

প্রশ্ন ১: তোমার PensionProBD ওয়েব অ্যাপের আর্কিটেকচার কেমন?

উত্তর: এটি একটি রোল-ভিত্তিক থ্রি-টিয়ার ওয়েব অ্যাপ।

- **Frontend:** React + Tailwind CSS
- **Backend:** Node.js + Express
- **Database:** MongoDB

সকল ইউজার অ্যাকশন API এর মাধ্যমে পরিচালিত হয়। Authentication JWT টোকেন দিয়ে করা হয়েছে।

রিয়ল টাইম নোটিফিকেশনের জন্য Socket.IO ব্যবহার করেছি।

ডকুমেন্ট আপলোড নিরাপদভাবে ক্লাউড (যেমন AWS S3) এ রাখা হয়।

ফ্রন্টএন্ড (React)

প্রশ্ন ২: ইউজার রোল অনুযায়ী রাউটিং কীভাবে কাজ করে?

উত্তর: লগইনের সময় ব্যাকএন্ড থেকে JWT টোকেনের সঙ্গে ইউজারের role পাওয়া যায়। React রাউটার Protected Route ব্যবহার করে সেই রোল অনুযায়ী ড্যাশবোর্ডে রিডাইরেক্ট করে —

যেমনঃ Pension Holder → /pension/dashboard,

Assistant Accountant General → /admin/aag-dashboard,

Head of Office → /admin/head-dashboard।

প্রশ্ন ৩: মাল্টি-ল্যাঙ্গুয়েজ (বাংলা/ইংরেজি) কীভাবে ইমপ্লিমেন্ট করেছো?

উত্তর: আমি react-i18next লাইব্রেরি ব্যবহার করেছি।

ভাষা টগল বাটন টপবারে আছে। ইউজার যে ভাষা সিলেক্ট করে সেটি localStorage-এ সংরক্ষিত হয়, এবং পুরো UI ও ফর্ম লেবেল সেই অনুযায়ী পরিবর্তিত হয়।

প্রশ্ন ৪: মাল্টি-স্টেপ ফর্ম কীভাবে বানিয়েছো?

উত্তর: React state দিয়ে প্রতিটি ধাপ আলাদা রাখা হয়েছে (যেমন personal info → job info → document upload)।

প্রতিটি ধাপে ভ্যালিডেশন হয় (Yup লাইব্রেরি দিয়ে)।

শেষ ধাপে সব ডেটা একত্রে ব্যাকএন্ডে পাঠানো হয়।

ব্যাকএন্ড (Node.js + Express)

প্রশ্ন ৫: গুরুত্বপূর্ণ API endpoint গুলো কী কী?

উত্তর:

- POST /api/auth/register/pension → পেনশন হোল্ডার রেজিস্ট্রেশন
 - POST /api/auth/login → লগইন (JWT সহ)
 - POST /api/pensions → নতুন পেনশন ফর্ম সাবমিট
 - PUT /api/pensions/:id → আপডেট / কমেন্ট
 - POST /api/complaints → অভিযোগ সাবমিট
 - PUT /api/users/:id/flag → রেড ফ্ল্যাগ যোগ
 - GET /api/notifications → নোটিফিকেশন ফেচ
-

প্রশ্ন ৬: রোল-বেসড এক্সেস কন্ট্রোল (RBAC) কীভাবে কাজ করে?

উত্তর: ব্যাকএন্ডে Middleware JWT টোকেন যাচাই করে। তারপর req.user.role দেখে নির্দিষ্ট রোল ছাড়া অন্য কেউ অ্যাক্সেস পায় না। যেমন Pension Holder অফিসারদের API অ্যাক্সেস করতে পারবে না।

ডেটাবেস (MongoDB)

প্রশ্ন ৭: প্রধান কালেকশনগুলো কী কী?

উত্তর:

- users → নাম, ইমেইল, রোল, স্ট্যাটাস, ফ্ল্যাগ সংখ্যা
 - pensions → ফর্ম ডেটা, হোল্ডার আইডি, স্ট্যাটাস, কমেন্ট, ডকুমেন্ট
 - complaints → অভিযোগের তথ্য
 - notifications → রিয়েল টাইম আপডেট
 - audit_logs → প্রতিটি পরিবর্তনের ইতিহাস
-

প্রশ্ন ৮: পেনশন অ্যামাউন্ট বা ফাইনাল রিপোর্ট কিভাবে সুরক্ষিত রাখো?

