experience for customers through an intuitive interface that allows browsing, searching, and purchasing products with ease. For administrators, it provides robust tools for managing inventory, orders, and user interactions, ensuring operational efficiency.

By integrating features like secure user authentication, detailed product pages, real-time inventory updates, and order tracking, Desk Delight bridges the gap between customers and sellers, delivering a streamlined solution tailored to the office furniture market.

### ***1.3 FEATURES OF THE PROPOSED SYSTEM***

The proposed system will offer the following key features:

#### **For Customers:**

1. **User Registration and Login:** Secure and user-friendly authentication system with OTP-based password recovery.
2. **Product Browsing:** View a variety of office furniture with detailed descriptions, images, and reviews.
3. **Shopping Cart:** Add products to a cart with the ability to modify quantities or remove items.
4. **Order Placement:** Effortless order checkout with options to provide shipping details and choose payment methods.
5. **Order Tracking:** Track the status of orders, including confirmation, dispatch, and delivery.
6. **Search and Filter Options:** Easily find products based on categories, price range, or popularity.

#### **For Administrators:**

1. **Product Management:** Add, update, or delete products, including details such as stock availability, pricing, and images.
2. **Inventory Control:** Monitor and update product quantities to ensure accurate stock levels.
3. **Order Management:** View, update, and manage customer orders with the ability to change order statuses.
4. **User Management:** View, manage, and delete customer accounts to maintain platform security.

5. **Dashboard:** A comprehensive admin dashboard providing insights into sales, user activity, and inventory trends.

**Additional Features:**

- **Responsive Design:** Optimized for desktop and mobile devices, ensuring accessibility and convenience.
- **Feedback System:** Allows customers to rate and review products, improving transparency and trust.

## **2. FUNCTIONAL REQUIREMENTS**

## **2. FUNCTIONAL REQUIREMENTS**

### ***Administrative Functions***

- The admin can log in to the system and manage all administrative functionalities of the website.
- The admin can add, update, and remove product details, including product name, description, pricing, and image.
- The admin can view user profiles, including order history and feedback (if implemented in the future).
- The admin can view and manage customer orders, including updating the order status (e.g., pending, confirmed, or canceled).

### ***User Registration***

- Users can sign up for an account by providing necessary details, such as name, email address, and password.
- The system validates the provided details to ensure secure registration, and users are redirected to a dashboard after successful registration.

### ***User Authentication***

- Registered users can securely log in to their accounts using their email and password.
- Includes a Forgot Password feature that allows users to reset their password through OTP-based email verification.

### ***Product Browsing and Viewing***

- Users can browse products based on categories (e.g., gaming chairs) and view detailed product information, including name, description, price, and images.
- The system allows users to filter products based on categories and display only available products.

### ***Shopping Cart Management***

- Users can add products to their shopping cart, specifying quantities and checking the total cost.
- Users can update their cart, change product quantities, or remove items.

### ***Checkout and Order Placement***

- Users can proceed to checkout, entering shipping details and confirming their order.
- The system ensures that inventory is updated when an order is placed.

### ***Order Tracking and Cancellation***

- Users can view their order status (e.g., pending, confirmed, or canceled) and track order history.
- Users can cancel orders if the status is still "Pending."

### ***Admin Product Management***

- The admin can add, update, and remove products listed on the platform.
- Admins can adjust product pricing, inventory quantities, and product images.

### ***Order Management by Admin***

- Admins can view, approve, or cancel orders placed by users.
- Admins can update the status of orders (e.g., from pending to confirmed or canceled).

### ***Logout***

- Users can securely log out of their account at any time to ensure account safety.

# **3. NON-FUNCTIONAL REQUIREMENTS**



### **3. NON-FUNCTIONAL REQUIREMENTS**

#### ***PERFORMANCE***

- The system can handle multiple user requests simultaneously, ensuring smooth experiences during product searches, user registrations, and bookings.
- Pages load within 2 to 3 seconds, even during peak times when many users are interacting with the platform.
- Database queries for operations like product browsing, order placements, and user history are optimized for quick response times.

#### ***SCALABILITY***

- The system is designed to handle an increasing number of users, products, and transactions as the platform grows.
- The system architecture supports future features, such as expanded product categories or additional user functionalities, without major performance degradation.

#### ***USABILITY***

- The interface provides a seamless and intuitive experience for both customers and administrators, with clear navigational paths and well-organized layouts
- The platform ensures user-friendly features, such as simple product selection, easy checkout, and efficient account management.
- Clear, actionable error messages are displayed, guiding users without exposing sensitive system information.

#### ***DEPENDABILITY AND ACCESSIBILITY***

- The system is available 24/7, ensuring uninterrupted access for users and admins, with redundancy to minimize disruptions.
- Regular automated backups are implemented to prevent data loss in case of system failures.

- The platform recovers quickly after unexpected failures, resuming operations with minimal downtime, ensuring business continuity.

### ***AVAILABILITY***

- The platform maintains 99.9% uptime, ensuring consistent service availability with minimal downtime for users and administrators.
- Failover mechanisms are in place, ensuring continued availability during server or network issues by switching to backup systems or servers

### ***MAINTAINABILITY***

- The system follows modular, well-documented coding practices to make it easier for developers to maintain, debug, and update the platform.
- Logs are generated for significant events, such as user activity, system errors, or admin actions, helping administrators monitor system performance and troubleshoot effectively.

### ***COMPATIBILITY***

- The platform is compatible with modern browsers (Chrome, Firefox, Safari, Edge) and supports responsive design, ensuring an optimal user experience across different devices, including desktops, tablets, and smartphones.

## **4. UML DIAGRAMS**

## 4.1 USE CASES

### 4.1.1 Register

**Use Case ID:** UC-01

**Primary Actor:** User

**Goal in Context:** The user successfully registers for a new account to access the platform's features.

**Preconditions:**

- The user is not yet registered.
- The user has access to the registration page.

**Postconditions:**

- The user's information is saved to the database, and the user can log in.

**Main Flow:**

1. The user navigates to the registration page.
2. The user fills in required details such as name, email, and password.
3. The system validates the input data.
4. The system stores the user's data in the database.
5. The user is redirected to the login page.

**Alternate Flow:**

- If the user provides invalid data, the system displays an error message prompting the user to correct the details.

### 4.1.2 Login

**Use Case ID:** UC-02

**Primary Actor:** User

**Goal in Context:** The user successfully logs into their account.

**Preconditions:**

- The user has a registered account.
- The user is on the login page.

**Postconditions:**

- The user is authenticated and directed to their dashboard.

**Main Flow:**

1. The user navigates to the login page.
2. The user enters their email and password.
3. The system verifies the credentials.
4. If the credentials are valid, the system redirects the user to the dashboard.
5. If the credentials are invalid, the system displays an error message.

**Alternate Flow:**

- If the user forgets their password, they can click the "Forgot Password" link to reset their password via email verification.

### ***4.1.3 Product Search***

**Use Case ID:** UC-03

**Primary Actor:** User

**Goal in Context:** The user searches for a specific product by name or category.

**Preconditions:**

- The user is logged into the system.
- The user is on the homepage or product catalog.

**Postconditions:**

- The search results are displayed based on the search query.

**Main Flow:**

1. The user enters a product name or category in the search bar.
2. The system processes the search query and filters the products based on the search term.
3. The system displays the search results with product names, prices, and images.

**Alternate Flow:**

- If no results match the search query, the system displays a "No results found" message.

### ***4.1.4 Add Product to Cart***

**Use Case ID:** UC-04

**Primary Actor:** User

**Goal in Context:** The user adds a product to their shopping cart.

**Preconditions:**

- The user is logged into the system.
- The user is on a product detail page.

**Postconditions:**

- The selected product is added to the user's cart.

**Main Flow:**

1. The user selects a product to view its details.
2. The user clicks the “Add to Cart” button.
3. The system adds the product to the user’s shopping cart.
4. The user is notified that the product has been successfully added.

#### ***4.1.5 Checkout and Payment***

**Use Case ID:** UC-05

**Primary Actor:** User

**Goal in Context:** The user completes a purchase by checking out and providing payment information.

**Preconditions:**

- The user has at least one item in the shopping cart.
- The user is logged in.

**Postconditions:**

- The order is processed, and an invoice is generated.

**Main Flow:**

1. The user clicks on the “Checkout” button in the shopping cart.
2. The user reviews their order and enters payment and shipping details.
3. The system processes the payment securely.
4. Upon successful payment, the system generates an order and sends a confirmation email with the invoice.

**Alternate Flow:**

- If the payment fails, the user is notified, and they can attempt payment again.

#### ***4.1.6 View Order History***

**Use Case ID:** UC-06

**Primary Actor:** User

**Goal in Context:** The user views their past order details.

**Preconditions:**

- The user is logged in.
- The user has made previous orders.

**Postconditions:**

- The user sees a list of all past orders.

**Main Flow:**

1. The user navigates to their account dashboard.
2. The user clicks on "Order History" to view past orders.
3. The system displays the order history with details such as product name, price, status, and invoice.

