TECHNICAL PROJECT REPORT

Court Case Management System

A full-stack web application for managing legal cases, lawyers, and court schedules

DEVELOPER	VERSION	DATE	STACK
Chrishen Silva	1.0.0 (Web)	February 2026	React + Express + MySQL

REPOSITORY

github.com/chrishensilva

📋 1. Project Overview

The **Court Case Management System** is a modern, full-stack web application designed for law firms and court administrators to efficiently manage cases, clients, lawyers, court dates, and assignments. The system provides role-based access control, real-time data, PDF/CSV report generation, email notifications, and a comprehensive activity log.

Frontend Pages	**9**
API Endpoints	**16**
Database Tables	**6**
User Roles	**2**

Live Deployment: The application is hosted on Railway.app with cloud MySQL database on Aiven. The GitHub repository is at **github.com/chrishensilva/Court-Managment-System**.

🛠️ 2. Technology Stack

Layer	Technology	Version	Purpose
FRONTEND	React	19.2.0	UI Component Framework
FRONTEND	React Router DOM	7.9.6	Client-side routing (HashRouter)
FRONTEND	Recharts	3.6.0	Interactive charts on dashboard
FRONTEND	Vite	7.2.4	Build tool & dev server
BACKEND	Node.js	22.x	Runtime environment
BACKEND	Express	5.2.1	REST API server
BACKEND	mysql2	3.16.1	MySQL database driver
BACKEND	Nodemailer	8.0.1	Email notifications (Gmail)
BACKEND	dotenv	17.3.1	Environment variable management
DATABASE	MySQL	8.x	Primary data store (Aiven Cloud)
DEVOPS	Railway.app	—	Application hosting (free tier)
DEVOPS	Aiven	—	Managed MySQL cloud database
DEVOPS	GitHub	—	Version control & CI/CD

🏗 3. System Architecture

The application uses a **monolithic full-stack architecture** where the Express server serves both the REST API and the compiled React frontend as static files in production.

Request Flow

Browser → Railway (HTTPS) → Express Server (Port 5000) → mysql2 Driver → Aiven MySQL

Project Structure

```
Court-Managment-System/ ├── src/ │ ├── js/ │ │ ├── server.js # Express API server (entry point)
│ │ └── db.js # MySQL connection pool │ ├── App.jsx # Route definitions (HashRouter) │ ├──
main.jsx # React entry point │ ├── AuthContext.jsx # Authentication state │ ├──
ToastContext.jsx # Global toast/confirm system │ ├── config.js # Dynamic API base URL │ ├──
DashboardPage.jsx # Dashboard with stats │ ├── Lawyers.jsx # Lawyers list & delete │ ├──
UserPage.jsx # Cases list & manage │ ├── Cases.jsx # Assign lawyers to cases │ ├── AddUser.jsx
# Add new case record │ ├── AddLawyer.jsx # Add new lawyer │ ├── AddEditor.jsx # Manage editor
accounts │ ├── GenerateReport.jsx # Print / CSV export │ ├── ActivityLogs.jsx # System activity
log │ ├── Login.jsx # Login page │ └── Sidebar.jsx # Navigation sidebar ├── public/ # Static
assets ├── .env # Environment variables (local) ├── package.json └── vite.config.js
```

Environment Configuration

```
# .env (local) — not committed to GitHub DB_HOST=mysql-352d00d5-chrishensilva-
308b.h.aivencloud.com DB_USER=avnadmin DB_PASSWORD=*********** DB_NAME=defaultdb DB_PORT=10541
PORT=5000
```

🗄️ 4. Database Design

The database is hosted on **Aiven Managed MySQL** (cloud). All tables are automatically created on first server start via the `initMySQL()` function.

Table	Primary Key	Columns	Purpose
userdata	nic (VARCHAR)	name, email, number, address, lawyer1/2/3, note, last_date, next_date, casetype, status	Stores all case/client records
lawyerdata	nic (VARCHAR)	name, email, contact, note	Stores lawyer profiles
case_assignments	user_nic + lawyer_name	assigned_date	Tracks lawyer–case assignments
todo	id (INT AUTO_INCREMENT)	task, date, time	Dashboard to-do list
editors	id (INT AUTO_INCREMENT)	username, password, permissions	Sub-user accounts with permissions
activity_log	id (INT AUTO_INCREMENT)	username, action, details, timestamp	Full audit trail of all actions

Security Note: Passwords for editor accounts are currently stored in plain text. A future enhancement would be to hash passwords using `bcrypt` before storing.

🔌 5. API Endpoints

All API routes are prefixed with `/api` and served by the Express server on port **5000**.

Method	Endpoint	Description
POST	/api/login	Authenticate admin or editor user
GET	/api/getLawyers	Fetch all lawyers (with optional search)
POST	/api/addLawyer	Add a new lawyer record
POST	/api/deleteLawyer	Delete a lawyer by NIC
GET	/api/getUsers	Fetch all cases (with optional search)
GET	/api/getUserCases	Fetch cases with assigned lawyer (JOIN)
POST	/api/addUser	Add a new case record
POST	/api/deleteUser	Delete a case by NIC
POST	/api/assignLawyer	Assign a lawyer to a case + optional email
POST	/api/updateStatus	Update case status (ongoing/concluded/other)
GET	/api/getTodos	Fetch all to-do tasks
POST	/api/addTodo	Add a to-do task
POST	/api/deleteTodo	Delete a to-do task
GET	/api/dashboard_counts	Fetch total clients, lawyers, cases
GET	/api/getCaseStats	Case count grouped by type (for chart)
GET	/api/getReportData	All case data for report generation
GET	/api/getEditors	Fetch all editor accounts
POST	/api/addEditor	Create new editor account with permissions
POST	/api/deleteEditor	Remove an editor account by ID
GET	/api/getActivityLogs	Fetch last 500 activity log entries
POST	/api/logAction	Log a user action to the activity log

✨ 6. Application Features

6.1 Authentication & Role-Based Access Control

The system has two user roles. Sessions are persisted in `localStorage`.

