🟩 PART 1 — HOW TO BUILD THE USER FRONTEND FIRST

This is your **Public User Portal** (GitHub Pages).

✅ Step 1 — Create Your Frontend Project

Use React (recommended):

npx create-react-app cureconnect-frontend

cd cureconnect-frontend

Project structure:

```
src/
 pages/
  Login.jsx
  Register.jsx
  BookAppointment.jsx
  Dashboard.jsx
 components/
  Navbar.jsx
  InputBox.jsx
 services/
  api.js
```

✅ Step 2 — Create the API Service

Inside src/services/api.js:

```
const BASE_URL = "https://your-tunnel-domain.trycloudflare.com/api";


export async function loginUser(data) {
 return fetch(`${BASE_URL}/auth/login`, {
  method: "POST",
  headers: {"Content-Type": "application/json"},
  body: JSON.stringify(data)
 }).then(res => res.json());
}
```

Later, you will replace:

https://your-tunnel-domain.trycloudflare.com

with your **real Cloudflare Tunnel URL**.

✅ Step 3 — Build Pages

✓ Login.jsx

- Input fields → username, password
- Send POST request to /auth/login
- Save JWT token to localStorage
- Redirect to dashboard

✓ Register.jsx

Basic form that POSTs to:

/auth/register

✓ BookAppointment.jsx

- GET doctors list
- POST appointment form

✓ Dashboard.jsx

- Fetch user profile from /auth/me
- Display upcoming appointments

✅ Step 4 — Test Frontend Locally

npm start

Use local API temporarily:

BASE_URL = "http://localhost:5000/api"

Make sure your backend is running locally:

python main.py

When login/register works → FRONTEND IS READY.

🟩 PART 2 — HOW TO DEPLOY FRONTEND TO GITHUB PAGES

1. Install gh-pages

npm install gh-pages --save-dev

2. Edit package.json

Add:

"homepage": "https://your-username.github.io/cureconnect-frontend",

"scripts": {

"predeploy": "npm run build",

"deploy": "gh-pages -d build"

}

3. Deploy:

npm run deploy

Your frontend is now online.

🟦 PART 3 — BUILD & CONNECT CLOUDFLARE TUNNEL

This is the MOST IMPORTANT PART.

This allows your backend (clinic PC) to be accessed from GitHub Pages.

☁️ What is Cloudflare Tunnel?

It creates a **secure path** from internet → your PC.

Your backend stays safe behind Cloudflare, no port forwarding needed.

🟦 STEP-BY-STEP TUNNEL SETUP

🔵 Step 1 — Install cloudflared on Clinic PC

Download from:
https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/installation/

Install and run:

cloudflared login

This opens browser → choose your Cloudflare domain.

🔵 Step 2 — Create Tunnel

Run:

cloudflared tunnel create cureconnect-tunnel

This creates:

- Tunnel ID
- Cert file

🔵 Step 3 — Create the Tunnel Config

Go to:

C:\Users\<YourUser>\.cloudflared\config.yml

Create/Update:

tunnel: cureconnect-tunnel

credentials-file: C:\Users\YourUser\.cloudflared\cureconnect-tunnel.json

ingress:

- hostname: api.cureconnect.in

  service: http://localhost:5000

- service: http_status:404

What you are doing:

- Exposing your backend to your domain
- Mapping domain → PC backend

🔵 Step 4 — Route DNS

Go to Cloudflare Dashboard → DNS
Add CNAME:

api CNAME <TUNNEL-ID>.cfargotunnel.com

🔵 Step 5 — Start Tunnel

cloudflared tunnel run cureconnect-tunnel

Now your backend is LIVE on HTTPS:

https://api.cureconnect.in/api/

🟩 PART 4 — CONNECT FRONTEND → BACKEND

Now go back to your frontend api.js and update:

const BASE_URL = "https://api.cureconnect.in/api";

Deploy frontend again:

npm run deploy

Now your GitHub Pages app communicates with clinic backend live.

🟩 PART 5 — CONFIRM DATA IS SAVED IN DATABASE

Test this:

Open your public site:

https://your-username.github.io/cureconnect-frontend

Try:

- Create account
- Login
- Book appointment

Then open SQLite database on clinic PC:

clinic.db

Check:

- User table → new user added
- Appointment table → new appointment
- TokenQueue table → new token

If these tables update → **Cloudflare tunnel + frontend + backend are connected successfully**.

🟦 FINAL WORKING FLOW

🟢 Public user:

GitHub Pages → Tunnel → Backend → SQLite

🟡 Admin user:

Local Admin UI → Backend → SQLite