# Capstone Project: Insurance & Reinsurance Policy and Claims Management System

## Business Context (Real-World)

Insurance companies underwrite policies and manage claims. Reinsurance companies share risk when exposure exceeds a threshold.

This project simulates:

- Policy issuance
- Claim lifecycle
- Automatic reinsurance allocation
- Risk exposure tracking

---

## Core Problem Statement

Build a system that manages insurance policies and claims, and automatically transfers part of the risk to reinsurers when exposure crosses predefined limits.

---

## High-Level Architecture

Angular (UI)

|

Node.js (API Gateway)

|

---

| Policy Service |

| Claims Service |

| Reinsurance Engine |

| User & Auth Service |

---

|

MongoDB (Collections)

# Core Modules

## User & Role Management

**Roles**

- Underwriter
- Claims Adjuster
- Reinsurance Manager
- Admin

**Features**

- JWT authentication
- Role-based access control (RBAC)

---

## Policy Management Module

**Policy Attributes**

- Policy Number (Auto-generated)
- Policyholder Details
- Policy Type (Health, Motor, Life, Property)
- Sum Insured
- Premium
- Risk Category
- Policy Status

**Key Logic**

- Exposure calculation
- Policy approval workflow

---

## Claims Management Module

**Claim Lifecycle**

Submitted → Under Review → Approved → Paid / Rejected

**Features**

- Claim amount validation

- Policy coverage checks
- Fraud flagging (rule-based)

---

## Reinsurance Allocation Engine ⭐ (Key Differentiator)

**Business Rule Example**

If Sum Insured > ₹50,00,000

→ 40% risk transferred to Reinsurer A

→ 20% risk transferred to Reinsurer B

**Capabilities**

- Treaty-based reinsurance
- Proportional risk sharing
- Exposure aggregation per reinsurer

---

## Financial & Exposure Dashboard

**Dashboards**

- Total exposure by policy type
- Claims ratio (Claims / Premium)
- Reinsurer-wise risk distribution
- Loss ratio trends

---

# MongoDB Data Modelling

## Collections

### users Collection

Represents **internal system users** (not customers).

```
{
  _id: ObjectId,
  username: String,
  email: String,
  passwordHash: String,
  role: "UNDERWRITER" | "CLAIMS_ADJUSTER" | "REINSURANCE_ANALYST" | "ADMIN",
```

permissions: [String],

status: "ACTIVE" | "INACTIVE",

lastLoginAt: Date,

createdAt: Date,

updatedAt: Date

}

## policies Collection

Stores **current version of a policy**.

{

_id: ObjectId,

policyNumber: String,

insuredName: String,

insuredType: "INDIVIDUAL" | "CORPORATE",

lineOfBusiness: "HEALTH" | "MOTOR" | "LIFE" | "PROPERTY",

sumInsured: Number,

premium: Number,

retentionLimit: Number,

status: "DRAFT" | "ACTIVE" | "SUSPENDED" | "EXPIRED",

effectiveFrom: Date,

effectiveTo: Date,

createdBy: ObjectId,

approvedBy: ObjectId,

createdAt: Date,

updatedAt: Date

}

**Notes**

- Endorsements update this record
- History goes to audit_logs

## claims Collection

Manages **claim lifecycle**.

{

```
  _id: ObjectId,
  claimNumber: String,
  policyId: ObjectId,
  claimAmount: Number,
  approvedAmount: Number,
  status: "SUBMITTED" | "IN_REVIEW" | "APPROVED" | "REJECTED" | "SETTLED",
  incidentDate: Date,
  reportedDate: Date,
  handledBy: ObjectId,
  remarks: String,
  createdAt: Date,
  updatedAt: Date
}
```

## reinsurers Collection

Master data for reinsurers.

```
{
  _id: ObjectId,
  name: String,
  code: String,
  country: String,
  rating: "AAA" | "AA" | "A" | "BBB",
  contactEmail: String,
  status: "ACTIVE" | "INACTIVE",
  createdAt: Date,
  updatedAt: Date
}
```

---

## treaties Collection

Defines **reinsurance agreements**.

```
{
  _id: ObjectId,
```

```
  treatyName: String,
  treatyType: "QUOTA_SHARE" | "SURPLUS",
  reinsurerId: ObjectId,
  sharePercentage: Number,
  retentionLimit: Number,
  treatyLimit: Number,
  applicableLOBs: ["HEALTH", "MOTOR"],
  effectiveFrom: Date,
  effectiveTo: Date,
  status: "ACTIVE" | "EXPIRED",
  createdAt: Date,
  updatedAt: Date
}
```

