

Project Proposal: Risk Engine SaaS Platform

Risk Engine SaaS Platform

This document outlines the proposal for developing a multi-tenant SaaS-based Risk Engine designed for enterprise clients. The system will include user tracking, real-time risk analysis, AI/ML integration, and a robust admin dashboard. It is aimed to be completed as an MCA capstone project and later monetized as a scalable SaaS solution.

1. Core Idea

- Build a SaaS-based risk engine for enterprise use.
- Project will serve as MCA capstone and revenue-generating post-graduation.
- Supports user login, tracking, risk analysis, and admin dashboards.
- Uses Laravel for backend, integrated with C++ for high-performance scoring and Python for AI/ML.

2. Project Goals

- Launch MVP in 1.5 months.
- Target enterprise clients with multi-tenant capabilities.
- Collect pilot feedback and evolve into a commercial SaaS offering.

3. Technology Stack

- Backend: Laravel (PHP 8.x)
- Frontend: Vue.js + Laravel Mix
- AI/ML: Python (scikit-learn, TensorFlow)
- High Performance Engine: C++ (gRPC/HTTP microservice)
- Database: MySQL/PostgreSQL
- Cache: Redis

Project Proposal: Risk Engine SaaS Platform

- Logging & Search: Elasticsearch, Kibana
- Messaging: RabbitMQ or Kafka
- Containerization: Docker + Kubernetes or Compose
- CI/CD: GitHub Actions
- Monitoring: Prometheus + Grafana

4. Project Timeline (6 Weeks)

Week 1: Planning & Setup

- Define ERD, APIs, and user stories
- Setup Laravel, Docker, Auth

Week 2: User & Admin Module

- CRUD operations for users, roles
- Enterprise onboarding, dashboards

Week 3: Data Tracking

- Activity logger APIs
- Store events, build UI feed

Week 4: Risk Engine MVP

- C++ microservice (rule-based)
- Integrate with Laravel via gRPC

Week 5: AI/ML Integration

- Train anomaly models
- Connect Python inference API

Week 6: SaaS Finalization

- Multi-tenant, subscription, billing

Project Proposal: Risk Engine SaaS Platform

- Logging stack, testing, deployment

5. Features Overview

- Secure authentication (Laravel Sanctum)
- User activity tracking and feed
- Rule and ML-based risk scoring
- Real-time and historical dashboards
- Admin role management and controls
- Alerts, thresholds, audit logs, reports
- SaaS billing and tenant separation

6. AI/ML Integration

- Pipeline: Stream events > Queue > DB
- Models: Isolation Forest, Autoencoders, XGBoost
- Service: Dockerized Python app
- Retraining via feedback loop

7. Deployment & SaaS

- Infrastructure as Code (Terraform)
- CI/CD and container registry
- Stripe or Razorpay integration
- Tenant data isolation in DB

8. Diagrams (To be Added Later)

Project Proposal: Risk Engine SaaS Platform

- ER Diagram: User & Enterprise Flow
- Architecture Flow: Laravel <-> C++ <-> Python <-> DB <-> Frontend