CRM System Guide

Complete Technical Documentation

Traincape Technology
7/11/2025

Table of Contents

1. System Overview

Our CRM system consists of the following main modules:

- Authentication System (Login/Register)
- Employee Management
- Payroll System
- Chat System

2. Page Connectivity

Frontend Pages and Their Connections:

Login Flow:

LoginPage.jsx
!“
AuthContext.jsx (Manages authentication state)
!“
DashboardPage.jsx (After successful login)

Employee Management Flow:

EmployeePage.jsx
% % % EmployeeList.jsx (Shows all employees)
% % % EmployeeForm.jsx (Add/Edit employee)
% % % EmployeeDetail.jsx (View employee details)

Payroll System Flow:

PayrollPage.jsx
% % % PayrollList.jsx (All payroll records)
% % % PayrollGenerate.jsx (Generate new payroll)
% % % PayrollDetail.jsx (View/Download slip)

3. API Flow

Authentication APIs:

POST /api/auth/login
% % % auth.controller.js !’ validateUser() !’ generateToken()

POST /api/auth/register
% % % auth.controller.js !’ createUser() !’ sendWelcomeEmail()

Employee APIs:

GET /api/employees
% % % employee.controller.js !’ findAll() !’ paginateResults()

POST /api/employees
% % % employee.controller.js !’ create() !’ validateData()

Payroll APIs:

POST /api/payroll/generate
% % % payroll.controller.js !’ calculateSalary() !’ generatePDF()

GET /api/payroll/:id/salary-slip
% % % payroll.controller.js !’ findPayroll() !’ createPDF()

4. Database Schema

User Schema:

```
const userSchema = {
  email: String (required, unique),
  password: String (hashed),
  role: String (Admin/HR/Manager/Employee),
  lastLogin: Date,
  status: String (Active/Inactive)
}
```

Employee Schema:

```
const employeeSchema = {
  userId: ObjectId (ref: User),
  fullName: String,
  department: String,
  designation: String,
  salary: Number
}
```

Payroll Schema:

```
const payrollSchema = {
  employeeId: ObjectId (ref: Employee),
  month: Number (1-12),
  year: Number,
  baseSalary: Number,
  daysPresent: Number,
  calculatedSalary: Number
}
```

5. Detailed Component Explanations

PayrollComponent Workflow:

1. User Input
   - Enter base salary
   - Enter days present
   - Select month/year

2. Calculation
   - Always use 30 days as base
   - Formula: (baseSalary / 30) * daysPresent
   - Example: (30000 / 30) * 25 = 25000

3. PDF Generation
   - Size limit: 10KB - 20KB
   - Includes company logo
   - Shows salary breakdown

ChatComponent Workflow:

1. Message Flow
   - User types message
   - Socket.IO emits to server
   - Server broadcasts to recipients
   - Recipients play notification

2. Notification System
   - WhatsApp-style sounds
   - Browser notifications
   - Unread message count

6. Code Examples

Salary Calculation:

```
// Frontend calculation
const calculateSalary = (baseSalary, daysPresent) => {
  return (baseSalary / 30) * daysPresent;
};

// Example usage
const salary = calculateSalary(30000, 25);
// Result: 25000 (for 25 days present)
```

PDF Generation:

```
const generatePDF = (data) => {
  const doc = new PDFDocument();
  let size = 0;
  doc.on("data", chunk => {
    size += chunk.length;
    if (size > 20480) {
      throw new Error("PDF too large");
    }
  });
};
```

Generated: 7/11/2025, 1:51:11 PM
Traincape Technology CRM System