

## FINAL REPORT CONTENT (Use for PDF)

### Title

**Financial Literacy RPG Quest — Simulation Based Personal Finance Learning System**

---

### 1. Objective

To design a simulation-driven financial learning platform that teaches real-world money management through decision-based gameplay.

---

### 2. System Overview

The system simulates a virtual financial life where a player earns income, manages expenses, handles debt, invests money, and faces random financial events. Each decision impacts financial stability, risk level, and long-term wealth growth.

---

### 3. Architecture

- Player Engine
  - Decision Engine
  - Financial Simulation Core
  - Risk Evaluation Model
  - Credit Score Engine
  - Advice Generation System
  - Random Event Generator
  - REST API Layer
- 

### 4. Simulation Model

Each month:

- Salary added
- Random life event applied
- Player decision executed
- Debt / savings updated

- Credit score recalculated
  - Risk and health evaluated
  - Financial score computed
- 

## 5. Financial Metrics

### Net Worth

Net Worth = Balance + Savings – Debt

### Financial Score

Measures overall financial stability using savings, debt, and credit score.

### Risk Levels

- LOW → Safe finances
  - MEDIUM → Manage debt carefully
  - HIGH → Financial instability risk
- 

## 6. Decision Engine

Players can:

- SAVE → Increase savings
  - INVEST → Grow wealth with risk
  - SPEND → Reduce balance
  - LOAN → Increase debt
- 

## 7. Credit Score Model

Credit score changes based on:

- Debt level
  - Savings stability
  - Negative balance
- 

## 8. Advice System

AI-based rule engine generates financial guidance based on:

- Risk level
  - Financial health
  - Savings behavior
- 

## **9. Game Progression Levels**

- BROKE
  - STABLE
  - GROWING
  - INVESTOR
  - WEALTHY
- 

## **10. Game Outcome**

- BANKRUPT → Excessive debt
- IN PROGRESS → Normal
- FINANCIALLY FREE → Strong wealth

## SAMPLE OUTPUT :

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Personal Workspace' selected, showing 'Collections', 'Environments', 'History', and 'Flows'. The main area has tabs for 'New Collection' and 'New Request'. A 'POST' request is selected, with the URL 'http://localhost:8060/game/play/c17d1ec1-3e5f-4c28-941c-937c3ffecab1'. The 'Body' tab is active, showing the JSON payload:

```
1 {  
2   "type": "SAVE",  
3   "amount": 10000  
4 }  
5
```

Below the request, the response is displayed in a 'Body' section with a 'JSON' dropdown. The status is '200 OK' with a response time of '1.58 s'. The response body is:

```
1 {  
2   "playerId": "c17d1ec1-3e5f-4c28-941c-937c3ffecab1",  
3   "balance": 90000.0,  
4   "savings": 10000.0,  
5   "debt": 15000.0,  
6   "creditScore": 650,  
7   "riskLevel": "MEDIUM",  
8   "financialHealth": "WEAK",  
9   "level": "STABLE",  
10  "advice": "Increase savings for financial security."  
11 }
```