Plan for quickship – logistic management platform

1. Scope & Objectives

Goal: Create a core logistics engine that handles order entry, driver assignment, and delivery proof.

- **Admin Web Portal:** Central command for managing orders and staff.
- **Driver Mobile Web:** A simplified interface for drivers to process deliveries.
- **Public Tracking:** A simple page for customers to check status via ID.

2. Detailed Requirements

Functional Requirements

- **Order Management:** Create, view, and edit shipments (Sender, Receiver, Weight, COD).
- **Assignment Logic:** Admin can manually assign a driver to a specific shipment.
- **Status Workflow:** 4 Stages: Pending → Picked Up → In Transit → Delivered.
- **Proof of Delivery (PoD):** Drivers must be able to upload one photo upon delivery.
- **Public Tracking:** A search bar that returns a timeline of the package's status.

Non-Functional Requirements

- **Mobile First:** The driver interface must work perfectly on Chrome/Safari (iOS/Android).
- **Reliability:** Status updates must be logged with a timestamp for audit trails.

3. Simplified Database Schema

SQL

```
-- Users (Admin & Drivers)
Table profiles {
  id uuid primary_key
  full_name text
  role text -- 'admin' or 'driver'
}

-- Shipments
Table shipments {
  id uuid primary_key
  tracking_number text unique
  status text -- 'pending', 'picked_up', 'in_transit', 'delivered'
  driver_id uuid references profiles.id
  sender_name text
```

receiver_address text

pod_image_url text -- Link to the delivery photo

created_at timestamp

}

4. Risk Management

Potential Risk	**Mitigation Strategy**
Scope Creep	**Rule:** No "Advanced Analytics" or "Auto-Routing" in the first 8 weeks. Stick to manual assignment.
Technical Debt	Use **Supabase** to handle Backend-as-a-Service, saving 2 weeks of manual API coding.
Testing Gaps	Spend the first 30 mins of every Monday "Live Testing" the previous week's work on real phones.