

# Wahid Auto Parts Hub – MVP Architecture with Code Skeleton

## 1. Executive Summary

This document defines the exact MVP architecture for Wahid Auto Parts Hub and includes starter backend and frontend code skeletons to accelerate execution. The code is intentionally minimal and production-oriented.

## 2. Backend Architecture (Node.js + Express)

Project Structure:

```
backend/
  └── src/
    ├── app.js
    └── routes/
      ├── products.js
      ├── sellers.js
      └── orders.js
    └── models/
      ├── Product.js
      ├── Seller.js
      └── Order.js
    └── db.js
  └── package.json
```

Database Connection (PostgreSQL):

```
javascript
// db.js
const { Pool } = require("pg");

const pool = new Pool({
  host: "localhost",
  user: "postgres",
  password: "password",
  database: "wahid_mvp",
  port: 5432,
});

module.exports = pool;
```

Product Model:

```
javascript
// models/Product.js
const pool = require("../db");

async function getProducts() {
```

```

        const res = await pool.query("SELECT * FROM products");
        return res.rows;
    }

module.exports = { getProducts };

```

### 3. API Layer

Product API Endpoint:

```

javascript
// routes/products.js
const express = require("express");
const router = express.Router();
const Product = require("../models/Product");

router.get("/", async (req, res) => {
    const products = await Product.getProducts();
    res.json(products);
});

module.exports = router;

```

### 4. Frontend Architecture (Next.js)

Basic Product Listing Page:

```

javascript
// pages/index.js
import { useEffect, useState } from "react";

export default function Home() {
    const [products, setProducts] = useState([]);

    useEffect(() => {
        fetch("http://localhost:3001/products")
            .then(res => res.json())
            .then(setProducts);
    }, []);

    return (
        <div>
            <h1>Wahid Auto Parts Hub</h1>
            {products.map(p => (
                <div key={p.id}>
                    <strong>{p.name}</strong> - {p.price} ETB
                    <a href={`https://wa.me/${p.whatsapp}`}>
                        Order via WhatsApp
                    </a>
                </div>
            )));
        </div>
    );
}

```

```
}
```

## 5. Database Schema (SQL)

```
sql
CREATE TABLE sellers (
    id SERIAL PRIMARY KEY,
    name TEXT,
    phone TEXT,
    verified BOOLEAN DEFAULT false
);

CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    name TEXT,
    sku TEXT,
    category TEXT
);

CREATE TABLE seller_products (
    id SERIAL PRIMARY KEY,
    seller_id INT REFERENCES sellers(id),
    product_id INT REFERENCES products(id),
    price NUMERIC,
    stock_status TEXT
);
```

## 6. WhatsApp Order Flow

Orders are handled as inquiries. Clicking the WhatsApp link sends a pre-filled message. Payment and delivery remain offline at MVP stage.

## 7. Deployment Notes

Frontend can be deployed on Vercel. Backend can be deployed on Railway or AWS Lightsail. Environment variables should be used for database credentials.