

Hệ thống Đặt lịch Khám Bệnh - Tài liệu Đầy đủ

❖ Tổng quan

- **Tên dự án:** Healthcare Booking System
- **Mục đích:** Đặt lịch khám, quản lý hồ sơ sức khỏe, nhắc nhở uống thuốc
- **Tech Stack:**
 - Backend: Node.js + Express + PostgreSQL + Prisma
 - Frontend: React + Tailwind + shadcn/ui
 - Auth: JWT

DATABASE SCHEMA

1. users

```
sql
```

```
CREATE TABLE users (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    phone VARCHAR(20),
    role VARCHAR(20) DEFAULT 'patient', -- 'patient', 'doctor', 'clinic_admin'
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
```

2. patient_profiles

```
sql
```

```
CREATE TABLE patient_profiles (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id) ON DELETE CASCADE,
    full_name VARCHAR(255) NOT NULL,
    date_of_birth DATE,
    gender VARCHAR(10),
    blood_type VARCHAR(5),
    allergies TEXT[],
    emergency_contact VARCHAR(255),
    emergency_phone VARCHAR(20),
    created_at TIMESTAMP DEFAULT NOW()
);
```

3. clinics

sql

```
CREATE TABLE clinics (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    name VARCHAR(255) NOT NULL,
    address TEXT,
    phone VARCHAR(20),
    specialty TEXT[],
    rating DECIMAL(2,1) DEFAULT 0,
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP DEFAULT NOW()
);
```

4. doctors

sql

```
CREATE TABLE doctors (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id),
    clinic_id UUID REFERENCES clinics(id),
    full_name VARCHAR(255) NOT NULL,
    specialty VARCHAR(100),
    license_number VARCHAR(100),
    created_at TIMESTAMP DEFAULT NOW()
);
```

5. appointments

sql

```
CREATE TABLE appointments (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID REFERENCES users(id),
    clinic_id UUID REFERENCES clinics(id),
    doctor_id UUID REFERENCES doctors(id),
    appointment_date TIMESTAMP NOT NULL,
    status VARCHAR(20) DEFAULT 'scheduled', -- scheduled, confirmed, completed, cancelled
    type VARCHAR(50), -- consultation, checkup, follow-up
    reason TEXT,
    notes TEXT,
    created_at TIMESTAMP DEFAULT NOW()
);
```

6. medical_records

sql

```
CREATE TABLE medical_records (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID REFERENCES users(id),
    appointment_id UUID REFERENCES appointments(id),
    diagnosis TEXT,
    treatment TEXT,
    doctor_notes TEXT,
    record_date DATE DEFAULT CURRENT_DATE,
    created_at TIMESTAMP DEFAULT NOW()
);
```

7. medications

sql

```
CREATE TABLE medications (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID REFERENCES users(id),
    record_id UUID REFERENCES medical_records(id),
    name VARCHAR(255) NOT NULL,
    dosage VARCHAR(100),
    frequency VARCHAR(100),
    start_date DATE NOT NULL,
    end_date DATE,
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP DEFAULT NOW()
);
```

8. reminders

```
sql

CREATE TABLE reminders (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    medication_id UUID REFERENCES medications(id) ON DELETE CASCADE,
    patient_id UUID REFERENCES users(id),
    reminder_time TIME NOT NULL,
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMP DEFAULT NOW()
);
```

9. health_metrics

```
sql

CREATE TABLE health_metrics (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    patient_id UUID REFERENCES users(id),
    metric_type VARCHAR(50), -- weight, blood_pressure, glucose, heart_rate
    value DECIMAL(10,2),
    unit VARCHAR(20),
    measured_at TIMESTAMP DEFAULT NOW(),
    created_at TIMESTAMP DEFAULT NOW()
);
```

🔌 API ENDPOINTS (20+)

Auth (6)

POST /api/auth/register	- Đăng ký
POST /api/auth/login	- Đăng nhập
POST /api/auth/logout	- Đăng xuất
POST /api/auth/refresh	- Refresh token
POST /api/auth/forgot-password	- Quên mật khẩu
POST /api/auth/reset-password	- Reset mật khẩu

Patient Profile (4)

GET /api/profile	- Xem profile
PUT /api/profile	- Cập nhật profile
POST /api/profile/avatar	- Upload avatar
GET /api/profile/:id	- Admin xem profile

Clinics (3)

