

**CASE STUDY  
ON  
WELLNESS-KART**

**Project Title:** Wellness-Kart: Automated E-Commerce Management System

**Subtitle:** A Spring Boot & MySQL Based Web Application for Wellness Products

**Presented By:** Devanand.S.K

**Register Number:** 43130366

**Department:** ECE

**Guided By:** Jayanth Vellingiri

**College/Institution:** Sathyabama University

## ABSTRACT

- Overview: Wellness-Kart is a centralized web platform designed to automate the sales and inventory management of personalized wellness products (bottles and fragrances).
- Problem Addressed: Replaces inefficient manual record-keeping and disjointed sales processes that lead to data errors and lack of business visibility.
- Solution: Implements a robust Spring Boot backend with a MySQL database to manage users, orders, and secure authentication.
- Key Outcome: Delivers streamlined operations for administrators and a seamless, digital ordering experience for customers.

# INTRODUCTION

- Domain: Web Application Development & E-Commerce.
- Core Concept: A specialized marketplace allowing users to customize wellness items (e.g., selecting specific bottle sizes and fragrance combinations).
- Motivation: The need to modernize traditional sales methods with a scalable, digital solution.
- Goal: To bridge the gap between wellness product suppliers and customers through automation and user-friendly design.

## **PROBLEM STATEMENT**

- Data Redundancy: Customer details were previously scattered across manual logbooks or spreadsheets, leading to duplication.
- Calculation Errors: Manual revenue tracking was prone to human error and time-consuming.
- Lack of Insights: Administrators had no real-time dashboard to view "Total Sales" or "Active User Counts."
- Poor UX: Customers had no digital way to track their order history or status.

## **OBJECTIVES OF THE PROJECT**

1. Automation: Automatically calculate order totals and track order status changes (Placed, Shipped, etc.).
2. Centralization: Store all User, Admin, and Order data in a single, secure Relational Database (RDBMS).
3. Role-Based Access: Create distinct, secure portals for Admins (Management) and Users (Customers).
4. Analytics: Provide immediate business insights via a dynamic Admin Dashboard.

# **SCOPE OF THE PROJECT**

## **In-Scope:**

- User Registration & Secure Login (Authentication).
- Product Selection (Bottle Types, Fragrances).
- Order Placement & History Tracking.
- Admin Dashboard & User Management.

## **Out-of-Scope:**

- Live Payment Gateway Integration (System uses "Payment on Handover").
- Third-party Logistics/Shipping API Integration.

# **SYSTEM ANALYSIS (EXISTING VS. PROPOSED)**

## **EXISTING SYSTEM:**

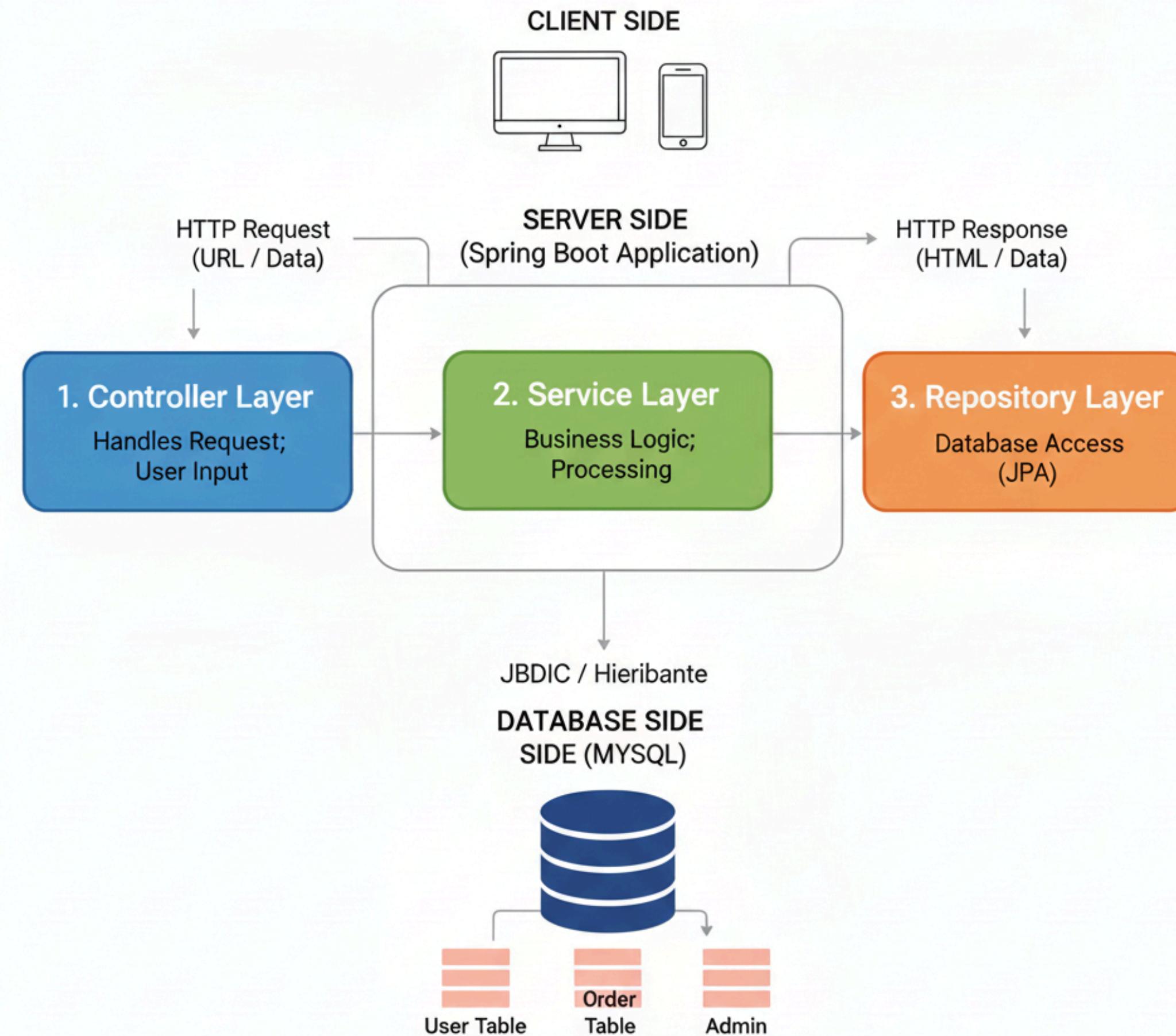
- Relies on Pen & Paper or Excel.
- High risk of data loss.
- Slow retrieval of customer history.
- Proposed Wellness-Kart System:
- Automated Database entry via Web Forms.
- Secure Login & Data Integrity.
- Instant data retrieval via Search & Sort functionality.

