Unified Healthcare Records Application

A centralized platform for managing digital health records with secure access, appointment scheduling, and AI-assisted voice dictation for enhanced healthcare workflows.

📌Notion Project Tracker (2-Month Plan)

Task	Assignee	Status	Due Date	Priority	Notes
Project Kickoff & Scope Finalization	-	Done	Jul 1	High	Define project scope and milestones
UI/UX Wireframe Design	-	In Progress	Jul 6	High	Create responsive wireframes for web app
User Auth Module (JWT + Roles)	-	To Do	Jul 10	High	Includes login, signup, and role access
Patient Records CRUD APIs	-	To Do	Jul 15	High	Mongo/PostgreSQL based APIs
Appointment Scheduler Module	-	To Do	Jul 20	Medium	Doctor/patient booking system
Prescription Management	-	To Do	Jul 25	Medium	E-prescriptions and medication logs
Voice Dictation AI (Google STT)	-	To Do	Jul 30	High	Transcribe consultation notes via mic
Dashboard & Analytics UI	-	To Do	Aug 5	Medium	Charts for appointments, records
Integration Testing	-	To Do	Aug 10	High	Validate module interconnectivity
Security Audit & MFA	-	To Do	Aug 15	High	Test data protection, MFA setup
Deployment on AWS EC2	-	To Do	Aug 20	High	Production push with SSL & backups
Final QA & Documentation	-	To Do	Aug 25	High	Cleanup, test cases, final PDF

📄PDF Documentation (Layout Summary)

1. Project Overview

The application is designed to unify scattered health records into one secure digital system accessible to doctors, patients, and healthcare administrators with advanced AI voice assistance.

2. Module Breakdown

- **Module 1:** JWT-based Auth and Role Management
- **Module 2:** Patient Profile and Health Data Management
- **Module 3:** Appointment & Slot Booking
- **Module 4:** Prescription Generation
- **Module 5:** AI Voice Integration for Dictation
- **Module 6:** Lab Upload System with Report Parsing
- **Module 7:** Admin Dashboard & Analytics Charts

3. Tech Stack

- **Frontend:** React.js + Tailwind CSS
- **Backend:** Node.js with Express
- **Database:** MongoDB or PostgreSQL
- **AI Voice:** Google Speech-to-Text or Whisper API
- **Auth:** JWT + Bcrypt + MFA
- **Cloud:** AWS EC2, S3, RDS (if SQL)

4. Workflow Description (Simplified)

- User logs in → token generated → navigates to dashboard
- Patient data fetched via secured API → shown in UI
- Doctor uploads notes or uses voice → converted to text
- Appointments shown in calendar view → book/update

5. Security Protocols

- HTTPS, AES-256 storage encryption, JWT validation
- Access logs, audit trail, and optional HIPAA readiness

6. Deployment Strategy

- Dockerized containers
- GitHub Actions CI/CD
- NGINX + SSL + Domain

7. Future Scope

- Mobile app (Flutter)
- Wearable health integration

- FHIR/HL7 APIs for hospital data sync

📊Comprehensive Action Plan & Resource Summary

1. Project Resources

- **Documentation PDF**: `Unified_Healthcare_Records_Kit.pdf`
- **Notion CSV Board**: `notion_project_tracker.csv`
- **Boilerplate Source ZIP**: `healthcare_app_boilerplate.zip`
- **GitHub README Template**: see Docs folder

2. Modules & Required Updates

Module	Key Enhancements
Auth & Role Management	Add **MFA**, password-reset flow, refresh tokens
Patient Records CRUD	Advanced search filters, strict file-upload validation
Appointment Scheduler	Email / SMS reminders via SendGrid & Twilio
Prescription Service	Integrate drug-interaction API (e.g., FDA OpenAPI)
Voice Dictation AI	Multi-language support, command parsing library
Dashboard Analytics	Real-time charts, CSV export, drill-down filters
Lab Integration	HL7 / FHIR parser, auto-tagging of report PDFs

3. Features to Build (Next 8 Weeks)

1. **Telemedicine (WebRTC) module**
2. Tailwind-based **mobile-responsive redesign**
3. **Admin audit-log viewer** interface
4. **Bulk patient import** (CSV → DB)
5. Role-based **PDF record export**

4. Technology Improvements

- Migrate API layer to **TypeScript**
- Introduce **GraphQL gateway** for complex queries
- Add **Docker Compose** for turnkey local stack
- Enforce code quality with **ESLint + Prettier**
- CI pipeline: **GitHub Actions** for lint/test/build
- Investigate **PostgreSQL logical replication** for read scaling

5. Process Recommendations

- **2-week Agile sprints** with planning & retro

- **Daily 15 min stand-ups**
- Pull-request **code reviews** with templates
- **Automated tests** (Jest/Supertest) in CI
- **Monthly security pentest** & **quarterly backup drill**

6. Eight-Week Action Checklist

Week	Focus	Deliverables
1	DB Schema v1.1, MFA	Finalized ERD, MFA endpoints
2	Email Service	SES/SendGrid integration, UI hooks
3	Reminders	Cron job, Twilio SMS prototype
4	Telemed P-0	WebRTC MVP, internal demo
5	Audit Log UI	Admin page, RBAC tests
6	Security Fixes	Pentest report & patches
7	Load/Perf Test	k6 scripts, query optimisation
8	Prod Deploy	Blue-green rollout, post-deploy QA

7. Ownership Matrix (RACI)

Role	Responsible	Accountable	Consulted	Informed
Product Owner		✓	✓	✓
Tech Lead	✓	✓	✓	✓
Frontend Dev	✓			
Backend Dev	✓			
DevOps	✓			
QA	✓			

8. Immediate Next Steps

1. Import CSV into Notion → convert to Kanban
2. Push boilerplate code to new GitHub repo
3. Add environment files (`.env`) for dev & prod
4. Schedule **Sprint 0 kickoff** meeting

📘GitHub README Template

~~~
# Unified Healthcare Records Application

A full-stack platform for managing electronic health records with secure user
roles, AI-based voice dictation, and modern UI/UX.

## 🗜️ Features
- Role-based user registration (Doctor, Patient, Admin)
- Patient record management (CRUD)
- Appointment booking system
- Voice-to-text AI for clinical notes
- Analytics dashboard
- JWT + MFA security

## 🛠️ Tech Stack
- **Frontend:** React.js
- **Backend:** Node.js + Express
- **Database:** MongoDB/PostgreSQL
- **AI:** Google STT or Whisper
- **Deployment:** AWS EC2 + Docker

## 📦 Installation
```bash
git clone https://github.com/your-username/healthcare-app.git
cd backend && npm install
cd ../frontend && npm install
~~~

🩲Environment Variables

- `.env` file in backend with DB_URI, JWT_SECRET, STT_API_KEY

📁Folder Structure

```
root/
├── backend/
│   └── routes/, controllers/, models/
├── frontend/
│   └── components/, pages/, services/
├── docker-compose.yml
└── README.md
```

Contributing

Pull requests welcome. Please lint and test your code.

📄License

MIT

```
---

Let me know if you'd like:
- A downloadable **PDF version**
- A **Notion template link**
- The full **GitHub repo structure** as a `.zip`

I can generate those next!
```