DIGITAL ASSIGNMENT - 5

Software Engineering Lab L41+L42

22BCE0544

APRAJITA NANDKEULIAR

Project Title

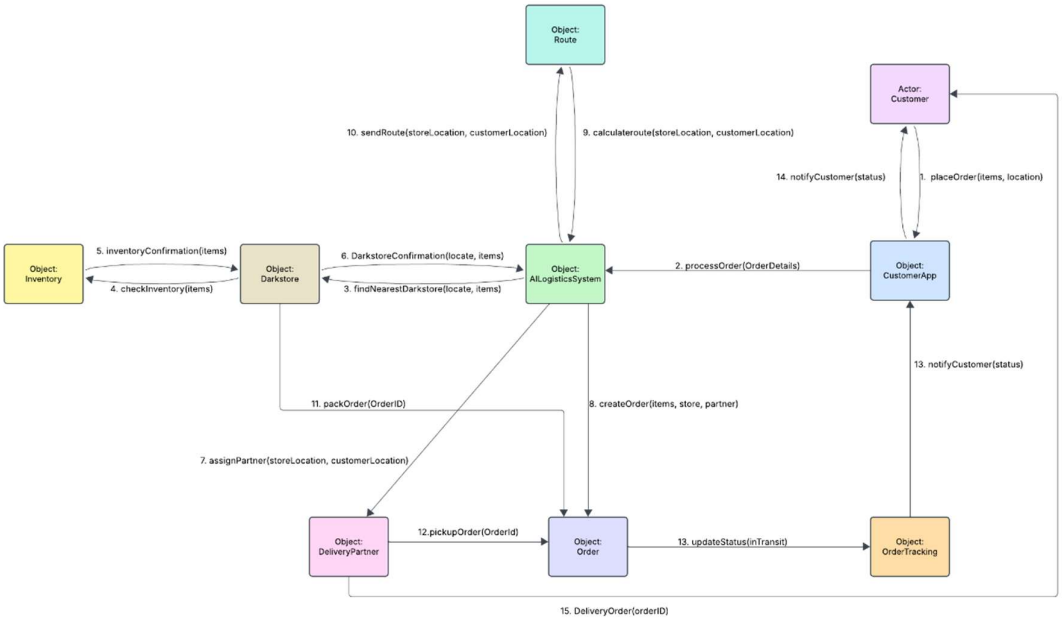
SwiftDrop: Ultra-fast Delivery Logistics System

Explanation of the Collaboration Diagram

This collaboration diagram shows the sequence of messages exchanged between objects in the SwiftDrop: ultra-fast Delivery Logistics System. The interaction represents the complete flow from when a customer places an order until it's delivered:

1. **Customer to CustomerApp**: The process begins with the customer (actor) sending a message to place an order through the CustomerApp, providing items and their location.
2. **CustomerApp to AILogisticsSystem**: The app forwards the order details to the AI Logistics System for processing.
3. **AILogisticsSystem to DarkStore**: The system locates the nearest dark store that can fulfill the order based on customer location and requested items.
4. **DarkStore to Inventory**: The dark store checks its inventory to confirm item availability.
5. **AILogisticsSystem to DeliveryPartner**: The system assigns an available delivery partner based on proximity to the dark store and customer location.
6. **DeliveryPartner to Route**: The delivery partner uses the route optimization to determine the fastest path from store to customer.
7. **AILogisticsSystem to Order**: The system creates a new order with the selected items, dark store, and delivery partner.
8. **DarkStore to Order**: The dark store packs the order and updates its status.
9. **Order to OrderTracking**: The order status is updated to "packed" in the tracking system.
10. **OrderTracking to CustomerApp**: The customer is notified that their order has been packed.
11. **DeliveryPartner to Order**: The delivery partner picks up the order from the dark store.
12. **Order to OrderTracking**: The order status is updated to "in transit."
13. **OrderTracking to CustomerApp**: The customer is notified that their order is on the way.
14. **DeliveryPartner to Order**: The delivery partner delivers the order to the customer.

15. **Order to OrderTracking**: The order status is updated to "delivered."
16. **OrderTracking to CustomerApp**: The customer is notified that their order has been delivered.



