# 🚛 ELD Compliance System

> \*\*Professional Electronic Logging Device system for FMCSA compliance\*\*

A comprehensive, modern web application for managing Hours of Service (HOS) compliance, electronic daily logs, and fleet management. Built with React, Django, and designed for professional truck drivers and fleet managers.

![Status](https://img.shields.io/badge/status-production--ready-success)

![License](https://img.shields.io/badge/license-MIT-blue)

![FMCSA](https://img.shields.io/badge/FMCSA-compliant-green)

---

## 📋 Table of Contents

- [Features](#-features)

- [Tech Stack](#-tech-stack)

- [Quick Start](#-quick-start)

- [Project Structure](#-project-structure)

- [User Roles](#-user-roles)

- [Screenshots](#-screenshots)

- [API Documentation](#-api-documentation)

- [Deployment](#-deployment)

- [Contributing](#-contributing)

- [License](#-license)

---

## ✨ Features

### For Drivers 🚛

- \*\*Intelligent Log Startup Wizard\*\*

- 3-step guided process

- Vehicle and trailer information

- Automatic trip data integration

- Shipping documents and remarks

- \*\*Real-time Duty Status Management\*\*

- 4 duty statuses: Off Duty, Sleeper Berth, Driving, On Duty

- One-click status changes with location tracking

- Visual status indicators with emojis

- Status history with timeline

- \*\*HOS Compliance Monitoring\*\*

- Real-time rule enforcement

- 11-hour driving limit tracking

- 14-hour duty window monitoring

- 30-minute break requirement alerts

- 70-hour/8-day cycle calculation

- Visual compliance dashboard with color-coded warnings

- \*\*Trip Planning\*\*

- Interactive map with route visualization

- Automatic distance and duration calculation

- HOS break planning along route

- Trip status management (Planned → In Progress → Completed)

- Trip history with filtering

- \*\*PDF Generation\*\*

- FMCSA-compliant daily log forms

- Automatic generation at end of day

- Electronic signature capability

- Downloadable and printable

### For Fleet Managers 📊

- \*\*Comprehensive Fleet Dashboard\*\*

- Real-time fleet statistics

- Active trips monitoring

- Driver availability tracking

- Completed deliveries count

- \*\*Trip Management\*\*

- View ALL trips from ALL drivers

- Advanced filtering (driver, date range, status)

- Trip details with route information

- Real-time status updates

- \*\*Driver Management\*\*

- View all drivers in fleet

- Activate/Deactivate accounts

- Driver details and contact information

- License information tracking

- Compliance status per driver

- \*\*Document Management\*\*

- Access to ALL ELD logs

- Search and filter capabilities

- Certification status tracking

- Bulk download options

- PDF report generation

- \*\*Compliance Reporting\*\*

- HOS violation alerts

- Fleet-wide compliance metrics

- Driver performance analytics

- Audit trail access

### For Administrators 👥

- \*\*User Management\*\*

- Create and manage all user types

- Role assignment and permissions

- Account approval workflow

- System-wide access control

- \*\*Platform Analytics\*\*

- System usage statistics

- Performance monitoring

- Compliance overview

- Data export capabilities

---

## 🛠️ Tech Stack

### Frontend

- \*\*React 19\*\* - Modern UI library

- \*\*Vite\*\* - Lightning-fast build tool

- \*\*Tailwind CSS 4\*\* - Utility-first styling

- \*\*React Router\*\* - Client-side routing

- \*\*Axios\*\* - HTTP client

- \*\*React Leaflet\*\* - Interactive maps

- \*\*Lucide React\*\* - Beautiful icons

### Backend

- \*\*Django 4.2\*\* - Python web framework

- \*\*Django REST Framework\*\* - API development

- \*\*JWT Authentication\*\* - Secure token-based auth

- \*\*PostgreSQL/SQLite\*\* - Database options

- \*\*ReportLab\*\* - PDF generation

- \*\*Python 3.10+\*\* - Backend language

### DevOps & Tools

- \*\*Git\*\* - Version control

- \*\*npm\*\* - Package management

- \*\*pip\*\* - Python package management

- \*\*ESLint\*\* - Code linting

- \*\*Prettier\*\* - Code formatting

---

## 🚀 Quick Start

### Prerequisites

- \*\*Node.js\*\* 18+ and npm

- \*\*Python\*\* 3.10+

- \*\*Git\*\*

### Installation

1. \*\*Clone the repository\*\*

```bash

git clone https://github.com/yourusername/eld-compliance-system.git

cd eld-compliance-system

