SYNOPSIS

Report on

Rezy

by

Vishal chaturvedi (202410116100248)
Simran Tyagi (202410116100208)
Shefali Yadav (202410116100195)
Vanshika Garg (202410116100235)

Session:2025-2026 (III semester)

Under the supervision of
Dr. Anita Yadav
Dr. Amit Kumar Gupta
Ms. Shruti Aggarwal

KIET Group of Institutions, Delhi-NCR, Ghaziabad

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206

ABSTRACT

The rapid digitalization of the food and beverage industry has transformed the way customers interact with restaurants and cafés. Traditional food ordering methods are often time-consuming and prone to inefficiencies such as long queues, manual errors, and lack of personalization. To address these challenges, this project introduces **Rezy – Online Café Food Booking System**, a web-based application designed to simplify and automate the process of ordering food in cafés and restaurants.

Rezy is built using **Flask (Python framework)** for backend development, **MySQL** for secure and structured data management, and **Docker** for containerization and efficient deployment. The system provides a user-friendly interface where customers can browse menus, select food items, place orders, and receive confirmation in real time. On the other side, restaurant administrators can manage menus, track customer orders, and update availability seamlessly.

The project aims to deliver an efficient, scalable, and reliable solution that enhances customer convenience while improving restaurant workflow management. By integrating modern web technologies, Rezy ensures minimal manual intervention, faster order processing, and improved customer satisfaction. This system can be extended further to include features such as online payments, loyalty rewards, and AI-driven recommendations, making it a robust platform for digital café management.

Keywords: Online Food Ordering, Café Management System, Restaurant Automation, Real-Time Food Ordering, User-Friendly Interface, Scalable Web Solution

TABLE OF CONTENTS

1. Introduction

In today's fast-paced digital world, the food and beverage industry is rapidly adopting technology to enhance customer convenience and improve operational efficiency. Traditional methods of food ordering in cafés and restaurants often involve manual booking, long waiting times, and communication gaps between customers and staff. To address these challenges, online food booking systems have emerged as an effective solution.

Rezy – Online Café Food Booking System is a web-based application developed using **Flask (Python framework)**, **MySQL database**, and **Docker** for containerized deployment. The primary aim of this project is to provide a seamless platform where customers can browse the café menu, place orders online, and receive instant confirmation without unnecessary delays. For café owners and staff, it offers efficient order management, real-time updates, and data-driven insights into customer preferences.

This project not only digitalizes the café ordering process but also ensures **user-friendliness, scalability, and reliability**. By integrating modern web technologies, Rezy reduces manual workload, minimizes errors, and enhances the overall dining experience. With the growing demand for smart restaurant solutions, this project demonstrates how technology can transform a traditional café into a modern, customer-centric service provider.

2. Literature Review

- **Initial Systems (2000s):** Early food ordering platforms were static websites with limited menu options and no real-time updates (Kimes, 2011).
- **Third-party Platforms (2010s):** Apps like Zomato, Swiggy, and Uber Eats revolutionized convenience but imposed high commission fees and reduced restaurant autonomy (Jhamb & Aggarwal, 2020).
- **Restaurant-owned Systems (2015+):** Cloud-based POS and web apps emerged, enabling restaurants to build their own ordering & reservation systems (Kaur & Singh, 2019).
- **Integration & Automation:** Research emphasized integrating payment gateways, table reservations, and order tracking for enhanced customer experience (Choudhary et al., 2021).
- **Modern Approaches:** Use of Flask, MySQL, and containerization (Docker) allows lightweight, scalable, and portable solutions tailored for small restaurants and cafés (Gupta, 2022).
- **Technology Trends:** AI chatbots, recommendation systems, and predictive analytics are enhancing personalized food booking experiences (Forbes Tech Council, 2023).
- **Challenges:** Adoption barriers include digital literacy, cost, data security, and customer trust (NIST, 2022).
- **Research Gap:** Need for a low-cost, customizable, and café-specific solution that balances usability, efficiency, and modern features → *Rezy addresses this gap*.

