ONLINE SHOPPING

AIM:

The ‘Online Shopping System’ Services department strives to provide solutions to develop and transfer easy and efficient way in the digital age and to help reduces the human pressure and time.

PROBLEM STATEMENT:

Despite the growth of online shopping, many platforms still struggle with issues related to user experience, payment security, inventory management, and personalized shopping experiences. Our goal is to address these challenges by creating an online shopping platform that not only meets but exceeds user expectations.

SOFTWARE REQUIREMENT SPECIFICATION:

Creating a Software Requirements Specification (SRS) for an online shopping platform involves detailing the functional and non-functional requirements, user interface designs, and other specifications necessary for the development and implementation of the system. Here's a structured outline for an SRS document for an online shopping system:

Purpose

The purpose of this document is to describe the functional and non-functional requirements for the development of an online shopping system.

Scope

This document covers all aspects of the online shopping platform including user registration, product brows Books and Publications:

Reference:

- *Books and Publications*:

"Designing Data-Intensive Applications" by Martin Kleppmann: This book provides principles for designing scalable and maintainable systems, which are crucial for online shopping platforms.

” Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown: A practical guide for building modern web applications, including e-commerce sites, using JavaScript.

- *Official Documentation and APIs*:

Stripe API Documentation: Comprehensive guide on integrating payment processing with Stripe.

PayPal Developer Documentation: Resources and guides for integrating PayPal into e-commerce websites.

Amazon Web Services (AWS) Documentation: For cloud-based infrastructure and services that support e-commerce platforms.

- *Case Studies and White Papers*:

"Case Study: How Amazon.com Became the King of E-commerce" on Investopedia: Insights into Amazon's strategies and technologies.

"E-commerce Usability: The Guidelines According to the Baynard Institute": Research-based guidelines to enhance the usability of e-commerce sites.

Technologies Used:

E-commerce Platforms

Payment Gateways

Mobile Applications

Internet of Things (IoT)

Tools to be Used:

Star UML

Python

OVERVIEW:

Objective:

The objective of online shopping is to provide a convenient, efficient, and accessible way for consumers to purchase goods and services. It aims to offer a seamless 24/7 shopping experience, a broad range of products, easy price comparisons, personalized recommendations, and various secure payment options.

Key Features:

Convenience: Shop anytime, anywhere with internet access.

Variety and Accessibility: Access to a vast selection of products and brands.

Price Comparisons and Reviews: Compare prices and read customer reviews easily.

Personalization: Recommendations based on browsing and purchase history.

Payment Options: Multiple payment methods including credit/debit cards and digital wallets.

Expected Outcomes:

Increased Consumer Convenience: Higher satisfaction due to flexibility and ease of access.

Broader Market Reach for Retailers: Expanded market beyond local boundaries.

Improved Sales and Revenue: Increased sales from 24/7 operations.

Data-Driven Insights: Valuable consumer data to improve products and marketing.

Enhanced Customer Experience: Personalized shopping and better customer loyalty.

Technological Advancements: Ongoing improvements in AI and AR to enhance shopping.

OVERALL DESCRIPTION:

Product Perspective:

From a product perspective, online shopping involves a digital marketplace where various products and services are listed by vendors for purchase by consumers. This marketplace is accessible through e-commerce websites or apps, offering a vast array of categories, such as electronics, clothing, groceries, and more. Each product typically includes detailed descriptions, images, prices, and customer reviews, helping buyers make informed decisions.

Software Interface:

The software interface of an online shopping platform includes the front-end and back-end components:

Front-End: The user interface (UI) that customers interact with. This includes web pages or mobile app screens where users can search for products, view product details, add items to the cart, and complete purchases.

Back-End: The server-side software that manages databases, handles transactions, processes orders, and communicates with third-party services like payment gateways and shipping providers.

Hardware Interface:

The hardware interface involves the physical infrastructure that supports online shopping platforms:

Servers:

Host the e-commerce website or app, databases, and handle network traffic.

User Devices:

Includes desktops, laptops, smartphones, and tablets that customers use to access the online store.

Payment Terminals:

Securely process online transactions.

Warehousing and Logistics Equipment:

Supports inventory management and order fulfillment.

Software Function:

Key software functions of an online shopping platform include:

Product Search and Filtering:

Allows users to search for products using keywords and filters.

Product Display:

Shows detailed product information, images, and reviews.

Shopping Cart and Checkout:

Enables users to add items to a cart and proceed through a secure checkout process.

Payment Processing:

Facilitates various payment methods like credit cards, digital wallets, and buy-now-pay-later options.

Order Management:

Tracks order status, processing, and shipping.