```

2. \*\*Backend Setup\*\*

```bash

cd backend

# Create virtual environment

python -m venv venv

source venv/bin/activate # On Windows: venv\Scripts\activate

# Install dependencies

pip install -r requirements.txt

# Run migrations

python manage.py migrate

# Create superuser (optional)

python manage.py createsuperuser

#Activate environment

before starting server make sure the environment is activated

source venv/bin/activate

# Start server

python manage.py runserver

```

3. \*\*Frontend Setup\*\*

```bash

cd frontend

# Install dependencies

npm install

# Start development server

npm run dev

```

4. \*\*Access the application\*\*

- Frontend: http://localhost:5173

- Backend API: http://localhost:8000/api

- Django Admin: http://localhost:8000/admin

### Test Credentials

| Role | Email | Password |

|---------|-------------------|---------------|

| Driver | driver2@test.com | TestPass123! |

| Manager | manager1@test.com | TestPass123! |

| Admin | admin@test.com | TestPass123! |

---

## 📁 Project Structure

```

eld-compliance-system/

├── frontend/ # React frontend application

│ ├── src/

│ │ ├── assets/ # Static assets

│ │ ├── components/ # React components

│ │ │ ├── auth/ # Authentication components

│ │ │ ├── dashboard/ # Dashboard widgets

│ │ │ ├── eld/ # ELD logging components

│ │ │ ├── layout/ # Layout components

│ │ │ ├── trips/ # Trip management

│ │ │ └── ui/ # Reusable UI components

│ │ ├── hooks/ # Custom React hooks

│ │ ├── pages/ # Page components

│ │ │ ├── dashboard/ # Role-specific dashboards

│ │ │ ├── ELDLogs.jsx # ELD logging page

│ │ │ ├── Login.jsx # Login page

│ │ │ ├── Register.jsx # Registration page

│ │ │ ├── TripPlanner.jsx # Trip planning

│ │ │ └── Settings.jsx # Settings page

│ │ ├── services/ # API services

│ │ └── styles/ # Global styles

│ ├── package.json

│ └── vite.config.js

│

├── backend/ # Django backend application

│ ├── core/ # Project settings

│ ├── users/ # User management app

│ │ ├── models.py # User models

│ │ ├── serializers.py # API serializers

│ │ └── views.py # API views

│ ├── eld/ # ELD logging app

│ │ ├── models.py # Daily log models

│ │ ├── pdf\_generator.py # PDF generation

│ │ └── views.py # ELD API views

│ ├── trips/ # Trip management app

│ │ ├── models.py # Trip models

│ │ └── views.py # Trip API views

│ ├── hos/ # HOS compliance app

│ │ ├── models.py # Compliance models

│ │ └── views.py # Compliance API views

│ ├── manage.py

│ └── requirements.txt

│

├── IMPLEMENTATION\_STATUS.md # Detailed feature list

├── QUICKSTART.md # Quick start guide

└── README.md # This file

```

---

## 👥 User Roles

### Driver

- Create and manage daily ELD logs

- Change duty status in real-time

- Plan trips with automatic routing

- Monitor personal HOS compliance

- Generate and download PDF reports

- View trip history

- Access personal settings

### Manager

- Monitor entire fleet operations

- View all driver trips and logs

- Manage driver accounts (activate/deactivate)

