

# CareConnect

Muni Chandrasekar Reddy.K

Hospitals need efficient systems to manage patients, doctors, appointments, and treatments. Currently, many hospitals use manual registers or disconnected software, which makes it difficult to manage records, avoid scheduling conflicts, and track patient history.

CareConnect is a web application that allows Admins, Doctors, and Patients to interact with the system based on their roles.

These are the frameworks that are used in project :

- Flask for application back-end
- Jinja2 templating, HTML, CSS and Bootstraps for application front-end
- SQLite for database

## Technologies and Frameworks Used:

Technology / Library	Purpose
Flask	Core backend web framework
SQLite	Lightweight local database for storing user data
SQLAlchemy	Object Relational Mapper for SQLite database
Jinja2	Template engine for rendering dynamic HTML pages

Bootstrap 5	Frontend styling
Chart.js	To visualize trends in admin page
Flask-session	User authentication

## Database Schema / ER Diagram:

Tables:

**Admin**-Stores id,username and password of admin(aid,username,password)

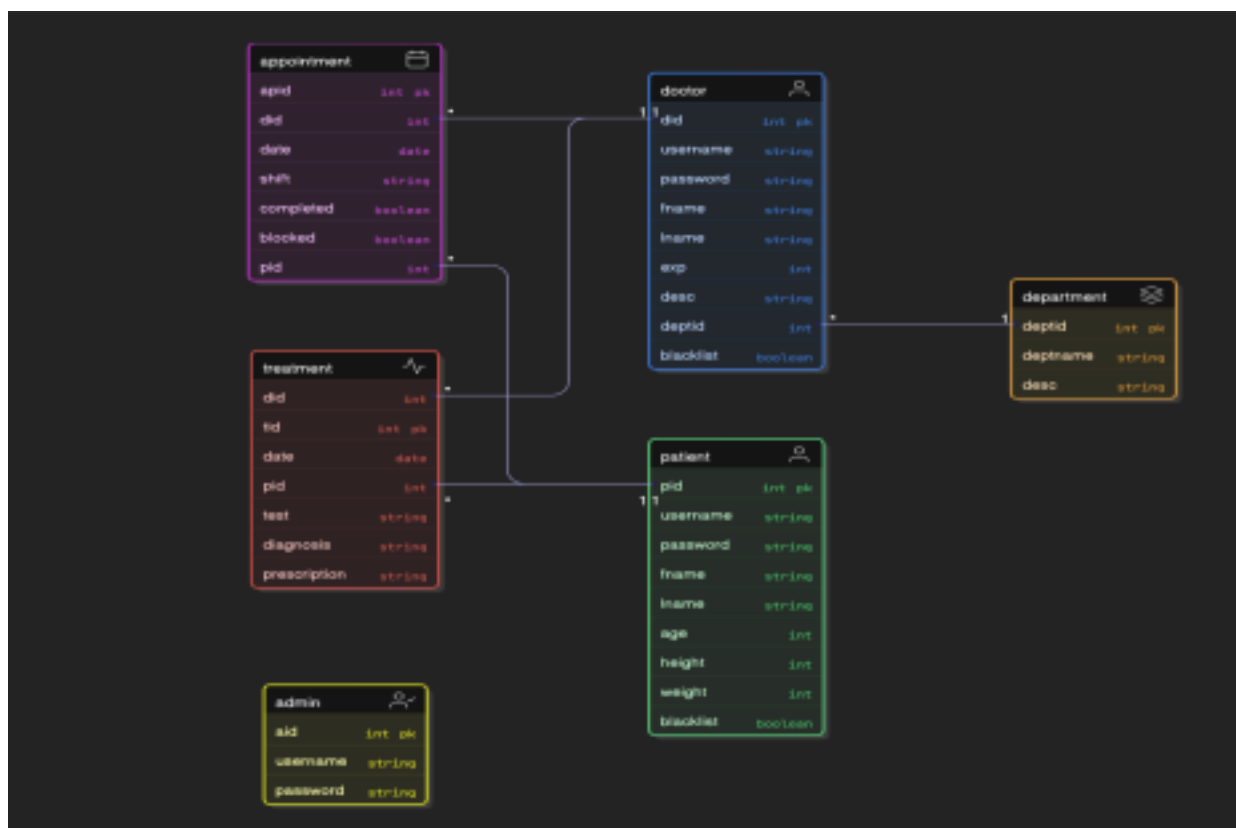
**Patient**-Stores id,username,password and patient details such as age, height (pid,username,password,fname,lname,age,height,weight,blacklist.)

**Doctor**-Store id,username,password and doctors details such as experience,speciality etc(did,username,password,fname,lname,exp,desc,deptid,blacklist.)

**Department**-Stores Id,Department name and description(deptid,deptname,desc)

**Appointment**- Store appointment details such as apid,date,shift,doctor's id,Patient's id etc(apid,did,pid,date,shift,completed,blocked)

**Treatment**- Stores Patient's Treatment details(tid,date,did,pid,test,diagnosis,prescription)



## Architecture and Features:

Folder Structure:

```

├── model.py
├── app.py
├── templates/
│   ├── treatment.html
│   ├── adddepartment.html
│   ├── patienthistory.html
│   ├── index.html
│   ├── patientedit.html
│   ├── doctorreg.html
│   ├── department.html
│   ├── adminlogin.html
│   ├── register.html
│   ├── Patientdash.html
│   ├── doctoredit.html
│   ├── admindash.html
│   ├── ppatientedit.html
│   ├── deptedit.html
│   └── docdash.html
├── instance/
│   └── HMS.db
└── static/

```

## Features:

There are 2 login pages one for admin and other for both patients and doctors .I have restricted routes by roles and protect sessions.following are their features: **Admin:**

- Admin can add,view,edit,remove and blacklist doctors.
- Admin can view,edit,remove and blacklist patients
- Admin can add,edit and remove departments.
- Admin can search for patients and doctors using a searchbar.
- Admin can view appointments and history of that patient.
- He/she can have an overview on no of patients,doctors and appointments using a bar chart at last.

## Patient:

- The patient can edit his/her profile.
  - He/she can view his treatment history.
  - He/she can book appointment based on the department,doctor,speciality. •
- He/she can view appointments made and cancel them

## Doctor:

- Doctors can view appointments ,add treatment , mark it as completed,or cancel the appointment.

- He can view the patients and their records/history.
- He can block slots when he is not available and can view these blocked appointments and cancel them.

## **Video Presentation:**

[https://drive.google.com/file/d/1xSr5KUzi\\_g2fyIDMVr5xl8Rmwa\\_EjDGo/view?usp=sharing](https://drive.google.com/file/d/1xSr5KUzi_g2fyIDMVr5xl8Rmwa_EjDGo/view?usp=sharing)