3. Project / Research Objective

- **To develop a user-friendly online food booking system** that allows customers to browse menus, place orders, and reserve tables seamlessly.
- **To design a restaurant-centric platform** that reduces dependency on third-party apps and eliminates high commission costs.
- **To integrate secure payment gateways** ensuring safe and smooth financial transactions for both customers and restaurants.
- **To implement real-time order management** enabling restaurants to track, confirm, and update customer orders efficiently.
- **To provide a scalable and customizable solution** using technologies like Flask, MySQL, and Docker for easy deployment across small and medium restaurants.
- **To enhance customer experience** with features such as order tracking, digital receipts, and instant booking confirmations.
- **To analyze adoption challenges** (e.g., cost, usability, security) and propose ways to overcome them.
- **To contribute to digital transformation in the food service industry** by creating a cost-effective, flexible, and modern alternative to third-party platforms.

4. Hardware and Software Requirements

4.1. Hardware Requirements

To ensure smooth development, deployment, and execution of the Rezy system, the following minimum and recommended hardware specifications are proposed:

a) For Development Environment (Developer System)

- Processor (CPU): Intel i5 / AMD Ryzen 5 (minimum), Intel i7 or higher (recommended)
- RAM: 8 GB (minimum), 16 GB or more (recommended for Docker and database handling)
- Storage: 256 GB SSD (minimum), 512 GB or more (recommended)
- Display: 14” or larger with 1920x1080 resolution
- Network: Stable internet connection (at least 10 Mbps)

b) For Server Deployment (Production Environment)

- Processor (CPU): Quad-Core Intel Xeon / AMD EPYC or equivalent cloud-based CPU
- RAM: 16 GB (minimum), scalable to 32 GB+ based on concurrent users
- Storage: 500 GB SSD with high read/write speed
- Network Bandwidth: 100 Mbps or higher for handling real-time requests
- Cloud Hosting Options: AWS, Azure, Google Cloud, or DigitalOcean

c) For End-Users (Customer Devices)

- Device: Smartphone, Tablet, Laptop, or Desktop
- Processor: Any modern mobile/PC processor
- RAM: 2 GB minimum
- Storage: 50 MB free space (for browser cache or mobile app if extended in future)
- **Browser Support: Chrome, Firefox, Safari, Edge (latest versions)**

4.2 Software Requirements

The project uses a modern technology stack for development, deployment, and database management.

a) Development Tools

- **Operating System (OS):** Windows 10/11, Ubuntu 20.04+, macOS Monterey or later
- **IDE/Editor:** Visual Studio Code / PyCharm
- **Version Control:** Git & GitHub/GitLab for collaboration
- **Virtualization/Containerization:** Docker & Docker Compose

b) Backend

- **Programming Language:** Python 3.9+
- **Framework**: Flask (for REST APIs and backend logic)
- **Database:** MySQL (Relational Database Management System)
- **ORM (Optional):** SQL Alchemy for object-relational mapping
- **Authentication:** Flask-Login / JWT for user sessions

c) Frontend

- **Framework:** HTML5, CSS3, JavaScript (vanilla or Bootstrap/Tailwind for styling)
- **Responsive UI:** Bootstrap / Tailwind CSS for cross-device compatibility

d) Deployment

- **Containerization:** Docker for packaging and deployment
- **Web Server:** Nginx / Gunicorn for serving Flask app in production
- **Hosting:** Cloud hosting (AWS EC2, Azure VM, Heroku, or DigitalOcean)

e) Database Management

- **Database Server:** MySQL 8.0+
- **Database Client**: MySQL Workbench / phpMyAdmin
- **Backup & Recovery:** Automated database backup scripts with Docker volumes

f) Testing Tools

- **Unit Testing:** PyTest / Unittest (Python)
- **API Testing:** Postman / Insomnia
- **Load Testing:** Apache JMeter

5 Project Flow

a) User Registration & Authentication

- **New Users:** Register with details (name, email, phone, password).
- **Existing Users:** Login using credentials.
- **Authentication Layer:** Managed using Flask session handling & hashed passwords (bcrypt).
- Ensures secure access to the system.

b) Menu Browsing & Selection

- **After login, customers can:**
    - Browse food categories (Snacks, Meals, Beverages, Desserts).
    - View dish details (ingredients, price, availability).
    - Select multiple items and add them to the cart.
- **Menu data is dynamically fetched from MySQL database.**

c) Cart Management & Order Placement

- **Customers can:**
    - Review selected items.
    - Update quantity or remove items.
    - View real-time total price.
- **Once satisfied → Proceed to order booking.**

d) Order Processing & Database Storage

- **Order details are sent to the Flask backend.**
- **Data stored in MySQL database:**
    - User Info
    - Items Ordered
    - Order ID
    - Payment Status
    - Timestamp
- **Backend ensures data consistency & integrity.**

e) Order Confirmation & Notification

- System generates an order confirmation receipt.
- User receives confirmation via on-screen notification (and email/SMS if integrated).
- Café staff dashboard updates with new order details.

