A PROJECT ON

"Online Courier Management System"

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:

Sahil Ganbhoj - 250244220060

UNDER THE GUIDENCE OF:
Mr. Snehal Jadhav
Faculty Member
Sunbeam Institute of Information Technology, PUNE.

CERTIFICATE

This is to certify that the project work under the title 'Online Courier Management System' is done by Sahil Ganbhoj in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Mr. Snehal Jadhav
Project Guide

Mr. Yogesh Kolhe
Course Co-Coordinator

Date: 11-08-2025

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mr. Yogesh Kolhe (Course Coordinator, SIIT ,Pune) .

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Sahil Ganbhoj

0225 PG-DAC

SIIT Pune

<u>Table of Contents</u>

1. Introduction to Project

The SwiftCourier – Online Courier Management Service is a fully integrated web application designed to modernize and streamline courier operations for businesses of any scale. It serves as a digital alternative to traditional, paper-based courier systems by providing a centralized, secure, and real-time platform that connects administrators, employees, and customers.

In the logistics industry, time, accuracy, and transparency are critical. Manual systems often lead to delays, misplaced shipments, inaccurate status updates, and customer dissatisfaction. SwiftCourier addresses these issues through automation, cloud-enabled accessibility, and role-specific features that ensure smooth and efficient operation from booking to delivery.

The system is built using a three-tier architecture:

1. **Presentation Layer –** Developed using React.js, this layer ensures a dynamic, responsive, and user-friendly interface accessible via web browsers.

2. **Business Logic Layer** – Powered by Spring Boot, it manages all application logic, validations, and API services.

3. **Data Layer** – Managed through MySQL, ensuring structured, secure, and efficient storage of shipment, user, and transaction data.

Key features include:

● Secure user registration and authentication for Admin, Employee, and Customer roles.

● Order placement with automated tracking number generation.

- Real-time shipment tracking with multiple status levels (Order Placed, Picked Up, Shipped, Out for Delivery, Delivered, Cancelled).

- Branch and employee management for scalable operations.

- Integrated payment gateway for seamless transactions.

By implementing SwiftCourier, courier companies can expect reduced operational costs, increased accuracy, and enhanced customer satisfaction through transparency and faster delivery updates.

2. Requirements

2.1 Functional Requirements

2.1.1 User Account Management

- The system supports role-based authentication, allowing Admin, Employee, and Customer accounts.

- Customers can register to create shipment orders, track packages, and view history.

- Employees can update shipment statuses, process orders, and manage deliveries.

- Admins have full system control, including managing branches, employees, and all shipments.

2.1.2 Registration and Profile Creation

- The registration form collects essential details: Name, Email, Contact Number, Address, and Password.

- Profile pages allow updating personal details without affecting existing shipment records.

- Password reset functionality ensures account security.

2.1.3 Order Creation and Tracking

- Customers can create orders by entering pickup and delivery addresses, package weight, and delivery type (standard/express).

- System auto-generates a unique tracking ID for each order.

- Tracking is available to all users with the tracking number, even without logging in.

2.1.4 Payment Processing

- Integration with secure payment gateways enables customers to pay online via debit/credit card, UPI, or net banking.

- Orders remain in "Pending Payment" state until payment confirmation is received.

2.1.5 Branch and Employee Management

- Admins can add, edit, or remove branches.

- Employee details can be updated, including designation, contact info, and assigned branch.

- Branch data includes Branch name, location, contact number and Email.

2.1.6 Viewing Order History

- Customers can view all previous orders with details such as delivery status, cost, and date.

- Admins and employees can filter order history by branch, date range, or status.

2.2 Non-Functional Requirements

2.2.1 Interface

- Clean, intuitive design with responsive elements for both desktop and mobile use.

- Consistent branding, color schemes, and typography for professional appearance.

2.2.2 Other Requirements

- **Hardware:**
  Client – Dual-core 2.0 GHz CPU, 4 GB RAM, 2 GB free disk space.
  Server – Quad-core 2.4 GHz CPU, 8 GB RAM, 50 GB disk space.