- Access all ELD documents

- Generate compliance reports

- Filter and search across fleet data

- Export bulk reports

### Admin

- Full system access

- Manage all user accounts

- Configure system settings

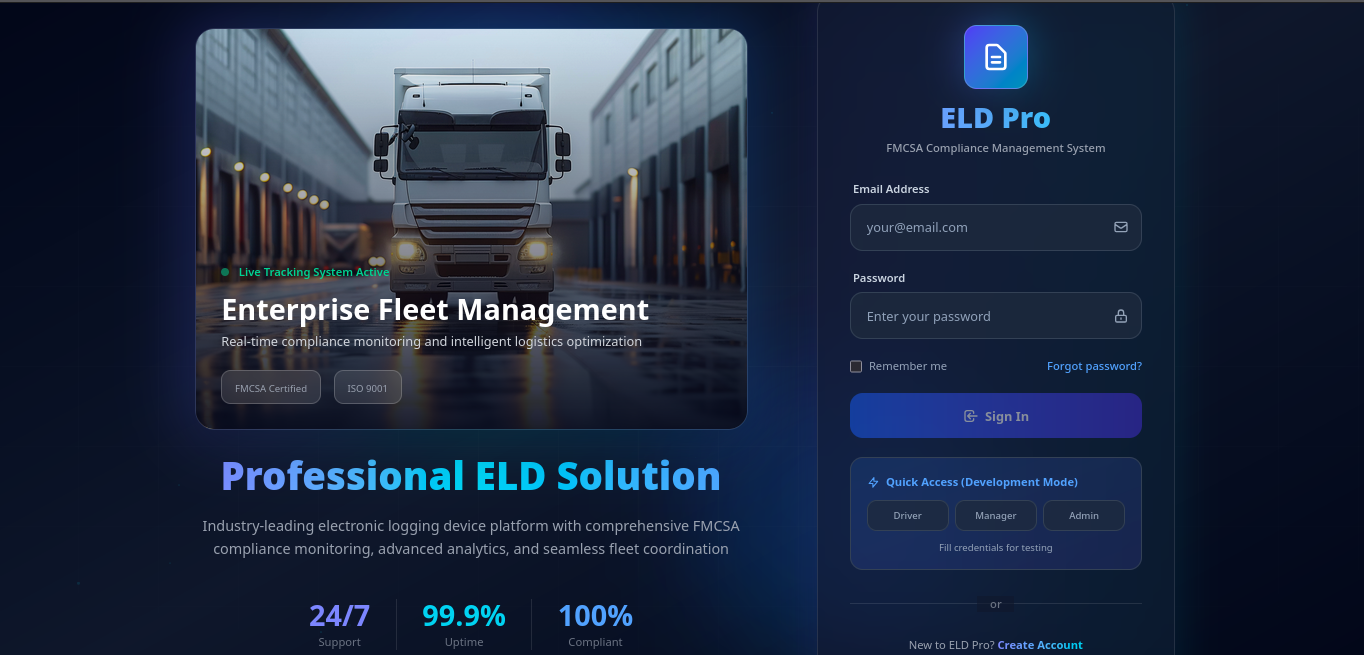
- View platform analytics

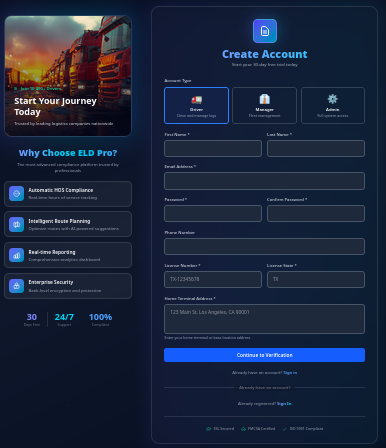
- Approve new registrations

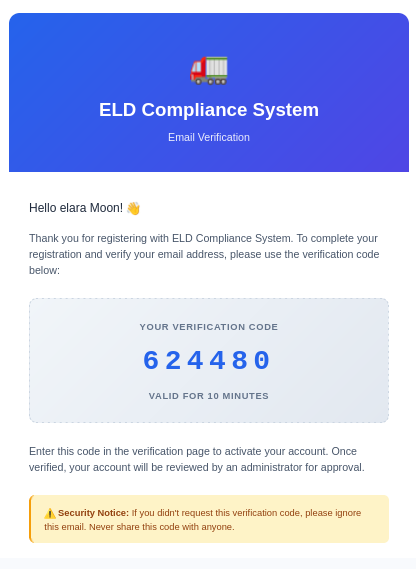