f) Payment Integration (Optional Enhancement)

- Integration of UPI/Wallet/Card gateways (Stripe/PayPal/Razorpay).
- Orders marked as "Paid" in database after successful transaction.

g) Admin / Café Management Dashboard

- **Admin can:**
    - Add / Update / Delete menu items.
    - Monitor active and past orders.
    - Track revenue & customer trends.
- **Access restricted via role-based authentication.**

h) Deployment with Docker

- Entire app (Flask + MySQL) containerized using Docker.
- Ensures easy deployment across cloud / local servers.
- Scalable and portable system architecture.

6. Project / Research Outcome

The implementation of Rezy – Online Café Food Booking System delivers significant outcomes for both end-users (customers) and café management (admins/staff). The outcomes are both technical and practical, ensuring efficiency, scalability, and better user experience.

1. Simplified Food Ordering for Customers

- Customers can easily browse café menus online.
- They can book food items in advance without standing in queues.
- Provides a user-friendly and responsive interface for quick order placement.

2. Enhanced Customer Experience

- Real-time order confirmation and notifications.
- Transparent price calculation with cart management.
- Reduces waiting time → increases customer satisfaction.

3. Efficient Café Management

- Café staff receives instant order updates.
- Centralized database of orders helps in better kitchen planning.
- Admin dashboard helps track sales, manage menu, and monitor performance**.**

4. Automation of Manual Tasks

- Removes the need for paper-based orders or manual tracking.
- Automates order logging, billing, and status management.
- Reduces chances of human error in order handling.

5. Scalable & Secure System

- Backend powered by Flask + MySQL ensures reliability and performance.

- Secure login/registration with password encryption.
- Scalable deployment using Docker containers, suitable for cafés of any size.

6. Research Contribution

- Demonstrates the practical use of web technologies (Flask, MySQL, Docker) in solving real-world problems.
- Can be extended with AI/ML for recommendation systems (e.g., suggesting food items based on history).
- Provides a base model for other food-service industries (restaurants, canteens, cloud kitchens).

7. Business & Social Impact

- Helps cafés handle large orders efficiently.
- Encourages digital adoption in small/medium food businesses.
- Promotes contactless food booking, supporting post-pandemic dining trends.

7. Proposed Time Duration

Phase 1: Requirement Analysis & Planning (Week 1 – Week 2)

- Collecting requirements from café owners & customers.
- Defining project scope and objectives.
- Preparing Software Requirement Specification (SRS).

Phase 2: System Design (Week 3 – Week 4)

- Designing **ER diagrams, UML diagrams, and Data Flow Diagrams (DFD)**.
- Creating **UI/UX wireframes** for customer and admin panels.
- Finalizing database schema (MySQL).

Phase 3: Backend & Database Development (Week 5 – Week 6)

- Setting up **Flask framework** for backend.
- Designing and implementing **MySQL database**.
- Implementing user authentication (login, signup).

Phase 4: Frontend Development (Week 7 – Week 8)

- Developing customer-facing pages (menu, cart, booking).
- Admin dashboard (orders, menu management, reports).
- Ensuring responsive design (mobile-friendly).

Phase 5: Integration & Testing (Week 9 – Week 10)

- Integrating **frontend with backend** APIs.
- Functional testing (order booking, notifications, payments simulation).