#### ***4.1.7 Admin Login***

**Use Case ID:** UC-07

**Primary Actor:** Admin

**Goal in Context:** The admin logs into the admin portal.

**Preconditions:**

- The admin has valid login credentials.
- The admin has access to the admin portal.

**Postconditions:**

- The admin is logged into the system and directed to the admin dashboard.

**Main Flow:**

1. The admin navigates to the admin login page.
2. The admin enters their username and password.
3. The system verifies the credentials.
4. If the credentials are correct, the admin is redirected to the admin dashboard.

**Alternate Flow:**

- If the credentials are invalid, the system displays an error message.

#### ***4.1.8 Add Product (Admin)***

**Use Case ID:** UC-08

**Primary Actor:** Admin

**Goal in Context:** The admin adds a new product to the platform's catalog.

**Preconditions:**

- The admin is logged into the system.
- The admin has product details ready to add.

**Postconditions:**

- The new product is available for users to view and purchase.

**Main Flow:**

1. The admin navigates to the product management page.
2. The admin fills in the product details (name, description, price, image).
3. The admin submits the form to add the product to the catalog.
4. The system confirms the successful addition of the product.



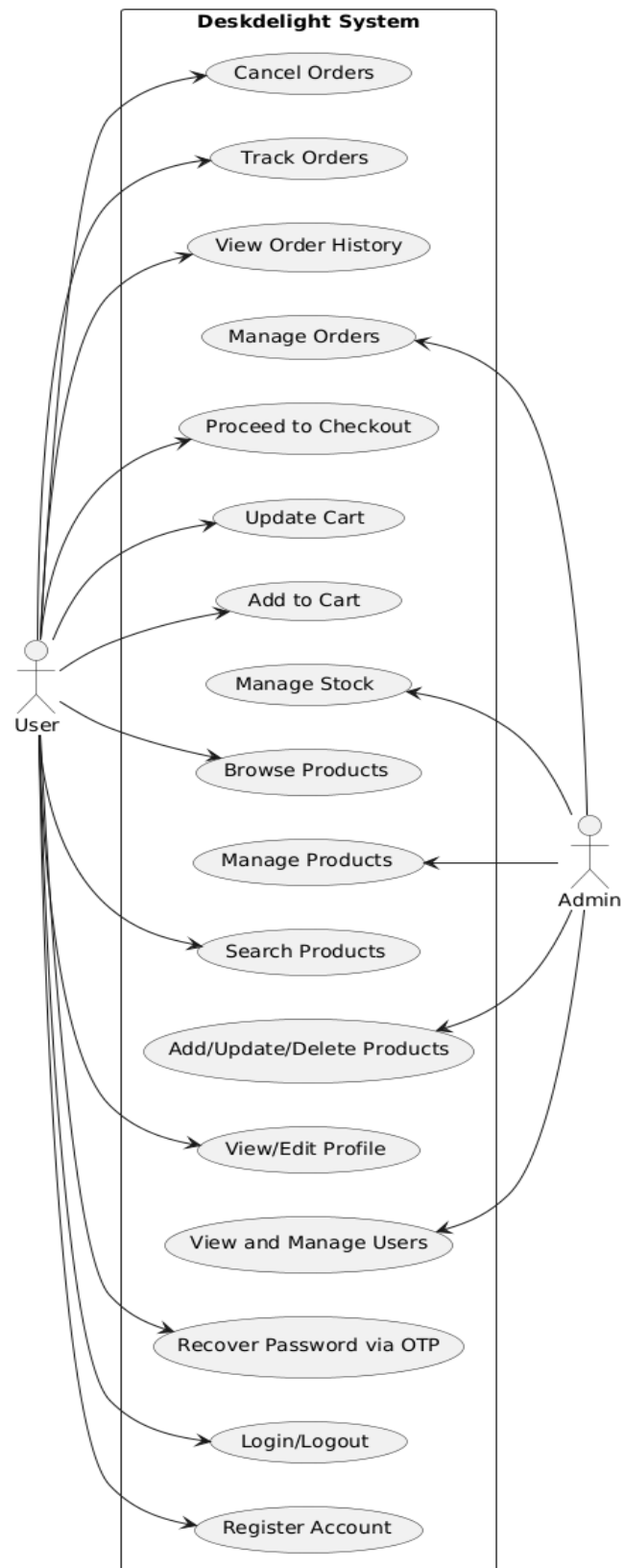
**4.2 USECASE DIAGRAM**

Figure 4.1

### 4.3 ACTIVITY DIAGRAM

#### USER

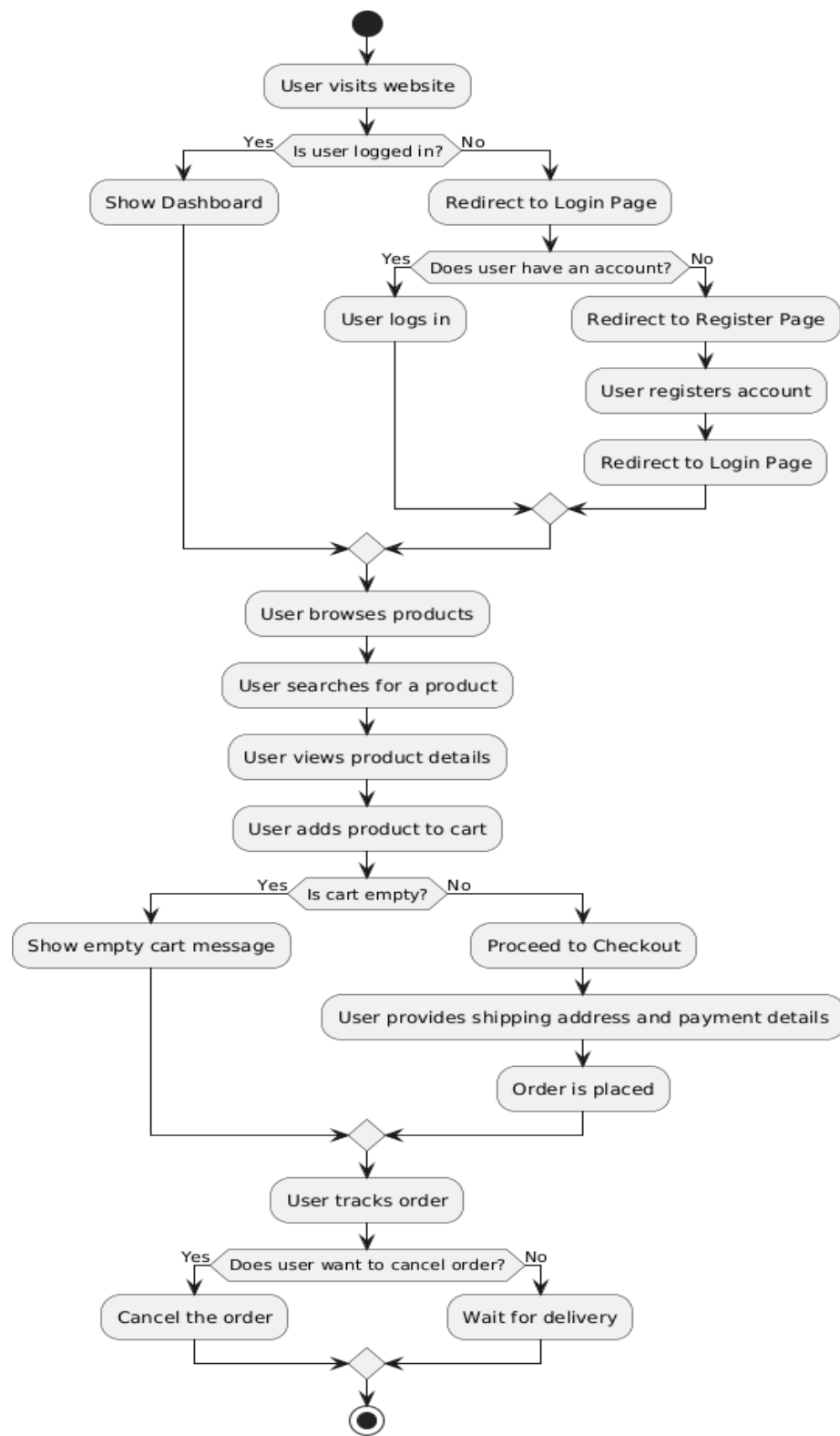


Figure 4.2

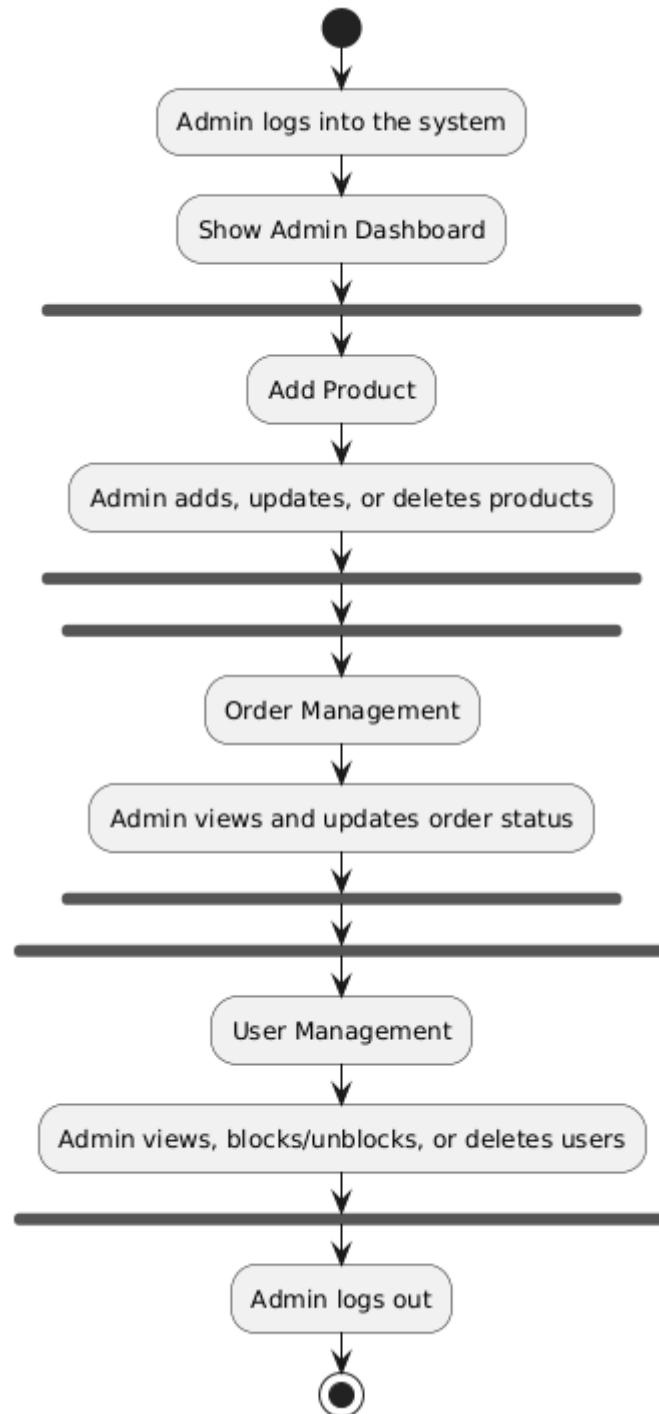
**ADMIN**

Figure 4.3

#### 4.4 CLASS DIAGRAM

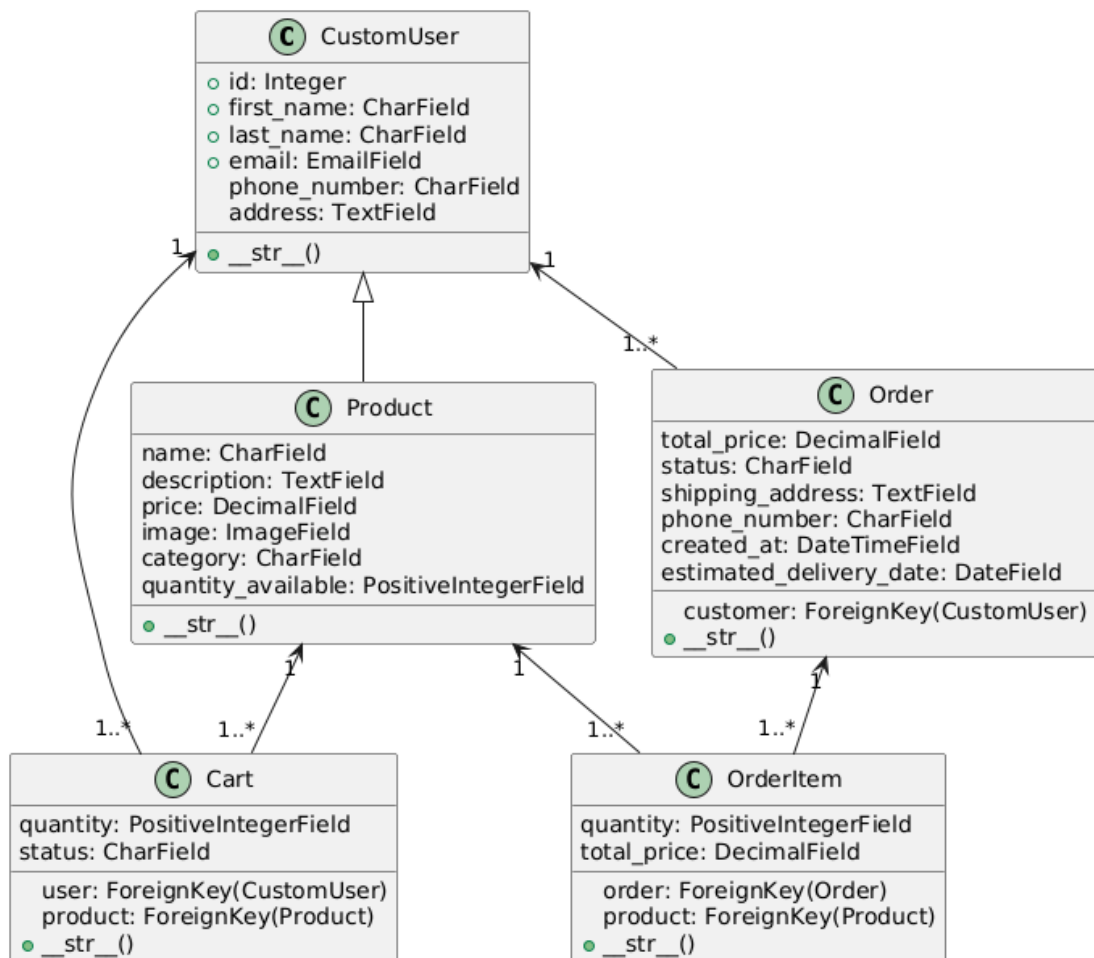


Figure 4.4

## **5. TEST CASES**

**5.1 LOGIN**

SR_NO	TEST CASE	FEATURE	DESCRIPTION	STEPS TO EXECUTE	EXPECTED RESULTS
1	TC-01	User Interface	Check the appearance of the login page	Load the login page of the system	UI elements are properly aligned. Navigation buttons and links are functional.
