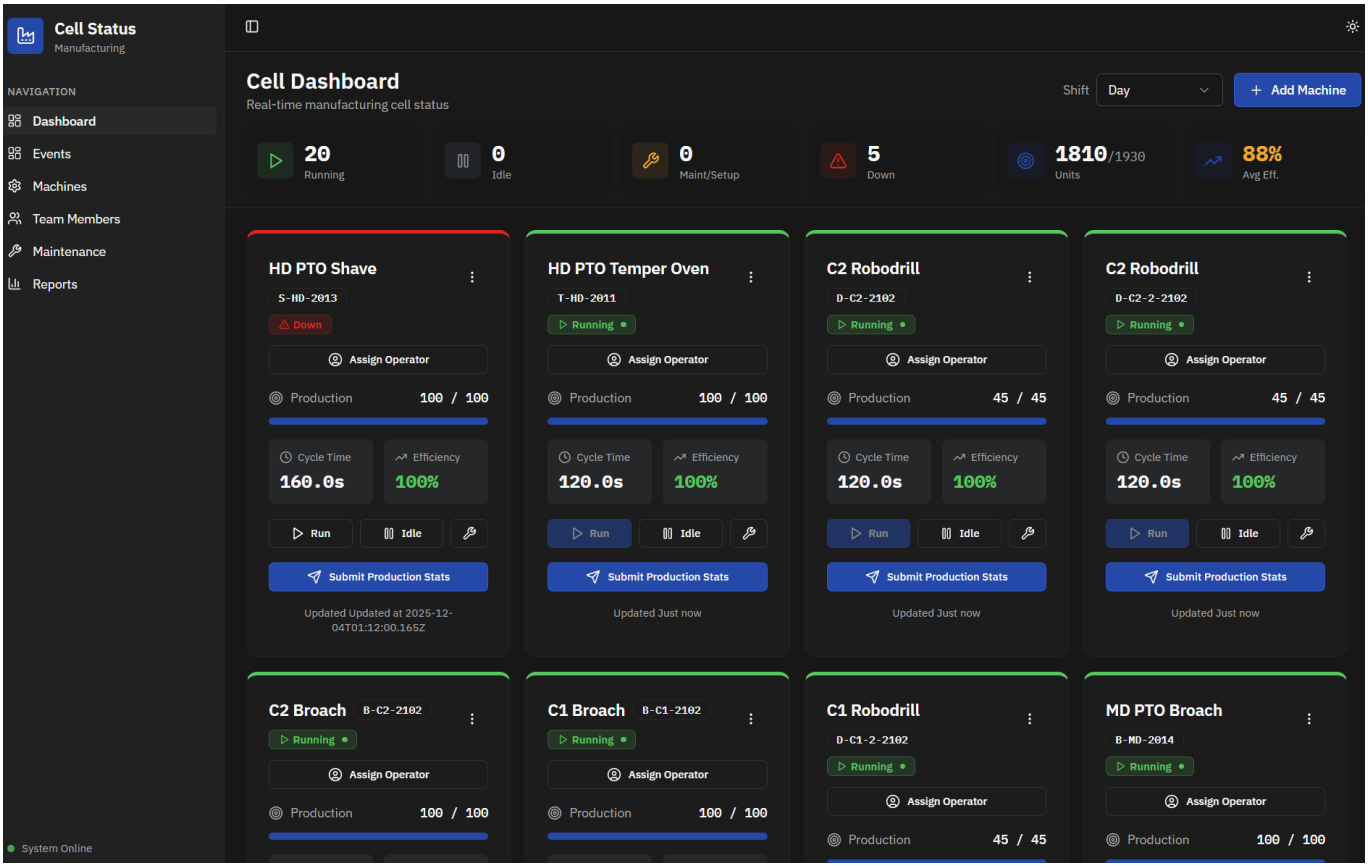
CellStatus - Manufacturing Cell Tracker

A modern, real-time manufacturing cell status tracking application built for production floor teams. Monitor machines, track production metrics, manage operator assignments, and log downtime incidents—all in one intuitive dashboard.