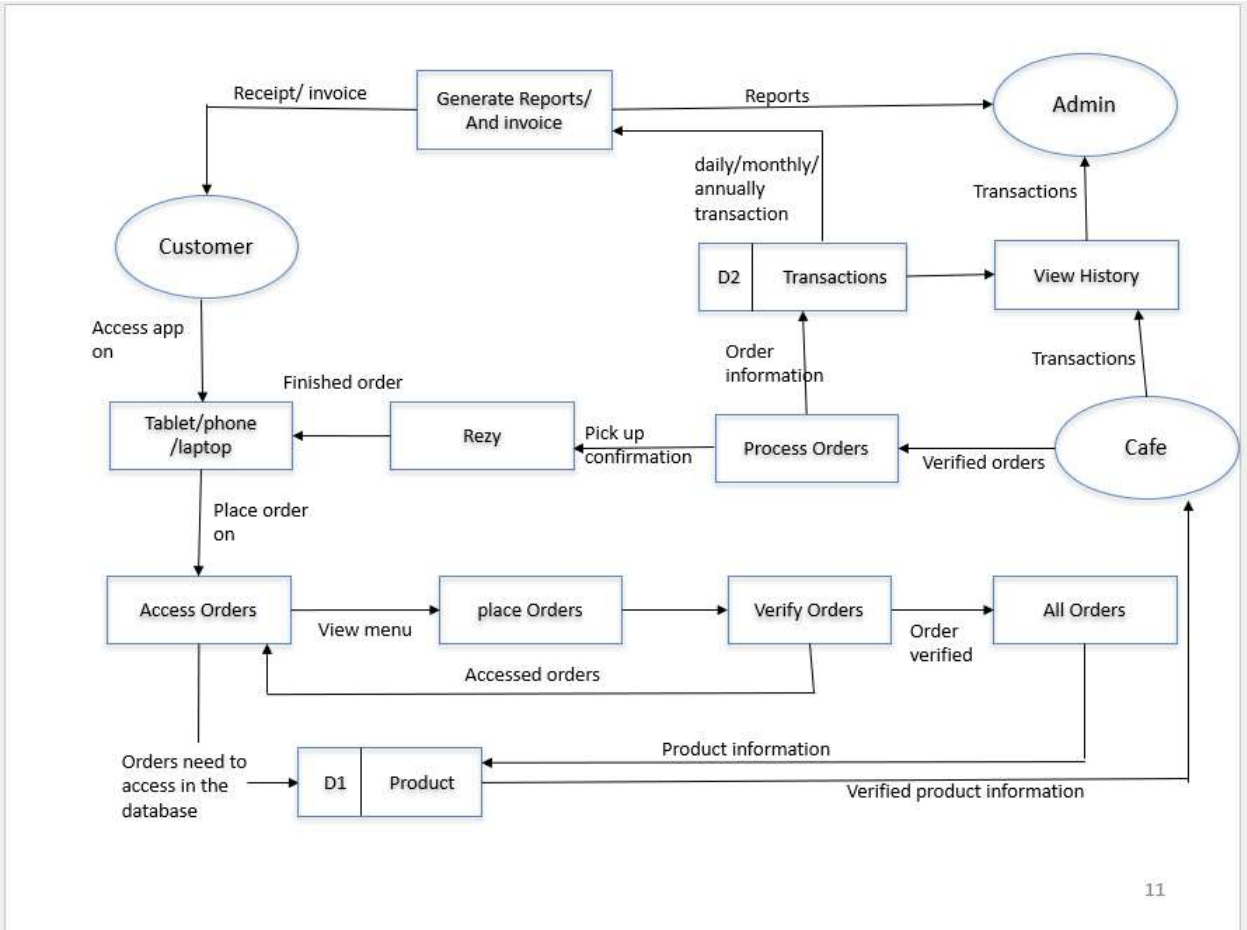
- Debugging and fixing errors.

Phase 6: Deployment & Documentation (Week 11 – Week 12)