2	TC-02	Login form	Verify that a registered user can log in with valid credentials	Enter valid email and password	User is redirected to the dashboard or home page.
3	TC-03	Login form	Verify that a user will stay on the login page when entering invalid credentials	Enter incorrect email or password	User stays on the login page with an error message.
4	TC-04	Password Recovery	Verify the forgot password functionality works	Click the 'Forgot Password' link. Enter the registered email and click 'Send OTP'.	A password reset link is sent to the registered email address.
5	TC-05	Login form	Verify that a user cannot log in without filling in both the email and password fields	Leave email or password field empty	A prompt indicates that both fields must be filled.

**5.2 REGISTRATION**

SR_NO	TEST CASE	FEATURE	DESCRIPTION	STEPS TO EXECUTE	EXPECTED RESULTS
1	TC-01	User Interface	Check the appearance of the registration page	Load the registration page	UI elements are properly aligned, and form fields are clearly labeled and easy to understand.
2	TC-02	Registration Form	Check the behavior of the registration form with empty fields	Leave all fields blank. Click "Register".	Error messages are displayed for empty fields with clear, informative prompts.
3	TC-03	Registration Form	Prevent invalid data during registration	Enter invalid data (incorrect email, weak password, mismatched password). Click "Register".	Specific error messages for invalid data. Registration should not proceed.
4	TC-04	Registration Form	Check successful registration with valid data	Enter valid data (first name, last name, email, password). Click "Register"	Registration is successful. A confirmation message and a welcome email are sent.
5	TC-05	Login Functionality	Verify login with valid credentials after registration	Enter registered email and password. Click "Login"	Login is successful. User is redirected to the dashboard or home page.

