Full Stack (MERN) Developer Task - OralVis Healthcare

Objective

Build a small MERN app where:
1) A Patient logs in and uploads a teeth photo with basic details.
2) An Admin logs into a portal, views submissions, annotates the image, saves the annotation, and generates a PDF.
3) **Bonus**: Store original image, annotated image, and generated PDF in AWS S3 and include their links in the PDF & UI.
4) Auth can be JWT or HTTP-only cookies.

Mandatory Features

1) Authentication & Roles

- Roles: patient, admin
- Auth: JWT or HTTP-only cookie sessions
- Routes: /auth/register, /auth/login, /auth/logout
- Access control:
  * Patient: upload & see their own submissions + report links
  * Admin: list & view all submissions, annotate, generate PDF

2) Patient – Upload Flow

- Form fields: Name, Patient ID, Email, Note, Upload Image
- Save file locally or on S3
- Create Submission record with details and status=uploaded

3) Admin – Review & Annotation

- Dashboard: list all submissions with details -
  View page: show original image + annotation canvas (shapes: rectangle, circle, arrow, freehand) -
  Save annotation: store JSON + flattened annotated image
- Status → annotated

4) PDF Report Generation

- Generate PDF: include patient details, annotated image, upload date/time, notes
- Store PDF locally or on S3
- Update DB with reportUrl, status=reported
- Patient can download their PDF

- For your reference, we have attached a sample patient images and a demo report PDF to illustrate the expected submission flow and output format.

- **“This demo report is only for reference. Your generated PDF should at minimum contain patient details, the original + annotated image, and structured notes.”**

Bonus Features

- Store all files in AWS S3
- Provide report links in PDF

Tech Stack (Suggested)

- Backend: Node.js, Express, MongoDB
- Frontend: React
- Auth: JWT or HTTP-only cookies
- PDF: pdfkit, pdf-lib, or puppeteer
- Storage: Local or S3
- Security: use .env for secrets

Acceptance Criteria

1) Auth works with role-based access
2) Patient upload creates DB record + stores image
3) Admin can annotate and save annotation
4) PDF generation works with embedded image and details
5) Patient can download report
6) Clear UX/UI
7) README explains setup
8) Bonus if S3 integration works

Submission Checklist

- GitHub repo (frontend + backend)
- README with setup instructions
- API list & sample requests
- Screenshots/GIFs
- Hosted demo link
- Test credentials