

# Agenda 2M

A comprehensive appointment and order management system for automotive workshops, built with Next.js and Supabase.

## Overview

Agenda 2M is a web application designed to streamline workshop operations by managing appointments, purchase orders, payments, and pending tasks. The system provides an intuitive interface for coordinating technician schedules and tracking customer service requests.

## Tech Stack

- **Next.js 14+** - React framework with App Router
- **TypeScript** - Type-safe development
- **Supabase** - Backend as a Service (Database & Authentication)
- **Tailwind CSS** - Utility-first CSS framework
- **React DatePicker** - Date selection component
- **Lucide React** - Icon library

## Prerequisites

- Node.js 18 or higher
- npm or yarn package manager
- Supabase account (free tier available)

## Installation

1. Clone the repository:

```
git clone <repository-url>
cd agenda-2m
```

2. Install dependencies:

```
npm install
```

3. Configure environment variables:

Create a `.env.local` file in the project root:

```
NEXT_PUBLIC_SUPABASE_URL=your_supabase_url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your_supabase_anon_key
```

Get these credentials from your Supabase project dashboard at:

[https://app.supabase.com/project/\\_/settings/api](https://app.supabase.com/project/_/settings/api)

## Running the Project

### Development Mode

```
npm run dev
```

Open <http://localhost:3000> in your browser.

## Production Build

```
# Build the application  
npm run build  
  
# Start production server  
npm run start
```

## Features

### Appointment Scheduler (Agenda)

- **Interactive Calendar:** View and manage appointments by date
- **Time Slots:** 30-minute intervals from 8:00 AM to 5:30 PM
- **Technician Assignment:** Assign appointments to specific team members (Botaguas, Keilor, Andrey, Dylan, Kenneth)
- **Status Tracking:** Visual color coding for appointment states
  - Red: Pending
  - Yellow: In Progress
  - Green: Completed
- **Customer Information:** Track customer name, phone, vehicle details, and service description
- **Real-time Updates:** Instant synchronization across all users

### Order Management

- **Purchase Order Creation:** Register new orders with customer and product details
- **Payment Tracking:** Monitor initial payments and remaining balances
- **Provider Information:** Track supplier details for each order
- **Order History:** View all orders sorted by date
- **Financial Overview:** Calculate totals and remaining amounts automatically

### Payment Tracking

- **Payment Records:** Register and monitor all incoming payments
- **Customer Association:** Link payments to specific customers
- **Date Tracking:** Organize payments by date
- **Amount Recording:** Track payment amounts and methods

### Pending Tasks

- **Quote Management:** Keep track of pending quotes
- **Task List:** Organize pending work and follow-ups
- **Status Updates:** Mark tasks as completed or in progress

### Authentication & Security

- **Supabase Auth:** Secure login system
- **Session Management:** Persistent user sessions
- **Role-based Access:** Admin and user permissions (where applicable)

## Project Structure

```
agenda-2m/  
├── app/  
│   ├── dashboard/  
│   │   ├── agenda/      # Appointment scheduler  
│   │   ├── orders/     # Purchase orders management
```

```
|   |   └── payments/    # Payment tracking
|   |   └── tasks/       # Pending tasks list
|   └── components/     # Reusable React components
|   └── utils/          # Utility functions
|   └── supabase/        # Supabase client configuration
└── public/            # Static assets
```

## User Workflow

1. **Login:** Access the system through Supabase authentication
2. **Dashboard:** Navigate to different sections via the sidebar menu
3. **Create Appointments:** Click on time slots in the agenda to create new appointments
4. **Manage Orders:** Add, edit, or delete purchase orders
5. **Track Payments:** Record incoming payments from customers
6. **Monitor Tasks:** Keep track of pending quotes and work items

## Deployment

The application can be easily deployed to Vercel:

1. Connect your GitHub repository to Vercel
2. Configure environment variables in Vercel dashboard
3. Deploy with automatic CI/CD

## Support

For issues, bugs, or feature requests, please create an issue in the repository.