

Healthcare Patient Management System

A full-stack web application for managing patient records, built with React (frontend) and Flask (backend) with SQLite database.

Prerequisites

Before you begin, ensure you have the following installed:

- **Node.js** (v14 or higher) - [Download here](#)
- **Python** (v3.8 or higher) - [Download here](#)
- **Git** - [Download here](#)

Project Structure

```
Healthcare-Patient-Management/  
├── client/                      # React frontend  
│   ├── src/  
│   ├── public/  
│   └── package.json  
└── server/                     # Flask backend  
    ├── app.py  
    ├── database.db             # SQLite database (created automatically)  
    └── requirements.txt
```

Installation & Setup

1. Clone the Repository

```
git clone  
git clone https://github.com/bhavv04/Healthcare-Patient-Management  
cd Healthcare-Patient-Management
```

2. Backend Setup (Flask)

Navigate to the server directory:

```
cd server
```

Create a virtual environment:

Windows (PowerShell):

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
```

macOS/Linux:

```
python3 -m venv .venv
source .venv/bin/activate
```

Install Python dependencies:

(Install in virtual environment once you're in server directory)

```
pip install -r requirements.txt
```

3. Frontend Setup (React)

Open a **new terminal** and navigate to the client directory:

```
cd client
```

Install Node.js dependencies:

```
npm install
```

Running the Application

You need to run **both** the backend and frontend servers simultaneously.

Terminal 1: Start the Flask Backend

```
cd server
# Activate virtual environment first
.\.venv\Scripts\Activate.ps1 # Windows
source .venv/bin/activate    # macOS/Linux

flask run --debug
```

The Flask server will start on <http://127.0.0.1:5000>

Terminal 2: Start the React Frontend

```
cd client
npm start
```

The React app will automatically open in your browser at <http://localhost:3000>

Usage

1. Open your browser to <http://localhost:3000>
2. Enter a patient name in the input field
3. Click "Submit" to add the patient to the database
4. You'll see a confirmation message with the patient ID
5. The database is stored in [server/database.db](#)

API Endpoints

Backend API (Flask)

- [GET /](#) - Health check endpoint
- [POST /patients](#) - Create a new patient
 - Request body: { "name": "string", "age": number, "email": "string" }
 - Response: { "id": number }
- [GET /patients](#) - Get all patients
 - Response: [{ "id": number, "name": "string", "age": number, "email": "string" }]

Troubleshooting

Python not found

- Make sure Python is installed and added to your PATH
- On Windows, you might need to use [python](#) instead of [python3](#)

Flask module not found

- Ensure your virtual environment is activated
- Run [pip install -r requirements.txt](#) again

React won't start

- Delete [node_modules](#) folder and [package-lock.json](#)
- Run [npm install](#) again

CORS errors

- Make sure Flask is running on port 5000

- Check that `flask-cors` is installed

Database errors

- Delete `database.db` and restart Flask to recreate it
- Make sure you're using `db.session.add()` and `db.session.commit()`

Database

The application uses SQLite for data storage. The database file (`database.db`) is created automatically when you first run the Flask server.

Viewing the Database

Option 1: VS Code Extension

- Install "SQLite Viewer" extension by Florian Klampfer
- Click on `database.db` file to view

Option 2: DB Browser for SQLite

- Download from sqlitebrowser.org
- Open the `database.db` file

Development

Adding New Features

Backend (Flask):

1. Define new models in `app.py`
2. Create new routes/endpoints
3. Remember to use `db.session.add()` and `db.session.commit()`

Frontend (React):

1. Create new components in `src/`
2. Use `fetch()` to call Flask API endpoints
3. Update state with `useState` or `useReducer`

Technologies Used

- **Frontend:** React, JavaScript, CSS
- **Backend:** Flask (Python)
- **Database:** SQLite
- **Other:** Flask-CORS, Flask-SQLAlchemy

License

This project is licensed under the MIT License.

Contributors

Bhavdeep Arora