Explanation of the Component Diagram

In UML, the interfaces are typically represented with "lollipop" (provided interface) and "socket" (required interface) notations.

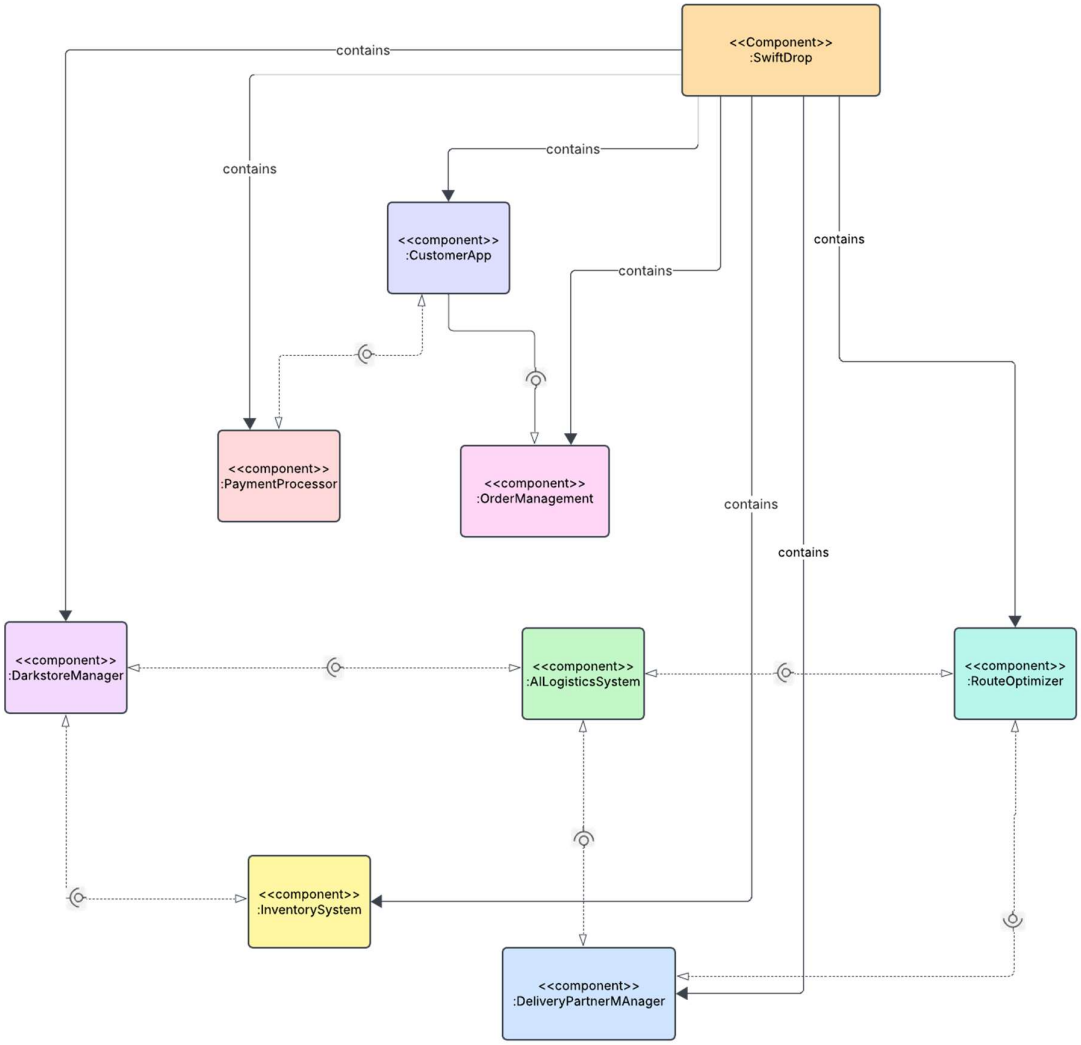
Key Elements:

1. **Components**: Each box represents a component with the <<Component>> stereotype, following the format in your reference image.
2. **Containment**: The SwiftDrop: ultra-fast Delivery Logistics System is the container component that contains all other components, shown with composition relationships (solid diamond).
3. **Interface Relationships**: The lines with <<interface>> stereotype represent the provided-required interface relationships between components, indicating that one component provides services that another component requires.