GET /api/clinics	- Danh sách phòng khám
GET /api/clinics/:id	- Chi tiết phòng khám
GET /api/clinics/search	- Tìm kiếm

Appointments (8)

GET /api/appointments	- Danh sách lịch (theo role)
GET /api/appointments/:id	- Chi tiết lịch
POST /api/appointments	- Đặt lịch mới
PUT /api/appointments/:id	- Cập nhật lịch
DELETE /api/appointments/:id	- Hủy lịch
GET /api/appointments/upcoming	- Lịch sắp tới
POST /api/appointments/:id/confirm	- Xác nhận (doctor)
GET /api/clinics/:id/available-slots	- Xem lịch trống

Medical Records (5)

GET /api/records	- Danh sách hồ sơ
GET /api/records/:id	- Chi tiết hồ sơ
POST /api/records	- Tạo hồ sơ (doctor)

PUT /api/records/:id - Cập nhật hồ sơ
DELETE /api/records/:id - Xóa hồ sơ

Medications & Reminders (6)

GET /api/medications - Danh sách thuốc
POST /api/medications - Thêm thuốc
PUT /api/medications/:id - Cập nhật thuốc
DELETE /api/medications/:id - Xóa thuốc
POST /api/medications/:id/reminders - Tạo lời nhắc
PUT /api/reminders/:id - Cập nhật lời nhắc

Health Metrics (3)

GET /api/health-metrics - Danh sách chỉ số
POST /api/health-metrics - Thêm chỉ số
GET /api/health-metrics/trends - Xu hướng

Reports (3)

GET /api/reports/medication-adherence - Tuân thủ uống thuốc
GET /api/reports/metric-trends - Xu hướng chỉ số
GET /api/reports/appointments - Tần suất khám

⌚ ROLES & PERMISSIONS

PATIENT (Bệnh nhân)

Có thể làm:

- Đăng ký, đăng nhập
- Xem/sửa profile của mình
- Tìm kiếm phòng khám
- Đặt lịch khám
- Xem/hủy lịch của mình
- Xem hồ sơ bệnh án của mình
- Quản lý thuốc và lời nhắc

- Nhập chỉ số sức khỏe
- Xem báo cáo cá nhân

KHÔNG thể:

- Xem thông tin bệnh nhân khác
- Tạo hồ sơ khám bệnh
- Xác nhận lịch hẹn

DOCTOR (Bác sĩ)

Có thể làm:

- Tất cả quyền của Patient
- Xem danh sách lịch khám của mình
- Xác nhận/tù chối lịch hẹn
- Xem thông tin bệnh nhân (khi có lịch)
- Tạo hồ sơ khám bệnh
- Kê đơn thuốc
- Cập nhật trạng thái khám

KHÔNG thể:

- Xem lịch của bác sĩ khác
- Quản lý phòng khám

CLINIC_ADMIN (Quản lý phòng khám)

Có thể làm:

- Quản lý thông tin phòng khám
- Quản lý danh sách bác sĩ
- Xem tất cả lịch của phòng
- Xác nhận/hủy lịch hẹn
- Đặt lịch hộ bệnh nhân
- Xem báo cáo phòng khám

FRONTEND SCREENS (12+)

1. Auth Pages (2)

- Login/Register
- Forgot Password

2. Patient Dashboard (1)

- Tổng quan: lịch sắp tới, nhắc thuốc hôm nay
 - Card: chỉ số sức khỏe gần nhất
 - Quick actions: Đặt lịch, thêm thuốc

3. Booking Flow (3)

- Tìm kiếm phòng khám
- Chọn ngày giờ khám
- Xác nhận đặt lịch

4. Appointments (2)

- Danh sách lịch đã đặt
- Chi tiết lịch khám

5. Medical Records (2)

- Danh sách hồ sơ
- Chi tiết hồ sơ

6. Medications (2)

- Tủ thuốc (danh sách)
- Lịch sử uống thuốc

7. Health Tracking (1)

- Nhập & xem chỉ số sức khỏe

8. Reports (1)

- Dashboard báo cáo (3 loại)

9. Settings (1)

- Cài đặt tài khoản

10. Doctor Dashboard (1)

- Danh sách lịch khám hôm nay
- Xác nhận/tù chối lịch
- Vào khám bệnh

3 BÁO CÁO + DASHBOARD

Dashboard (Overview)

Metrics:

- Số lịch khám tháng này
- Thuốc đang dùng
- Tuân thủ uống thuốc (%)
- Lịch sắp tới