উত্তর: ফাইনাল রিপোর্ট তৈরি হলে সেটি সার্ভার সাইডে PDF হিসেবে জেনারেট হয় এবং DB তে রেকর্ড করা হয়। একবার অনুমোদন হয়ে গেলে সেটি আর পরিবর্তন করা যায় না।

ফাইল আপলোড ও সিকিউরিটি

প্রশ্ন ৯: NID বা জব ডকুমেন্ট কীভাবে সিকিউরলি সংরক্ষণ করো?

উত্তর:

ডকুমেন্ট ক্লাউড স্টোরেজে (যেমন AWS S3) আপলোড হয় **Presigned URL** ব্যবহার করে।

ডাটাবেজে শুধু ফাইল নাম, URL ও আপলোড টাইম সংরক্ষণ করা হয়।

ফাইল ডাউনলোড করার সময় সার্ভার নতুন short-lived signed URL তৈরি করে, যাতে অন্য কেউ এক্সেস না পায়।

প্রশ্ন ১০: কী ধরনের ফাইল আপলোড করা যায়?

উত্তর: শুধু PDF, JPG, PNG ফাইল আপলোড করা যায় এবং সাইজ 10MB এর বেশি হলে ব্লক হয়।

রিয়েল টাইম নোটিফিকেশন

প্রশ্ন ১১: রিয়েল টাইম আপডেট কিভাবে কাজ করে?

উত্তর: Socket.IO ব্যবহার করেছি। যখন অফিসার কোনো অ্যাকশন নেয় (যেমন Approved বা Comment), তখন সার্ভার Pension Holder-এর socket ID-তে ইভেন্ট পাঠায়। ফ্রন্টএন্ডে টোস্ট/পপআপ আসে সাথে সাথে।

অনুমোদন ও রেড ফ্ল্যাগ সিস্টেম

প্রশ্ন ১২: পেনশন ফর্মের অ্যাপ্রভাল প্রক্রিয়া ব্যাখ্যা করো।

উত্তর:

- 1** Pension Holder ফর্ম সাবমিট করে → **Pending**
 - 2** Assistant Accountant General রিভিউ করে → সঠিক হলে Head of Office-এ ফরওয়ার্ড করে, না হলে কमेंট দেয়
 - 3** Head of Office ফাইনাল অ্যাপ্রভ করে → ফাইনাল PDF জেনারেট হয়
-

প্রশ্ন ১৩: Red Flag সিস্টেম কীভাবে কাজ করে?

উত্তর: Head যদি কোনো অফিসারের বিরুদ্ধে অভিযোগে সত্যতা পায়, তবে ১টি Red Flag যোগ হয়।

যখন কোনো অফিসারের ৩টি ফ্ল্যাগ হয়, তখন স্বয়ংক্রিয়ভাবে তার অ্যাকাউন্ট **Disabled** হয়ে যায়।

সিকিউরিটি

প্রশ্ন ১৪: Authentication কীভাবে সুরক্ষিত রেখেছে?

উত্তর: Access Token (JWT) ১৫ মিনিটের জন্য বৈধ এবং Refresh Token ৭ দিনের জন্য।

Refresh Token HTTP-only কুকিতে থাকে, তাই জাভাস্ক্রিপ্ট এক্সেস করা যায় না।

Rate Limiting ও Input Validation আছে।

প্রশ্ন ১৫: অন্য ইউজারের ডেটা কেউ যেন না পায় সেটা কীভাবে নিশ্চিত করো?

উত্তর: প্রতিটি রিকোয়েস্টে ব্যাকএন্ড চেক করে pension.holderId === req.user.id কিনা।

অর্থাৎ এক ইউজার অন্য ইউজারের পেনশন ফর্ম বা ডকুমেন্ট দেখতে পারবে না।

ভ্যালিডেশন ও বিজনেস রুল

প্রশ্ন ১৬: job_age 19 বছরের কম হলে কী হয়?

