**Insider-Threat Detection Dashboard – Full Technical Overview**

## ✅ What Is This Project About?

A **real-time internal security dashboard** designed to monitor, detect, and visualize suspicious activity in small-to-medium (SMB) environments. It tracks events like file deletions and unusually large downloads, calculates a risk score per user, and surfaces high-risk behavior through live alerts and visual charts.

## 🌟 Why Build It?

* **Rising demand**: With remote work on the rise, insider threats—whether accidental or malicious—are becoming a serious risk.
* **Skill showcase**: Demonstrates integration of real-time data pipelines, backend API services, data storage, machine learning logic, and frontend visualization.
* **Real-world utility**: Goes beyond toy projects by addressing actual concerns in enterprise cybersecurity.

## 🌞 What’s Innovative?

* **Live risk alerts**: Automatically flags critical behaviors (e.g., delete or huge file size), notifying users via an alert banner.
* **Dynamic risk scoring**: Calculates user risk based on behavioral weights (e.g., delete = +3, download = +1).
* **User-friendly dashboard**: Combines table views with a D3.js-powered bar chart to visualize risk per user.
* **Full-stack real-time flow**: Simulates Kafka-like data ingestion, real-time API ingestion, and frontend polling.

## ⚙️ How It Works

1. **Behavior generator**: A Python Kafka-simulator (producer.py) sends logs every few seconds.
2. **FastAPI backend**:
   * Accepts POSTed behavior logs via /api/behavior.
   * Stores them in PostgreSQL (insider\_db.behavior\_logs table).
   * Supports CORS, enabling React frontend to fetch logs.
3. **Frontend in React**:
   * Presents a login page (hardcoded admin/admin123).
   * Once authenticated, polls /api/behavior every ~10 seconds.
   * Presents:
     + A **table** of recent behaviors.
     + A **risk bar chart** showing risk scores per user.
     + A **live alert banner** for recent risky actions.
4. **Risk logic**:
   * delete events add +3 to user risk; download +1.
   * file\_size > 750 KB flagged instantly.
   * Top 3 recent risky events displayed as alert banners.

## 📦 Technology Stack

* **Frontend**:
  + **React.js** (JavaScript, functional components, Hooks)
  + **Tailwind CSS** for quick, mobile-first layout
  + **D3.js** for interactive, animated bar chart
* **Backend**:
  + **FastAPI** (Python) — high-speed API, auto docs
  + **SQLAlchemy ORM** connecting to PostgreSQL
* **Database**:
  + **PostgreSQL** 17 — local instance with insider\_db
* **Simulator**:
  + Python script emulating Kafka — posts directly to backend API
* **Development Environment**:
  + Docker (optional), uvicorn, Python 3.x, Node.js & npm
* **Infrastructure**:
  + Local environment; can be containerized or deployed on cloud VMs
* **Future scope**:
  + Real Kafka + Kafka Streams
  + Kubernetes, Dockerfile, CI/CD (GitHub Actions)
  + Authentication, RBAC, analytics dashboard

## 🧰 Project Components & Folder Structure

/insider-threat-ai-dashboard  
 ├─ backend/  
 │ ├ main.py # FastAPI endpoints, DB definitions  
 │ └ kafka-simulator/producer.py  
 └─ frontend/  
 ├ src/  
 │ ├ App.js # Main React logic  
 │ ├ RiskChart.js  
 │ ├ RiskAlert.js  
 │ ├ Login.js  
 │ ├ index.js # Entry point  
 │ └ styles/... # Tailwind config

## 💪 Setup & Run (Local)

1. **Database**:

* CREATE DATABASE insider\_db;

1. **Backend**:

* pip install fastapi uvicorn sqlalchemy psycopg2  
  uvicorn backend.main:app --reload

1. **Frontend**:

* cd frontend  
  npx create-react-app . # or yarn  
  npm install tailwindcss d3  
  npm start

1. **Simulator**:

* python backend/kafka-simulator/producer.py

## 🥇 Demonstration Flow

* Open React UI → click **Log In** → enter admin/admin123
* After login:
  + **Behavior Table** displays recent actions
  + **Alerts** pop up for delete actions or file\_size > 750KB
  + **Bar Chart** shows risk score evolving live

## 📊 Future Enhancements

* WebSocket + Kafka for push-driven updates
* User-level authentication and roles (admin/analyst/etc.)
* Automated deployment (Docker, cloud)
* Enhanced ML: anomaly detection via LSTM or Isolation Forest
* Audit log visualization, risk trend analysis

## 🎓 Skills & Value Demonstrated

* Full-stack architecture (frontend ↔ backend ↔ DB)
* Real-time event streaming and processing
* Data visualization (D3.js)
* Typed, high-performance APIs (FastAPI)
* Clean modern styling (Tailwind CSS)
* Version control and deployment readiness