# **SYSTEM ANALYSIS (EXISTING VS. PROPOSED)**

## **EXISTING SYSTEM:**

- Relies on Pen & Paper or Excel.
- High risk of data loss.
- Slow retrieval of customer history.
- Proposed Wellness-Kart System:
- Automated Database entry via Web Forms.
- Secure Login & Data Integrity.
- Instant data retrieval via Search & Sort functionality.

# Wellness-Kart System: MVC Architecture



## **TECHNOLOGY STACK**

- Backend Framework: Java Spring Boot (Spring MVC, Spring Data JPA).
- Database: MySQL (Relational Data Storage).
- Frontend Engine: Thymeleaf (Server-side Java Template Engine), HTML5, CSS3.
- Development Tools: Maven (Dependency Management), Spring Tool Suite (IDE).

## MODULES OVERVIEW

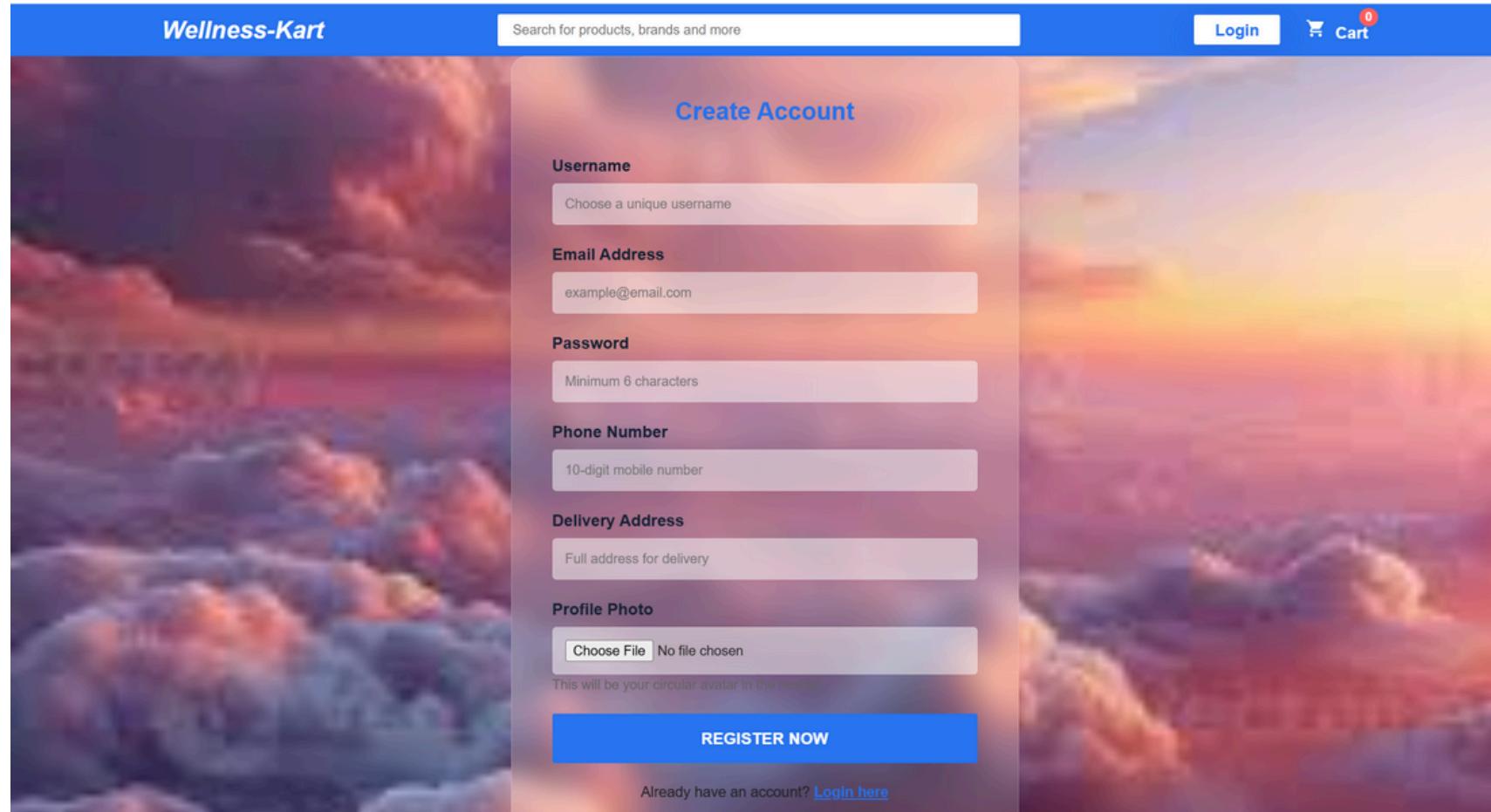
1. Admin Module: The control center for business logic, data viewing, and system oversight.
2. User Module: The customer-facing interface for registration, shopping, and profile management.
3. Order Module: The transaction processing engine handling product selection and cost calculation.
4. Security Module: Manages Login/Logout authentication and session stability.

# ADMIN MODULE FEATURES

The screenshot displays the 'WK Admin Panel' interface. At the top, there's a dark header bar with the text 'WK Admin Panel' on the left, 'Manage Users' in the center, and 'Logout' on the right. Below the header is a light gray section titled 'Business Overview' which contains three cards: 'TOTAL REVENUE ₹1330.0', 'TOTAL ORDERS 1', and 'TOTAL CUSTOMERS 1'. Underneath this is a section titled 'Recent Orders' featuring a table with columns: Order ID, Customer, Details, Amount, Date, and Status Action. A single row is shown: #WK-1, dev, 1500ml Vanilla(x4), ₹1330.0, 19/12/25 12:06, and a dropdown menu for status with options Placed, Processing (which is selected), Shipped, Delivered, and Cancelled. There's also an 'Update' button next to the dropdown.