**5.3 PRODUCT ADDITION (Admin Dashboard)**

SR_NO	TEST CASE	FEATURE	DESCRIPTION	STEPS TO EXECUTE	EXPECTED RESULTS
1	TC-01	Admin Dashboard	Verify admin can add a new product	Log in as admin. Navigate to 'Add Product' page. Enter product details (name, description, price, category, quantity). Click 'Save'.	Product is successfully added and visible in the product list.
2	TC-02	Admin Dashboard	Verify all fields are required for adding a product	Log in as admin. Navigate to 'Add Product' page. Leave one or more fields blank. Click 'Save'.	Error message is displayed for missing required fields.
3	TC-03	Admin Dashboard	Verify admin can update product details	Log in as admin. Navigate to 'Product Management' page. Select a product, edit details, and click 'Save'.	Product details are successfully updated.
4	TC-04	Admin Dashboard	Verify admin can remove a product	Log in as admin. Navigate to 'Product Management' page. Select a product, click 'Delete', and confirm.	Product is removed from the list.
5	TC-05	Admin Dashboard	Verify image upload functionality for product	Log in as admin. Navigate to 'Add Product' page. Upload an image for	Image is uploaded and displayed correctly on the product detail page.



				the product and click 'Save'.	
<b>5.3 ORDER MANAGEMENT (Admin Dashboard)</b>					
SR_NO	TEST CASE	FEATURE	DESCRIPTION	STEPS TO EXECUTE	EXPECTED RESULTS
1	TC-01	Order Management	Verify admin can view the list of orders	Log in as admin. Navigate to 'Order Management' page. View the list of orders.	Admin can view all orders with their details (status, total price, etc.).
2	TC-02	Order Management	Verify admin can update the status of an order	Log in as admin. Navigate to 'Order Management' page. Select an order, change status (e.g., from 'Pending' to 'Shipped'), and click 'Save'.	Order status is updated successfully.
3	TC-03	Order Management	Verify admin can view detailed information of an order	Log in as admin. Navigate to 'Order Management' page. Select an order to view its details (items, shipping address, etc.).	Admin can view detailed information of the selected order.
4	TC-04	Order Management	Verify admin can cancel an order	Log in as admin. Navigate to 'Order Management' page. Select an	Order is successfully canceled and its status is updated to

				order and click 'Cancel'.	'Cancelled'.
5	TC-05	Order Management	Verify admin can remove an order from the system	Log in as admin. Navigate to 'Order Management' page. Select an order, click 'Delete', and confirm.	Order is removed from the system and no longer appears in the order list.

#### **5.4 CART MANAGEMENT (User Feature)**

SR_NO	TEST CASE	FEATURE	DESCRIPTION	STEPS TO EXECUTE	EXPECTED RESULTS
1	TC-01	Cart Management	Verify that a user can add a product to the cart	Log in as user. Navigate to product page. Select a product and click 'Add to Cart'.	Product is added to the cart, and the cart count is updated.
2	TC-02	Cart Management	Verify that the cart page shows all added products	Log in as user. Navigate to 'Cart' page.	All products added to the cart are displayed with their details (name, quantity, price).
3	TC-03	Cart Management	Verify that a user can update the quantity of items in the cart	Log in as user. Navigate to 'Cart' page. Select a product and change its quantity. Click 'Update Cart'.	Cart is updated with the new quantity, and the total price is recalculated.
4	TC-04	Cart Management	Verify that a user can remove a product from the cart	Log in as user. Navigate to 'Cart' page. Click 'Remove' next to a product.	Product is successfully removed from the cart.

## DESK DELIGHT

5	TC-05	Cart Management	Verify that a user can checkout with items in the cart	Log in as user. Navigate to 'Cart' page. Click 'Checkout'. Enter shipping details and payment information.	Checkout process is successful, and an order is created.
---	-------	-----------------	--	--	--

## **6. INPUT DESIGN AND OUTPUT DESIGN**

## 6.1 INPUT DESIGN

The input design for **Desk delight** ensures that both customers and admins can easily interact with the platform. The system focuses on user-friendly data entry with proper validation and error handling to minimize mistakes.

### Customer Input Design:

#### 1.Login and Registration Forms:

##### Login:

- **Input Fields:**
  - Email (required, valid email format)
  - Password (required, minimum 8 characters, must contain at least one letter and one number)
- **Validation:**
  - Ensure the email format is valid (e.g., example@domain.com).
  - Password must meet security standards (8+ characters, alphanumeric).
  - If login fails, display an error message (e.g., "Invalid email or password").
- **Error Handling:**
  - If either the email or password is empty, prompt with "This field cannot be empty".
  - Invalid email format should display "Please enter a valid email address".

##### Registration:

- **Input Fields:**
  - Full Name (required, alphanumeric, max 100 characters)
  - Email (required, valid email format)
  - Password (required, minimum 8 characters, alphanumeric)
  - Confirm Password (required, must match the password)
  - Phone Number (optional, numeric, valid format)
- **Validation:**

- Ensure the email format is correct and does not already exist in the database.
- Ensure the password is secure (at least 8 characters, includes both letters and numbers).
- Ensure the password and confirm password fields match.
- **Error Handling:**
  - If any required field is empty, display an error message ("This field is required").
  - If passwords do not match, show "Passwords do not match".
  - If email format is invalid, show "Please enter a valid email address".

## 2. Product Booking

### Input Fields:

- Product Name (pre-filled)
- Quantity (required, numeric)
- Delivery Date (required, date picker)
- Delivery Address (required, text area)

### Validation:

- Ensure quantity is a positive integer.
- Ensure delivery date is not in the past.
- Ensure delivery address is not empty.

### Error Handling:

- If any field is empty, display "This field is required".
- If quantity is not a positive number, show "Please enter a valid quantity".
- If the delivery date is in the past, show "Please choose a future delivery date".

## 3. Payment Details:

### Input Fields:

- Credit Card Number (required, valid card format)

- Expiry Date (required, valid date format)
- CVV (required, numeric, 3 digits)

**Validation:**

- Ensure credit card number is in valid format (16 digits).
- Ensure expiry date is a valid future date.
- Ensure CVV is a 3-digit number.

**Admin Input Design:**

**1. Product Management**

**Input Fields:**

- Product Name (required, alphanumeric, max 100 characters)
- Description (required, max 500 characters)
- Price (required, numeric, 2 decimal points)
- Quantity (required, numeric)
- Product Image (optional, file upload)

**Validation:**

- Ensure price and quantity are valid numeric values.
- Ensure product name and description are not empty.

**Error Handling:**

- If required fields are empty, show error messages like "This field is required".
- If price or quantity is invalid, display "Please enter a valid price/quantity".

## **2. Order Management:**

### **Input Fields:**

- Order ID (auto-generated)
- Order Status (required, select from "Pending", "Shipped", "Delivered", "Cancelled")
- Delivery Date (optional, date picker)

### **Validation:**

- Ensure order status is selected.
- If a delivery date is provided, ensure it is valid.

### **Error Handling:**

- If order status is not selected, display "Please select a status".
- If delivery date is not in the future, show "Please choose a valid delivery date".

### **Data Validation:**

#### **Real-Time Validation:**

- Immediate feedback is provided when incorrect data is entered, for example, when an invalid email format is detected during registration or login.
- On the booking page, if the selected time slot is unavailable, the system will automatically suggest available slots.

#### **Required Fields:**

- All fields that are necessary for a successful submission (such as registration, booking, feedback) are validated to ensure they are not left empty.

#### **Error Alerts:**

- Error messages are shown when users fail to enter valid information, with clear descriptions of the issue (e.g., "Invalid email format", "Password too short").



## 6.2 OUTPUT DESIGN

The output design for Desk delight is structured to provide clear, relevant, and timely information to both customers and admins, ensuring seamless interaction with the system.

### Customer Output Design:

#### 1. User Dashboard:

- The dashboard shows relevant customer information such as their current cart, past orders, and recent interactions with the store.
- It provides a summary of order statuses (pending, shipped, delivered, or canceled) with quick links to manage orders or view product details.

#### 2. Order Confirmation:

- After completing an order, customers will see a confirmation message displaying the order ID, product names, order date, and total price.
- A downloadable invoice is automatically generated, summarizing the order details, including pricing, quantity, shipping details, and payment status.

#### 3. Product Pages:

- Customers can browse products with detailed descriptions, images, and pricing information. Available products will be listed with attributes such as name, category, and price.
- Product pages will also display stock availability to help customers make informed purchase decisions.

#### 4. Order History:

- Customers can view their order history, including detailed information about each past order such as order status, products purchased, and shipping address.

### Admin Output Design:

#### 1. Admin Dashboard:

- The admin dashboard displays an overview of system activities, such as:
  - Recent orders (pending, shipped, delivered).
  - Order statistics (total sales, total pending orders).
  - Quick links for managing orders, products, and customers.

## 2. Order Details:

- Admins can view detailed order records, including customer information, product details, order status, and payment status.
- Admin can update the order status (e.g., pending, shipped, or delivered) and add notes if needed.

## 3. Product Management:

- Admins can view, add, update, or delete products from the product catalog.
- Product information will include product name, description, category, price, stock quantity, and images.

## 4. Customer Management:

- Admins can view customer profiles, including their order history and contact details. Admin can manage customer accounts, including the ability to deactivate or modify customer information.

## Key Output Features:

- **Real-time updates:** Confirmation messages will be displayed after customer orders, and admin updates.
- **Clear and actionable messages:** Guidance will be provided after each action (order confirmation, product update).
- **Downloadable Invoices:** Customers will receive invoices for their purchases.
- **Administrative Insights:** Admins will receive summaries and detailed views of sales, orders, and customer data to manage the platform efficiently.

## **7. SYSTEM IMPLEMENTATION**

## 7.1 Introduction

The Deskdelight project is an e-commerce platform specializing in selling premium chairs, designed to offer a user-friendly and efficient online shopping experience. Built using Django as the backend framework, the frontend leverages HTML, CSS, and Bootstrap for a responsive and visually appealing design. JavaScript is utilized to add dynamic and interactive elements to enhance usability. The project environment is developed with Python 3.9, Django 4.x, and other dependencies, including SQLite for database management, Pillow for image handling, and Django Crispy Forms for improved form styling.

