Assignment 2 — Secure Clinic Visit & Diagnostic Upload System

Technology: .NET / Java / PHP (any one)
Domain: Medical Clinic / Diagnostic Center
Focus: JWT Auth + Cascading dropdown + CRUD + Multiple file upload + Summary from related data

---

1) Problem Statement

Build a small application for a diagnostic center where staff can **register/login**, manage **patients and their visits**, select **medical tests using cascading dropdowns**, and upload **multiple diagnostic files** for a visit. The system should also provide a **summary listing** that shows visit details along with patient, selected test(s), and file information.

---

2) Functional Requirements

A) Authentication & Authorization (JWT)

1. **Registration**
    - Fields: Full Name, Email (unique), Password
    - Store passwords securely (hashed)
2. **Login**
    - Email + Password
    - Return JWT on success
3. **Route Protection**
    - All APIs except register/login must require Authorization: Bearer <token>

B) Master Data + Cascading Dropdown

Create a cascading dropdown flow (API-driven/AJAX):

- **Department → Test Category → Test Name**

Rules:

- Only Department enabled initially
- Selecting Department loads Test Category
- Selecting Test Category loads Test Name
- Changing a parent resets the child dropdown(s)

Provide seed/hardcoded data such as:

- Cardiology → ECG Tests → ECG, TMT
- Pathology → Blood Tests → CBC, ESR
- Radiology → Imaging → MRI Brain, CT Brain, etc.

C) Patient Management (CRUD)

Create APIs and UI :

- Add patient
- Edit patient
- Delete patient (soft delete preferred)
- List/search patients
- View patient details

Minimum patient fields:

- Patient Name, Age, Mobile/Email (optional), Gender (optional)

D) Visit & Test Ordering (CRUD)

For a selected patient, staff should be able to:

- Create a **Visit**
    - Visit Date (required)
    - Notes (optional)
- Add **one or more Tests** to that visit using cascading dropdown selections
- Update visit details and selected tests (basic update is enough)
- Delete/cancel visit (soft delete preferred)
- List visits for a patient

E) Multiple File Upload for Visit

Enable uploading **multiple files** per visit (or per selected test—your choice).

Rules:

- Allowed types: PDF, JPG, PNG
- Max size: 5 MB per file
- Max files per request: 5
- Store and return file metadata:
    - Original file name, stored file name/path, file size, uploaded date, uploaded by user

Required operations:

- Upload files for a visit
- List files for a visit
- Download/view a file
- Delete a file

F) Summary & Details Views (Related Data Fetch)

Create an endpoint (and UI if possible) that shows a consolidated summary:

Visit Summary List should include:

- Patient Name, Age
- Visit Date
- Selected Department/Category/Test(s)
- Uploaded File Count
- Created/Uploaded by (logged-in user)

Also create a **Visit Full Details** endpoint that returns:

- Patient info
- Visit info
- All selected tests
- Uploaded files (grouping optional)

3) API Requirements (Minimum Endpoints)

You may name routes differently, but the system must support equivalent behavior.

Auth

- POST /api/auth/register
- POST /api/auth/login

Master Data

- GET /api/departments
- GET /api/categories?departmentId=
- GET /api/tests?categoryId=

Patients

- POST /api/patients
- PUT /api/patients/{id}
- DELETE /api/patients/{id}
- GET /api/patients
- GET /api/patients/{id}

Visits

- POST /api/visits
- PUT /api/visits/{id}
- DELETE /api/visits/{id}
- GET /api/visits?patientId=
- GET /api/visits/{id}

Files

- POST /api/visits/{visitId}/files (multipart, multiple)
- GET /api/visits/{visitId}/files
- GET /api/files/{fileId}/download
- DELETE /api/files/{fileId}

Summary

- GET /api/visit-summary
- GET /api/visits/{id}/full

4) Validation & Error Handling

- Validate required fields for patient/visit/test selection
- Validate file type/size/count
- Return proper HTTP codes: 400, 401, 403, 404, 500
- If saving visit/tests/files fails, avoid partial/inconsistent saved data

5) Data Storage Expectations

Design the storage so that:

- Users, patients, visits, selected tests, and uploaded files are stored separately and linked appropriately.
- Summary and details endpoints can be generated from stored data without hardcoding.

(You decide the exact schema.)

6) Deliverables

1. Source code
2. DB script/migrations + seed data for dropdowns
3. README with:
    - Setup steps
    - How to register/login and use JWT
    - How to test APIs (Postman collection or Swagger/OpenAPI preferred)

7) Testings

- JWT auth implemented correctly + protected routes
- Cascading dropdown works correctly (reset + load)
- Patient and visit CRUD works
- Multiple file upload works with validation + download
- Summary & full details endpoints correctly return consolidated data
- Code quality (structure, naming, error handling)

Sample Seed Dataset

Medical Master Data (Cascading Dropdown)

1) Departments

DepartmentId	DepartmentName
1	Cardiology
2	Neurology
3	Orthopedics
4	Pathology
5	Radiology

2) Test Categories

CategoryId	DepartmentId	CategoryName
1	1	ECG Tests
2	1	Blood Tests
3	2	Imaging
4	2	Neuro Tests
5	3	Imaging
6	3	Physical Tests
7	4	Blood Tests
8	4	Urine Tests
9	5	Imaging
10	5	Ultrasound

3) Test Names

TestId	CategoryId	TestName
1	1	ECG
2	1	TMT
3	1	Holter Monitoring
4	2	Troponin
5	2	Lipid Profile
6	2	CK-MB
7	3	MRI Brain
8	3	CT Brain
9	4	EEG
10	4	Nerve Conduction Study
11	5	X-Ray Knee
12	5	X-Ray Spine
13	6	Bone Density Test
14	6	Joint Mobility Test
15	7	Complete Blood Count (CBC)
16	7	ESR
17	7	Blood Sugar
18	8	Urine Routine
19	8	Urine Culture
20	9	CT Chest
21	9	MRI Abdomen
22	10	USG Abdomen
23	10	USG Pelvis