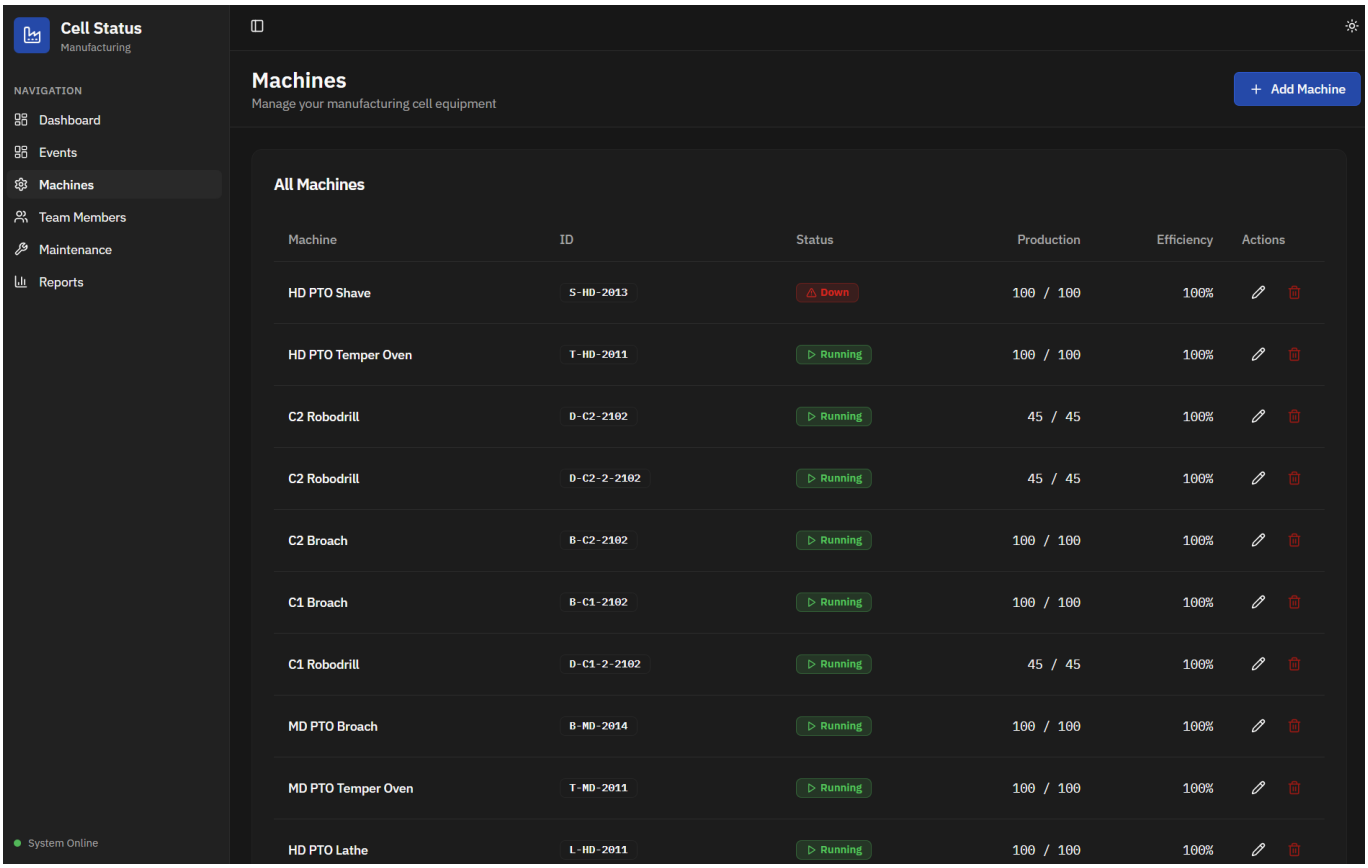
Screenshots

Dashboard - Real-Time Cell Overview



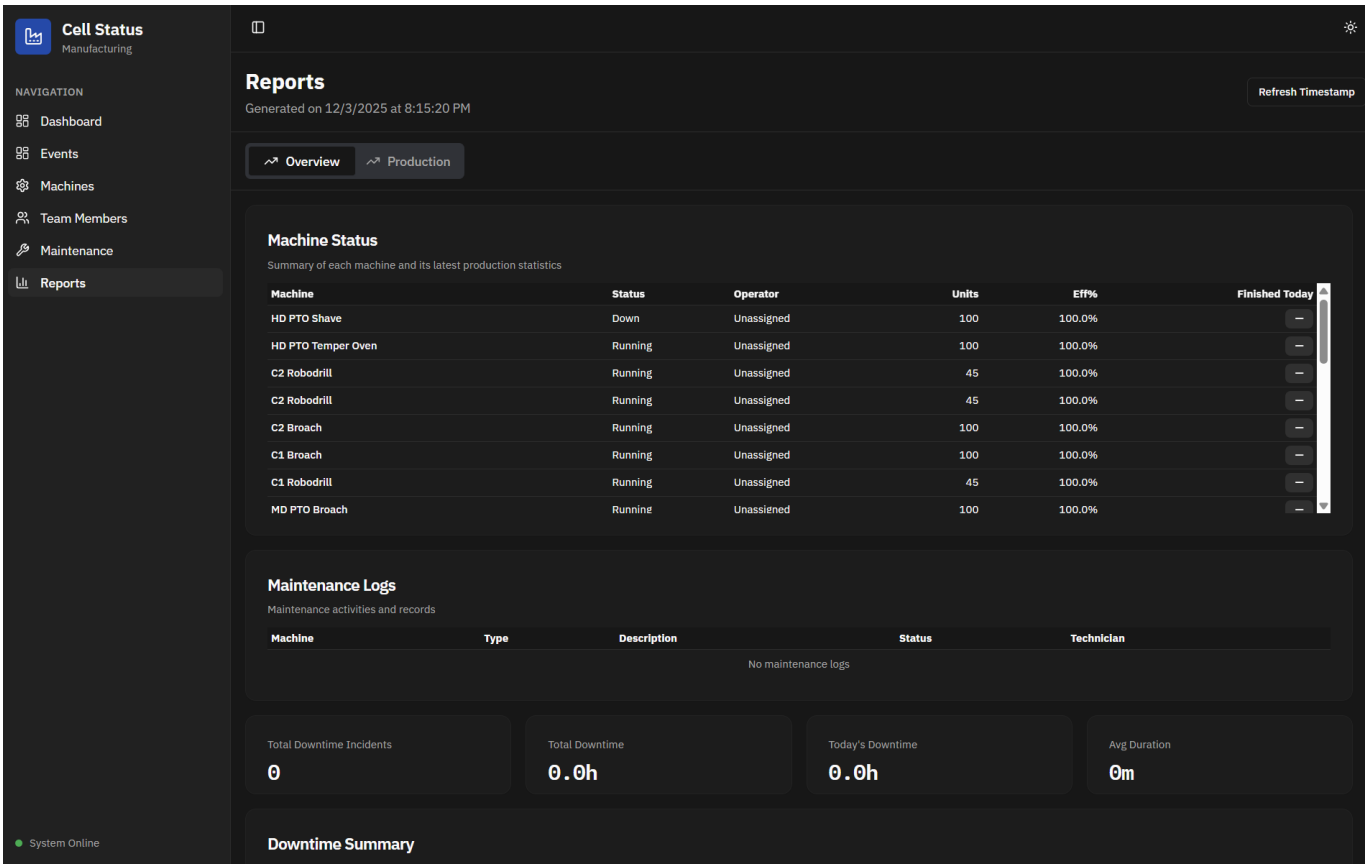
Live dashboard showing all machines with status cards, production stats, and shift selection