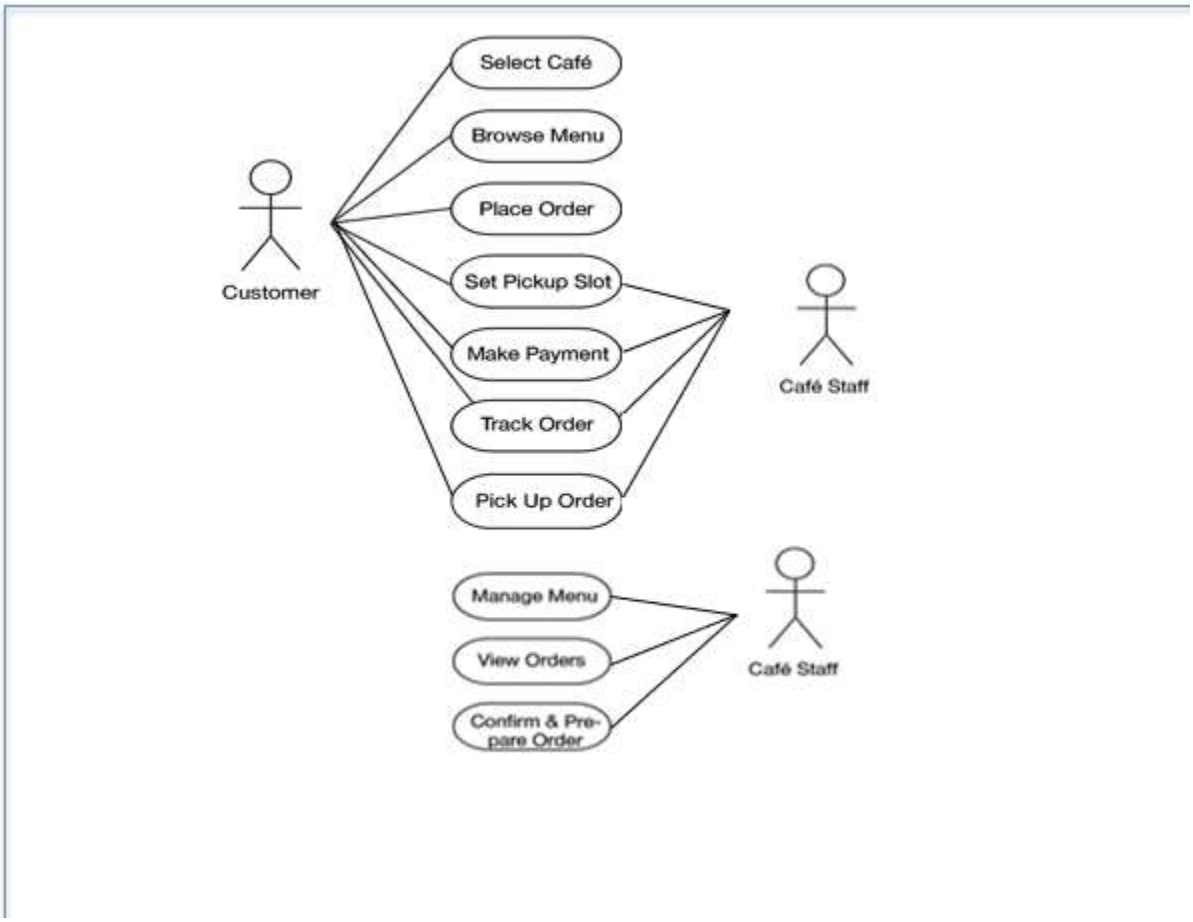
- Deploying application with **Docker** for scalability.
- Writing **project documentation & user manual**.
- Preparing final report & presentation.

