**Financial Risk Analysis Tool: Hackathon Project Documentation**

**1. Project Overview**

**Objective**

Develop a web-based platform that integrates Monte Carlo Simulations (MCS) and predictive analytics to assess investment risks, delivering probabilistic insights (e.g., NPV distributions, risk probabilities) through an intuitive UI/UX for actionable decision-making.

**Hackathon Track**

"Financial Risk Analysis Tool" – Assessing investment risks using mathematical models like MCS and predictive analytics.

**Duration**

30 hours (March 25-26, 2025 assumed).

**Team**

4 members.

**2. Problem Statement**

**Definition**

Traditional financial risk assessment tools rely on static assumptions and manual processes, failing to capture the uncertainty and volatility of investment decisions in dynamic markets. Current MCS implementations are slow, limited in scope, and lack predictive capabilities to forecast key variables (e.g., revenues, costs), while standalone analytics struggle with data complexity, leaving investors with inaccurate risk insights.

**Impact**

* **Decision Accuracy**: Poor risk quantification risks financial losses or missed opportunities.
* **Efficiency**: Manual tools delay real-time or portfolio-level analysis.
* **Scalability**: Non-automated systems limit multi-user or multi-scenario support.
* **Relevance**: Lack of predictive adaptability reduces utility in 2025’s data-driven financial landscape.

**3. Suggested Solution**

**Overview**

We propose a web-based **Financial Risk Analysis Platform** that combines Monte Carlo Simulations (MCS) and predictive analytics to assess investment risks effectively. The platform will:

* Use MCS to simulate thousands of scenarios, generating probabilistic outcomes like NPV distributions and risk probabilities.
* Apply machine learning (e.g., linear regression) to forecast variables such as revenues or costs using mock data.
* Automate analysis via an intuitive UI/UX, ensuring rapid, scalable results.
* Feature a preloaded demo case (e.g., €40k startup investment) for instant risk insights.

This solution merges robust simulation with predictive forecasting, empowering investors with data-driven decisions.

**Key Features**

1. **Input Interface**: Clean form for investment details (e.g., initial cost, duration, revenue/cost ranges).
2. **MCS Engine**: Simulates outcomes, outputs NPV mean, std. dev., and P(NPV < 0).
3. **Predictive Analytics**: Forecasts variable trends (e.g., revenue growth).
4. **UI/UX**: Sleek dashboard with interactive charts (histogram, probability curve) and clear stats.
5. **Demo**: Preloaded €40,000 investment case for immediate demo impact.

**4. Tech Stack**

* **Frontend**: React.js, Chart.js
* **Backend**: Python, Flask
* **Monte Carlo Simulation**: NumPy, SciPy
* **Predictive Analytics**: scikit-learn
* **Data**: In-memory/CSV
* **Deployment**: Heroku

**5. Workflow**

1. **User Interaction**:
   * User accesses platform, selects demo or inputs custom data (e.g., €40k initial cost, 5 years, revenue €40k-€60k).
2. **Predictive Processing**:
   * ML model (linear regression) forecasts trends (e.g., +5% revenue growth) from mock data.
3. **Simulation**:
   * MCS runs thousands of trials using predicted inputs, calculates NPV and risk metrics.
4. **Output Display**:
   * UI renders interactive visuals (NPV histogram, probability curve) and stats (e.g., “90% chance NPV > 0”).
5. **Decision**:
   * User interprets results for investment action (e.g., “Invest!”).

**6. UI/UX Design**

**Principles**

* **Simplicity**: Minimalist design, easy navigation.
* **Interactivity**: Clickable charts, responsive inputs.
* **Clarity**: Bold stats, color-coded risk levels (e.g., green for low risk).

**Components**

1. **Landing Page**:
   * Welcome text: “Assess Investment Risks Instantly.”
   * “Run Demo” button + custom input option.
2. **Input Section**:
   * Form with fields: Initial Investment, Years, Revenue Range, Cost Range, Discount Rate.
   * Tooltips for guidance (e.g., “Enter min/max revenue”).
3. **Results Dashboard**:
   * Top: Key stats (Mean NPV, Risk Probability).
   * Middle: Interactive NPV histogram (Chart.js).
   * Bottom: Cumulative probability curve + recommendation (e.g., “Low Risk: Proceed”).
4. **Styling**:
   * Clean fonts (e.g., Roboto), blue/white color scheme, mobile-responsive layout.

**7. Team Segregation (4 Members)**

**Roles and Tasks**

1. **Member 1: Frontend/UI Developer**
   * Build React.js UI with Chart.js visuals.
   * Design input form and interactive dashboard.
   * Ensure responsive, polished UI/UX.
   * **Deliverable**: Sleek interface with demo-ready visuals.
2. **Member 2: Backend Developer**
   * Setup Flask API for data flow.
   * Manage mock data (e.g., €40k case in CSV).
   * Integrate MCS and ML outputs into API.
   * **Deliverable**: Functional backend with demo data.
3. **Member 3: MCS Developer**
   * Code MCS engine (NumPy, SciPy).
   * Simulate thousands of trials for NPV/risk metrics.
   * Test with mock data for accuracy.
   * **Deliverable**: Working MCS module.
4. **Member 4: ML + Deployment Developer**
   * Build ML model (scikit-learn linear regression).
   * Forecast variables (e.g., revenue trends).
   * Deploy to Heroku, prep pitch visuals.
   * **Deliverable**: Predictive module + live platform.

**Collaboration**

* **Frontend + Backend**: Connect UI to API.
* **MCS + ML**: Link predictions to simulations.
* **All**: Test demo case end-to-end.

**8. 30-Hour Timeline**

**Hour 0-3: Setup (3h)**

* **All**: Sync, setup tools (Git, React, Flask, Python).
* **Frontend**: Scaffold UI, install Chart.js.
* **Backend**: Initialize Flask, mock data.
* **MCS**: Start MCS logic.
* **ML/Deploy**: Setup scikit-learn, Heroku.

**Hour 3-12: Core Development (9h)**

* **Frontend**: Build input form, placeholder dashboard.
* **Backend**: API endpoints, data integration.
* **MCS**: Finalize simulation (NPV, stats).
* **ML/Deploy**: Train ML model on mock data.

**Hour 12-21: Integration + UI/UX (9h)**

* **Frontend**: Add interactive charts, polish UI.
* **Backend**: Link MCS/ML to API, test flow.
* **MCS**: Validate outputs (e.g., NPV range).
* **ML/Deploy**: Integrate forecasts, start deployment.

**Hour 21-27: Testing + Polish (6h)**

* **Frontend**: Refine UX (tooltips, colors).
* **Backend**: Stabilize API, demo case.
* **MCS**: Optimize speed if needed.
* **ML/Deploy**: Deploy to Heroku, test live.

**Hour 27-30: Final Prep (3h)**

* **All**: Run demo (€40k case), debug.
* **ML/Deploy**: Prep pitch (5-min: problem → solution → demo).

**9. Deliverables**

* **Platform**: Live on Heroku, with demo case (€40k, 5 years, 12% rate).
* **UI/UX**: Clean, interactive interface (input form, charts).
* **Outputs**: NPV histogram, risk stats (e.g., “90% chance > 0”).
* **Pitch**: 5-min demo showing risk insights.

**10. Conclusion**

This platform fulfills the hackathon goal by assessing investment risks with MCS and predictive analytics, delivered through a scalable, user-friendly UI/UX. The 30-hour plan and four-member segregation ensure a polished MVP that impresses judges with actionable insights.