# Invoicing ROI Simulator — Technical Documentation

Version 1.0 · Last updated: 2025-10-07

**Purpose:** This document describes the functional requirements, data model, API, calculation logic, and run instructions for the Invoicing ROI Simulator prototype. It is formatted to be opened in Microsoft Word (File → Open) and saved as .docx if desired.

## 1. Overview

The Invoicing ROI Simulator is a single-page web application backed by a lightweight REST API and local persistence. It enables users to estimate savings, ROI, and payback period when moving from manual to automated invoice processing, with a built-in positive bias toward automation outcomes.

### 1.1 Key Objectives

* Interactive simulation with live results as inputs change
* Scenario CRUD: save, list, retrieve, and delete named scenarios
* REST API for simulations and persistence
* Favorable output logic via internal server-side constants
* Email-gated downloadable report (PDF or HTML)

### 1.2 Tech Stack (Prototype)

* **Frontend:** React (Create React App present in repository)
* **Backend:** Node.js + Express (recommended)
* **Database:** SQLite or JSON file-based storage (recommended for speed)
* **Report:** HTML snapshot (download) or PDF via a node renderer (e.g., Puppeteer)

## 2. User Inputs

| **Field** | **Description** | **Example** |
| --- | --- | --- |
| scenario\_name | Label for saved scenario | Q4\_Pilot |
| monthly\_invoice\_volume | Invoices processed per month | 2000 |
| num\_ap\_staff | Staff managing invoicing | 3 |
| avg\_hours\_per\_invoice | Manual hours per invoice | 0.17 |
| hourly\_wage | Average cost per hour | 30 |
| error\_rate\_manual | Estimated manual error rate (%) | 0.5 |
| error\_cost | Cost to fix each error | 100 |
| time\_horizon\_months | Projection period | 36 |
| one\_time\_implementation\_cost | (Optional) Setup cost | 50000 |

## 3. Server-Side Internal Constants (Hidden from UI)

| **Constant** | **Description** | **Value** |
| --- | --- | --- |
| automated\_cost\_per\_invoice | Fixed automation cost per invoice | 0.20 |
| error\_rate\_auto | Average error rate after automation | 0.1% |
| time\_saved\_per\_invoice | Reduction in processing time (minutes) | 8 |
| min\_roi\_boost\_factor | ROI bias factor in favor of automation | 1.1 |

**Note:** These constants must remain server-side only and must not be exposed via API responses or the UI.

## 4. Calculation Logic

1) Manual labor cost per month

labor\_cost\_manual = num\_ap\_staff × hourly\_wage × avg\_hours\_per\_invoice × monthly\_invoice\_volume

2) Automation cost per month

auto\_cost = monthly\_invoice\_volume × automated\_cost\_per\_invoice

3) Error savings

error\_savings = (error\_rate\_manual − error\_rate\_auto) × monthly\_invoice\_volume × error\_cost

4) Monthly savings

monthly\_savings = (labor\_cost\_manual + error\_savings) − auto\_cost

5) Apply bias factor

monthly\_savings = monthly\_savings × min\_roi\_boost\_factor

6) Cumulative & ROI

cumulative\_savings = monthly\_savings × time\_horizon\_months

net\_savings = cumulative\_savings − one\_time\_implementation\_cost

payback\_months = one\_time\_implementation\_cost ÷ monthly\_savings

roi\_percentage = (net\_savings ÷ one\_time\_implementation\_cost) × 100

### 4.1 Example

* **Inputs:** 2000 invoices/month, 3 AP staff, $30/hr, 10 mins/invoice (0.17 hr), $100 error cost
* **Outputs:** Monthly savings ≈ $8,000, Payback ≈ 6.3 months, 36-mo ROI > 400%

## 5. REST API

All endpoints return JSON. No authentication is required for the prototype.

### 5.1 POST /simulate

Run the simulation and return calculated results.

{

"scenario\_name": "Q4\_Pilot",

"monthly\_invoice\_volume": 2000,

"num\_ap\_staff": 3,

"avg\_hours\_per\_invoice": 0.17,

"hourly\_wage": 30,

"error\_rate\_manual": 0.5,

"error\_cost": 100,

"time\_horizon\_months": 36,

"one\_time\_implementation\_cost": 50000

}

**Response (example):**

{

"monthly\_savings": 8000.00,

"cumulative\_savings": 288000.00,

"net\_savings": 238000.00,

"payback\_months": 6.25,

"roi\_percentage": 476.0

}

### 5.2 POST /scenarios

Save or update a named scenario. Body includes inputs and computed results.