- Secure Login: Credential verification against the secure admin database table.
- Dashboard: Real-time counters displaying "Total Orders," "Total Revenue," and "Active Users."
- User Management: View a comprehensive list of all registered students/users.

# USER MODULE FEATURES



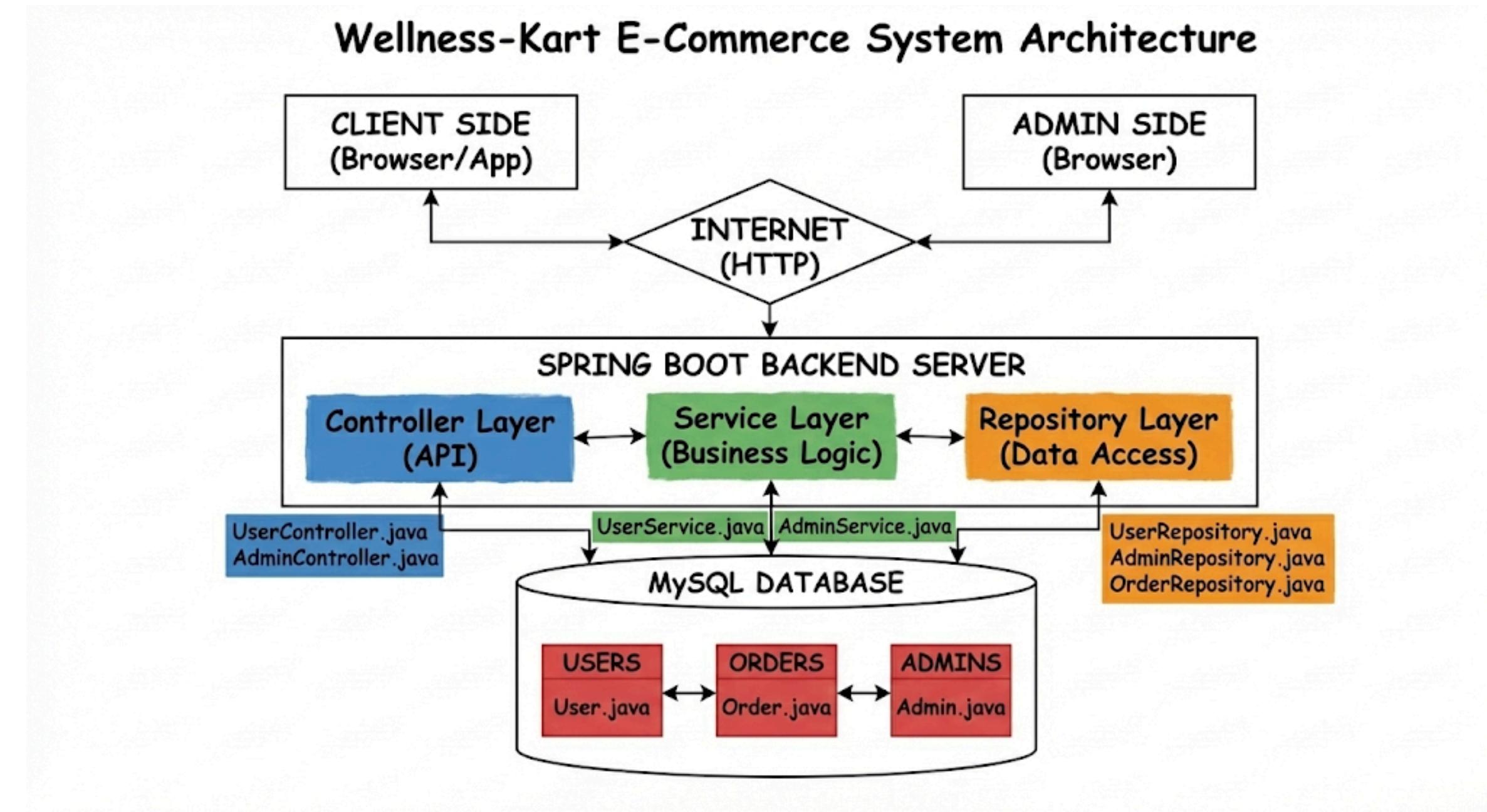
- Registration: A form to capture Name, Email, Phone, Address, and upload a Profile Picture.
- Profile Management: Users can view their details and uploaded profile image.
- Shopping Interface: A selection menu for choosing Bottles and Fragrances.

## **ORDER MODULE LOGIC**

- Selection: User selects Bottle Type + Fragrance.
- Processing: The system captures the Current Date, the logged-in User ID, and calculates the Total Amount.
- Status: Default status is set to "Placed."
- Payment: Default mode is set to "Payment on Handover."

# **DATABASE DESIGN (ER DIAGRAM)**

- ENTITIES: USER, ADMIN, ORDER.
- RELATIONSHIPS:
- USER ↔ ORDER: ONE-TO-MANY (1:N) - ONE USER CAN PLACE MULTIPLE ORDERS.
- ADMIN ↔ SYSTEM: MANAGES THE OVERALL DATA FLOW.