Role	Access
ADMIN	Full access to all modules including editor management, activity logs, and delete operations
EDITOR	Configurable access — admin grants permissions per module (dashboard, lawyers, cases, assign, addlawyer, adduser, report)

6.2 Dashboard

- Live count cards for **Total Clients**, **Lawyers**, and **Cases**
- **Pie chart** (Recharts) showing cases by court type
- **To-Do App** — tasks with date/time, stored in MySQL

6.3 Case Management

- Add/delete case records with full client details (name, NIC, email, contact, address, lawyers, notes, dates)
- Status tracking: **Ongoing** / **Concluded** / **Other** with color-coded rows
- Search by case number or name

6.4 Lawyer Assignment

- Assign lawyers to cases via an inline dropdown on the Assign Cases page
- Optional **email notification** to the lawyer via Gmail (Nodemailer)
- Modern confirm dialog — Cancel aborts the entire assignment

6.5 Report Generation

- Filter cases by court type
- Summary statistics: total, assigned, unassigned, upcoming hearings (within 30 days)
- **Print to PDF** via browser print dialog
- **Export to CSV** with all case fields

6.6 Editor Management (Admin Only)

- Create sub-user accounts with granular permission control
- Delete editor accounts
- Permissions stored as JSON array in MySQL

6.7 Activity Logs (Admin Only)

- Every action (login, insert, delete, update, assign) is automatically logged
- Shows last 500 log entries sorted by timestamp

6.8 Modern Notification System

- Replaced all native `alert()` and `window.confirm()` calls
- **Toast Notifications**: slide-in from top-right, 4 types (success/error/warning/info), auto-dismiss with progress bar

- **Confirm Dialogs**: glassmorphism modal with dark theme, returns Promise for async/await usage

7. Application Pages & Routes

Route	Component	Access	Description
/login	Login.jsx	Public	Username/password login form
/	DashboardPage.jsx	dashboard perm	Stats cards, chart, to-do
/lawyers	Lawyers.jsx	lawyers perm	List & delete lawyers
/clients	UserPage.jsx	cases perm	View & manage case records
/cases	Cases.jsx	assign perm	Assign lawyers to cases
/adduser	AddUser.jsx	adduser perm	Add new case/client record
/addlawyer	AddLawyer.jsx	addlawyer perm	Add new lawyer
/report	GenerateReport.jsx	report perm	View, filter, print, export cases
/addeditor	AddEditor.jsx	Admin only	Create/delete editor accounts
/logs	ActivityLogs.jsx	Admin only	View full audit log

8. Deployment Architecture

Railway.app (Backend + Frontend)

- **Build Command:** `npm install && npm run build`
- **Start Command:** `node src/js/server.js`
- In production (`NODE_ENV=production`), Express serves the compiled `/dist` folder as static files
- A catch-all `app.use()` middleware serves `index.html` for all non-API routes (SPA support)

Aiven (MySQL Database)

- Managed cloud MySQL with SSL connection on port **10541**
- Connection details passed via Railway environment variables
- Tables auto-created on startup if they don't exist

GitHub (CI/CD)

- Repository: **github.com/chrishensilva/Court-Managment-System**
- Every push to `main` triggers automatic redeploy on Railway
- `.env`, `dist/`, `dist_electron/`, `node_modules/` excluded via `.gitignore`

💡 9. Key Design Decisions

Decision	Rationale
HashRouter instead of BrowserRouter	Works on static file hosts without server-side routing configuration
Monolithic server (API + static files)	Simplifies deployment — only one Railway service needed (free tier)
`app.use()` for SPA catch-all	Express v5 changed wildcard route syntax; `app.use()` avoids path-to-regexp entirely
MySQL connection pool	Reuses connections for better performance under load; 10-connection limit
Environment variables via dotenv	Keeps credentials out of source code; compatible with Railway's variable injection
Toast Context (no library)	Avoids adding dependencies; gives full control over styling and behavior
`import.meta.env.PROD` for API URL	Automatically uses relative `/api` in production and `localhost:5000` in dev

🔮 10. Future Improvements

- **Password hashing:** Implement `bcrypt` for editor account passwords
- **JWT Authentication:** Replace localStorage sessions with secure JWT tokens
- **Pagination:** Add server-side pagination for large datasets
- **File Uploads:** Attach court documents (PDFs) to case records
- **Calendar View:** Visual calendar for upcoming court dates
- **Real-time Updates:** Use WebSockets or Server-Sent Events for live data
- **Mobile App:** React Native version using the same Express API
- **Audit Improvements:** IP address logging in activity log
- **Multi-language Support:** i18n for Sinhala/Tamil interface

⚠️ 11. Known Issues & Limitations

- Passwords for editor accounts are stored in plain text (no hashing)
- Admin password (`admin/123`) is hardcoded in `server.js`
- No rate limiting on API endpoints — could be vulnerable to brute force
- First load on Railway may be slow (cold start on free tier)
- Email credentials (Gmail App Password) are stored in environment variables — should be rotated periodically

Court Case Management System — Project Report v1.0

Developed by Chrishen Silva | February 2026 | github.com/chrishensilva/Court-Managment-System