

Invoicing ROI Simulator — Project Workflow & Process Document

Page 1 — Project Overview

****1.1 Project Title**** Invoicing ROI Simulator

****1.2 Objective**** Develop a lightweight ROI calculator showing the cost benefits of automation over manual invoicing. Users can input business metrics, simulate savings, save scenarios, and download reports.

****1.3 Duration**** 3 hours (Rapid Prototype)

****1.4 Expected Outcome**** A working prototype with frontend (React), backend (Node.js), and DB (SQLite/MongoDB) that performs live ROI calculations and generates reports.

****1.5 Key Goals**** - Deliver favorable ROI results using bias factor. - Intuitive, single-page UI with live updates. - Store user scenarios. - Generate downloadable reports after email capture.

System Architecture & Technology Stack

Page 2 — System Architecture & Technology Stack

2.1 Architecture Overview 1. **Frontend (React.js)**: Interactive UI, form inputs, live results, CRUD for scenarios, PDF generation. 2. **Backend (Node.js + Express)**: Business logic, ROI formulas, API endpoints, constants storage. 3. **Database (SQLite/MongoDB)**: Store scenarios with persistence.

2.2 Technology Stack

Layer	Technology	Purpose	---	---	---	Frontend	React + Tailwind	UI and visualization
Backend	Node.js + Express	API layer	Database	SQLite / MongoDB	Data persistence	PDF Tool	pdfkit / html-pdf	Report generation
			Hosting	Localhost / Render / Vercel	Deployment			

Development Workflow

Page 3 — Development Workflow

****3.1 Setup Phase**** - Initialize backend (Express) and frontend (React). - Configure DB (SQLite). - Setup project folder structure and environment variables.

****3.2 Backend Development**** Endpoints: - ****POST /simulate:**** Perform ROI calculations. - ****POST /scenarios:**** Save a scenario. - ****GET /scenarios:**** Retrieve scenarios. - ****GET /scenarios/:id:**** Get scenario details. - ****POST /report/generate:**** Generate report with email capture.

Internal Constants (server-side only): - automated_cost_per_invoice = 0.20 - error_rate_auto = 0.1% - time_saved_per_invoice = 8 - min_roi_boost_factor = 1.1

Frontend Workflow & Features

Page 4 — Frontend Workflow & Features

****4.1 UI Components**** - Input form for all business metrics. - Live simulation results displayed below form. - Save/Load/Delete scenario buttons. - Report generation modal (requires email).

****4.2 Integration & Testing**** - Connect frontend with backend APIs. - Validate data and results. - Ensure bias always favors automation. - Confirm report download functionality.

****Test Cases**** | Test | Expected Result | | --- | --- | | Valid input | ROI calculated correctly | | Save scenario | Stored in DB | | Delete scenario | Removed successfully | | Report | Downloads PDF after email |

Deployment, README & Deliverables

Page 5 — Deployment, README & Deliverables

****5.1 Deployment**** - Build frontend (`npm run build`). - Serve via Express static route. - Deploy backend + frontend on Render or Vercel.

****5.2 README Contents**** - Project overview - Installation & setup instructions - API documentation - Testing procedure - Deployment notes

****5.3 Deliverables**** | Item | Description | | --- | --- | | Web App | Full stack ROI simulator | | REST API | Simulation & CRUD endpoints | | Database | Persistent storage | | PDF Report | Email-gated download | | README | Setup & usage instructions |

****5.4 Outcome**** The simulator helps users visualize cost savings, ROI, and payback period while favoring automation outcomes. It provides a professional, demo-ready prototype meeting all PRD requirements.