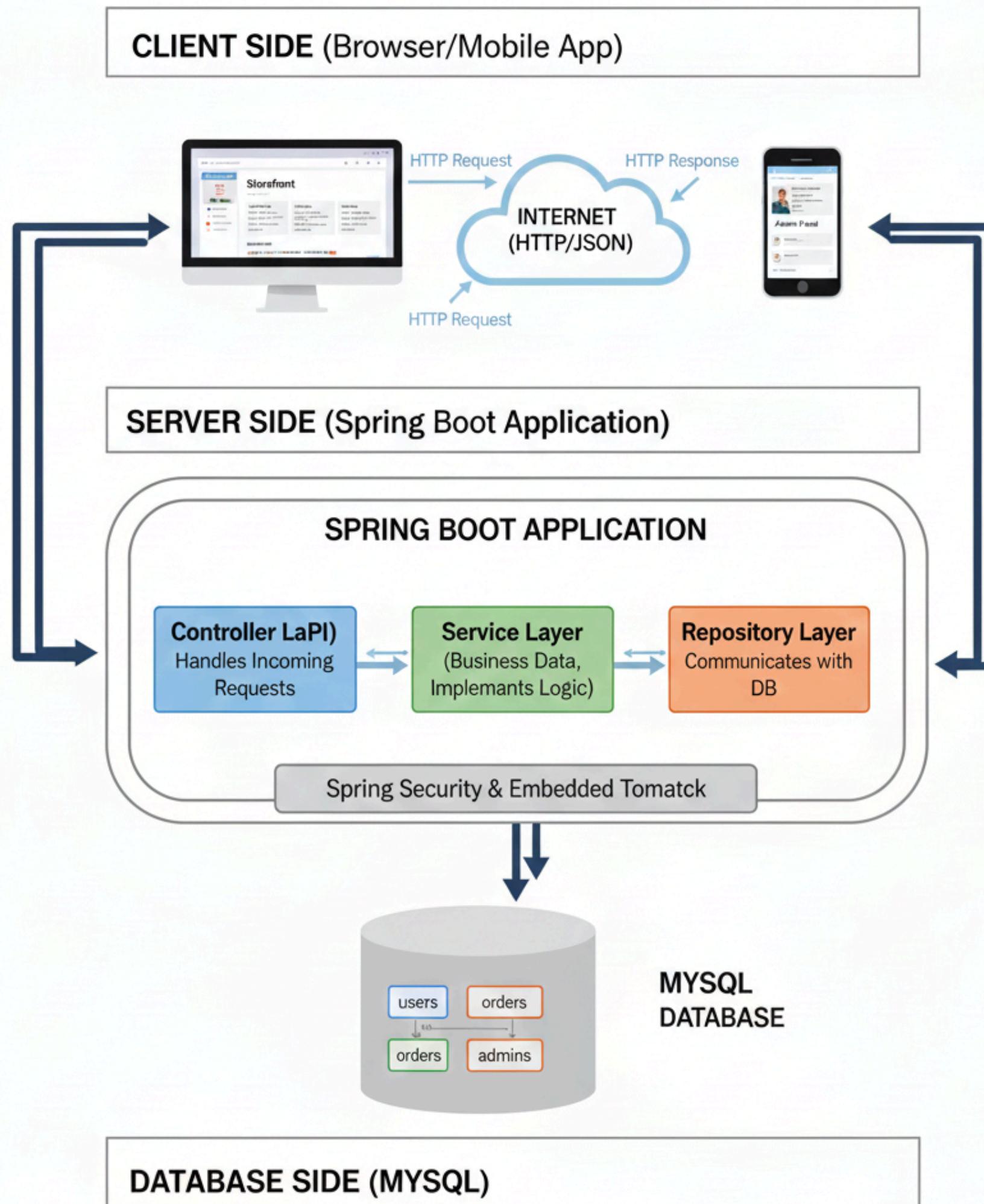
# DATABASE SCHEMA (TABLES)

- Table users: id (PK), username, password, email, phone, address, profile\_image.
  - Table orders: id (PK), user\_id (FK), bottle\_type, fragrance, total, status, date.
  - Table admins: id (PK), username, password, role.

