🚛 Truckoria Developer Manual

Overview

Truckoria is a high-performance logistics platform designed for industrial-grade cargo delivery across Ethiopia and beyond. This developer manual outlines how to architect, build, and deploy the full-stack application, ensuring secure onboarding, real-time operations, and multi-role support.

🔧 Tech Stack Recommendation

Frontend:

- React.js with Tailwind CSS
- Redux Toolkit for state management
- i18next for multi-language support
- Socket.IO for real-time communication

Backend:

- Node.js + Express.js
- MongoDB (Mongoose ORM)
- Redis (Job queues, caching)
- Firebase or Twilio for OTP
- Cloudinary/S3 for media uploads
- OpenCV + AWS Rekognition (for AI-KYC verification)

Deployment:

- Dockerized microservices
- NGINX for reverse proxy
- CI/CD with GitHub Actions or GitLab CI
- Deployed on AWS or Hetzner

📚 Required Libraries

Frontend Libraries

- **react** – UI library
- **react-router-dom** – Routing

- **redux**, **@reduxjs/toolkit** – State management
- **axios** – HTTP requests
- **i18next**, **react-i18next** – Language translation
- **socket.io-client** – Real-time sockets
- **formik**, **yup** – Forms and validation
- **tailwindcss** – Styling
- **react-icons** – Icons
- **react-toastify** – Notifications
- **moment.js** – Time formatting

Backend Libraries

- **express** – Core backend framework
- **mongoose** – MongoDB ORM
- **jsonwebtoken** – Auth tokens
- **bcryptjs** – Password hashing
- **nodemailer** – Email notifications
- **twilio** or **firebase-admin** – OTP services
- **multer** – File uploads
- **cloudinary** – Media storage
- **socket.io** – WebSocket handling
- **node-cron** – Scheduled jobs
- **redis** – Caching and queues
- **aws-sdk**, **opencv4nodejs** – AI & document verification

🌍 Multi-Language Integration

Use `i18next` with JSON language files:

- `en.json`
- `am.json`
- `om.json`

Load translations via language switcher in the header and store preference in localStorage.

📲 1. User Signup & Onboarding

API Endpoints:

- `POST /api/auth/signup`
- `POST /api/auth/verify-otp`
- `POST /api/auth/upload-docs`
- `POST /api/auth/kyc-check`

Flow:

1. **Signup Form:** Submit name, phone, email, password, role
2. **OTP Generation:** Send via Firebase/Twilio, valid for 5 mins
3. **OTP Verification:** Max 3 attempts
4. **Role-Based Doc Upload:** See roles below
5. **Trigger AI-KYC System**: Upload files to Cloudinary/S3 → KYC service
6. **Notify Result:** Show approval/rejection UI

👥 2. Role Management & User Flows

A. Employers (Customers)

- **Post Job:** `POST /api/jobs`
- **View Drivers:** `GET /api/drivers`
- **Assign Driver:** `PATCH /api/jobs/:id/assign`
- **Track Job:** `GET /api/jobs/:id/track`
- **Rate Driver:** `POST /api/ratings`

B. Drivers

- **Job List:** `GET /api/jobs?status=open`
- **Accept Job:** `PATCH /api/jobs/:id/accept`
- **Start Job:** `PATCH /api/jobs/:id/start`
- **Upload Proof:** `POST /api/jobs/:id/proof`
- **Update Status:** `PATCH /api/jobs/:id/status`

C. Companies

- **Add Driver:** `POST /api/company/drivers`
- **Post Bulk Jobs:** `POST /api/company/jobs`
- **Dashboard Metrics:** `GET /api/company/analytics`

- **Bulk Payment:** `POST /api/company/payments`

📍 3. Real-Time Features

Socket.IO Events:

- `job:update-status`
- `user:notification`
- `chat:message`

Push Notification Service:

- Use Firebase Cloud Messaging
- WebSocket fallback for instant updates

💬 In-App Chat

- Real-time driver ↔ employer communication
- Save chat logs in MongoDB
- Media sharing enabled

🔒 AI-KYC Verification

- On document upload, trigger KYC microservice
- Use ML libraries (OpenCV, AWS Rekognition) to:
  - Compare selfies with ID
  - Validate document types
- Return decision and store metadata

💼 Admin Panel

Access:

- Route: `/admin`
- Login secured by JWT & IP restriction

Features:

- Approve/reject documents
- Ban/suspend users
- View job/payment history

- Moderate chat logs
- Analytics dashboard

💰 Payments

- Integrate Telebirr SDK
- Support bank transfer receipts
- Escrow account model (driver gets paid upon delivery)

Payment Flow:

1. Job marked complete
2. Funds released to driver via backend service

📊 Analytics

Use MongoDB Aggregations or Apache Kafka for large-scale analytics:

- Delivery success rate
- Job duration trends
- Earnings by region
- Top rated drivers/fleets

🔁 CI/CD & Deployment

- Use Docker Compose for local dev
- GitHub Actions for automated tests + builds
- Deploy via Kubernetes (optional for scaling)
- Monitor with Grafana + Prometheus

🧪 Testing

- Jest + Supertest for API
- Cypress for end-to-end UI
- Load testing with Artillery

🚀 Summary

Truckoria is engineered for scale, trust, and performance. With its robust KYC, real-time tracking, role separation, and secure payment system, it's set to become Ethiopia’s premier logistics platform for industrial freight.

📧 For questions or contributions, contact the Truckoria Dev Team: dev@truckoria.et