Compliance Management System

An API-Based Solution for Legal Professionals

Srishti

April 22, 2025

Bennett University

Abstract

This report details the design and implementation of the Compliance Management System, a comprehensive API-based solution tailored for legal professionals operating within the construction industry. The system aims to provide real-time retrieval and analysis of compliance information, streamlining workflows and enhancing accuracy. Key features include direct access to up-to-date compliance data, automated processing and analysis of legal documents using Optical Character Recognition (OCR) and AI, and seamless integration with external APIs like Google Gemini and Mistral. The backend is built using Node.js, Express.js, and TypeScript, supported by a MongoDB database structured for development, testing, and production environments. The frontend utilizes React.js and Tailwind CSS. This report covers the system's architecture, core functionalities, technological stack, setup procedures, and security considerations.

Contents

1 Introduction

1.1 Background

The construction industry operates within a complex web of regulations, standards, and legal requirements that are subject to frequent updates. Legal professionals in this sector face significant challenges in staying abreast of real-time compliance information and efficiently processing large volumes of legal documents. Traditional methods often involve manual research across disparate sources and time-consuming document review, leading to potential delays, errors, and increased risk of non-compliance.

1.2 Problem Statement and Importance

There is a critical need for a centralized, efficient system that provides legal professionals in construction with immediate access to current compliance data and automates the analysis of relevant legal documents. The Compliance Management System addresses this need by offering an integrated platform that leverages modern APIs and AI technologies. This system aims to significantly reduce research time, minimize errors associated with manual processing, enhance the accuracy of compliance checks, and ultimately allow legal teams to focus on higher-value strategic tasks. Ensuring timely and accurate compliance is paramount in mitigating legal risks and financial penalties within the construction sector.

1.3 Project Objectives

The primary objectives of the Compliance Management System project are:

- To develop a robust and scalable API backend providing endpoints for compliance information retrieval and document processing.
- To integrate external APIs (Google Gemini, Mistral OCR) for accessing real-time data, performing OCR, and conducting AI-powered text analysis.
- To implement a secure and efficient MongoDB database structure with separate environments for development, testing, and production.
- To build a user-friendly frontend interface using React.js for interacting with the API.
- To establish comprehensive API documentation using OpenAPI/Swagger.
- To ensure system security through best practices in credential management and environment configuration.

2 System Architecture and Technologies

2.1 Architectural Approach

The system is designed following an API-first approach, separating the backend logic and data management from the frontend presentation layer. This promotes modularity, scalability, and allows for potential integration with other client applications in the future. The architecture emphasizes clear separation of concerns, utilizing controllers for handling requests, services for business logic, models for data representation, and dedicated modules for external API integrations.

2.2 Technology Stack

The project utilizes a modern technology stack chosen for performance, developer productivity, and ecosystem support:

- **Frontend:** React.js, Tailwind CSS
- **Backend:** Node.js, Express.js, TypeScript

- **Database:** MongoDB (using Mongoose ODM)
- **API Documentation:** OpenAPI (Swagger)
- **External APIs:**
  - Google Gemini 2.5 Flash (Web Search Analysis/Summaries)
  - Mistral OCR (Document Text Extraction)
- **Development Build Tools:** npm, TypeScript Compiler (tsc)

2.3 Project Structure

The codebase is organized into distinct modules to enhance maintainability and clarity. The primary structure is as follows:

```
compliance-management/
          src/
                api/
                      controllers/     # Request handlers
                      middlewares/     # Express middlewares
                      routes/          # API routes
                      validators/      # Request validation
                config/              # Application configuration
                db/                  # Database setup and models
                integrations/        # External API integration
                      perplexity/      # Perplexity API (deprecated)
                      mistral/         # Mistral OCR API
                      gemini/          # Google Gemini API
                models/              # Mongoose models (redundant, likely within db/)
                services/            # Business logic
                types/               # TypeScript types
                utils/               # Utility functions
          tests/                   # Test files
                unit/                # Unit tests
                integration/         # Integration tests
                e2e/                 # End-to-end tests
          dist/                    # Compiled code
          .env.example             # Environment variable template
          .gitignore
          package.json
          tsconfig.json
```

Listing 1: Project Directory Structure

Note: The original description listed ‘models/‘ separately; it’s often integrated within ‘db/‘ when using Mongoose.

3 Features and Implementation

3.1 Core Features

The Compliance Management System provides the following key functionalities:

- **Compliance Information Retrieval:** Enables users to request real-time compliance updates and specific legal information relevant to the construction industry. This leverages the Google Gemini API for web searches across government and news sources.
- **Document Processing:** Allows users to upload legal documents (e.g., PDFs, images). The system uses the Mistral OCR API to extract text content from these documents.
- **Text Analysis and Summarization:** Utilizes the Google Gemini API to analyze the extracted text from documents, providing summaries, identifying key clauses, or answering specific questions about the content.

- **API Integration:** Provides a unified interface abstracting the complexities of interacting with multiple external APIs (Gemini, Mistral).
- **Secure Data Storage:** Employs MongoDB with distinct databases for different operational environments, managed via Mongoose models.