- **Software:**
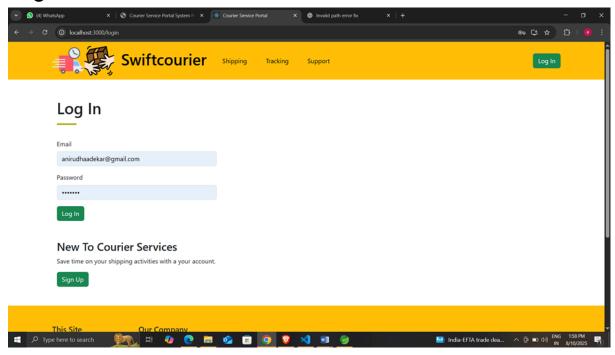 Frontend – React.js, Node.js, npm
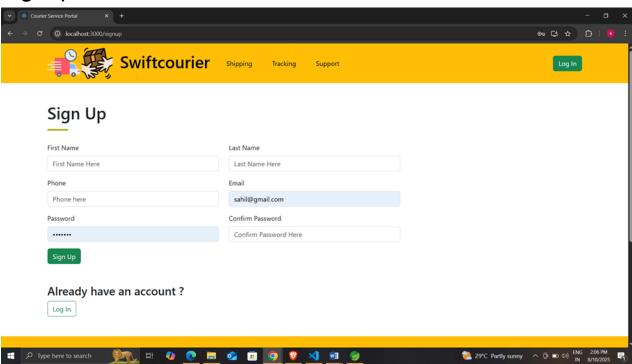 Backend – Java 17+, Spring Boot
  Database – MySQL 8.0+
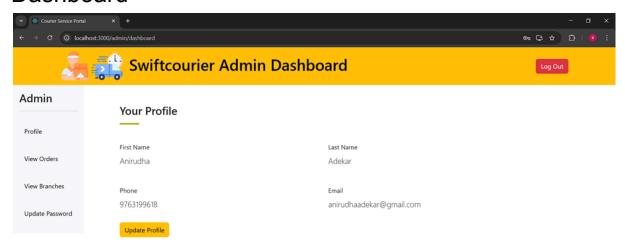 Browsers – Chrome, Edge, Firefox, Safari
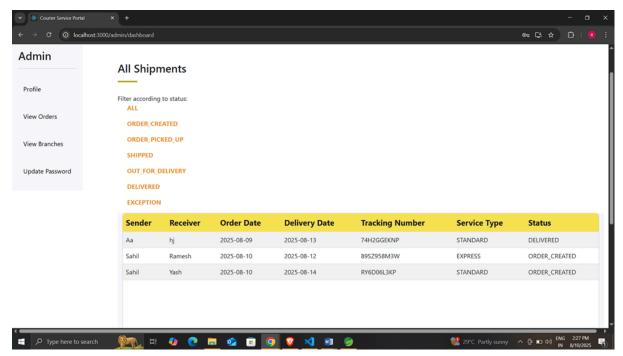
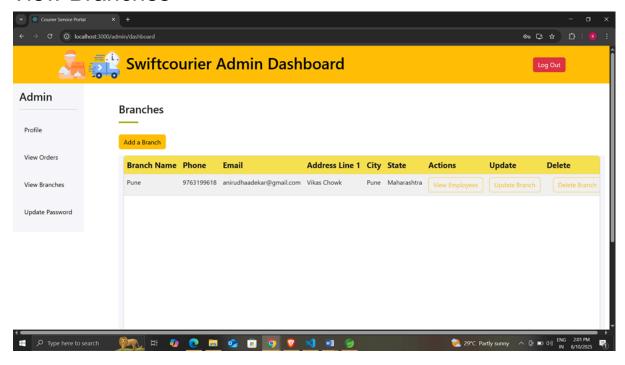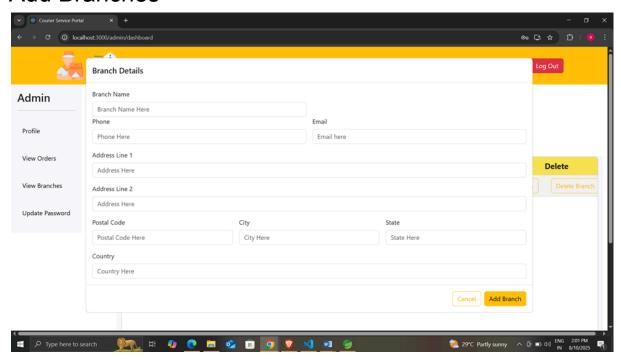# 3. Design

Login:

Signup:

Admin

Dashboard

View Orders
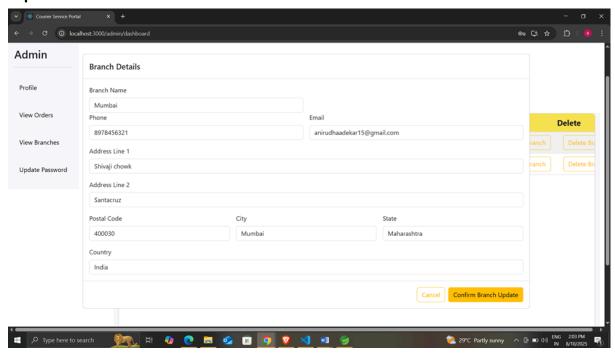
View Branches
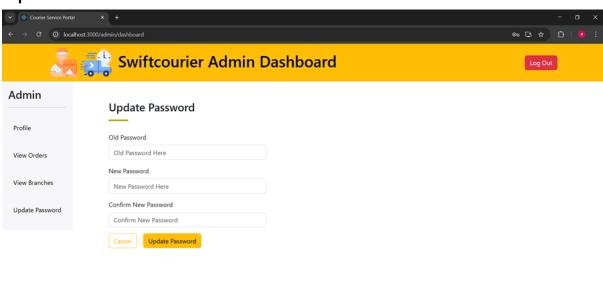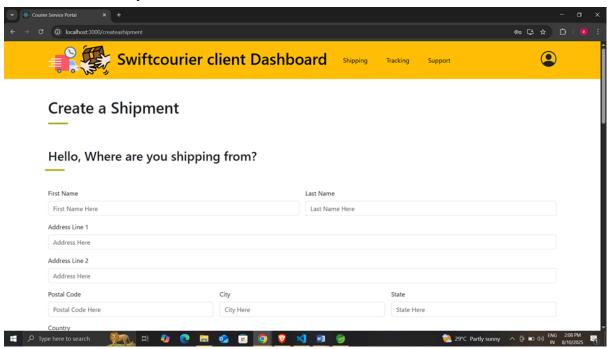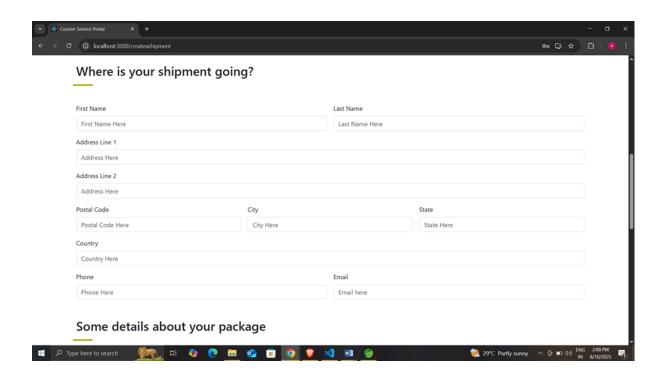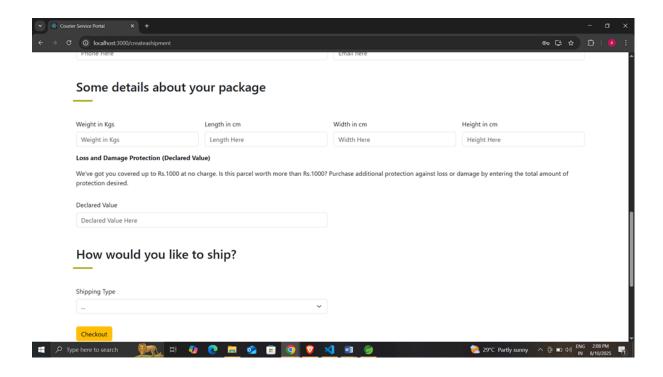

Add Branches

Update Branches

Update Password

Client

Create a Shipment

Some details about your package

Weight in Kgs

Length in cm

Width in cm

Height in cm

Loss and Damage Protection (Declared Value)

We've got you covered up to Rs.1000 at no charge. Is this parcel worth more than Rs.1000? Purchase additional protection against loss or damage by entering the total amount of protection desired.

Declared Value

How would you like to ship?

