Secure Incident Reporting System (MERN Stack)

Objective:

Develop a **highly secure and scalable** cybersecurity **Incident Reporting System** with **advanced dashboard functionalities, filtering, logging, and role-based access control (RBAC)**.

🛠️ Features & Requirements

◇ Frontend

✅ **Authentication & Role-Based Access (RBAC)**
- **Users**: Can **submit incidents** and view only their own reports.
- **Admins**: Can **view, update, filter, and manage all incidents**.
- **Super Admin**: Can **manage users, assign roles, and delete incidents permanently**.

✅ **Advanced Dashboard (Role-Based Views)**
📌 **For Users**
- View their **submitted incidents** in a clean **table format**.
- **Search, sort, and filter incidents** by category, status, or date.
- Receive **real-time notifications** when an admin updates their incident status.

📌 **For Admins**
- **Analytics Dashboard** with the following widgets:
  - **Total Incidents** (Count of all incidents).
  - **Open vs Resolved Incidents** (Pie chart).
  - **Most Common Categories** (Bar chart: Phishing, Malware, Unauthorized Access, etc.).
  - **Average Resolution Time** (How long admins take to resolve issues).

- **Incident Filters & Sorting**
  - **Filter by Status** (Open, In Progress, Resolved).
  - **Filter by Category** (Phishing, Malware, Ransomware, etc.).
  - **Sort by Date, Severity, or User Reports**.

- **Bulk Actions**
  - Mark multiple incidents as “Resolved”.
  - Assign incidents to different admins.
  - Export incidents as **CSV or PDF**.

📌 **For Super Admin**

- **User Management System**
  - Add, edit, or delete users.
  - Assign roles (User, Admin, Super Admin).
  - Block or unblock users.
- **Audit Logs** (Track who created, modified, or deleted incidents).

✅ **Incident Reporting Form**

- **Fields:** Title, Description, Category (Dropdown), Priority (Low, Medium, High), Date, Evidence Upload (Screenshots, PDFs).
- **Form Validation** (Required fields, file size limit, accepted formats).

✅ **Real-time Updates & Notifications (WebSockets - Socket.io)**

- **Users get notifications** when an admin updates their incident.
- **Admins see real-time incident reports** without page refresh.

◇ Backend (Node.js + Express + MongoDB)

✅ **Authentication (JWT & Refresh Tokens)**
✅ **Role-Based Access Control (RBAC)**

✅ **Advanced Logging & Audit Trail**

- Track **who created, updated, or deleted incidents** with timestamps & IP addresses.
- Maintain **a separate collection for logs** in MongoDB.

Deliverables

1. **GitHub Repository** (with README & setup instructions).
2. **Presentation or Video Walkthrough**