উত্তর: ক্লায়েন্ট এবং সার্ভার উভয় দিকেই চেক করা হয়।

যদি ১৯ বছরের নিচে হয়, সার্ভার সরাসরি **Bad Request (400)** দেয় এবং ফর্ম সাবমিট হয় না।

প্রশ্ন ১৭: ৩ দিনের মধ্যে ভুল থাকলে Rejection সিস্টেম কীভাবে কাজ করে?

উত্তর: AG (Accountant General) ৩ দিনের মধ্যে রিজেক্ট করতে পারে। যদি ৩ দিনের পরে হয়, Head of Office-এর অনুমতি ছাড়া রিজেক্ট করা যায় না।

টেস্টিং

প্রশ্ন ১৮: টেস্টিং কীভাবে করেছে?

উত্তর:

- **Unit Test:** Jest দিয়ে Authentication ও Pension Calculation ফাংশন টেস্ট করেছি।

- **Integration Test:** Supertest দিয়ে API endpoint টেস্ট করেছি।
 - **E2E Test:** React Testing Library এবং Cypress দিয়ে পুরো ইউজার ফ্লো টেস্ট করেছি।
-

ডিপ্লয়মেন্ট ও স্কেলিং

প্রশ্ন ১৯: ডিপ্লয়মেন্ট কিভাবে করেছো?

উত্তর: Docker ব্যবহার করেছি।

Frontend, Backend এবং Notification Worker আলাদা কন্টেইনারে চলে।

MongoDB ক্লাউড (MongoDB Atlas) এ হোস্ট করা।

CI/CD পাইপলাইন দিয়ে কোড অটো ডিপ্লয় হয়।

প্রশ্ন ২০: স্কেলিং কিভাবে করবে?

উত্তর: সার্ভার স্টেটলেস রাখায় একাধিক ইনস্ট্যান্স চালানো যায়।

লোড ব্যালেন্সার ব্যবহার করে ট্রাফিক ভাগ করা হয়।

Socket.IO স্কেল করার জন্য Redis অ্যাডাপ্টার ব্যবহার করা যায়।

প্রাইভেসি ও অ্যাক্সেসিবিলিটি

প্রশ্ন ২১: ইউজারের ব্যক্তিগত তথ্য সুরক্ষা কিভাবে রাখো?

উত্তর: শুধুমাত্র প্রয়োজনীয় ডেটা সংরক্ষণ করি।

NID ফাইল এনক্রিপ্টেড স্টোরেজে রাখা হয়।

অ্যাক্সেস লগ রেকর্ড করা হয়, যাতে কে কখন অ্যাক্সেস করেছে সেটা ট্র্যাক করা যায়।

প্রশ্ন ২২: বাংলা ও ইংরেজি দুই ভাষায় এক্সেসিবল কিভাবে করেছো?

উত্তর: i18n সিস্টেমে সব টেক্সট অনুবাদ করা আছে।

ফন্ট, তারিখ ফরম্যাট, এবং বাটন টেক্সট উভয় ভাষায় পরিবর্তিত হয়।

Light/Dark মোডও আছে যাতে ভিজ্যুয়ালি আরামদায়ক হয়।

মনিটরিং ও পারফরম্যান্স

প্রশ্ন ২৩: পারফরম্যান্স অপটিমাইজেশনের জন্য কী করেছো?

উত্তর:

- React কোড স্প্লিটিং
 - পেজিনেশন ও সার্ভার-সাইড ফিল্টারিং
 - MongoDB ইনডেক্স ব্যবহার
 - Lazy Loading
 - অপয়োজনীয় রিকোয়েস্ট কমানো
-

প্রশ্ন ২৪: সার্ভার মনিটরিং কিভাবে করো?

উত্তর: লগ গুলো JSON আকারে সংরক্ষণ হয়।

এরর রোট বা সার্ভার ডাউন হলে অ্যালার্ট পাঠানো হয়।

Grafana বা ELK Stack দিয়ে রিয়েল টাইম মনিটরিং করা যায়।

UI/UX ও ডিজাইন

প্রশ্ন ২৫: “Bangladeshi Cultural UI” বলতে কী বোঝাও?