Shipping Type

Checkout

Your Order Was Placed Successfully

SENDER
Sahil Ganbhoj
The Crown Greens
Hinjwadi
Pune
Maharashtra
India
411041
7894561230
sahil@gmail.com

RECEIVER
Ramesh Dev
Radha Chowk
Market road
Nagpur
Maharashtra
India
450319
9874563210
ramesh@gmail.com

TrackingNumber : 89SZ958M3W

Order Date : 2025-08-10

Delivery Date : 2025-08-12

Shipping Type : EXPRESS

Declared Value : 5000

Status : ORDER_CREATED

Please note down the tracking number. You can use it to track your order and check its status.

View Shipments

View Shipping History

Calculate Shipping Cost

Find a Location

Contact Us

Employee

Employee Dashboard

View Orders
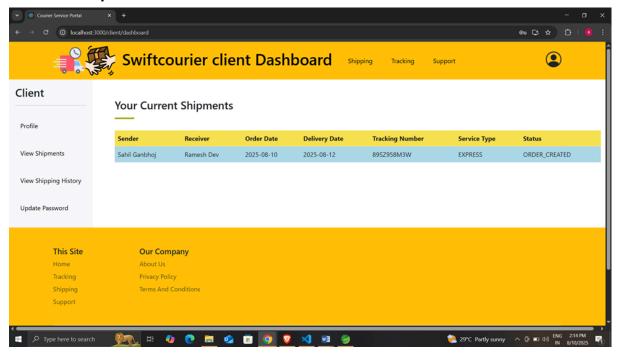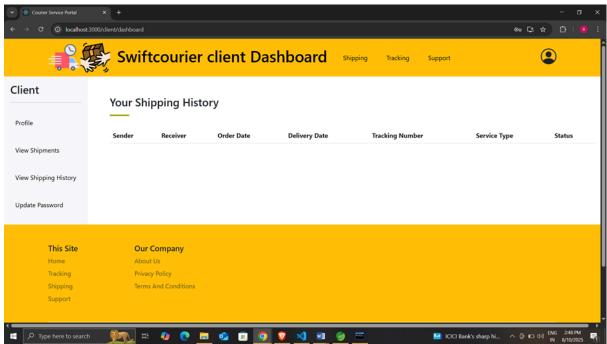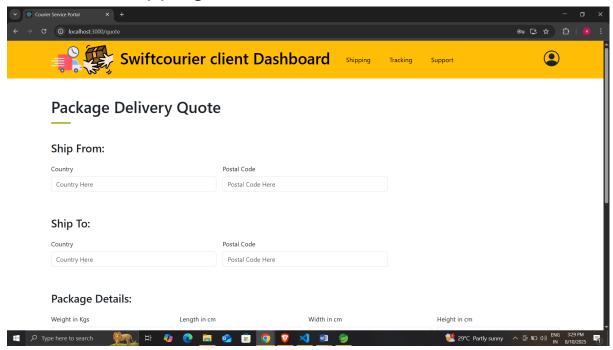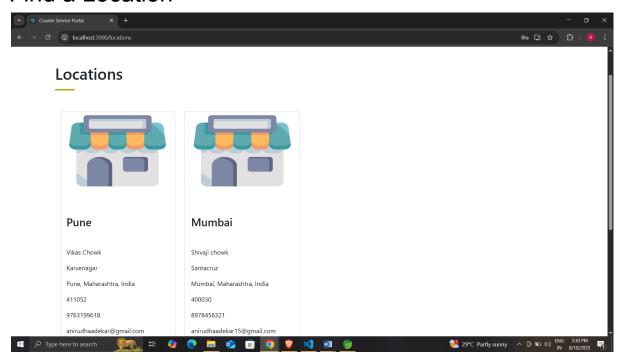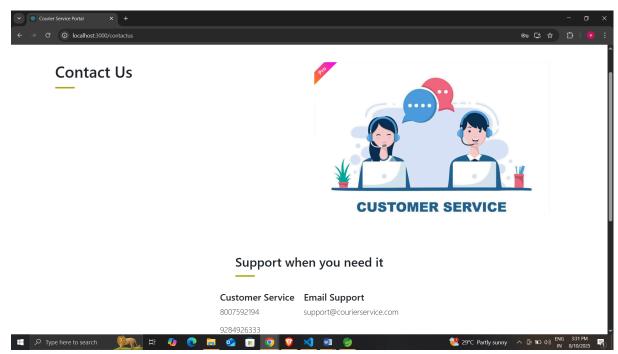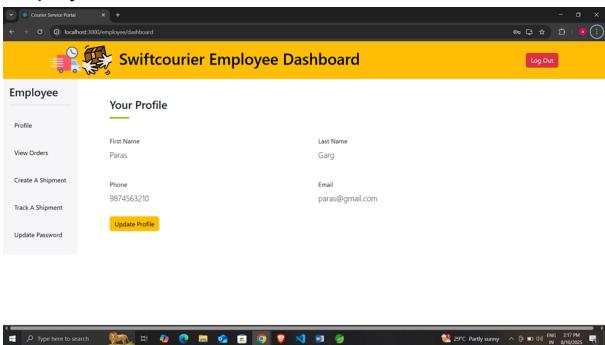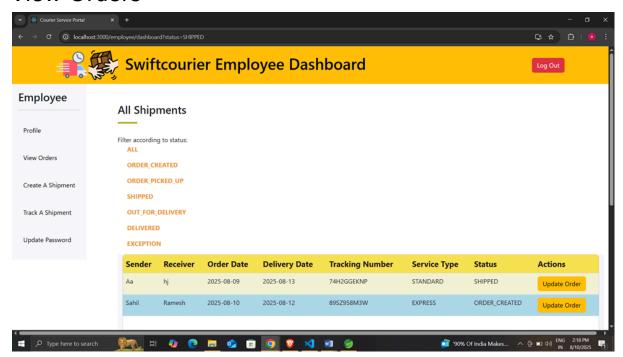
Create a Shipment