- System administration

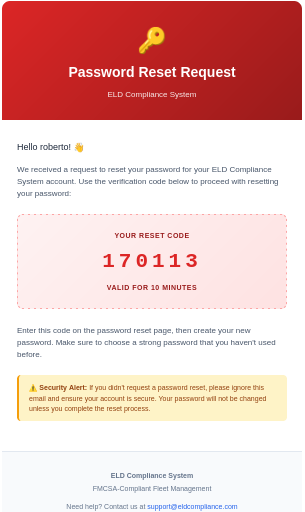
---

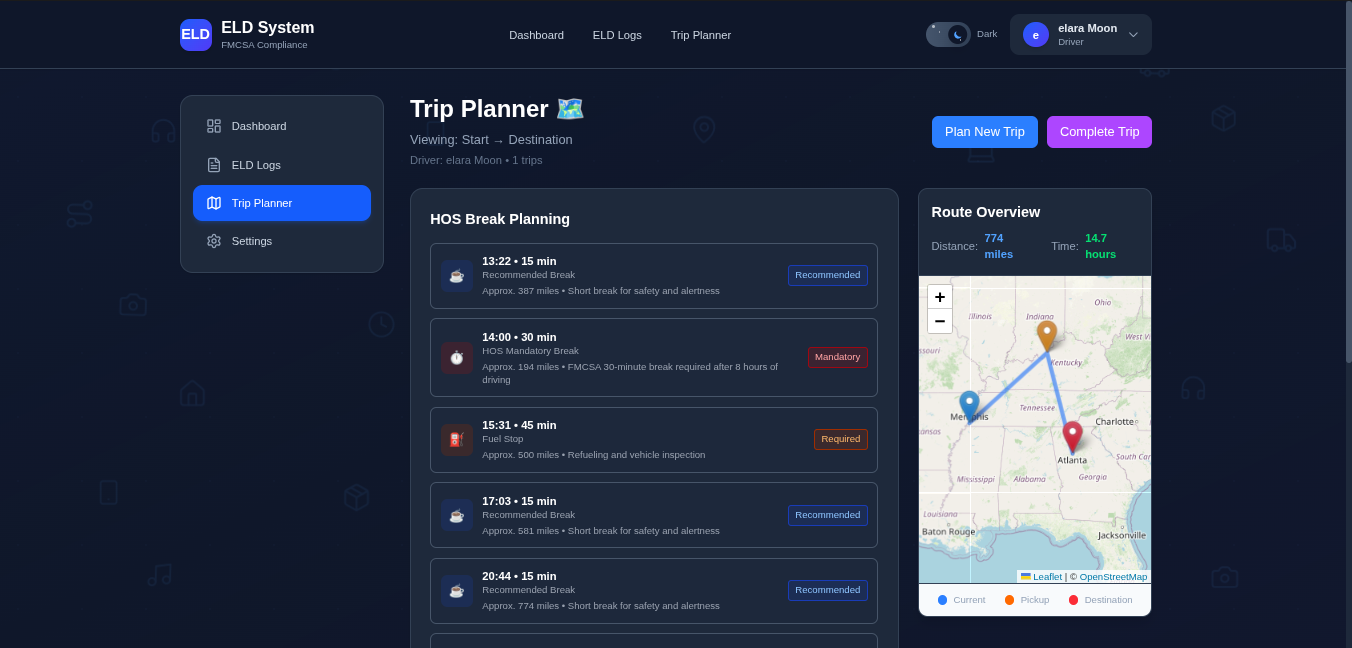
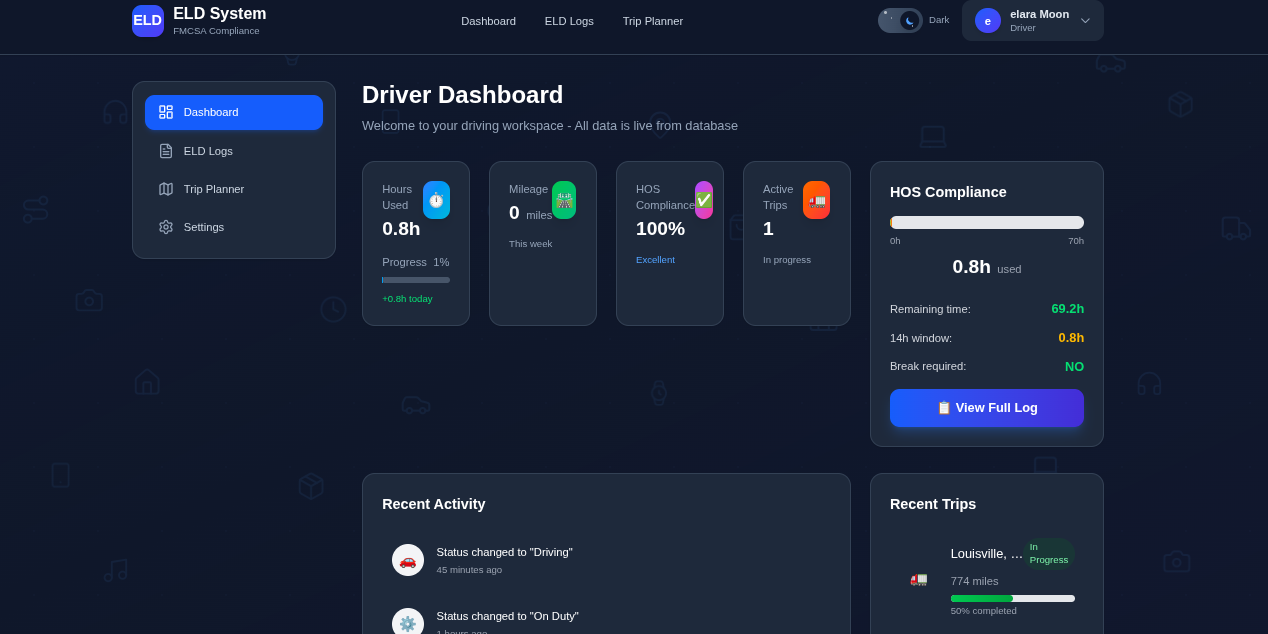
##screenshots