User Accounts:

Manages user profiles, order history, and wish lists.

Customer Support:

Provides help through chatbots, FAQs, and customer service contacts.

Use Characteristics:

Use characteristics of online shopping platforms include:

Accessibility:

Available 24/7 from any location with internet access.

Ease of Use:

User-friendly interfaces that facilitate easy navigation and purchase.

Personalization:

Recommends products based on user behavior and preferences.

Security: Implements measures like SSL encryption to protect user data and transactions.

Scalability:

Can handle varying amounts of traffic and transactions without performance degradation.

Constraints:

Constraints in online shopping systems include:

Security Risks:

Vulnerability to cyber-attacks, requiring robust security measures.

Privacy Concerns:

Ensuring user data is protected and used ethically.

Technical Issues:

Downtime, slow loading times, and bugs can affect the user experience.

Logistics:

Dependence on reliable delivery services for timely order fulfillment.

Regulatory Compliance:

Adherence to laws and regulations regarding online commerce, data protection, and consumer rights.

User Trust:

Building and maintaining consumer trust through reliable service and transparent practices.

UML DIAGRAMS:

The following UML diagrams describe the process involved in the online recruitment system

- Use case diagram
- Class diagram
- Sequence diagram

USE CASE DIAGRAM:

This diagram will contain the actors, use cases which are given below

Actors:

Users or customer, Admin

Use case:

Login

Register

View product

Ordered product

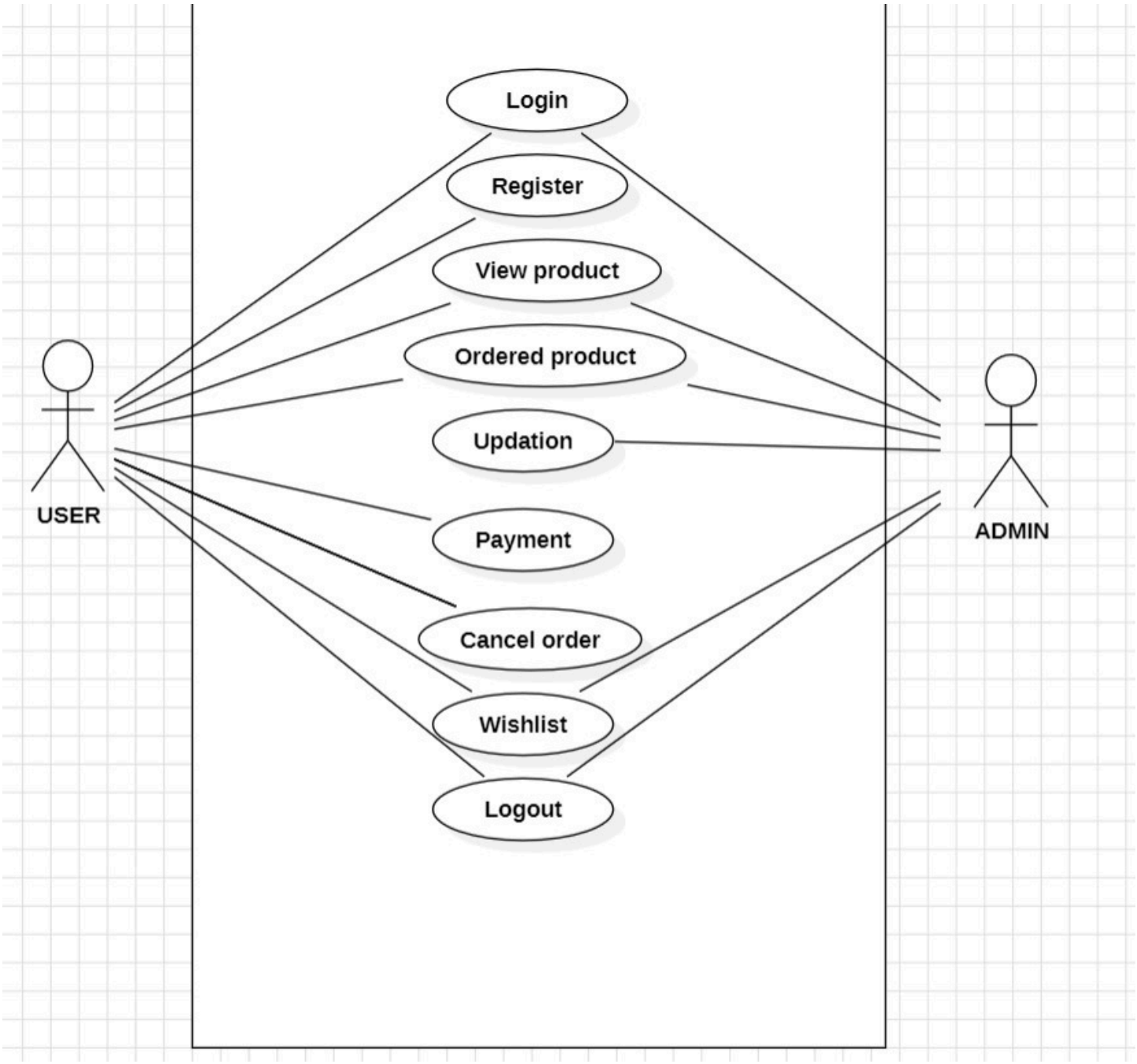
Updation

Payment

Cancel order

Wishlist

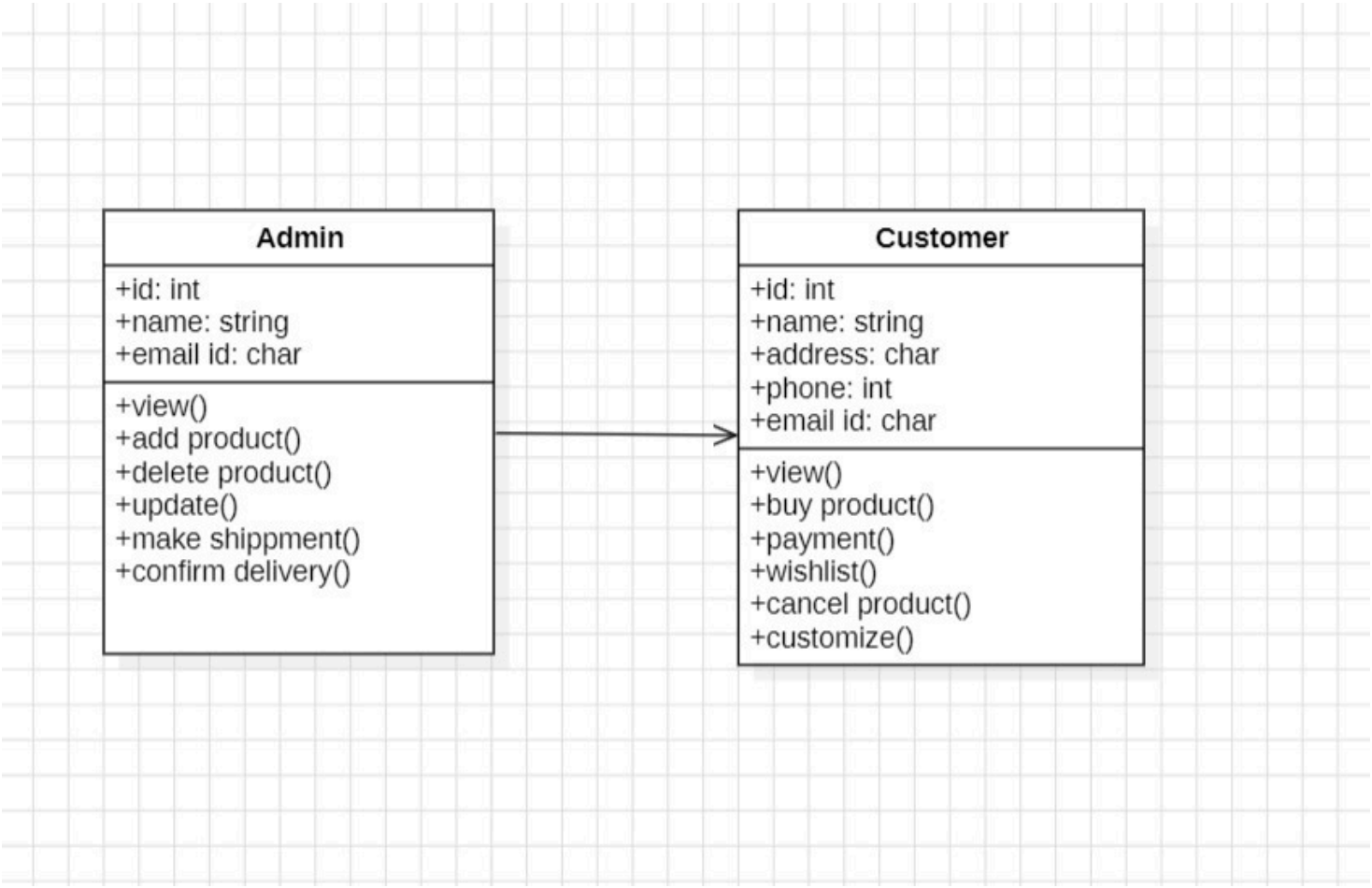
Logout



CLASS DIAGRAM:

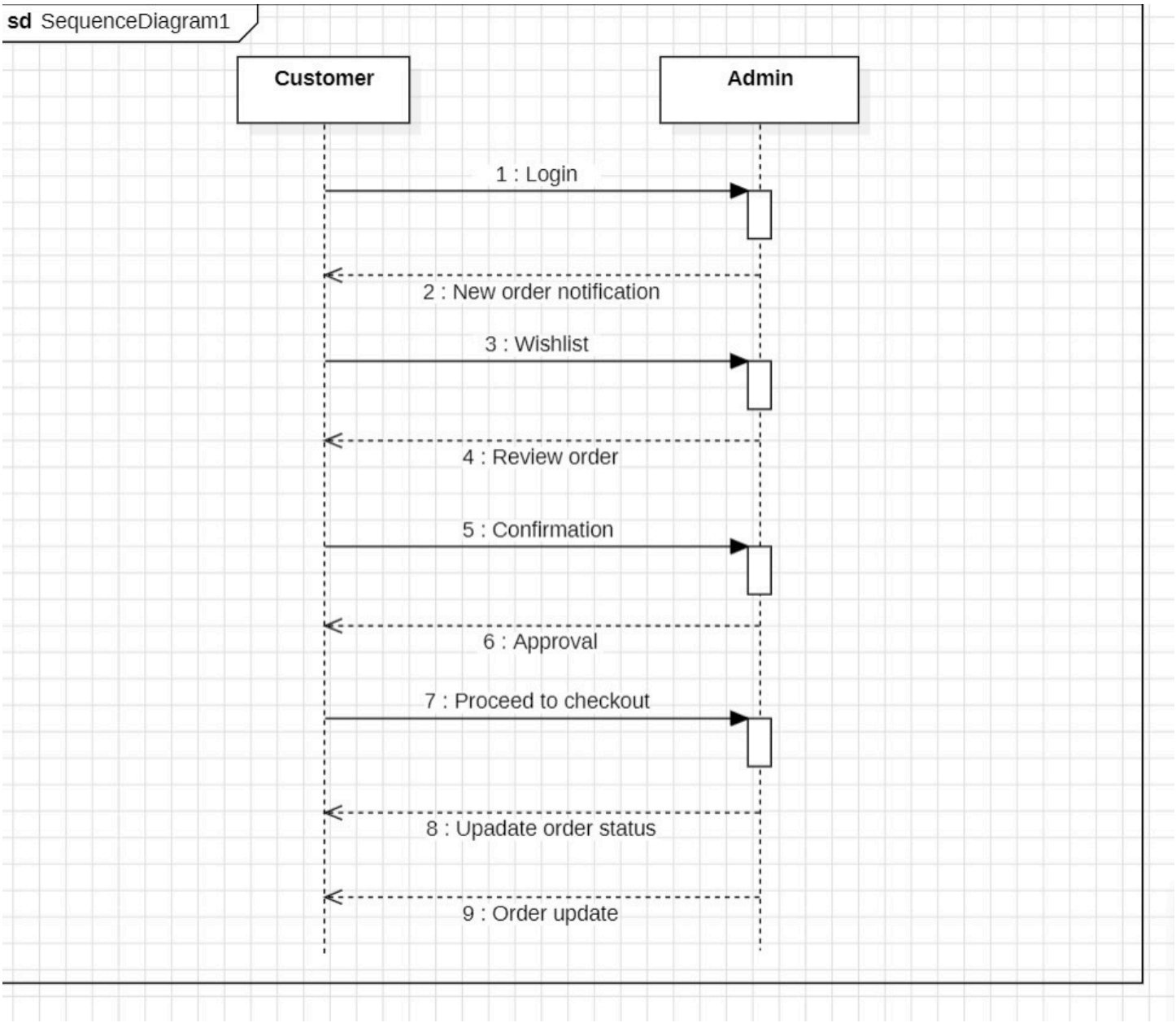
This diagram consists of the following classes, attributes and their operations.