Track a Shipment

4. Diagram

Database Design

Addresses table							
address_id (PK)	address	address_2	city	country	postal_code	state	user_id (FK)

Branches table				
branch_id (PK)	branch_name	email	phone	address_id (FK)

Complaints table					
complaint_id (PK)	complaint_date	complaint_status	description	remark	order_id (FK)

Customers table						
customer_id (PK)	customer_type	email	first_name	last_name	phone	address_id (FK)

Employees table			
hire_date	salary	user_id (PK)	branch_id (FK)

Orders table											
order_id (PK)	declared_value	delivery_date	order_date	service_type	status	tracking_number	parcel_id_parcel_id (FK)	payment_id (FK)	receiver_id (FK)	sender_id (FK)	user_id (FK)

Parcels table

parcel_id (PK)	height	length	weight	width

Payments table

payment_id (PK)	amount	payment_date	payment_method

Users table

user_id (PK)	email	first_name	last_name	password	phone	role

ER Diagram

5. Tools and Technologies Used

Frontend – React.js

- Enables modular, reusable components.

- Virtual DOM ensures fast UI rendering.

Backend – Spring Boot (Java)

- Simplifies configuration and deployment.

- Offers built-in support for security, REST APIs, and database connectivity.

Database – MySQL

- ACID-compliant, ensuring reliable transaction processing.

- Supports indexing for fast searches.

Version Control – Git & GitHub

- Tracks code changes and supports collaborative development.

Payment Gateway Integration

- Secures financial transactions with encryption and tokenization.

6. Advantages

The SwiftCourier platform offers a wide range of benefits for courier companies, employees, and customers alike. By integrating modern web technologies and an efficient workflow, the system ensures that operations are not only smooth but also scalable and secure. The following advantages outline the core strengths of the system:

1. **Centralized Management**
   All courier operations — from order booking and payment processing to branch and employee management — are consolidated into a single, unified portal. This eliminates the need for multiple disconnected systems, making it easier for administrators to monitor, control, and update business processes in real time.

2. **High Scalability**
   SwiftCourier is designed with scalability in mind. Whether operating within a single city or across multiple regions, the system can easily handle increased data volumes, additional branches, and more concurrent users without major reengineering. The modular architecture supports smooth expansion without service downtime.

3. **Customer Transparency and Trust**
   Customers can track their shipments in real time, from the moment an order is placed until it is delivered. Status updates such as *Order Placed*, *Order Picked Up*, *Shipped*, *Out for Delivery*, and *Delivered* are easily accessible through the tracking feature. This transparency builds customer trust and reduces the need for repeated customer service inquiries.

4. **Secure Transactions**
   With sensitive information like personal details and payment data involved, security is a top priority. All financial transactions are processed through encrypted payment gateways, and user

authentication is handled through **JWT (JSON Web Tokens)** to prevent unauthorized access. This ensures the confidentiality and integrity of user data.