3.2 External API Integrations

- **Google Gemini 2.5 Flash (Web Search):** Integrated via the ‘src/integrations/gemini/‘ module to perform targeted web searches for retrieving the latest compliance information, regulations, and relevant news articles.
- **Mistral OCR:** Integrated via the ‘src/integrations/mistral/‘ module. This API is called upon document upload to accurately extract textual content from various document formats.
- **Google Gemini 2.5 Flash (Analysis):** Also integrated via the ‘src/integrations/gemini/‘ module. It takes the text extracted by Mistral OCR or user queries as input to perform natural language processing tasks like summarization, analysis, and question-answering.

4 Setup and Configuration

4.1 Prerequisites

Before running the system, ensure the following are installed:

- Node.js (v16.x or higher)
- MongoDB (v4.x or higher, local instance or Atlas cluster)
- Valid API keys for Mistral and Google Gemini

4.2 Installation

Follow these steps to set up the project locally:

1. Clone the repository:

```
git clone https://github.com/your-username/compliance-management.git
cd compliance-management

```

2. Install dependencies:

```
npm install

```

3. Create the environment configuration file:

```
cp .env.example .env

```

4. Update the ‘.env‘ file with your MongoDB connection strings and API keys.

4.3 Running the Application

- Start the development server (with hot-reloading):

```
npm run dev

```

- Build the project for production:

```
npm run build
```

- Start the production server:

```
npm start
```

- Run linters:

```
npm run lint
```

- Run automated tests:

```
npm test
```

4.4 API Documentation

Comprehensive API documentation, generated using OpenAPI/Swagger, is available at the following endpoint once the server is running:

`http://localhost:3000/api/v1/docs`

(Assuming the default port is 3000 and base path is '/api/v1')

5 Environment and Database Configuration

5.1 Environment Variable Management

The application utilizes a centralized configuration system managed via the 'src/config/' directory and environment variables loaded from a '.env' file.

- **Centralized:** A single source defines configuration requirements.
- **Type Safe:** Uses TypeScript for defining configuration structure.
- **Validation:** Automatically validates the presence of required environment variables (e.g., 'MISTRAL$_API_KEY$', '$GEMINI_API$ $specific$'$MONGODB_URI$')$on startup. The application will fail to start if required variables are missing, logging the speci$

- **Defaults:** Provides sensible defaults for optional parameters.

5.1.1 Required Environment Variables

- 'MISTRAL$_API_KEY$' : $API key for Mistral OCR.$'$GEMINI_API_KEY$' : $API key for Google Gemini.$

- 'MONGODB$_URI_DEV$' : $MongoDB connection string for development.$'$MONGODB_URI_TEST$' : $MongoDB connectionstri$

- 'MONGODB$_URI_PROD$' : $MongoDB connection string for production.$'$MONGODB_PASSWORD$' : $Password for the Mong$

5.2 MongoDB Configuration

The system connects to MongoDB Atlas, using distinct databases for separation of concerns:

- Development: 'compliance-dev'
- Testing: 'compliance-test'

- Production: ‘compliance-prod‘

The appropriate database connection string is selected based on the ‘NODE$_ENV$‘$environment variable$.

5.2.1 Connection Features

The database connection is configured for robustness and performance:

- Connection pooling (defaulting to 5-10 connections).
- Automatic retries on transient network errors for reads and writes.
- Configured timeouts to prevent indefinite hangs.
- ’Majority’ write concern for enhanced data durability.
- Secure handling of connection strings (passwords should not be logged).

5.2.2 Testing Database Connections

A dedicated script can be used to verify connectivity to all configured databases:

```
npm run test:db
```

This script attempts to connect, write, read, and delete a test document in each environment’s database.

6 Security Considerations

Security is a critical aspect of the system, particularly due to the handling of API keys and potentially sensitive document data.

6.1 Handling Sensitive Data

The following practices are employed to protect credentials:

- **Environment Variables:** All sensitive keys (API keys, database passwords) are managed exclusively through environment variables, never hardcoded in the source code.
- **‘.gitignore‘:** The ‘.env‘ file containing actual secrets is listed in ‘.gitignore‘ to prevent accidental commits. An ‘.env.example‘ file serves as a template.
- **Database Users:** Separate MongoDB users with the least privilege necessary are recommended for each environment (dev, test, prod).
- **API Key Rotation:** Regularly rotating external API keys is recommended as a security best practice.
- **Production Secrets Management:** For production deployments, using dedicated secrets management services (e.g., AWS Secrets Manager, HashiCorp Vault, platform-specific environment variable stores) is strongly advised over static ‘.env‘ files.

6.2 Dependency Auditing

Regularly auditing project dependencies for known vulnerabilities is crucial:

```
# Check for vulnerabilities
npm audit

# Attempt to automatically fix vulnerabilities
npm audit fix
```