CLASSES	ATTRIBUTES	OPERATIONS
Admin	Id Name Email id	View() Add product() Delete product() Update() Make shipment() Confirm delivery()
User or Customer	Id Name Address Phone Email id	View() Buy product() Payment() Wishlist() Cancel product()



SEQUENCE DIAGRAM:

This diagram consists of the objects, messages and return messages. Object: Customer, Stock dealer, Central stock system



PROGRAM:

```
import tkinter as tk

from tkinter import ttk, messagebox

# Sample product data

products = [

    {"name": "Dress 1", "price": "$49.99", "description": "Floral Print Maxi Dress", "sizes": ["S", "M", "L", "XL"], "colors": ["Red", "Blue", "Green"], "gender": "Women"},

    {"name": "Dress 2", "price": "$59.99", "description": "Ruffle Wrap Dress", "sizes": ["S", "M", "L"], "colors": ["Black", "White", "Pink"], "gender": "Women"},

    {"name": "Dress 3", "price": "$69.99", "description": "Off-Shoulder Midi Dress", "sizes": ["S", "M", "L", "XL"], "colors": ["Blue", "Yellow", "Orange"], "gender": "Women"},
```

{"name": "Dress 4", "price": "$79.99", "description": "Lace Bodycon Dress", "sizes": ["S", "M", "L"], "colors": ["Red", "White", "Black"], "gender": "Women"},

{"name": "Dress 5", "price": "$89.99", "description": "Embroidered Shift Dress", "sizes": ["S", "M", "L", "XL"], "colors": ["Pink", "Purple", "Green"], "gender": "Women"},

{"name": "Shirt 1", "price": "$29.99", "description": "Casual Plaid Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["Red", "Blue", "Green"], "gender": "Men"},

{"name": "Shirt 2", "price": "$39.99", "description": "Striped Oxford Shirt", "sizes": ["S", "M", "L"], "colors": ["Black", "White", "Pink"], "gender": "Men"},

{"name": "Shirt 3", "price": "$49.99", "description": "Classic Fit Polo Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["Blue", "Yellow", "Orange"], "gender": "Men"},

{"name": "Shirt 4", "price": "$59.99", "description": "Printed Graphic T-Shirt", "sizes": ["S", "M", "L"], "colors": ["Red", "White", "Black"], "gender": "Men"},

{"name": "Shirt 5", "price": "$69.99", "description": "Slim Fit Button-Down Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["Pink", "Purple", "Green"], "gender": "Men"},

{"name": "Dress 6", "price": "$99.99", "description": "Elegant Evening Gown", "sizes": ["S", "M", "L", "XL"], "colors": ["Black", "Blue", "Red"], "gender": "Women"},

{"name": "Dress 7", "price": "$109.99", "description": "Sleeveless Cocktail Dress", "sizes": ["S", "M", "L"], "colors": ["White", "Pink", "Purple"], "gender": "Women"},

{"name": "Dress 8", "price": "$119.99", "description": "Embellished Party Dress", "sizes": ["S", "M", "L", "XL"], "colors": ["Gold", "Silver", "Black"], "gender": "Women"},

{"name": "Dress 9", "price": "$129.99", "description": "Bohemian Maxi Dress", "sizes": ["S", "M", "L"], "colors": ["Green", "Yellow", "Orange"], "gender": "Women"},

{"name": "Dress 10", "price": "$139.99", "description": "Satin Slip Dress", "sizes": ["S", "M", "L", "XL"], "colors": ["Pink", "Purple", "Blue"], "gender": "Women"},

{"name": "Shirt 6", "price": "$79.99", "description": "Casual Linen Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["White", "Beige", "Blue"], "gender": "Men"},

{"name": "Shirt 7", "price": "$89.99", "description": "Hawaiian Print Shirt", "sizes": ["S", "M", "L"], "colors": ["Blue", "Green", "Orange"], "gender": "Men"},

{"name": "Shirt 8", "price": "$99.99", "description": "Slim Fit Dress Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["White", "Black", "Gray"], "gender": "Men"},

{"name": "Shirt 9", "price": "$109.99", "description": "Long Sleeve Polo Shirt", "sizes": ["S", "M", "L"], "colors": ["Red", "Blue", "Green"], "gender": "Men"},

{"name": "Shirt 10", "price": "$119.99", "description": "Crew Neck T-Shirt", "sizes": ["S", "M", "L", "XL"], "colors": ["Black", "White", "Gray"], "gender": "Men"}

