Courier Website Development Documentation

1. Project Overview

Develop a courier service platform facilitating:

- Customer Functions: Register, log in, create shipments, track packages, view shipment history, and manage profiles.
- Delivery Partner Functions: Register, log in, view assigned deliveries, update delivery statuses, and manage profiles.
- Admin Functions: Oversee system operations, manage users, and monitor shipments.

2. Technology Stack

- Backend: Node.js with Express.js
- Frontend: HTML, CSS, JavaScript (with optional frameworks like React or EJS)
- Database: MySQL
- Authentication: JSON Web Tokens (JWT) or session-based authentication
- Deployment: Platforms like Heroku, Vercel, or AWS

3. Database Schema

3.1. Customers Table

```
CREATE TABLE customers (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100) UNIQUE,
  phone VARCHAR(15),
  password VARCHAR(255),
  address TEXT,
```

```
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

3.2. Delivery Partners Table

```
CREATE TABLE delivery_partners (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100) UNIQUE,
  phone VARCHAR(15),
  vehicle_type VARCHAR(50),
  password VARCHAR(255),
  is_available BOOLEAN DEFAULT TRUE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

3.3. Shipments Table

```
CREATE TABLE shipments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  customer_id INT,
  tracking_id VARCHAR(50) UNIQUE,
  sender_name VARCHAR(100),
  sender_address TEXT,
  receiver_name VARCHAR(100),
  receiver_address TEXT,
  package_weight FLOAT,
  delivery_type ENUM('Normal','Express'),
  status ENUM('Booked', 'Picked_Up', 'In_Transit', 'Delivered') DEFAULT 'Booked',
```

```
  assigned_partner_id INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (customer_id) REFERENCES customers(id),
  FOREIGN KEY (assigned_partner_id) REFERENCES delivery_partners(id)
);
```

3.4. Payments Table (Optional)

```
CREATE TABLE payments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  shipment_id INT,
  amount DECIMAL(10,2),
  payment_mode VARCHAR(50),
  status ENUM('Pending', 'Paid'),
  paid_at TIMESTAMP,
  FOREIGN KEY (shipment_id) REFERENCES shipments(id)
);
```

4. API Endpoints

4.1. Customer Routes

- POST /api/customers/register
- POST /api/customers/login
- GET /api/customers/dashboard
- POST /api/shipments
- GET /api/shipments
- GET /api/shipments/:tracking_id
- PUT /api/customers/profile

4.2. Delivery Partner Routes

- POST /api/partners/register
- POST /api/partners/login
- GET /api/partners/dashboard
- GET /api/partners/shipments
- PUT /api/shipments/:id/status
- PUT /api/partners/profile

5. Frontend Pages

5.1. Customer Pages

- Dashboard
- Create Shipment
- My Shipments
- Track Shipment
- Profile

5.2. Delivery Partner Pages

- Dashboard
- Assigned Deliveries
- Update Delivery Status
- Profile

6. User Flows

6.1. Customer Flow

- Register/Login
- Dashboard
- Create Shipment
- My Shipments
- Track Shipment
- Profile

6.2. Delivery Partner Flow

- Register/Login
- Dashboard
- Assigned Deliveries
- Update Delivery Status
- Profile

7. Deployment Considerations

- Environment Variables
- Database Hosting
- Backend Hosting
- Frontend Hosting
- Domain Management