7 Conclusion and Future Work

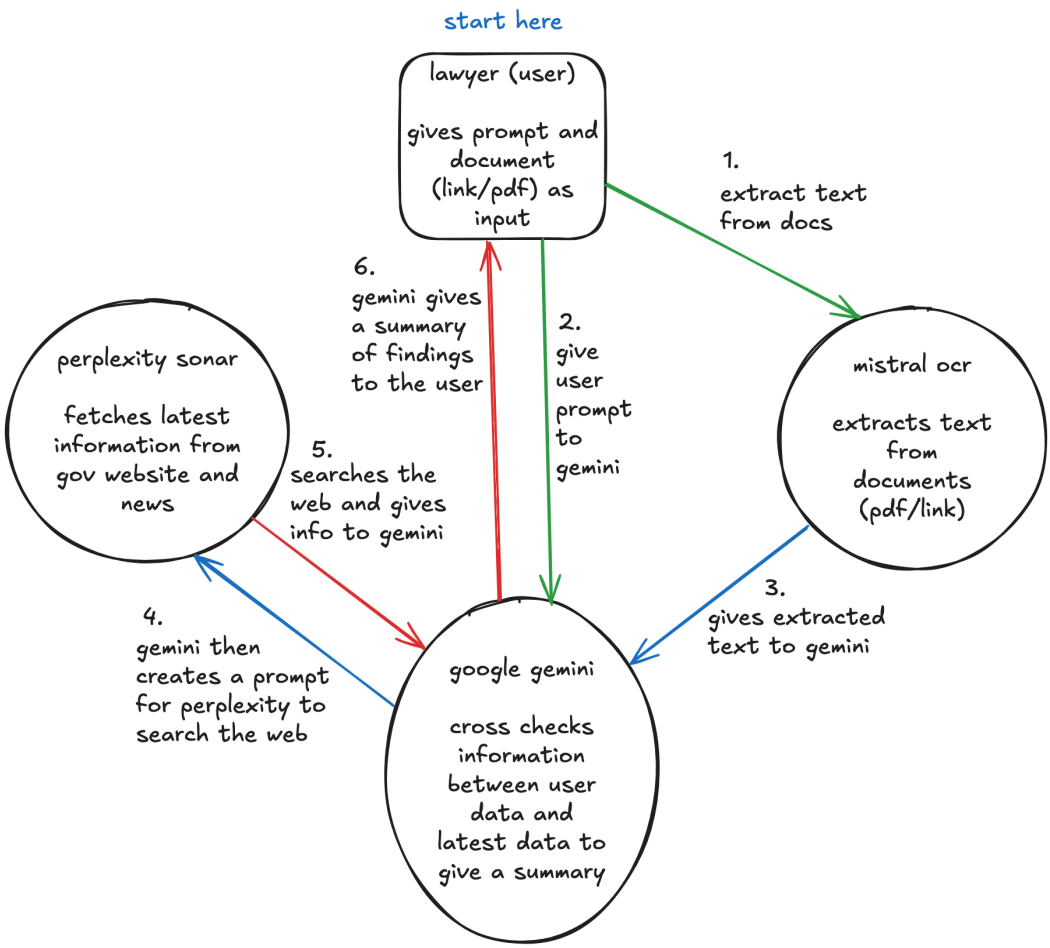
7.1 Summary

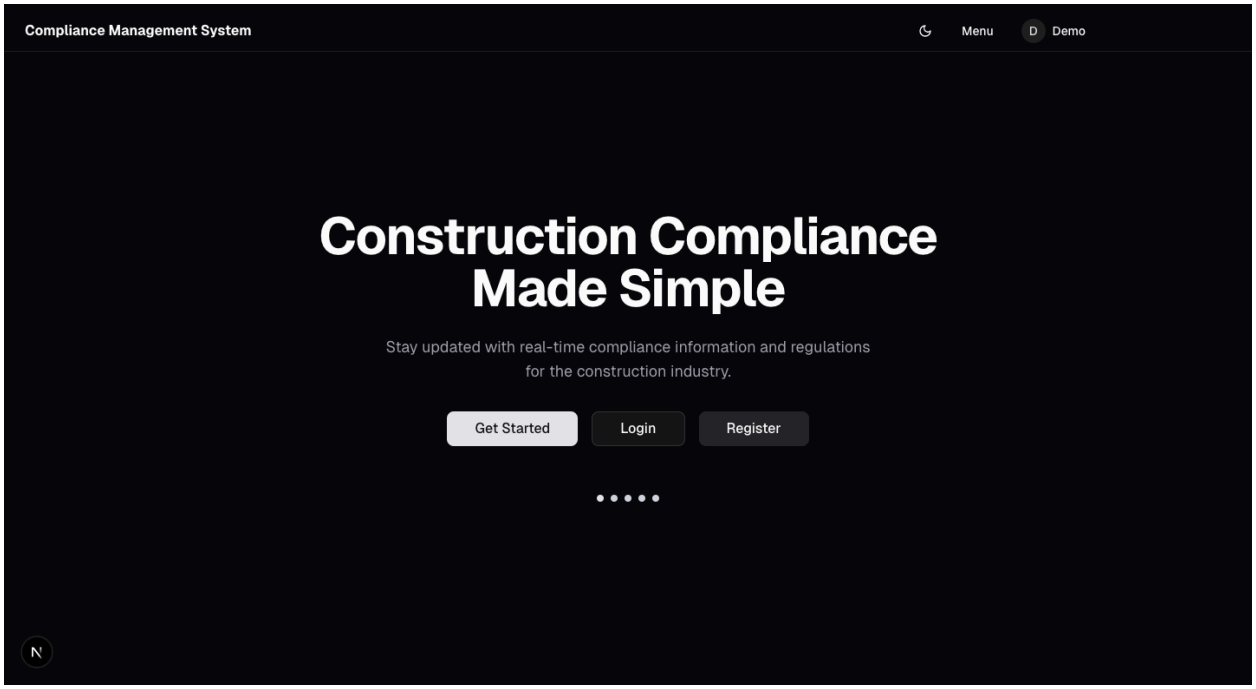
The Compliance Management System provides a much-needed solution for legal professionals in the construction industry by automating and centralizing compliance information retrieval and document analysis. Leveraging Node.js, React, MongoDB, and powerful external APIs like Google Gemini and Mistral OCR, the system offers significant improvements in efficiency, accuracy, and risk mitigation compared to traditional methods. The project follows modern development practices, including an API-first architecture, TypeScript for type safety, environment-based configuration, and robust security measures.