Component Interactions:

- **CustomerApp** has interface relationships with:
    - OrderManagement: For order placement and tracking
    - PaymentProcessor: For handling payment transactions

- **OrderManagement** has interface relationships with:
    - AILogisticsSystem: For processing order logistics
    - InventorySystem: For checking item availability
- **AILogisticsSystem** has interface relationships with:
    - DarkStoreManager: For store selection and coordination
    - DeliveryPartnerManager: For delivery partner assignment
- **DarkStoreManager** has an interface relationship with:
    - InventorySystem: For inventory management and updates
- **DeliveryPartnerManager** has an interface relationship with:
    - RouteOptimizer: For calculating optimal delivery routes



Explanation of the Deployment Diagram

This deployment diagram shows the physical architecture of the SwiftDrop: ultra-fast Delivery Logistics System. It illustrates how the system components are deployed across various hardware devices and how they communicate.

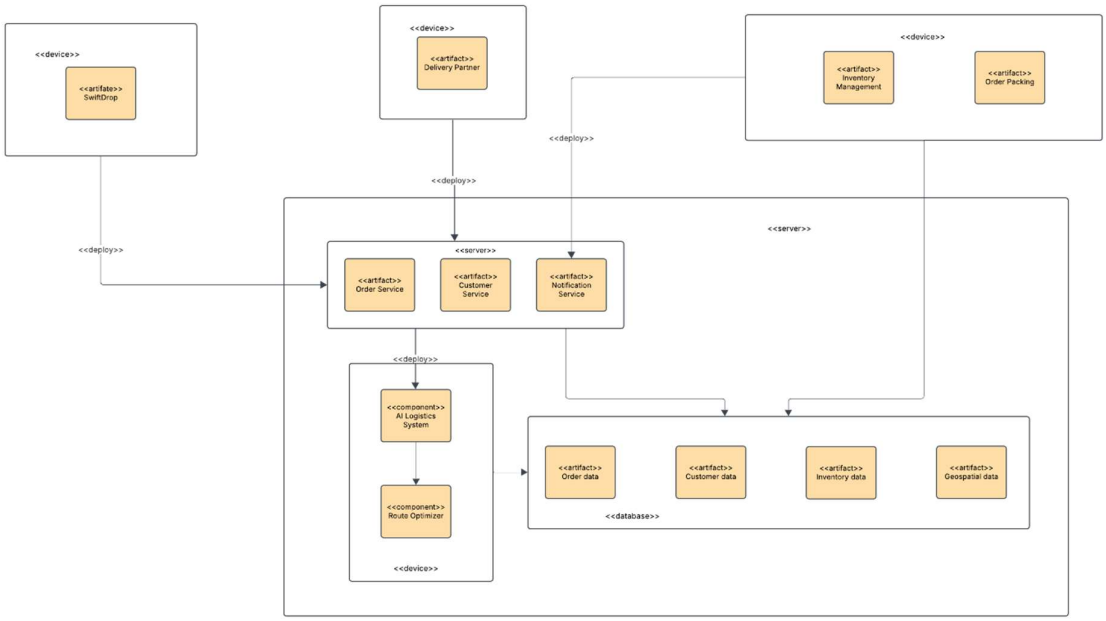
Key Elements in the Diagram:

1. **Devices (Nodes)**:
    - **Cloud Server**: The main server infrastructure hosting application servers, AI processing servers, and databases
    - **Customer Smartphone**: Customer's mobile device running the SwiftDrop app
    - **Delivery Partner Smartphone**: Mobile device used by delivery personnel
    - **Dark Store Terminal**: In-store system for inventory management and order packing
2. **Components and Artifacts**:
    - **Components** (shown with <<component>> stereotype): Executable software modules
        - Order Service, Customer Service, Notification Service
        - AI Logistics System, Route Optimizer
        - Inventory Management, Order Packing System
    - **Artifacts** (shown with <<artifact>> stereotype): Physical pieces of information
        - Zepto Mobile App
        - Delivery Partner App
        - Database artifacts (Order Data, Customer Data, Inventory Data, GeoSpatial Data)
3. **Relationships**:
    - **Communication links** (shown as lines with protocol labels):
        - HTTPS connections between mobile devices and servers
        - Internal connections between components and databases
    - **Deployment** relationships between components:

- Application Servers deploy to AI Logistics System
- AI Logistics System deploys to Route Optimizer

Key Aspects of the System Architecture:

1. **Distributed Computing**: The system is distributed across cloud servers and edge devices (smartphones and dark store terminals).
2. **Cloud-Based Processing**: The core intelligence of the system (AI Logistics System and Route Optimizer) runs on dedicated servers in the cloud.
3. **Database Integration**: A centralized database stores all critical data including orders, customers, inventory, and geospatial information.
4. **Mobile Access**: Both customers and delivery partners access the system through mobile applications that communicate with the backend servers via HTTPS.
5. **Dark Store Integration**: Each dark store has its own terminal with inventory management and order packing systems that connect to the central cloud infrastructure.



2. Design and demonstration of test cases. Functional Testing and Non- Functional Testing.

Functional Test Cases

1. Customer Order Placement

Test Case	Input	Expected Output	Result
Valid Order Placement	User selects valid items (Milk, Bread, Eggs), provides delivery address, payment info	Order successfully created, confirmation displayed	Pass/Fail
Order with Out-of-Stock Item	User selects unavailable item (Sold out ice cream)	Real-time notification that item is unavailable, suggestion for alternatives	Pass/Fail
Order with Invalid Address	User enters incomplete/invalid address	Error message prompting correct address format	Pass/Fail
Order Cancellation	User cancels order within 1 minute of placement	Order successfully cancelled, confirmation message displayed	Pass/Fail
Add/Remove Items from Cart	User adds 3 items, removes 1 item from cart	Cart accurately reflects changes, price updates automatically	Pass/Fail

2. Dark Store Identification

Test Case	Input	Expected Output	Result
Nearest Dark Store Selection	Customer at location (12.9716° N, 77.5946° E) places order	System identifies closest dark store with items in stock	Pass/Fail
Multiple Dark Stores Available	Order with items available in 3 nearby dark stores	System selects optimal dark store based on proximity and inventory	Pass/Fail
Item Split Across Dark Stores	Order with items not all available in single dark store	System either splits order or chooses store with most items	Pass/Fail
GPS Location Unavailable	Customer denies location permission	System prompts user to enter address manually	Pass/Fail
Dark Store Offline	Nearest dark store is temporarily offline	System reroutes to next nearest available dark store	Pass/Fail

3. Delivery Partner Assignment

Test Case	Input	Expected Output	Result
Available Partner Assignment	Order ready at dark store, 3 partners available nearby	Nearest available partner assigned, notification sent	Pass/Fail
No Partners Available	Order ready, no delivery partners within 2km	Graceful waiting or customer notification about slight delay	Pass/Fail
Partner Rejection	Partner rejects delivery assignment	System quickly reassigns to next available partner	Pass/Fail
Partner Status Change	Partner changes status from "available" to "busy"	Partner not included in available pool for new assignments	Pass/Fail
Multiple Orders Assignment	Multiple orders going to same area	Efficient batching of deliveries to same partner when feasible	Pass/Fail

4. Route Optimization

Test Case	Input	Expected Output	Result
Optimal Route Generation	Delivery from Dark Store A to Customer at 2.5km distance	System generates fastest route accounting for current traffic	Pass/Fail
Traffic Congestion Rerouting	Heavy traffic on primary route	System calculates and offers alternative faster route	Pass/Fail
Weather-Affected Routing	Rainy conditions affecting certain roads	Routes adjusted to account for weather conditions	Pass/Fail
Multiple Delivery Optimization	Partner with 3 deliveries in same area	Efficient sequence of deliveries minimizing total distance/time	Pass/Fail
Turn-by-Turn Navigation	Partner following route to customer	Clear, accurate turn-by-turn directions provided	Pass/Fail