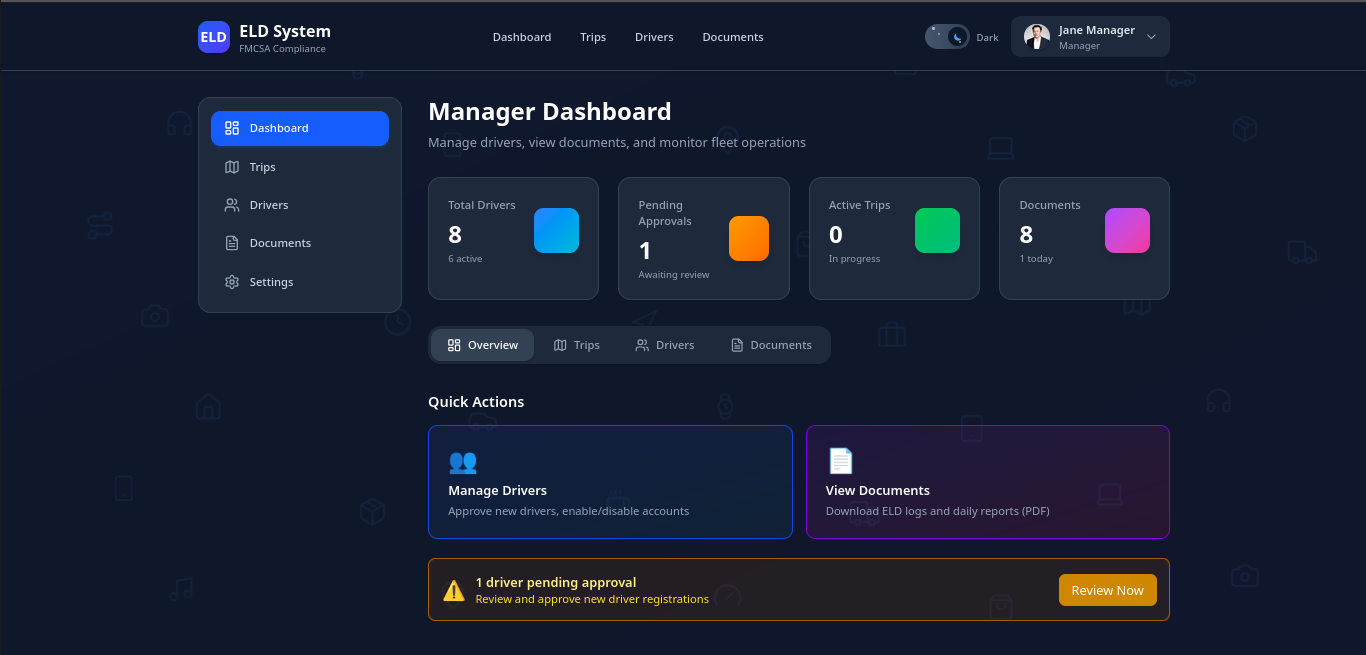
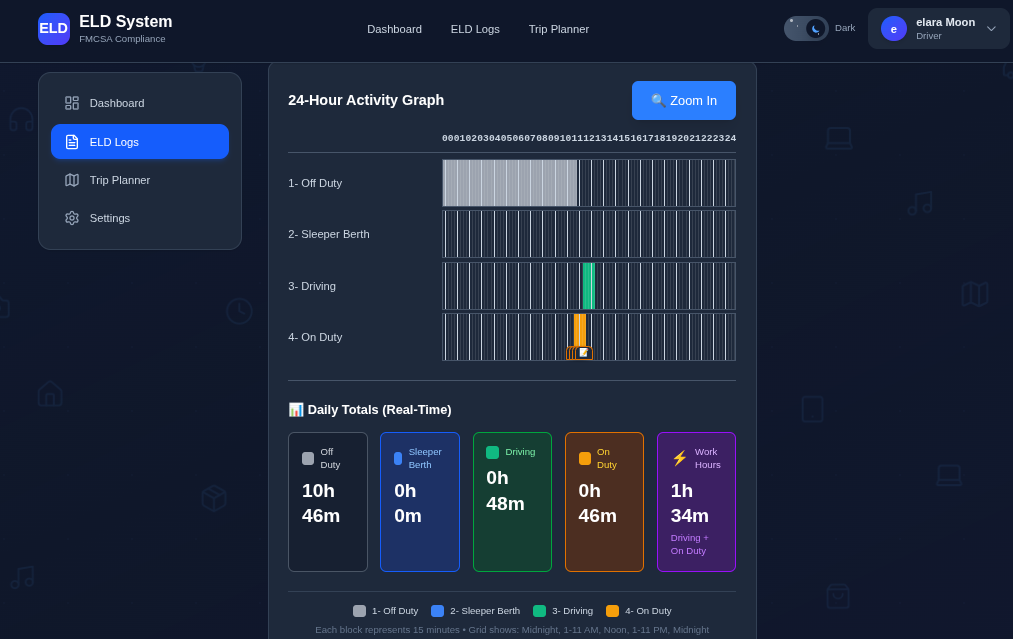