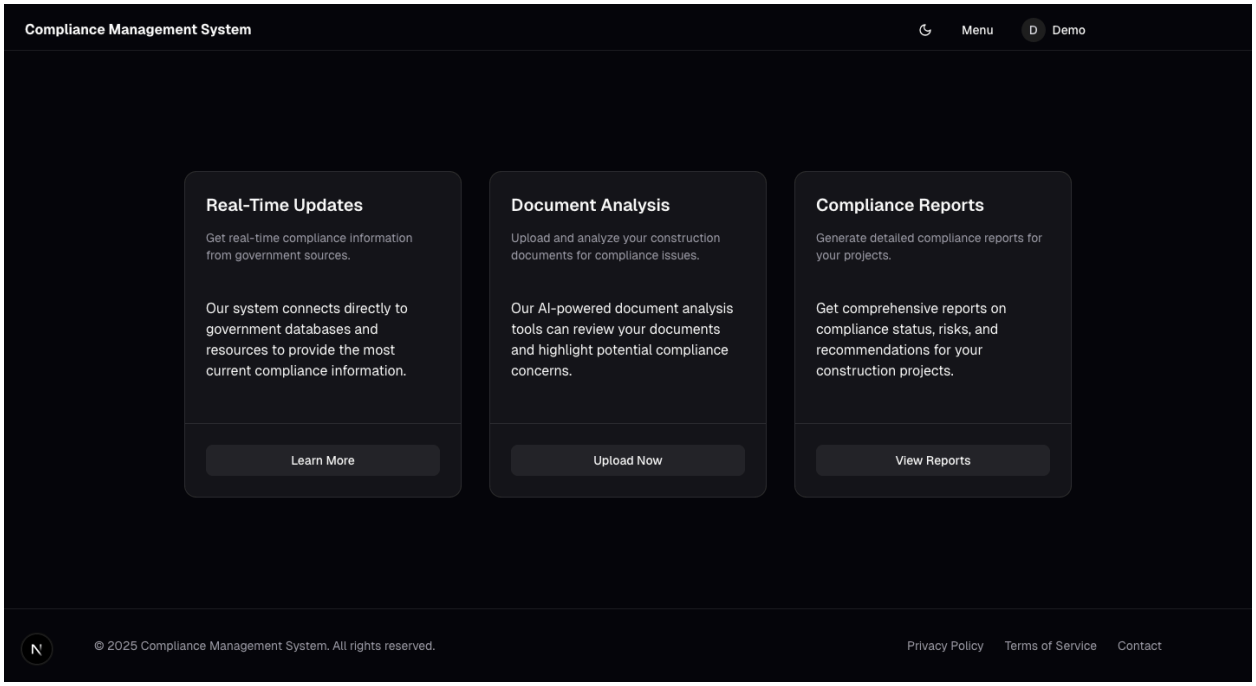
7.2 Future Work

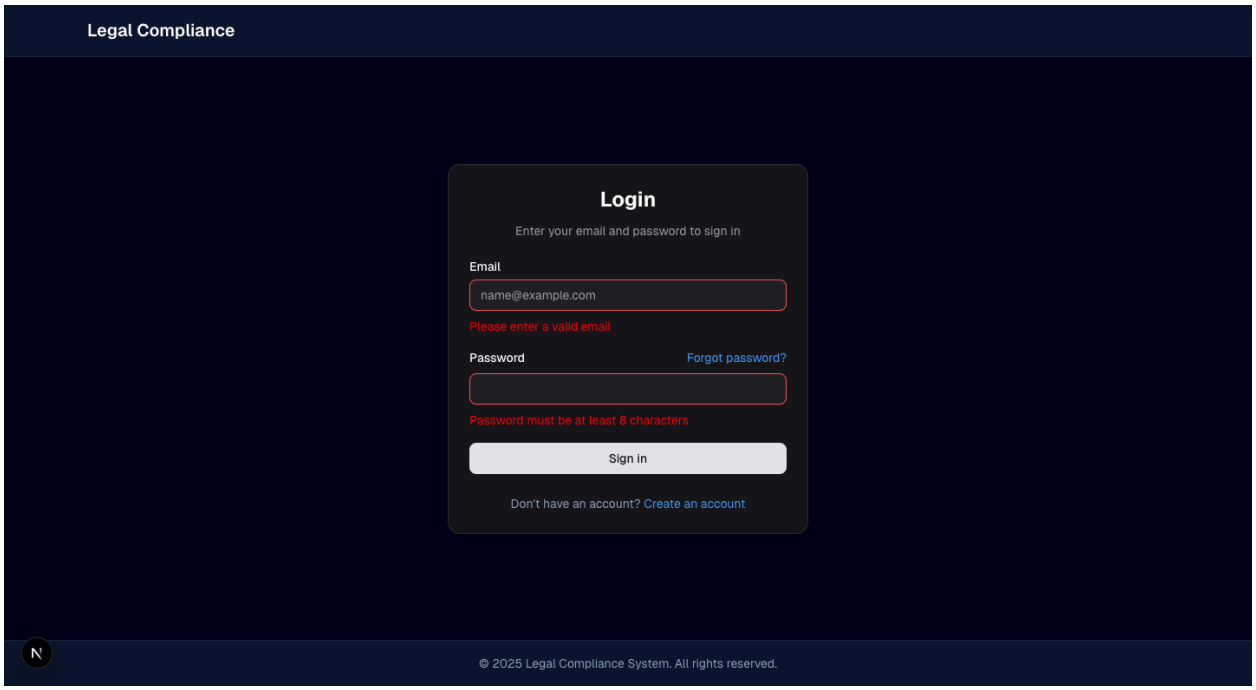
Potential areas for future development and enhancement include:

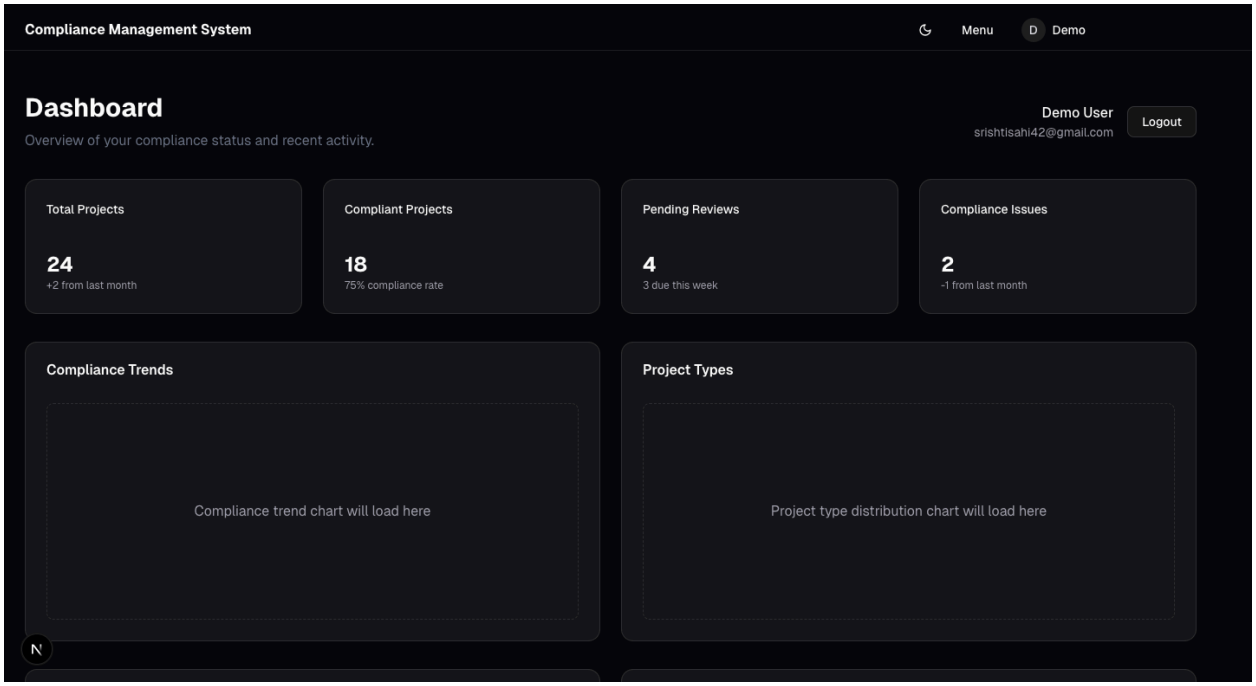
- **Advanced Analytics:** Implementing more sophisticated analysis features on extracted document text, such as identifying specific clauses, risks, or obligations automatically.
- **User Roles and Permissions:** Adding role-based access control to manage user permissions within the system.
- **Expanded Integrations:** Integrating with additional legal databases, project management tools, or industry-specific data sources.
- **Enhanced Frontend Features:** Developing more interactive UI components for document comparison, annotation, or workflow management.
- **Real-time Notifications:** Implementing a system to notify users proactively about relevant compliance updates.
- **CI/CD Pipeline:** Setting up a continuous integration and continuous deployment pipeline for automated testing and deployment.