# **CLIENT-SERVER ARCHITECTURE**

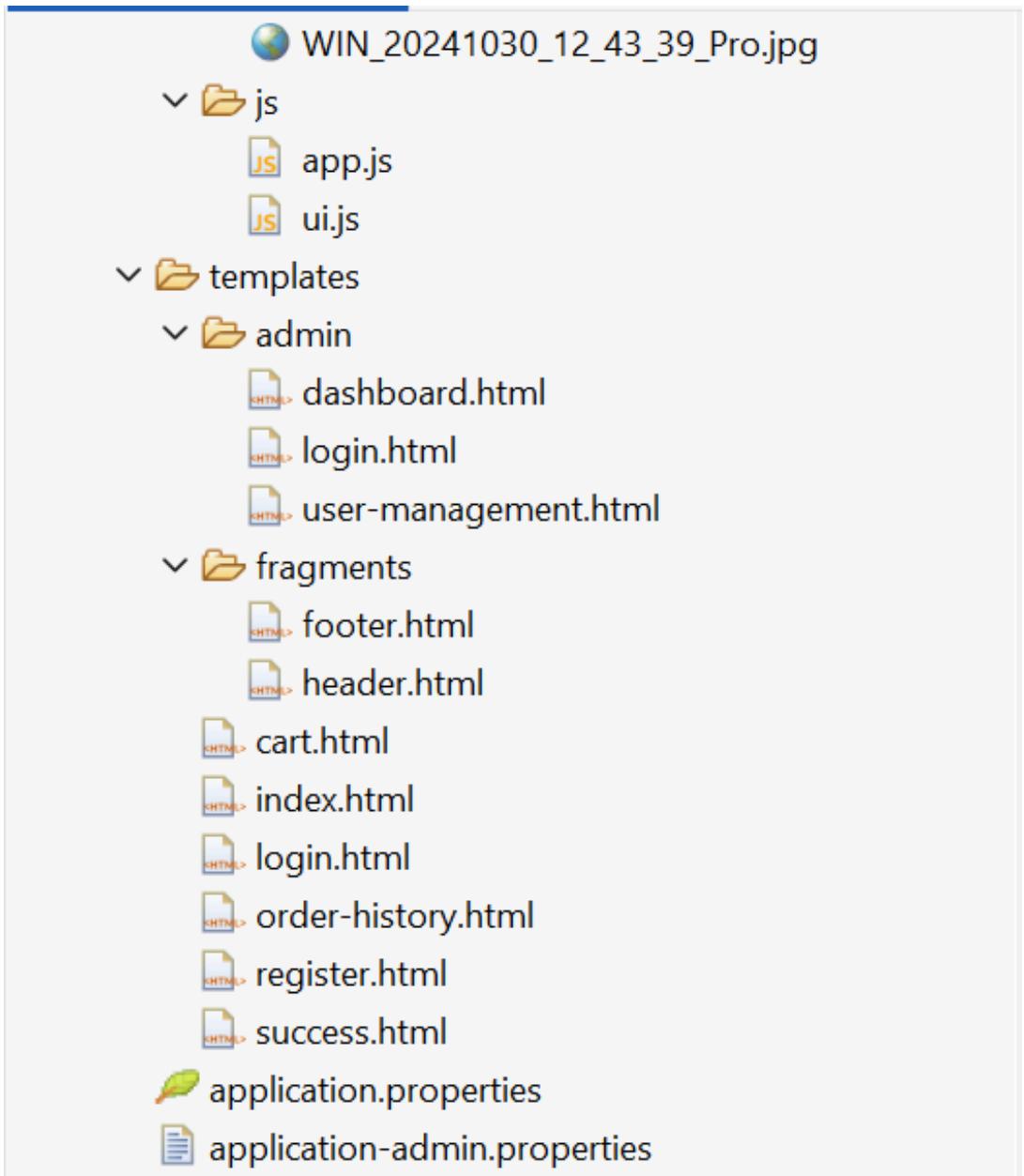
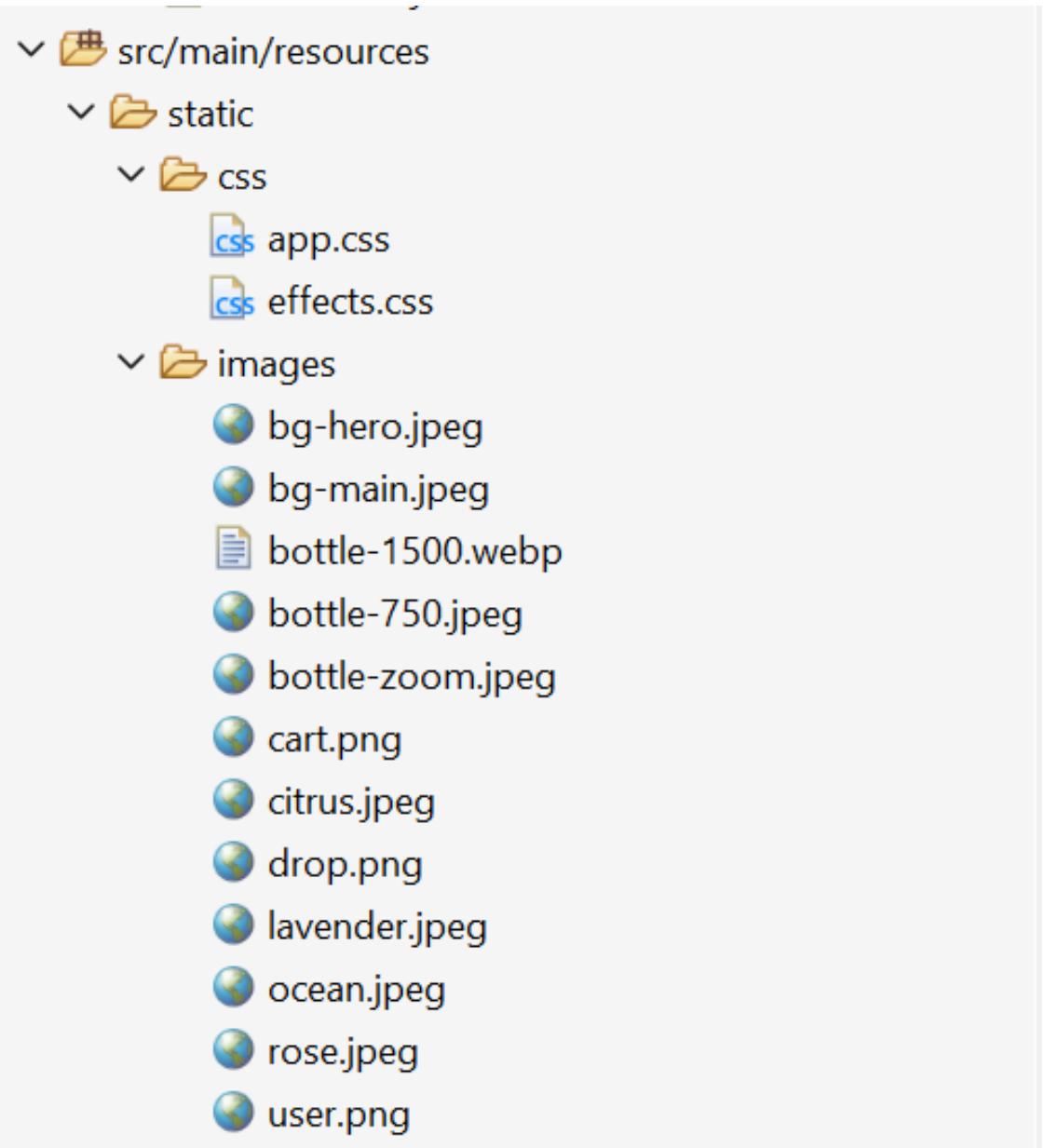
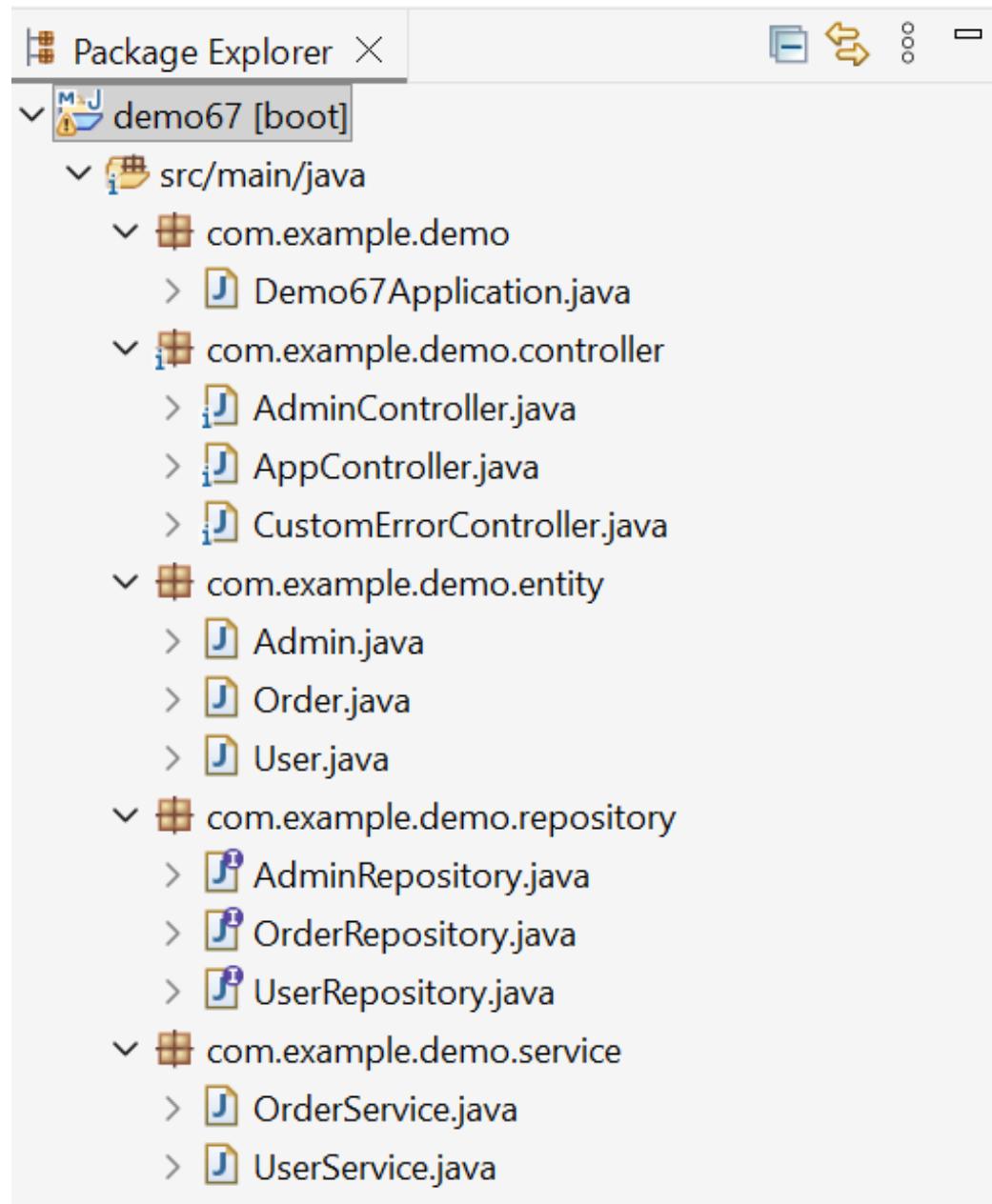
- Client: The Web Browser (Chrome/Edge) sending HTTP Requests.
- Server: Tomcat (Embedded in Spring Boot) receiving and processing requests.
- Data Flow: Client → Controller → Service → Repository → Database.

