SmartCare – AI-Powered Telemedicine Platform

> Build a full-stack telemedicine app with support for real-time instant/emergency consultations, intelligent doctor suggestions, and interactive analytics dashboards using Dash. The goal is to challenge your skills in backend logic, real-time workflows, creative problem solving, and data science.

🧑‍💻 Tech Stack

Layer	Technologies
Frontend	React.js + Tailwind CSS
Backend	Python
Realtime	WebSockets (Socket.IO / FastAPI WebSocket)
DB	PostgreSQL / mysql
Dashboards	Dash (Plotly)
Data Science	Pandas, NumPy, scikit-learn, Voice Recognition (`speech_recognition` , `whisper` , or similar)

🐳 Docker Integration

Features

- Containerized Python backend
- `.env` support for environment variables (DB, ports, keys)
- Docker Compose for backend + DB setup

Deliverables

- `Dockerfile` for Python backend (Flask/FastAPI)
- `docker-compose.yml` (if DB or Dash is to be containerized too)
- Instructions in `README.md` to run: docker build -t smartcare-backend .
  docker run -p 8000:8000 smartcare-backend

🔐 Module 1: Auth & Roles

Features

- Sign Up / Login for 3 roles: `Patient` , `Doctor` , `Admin`
- Role-based routing and access
- Admin can block users

Deliverables

- Secure auth (JWT or session)
- Role-based UI rendering

🧑‍⚕️ Module 2: Doctor Panel

Features

- Doctor Profile (specialization, availability, pricing)
- **Toggle instant consultation availability**
- View appointments
- Accept/Reject instant requests
- Create prescriptions (text + downloadable PDF)

Challenges

- **Time-slot management logic** (no overlaps)
- **Availability check** (for instant consults)

🧑‍💻 Module 3: Patient Panel

Features

- Book appointments (choose `Normal` or `Emergency`)
- View doctor list (filter: specialization, rating, availability)
- Upload symptoms & past reports
- See available doctors for **Instant Consult** (only if eligible)
- Join live chat after booking

📅 Module 4: Appointment System

Normal Appointment

- Patients select doctor, date, time
- Doctor sees request, can accept/reject

Emergency Appointment

- Can be requested by patient
- Must be shown **with priority** in doctor's dashboard
- Can override other time slots

Logic Required

- Conflict detection engine
- Emergency override with permission

⚡ Module 5: Instant Consultation System

Logic

- A doctor appears for instant consult **only if**:
  - They are **online**
  - Instant toggle is ON
  - No appointments in **next 15–30 minutes**

Flow

1. Patient sees "Instant Consultation" button
2. Click sends **real-time request**
3. Doctor can **Accept/Reject**
4. If accepted → Start live consultation

> **All instant consultations are treated as Emergency**

💬 Module 6: Real-Time Chat System

Features

- WebSocket-powered chat between doctor and patient
- Send messages during consultation
- Save chat history to DB
- ❌ **Typing indicators are NOT required**

🗣️ Module 7: Voice-to-Text During Consultation

Features

- Both patient and doctor can use **voice input**
- Voice is converted to **text using Python speech-to-text model**
- Displayed in chat interface as messages

Tools

- `speech_recognition` (with Google API or offline)
- OR `whisper` , `vosk` , or `transformers` models

📈 Module 8: Admin Dashboard + Dash Analytics

Dash Features

- Show separate consultation summaries:
  - 📊 Instant
  - 📅 Appointment (Normal & Emergency)
- Trends:
  - Appointments over time
  - Top symptoms
  - Most active doctors
  - Emergency vs Normal consult ratios
  - Specialization demand

Admin Tools

- View and manage all users
- View doctor performance metrics
- Revenue summary (if pricing used)

🧠 Module 9: Intelligence & Data Science Features

🔍 Doctor Suggestion Engine

- Input: Symptoms
- Output: Ranked list of doctors (based on:
  - Specialization match
  - Rating
  - Availability
- Logic or ML-based (classifier optional)

🚑 Health Risk Predictor

- Inputs: Age, Gender, BP, Sugar, symptoms
- Output: Low / Medium / High risk
- Use rules or logistic regression

✅ Final Deliverables

- Full GitHub repo
- Live or demo-ready version
- Dashboards of admin and doctor
- Voice-to-text working in consultation
- PDF reports (prescriptions)
- Sample dataset(s)
- Optional: Loom/video walkthrough
- Brief documentation (README + data flow)