# **Capstone Project 2: Risk Exposure Tracking & Reporting System**

# **Domain: Risk Management**

## **Problem Statement**

Consulting firms and insurers need to monitor client risk exposures across categories such as credit, market, operational, and compliance risk. Traditionally, these are tracked via spreadsheets or manual reports, which are prone to errors and lack real-time insights. Without a centralized system, clients cannot assess overall exposure or track mitigation actions, leading to poor decision-making and regulatory gaps.

This project builds a **Risk Exposure Tracking & Reporting System** where clients can:

* Record different categories of risks.
* View aggregated exposures.
* Monitor mitigation actions.
* Generate risk trend reports.
* Manage organizational profiles.

## **Feature 1: Risk Recording**

### **Backend**

* POST /api/risks — Add a new risk record.
* GET /api/risks/{orgId} — Fetch risks by organization.

### **Database Schema**

**Table: Risks**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| riskId | BIGINT (PK) | Unique risk identifier |
| orgId | BIGINT (FK) | Organization owner |
| category | VARCHAR(50) | Credit, Market, Operational, etc. |
| description | VARCHAR(200) | Risk description |
| exposure | DECIMAL(12,2) | Exposure amount |
| status | VARCHAR(20) | Open, Mitigated, Closed |
| createdAt | DATETIME | Creation timestamp |

### **Frontend**

* RiskForm — Add/edit risks.
* RiskList — Display risks with filters by category/status.

### **Deployment**

**Offline:**

* Backend runs via dotnet run.
* Local SQL DB stores risk records.
* Frontend runs via npm start and accessed in browser.

**Cloud (Optional):**

* API hosted on Azure App Service.
* Frontend hosted on Azure Static Web Apps.
* Database on Azure SQL.

## **Feature 2: Exposure Aggregation**

### **Backend**

* GET /api/exposures/{orgId} — Summarize exposure by category.

### **Database Schema**

**Table: ExposureSummary**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| summaryId | BIGINT (PK) | Summary record ID |
| orgId | BIGINT (FK) | Organization |
| category | VARCHAR(50) | Risk category |
| totalAmt | DECIMAL(12,2) | Total exposure amount |
| lastUpdate | DATETIME | Last updated timestamp |

### **Frontend**

* ExposureDashboard — Chart view by category.

### **Deployment**

**Offline:**

* Backend runs via dotnet run.
* Local SQL DB performs aggregation queries.
* Frontend runs via npm start and accessed in browser.

**Cloud (Optional):**

* API hosted on Azure App Service.
* Frontend hosted on Azure Static Web Apps.
* Database on Azure SQL with precomputed aggregates.

## **Feature 3: Mitigation Tracking**

### **Backend**

* POST /api/mitigations — Log mitigation action.
* GET /api/mitigations/{riskId} — Fetch actions per risk.

### **Database Schema**

**Table: Mitigations**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| mitigationId | BIGINT (PK) | Unique mitigation identifier |
| riskId | BIGINT (FK) | Related risk |
| action | VARCHAR(200) | Mitigation step description |
| owner | VARCHAR(100) | Person responsible |
| deadline | DATE | Target completion date |
| status | VARCHAR(20) | Open, Completed |

### **Frontend**

* MitigationList — Display and track actions.

### **Deployment**

**Offline:**

* Backend runs via dotnet run.
* Local SQL DB stores mitigation steps linked to risks.
* Frontend runs via npm start and accessed in browser.

**Cloud (Optional):**

* API hosted on Azure App Service.
* Frontend hosted on Azure Static Web Apps.
* Database on Azure SQL with secured connections.

## **Feature 4: Risk Trend Reports**

### **Backend**

* GET /api/reports/{orgId} — Generate risk trends by date range.

### **Database Schema**

**Table: RiskReports**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| reportId | BIGINT (PK) | Unique report ID |
| orgId | BIGINT (FK) | Organization |
| period | VARCHAR(20) | Monthly/Quarterly |
| metrics | VARCHAR(500) | JSON/summary metrics |
| createdAt | DATETIME | Report timestamp |

### **Frontend**

* RiskReport — Chart/table visualization of risk trends.

### **Deployment**

**Offline:**

* Backend runs via dotnet run.
* Local SQL DB generates reports.
* Frontend runs via npm start and accessed in browser.

**Cloud (Optional):**

* API hosted on Azure App Service.
* Frontend hosted on Azure Static Web Apps.
* Database on Azure SQL.
* Optional integration with Power BI for dashboards.

## **Feature 5: Organization Profiles**

### **Backend**

* GET /api/orgs/{orgId} — Fetch organization details.
* PUT /api/orgs/{orgId} — Update profile.

### **Database Schema**

**Table: Organizations**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| orgId | BIGINT (PK) | Organization ID |
| name | VARCHAR(100) | Organization name |
| sector | VARCHAR(50) | Industry sector |
| region | VARCHAR(50) | Region |
| contact | VARCHAR(100) | Contact person |
| email | VARCHAR(100) | Contact email |

### **Frontend**

* OrgProfile — Manage organization details.

### **Deployment**

**Offline:**

* Backend runs via dotnet run.
* Local SQL DB stores organization profiles.
* Frontend runs via npm start and accessed in browser.

**Cloud (Optional):**

* API hosted on Azure App Service.
* Frontend hosted on Azure Static Web Apps.
* Database on Azure SQL with secured connections.