Key features include user authentication, product browsing, a shopping cart system, product reviews and ratings, OTP-based password recovery, and an admin interface for product management.

## 7.2 Project Structure

The project follows Django's standard project structure, with a clear separation of concerns for the frontend, backend, and static resources. The structure is organized as follows:

```

deskdelight/
|
|— deskdelight/
|   |— __init__.py
|   |— settings.py
|   |— urls.py
|   |— wsgi.py
|   |— asgi.py
|
|— myapp/
|   |— __init__.py
|   |— admin.py
|   |— models.py
|   |— views.py

```

```
| └─ urls.py  
| └─ templates/  
| └─ static/  
└─ manage.py
```

### 7.3 Database Design and Models

The database is implemented using Django's ORM with SQLite as the backend. The key models include:

- **User:** Custom user model for authentication, managing user accounts, and storing relevant information such as name, email, and address.
- **Product:** Fields to store product details, including name, description, price, category, stock status, and image.
- **Cart:** Represents a user's shopping cart, with fields to store the user, product, quantity, and total price.
- **Order:** Fields to handle order placement, including user, order items, total amount, order date, and status (pending, shipped, delivered).
- **Review:** Stores product reviews and ratings, including user, product, review content, rating, and timestamp.
- **OTP:** Temporary model to store OTPs for user authentication and password recovery, including user, OTP value, and expiration timestamp.
- **AdminProductManagement:** Model to manage products added or updated by the admin, including product details and modification timestamps.

<b>7.3.1 Table Name: Customer</b>			
Field Name	Data Type	Constraints	Description
Customer_id	int	PRIMARY KEY	Unique identifier for the customer
first_name	VARCHAR(50)	NOT NULL	Customer's first name
last_name	VARCHAR(50)	NOT NULL	Customer's last name
email	VARCHAR(255)	UNIQUE, NOT NULL	Customer's email address.
phone_number	VARCHAR(15)	UNIQUE, NOTNULL	Customer's phone number
password	VARCHAR(255)	NOT NULL	Customer's hashed password.
address	TEXT	NULLABLE	User's address
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time the user was created.

Table 7.1 Customer Registration

**7.3.2 Table Name: Product Table**

Field Name	Data Type	Constraints	Description
product_id	int	PRIMARY KEY	Unique identifier for the product.
name	VARCHAR(255)	NOT NULL	Name of the product.
description	TEXT	NULLABLE	Detailed description of the product.
price	DECIMAL(10, 2)	NOT NULL	Price of the product.
category	VARCHAR(100)	NOT NULL	Category of the

			product (e.g., chairs, desks).
image_url	DECIMAL(10,2)	NULLABLE	URL of the product image.
stock_quantity	INTEGER	DEFAULT 0	Quantity available in stock.
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time the product was added.

Table 7.2 Product Page

**7.3.3 Table Name: Order Table**

Field Name	Data Type	Constraints	Description
order_id	SERIAL	PRIMARY KEY	Unique identifier for the order.
customer_id	INTEGER	FOREIGN KEY REFERENCES Customer(Customer_id)	Reference to the Customer table
total_price	DECIMAL(10, 2)	NOT NULL	Total price of the order.
Status	VARCHAR(20)	CHECK (status IN ('pending', 'shipped', 'delivered'))	Status of the order.
payment_status	VARCHAR(20)	CHECK (payment_status IN ('pending', 'paid'))	Payment status of the order.
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the order was placed.

Table 7.3 Order Management

**7.3.4 Table Name: Cart table**

Field Name	Data Type	Constraints	Description
cart_id	int	PRIMARY KEY	Unique identifier for the cart
customer_id	INTEGER	FOREIGN KEY REFERENCES customer (customer_id)	Reference to the Customer table
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Date and time when the cart was created.

Table 7.4 Cart

**7.3.5 Table Name: Cart Item Table**

Field Name	Data Type	Constraints	Description
cart_item_id	INTEGER	PRIMARY KEY	Unique identifier for the cart item.
cart_id	INTEGER	FOREIGN KEY REFERENCES Cart(cart_id)	Reference to the Cart the item belongs to.
product_id	INTEGER	FOREIGN KEY REFERENCES Product(product_id)	Reference to the Product in the cart.
Quantity	INTEGER	NOT NULL	Quantity of the product in the cart.
total_price	DECIMAL(10, 2)	NOT NULL	Total price for the quantity of the product.

Table 7.5 Cart item



**7.3.6 Table Name: OTP Table**

Field Name	Data Type	Constraints	Description
otp_id	INTEGER	PRIMARY KEY	Unique identifier for the OTP record.
customer_id	INTEGER	FOREIGN KEY REFERENCES Customer(customer_id)	Reference to the User requesting the OTP.
otp_code	VARCHAR(6)	NOT NULL	OTP code generated for password reset.
generated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Timestamp when the OTP was generated.
expired_at	TIMESTAMP	NOT NULL	Timestamp when the OTP expires.

Table 7.6 OTP table

**7.4 URL Routing and Views****URL Routing:**

In the Desk delight project, URL routing is modularized into app-specific urls.py files. These files are then included in the main urls.py for better maintainability and scalability. The URLs are structured to handle product browsing, user actions (authentication, registration), admin management, and order handling.

### Example URLs:

- /products/ - Displays a list of available products.
- /product/<int:product\_id>/ - Shows detailed information about a specific product.
- /cart/ - Displays the items added to the cart.
- /checkout/ - Handles the checkout process.
- /register/ - Handles user registration.
- /login/ - User authentication (login).
- /profile/ - Displays user profile information.
- /admin/products/ - Admin interface for managing products.
- /admin/orders/ - Admin interface to manage and update orders.
- /order/track/ - Track the status of an order.

### Views:

The views of the Desk delight project are categorized based on functionality. Each view is responsible for handling specific user actions, such as viewing products, managing the cart, and handling user authentication.

### User Views:

- **User Authentication Views:**
  - **Login:** Displays a login form for users to access their accounts.
  - **Register:** Handles the registration of a new user by collecting necessary details.
  - **Profile:** Displays the logged-in user's profile page where they can view and update personal information.
  - **Logout:** Logs the user out of the system.
- **Password Reset Views:**
  - **Forgot Password:** Page for users to request a password reset by providing their email.
  - **Send OTP:** Sends an OTP to the user's email address for verification during password reset.
  - **Verify OTP:** Verifies the OTP entered by the user.

- **Reset Password:** Allows the user to set a new password after OTP verification.

#### **Product Views:**

- **Product Display:**
  - **Product List:** Displays a list of all available products.
  - **Product Details:** Shows detailed information of a selected product, such as specifications, price, and images.
- **Cart and Checkout Views:**
  - **Add to Cart:** Adds a selected product to the user's shopping cart.
  - **Cart:** Displays all the items the user has added to their cart, along with the total price.
  - **Checkout:** Handles the process of reviewing the cart, entering shipping details, and confirming the order.
  - **Place Order:** Finalizes the checkout and places the order for the user.

#### **Admin Views:**

- **Product Management:**
  - **Manage Products:** Admin can add, edit, or delete products.
  - **Product Listing:** Displays a list of all products, with options for the admin to manage them.
- **Order Management:**
  - **Manage Orders:** Admin can view all orders and update their statuses (e.g., processing, shipped, completed).
  - **Order Status Update:** Admin can update the status of an individual order.

#### **HTTP Methods:**

- **GET:** Used to retrieve information such as viewing the list of products, the cart contents, user profile, and order details.
- **POST:** Used for submitting data such as registering a new user, logging in, adding items to the cart, placing an order, or submitting feedback.

## 7.5 Templates and Frontend Integration

**Templates:** The system uses Django's template engine, leveraging its capabilities for dynamic content like loops, conditionals, and form rendering.

**Template Inheritance:**

- A base layout (base.html) includes common elements like headers, footers, and a responsive navbar for consistent design across pages.

**Static Files:** Static resources, such as images (products), CSS files, and JavaScript files, are managed and loaded via the `static/` directory.

## 7.6 User Authentication and Authorization

**Custom User Authentication:** The project includes custom authentication for users. Users will register by providing their details (name, email, and password). The passwords are securely hashed before being saved in the database using Django's `check_password` and `make_password` methods.

**Custom Login/Logout:** Users log in using their email and password. The authentication process is handled via custom views, which verify the credentials, initiate a session for the logged-in user, and store the session data for future requests.

**Password Reset:** Users can request a password reset through OTP (One-Time Password). The system generates a 6-digit OTP, stores it temporarily, and sends it to the user's registered email. The user can then reset their password using the OTP within a limited time frame (usually 15 minutes). This functionality ensures that users can regain access to their account securely.

**Role-based Access:**

- **Admin:** Admin users are able to manage product (add, edit, delete), manage order, and manage user details.
- **Customer:** Regular users (customers) can browse products, make order, and view their order history. They are not able to perform administrative actions like managing users or viewing all feedback.

**Session Management:** The system handles user sessions manually. Once the user logs in successfully, a session is initiated, and the session data (including user identification) is stored to maintain user context for subsequent requests.