# Wellness-Kart System: Client-Server Architecture



# IMPLEMENTATION - PROJECT STRUCTURE

- Controller Package: Contains AppController.java and AdminController.java.
- Entity Package: Contains User.java and Order.java (POJO Classes).
- Repository Package: Contains UserRepository and OrderRepository (Interfaces).
- Templates: Contains login.html, register.html, index.html.



## **KEY ALGORITHMS & LOGIC**

- Login Logic: `findByUsername(user)` → Check if password matches → `Session.setAttribute`.
- Search Logic: Uses `stream().filter()` on the Order list to match the query string against `bottleType` or `fragrance`.

## **TESTING - UNIT TESTING**

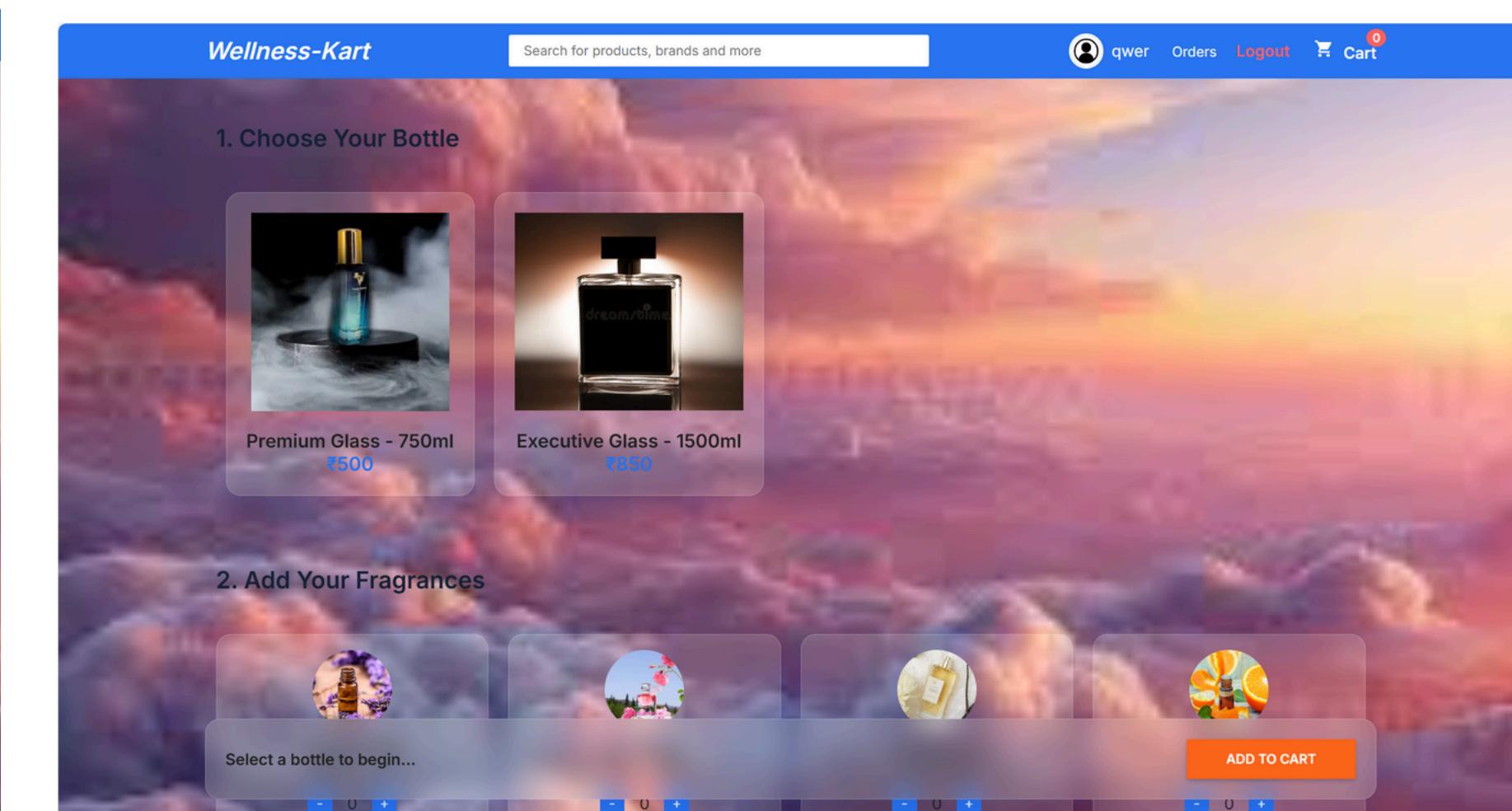
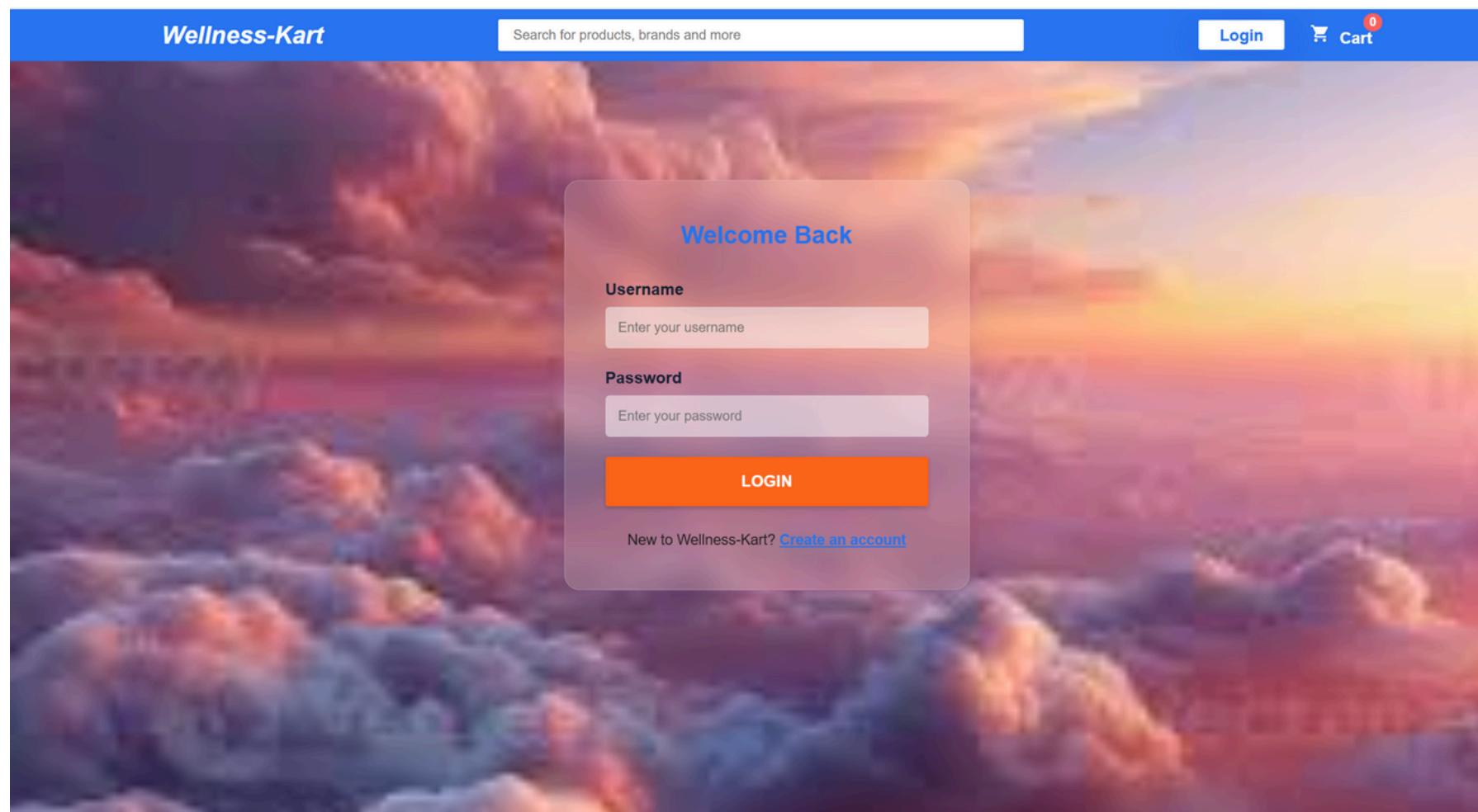
- Tools: JUnit 5.
- Tests Performed:
  - Test User Save (Repository Layer validation).
  - Test Order Retrieval (Service Layer logic).
  - Test Null Validations (Ensuring mandatory fields are present).

## **TESTING - API TESTING (POSTMAN)**

- GET Request: /api/users - Retrieves all user data.
- POST Request: /do\_register - Creates a new user entry.
- Status Codes: Tested for 200 OK (Success), 404 Not Found, and 500 Server Error.

## **OUTPUT - REGISTRATION & LOGIN**

- Registration: Demonstrates successful creation of a user with a profile picture.
- Login: Shows successful redirection to the Home Page upon entering valid credentials.
- Error Handling: Displays "Invalid Username/Password" message on failure.



# OUTPUT - PLACING AN ORDER

- Action: User selects items → Clicks "Place Order."
- Result: Order is saved to the DB → User is redirected to the "Success" page.

The screenshot shows the 'Your Orders' section of the Wellness-Kart website. At the top, there is a search bar and a navigation bar with links for 'dev', 'Orders', 'Logout', and a shopping cart icon showing '6'. Below the header, a large banner displays a sunset over water. A modal window titled 'Your Orders' is open, showing details for 'Order #WK-1' placed on '19 Dec 2025, 12:06'. The order summary includes: 'Bottle: 1500ml', 'Fragrances: Vanilla(x4)', and 'Amount: ₹1330.0'. To the right of the summary is a 'PLACED' button with the text 'Payment on Handover'. At the bottom of the page, there are sections for 'ABOUT', 'HELP', 'POLICY', and 'REGISTERED OFFICE ADDRESS', along with a copyright notice at the very bottom.