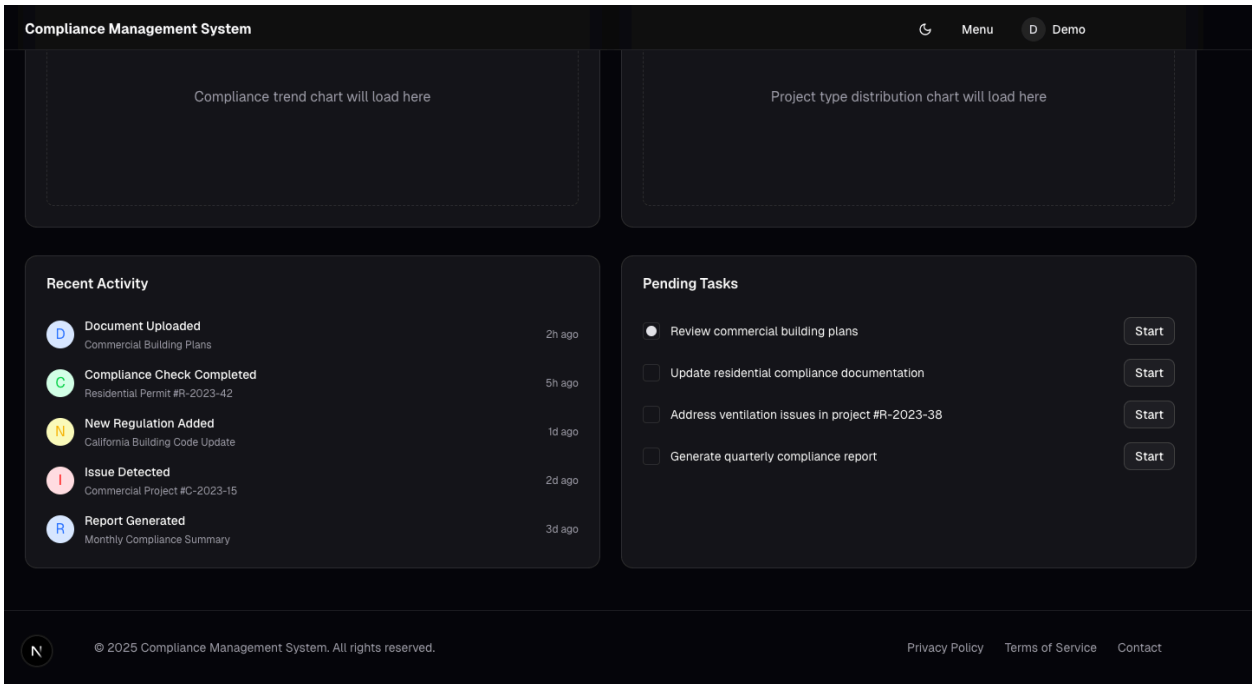

start here
lawyer (user)
gives prompt and document (link/pdf) as input
1.
extract text from docs
2.
give user prompt to gemini
3.
gives extracted text to gemini
4.
gemini then creates a prompt for perplexity to search the web
5.
searches the web and gives info to gemini
6.
gemini gives a summary of findings to the user
perplexity sonar
fetches latest information from gov website and news
mistral ocr
extracts text from documents (pdf/link)
google gemini
cross checks information between user data and latest data to give a summary

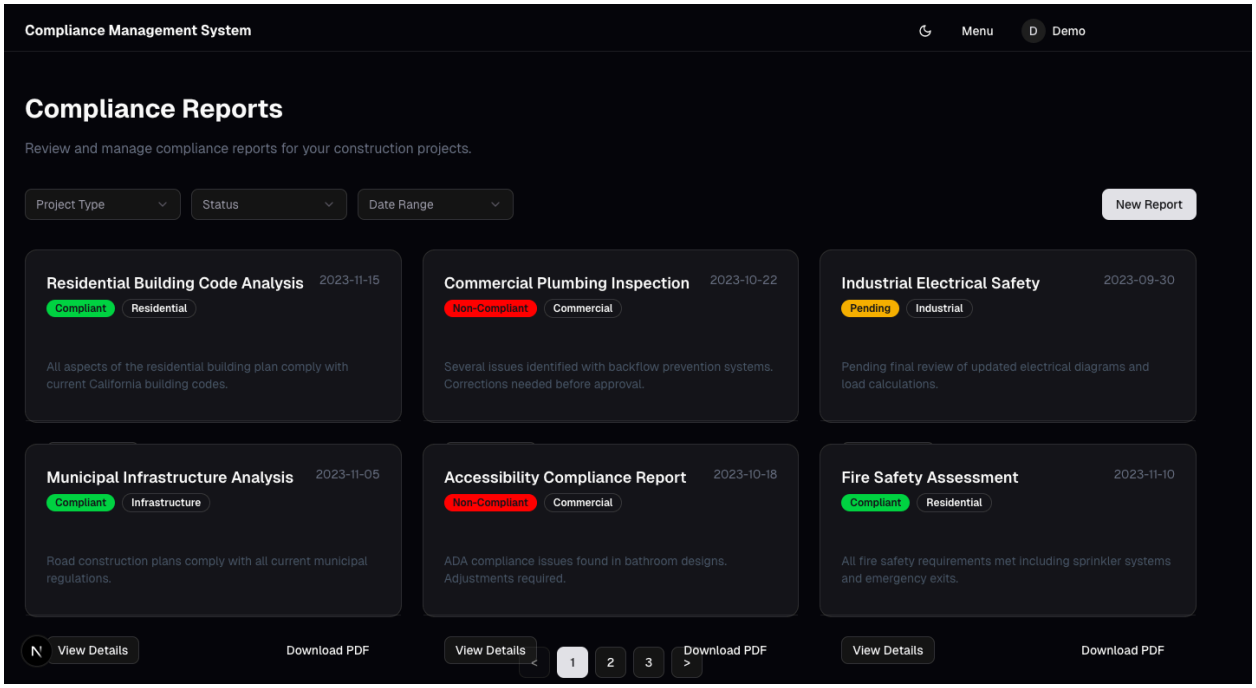

Compliance Management System
Menu
D
Demo
Construction Compliance Made Simple
Stay updated with real-time compliance information and regulations for the construction industry.
Get Started
Login
Register

