Invoicing ROI Simulator – Java Version

Problem Statement: Create a lightweight ROI calculator that helps users visualize cost savings and payback when switching from manual to automated invoicing. The application should take basic business metrics as input and produce clear, favorable results that demonstrate the advantages of automation.

Purpose: This project is a small Java web application that helps users estimate how much money they can save by moving from manual to automated invoicing. It calculates monthly savings, ROI, and payback period using simple formulas.

Approach & Architecture: The app consists of two main parts:

1. Frontend (UI):

* A single-page web form where users enter details like invoice volume, wage, and error rate.
* Results appear instantly as the user modifies inputs.

1. Backend (API):

* Java + Spring Boot server that:
  + Calculates ROI and savings
  + Stores and retrieves scenarios using a local database or JSON file
  + Generates a simple PDF/HTML report using libraries like iText or Apache PDFBox

Technologies Used:

| Layer | Technology | Purpose |
| --- | --- | --- |
| Frontend | HTML, CSS, JavaScript | User interface |
| Backend | Java + Spring Boot | REST API and business logic |
| Database | JSON file / H2 / SQLite | Store user scenarios |
| PDF Tool | iText / Apache PDFBox | Generate reports |

Key Features: - ROI & savings calculator - Save and load scenarios - Email-gated report download - Always shows a positive ROI (favorable bias)

Instant UI Feedback: - As the user edits inputs, the frontend shows real-time recalculated results. - Can calculate locally in JS or request backend recalculation dynamically.

Validation & Safety: - Ensure inputs are valid (non-negative numbers, reasonable ranges). - Sanitize inputs to prevent malicious data.

Simple & Self-Documented API: - All responses in JSON format - Consistent structure with proper error handling

Next Steps: 1. Build backend endpoints using Spring Boot REST controllers. 2. Create a simple frontend form and connect it to the API using AJAX/Fetch. 3. Test calculations, scenario saving, and report generation.