5. Order Tracking

Test Case	Input	Expected Output	Result
Real-Time Location Tracking	Customer opens tracking screen for active order	Accurate real-time location of delivery partner displayed	Pass/Fail
Status Update Notification	Order status changes from "preparing" to "out for delivery"	Push notification sent to customer about status change	Pass/Fail
ETA Calculation	Delivery in progress with traffic conditions changing	ETA updates dynamically based on current conditions	Pass/Fail
Delivery Confirmation	Partner completes delivery	Order status updates to "delivered", notification sent	Pass/Fail
Lost GPS Signal	Delivery partner temporarily loses GPS signal	System handles interruption gracefully, resumes tracking when signal returns	Pass/Fail

Non-Functional Test Cases

1. Performance Testing

Test Case	Input	Expected Output	Result
Order Processing Speed	New order submitted	Order processed and dark store assigned within 3 seconds	Pass/Fail
Route Calculation Time	Request for delivery route	Route calculated and displayed within 2 seconds	Pass/Fail
App Load Time	User opens mobile app	App loads completely within 2 seconds on standard connection	Pass/Fail
Real-time Inventory Updates	Item purchased, inventory decremented	Inventory count updates across system within 1 second	Pass/Fail
Payment Processing Speed	Customer completes payment	Payment processed and confirmed within 3 seconds	Pass/Fail

2. Scalability Testing

Test Case	Input	Expected Output	Result
Peak Load Handling	Simulate 1000 concurrent orders	System processes all orders without significant slowdown	Pass/Fail
New Dark Store Addition	Add new dark store to network	System incorporates new store in routing algorithms immediately	Pass/Fail
Geographic Expansion	Expand service to new city area	System scales to support new area without performance degradation	Pass/Fail
Database Growth	Database grows to 10 million orders	Query response times remain under 100ms	Pass/Fail
Holiday Rush Simulation	5x normal order volume during flash sale	System handles increased load with <10% performance impact	Pass/Fail

3. Reliability Testing

Test Case	Input	Expected Output	Result
Server Failover	Primary server failure	Automatic failover to backup with <5 seconds downtime	Pass/Fail
Data Consistency	Simultaneous inventory updates from multiple sources	Data remains consistent across all system components	Pass/Fail
Network Interruption	5-second network outage during order processing	System recovers and completes transaction after reconnection	Pass/Fail
Long-Term Stability	System runs continuously for 7 days	No degradation in performance or memory leaks	Pass/Fail
Error Recovery	Force application crash during order process	System recovers with no data loss, transaction completes	Pass/Fail

4. Security Testing

Test Case	Input	Expected Output	Result
Data Encryption	Sensitive customer data in transit	All sensitive data properly encrypted using industry standards	Pass/Fail
Authentication Testing	Login with invalid credentials	Access denied with appropriate error message	Pass/Fail
SQL Injection Attempt	Input with SQL injection code in search field	Query properly sanitized, injection attempt blocked	Pass/Fail
Authorization Testing	Partner attempts to access admin features	Access denied based on role permissions	Pass/Fail
Session Management	User inactive for 30 minutes	Session expires, user required to re-authenticate	Pass/Fail

5. User Experience Testing

Test Case	Input	Expected Output	Result
UI Responsiveness	User interaction with app interface	UI responds to touch inputs within 100ms	Pass/Fail
Interface Consistency	Navigate through different app sections	Consistent design, navigation, and behavior throughout	Pass/Fail
Error Message Clarity	Force validation error by submitting invalid data	Clear, actionable error message displayed	Pass/Fail
Accessibility Compliance	Screen reader enabled on app	All elements accessible and properly labeled for screen readers	Pass/Fail
Multi-device Compatibility	App installed on various Android/iOS devices	App functions correctly across all supported devices	Pass/Fail

Story Boarding

1. Customer Journey Storyboard

Scene 1: App Launch & Authentication

- **Screen**: Splash screen with Zepto logo, followed by login/signup options
- **User Action**: User opens app and logs in with credentials or social media
- **System Response**: Verifies identity and loads personalized home screen

Scene 2: Home Screen & Product Browse

- **Screen**: Grid of product categories, search bar at top, "Quick Reorder" section
- **User Action**: User browses categories or uses search to find specific items
- **System Response**: Shows products with real-time availability and pricing