Machines Management



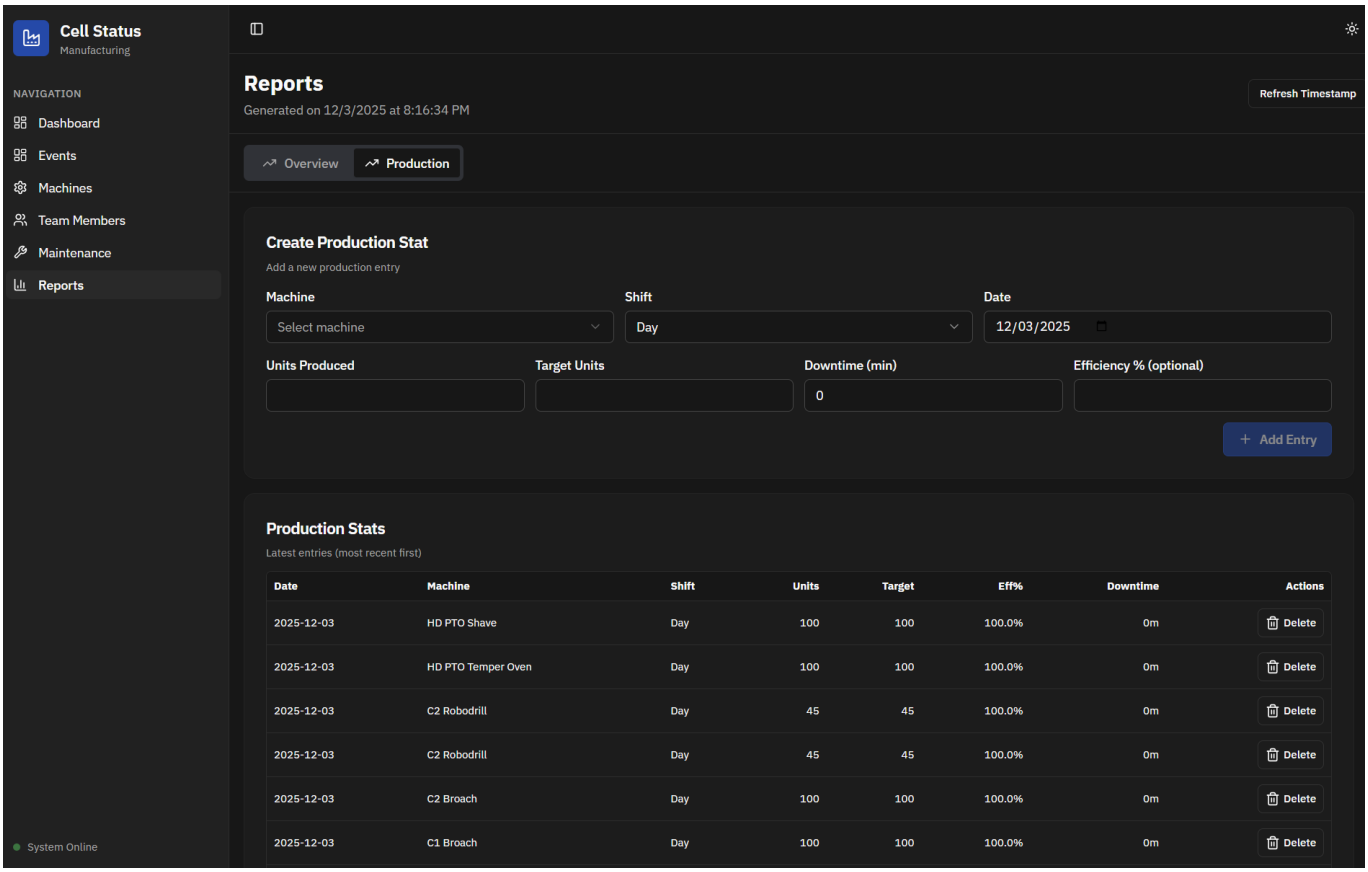
Manage your manufacturing equipment with detailed status tracking

