project: India Business Daily (MVP)

goal

Pull a small set of India-focused business RSS items daily, dedupe, LLM-summarize into tight bullets, lightly categorize, and save a Markdown digest. Keep it minimal, testable, and cheap.

non-goals (for v1)

- No heavy scraping beyond RSS.
- No database. Files only.
- No web UI. Output is a markdown file.

target outputs (acceptance criteria)

- digest.md generated locally with:
  - Title, date, and 6–12 stories.
  - Each story has: source name, one-liner (≤22 words), up to 2 bullets, and the original link.
  - Each story labeled with one of: **policy**, **markets**, **startups**, **infra**, **energy** (multi-label allowed).
- Run time < 2 minutes on a laptop.
- If a feed is down, the run still completes with remaining sources.

minimal sources (MVP)

- **PIB Press Releases** (policy, official)
- **RBI Press Releases** (policy/finance)
- **Reuters India Business** (broad business)

(we'll add more after MVP passes)

data flow (v1)

1. **Load config** (list of feeds, max items).
2. **Ingest**: Pull RSS → extract fields: title, summary/description, link, published, source.
3. **Normalize**: Strip HTML, collapse whitespace.
4. **Dedupe**: Drop items with same (lower(title[0:120]), link).
5. **LLM step** (single batch):
   - Summarize: one-liner + up to 2 bullets, facts only.
   - Classify into labels set.
6. **Render**: Assemble digest.md grouped by label.
7. **Persist**: Save digest.md to disk (and a JSON copy of raw items + LLM output for audit).

configuration

- .env: API key + model name + MAX_ITEMS.
- sources.yml: list of RSS feeds (name, url).

prompting standards (paste into Claude Code when it asks for spec)

System prompt (summarizer/classifier)

You are a precise India business analyst. Use only the provided item fields (title, summary, source, link, published). Do not invent facts. When unsure, say “unclear”. Return strict JSON only.

User payload structure

- INPUT JSON: list of items with title, summary, source, link, published.
- TASKS:
    1. For each item, produce:
        - one_liner (≤22 words, factual, no adjectives without evidence).
        - bullets (array, ≤2, each = *what happened* + *why it matters to India or investors* + *numbers if present*).
    2. Classify each item into zero or more labels from:
        - policy, markets, startups, infra, energy
        - If none fit, use misc.
    3. Return exactly:

```
[{
  "title": "...",
  "source": "...",
  "link": "...",
  "one_liner": "...",
  "bullets": ["...", "..."],
  "labels": ["policy"]
}]
```

Output acceptance checks (Claude should self-enforce)

- JSON parses.
- one_liner ≤ 22 words.
- No claims not present in the input.

files to create (by Claude Code)

1. README.md
    - What the tool does, requirements, how to run, and what files are produced.
2. sources.yml
    - Keys: feeds:[{name, url}]. Start with the 3 sources listed above.
3. .env.example
    - ANTHROPIC_API_KEY=...
    - LLM_MODEL=claude-3-5-sonnet-20240620
    - MAX_ITEMS=12

4. app.py (single entry point for now)
    - Loads config & env.
    - Fetches RSS from sources.yml.
    - Cleans/describes data (no HTML).
    - Dedupe.
    - Sends one batched LLM call with the prompts above.
    - Renders digest.md grouped by label (section headers).
    - Saves run_YYYY-MM-DD.json (input + output) for audit.

(you'll ask Claude Code to implement each file one by one; no need to prewrite code yourself.)

task cards for Claude Code (copy/paste these one at a time)

Task 1 — Repo bootstrap (no code yet)

- Create a new project called india-biz-daily.
- Initialize README.md with the project purpose, minimal run instructions, and "MVP scope".
- Add .gitignore for Python and local env files.
- Create empty sources.yml and .env.example with the keys above.

Acceptance: files exist; README states the goal and how to run; .env.example present.

Task 2 — Define sources

- Fill sources.yml with 3 feeds (PIB, RBI, Reuters India Business) using name and url.
- Ensure YAML is valid.

Acceptance: sources.yml parses; each item has name and url.

Task 3 — App skeleton

- Create a single script app.py that:
    - Loads .env and sources.yml.
    - Fetches entries from each feed (limit total items by MAX_ITEMS).
    - Extracts title, summary, link, published, source.
    - Cleans HTML tags and collapses whitespace.
    - Dedupe on (lower(title[0:120]), link).
    - Prints how many items were fetched and kept.
- No LLM call yet; just outputs a JSON preview to console.

Acceptance: Running the script prints ≥1 item for a working internet connection and shows deduped count.

Task 4 — LLM integration (summarize + classify)

- Add a function that sends the deduped list to Claude using the **prompting standards** above.

- Return parsed JSON. If parsing fails, retry once; if it still fails, fall back to a minimal one-liner (“unclear”) per item.
- Add a small validator:
  - one_liner length check ≤22 words,
  - bullets length ≤2,
  - labels subset of the allowed set.

Acceptance: The function returns valid JSON for the fetched inputs; validator passes; failures are logged but run completes.

Task 5 — Render markdown digest

- Group items by the first label (or misc).
- Write digest.md with:
  - H1 = “India Business Daily — <date>”
  - For each label section: H2 header, then each story with:
    - Title (plain text), Source (italic), two bullets (list), and a “Source” link.
- Also write a run_<date>.json to disk containing:
  - raw_items: the cleaned RSS items,
  - llm_output: the model’s JSON for traceability.

Acceptance: A markdown file is saved with at least one section and at least one story; links are clickable; a JSON audit file exists.

Task 6 — Failure handling / resilience (still code-light)

- If any feed returns 0 items or errors out, skip it and continue.
- Log a summary line per feed: name, items_ok, items_failed.
- If all feeds fail, produce digest.md with a polite note “No items today”.

Acceptance: Simulate one broken feed URL; run still completes with remaining feeds.

QA checklist you’ll run manually (5 minutes)

- Open run_<date>.json: does raw_items count match dedupe count?
- Skim digest.md:
  - any hallucinated numbers? (shouldn’t be)
  - one-liners short and factual?
  - links open to correct sources?
- Turn off one feed in sources.yml: does the run still produce a digest?

next milestones (after MVP passes)

- **Add two more feeds** (Mint, Business Standard) and watch for duplicates → clustering later.
- **Add label set expansion** (defense & aerospace, auto & EV, telecom, healthcare).

- **Schedule** with cron/n8n at 7:15 IST to save digest.md to a dated folder.

risks & guardrails

- **Paywalls / ToS**: use RSS only for now; we’re not storing full articles.
- **Hallucinations**: prompt forbids invention; keep summaries short; manual spot-check.
- **Flaky feeds**: resilience task ensures partial success.
- **Costs**: cap MAX_ITEMS (start with 12); single batched LLM call.