1.Overview

This Clinic Management System (CMS) is a full-stack application for managing patients, doctors, appointments, schedules, prescriptions, medical records, billing, and staff. It consists of:
• A PostgreSQL database named clinic_db

• A Node.js/Express backend serving RESTful APIs on port 5000

• A React frontend running on port 3000

2. Prerequisites

Before you begin, make sure you have installed: • Node.js (v14+ with npm) → https://nodejs.org/ • PostgreSQL (v12+) → https://www.postgresql.org/ - Command-line tools: psql, pg_dump, createdb, dropdb • Git (optional, for cloning)

3. Database Setup

Ensure PostgreSQL service is running on your machine.

Create the database (run once):
createdb -U postgres -h localhost clinic_db (If it already exists, skip this step.)

Load the SQL dump into clinic_db:
psql -U postgres -h localhost -d clinic_db -f clinic_db.sql (You should see messages like CREATE TABLE and INSERT 0 N without errors)

DATABASE Troubleshooting

`createdb` or `psql` not recognized
• Add your PostgreSQL `bin/` folder to your PATH, or
• Open the “PostgreSQL Command Prompt” that ships with your install.

Authentication / password failures
• Double-check you’re using the right Postgres password.
• Try logging in interactively:

```
psql -U postgres -h localhost

# then inside psql:
CREATE DATABASE clinic_db;
\q
```

- **“permission denied” or access errors**
  • Make sure your OS user has rights, or use a superuser account.

• On Windows you may need to launch your terminal as Administrator.

- `psql: error reading file “clinic_db.sql”`
  • Confirm you ran the command from the folder where `clinic_db.sql` lives.
  • Check that the file isn’t locked or in use by another process.

- **Encoding/corruption warnings on Windows**
  • If you see “code page” warnings, run `chcp 1252` before invoking `psql`
  • Or add `--encoding=UTF8` to your import command:

```
 psql -U postgres -h localhost -d clinic_db \
--encoding=UTF8 -f clinic_db.sql
```

With that in place, no extra magic is needed—just that one `psql −f clinic_db.sql` command (after the optional `createdb`). Everything else in your `.env`, backend and frontend will pick up the freshly loaded `clinic_db` automatically.

4. Backend Setup

We have a file named .env in the project **root/backend(folder)** with this content:
DATABASE_URL=postgres://postgres:admin@localhost:5432/clinic_db PORT=5000
(Use your actual Postgres password in place of admin.)

Install dependencies and start the server (NEW TERMINAL):

First Run : cd backend
Then : npm install
Then: npm run dev

The API is now running at http://localhost:5000 Test with:
curl http://localhost:5000/api/patients

5.Frontend Setup

Install dependencies and start React: (NEW TERMINAL)
First Run: cd frontend
Then: npm install
Then: npm start

The app will open in your browser at http://localhost:3000 It proxies API calls to the backend at port 5000.

6.Default Login

Use these credentials to log in at the application’s landing page:

Staff ID
Full Name Or Staff Id : **Alice Smith** OR **Bob Jones** OR **100** OR **101** OR **Gaurav G**
Password: **clinic123**

You may enter either the numeric Staff ID or the full name (case-insensitive), with the password clinic123.

7. Usage

• After logging in, use the sidebar to navigate through Patients, Doctors, Appointments, Schedules, Prescriptions, Medical Records, Billing, and Staff.
• Add, edit, or delete records as needed.
• You cannot delete the staff member you are currently logged in as.

8. Troubleshooting

• If ports 3000 or 5000 are in use, stop the conflicting service or change ports.
• If database connection fails, verify your .env DATABASE_URL is correct.
• Ensure clinic_db is loaded before starting the backend.

9. Support

For questions or issues, please reach out to any of our team member

Thank you for using Clinic CMS!