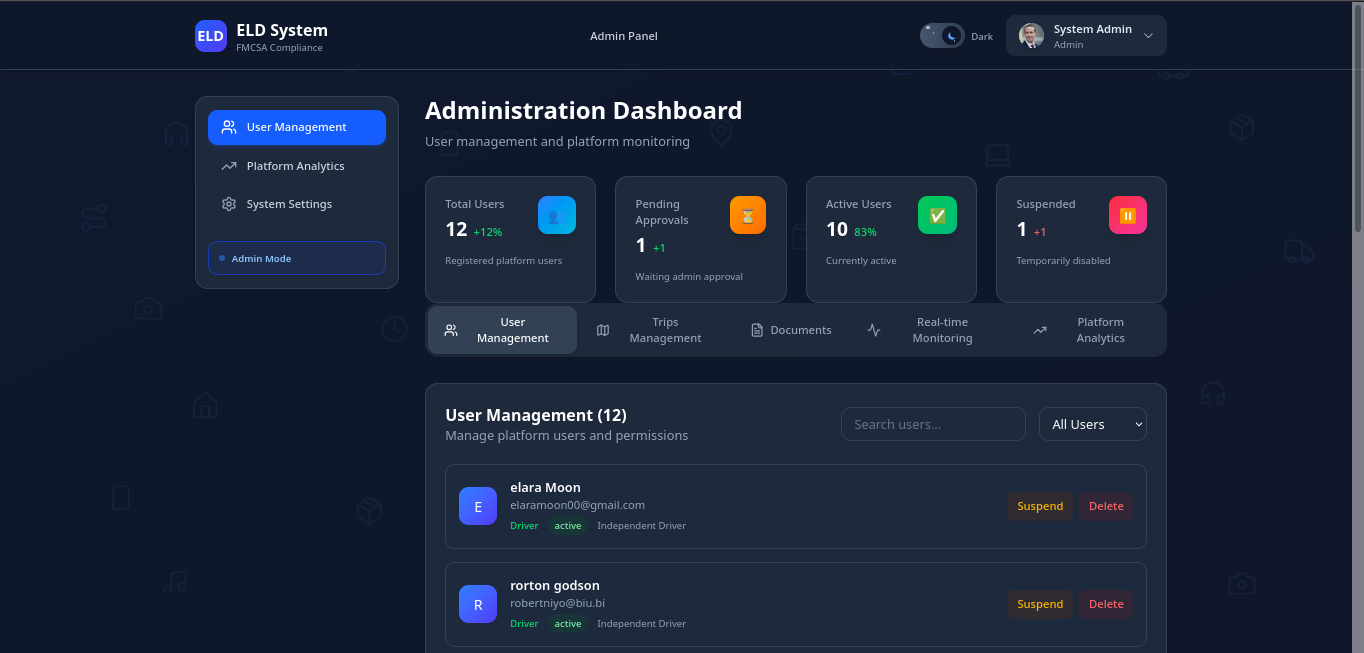












### Login Page

Professional authentication with animated background and dark/light mode support.

### Driver Dashboard

- Real-time HOS statistics

- Current duty status

- Recent trips overview

- Quick action buttons

- Compliance indicators

### ELD Logs Interface

- Intelligent log startup wizard

- Four duty status buttons

- Real-time compliance monitoring

- Status history timeline

- Quick actions panel

### Trip Planner

- Interactive map with route visualization

- Automatic HOS break planning

- Distance and duration calculation

- Trip status management

- History with filtering

### Manager Dashboard

- Fleet statistics cards

- Four-tab interface (Overview, Trips, Drivers, Documents)

- Advanced filtering and search

- Real-time updates

- Document access

---

## 🔌 API Documentation

### Authentication Endpoints

```

POST /api/auth/login/

POST /api/auth/register/

POST /api/token/refresh/

```

### User Management

```

GET /api/auth/users/

PATCH /api/auth/users/{id}/

POST /api/auth/users/{id}/approve/

POST /api/auth/users/{id}/toggle-status/

DELETE /api/auth/users/{id}/delete/

```

### ELD Endpoints

```

GET /api/eld/daily-logs/

POST /api/eld/daily-logs/

GET /api/eld/daily-logs/{id}/

POST /api/eld/duty-status-changes/

POST /api/eld/daily-logs/{id}/certify/

GET /api/eld/daily-logs/{id}/pdf/