Reports & Analytics



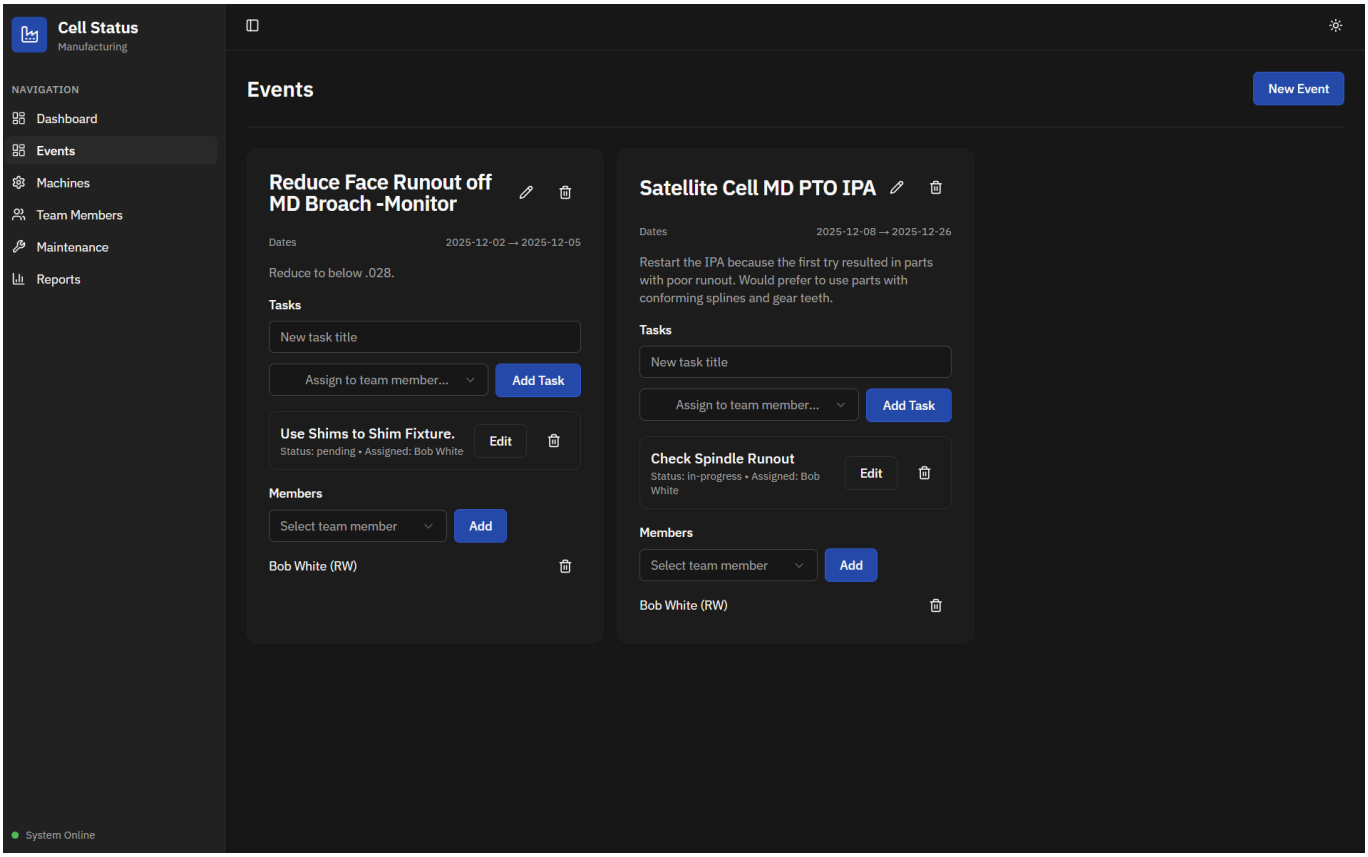
Comprehensive production reports with machine history, maintenance logs, and downtime analysis

Production Tracking



Log and review production statistics by shift and date

Event Cards and Team Collaboration



Document events and assign tasks for team collaboration

✨ Key Features

🎯 Dashboard

- **Live Status Cards**: Color-coded machine status at a glance (Running, Idle, Down, Maintenance, Setup)
- **Shift Management**: Track production across Day, Afternoon, and Midnight shifts
- **Summary Metrics**: Instant view of total running/idle/down machines, units produced, and average efficiency
- **Active Downtime Tracking**: See live downtime duration for machines currently down

📆 Events & Team Collaboration

- **Event Creation**: Create events to organize team efforts around specific issues or projects
- **Task Management**: Break down events into actionable tasks with:
  - Task titles and descriptions
  - Start and end dates for scheduling
  - Status tracking (Not Started, In Progress, Completed, Blocked)
  - Assignee selection from your operator roster
- **Team Assignments**: Add multiple team members (operators) to events for coordinated collaboration
- **Issue Tracking**: Link events to specific problems or improvements, keeping all related tasks and team members organized in one place
- **Timeline View**: Track task progress with clear start/end dates and status indicators
- **Collaborative Workflow**: Enable cross-functional teams to work together on maintenance issues, improvement projects, or production challenges

🏭 Machine Management

- **Machine Cards**: Visual cards showing:
  - Current status with color indicators
  - Assigned operator and shift
  - Units produced vs. target
  - Real-time efficiency percentage
  - Active downtime alerts
- **Quick Actions**:
  - Change machine status with one click
  - Assign/reassign operators
  - Log maintenance activities
  - Record downtime incidents
  - Submit production stats
  - Resolve active downtime

📊 Production Statistics

- **Daily Tracking**: Log units produced, targets, downtime, and efficiency per shift
- **Automatic Calculations**: Efficiency computed from production data
- **Historical View**: Review past production performance by machine and date
- **Operator Attribution**: Production stats automatically linked to current machine operator