Charts:

- Bar chart: Tần suất khám 6 tháng
- Line chart: Xu hướng cân nặng

1. Báo cáo Tuân thủ Uống Thuốc

Filters: Khoảng thời gian, loại thuốc

Hiển thị:

- Tỷ lệ tuân thủ (%)
- Số lần đúng/tổng số

- Pie chart: Đã uống/Bỏ lỡ
- Table: Chi tiết từng loại thuốc

Export: CSV, PDF

2. Báo cáo Xu hướng Chỉ số

Filters: Loại chỉ số, thời gian

Hiển thị:

- Giá trị TB/Max/Min
- Line chart: Xu hướng theo thời gian
- So sánh với ngưỡng chuẩn

Export: CSV, PDF

3. Báo cáo Tần suất Khám

Filters: Loại khám, thời gian

Hiển thị:

- Tổng số lượt khám
- Tần suất TB (lần/tháng)
- Bar chart: Số lượt theo tháng
- Pie chart: Phân bố theo loại

Export: CSV, PDF

🔒 BẢO MẬT

1. Authentication

javascript

```
//JWT với access token (15 phút) + refresh token (7 ngày)
const token = jwt.sign({ userId, role }, SECRET, { expiresIn: '15m' });
```

2. Password

```
javascript

// Bcrypt hash với salt rounds = 10
const hashedPassword = await bcrypt.hash(password, 10);
```

3. Authorization Middleware

```
javascript

const authorize = (...roles) => {
  return (req, res, next) => {
    if (!roles.includes(req.user.role)) {
      return res.status(403).json({ error: 'Forbidden' });
    }
    next();
  };
};
```

4. Input Validation

```
javascript

// Joi validation cho tất cả input
const schema = Joi.object({
  email: Joi.string().email().required(),
  password: Joi.string().min(8).required()
});
```

5. Rate Limiting

```
javascript

// Express rate limit: 100 requests/15 phút
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000,
  max: 100
});
app.use('/api/', limiter);
```

WORKFLOWS CHÍNH

Workflow 1: Đặt lịch khám

1. Patient login
2. Tìm phòng khám (GET /api/clinics/search)
3. Xem lịch trống (GET /api/clinics/:id/available-slots)
4. Chọn ngày giờ
5. Điền thông tin (lý do, triệu chứng)
6. POST /api/appointments
7. Nhận thông báo xác nhận
8. Doctor xem lịch (GET /api/appointments)
9. Doctor xác nhận (POST /api/appointments/:id/confirm)
10. Patient nhận thông báo

Workflow 2: Khám bệnh

1. Doctor xem danh sách lịch hôm nay
2. Chọn bệnh nhân → cập nhật status = 'in_progress'
3. Xem hồ sơ bệnh nhân (GET /api/records?patient_id=...)
4. Khám bệnh
5. Tạo hồ sơ mới (POST /api/records)

```
{  
  "diagnosis": "...",  
  "treatment": "...",  
  "doctor_notes": "..."  
}
```
6. Kê đơn thuốc (POST /api/medications)
7. Cập nhật status = 'completed'
8. Patient xem hồ sơ mới

Workflow 3: Quản lý thuốc

1. Patient/Doctor thêm thuốc (POST /api/medications)
2. Tạo lời nhắc (POST /api/medications/:id/reminders)

```
{  
  "reminder_time": "08:00",  
  "is_active": true  
}
```
3. Hệ thống gửi thông báo vào 08:00 hàng ngày

4. Patient ghi nhận đã uống

5. Xem báo cáo tuân thủ

SEED DATA

Users (4)

```
javascript
```

```
[  
  { email: 'patient1@test.com', password: '12345678', role: 'patient' },  
  { email: 'patient2@test.com', password: '12345678', role: 'patient' },  
  { email: 'doctor1@test.com', password: '12345678', role: 'doctor' },  
  { email: 'admin1@test.com', password: '12345678', role: 'clinic_admin' }  
]
```

Clinics (3)

```
javascript
```

```
[  
 {  
   name: 'Phòng khám Đa khoa ABC',  
   address: '123 Lê Lợi, Q1, TP.HCM',  
   phone: '028-1234-5678',  
   specialty: ['internal_medicine', 'cardiology'],  
   rating: 4.8  
 },  
 {  
   name: 'Bệnh viện XYZ',  
   address: '456 Nguyễn Huệ, Q1, TP.HCM',  
   phone: '028-8765-4321',  
   specialty: ['surgery', 'pediatrics'],  
   rating: 4.5  
 },  
 {  
   name: 'Phòng khám Nhi ABC',  
   address: '789 Điện Biên Phủ, Q3, TP.HCM',  
   phone: '028-1111-2222',  
   specialty: ['pediatrics'],  
   rating: 4.7  
 }  
 ]
```