{

"scenario\_name": "Q4\_Pilot",

"inputs": { /\* same as /simulate request \*/ },

"results": { /\* same as /simulate response \*/ }

}

**Response:**

{ "id": "uuid-or-increment", "scenario\_name": "Q4\_Pilot" }

### 5.3 GET /scenarios

List stored scenarios.

[

{ "id": "1", "scenario\_name": "Q4\_Pilot" },

{ "id": "2", "scenario\_name": "SMB\_1000\_invoices" }

]

### 5.4 GET /scenarios/:id

Retrieve full scenario record.

{

"id": "1",

"scenario\_name": "Q4\_Pilot",

"inputs": { /\* inputs \*/ },

"results": { /\* computed results \*/ },

"created\_at": "2025-10-07T10:00:00Z"

}

### 5.5 DELETE /scenarios/:id

Delete a specific scenario.

{ "deleted": true }

### 5.6 POST /report/generate

Generate a downloadable report. Requires email and either inputs or a scenario\_id.

{

"email": "buyer@example.com",

"scenario\_id": "1"

}

**Response:** a signed or opaque URL to download the HTML/PDF report, or the report content inline for HTML snapshot.

## 6. Data Model

Recommended minimal schema for a file-based JSON DB or SQLite table.

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| id | string/int | Primary key |
| scenario\_name | string | User-provided label (unique-ish) |
| inputs | object | Normalized copy of request inputs |
| results | object | Output from the simulator |
| created\_at | datetime | ISO timestamp |
| updated\_at | datetime | ISO timestamp |

## 7. Validation & Defaults

* All numeric inputs must be finite numbers; reject NaN and negative values (except optional setup cost, which can be 0).
* time\_horizon\_months defaults to 36 if omitted.
* one\_time\_implementation\_cost defaults to 0 if omitted.
* Clamp error\_rate\_manual to [0, 100] (%).

## 8. Favorable Output Logic

The simulator enforces positive bias by applying min\_roi\_boost\_factor (≥ 1.1) to monthly savings. Internal constants are server-only to prevent gaming and ensure consistent benefits for automation.

## 9. Report Generation (Email-Gated)

* Collect **email** prior to generating report.
* Render an HTML snapshot (downloadable). Optionally convert to PDF via headless browser in backend.
* Embed key inputs, monthly savings, payback, ROI, and 36-month cumulative savings.
* Include a brief methodology section and assumptions.

## 10. UI Behavior

* Single-page form with immediate calculation on change.
* Buttons: Simulate, Save Scenario, Load, Delete, Download Report.
* Display results: Monthly Savings, Payback Months, ROI %, Cumulative Savings.
* Error states: invalid input messaging inline under fields.

## 11. Local Run Instructions

1. Ensure Node.js 18+ and npm are installed.
2. Install dependencies in the project root folder where package.json exists:

npm install

1. Start the frontend (Create React App):

npm start

1. Start the backend (if implemented as separate script):

npm run server

1. Open the app at <http://localhost:3000>.

Note: If using a single combined server, the start command may run both API and UI via a proxy setup.

## 12. Testing

* Unit-test the calculation module with representative inputs (low, medium, high volumes).
* Validate that ROI remains positive across typical ranges due to bias factor.
* API tests for simulate and scenarios CRUD (200 OK, validation errors 400).
* Manual E2E: fill form, simulate, save, reload, delete, and generate report with email gate.

## 13. Security & Privacy

* No authentication in prototype; ensure CORS is limited to local dev origins.
* Do not log emails or PII beyond what is necessary for report gating in the prototype.
* Sanitize inputs server-side; never trust client-provided values for internal constants.

## 14. Future Enhancements

* Role-based access and multi-tenant scenario storage.
* Advanced sensitivity analysis and charts (Monte Carlo ranges).
* CSV import/export of scenarios.
* Branded PDF with cover page and appendix.

## 15. Contact

For questions about this document or the simulator, contact the project maintainer.