Scene 3: Product Selection & Cart Building

- **Screen**: Product detail view with quantity selector, "Add to Cart" button
- **User Action**: User selects quantity and adds items to cart
- **System Response**: Cart icon updates with item count and running total

Scene 4: Checkout Process

- **Screen**: Cart review, delivery address confirmation, payment method selection
- **User Action**: User reviews cart, confirms address, selects payment method
- **System Response**: Displays estimated delivery time based on nearest dark store

Scene 5: Order Confirmation & Tracking

- **Screen**: Order confirmation with timer showing delivery countdown
- **User Action**: User can tap "Track Order" to see real-time status
- **System Response**: Shows order being packed, assigned to delivery partner, and en route

Scene 6: Delivery Complete & Feedback

- **Screen**: Delivery confirmation with order details and feedback request
- **User Action**: User rates delivery experience
- **System Response**: Thanks user and offers quick reorder option

2. Delivery Partner App Storyboard

Scene 1: Partner Login & Availability

- **Screen**: Login screen followed by availability toggle
- **User Action**: Partner marks themselves as available for deliveries
- **System Response**: Begins monitoring location for potential order assignments

Scene 2: Order Assignment

- **Screen**: New order notification with accept/decline options
- **User Action**: Partner accepts order
- **System Response**: Shows dark store location and navigation instructions

Scene 3: Order Pickup

- **Screen**: Dark store arrival confirmation, order pickup checklist
- **User Action**: Partner confirms order pickup
- **System Response**: Updates order status and shows customer delivery route

Scene 4: Navigation & Delivery

- **Screen**: Turn-by-turn navigation with optimized route
- **User Action**: Partner follows route to customer location
- **System Response**: Provides real-time traffic updates and route adjustments

Scene 5: Delivery Completion

- **Screen**: Delivery confirmation
- **User Action**: Partner marks order as delivered
- **System Response**: Updates availability status for new devices

3. UI Components Breakdown

Customer App Components

Header Elements

- **App Logo & Navigation Bar**: Prominent placement with intuitive navigation
- **Search Bar**: Intelligent search with predictive text and filters
- **Location Indicator**: Shows delivery address with change option
- **ETA Display**: Shows estimated delivery time prominently

Product Browse Elements

- **Category Navigation**: Visual grid of grocery categories
- **Product Cards**: Image, name, price, weight/quantity, availability indicator
- **Quick Add Buttons**: "+/-" buttons for rapid cart building
- **Stock Status Indicators**: Real-time availability (In stock/Low stock/Out of stock)
- **Sorting & Filtering**: Options to sort by price, popularity, etc.

Cart & Checkout Elements

- **Cart Summary**: Itemized list with quantities and subtotals
- **Delivery Address Selector**: Map integration for precise location selection
- **Payment Method Selector**: Multiple payment options with saved methods
- **Promo Code Input**: Field for applying discounts
- **Order Timeline**: Visual representation of 10-minute delivery promise
- **Real-time Tracking Map**: Shows order status and delivery partner location

Delivery Partner App Components

Order Management Elements

- **Order Queue**: List of pending, active, and completed deliveries
- **Order Details Card**: Customer location, items, special instructions
- **Accept/Decline Buttons**: Option to accept or decline new orders
- **Navigation Interface**: Turn-by-turn directions with optimal route
- **Status Update Controls**: Buttons to mark key milestones (picked up, delivered)
- **Problem Reporting Tool**: Interface to report issues during delivery

Performance Dashboard Elements

- **Earnings Tracker**: Daily/weekly earnings and incentives
- **Rating Display**: Customer feedback and performance metrics
- **Heatmap View**: Shows high-demand areas for strategic positioning
- **Status Toggle**: Online/offline availability control

4. System Dashboard (Admin View)

Real-time Monitoring Interface

- **Dark Store Map**: Geographic view of all dark stores with status indicators
- **Inventory Levels**: Real-time stock levels across all locations
- **Delivery Partner Network**: Active partners and their current status/location
- **Order Flow Visualization**: Visual representation of orders in different stages
- **Performance Metrics**: KPIs including average delivery time, order completion rate
- **Alert System**: Notifications for inventory issues or delivery delays