5. **Operational Cost Savings**

   The shift from paper-based operations to digital processing leads to significant cost reductions. Businesses save on printing, manual labor, and physical storage space. Additionally, automated workflows reduce delays and operational inefficiencies, which in turn lowers overhead costs.

7. Coding Standards Implemented

To maintain code quality, ensure system reliability, and facilitate future scalability, SwiftCourier follows a set of well-defined coding standards and best practices. These standards cover naming conventions, authentication practices, and security measures.

6.1 Naming Conventions

- **Classes – PascalCase**
   All class names start with a capital letter, and each subsequent word is also capitalized. For example:
   `OrderController, BranchService, ShipmentTracker.`
   This convention improves readability and clearly distinguishes classes from variables or methods.

- **Methods – camelCase**
 Method names start with a lowercase letter, and subsequent words are capitalized. Method names are always verbs or verb phrases to indicate their function.

Example:
`createShipmentOrder(),        updateOrderStatus(), calculateShippingCost().`

- **Variables – camelCase**
Variables use lowercase for the first word, and uppercase for the first letter of each subsequent word. Descriptive names are preferred over abbreviations for clarity.

Example:
`trackingId, branchLocation, orderStatus.`

6.2 Security Best Practices

Security is integrated into every layer of the SwiftCourier architecture to ensure data protection and prevent unauthorized access:

● **JWT (JSON Web Token) Authentication**
All user authentication is handled using JWTs, which ensures secure, stateless authentication between the client and server. Tokens are signed and verified to prevent tampering and can be set with expiration times to enhance security.

● **Password Encryption**
Passwords are hashed using strong cryptographic algorithms before being stored in the database. This ensures that even if database access is compromised, plaintext passwords are never exposed.

● **Parameterized SQL Queries**
All database queries are parameterized to protect against SQL injection attacks. This prevents malicious users from executing arbitrary queries through form inputs.

● **Role-Based Access Control (RBAC)**
Access to system features is controlled based on the role of the user (Admin, Employee, Customer). This prevents unauthorized users from performing restricted operations such as deleting branches or modifying shipment data.

8. Conclusion

The SwiftCourier – Online Courier Management Service successfully delivers a modern, reliable, and scalable solution for managing courier operations in an increasingly fast-paced and competitive logistics environment. By integrating advanced web technologies such as React.js for the frontend, Spring Boot for the backend, and MySQL for database management, the system offers high performance, real-time responsiveness, and strong data integrity.

Through its role-based access architecture, SwiftCourier effectively addresses the different needs of Administrators, Employees, and Customers. Administrators benefit from centralized control over branches, employees, and orders, while Employees gain efficient tools for shipment handling and status updates. Customers experience convenience and transparency with features like instant order booking, automated tracking IDs, and real-time delivery status updates.

Security is a cornerstone of the platform. With JWT (JSON Web Token) authentication, encrypted password storage, and secure payment gateway integration, the system ensures that sensitive user data and financial transactions remain protected against unauthorized access or tampering. Additionally, the use of parameterized queries safeguards against common web vulnerabilities such as SQL injection attacks.

From a business perspective, SwiftCourier brings significant operational advantages:

- Reduction in manual workloads and human error through automation.

- Enhanced customer trust via transparent tracking and reliable delivery timelines.

- Cost savings by minimizing paper usage, streamlining workflows, and optimizing resource allocation.

- Scalability to expand from local operations to multi-city or even nationwide coverage without substantial redevelopment.

Beyond operational efficiency, the system also serves as a foundation for data-driven decision-making. The data stored in the system can be used for analytics, helping management identify performance bottlenecks, optimize delivery routes, and improve service quality over time.

In conclusion, SwiftCourier is not just a courier management application — it is a comprehensive, future-ready logistics solution capable of adapting to evolving customer expectations and market demands. By combining efficiency, security, scalability, and user-centric design, it positions itself as an essential tool for any courier service provider aiming to remain competitive in the digital era.

9. References

1. https://spring.io/projects/spring-boot

2. https://spring.io/projects/spring-data-jpa

3. https://restfulapi.net/

4. https://www.mysql.com/

5. https://spring.io/projects/spring-web

6. https://reactjs.org/