

PRE-HACKATHON FULL PROJECT DRY-RUN PLAN

Goal of This Dry-Run

Before the 24-hour hackathon, you should be able to:

-  Upload a transaction file
-  Detect subscriptions
-  Show savings insights
-  Demo smoothly in 2 minutes

Not perfection — **working clarity**.

PHASE 1: Freeze the Scope (VERY IMPORTANT)

Do NOT try everything

For hackathon success, **scope control = survival**.

Final MVP Scope

You WILL build:

1. Transaction upload (CSV)
2. Subscription detection
3. Risk / forgotten label
4. AI explanation (optional but recommended)
5. Simple mobile-style UI

You will NOT build:

-  Bank integrations
 -  Payment cancellation
 -  User accounts
-

PHASE 2: Define the Exact User Flow

One Simple Flow (This is your demo)

Open App

- Upload CSV
- Click Analyze
- See subscriptions
- See "Potentially Forgotten"
- See savings summary

If this works → **project is successful**.

● PHASE 3: Prepare Data FIRST (Don't Code Yet)

📁 Create 3 Sample CSV Files

1 Normal User

- Netflix
- Spotify
- Swiggy
- Amazon

2 Heavy Subscription User

- Netflix
- Prime
- Adobe
- Notion
- Google One

3 Almost No Subscriptions

- Mostly UPI spends

📌 This helps testing + demo.

● PHASE 4: Backend (Core Brain)

🧠 Step 1: CSV Parser

- Read file
- Clean data
- Keep only:
 - Date
 - Description
 - Amount (Debit only)

🧠 Step 2: Subscription Detection Logic

Start **simple & reliable**:

Group by merchant name

Check if repeats monthly

Check amount similarity

Label as:

- Subscription
 - Non-subscription
-

Step 3: Risk Scoring (Simple Formula)

Example:

Risk Score =

+ Repeated > 6 months

+ High cost

+ No change in amount

Output:

- Low
 - Medium
 - High
-

PHASE 5: AI Insight Layer (Optional but Powerful)

What AI Does:

- Explains results in human language
- NOT detection

Example Prompt:

"Explain which subscriptions might be forgotten and how much money the user can save yearly."

This avoids over-dependency on AI.

PHASE 6: Frontend (Mobile-First UI)

UI Pages (Minimum)

Upload Screen

- Upload CSV
- Button: "Analyze"

2 Results Screen

- List subscriptions
- Show:
 - Monthly cost
 - Risk badge

3 Savings Summary

- “You can save ₹3,240/year”
-

Design Rules

- Max width: 390–412px
- Large text
- Clean colors
- No clutter

Judges care about clarity, not animations.

PHASE 7: Connect Frontend ↔ Backend

- API endpoint:
- POST /analyze
- Return JSON:
 - Subscriptions
 - Risk
 - Savings

Test with **Postman** first.

PHASE 8: Full Dry-Run Test (CRITICAL)

Run this checklist:

- Upload works
- Wrong file handled
- Results appear < 3 sec
- UI works on mobile view
- Demo flow smooth

Fix **only critical bugs**.