8. Diagrams

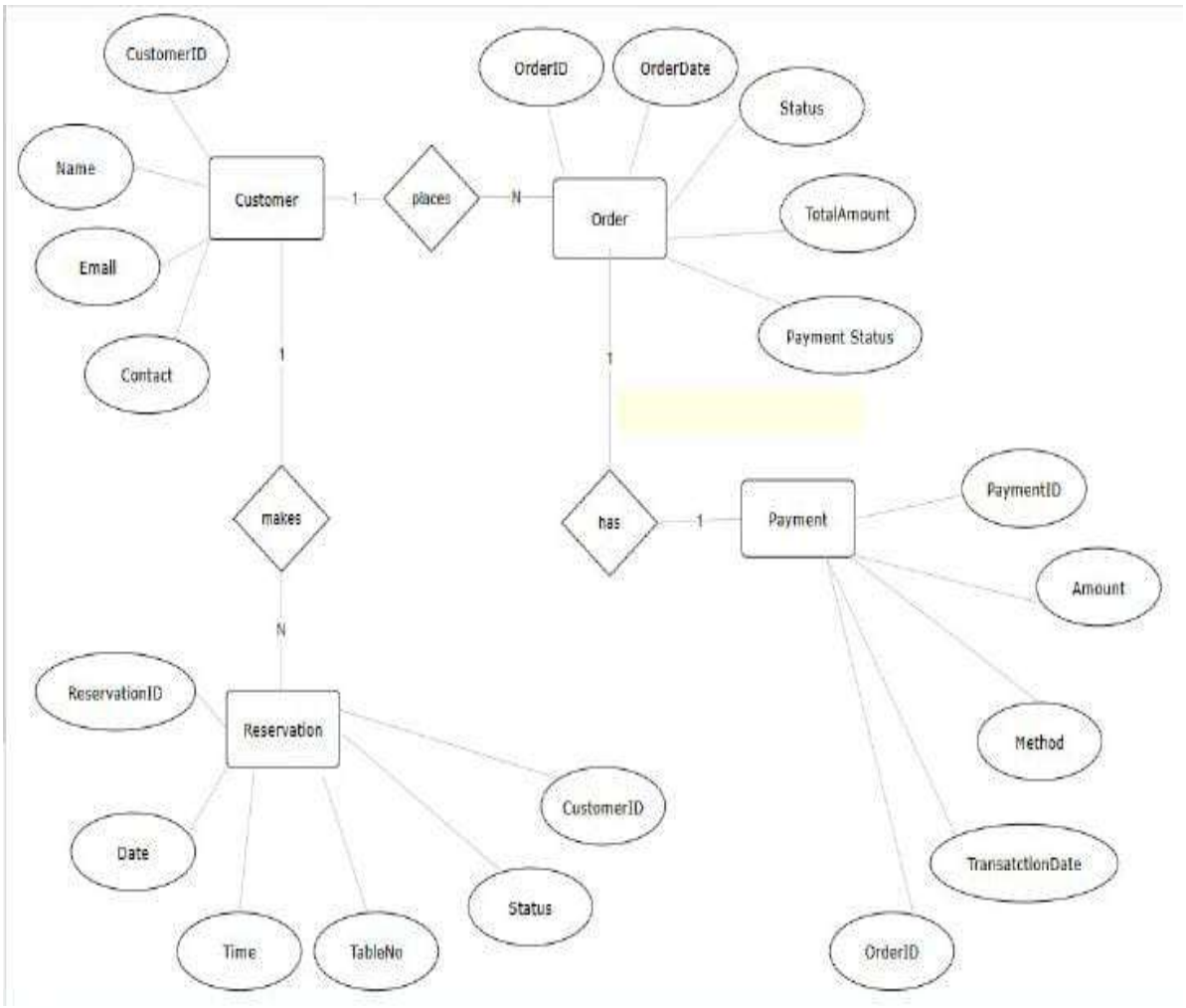
- **Data flow diagram**



- **Use case diagram**



- **E-R diagram**


CustomerID
Name
Email
Contact
Customer
1
places
N
OrderID
OrderDate
Status
Order
TotalAmount
Payment Status
1
1
makes
N
has
1
Payment
PaymentID
Amount
Method
TransatctionDate
OrderID
ReservationID
Reservation
CustomerID
Date
Time
TableNo
Status

10. REFERENCES

1. Phaal, R., Farrukh, C. J. P., & Probert, D. R. (2004). *Technology roadmapping—A planning framework for evolution and revolution.* Technological Forecasting and Social Change, 71(1-2), 5–26.
2. Lee, J., & Park, Y. (2005). *Customization of technology roadmaps according to roadmapping purposes: Overall process and detailed modules.* Technological Forecasting and Social Change, 72(5), 567–583.
3. Kerr, C., Phaal, R., & Probert, D. (2012). *Formalizing roadmapping for strategy alignment: Principles and practical approaches.* Research-Technology Management, 55(3), 57–66.
4. Smith, A., & Tran, M. (2018). *Agile roadmapping: Aligning product strategy with fast-changing markets.* Journal of Software Project Management, 12(4), 233–245.
5. Schilling, M. A. (2021). *Strategic Management of Technological Innovation.* McGraw-Hill Education.
6. Forrester Research. (2023). *The State of Product Roadmaps: AI, Cloud, and Automation in Strategy Planning.* Forrester Research Report.
7. National Institute of Standards and Technology (NIST). (2022). *Cybersecurity considerations for cloud-based applications.* U.S. Department of Commerce.
8. Gronroos, C. (2007). *Service Management and Marketing: Customer Management in Service Competition.* John Wiley & Sons.
9. Kurniawan, S., & Zaphiris, P. (2005). *Research-derived web design guidelines for older people.* Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility, 129–135.
10. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.