## 7.7 Forms and Validation

**Django Forms:** Django forms are used throughout the application to capture user input, such as registration details, login credentials, booking information, and feedback submissions. These forms ensure that the data entered by users is properly validated and processed before being stored in the database.

### **Validation:**

- Ensures that required fields (like email, and password) are filled.
- Email validation to ensure the email is in the correct format.
- Password validation to ensure secure passwords (e.g., minimum length of 8 characters, no similarity to email).
- Custom error messages are provided to guide users in correcting invalid entries.

## 7.8 Business Logic and Core Functionality

### **Product Display:**

The products, including various gaming chairs, are dynamically fetched from the database and displayed on the homepage. Each product is shown with its image, name, and price. Users can filter products based on categories such as type (e.g., ergonomic, executive), price range, or specific features like color or material.

**Search and Filter:**

Users have the option to search for products by name, category, or specific features. A price range filter is also provided, allowing users to select products that fit within their budget. This makes it easier for users to find products that meet their specific needs.

**Cart and Checkout Process:**

The cart system collects user-selected products, quantities, and any discounts or promotions applied. When users proceed to checkout, they are prompted to enter their shipping details, review their order, and confirm payment. The backend processes the order, updates the product inventory, and stores the order details for future reference.

**Price Calculations:**

The backend calculates the total price based on the products in the user's cart, considering individual product prices, quantities, and any applicable taxes or discounts. This ensures accurate billing during checkout. If a product is removed or modified in the cart, the price calculation is dynamically updated.

**User Account Management:**

Users can create and manage their accounts, including personal information (name, email, address) and passwords. They can also view their order history, update their details, and manage preferences for notifications or promotional offers. Authentication ensures that only registered users can make purchases or track orders.

**Order and Payment Processing:**

Once an order is placed, payment processing is handled securely, ensuring that all sensitive information (credit card details, etc.) is encrypted. The payment status is updated in the database, and an order confirmation is sent to the user via email. The order status is also made available to admins for further management.

**Admin Panel:**

The admin panel allows administrators to manage products, view orders, and update product details (e.g., prices, availability). Admins can also view customer details, monitor order statuses, and manage returns or cancellations. This functionality is protected by role-based access control to ensure that only admins can access sensitive management features.

## 7.9 Testing and Debugging

**Testing:**

- **Unit Tests for Models:** Unit tests are written to ensure that models are functioning as expected. These tests validate that product data (such as name, price, and availability), user information (like email, password), and cart contents are being correctly saved and retrieved from the database. The models for products, user authentication, and orders are tested for edge cases, like invalid email formats or missing required fields.
- **Functional Tests for Views:** Functional tests are implemented to verify that the views render the correct templates based on user actions. For example, tests ensure that the registration form correctly processes user data, that users are redirected to the correct pages after logging in, and that the cart page displays the correct products. Additionally, functional tests check if forms are properly submitting data, whether product details are accurately shown, and if users can view their order confirmations.

**Debugging:**

- **Django's Built-in Debugging Mode:** During development, the application utilizes Django's built-in debugging mode (run server `--debug`). This allows developers to track errors by displaying detailed stack traces in the console, making it easier to identify where and why an error occurs in the code. The server provides insights into issues with database queries, form submissions, and template rendering.

- **Browser Developer Tools:** For frontend issues, browser developer tools (such as Chrome Dev Tools) are used to inspect the HTML, CSS, and JavaScript. These tools help identify issues like unresponsive elements, JavaScript errors, layout problems, or broken styles. The responsive design of the website is also tested using these tools to ensure that the platform works seamlessly across different screen sizes.

**Error Handling:**

- **Proper Error Messages:** The system ensures that users receive clear and informative error messages. If invalid data is entered during registration (e.g., invalid email format or missing required fields), the user is prompted with specific error messages. If a user tries to book a turf that is unavailable or already booked, the system provides an appropriate error message, such as "This turf is unavailable at the selected time."
- **404 and 500 Error Pages:** Custom error pages are displayed when a user tries to access a non-existent page (404 error) or when an internal server issue occurs (500 error). These pages are user-friendly and direct users to the homepage or a contact page for further assistance.
- **Form Validation Errors:** When submitting forms (e.g., for login, registration, or order placement), if any validation fails, users are shown specific error messages near the form fields. This ensures that users understand what needs to be corrected before resubmitting their data.



## **8. FUTURE ENHANCEMENT**

## 8. Future Enhancements

- **Automated Order and Sales Reports:** Desk delight could incorporate a feature to automatically generate sales and order reports. This feature would provide administrators with detailed insights into revenue, popular products, customer behavior, and trends. Automating report generation would reduce manual tracking, improve efficiency, and help the management team make data-driven decisions.
- **Online Payment Gateway Integration:** A future enhancement could involve integrating online payment systems such as Stripe or PayPal. This would enable users to securely pay for their orders, including products and bookings, directly from the website. It would streamline the checkout process, offer more convenience to customers, and reduce administrative overhead related to payment management.
- **Mobile Application Development:** Developing a mobile app for Desk delight could significantly enhance user engagement by allowing customers to browse products, manage their orders, and track their purchases on the go. The app could also provide push notifications for promotions, updates, or special offers, ensuring higher customer retention and satisfaction.
- **Personalized Product Recommendations:** By leveraging machine learning or AI algorithms, Desk delight could offer personalized product recommendations based on users' browsing and purchase history. This would improve the shopping experience, guiding users to products they may be more interested in, thereby increasing sales and customer satisfaction.
- **Customer Support Chatbot Integration:** A chatbot could be integrated into Desk delight's platform to provide 24/7 customer support. The chatbot would be able to assist users with common inquiries such as order status, product details, troubleshooting, or payment issues. This feature would enhance customer service, reduce wait times, and improve user satisfaction.
- **Enhanced Product Search and Filtering Options:** The current search and filter functionality could be expanded to include more specific attributes such as product ratings, availability, price range, and product categories (e.g., gaming chairs, desks). This would help customers find exactly what they are looking for more quickly, leading to a better user experience.

- **Loyalty Program for Frequent Shoppers:** A loyalty program could be introduced to reward repeat customers. By offering discounts, points, or exclusive deals based on customer purchase frequency, Desk delight would encourage customer retention. Users could redeem points for future purchases or gain access to special promotions, increasing brand loyalty.
- **Social Media Integration:** Allowing users to share their products, feedback, and experiences on social media platforms could help promote Desk delight and attract new customers. Social media sharing buttons integrated on product pages or after purchase could make it easier for users to spread the word about their shopping experience.
- **Improved Website Usability and UX:** To further enhance the user experience, Desk delight could simplify the website's navigation, streamline the checkout process, and refine its visual design. Adding features such as live product availability updates, customer reviews for products, and dynamic search results would improve the user journey and increase conversions.
- **Enhanced Security Features:** To protect user accounts and sensitive information, Desk delight could implement advanced security features, such as multi-factor authentication (MFA). Encryption of payment and user data would provide an additional layer of security and build user trust, ensuring that all transactions and personal data are safely handled.
- **Scalability and Performance Optimization:** As Desk delight grows, ensuring that the platform can handle increased traffic and a larger database will be vital. Future enhancements could include optimizing the backend, refining database queries, and scaling the infrastructure to ensure fast load times and smooth operation even as user numbers rise.

## **9. CONCLUSION**

## 9. Conclusion

The Desk delight platform is designed to provide an intuitive and efficient experience for customers seeking to purchase high-quality chairs. With a focus on user-friendly design and seamless functionality, the platform allows users to easily browse, select, and purchase chairs that suit their needs. The system is built to support various customer actions, including viewing detailed product information, adding products to the cart, and completing purchases with secure checkout.

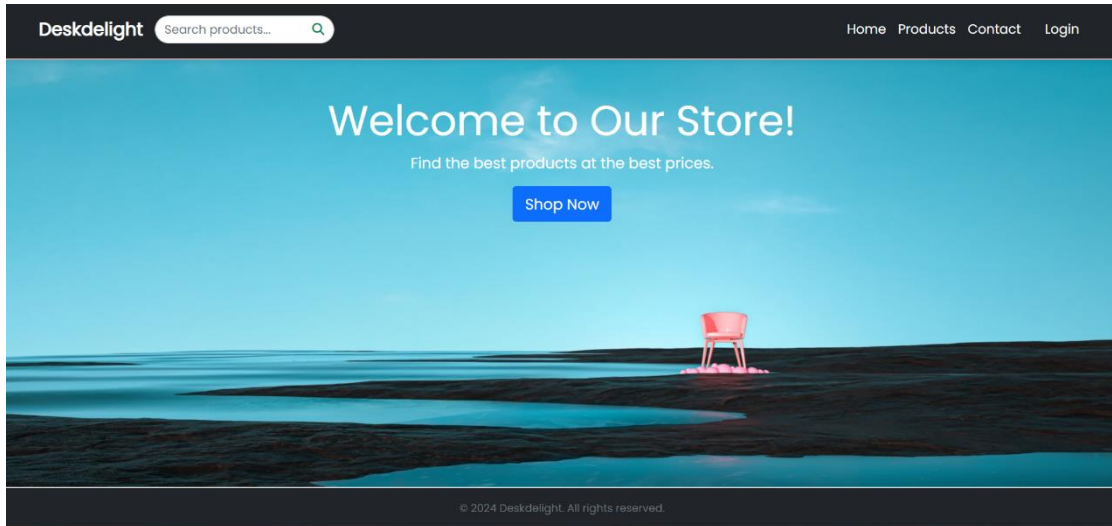
Looking ahead, the platform will continue to evolve with enhancements such as integrated payment solutions for smoother transactions, more personalized shopping experiences, and improved customer support through chatbots or other AI-powered tools. These future updates will aim to provide a more tailored and convenient shopping experience for users. Security and privacy are at the forefront of the platform's design, with robust measures like OTP-based password recovery and secure data handling ensuring that customers' personal and payment information remains safe. Desk delight aims to continually improve its performance and scalability, offering faster load times and a more responsive interface to meet growing customer demand.