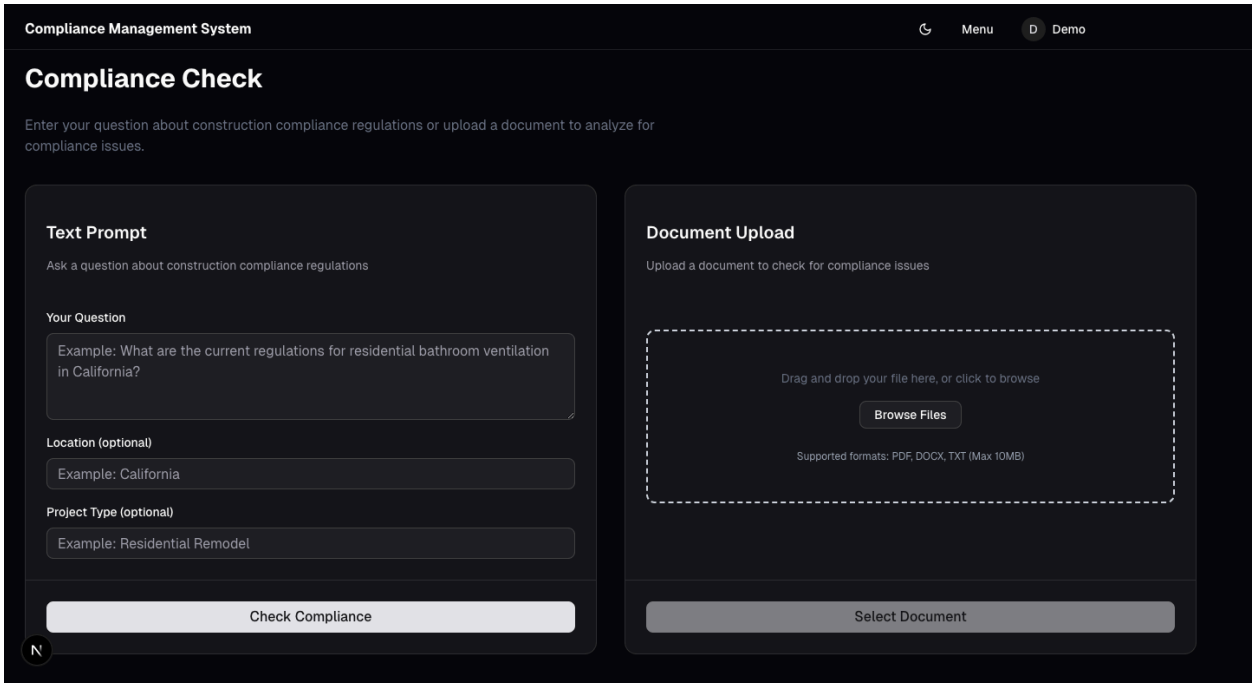

Compliance Management System
Menu
D
Demo
Real-Time Updates
Get real-time compliance information from government sources.
Our system connects directly to government databases and resources to provide the most current compliance information.
Learn More
Document Analysis
Upload and analyze your construction documents for compliance issues.
Our AI-powered document analysis tools can review your documents and highlight potential compliance concerns.
Upload Now
Compliance Reports
Generate detailed compliance reports for your projects.
Get comprehensive reports on compliance status, risks, and recommendations for your construction projects.
View Reports
© 2025 Compliance Management System. All rights reserved.
Privacy Policy
Terms of Service
Contact