---

## risk_allocations Collection

Stores **calculated reinsurance splits per policy**.

```
{
  _id: ObjectId,
  policyId: ObjectId,
  allocations: [
    {
      reinsurerId: ObjectId,
      treatyId: ObjectId,
      allocatedAmount: Number,
      allocatedPercentage: Number
    }
  ],
  retainedAmount: Number,
  calculatedAt: Date,
  calculatedBy: ObjectId
}
```

---

### audit_logs Collection

Tracks **who did what and when**.

```
{
  _id: ObjectId,
  entityType: "POLICY" | "CLAIM" | "TREATY" | "USER",
  entityId: ObjectId,
  action: "CREATE" | "UPDATE" | "DELETE" | "APPROVE",
  oldValue: Object,
  newValue: Object,
  performedBy: ObjectId,
  performedAt: Date,
  ipAddress: String
}
```

---

## Security & Compliance

- JWT + Refresh tokens
- Encrypted sensitive fields
- Audit trail for policy & claim updates
- Soft deletes (regulatory requirement)

---

# FRONTEND ARCHITECTURE

## TOP-LEVEL REACT STRUCTURE

```
src/
 ├— app/
|   ├— layout/
|   ├— routes/
|   ├— features/
|   ├— shared/
```

| ├── hooks/

| └── services/

---

## LAYOUT COMPONENTS

**Core Layout Components**

- AppShell
- Sidebar
- TopBar
- Breadcrumbs
- NotificationCenter
- Unauthorized

**Responsibilities**

- Role-based navigation
- Persistent layout
- Global feedback & alerts

---

## ROUTING & SECURITY COMPONENTS

**Routing Components**

- AppRoutes
- ProtectedRoute
- RoleRoute

**Responsibilities**

- Authentication checks
- Role-based access control
- Read-only vs edit access

---

## AUTH & USER CONTEXT COMPONENTS

**Auth Components**

- LoginPage
- LogoutHandler

**Context / Providers**

- AuthProvider
- PermissionProvider

**Purpose**

- User identity
- Role awareness
- Permission checks at component level

---

## POLICY MODULE – REACT COMPONENTS

**Policy Feature Folder**

features/policy/

├— PolicyList

├— PolicyDetails

├— CreatePolicyWizard

├— PolicyStepGeneral

├— PolicyStepCoverage

├— PolicyStepReview

├— PolicyStatusBadge

├— PolicyActions

├— PolicyHistory

**Component Responsibilities**

- CreatePolicyWizard
  → Multi-step form (enterprise standard)
- PolicyStepCoverage
  → Conditional fields & validations
- PolicyActions
  → Approve / Reject / Endorse (role-aware)
- PolicyHistory
  → Versioning timeline (very enterprise)

---

## CLAIMS MODULE – WORKFLOW-DRIVEN COMPONENTS

**Claims Feature Folder**

features/claims/

├— ClaimsList

├— ClaimDetails

├— ClaimCreateForm

├— ClaimStatusTimeline

├— ClaimActionPanel

├— ClaimNotes

├— ClaimDocuments

**Component Responsibilities**

- ClaimStatusTimeline
  → Visual workflow state machine
- ClaimActionPanel
  → Dynamic buttons based on role + status
- ClaimNotes
  → Collaborative enterprise feature

---

## REINSURANCE MODULE

**Reinsurance Feature Folder**

features/reinsurance/

├— TreatyList

├— TreatyForm

├— RiskAllocationView

├— AllocationTable

├— AllocationSummary

├— AllocationValidation

**Key Enterprise Components**

- AllocationTable
  → Editable grid (real insurance UI)
- AllocationValidation
  → Treaty limit checks
- AllocationSummary
  → Insurer vs reinsurer exposure split

---

## ADMIN & CONFIGURATION COMPONENTS

**Admin Feature Folder**

features/admin/

├— UserList

├— RoleMatrix

├— PermissionEditor

├— TreatyConfiguration

├— ThresholdConfig

---

## SHARED / REUSABLE COMPONENTS

**Shared Components**

shared/

├— DataTable

├— StatusBadge

├— ConfirmDialog

├— EmptyState

├— ErrorState

├— Loader

├— FormField

├— CurrencyInput

---

## STATE & LOGIC

**Hooks**

- useAuth
- usePermissions
- usePolicies
- useClaims
- useReinsurance

**Services**

- apiClient
- policyService

- claimsService
- reinsuranceService

---

## ERROR & EDGE-CASE COMPONENTS

- ErrorBoundary
- ApiErrorFallback
- AccessDenied
- SessionExpiredModal

---