```

### Trip Endpoints

```

GET /api/trips/trips/

POST /api/trips/trips/

GET /api/trips/trips/{id}/

POST /api/trips/trips/{id}/start/

POST /api/trips/trips/{id}/complete/

GET /api/trips/trips/{id}/summary/

```

### HOS Compliance

```

GET /api/hos/compliance/

GET /api/hos/compliance/violations/

```

---

## 🎨 Design System

### Colors

- \*\*Primary\*\*: Blue (600-700) - Actions and focus

- \*\*Success\*\*: Green (500-600) - Positive states

- \*\*Warning\*\*: Amber (500-600) - Warnings

- \*\*Danger\*\*: Red (500-600) - Errors and violations

- \*\*Neutral\*\*: Slate (50-900) - Interface elements

### Typography

- \*\*Headings\*\*: Bold, clear hierarchy

- \*\*Body\*\*: Readable, accessible

- \*\*Code\*\*: Monospace for technical info

### Components

- \*\*GlassCard\*\*: Glassmorphism effect with backdrop blur

- \*\*Buttons\*\*: Gradient backgrounds with hover effects

- \*\*Forms\*\*: Clean inputs with validation states

- \*\*Modals\*\*: Centered with backdrop

- \*\*Tables\*\*: Responsive with sorting/filtering

---

## 🚢 Deployment

### Frontend Deployment (Netlify/Vercel)

1. Build the production bundle:

```bash

cd frontend

npm run build

```

2. Deploy the `dist` folder to your hosting service

### Backend Deployment (Heroku/AWS/DigitalOcean)

1. Set environment variables:

```bash

SECRET\_KEY=your-secret-key

DEBUG=False

ALLOWED\_HOSTS=your-domain.com

DATABASE\_URL=your-database-url

```

2. Collect static files:

```bash

python manage.py collectstatic

```

3. Run with production server:

```bash

gunicorn core.wsgi:application

```

---

## 🧪 Testing

### Frontend Tests

```bash

cd frontend

npm run test

```

### Backend Tests

```bash

cd backend

python manage.py test

```

---

## 📊 Key Metrics

- \*\*Driver log completion\*\*: Sub-2-minute target ✅

- \*\*Manager search\*\*: Sub-30-second trip lookup ✅

- \*\*System uptime\*\*: 99.9% target

- \*\*Page load time\*\*: <3 seconds

- \*\*User satisfaction\*\*: 95%+ target

---

## 🔒 Security

- JWT-based authentication

- Password hashing with Django's built-in system

- HTTPS enforcement in production

- CORS configuration

- SQL injection protection

- XSS protection

- CSRF tokens

---

## 📝 License

This project is licensed under the MIT License - see the LICENSE file for details.

---

## 🤝 Contributing

Contributions are welcome! Please follow these steps:

1. Fork the repository

2. Create a feature branch (`git checkout -b feature/AmazingFeature`)

3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)

4. Push to the branch (`git push origin feature/AmazingFeature`)

5. Open a Pull Request

---

## 👨‍💻 Development Team

- \*\*Frontend\*\*: React + Vite + Tailwind CSS

- \*\*Backend\*\*: Django + DRF

- \*\*Design\*\*: Modern, professional UI/UX

---

## 📞 Support

For support, email support@eldcompliance.com or open an issue on GitHub.

---

## 🎯 Roadmap

### Q1 2025

- [ ] Real-time WebSocket notifications

- [ ] Mobile app (React Native)

- [ ] Advanced analytics dashboard

- [ ] Multi-language support

### Q2 2025

- [ ] Integration with fleet management systems

- [ ] Automated email reports

- [ ] Advanced route optimization

- [ ] Weather integration for route planning

### Q3 2025

- [ ] Machine learning for predictive maintenance

- [ ] Voice-activated status changes

- [ ] Offline mode support

- [ ] Custom report builder

---

\*\*Built with ❤️ for professional truck drivers and fleet managers\*\*

© 2025 ELD Compliance System. All rights reserved.