Appointments (10)

- 5 lịch quá khứ (status: completed)
- 3 lịch tương lai (status: confirmed)
- 2 lịch đợi xác nhận (status: scheduled)

Medical Records (8)

- Mỗi appointment completed có 1 record

Medications (6)

- 6 loại thuốc với start_date khác nhau

Reminders (10)

- Mỗi medication có 1-2 reminders

Health Metrics (50)

- Weight, blood_pressure, glucose
 - Data cho 3 tháng gần nhất
-

API RESPONSE FORMAT

Success Response

```
json

{
  "success": true,
  "data": { ... },
  "message": "Operation successful"
}
```

Error Response

```
json

{
  "success": false,
  "error": "Error message",
  "details": { ... }
}
```

Paginated Response

```
json
```

```
{  
  "success": true,  
  "data": [ ... ],  
  "pagination": {  
    "total": 100,  
    "page": 1,  
    "limit": 10,  
    "totalPages": 10  
  }  
}
```

🚀 IMPLEMENTATION ORDER

Phase 1: Core Setup (Week 1)

1. Setup database + Prisma schema
2. Seed data
3. Auth APIs (register, login, JWT)
4. Basic CRUD cho users, profiles

Phase 2: Booking System (Week 2)

1. Clinics APIs
2. Appointments APIs
3. Frontend: Tìm phòng khám
4. Frontend: Đặt lịch flow
5. Frontend: Danh sách lịch

Phase 3: Medical Management (Week 3)

1. Medical Records APIs
2. Medications & Reminders APIs
3. Health Metrics APIs
4. Frontend: Các màn hình tương ứng

Phase 4: Reports & Polish (Week 4)

1. Reports APIs với export CSV/PDF
 2. Dashboard với charts
 3. Notifications (email/push)
 4. Testing & bug fixes
 5. Documentation
-

DOCUMENTATION CẦN TẠO

1. README.md

markdown

```
# Healthcare Booking System
```

```
## Setup
```

```
npm install
```

```
npm run db:migrate
```

```
npm run db:seed
```

```
## Environment Variables
```

```
DATABASE_URL=...
```

```
JWT_SECRET=...
```

```
JWT_REFRESH_SECRET=...
```

```
## Run
```

```
npm run dev
```

2. API Documentation (Swagger)

- Auto-generate từ code

3. ERD Diagram

- Xuất từ database tool

4. Postman Collection

- Export tất cả APIs để test
-

CHECKLIST NGHIỆM THU

Chức năng

- Auth: Register, Login, Logout
- Patient: Đặt lịch, xem lịch, hủy lịch
- Doctor: Xem lịch, xác nhận, khám bệnh
- CRUD cho 9 entities
- 3 báo cáo + 1 dashboard
- Export CSV/PDF

Technical

- ≥20 API endpoints
- ≥12 màn hình frontend
- JWT authentication
- Role-based authorization
- Input validation
- Rate limiting
- Error handling

Quality

- Seed data đầy đủ
 - API response format nhất quán
 - Code clean, có comments
 - README rõ ràng
 - Swagger docs
 - Demo được end-to-end
-

NOTES QUAN TRỌNG

1. Đơn giản hóa:

- Không cần realtime (WebSocket) → dùng polling

- Không cần email thật → log ra console
- Không cần payment → giả lập

2. **Ưu tiên:**

- Hoàn thành CRUD trước
- UI đơn giản, functional
- Báo cáo dùng thư viện sẵn (recharts)

3. **Có thể bỏ qua (nếu thiếu thời gian):**

- AI features
- Notifications thật
- File upload (dùng URL thay vì file)
- Audit logs chi tiết

4. **Must have:**

- Auth + Authorization
 - Đặt lịch workflow
 - 3 báo cáo + dashboard
 - Seed data
-

SUPPORT

Nếu AI coder gặp vấn đề:

1. Kiểm tra lại database schema
2. Đảm bảo JWT middleware đúng
3. Check role-based permissions
4. Verify API response format
5. Test với Postman trước khi làm FE

READY TO CODE! 