The screenshot shows the 'Order Placed Successfully!' confirmation page. The background features a sunset over water. A central message says 'Order Placed Successfully!' with a green checkmark icon. Below the message, it says 'Thank you for shopping with Wellness-Kart.' and 'Payment Method: Cash on Delivery' with the note 'Amount will be collected on handover'. At the bottom, there are buttons for 'VIEW MY ORDERS' and 'CONTINUE SHOPPING'. The footer contains sections for 'ABOUT', 'HELP', 'POLICY', and 'REGISTERED OFFICE ADDRESS', identical to the first screenshot.

# OUTPUT - ADMIN DASHBOARD

- Metrics: Visual cards showing accurate, real-time counts of data in the system.
- Data Grid: A tabular view of all registered users and their details.

WK Admin Panel

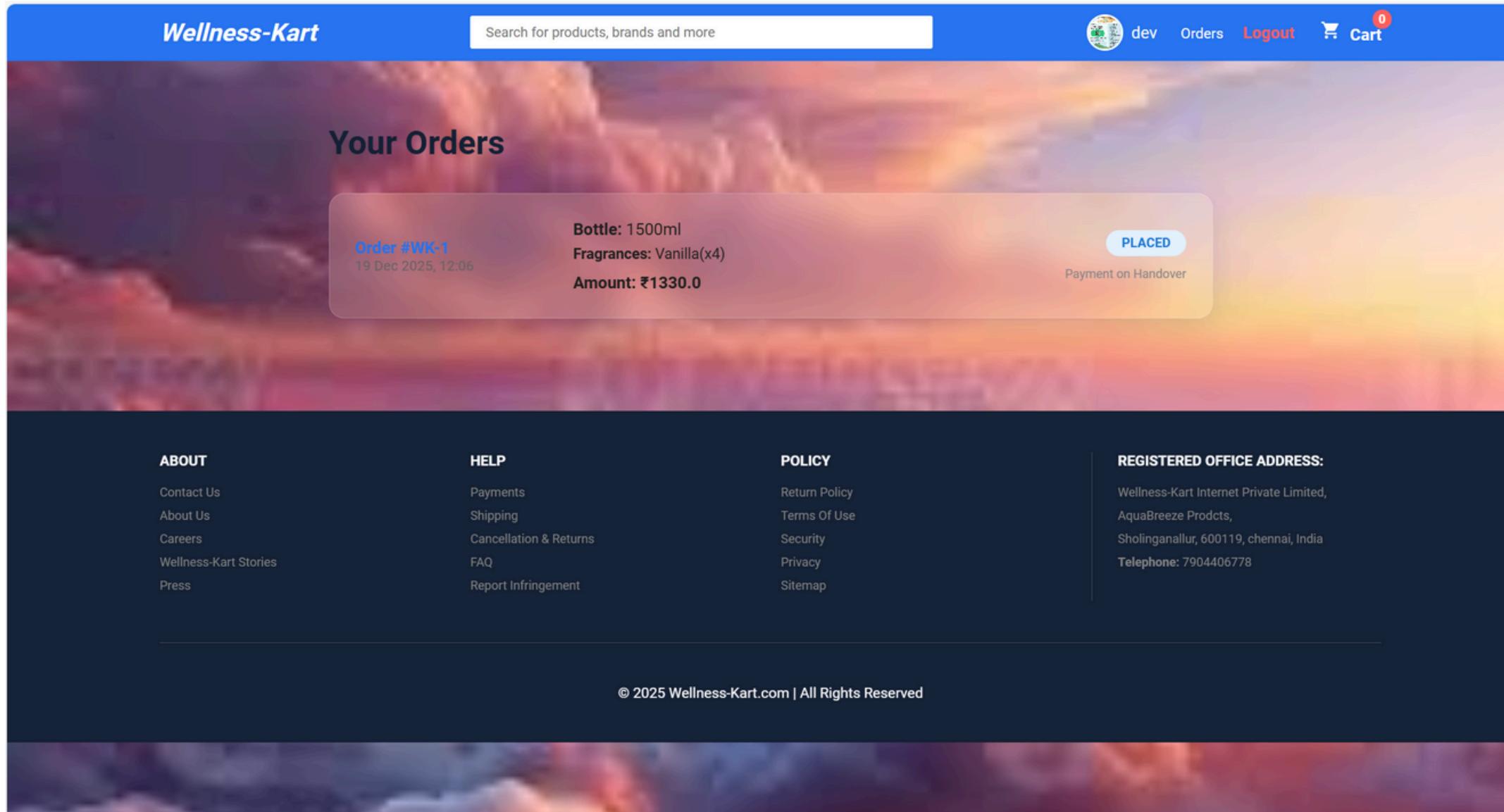
Customer Directory

Total Registered Users: 1

Profile	Username	Email	Phone	Address	Actions
	dev	dev@gmail.com	7904606778	Chennai/Tamil Nadu	<button>DELETE</button>

# OUTPUT - ORDER HISTORY & SEARCH

- Feature: User views their past orders in a modern glass-morphism card layout.
- Search: Demonstrates filtering orders by keywords like "Rose" or "Steel Bottle."



## **CHALLENGES FACED**

- Image Uploads: Configuring MultipartFile and ensuring the static/images folder refreshes correctly in the IDE.
- Session Management: Maintaining the "Logged In" state securely across different pages.
- Database Mapping: Correctly setting up the Foreign Key (@ManyToOne) relationship between Order and User entities.

## FUTURE ENHANCEMENTS

- Payment Gateway: Integration with APIs like Stripe or Razorpay for real-time payments.
- Email Alerts: Implementing an SMTP server to send automated email confirmations on order placement.
- Mobile App: Developing a React Native mobile frontend for the system.
- Admin Tools: Adding an interface to dynamically add/remove products from the catalog.

## CONCLUSION

- Wellness-Kart successfully digitizes the wellness product sales process, replacing manual inefficiency with automated precision.
- The system improves data accuracy, processing speed, and user experience.
- The modular Spring Boot architecture ensures the system is maintainable, secure, and scalable for future business growth.

## REFERENCES

- Spring Boot Official Documentation.
- Thymeleaf Standard Dialect Documentation.
- MySQL 8.0 Reference Manual.
- "Java Persistence with Hibernate" (Technical Book).

# **THANK YOU**

Wellness-Kart

Contact: devanandsanjay9543@gmail.com