**Prompt for Building an AI-Powered Risk Management Web Application**

*(Structured in User Stories with Clear Requirements)*

**🧠 Goal**

Build a professional, AI-enhanced web application for **Risk Management** designed for project managers and operations teams to manage, update, monitor, and report risks across projects, with an elegant and interactive dashboard. The application must support AI-assisted risk creation and risk insights.

**👤 User Groups**

1. **Admin**
2. **Risk Manager**
3. **Project Manager**
4. **Viewer / Read-only User**

**🚀 Key Functionalities (User Stories)**

**1. Risk Register Management**

* As a *Project Manager*, I want to **create new risks** so that I can log all project-related uncertainties.
* As a *Project Manager*, I want to **edit or update existing risks** so I can keep data up to date.
* As a *Risk Manager*, I want to **delete irrelevant or resolved risks**.
* As a *Risk Manager*, I want to **view all risk registers filtered by project, owner, impact, and status**.
* As a *Viewer*, I want to **view the list of risks only, without editing access**.

**2. AI-Powered Risk Creation**

* As a *Project Manager*, I want to **generate suggested risks based on project description or uploaded documents using AI**, so I can discover hidden risks.
* As a *Risk Manager*, I want to **get AI-based risk mitigation suggestions** for each high or critical risk.
* As a *User*, I want the AI assistant to **classify risks by probability and impact automatically**.

**3. Risk Monitoring and Updates**

* As a *Risk Manager*, I want to **track each risk’s current status, mitigation progress, and action owner**.
* As a *Project Manager*, I want to **update the risk response strategies** and log risk-related events.
* As a *Viewer*, I want to **view real-time updates on risk heatmap and charts**.

**4. Elegant Dashboards and Reports**

* As an *Admin*, I want to **configure and customize dashboards** by selecting widgets (e.g., Risk Heatmap, Pie Chart by Category, Top 10 Risks).
* As a *User*, I want to **see a summary dashboard** that shows the number of active risks, risks by category, and trends over time.
* As a *Risk Manager*, I want to **export reports in PDF and Excel formats** for stakeholders.

**5. User Management and Access Control**

* As an *Admin*, I want to **create and manage user accounts with roles and permissions**.
* As a *User*, I want to **log in securely using email/OTP or organizational login**.
* As an *Admin*, I want to **track login/logout history and activity log** for auditing.

**6. Risk and Operations Integration**

* As a *Project Manager*, I want to **link each risk with a project task or operational activity**, so risks are contextualized.
* As a *Risk Manager*, I want to **see which risks are tied to which operational processes**.

**📊 Inputs & Outputs**

**Inputs:**

* Manual Risk Entry (Name, Type, Impact, Probability, Owner, Mitigation Plan)
* AI-assisted Risk Creation (via text prompt or document upload)
* Status Updates
* Comments / Attachments per Risk
* Risk Event Logs

**Outputs:**

* Dynamic Risk Register Table
* Risk Heatmap
* AI Risk Suggestions
* Dashboards (KPIs, Trends)
* Exported Reports (PDF, XLSX)
* Notifications & Alerts

**🎨 UI/UX Requirements**

* Clean, modern interface using cards and widgets
* Responsive design for desktop and mobile
* Left menu for navigation: Dashboard, Risk Register, AI Generator, Reports, Settings
* Color-coded risks based on severity
* Real-time loading and updates (React.js / Next.js preferred)

**⚙️ Backend Requirements**

* Role-Based Access Control (RBAC)
* Integration-ready RESTful APIs
* AI module integration for risk generation (e.g., OpenAI or custom LLM)
* Secure database (PostgreSQL / MongoDB)

**🧩 Optional Features**

* AI chatbot for answering questions about risks
* Email alerts for high severity risks
* Version control for risk edits
* Archive for closed risks

**Project Name Suggestion:** RiskAI Pro

**Development Stack Suggestion:**

* Frontend: React.js / Next.js
* Backend: Node.js / Python (FastAPI preferred)
* AI Integration: OpenAI API or fine-tuned LLM
* Database: PostgreSQL or MongoDB
* Hosting: AWS / Azure / Vercel