Analytics Dashboard

- **Demand Forecasting**: AI-powered predictions for inventory planning
- **Heat Maps**: Geographic distribution of orders by time and location
- **Performance Trends**: Historical data on delivery times and service quality
- **Route Efficiency Analysis**: Optimization suggestions based on historical data

User Interface Modelling

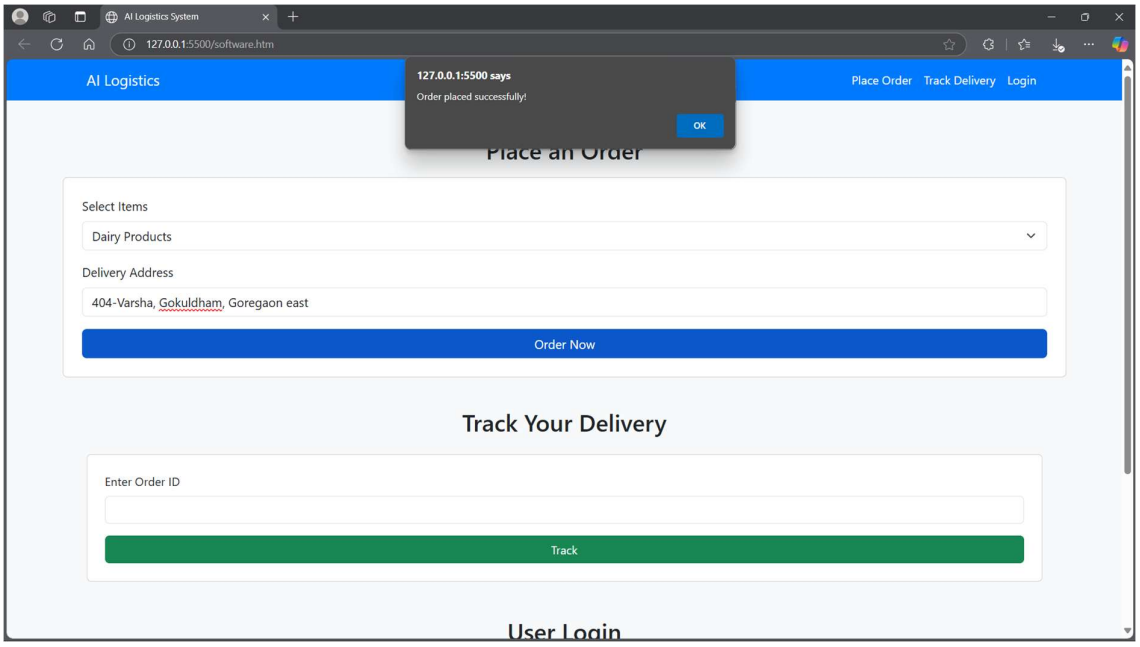
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AI Logistics System</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
    }
    .navbar {
      background-color: #007bff !important;
    }
    .navbar-brand, .nav-link {
      color: white !important;
    }
    .card {
      transition: transform 0.3s;
    }
    .card:hover {
      transform: scale(1.05);
    }
  </style>
```

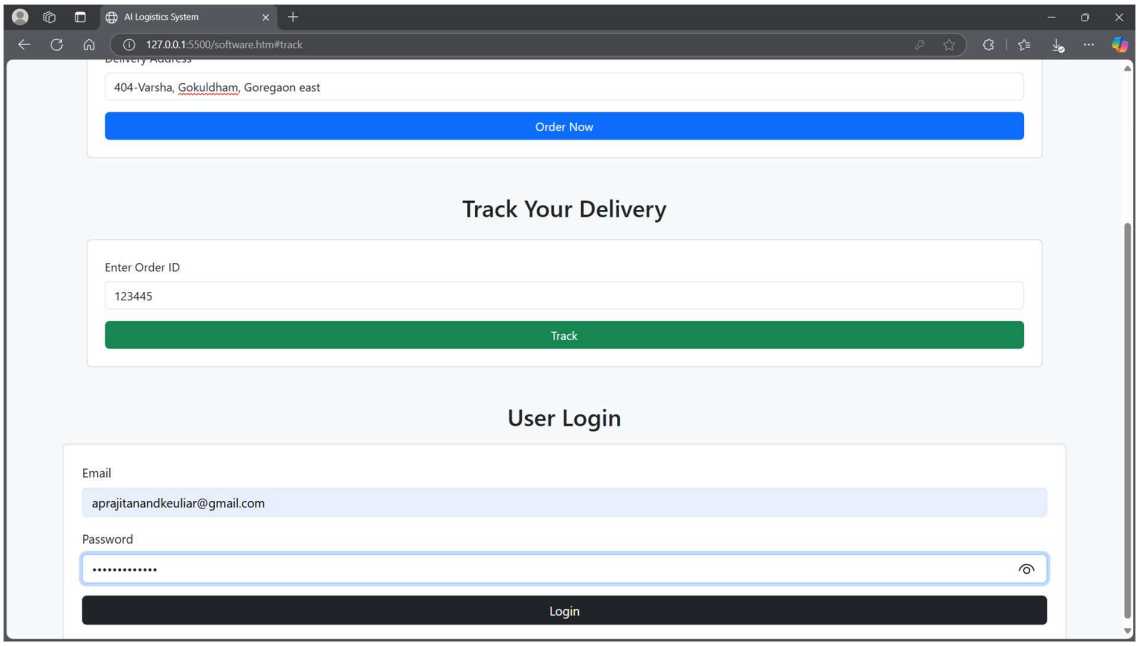
```
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark">
    <div class="container">
      <a class="navbar-brand" href="#">AI Logistics</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item"><a class="nav-link" href="#order">Place Order</a></li>
          <li class="nav-item"><a class="nav-link" href="#track">Track Delivery</a></li>
          <li class="nav-item"><a class="nav-link" href="#login">Login</a></li>
        </ul>
      </div>
    </div>
  </nav>