🔧 Maintenance Logging

- **Maintenance Types**: Preventive, Corrective, Emergency, and Inspection

- **Status Tracking**: Scheduled, In Progress, and Completed
- **Technician Assignment**: Track who performed each maintenance task
- **History**: Full maintenance records per machine

⏱️ Downtime Management

- **Reason Categorization**: Log downtime with specific reason codes:
  - **Mechanical**: Equipment Failure, Hydraulic Issue, Pneumatic Issue, Bearing Failure, Lubrication Issue
  - **Electrical**: Motor Failure, Sensor Malfunction, Control System Error, Power Supply Issue, Wiring Problem
  - **Material**: Material Shortage, Wrong Material, Material Defect, Loading Issue, Feed Problem
  - **Operator**: Break Time, Training, Shift Change, Absence, Setup Time
  - **Quality**: Quality Check, Rework Required, Calibration, Inspection, Cleaning
  - **Other**: Unplanned Downtime, Emergency Stop, Other
- **Duration Tracking**: Automatic calculation of downtime duration in minutes
- **Active Alerts**: Real-time indicators for machines currently experiencing downtime
- **Resolution Logging**: Record who resolved each incident and any notes

📈 Reports & Analytics

- **Overview Tab**:
  - Machine Status summary table
  - Maintenance logs overview
  - Machine history with production stats and maintenance records
  - Downtime summary with total incidents, duration, and per-incident details
- **Production Tab**:
  - Create new production entries
  - View all historical production stats
  - Delete individual entries
- **Downtime Analysis** (shown in Overview):
  - Total downtime incidents count
  - Total downtime hours
  - Today's downtime
  - Average incident duration
  - Per-row delete actions for downtime logs

🎨 Modern UI/UX

- **Responsive Design**: Works seamlessly on desktop, tablet, and mobile
- **Dark Mode**: Toggle between light and dark themes for 24/7 operation
- **Accessible**: Built with Radix UI primitives for keyboard navigation and screen readers
- **Intuitive**: Clean, industrial-themed design with semantic color coding

👥 Operator Management

- **Operator Database**: Maintain a list of operators with names, shifts, and availability
- **Assignment Tracking**: See which operator is running each machine

- **Shift-Based Views**: Filter and track by shift (Day, Afternoon, Midnight)

🚀 Getting Started

Prerequisites

- **Node.js** 20 or higher
- **PostgreSQL** database (free tier available at Neon.tech)

Installation

1. **Clone the repository**

```
git clone https://github.com/rwaynewhite15/CellStatus.git
cd CellStatus
```

2. **Install dependencies**

```
npm install
```

3. **Set up environment variables**

   Create a `.env` file in the root directory:

```
DATABASE_URL=postgresql://user:password@host/database
SESSION_SECRET=your-random-secret-here
```

4. **Initialize the database**

```
npm run db:push
```

5. **Start development server**

```
npm run dev
```

   Access the app at `http://localhost:5000`

🌐 Live Demo

Frontend: https://rwaynewhite15.github.io/CellStatus/
Backend API: Hosted on Render (serverless)

Tech Stack

Frontend

- React 18 + TypeScript
- Vite (fast build tool)
- TanStack Query (data fetching and caching)
- Shadcn UI + Radix UI (accessible components)
- Tailwind CSS (utility-first styling)
- Lucide React (beautiful icons)
- Recharts (production charts)

Backend

- Node.js + Express
- TypeScript
- Drizzle ORM (type-safe database queries)
- PostgreSQL (Neon serverless)
- CORS & Rate Limiting (security)

Deployment

- Frontend: GitHub Pages (static hosting)
- Backend: Render (Node.js service)
- Database: Neon (serverless PostgreSQL)

Security Features

- ☑ **CORS Protection**: Whitelist-based origin control
- ☑ **Rate Limiting**: 100 requests per 15 minutes per IP
- ☑ **Environment Isolation**: Secure credential management
- ☑ **SQL Injection Protection**: Parameterized queries via Drizzle ORM
- ☑ **HTTPS Enforced**: Secure communication in production
- ☑ **No-Cache Headers**: Prevent stale data issues

License

MIT License - see LICENSE file for details

Acknowledgments

- Built with Shadcn UI for beautiful, accessible components
- Icons by Lucide

- Database by Neon
- Hosted on Render and GitHub Pages

📧 Support

For issues, feature requests, or questions:

- Open an issue on GitHub Issues
- Check existing issues for solutions

Built for manufacturing teams to track production in real-time 🏭