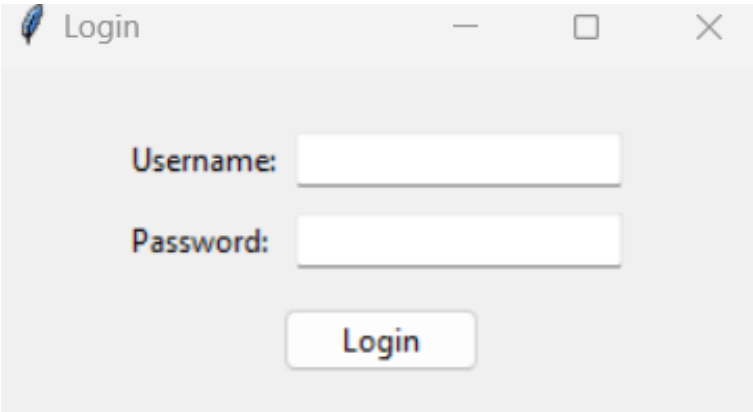
```
]

# Function to handle login

def login():

    # Replace this with actual login functionality

    messagebox.showinfo("Login", "Login successful!")

# Function to display products in the shopping window

def display_products(products, start_index, end_index):

    # Clear existing product frames

    for frame in product_listing_frame.winfo_children():

        frame.destroy()

 # Display products for the current page

    row = 0

    col = 0

    for i in range(start_index, end_index):

        product = products[i]

        create_product_frame(product, row, col)

        row += 1

        if row > 1:

            row = 0

            col += 1

# Function to create a frame for a product

def create_product_frame(product, row, col):

    product_frame = ttk.Frame(product_listing_frame, padding=10, relief="raised")

    product_frame.grid(row=row, column=col, padx=10, pady=5, sticky="ew")

    name_label = ttk.Label(product_frame, text=product["name"], font=("Helvetica", 14, "bold"))

    name_label.grid(row=0, column=0, columnspan=2, pady=(0, 5), sticky="w")

    price_label = ttk.Label(product_frame, text=product["price"])
```

```
    price_label.grid(row=1, column=0, pady=(0, 5), sticky="w")

    view_button = ttk.Button(product_frame, text="View Details", command=lambda p=product: view_product_details(p))

    view_button.grid(row=1, column=1, pady=(0, 5), sticky="e")

    add_to_cart_button = ttk.Button(product_frame, text="Add to Cart", command=lambda p=product: add_to_cart(p))

    add_to_cart_button.grid(row=2, column=0, columnspan=2, pady=(0, 5), sticky="ew")

# Function to handle clicking on the "View Details" button

def view_product_details(product):

    messagebox.showinfo("Product Details", f"Name: {product['name']}\nPrice: {product['price']}\nDescription: {product['description']}\nSizes: {', '.join(product['sizes'])}\nColors: {', '.join(product['colors'])}")

# Function to handle clicking on the "Add to Cart" button

def add_to_cart(product):

    add_to_cart_window = tk.Toplevel(root)

    add_to_cart_window.title("Add to Cart")

    payment_method_label = ttk.Label(add_to_cart_window, text="Select Payment Method:")

    payment_method_label.grid(row=0, column=0, padx=10, pady=5, sticky="w")

    payment_methods = ["Credit Card", "Debit Card", "PayPal"]

    payment_method_var = tk.StringVar(value=payment_methods[0])

    payment_method_combobox = ttk.Combobox(add_to_cart_window, textvariable=payment_method_var, values=payment_methods, state="readonly")

    payment_method_combobox.grid(row=0, column=1, padx=10, pady=5, sticky="w")

    delivery_address_label = ttk.Label(add_to_cart_window, text="Delivery Address:")

    delivery_address_label.grid(row=1, column=0, padx=10, pady=5, sticky="w")

    delivery_address_entry = ttk.Entry(add_to_cart_window)

    delivery_address_entry.grid(row=1, column=1, padx=10, pady=5, sticky="w")

    delivery_mode_label = ttk.Label(add_to_cart_window, text="Select Delivery Mode:")

    delivery_mode_label.grid(row=2, column=0, padx=10, pady=5, sticky="w")
```

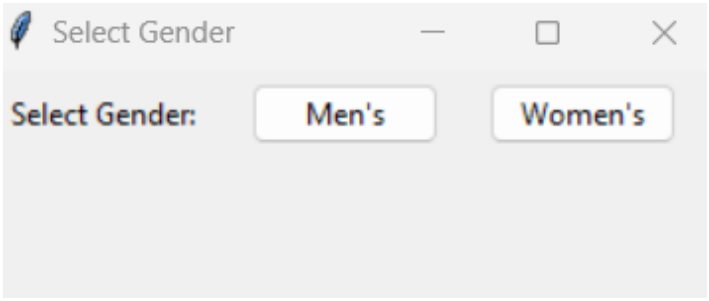
```
delivery_modes = ["Standard", "Express"]
    delivery_mode_var = tk.StringVar(value=delivery_modes[0])
    delivery_mode_radiobuttons = []
    for i, mode in enumerate(delivery_modes):
        rb = ttk.Radiobutton(add_to_cart_window, text=mode, variable=delivery_mode_var, value=mode)
        rb.grid(row=2, column=i+1, padx=10, pady=5, sticky="w")
        delivery_mode_radiobuttons.append(rb)
    confirm_button = ttk.Button(add_to_cart_window, text="Confirm", command=lambda:
confirm_order(add_to_cart_window, product))
    confirm_button.grid(row=3, column=0, columnspan=2, padx=10, pady=10)
# Function to confirm the order
def confirm_order(window, product):
    window.destroy()
    messagebox.showinfo("Order Confirmation", f"Your order for {product['name']} has been placed
successfully!")
# Function to handle clicking on the "Place Order" button
def place_order():
    messagebox.showinfo("Order Confirmation", "Your order has been placed successfully!")
# Function to handle clicking on the "Login" button
def login_clicked():
    username = username_entry.get()
    password = password_entry.get()
    if username == "user" and password == "password":
        login()
        root.deiconify()  # Show the main window
        login_window.destroy()  # Close the login window
        open_shopping_window()  # Open the shopping window after successful login
    else:
```

