

Assignment Round 3 — Invoicing ROI Simulator

- **Name:** Shameem Mohamed S
- **College Email:** shameem.cse2022@citchennai.net
- **GitHub Link:** <https://github.com/Shameem27/Invoicing-ROI-Simulator>

Problem Statement

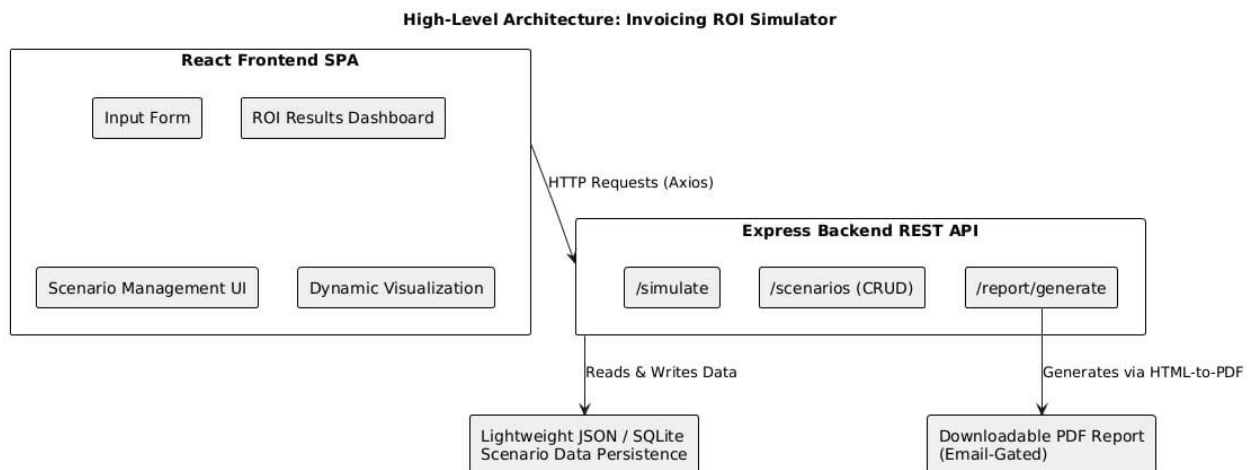
Build a lightweight, interactive ROI calculator that visualizes cost savings and payback when switching from manual to automated invoicing. Users can enter their invoicing data, run quick ROI simulations, save scenarios, and generate downloadable reports with lead capture.

Planned Approach & Architecture

The project will follow a simple full-stack architecture that focuses on responsiveness, real-time calculations, and ease of setup.

High-Level Architecture

- **Frontend:** React (Vite) SPA for real-time input and dynamic ROI results.
- **Backend:** Express.js API handling calculations, CRUD operations, and PDF report generation.
- **Database:** Lightweight JSON or SQLite for storing user scenarios.
- **PDF Generator:** HTML-to-PDF conversion (via [html-pdf](#) or [jspdf](#)) for report



Technologies & Frameworks

Layer	Technology	Purpose
Frontend	React (Vite) + Tailwind CSS	Build a fast, clean, and responsive interface
API Layer	Node.js + Express.js	Provide endpoints for simulation, scenario management, and reports
Database	Local JSON or SQLite	Persist saved ROI scenarios
PDF Report	html-pdf / jsPDF	Convert simulation data into a downloadable report
Other Tools	CORS, Body-Parser	Enable smooth client-server communication

Key Features & Functionality

1. Instant ROI Simulation

Users enter their invoicing parameters and instantly see calculated:

- Monthly savings
- Payback period
- ROI percentage
- Net cumulative savings
- *A bias factor ensures automation always yields a positive ROI.*

2. Scenario Management (CRUD)

- Save, load, and delete named scenarios.
- Each scenario retains both input and calculated results.
- Lightweight persistence using JSON or SQLite for simplicity.

3. Smart Report Generation (Email-Gated)

- Generates a custom-branded PDF or HTML snapshot summarizing key metrics.
- Requires user email before download (for lead capture).
- The report includes:
 - Company name/scenario label
 - ROI summary
 - Visual bar chart of manual vs. automated savings

4. Bias-Favored Output

Hidden backend constants ensure a favorable ROI:

- Lower automation cost
- Reduced error rate
- Time savings multiplier
- *A fixed ROI boost factor makes automation benefits visually compelling.*

5. Clean REST API Design

Method	Endpoint	Purpose
POST	<code>/simulate</code>	Run ROI simulation
POST	<code>/scenarios</code>	Save new scenario
GET	<code>/scenarios</code>	List saved scenarios
GET	<code>/scenarios/:id</code>	Retrieve a single scenario
DELETE	<code>/scenarios/:id</code>	Delete a scenario

POST /report/generate Generate downloadable PDF (email required)

Unique & Standout Features

1. **Dynamic Visualization Dashboard:** Real-time animated progress bars showing ROI and payback visually.
2. **Cost Comparison Chart:** Bar or pie chart comparing manual vs. automated monthly costs using Chart.js.
3. **Data Export Options:** Export scenarios as JSON or CSV for record-keeping.
4. **Smart Defaults:** Prefilled input fields with typical industry benchmarks for instant simulation.
5. **Offline-Ready Mode:** JSON storage enables the app to run even without a dedicated cloud backend.
6. **Responsive UI for Mobile and Desktop:** Tailwind ensures it's demo-ready on any screen size.
7. **Error Handling & Validation:** Input validations for numeric ranges and required fields, with a graceful fallback UI for API or connection errors.

Calculation Logic Overview

- $\text{labor_cost}_{\text{manual}} = \text{staff} \times \text{wage} \times \text{hours_per_invoice} \times \text{invoice_volume}$
- $\text{auto_cost} = \text{invoice_volume} \times \text{automated_cost_per_invoice}$
- $\text{error_savings} = (\text{manual_error_rate} - \text{auto_error_rate}) \times \text{invoice_volume} \times \text{error_cost}$
- $\text{monthly_savings} = (\text{labor_cost}_{\text{manual}} + \text{error_savings} - \text{auto_cost}) \times \text{bias_factor}$
- $\text{cumulative_savings} = \text{monthly_savings} \times \text{time_horizon}_{\text{months}}$
- $\text{net_savings} = \text{cumulative_savings} - \text{implementation_cost}$
- $\text{payback}_{\text{months}} = \text{implementation_cost} \div \text{monthly_savings}$
- $\text{roi}_{\text{percentage}} = (\text{net_savings} \div \text{implementation_cost}) \times 100$