Overall, Desk delight serves as a one-stop shop for high-quality chairs, ensuring a smooth and enjoyable shopping experience for users while providing efficient management and monitoring capabilities for administrators.

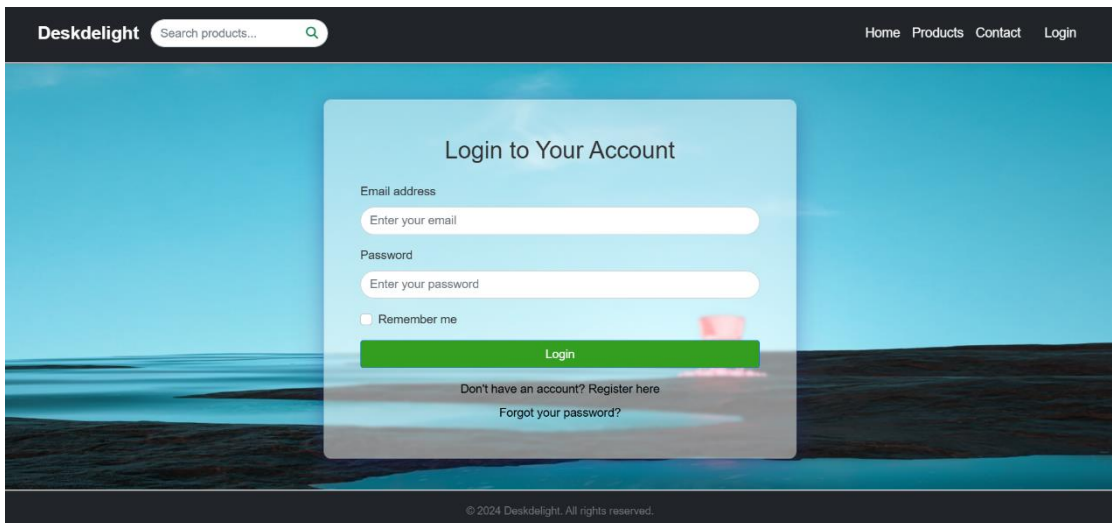
## **10. ANNEXURE**

## 10. SCREENSHOTS

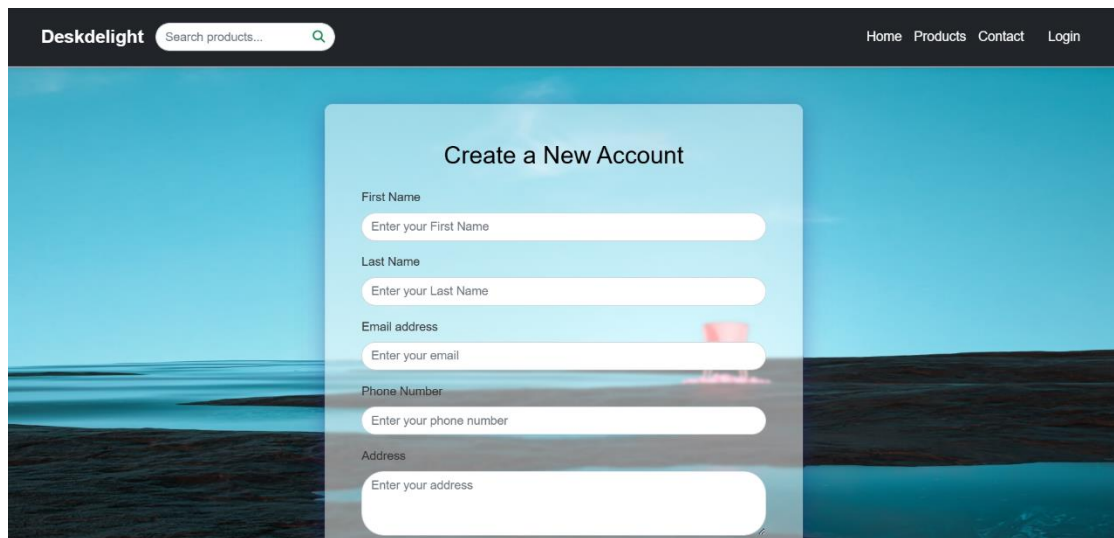
### 10.1 Home page



### 10.2 Login page



### 10.3 Register page

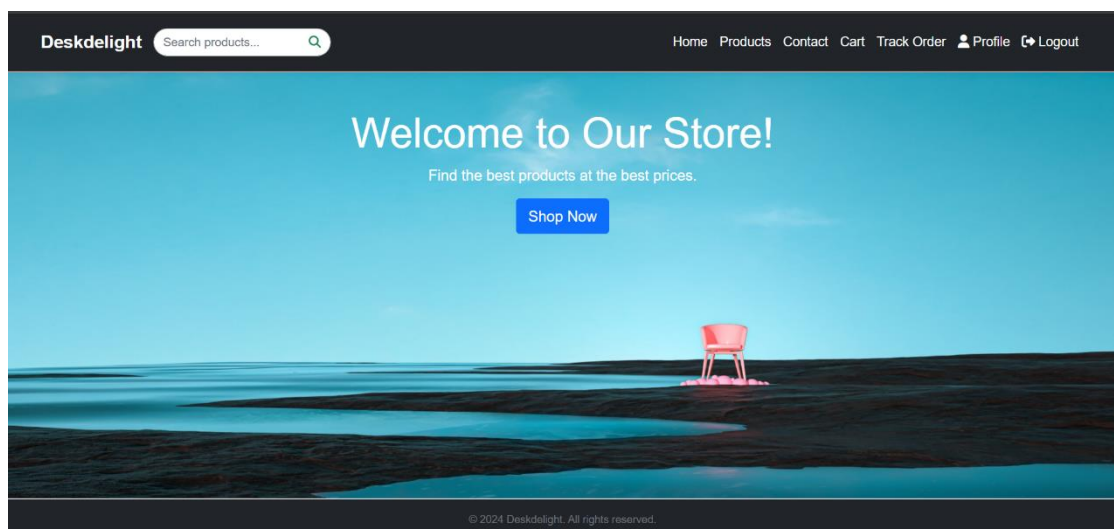


The screenshot shows the 'Create a New Account' registration form on the Deskdelight website. The form is centered on a background image of a beach with a red lifeguard stand in the distance. The form fields are as follows:

- First Name**: Input field with placeholder text 'Enter your First Name'.
- Last Name**: Input field with placeholder text 'Enter your Last Name'.
- Email address**: Input field with placeholder text 'Enter your email'.
- Phone Number**: Input field with placeholder text 'Enter your phone number'.
- Address**: Input field with placeholder text 'Enter your address'.

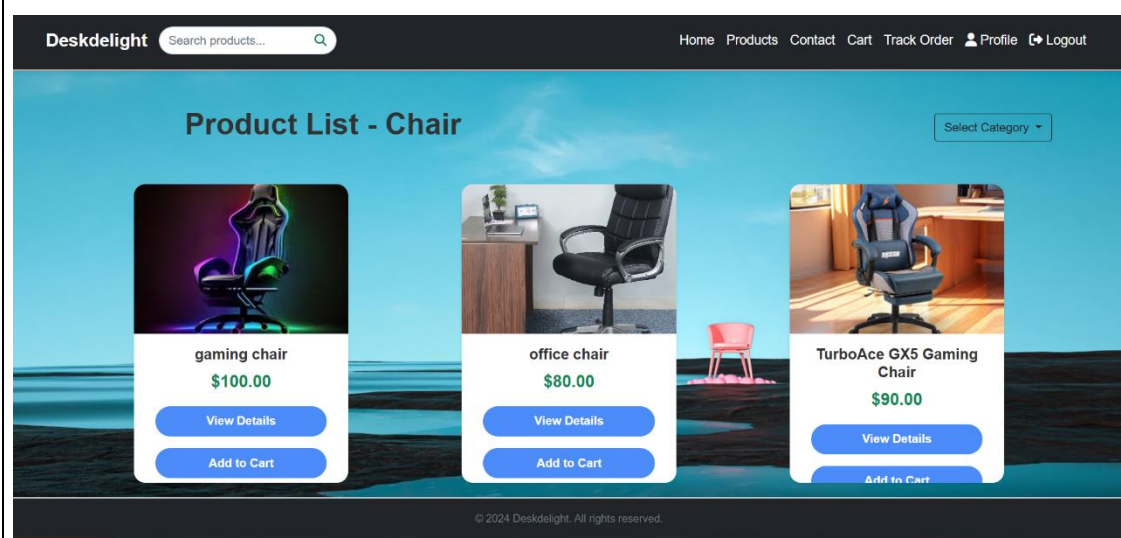
The top navigation bar includes the Deskdelight logo, a search bar, and links for Home, Products, Contact, and Login.