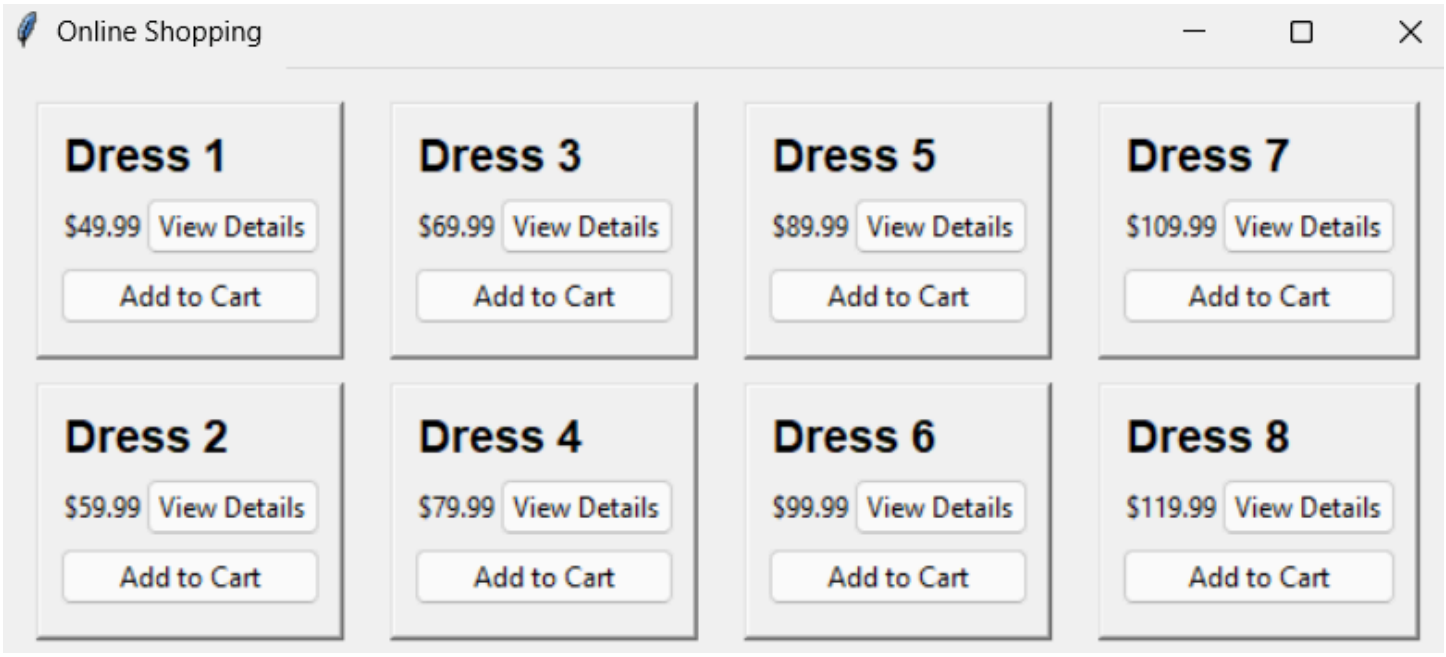
```
        messagebox.showerror("Login Failed", "Invalid username or password.")

# Function to open the shopping window

def open_shopping_window():

    shopping_window = tk.Toplevel(root)

    shopping_window.title("Select Gender")

    shopping_window.geometry("300x150")

    shopping_window.attributes("-topmost", True)

    select_gender_label = ttk.Label(shopping_window, text="Select Gender:")

    select_gender_label.grid(row=0, column=0, padx=10, pady=5, sticky="w")

   men_button = ttk.Button(shopping_window, text="Men's", command=lambda:
display_products_by_gender("Men"))

    men_button.grid(row=0, column=1, padx=10, pady=5, sticky="w")

    women_button = ttk.Button(shopping_window, text="Women's", command=lambda:
display_products_by_gender("Women"))

    women_button.grid(row=0, column=2, padx=10, pady=5, sticky="w")

# Function to display products by gender

def display_products_by_gender(gender):

    filtered_products = [product for product in products if product["gender"] == gender]

    display_products(filtered_products, 0, min(len(filtered_products), 8))

# Create main window

root = tk.Tk()

root.title("Online Shopping")

# Create login window

login_window = tk.Toplevel(root)

login_window.title("Login")

login_window.geometry("300x150")

login_window.attributes("-topmost", True)

login_frame = ttk.Frame(login_window)
```