উত্তর: ওয়েবসাইটে বাংলাদেশের সবুজ-লাল কালার স্কিম ব্যবহার করেছি।

ফন্ট, আইকন ও ব্যাকগ্রাউন্ডে সরকারি ওয়েবসাইটের (gov.bd) ধাঁচ বজায় রেখেছি।

ডিজাইনে Digital Bangladesh ব্র্যান্ডিং যুক্ত করেছি।

প্রশ্ন ২৬: Light/Dark মোড কিভাবে কাজ করে?

উত্তর: Tailwind এর Dark Mode ক্লাস ব্যবহার করেছি।

ইউজার পছন্দ অনুযায়ী মোড localStorage এ সংরক্ষিত হয়, পরের বার ভিজিট করলে আগের সেটিং বজায় থাকে।

লাইভ ডেমো / ভাইভা টাইপ প্রশ্ন

প্রশ্ন ২৭: এখন দেখাও – রেজিস্টার থেকে ফাইনাল রিপোর্ট পর্যন্ত পুরো ফ্লো কেমন?

উত্তর:

- 1 Pension Holder রেজিস্ট্রেশন → লগইন → ফর্ম পূরণ ও ডকুমেন্ট আপলোড।
- 2 AG লগইন করে ফর্ম যাচাই করে।
- 3 সঠিক হলে Head-এর কাছে ফরওয়ার্ড করে, ভুল হলে কमेंট দেয়।

- 4 Pension Holder ফিডব্যাক ঠিক করে আবার সাবমিট করে।
 - 5 Head of Office ফাইনাল অ্যাপ্রভ করলে ফাইনাল PDF রিপোর্ট ডাউনলোড করা যায়।
-

প্রশ্ন ২৮: রেড ফ্ল্যাগ সিস্টেম লাইভ দেখাও।

উত্তর: Head যদি কোনো অফিসারকে ৩ বার Flag দেয়, তখন তার অ্যাকাউন্ট অটো **Disabled** হয়ে যায়।

Login করলে “Account Disabled” মেসেজ আসে।

নন-টেকনিক্যাল প্রশ্ন

প্রশ্ন ২৯: কারা এই সিস্টেমের মূল ইউজার?

উত্তর:

- Pension Holder (অবসরপ্রাপ্ত কর্মচারী)
 - Assistant Accountant General
 - Head of Office
-

প্রশ্ন ৩০: এই সিস্টেমের উপকারিতা কী?

উত্তর: ম্যানুয়াল কাগজপত্র কমায়, দ্রুত পেনশন প্রক্রিয়া সম্পন্ন হয়, অভিযোগ ও ট্র্যাকিং সহজ হয়, এবং দুর্নীতি কমে যায় কারণ প্রতিটি কাজ লগ হয়।

প্রশ্ন ৩১: ডেভেলপমেন্টের সময় কী কী চ্যালেঞ্জ ছিল?

উত্তর: ফাইল সিকিউরিটি, রিয়েল টাইম আপডেট, মাল্টি-ল্যাঙ্গুয়েজ হ্যান্ডলিং, এবং রেড ফ্ল্যাগ অটো সিস্টেম ইমপ্লিমেন্ট করাই সবচেয়ে বড় চ্যালেঞ্জ ছিল।

প্রশ্ন ৩২: ভবিষ্যতে কী কী ফিচার যুক্ত করতে চাও?

উত্তর:

- ই-সিগনেচার সিস্টেম
- NID API ইন্টিগ্রেশন
- মোবাইল অ্যাপ

- OTP-ভিত্তিক লগইন
 - পেনশন ক্যালকুলেশনের AI সাজেশন সিস্টেম
-

প্রশ্ন ৩৩: এই প্রজেক্টে তোমার ভূমিকা কী ছিল?

উত্তর: আমি ফুল-স্ট্যাক ডেভেলপার হিসেবে কাজ করেছি।

UI ডিজাইন, React ফ্রন্টএন্ড, Node.js ব্যাকএন্ড, MongoDB মডেল, এবং রোল-বেসড লজিক — সব আমি তৈরি করেছি।

প্রশ্ন ৩৪: এই সিস্টেমের নাম PensionProBD কেন রেখেছো?

উত্তর: “Pension” মানে অবসর ভাতা, আর “ProBD” মানে Professional Bangladesh — অর্থাৎ বাংলাদেশের জন্য পেশাদার পেনশন সিস্টেম।