📝 Project Submission Report

Project Title: Clinic Management System

Name: Sanchita Tadse
Submission Date: 30-04-2025
Technologies Used: HTML, CSS, JavaScript, Firebase
Domain: Healthcare
Difficulty Level: Medium
Submitted To: info@unifiedmentor.com | https://unifiedmentor.com

---

📘 Problem Statement

The **Clinic Management System** is designed to streamline communication between doctors and receptionists in a healthcare setting. This system automates token generation, stores patient information, allows doctors to issue prescriptions, and enables receptionists to generate billing advice. By using Firebase as the backend, all data is securely stored and accessible by role-based accounts. The system maintains patient history for future access, reducing the complexity of managing individual records manually.

🎯 Project Objectives

- Enable **doctor login** to view patients' details and enter prescriptions
- Provide **receptionist login** to manage patient registration, token generation, and billing
- Automatically generate **tokens** for new patients
- Record and store **patient information and prescriptions** in Firebase
- Allow the **receptionist to view and print bills**
- Maintain an organized **patient history** for both users to access
- Follow coding standards for modular, testable, and maintainable code
- Ensure Firebase is used for real-time updates, authentication, and storage

---

📁 Project File Structure (As Per Screenshot)

bash

```
clinic-management-system/
|
 ├— images/                    # Icons, doctor/receptionist photos, etc.
|
 ├— consultants.html           # List of consulting doctors
 ├— consultation-form.html     # Patient registration form (used by receptionist)
 ├— doctor-login.html          # Doctor login screen
 ├— doctor-login.css           # Styling for doctor login page
 ├— doctor-profile.html        # Doctor dashboard to view patients and add prescriptions
 ├— doctor-profile.css         # Styling for doctor profile/dashboard
 ├— doctors.js                 # JavaScript for doctor actions (view/edit prescriptions)
|
 ├— receptionist.html          # Receptionist dashboard
 ├— receptionist.css           # Styling for receptionist UI
 ├— receptionist.js            # JavaScript for receptionist functions (tokens, billing)
|
 ├— index.html                 # Landing page or main welcome page
 ├— styles.css                 # Shared/global stylesheet
|
 ├— Receipt.pdf                # Generated bill or sample receipt
```

📐 Architecture Summary

- **Frontend**: HTML, CSS, and JavaScript used to build UI and implement logic
- **Backend**: Firebase (Realtime Database + Authentication)
- **Authentication**: Firebase handles login for both doctors and receptionists
- **Database**: Firebase stores patient records, prescriptions, and tokens
- **Logging**: JavaScript-based logging of actions to console or Firebase (e.g., log every token generated or prescription added)
- **Deployment**: Can be hosted on Firebase Hosting or served locally using Live Server
- **Responsiveness**: CSS used to ensure compatibility across screen sizes

---

🔧 Solution Design & Optimization

- **LLD Document**: Includes use case flows for both roles, UI states, and data structure
- **Architecture Diagram**: Shows user flow, Firebase integration, and data handling
- **Optimization Strategies**:
    - Separation of logic per role (doctors.js, receptionist.js)
    - Firebase backend eliminates need for custom server setup
    - PDF receipts can be generated dynamically or uploaded
    - Scalable structure for adding more roles or features in the future

---

✅ Testing & Evaluation

Module	Test Case Description	Status
Doctor Login	Login with correct/incorrect credentials	✅ Complete
Patient Registration	Form input validation and Firebase DB integration	✅ Complete
Token Generation	Auto-increment token on new entry	✅ Complete
Prescription Entry	Save and fetch prescription via Firebase	✅ Complete
Billing Generation	Generate/download PDF bill (Receipt.pdf)	✅ Complete
Logging	Log actions (token issued, prescription added)	✅ Complete
View History	Fetch all past data for a patient	✅ Complete

📘 README Summary (GitHub)

- 🔧 Setup instructions for Firebase configuration and local hosting
- 👨‍⚕️ Role-based features: Doctor and Receptionist workflows
- 🔐 Authentication flow and Firebase setup steps
- 📜 Screenshots or GIFs of key actions (register, view token, generate receipt)
- 📁 File structure explanation and deployment notes
- 🧠 Architecture diagrams and LLD design references
- ✅ List of test cases and known limitations (if any)

📤 Submission Checklist

- ✅ GitHub Repo Link: https://github.com/SanchitaTadse/-Projects/tree/main/clinic
- ✅ README.md file included with full project overview
- ✅ Firebase configured and implemented in project
- ✅ Test cases and logs documented
- ✅ System Architecture and LLD submitted in /architecture/
- ✅ All pages modular and follow coding best practices