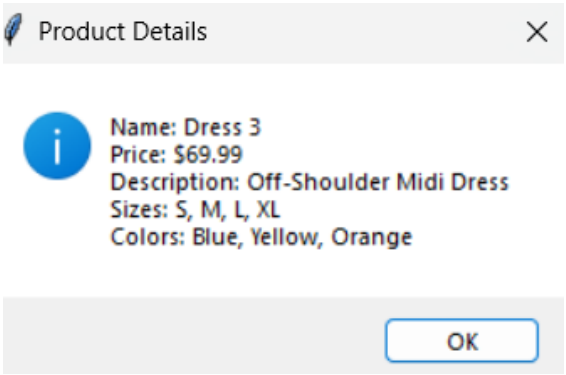
```
login_frame.pack(padx=20, pady=20)

username_label = ttk.Label(login_frame, text="Username:")

username_label.grid(row=0, column=0, sticky="w")

username_entry = ttk.Entry(login_frame)

username_entry.grid(row=0, column=1, padx=5, pady=5)

password_label = ttk.Label(login_frame, text="Password:")

password_label.grid(row=1, column=0, sticky="w")

password_entry = ttk.Entry(login_frame, show="*")

password_entry.grid(row=1, column=1, padx=5, pady=5)

login_button = ttk.Button(login_frame, text="Login", command=login_clicked)

login_button.grid(row=2, column=0, columnspan=2, pady=10)

# Hide main window until login

root.withdraw()

# Frame to display products

product_listing_frame = ttk.Frame(root)

product_listing_frame.pack(padx=10, pady=10)

# Run the main event loop

root.mainloop()
```

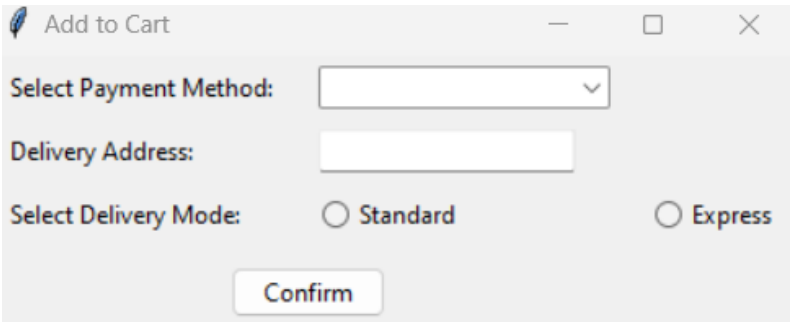


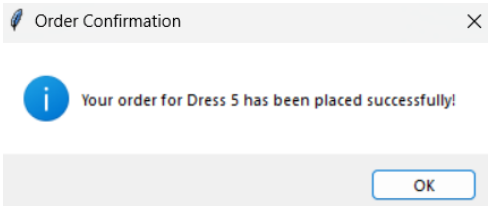

Login
Login successful!
OK


Select Gender
Select Gender:
Men's
Women's


Online Shopping
Dress 1
$49.99
View Details
Add to Cart
Dress 3
$69.99
View Details
Add to Cart
Dress 5
$89.99
View Details
Add to Cart
Dress 7
$109.99
View Details
Add to Cart
Dress 2
$59.99
View Details
Add to Cart
Dress 4
$79.99
View Details
Add to Cart
Dress 6
$99.99
View Details
Add to Cart
Dress 8
$119.99
View Details
Add to Cart


Product Details
Name: Dress 3
Price: $69.99
Description: Off-Shoulder Midi Dress
Sizes: S, M, L, XL
Colors: Blue, Yellow, Orange
OK


Add to Cart
Select Payment Method:
Delivery Address:
Select Delivery Mode:
Standard
Express
Confirm



RESULT:

Thus, online shopping is experiencing continuous growth, impacting traditional retail while offering convenience and global reach. Mobile shopping is on the rise, presenting both challenges and opportunities for retailers. The future promises further advancements in technology, shaping the online shopping experience.