  <div class="container mt-5">
    <section id="order" class="mb-5">
      <h2 class="text-center mb-4">Place an Order</h2>
      <div class="card p-4">
        <form id="orderForm">
          <div class="mb-3">
            <label class="form-label">Select Items</label>
            <select class="form-select">
              <option>Fresh Produce</option>
```

```
            <option>Dairy Products</option>
            <option>Grocery Essentials</option>
          </select>
        </div>
        <div class="mb-3">
          <label class="form-label">Delivery Address</label>
          <input type="text" class="form-control" required>
        </div>
        <button type="submit" class="btn btn-primary w-100">Order Now</button>
      </form>
    </div>
  </section>

  <section id="track" class="mb-5">
    <h2 class="text-center mb-4">Track Your Delivery</h2>
    <div class="card p-4">
      <form id="trackForm">
        <div class="mb-3">
          <label class="form-label">Enter Order ID</label>
          <input type="text" class="form-control" required>
        </div>
        <button type="submit" class="btn btn-success w-100">Track</button>
      </form>
    </div>
  </section>

  <section id="login">
    <h2 class="text-center mb-4">User Login</h2>
```

```
        <div class="card p-4">
          <form id="loginForm">
            <div class="mb-3">
              <label class="form-label">Email</label>
              <input type="email" class="form-control" required>
            </div>
            <div class="mb-3">
              <label class="form-label">Password</label>
              <input type="password" class="form-control" required>
            </div>
            <button type="submit" class="btn btn-dark w-100">Login</button>
          </form>
        </div>
      </section>
    </div>

    <script>
      document.getElementById("orderForm").addEventListener("submit",
function(event) {
        event.preventDefault();
        alert("Order placed successfully!");
      });

      document.getElementById("trackForm").addEventListener("submit",
function(event) {
        event.preventDefault();
        alert("Tracking order...");
      });
```

```
    document.getElementById("loginForm").addEventListener("submit", function(event)
{
        event.preventDefault();
        alert("Login successful!");
    });
  </script>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></
script>
</body>
</html>
```




AI Logistics System
127.0.0.1:5500/software.htm#track
404-Varsha, Gokuldham, Goregaon east
Order Now
Track Your Delivery
Enter Order ID
123445
Track
User Login
Email
aprajitanandkeuliar@gmail.com
Password
Login