# 7-Day MVP Execution Plans

Prepared for: Masoud Arouni  
Date: 2025-09-09

# Real Estate Deal Evaluator — MVP in 7 Days (B2C)

🎯 Target Audience: First-time or small-scale real estate investors

📝 Summary:  
A consumer-facing web app that helps small or first-time real estate investors quickly evaluate deals using key metrics like Cap Rate, Cash-on-Cash Return, and ROI. Built using Streamlit for fast iteration and mobile-friendliness.

## 🗓 7-Day Execution Plan

### Day 1: Scoping & UI Wireframe

* • Define core features (inputs: price, rent, loan %, expenses).
* • Sketch layout using Streamlit or Figma.
* • Set up GitHub repo + .env.template.

### Day 2: Metric Calculations

* • Implement logic for Cap Rate, CoC Return, ROI.
* • Validate against real property examples.

### Day 3: Interactive Dashboard

* • Add dynamic charts (e.g., cash flow projection, breakeven).
* • Display colored indicators (e.g., good vs. bad deal).

### Day 4: PDF Export or Email Report

* • Create downloadable PDF report with results.
* • Optional: Email report using SMTP or SendGrid.

### Day 5: Deal Grade + Comparison

* • Add logic to auto-grade deals A/B/C/F.
* • Allow user to compare multiple properties side-by-side.

### Day 6: Mobile UX + Finishing Touches

* • Optimize layout for mobile view.
* • Add form validations and field defaults.

### Day 7: Demo Launch + User Testing

* • Publish to Streamlit Cloud.
* • DM 5–10 real estate contacts or LinkedIn leads.
* • Collect feedback and revise.

# Legacy PDF → Smart Data Extractor — MVP in 7 Days

🎯 Target Audience: Legal ops, small city departments, utilities, and contractors dealing with legacy PDFs

📝 Summary:  
A backend-focused tool that ingests legacy PDFs (city forms, contracts, bills) and extracts structured data using a mix of PDF parsing, regex, and LLM-enhanced tagging. Outputs data in CSV or JSON for use in other systems.

## 🗓 7-Day Execution Plan

### Day 1: Project Setup & Sample PDFs

* • Gather 3–5 real-world sample PDFs (RFPs, NDAs, bills).
* • Set up project structure and GitHub repo.

### Day 2: Basic PDF Text Extraction

* • Use PyMuPDF or pdfplumber to extract text/tables.
* • Handle layout issues, headers, footers.

### Day 3: Regex-Based Field Extraction

* • Build regex for common fields (dates, company, clauses).
* • Test on multiple PDFs and refine.

### Day 4: LLM Extraction Add-on

* • Integrate OpenRouter + prompt to extract key values.
* • Compare performance vs regex.

### Day 5: Export & UI Display

* • Build Streamlit interface to preview extracted data.
* • Allow export to CSV and JSON.

### Day 6: Extraction Confidence Scores

* • Highlight confidence per field (LLM or regex match).
* • Let user edit/fix extracted fields before export.

### Day 7: Launch + Share

* • Deploy to Streamlit Cloud or Replit.
* • Demo to legal, gov, or data-heavy users.
* • Collect use cases for next version.