

PROJECT DOCUMENTATION FORMAT

CRM Application for Jewellery Management

Full Stack Developer Report – MERN Stack

1. Introduction

- **Project Title:** CRM Application for Jewellery Management
 - **Team Members:**
 - Munneppa Chennappa – Backend Developer
 - M Prakash –Developer
 - C Supriya – UI/UX Designer
 - A Manhu – Database & API Integration
 - G.MADAN KUMAR – Frontend Developer
-

2. Project Overview

- **Purpose:**

This CRM system is designed to streamline and digitize customer relationship management in jewellery businesses, covering customer profiles, purchase history, inventory, repair tracking, and follow-ups.
 - **Features:**
 - Customer registration & contact management
 - Jewellery item catalog with categories
 - Sales order & invoice generation
 - Service/repair tracking
 - Inventory management
 - Reminder and follow-up system (SMS/email)
 - Admin dashboard with analytics
-

3. Architecture

- **Frontend:**

Built using **React.js** with component-based architecture. Implements Redux for state management and Axios for API calls.
- **Backend:**

Developed with **Node.js** and **Express.js**, featuring RESTful APIs for CRUD operations and JWT-based authentication.

PROJECT DOCUMENTATION FORMAT

- **Database:**

Uses **MongoDB** to store:

- Customer info
 - Jewellery inventory
 - Transactions
 - Service records
-

4. Setup Instructions

- **Prerequisites:**

- Node.js (v18+)
- MongoDB (Atlas/local)
- Git

- **Installation:**

bash

CopyEdit

```
git clone https://github.com/your-repo/jewellery-crm.git
```

```
cd jewellery-crm
```

```
npm install
```

```
cd client
```

```
npm install
```

- **Environment:**

Create .env files in both client and server with API keys, Mongo URI, and JWT secrets.

5. Folder Structure

- **Client (React):**

bash

CopyEdit

```
/client
```

```
  /src
```

```
    /components
```

```
    /pages
```

```
    /redux
```

PROJECT DOCUMENTATION FORMAT

/services

- **Server (Node.js):**

bash

CopyEdit

/server

/routes

/controllers

/models

/middleware

6. Running the Application

- **Frontend:**

bash

CopyEdit

cd client

npm start

- **Backend:**

bash

CopyEdit

cd server

npm start

7. API Documentation

Endpoint	Method	Description
/api/customers	GET	Fetch all customers
/api/customers/:id	PUT	Update a customer
/api/inventory	POST	Add new jewellery item
/api/orders	POST	Create new sales order
/api/auth/login	POST	User login
/api/auth/register	POST	Register admin/staff

PROJECT DOCUMENTATION FORMAT

8. Authentication

- JWT token-based authentication
 - Secure login for admin and staff
 - Role-based access control
-

9. User Interface

Includes:

- Dashboard
- Customer Management View
- Inventory List with Filter
- Order/Invoice Generator
- Calendar Reminders

(Screenshots can be added here)

10. Testing

- **Tools Used:** Jest, Postman
 - **Strategy:** Unit testing (backend), manual testing (frontend)
-

11. Screenshots or Demo

(Add links to screenshots or a hosted demo here if available)

12. Known Issues

- No image compression on upload
 - Reminder notifications depend on manual cron job or third-party scheduler
-

13. Future Enhancements

- Multi-store support
- WhatsApp integration for reminders
- AI-based customer behavior analysis
- Mobile app version (React Native)

PROJECT DOCUMENTATION FORMAT