### 10.4 User page

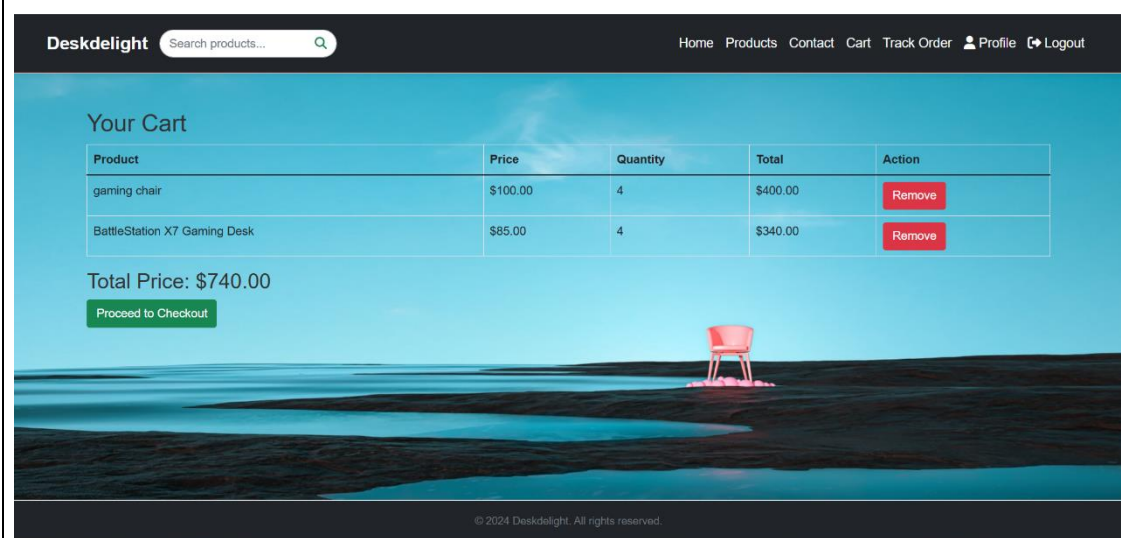




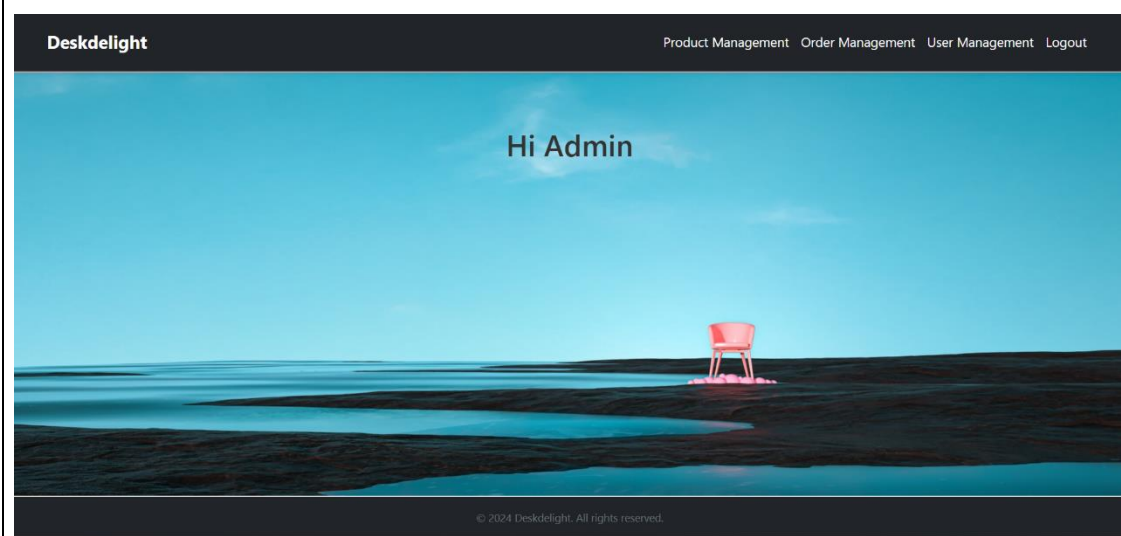
10.5 Product list page



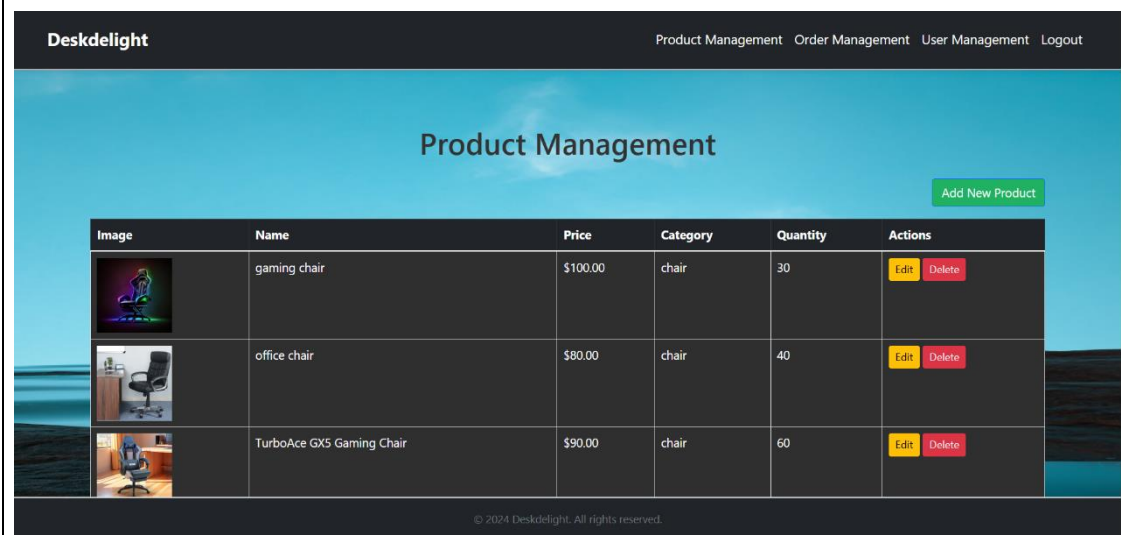
10.6 cart page



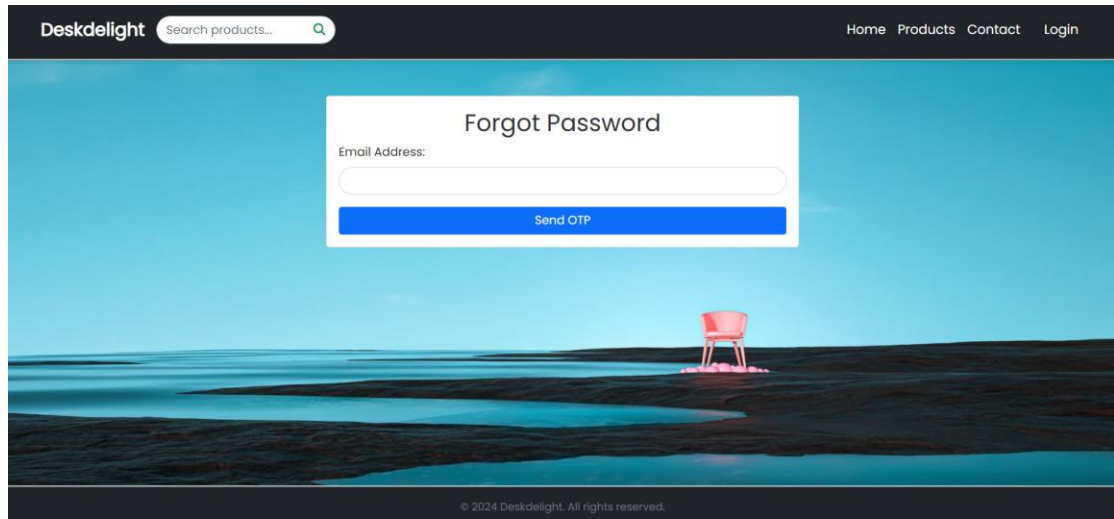
10.6 Admin page



10.7 product management page



### *10.8 Forgot password*



Deskdelight Search products... Q

Home Products Contact Login

### Forgot Password

Email Address:

Send OTP

© 2024 Deskdelight. All rights reserved.