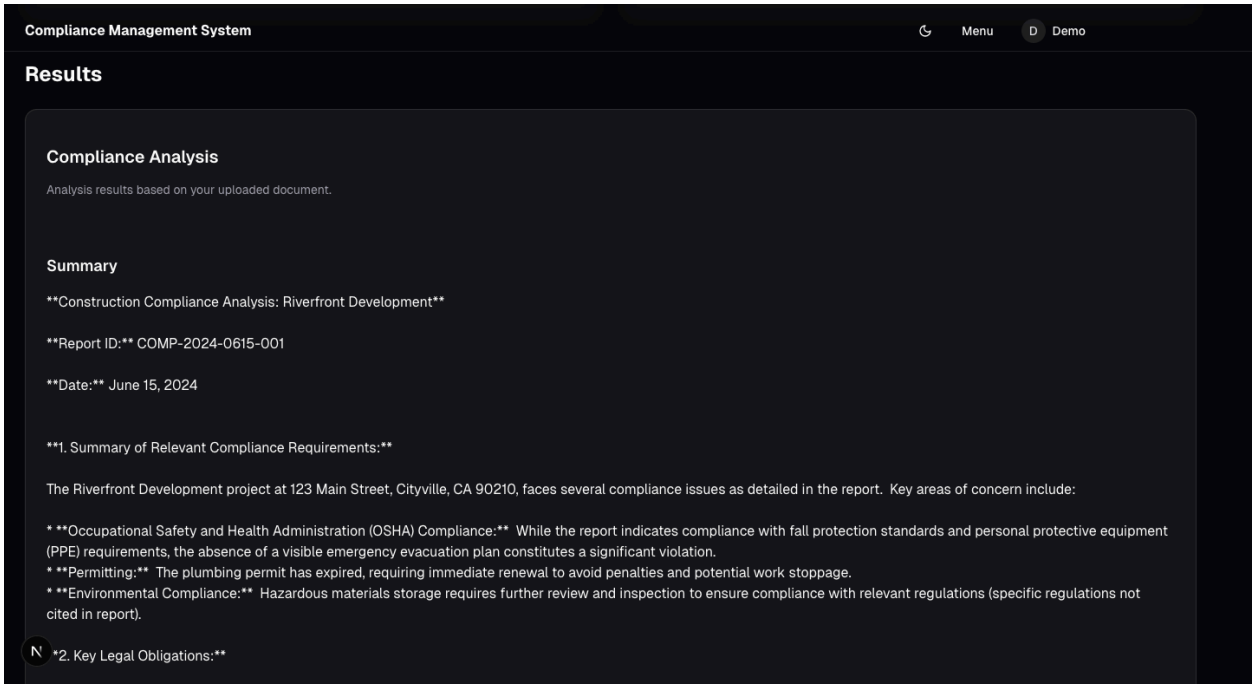

Legal Compliance
Login
Enter your email and password to sign in
Email
name@example.com
Please enter a valid email
Password
Forgot password?
Password must be at least 8 characters
Sign in
Don't have an account? Create an account
© 2025 Legal Compliance System. All rights reserved.


Compliance Management System
Menu
D
Demo
Dashboard
Overview of your compliance status and recent activity.
Demo User
srishtisahi42@gmail.com
Logout
Total Projects
24
+2 from last month
Compliant Projects
18
75% compliance rate
Pending Reviews
4
3 due this week
Compliance Issues
2
-1 from last month
Compliance Trends
Compliance trend chart will load here
Project Types
Project type distribution chart will load here


Compliance Management System
Menu
D
Demo
Compliance trend chart will load here
Project type distribution chart will load here
Recent Activity
D
Document Uploaded
Commercial Building Plans
2h ago
C
Compliance Check Completed
Residential Permit #R-2023-42
5h ago
N
New Regulation Added
California Building Code Update
1d ago
I
Issue Detected
Commercial Project #C-2023-15
2d ago
R
Report Generated
Monthly Compliance Summary
3d ago
Pending Tasks
Review commercial building plans
Start
Update residential compliance documentation
Start
Address ventilation issues in project #R-2023-38
Start
Generate quarterly compliance report
Start
© 2025 Compliance Management System. All rights reserved.
Privacy Policy
Terms of Service
Contact


Compliance Management System
Menu
D
Demo
Compliance Reports
Review and manage compliance reports for your construction projects.
Project Type
Status
Date Range
New Report
Residential Building Code Analysis
2023-11-15
Compliant
Residential
All aspects of the residential building plan comply with current California building codes.
Commercial Plumbing Inspection
2023-10-22
Non-Compliant
Commercial
Several issues identified with backflow prevention systems. Corrections needed before approval.
Industrial Electrical Safety
2023-09-30
Pending
Industrial
Pending final review of updated electrical diagrams and load calculations.
Municipal Infrastructure Analysis
2023-11-05
Compliant
Infrastructure
Road construction plans comply with all current municipal regulations.
Accessibility Compliance Report
2023-10-18
Non-Compliant
Commercial
ADA compliance issues found in bathroom designs. Adjustments required.
Fire Safety Assessment
2023-11-10
Compliant
Residential
All fire safety requirements met including sprinkler systems and emergency exits.
View Details
Download PDF
View Details
Download PDF
View Details
Download PDF
1
2
3


Compliance Management System
Menu
D
Demo
Compliance Check
Enter your question about construction compliance regulations or upload a document to analyze for compliance issues.
Text Prompt
Ask a question about construction compliance regulations
Your Question
Example: What are the current regulations for residential bathroom ventilation in California?
Location (optional)
Example: California
Project Type (optional)
Example: Residential Remodel
Check Compliance
Document Upload
Upload a document to check for compliance issues
Drag and drop your file here, or click to browse
Browse Files
Supported formats: PDF, DOCX, TXT (Max 10MB)
Select Document


Compliance Management System
Menu
D
Demo
Results
Compliance Analysis
Analysis results based on your uploaded document.
Summary
Construction Compliance Analysis: Riverfront Development
Report ID: COMP-2024-0615-001
Date: June 15, 2024
1. Summary of Relevant Compliance Requirements:
The Riverfront Development project at 123 Main Street, Cityville, CA 90210, faces several compliance issues as detailed in the report. Key areas of concern include:
* **Occupational Safety and Health Administration (OSHA) Compliance:** While the report indicates compliance with fall protection standards and personal protective equipment (PPE) requirements, the absence of a visible emergency evacuation plan constitutes a significant violation.
* **Permitting:** The plumbing permit has expired, requiring immediate renewal to avoid penalties and potential work stoppage.
* **Environmental Compliance:** Hazardous materials storage requires further review and inspection to ensure compliance with relevant regulations (specific regulations not cited in report).
*2. Key Legal Obligations:**