



# RTO Management System (RMS)

A comprehensive **RTO Management System** designed to digitize and automate Regional Transport Office operations such as **Driving License management, Vehicle Registration, Appointments, Payments, Violations, and Reporting** using **Agile/Scrum methodology**.

## 📌 Project Overview

The **RTO Management System (RMS)** aims to replace manual and paper-based RTO processes with a secure, scalable, and user-friendly digital platform. The system serves **citizens, RTO officers, inspectors, and administrators**, ensuring transparency, efficiency, and regulatory compliance.

## 🎯 Objectives

- Automate Learner License (LL) and Driving License (DL) processes
- Digitize vehicle registration and ownership transfer
- Enable online appointment scheduling and test management
- Provide secure online fee and fine payments
- Centralize citizen, vehicle, and license records
- Generate real-time reports and analytics
- Improve service delivery and reduce processing time



## Team Structure (Agile – 5 Members)

| Role                 | Responsibility                             |
|----------------------|--|
| Scrum Master         | Sprint facilitation, blocker removal       |
| Product Owner        | Backlog management, requirement definition |
| Backend Developer    | APIs, database, business logic             |
| Frontend Developer   | UI/UX, React components                    |
| Full-Stack Developer | Integration, testing, deployment           |

# System Architecture

---

Frontend (React.js)

↓

API Gateway (Express.js)

↓

Backend Microservices

↓

PostgreSQL Database

↓

Redis Cache

## Core Modules

---

### 1. User Authentication & Authorization

- Citizen, Officer, Inspector, Admin roles
- JWT-based authentication
- Role-Based Access Control (RBAC)

### 2. Citizen Management

- Citizen profile creation & management
- Aadhaar & contact information
- Search and verification

### 3. Driving License Management

- Learner License (LL) application
- Driving License (DL) issuance & renewal
- License status tracking

### 4. Vehicle Registration Management

- New vehicle registration (RC)
- Ownership transfer
- Vehicle verification

### 5. Appointment & Driving Test Management

- Slot booking for RTO services
- Driving test scheduling
- Inspector test result updates

## 6. Fee & Payment Management

- Online payments (License, RC, fines)
- Invoice & receipt generation
- Payment reports

## 7. Document Management

- Aadhaar, address proof, insurance, PUC uploads
- Secure document storage

## 8. Violation & Penalty Management

- Traffic violation recording
- Online fine payments
- Violation history tracking

## 9. Notifications & Communication

- Email & SMS alerts
- Application status updates
- Announcements

## 10. Reporting & Analytics

- License & vehicle statistics
- Revenue reports
- Operational dashboards

## Technology Stack

---

### Frontend

- React.js 18+
- TypeScript
- Redux Toolkit
- Material UI / Ant Design

- Axios & React Query

## Backend

- Node.js 20+
- Express.js
- RESTful APIs
- JWT Authentication

## Database & Caching

- PostgreSQL 15+
- Redis

## DevOps & Deployment

- Docker
- GitHub Actions (CI/CD)
- AWS / GCP
- Nginx

## Testing

- Jest (Unit Testing)
- React Testing Library
- Postman (API Testing)
- Cypress (E2E Testing)

## Project Structure

---

```
rto-management-system/
|__ frontend/
|   |__ src/
|   |__ public/
|   |__ package.json
|
|__ backend/
|   |__ src/
|   |__ controllers/
|   |__ routes/
|   |__ models/
|   |__ package.json
```

```
|  
|   |-- database/  
|   |   |-- schema.sql  
|   |   |-- migrations/  
|  
|   |-- docs/  
|   |   |-- API_Documentation.md  
|   |   |-- Architecture.md  
|   |   |-- User_Manual.md  
|  
|   |-- docker-compose.yml  
|   |-- README.md
```

## Agile Process

---

- Sprint Duration: **2 Days**
- Total Sprints: **26**
- Daily Stand-ups
- Sprint Planning & Retrospectives
- Continuous Integration & Deployment

## Key Metrics

---

- Velocity (Story points per sprint)
- Sprint Burndown
- Test Coverage (>80%)
- Bug Rate
- System Uptime (99.9%)

## Risk Management

---

| Risk          | Mitigation              |
|---------------|-------------------------|
| Data Security | Encryption, RBAC        |
| High Traffic  | Caching, load balancing |
| Scope Creep   | Strict backlog control  |

| Risk               | Mitigation          |
|--------------------|---------------------|
| Regulatory Changes | Config-driven rules |

## Getting Started

---

### Prerequisites

- Node.js 20+
- PostgreSQL 15+
- Docker (optional)

### Setup Instructions

```
# Clone repository
git clone https://github.com/your-org/rto-management-system.git

# Backend setup
cd backend
npm install
npm run dev

# Frontend setup
cd frontend
npm install
npm start
```

## Deliverables

---

- Source Code
- API Documentation (Swagger)
- Database Schema
- User & Admin Manuals
- Deployment Guide