Pawzz Data Analyst Assignment – Shourya Garg

Q1) Tools + System Design for Community-Verified Directory (Long Answer)

As the lead Data Analyst, my goal would be to create a repeatable, mostly automated, auditable data pipeline that handles messy, multi-source inputs while ensuring high accuracy for publication.

1. Required Fields & Proof Standards per Category

- **Preferred Schema (Column names) :** Business Shop Name, Category (vet/shelter/feeder), Primary Phone No., Alternate Phone No., Full Address, Google Maps Link, Contact Person Name, Operating Hours, Services Offered, Last Updated Date.
- **Proof standards:**
  - **High-proof (required for publish):** Photo of signboard, Business card scan, Official registration (GST), Google Maps pin match (approximately), WhatsApp profile link.
  - **Low-proof (needs verification):** Text-only submission will be flagged as 'needs fix'.

I will store this in a centralized schema (Airtable base or Google Sheets master or SQL Databases) with validation rules (ex - phone must be 10 digits, address must have PIN).

2. Data Pipeline Stages

1. **Submitted(raw)**
   - **Sources:** Google Form (embedded on website/WhatsApp bot), Airtable form, Zapier from WhatsApp Business API/Instagram DMs, manual intern entry from Google Maps/Justdial.
   - **Auto-actions:** Zapier → enrich with Google Maps API (address → lat/long), deduplicate check (basic name + phone fuzzy match via Airtable script).
2. **Needs Fix(incomplete)**
   - Auto-flagged if missing core fields or low-proof.
   - Team member (intern) assigned → fix or reject.
3. **Ready(cleaned & proof-attached)**
   - Manual review + attachment upload.
   - Auto-score quality
- **Verified(community + internal check)**
   - **Cross-check:** Call/SMS verification (Twilio/Zapier), community flag form (public "Is this accurate?" link), cross-reference with existing Pawzz network.
   - **Status:** Verified / Re-check Needed.
4. **Published**
   - Sync to live directory (Airtable to website via API).

- Auto-archive old versions for audit.

Tools stack:

- **Intake:** Google Forms / Typeform + Zapier (for WhatsApp/Instagram).
- **Storage & Workflow:** Airtable (visual, relational, automations, free tier sufficient initially) or Notion Databases.
- **Cleaning:** I would prefer python because I have used python pandas many times and I am very confident with pandas in Data Cleaning.
- **Enrichment:** Google Maps API, Clearbit/Truecaller-like for phone validation (if budget allows).
- **Visualization:** Power BI

3. Dashboards / KPIs to Track

Build in Power BI / Looker Studio:

- **Throughput:** Listings submitted/week, → ready, → verified, → published (funnel chart).
- **Quality:** % with high-proof, % rejected, average time per stage.
- **Backlog:** Open items in "Needs Fix" + "Verified – Re-check"
- **Source performance:** Accuracy percentage from Google Maps vs Community vs Interns.
- **Alerts:** Slack/email if backlog greater than 50 or quality less than 80%.

This setup lets interns run 80–90% autonomously; founder only approves edge cases.

4. Handling Unstructured Inputs

- **WhatsApp/screenshots -** Zapier to Airtable (extract text via OCR if possible, or manual transcription box).
- **Inconsistent formats -** Python/Pandas standardization scripts (ex - remove +91 from phone numbers).
- **Bulk intern research -** shared Google Sheet template and import to Airtable weekly via CSV.
- **Long-term:** Build simple Python ETL (using Pandas + requests for scraping) run on schedule via Google Cloud Functions.

Q2) Red Flags + Verification + Duplicate Prevention (Long Answer)

Top Red Flags for Unreliable Listings

1. No proof attached (no photo) .
2. Invalid mobile no.
3. Address mismatch (Google Map pins far from submitted).
4. Multiple similar submissions in short time (spam).
5. Inconsistent details .
6. Negative community flags or reports (via public form).
7. Duplicate-like patterns (same photos across different names).

End-to-End Verification + Duplicate-Prevention System

1. **Duplicate Prevention:**
   - **Fuzzy matching at intake:** Compare Name + Phone + Address using Python, if similiarity is greater than 80% (or threshold), then deemed fake.
   - **Tools:** Airtable extension or Python script for bulk.
2. **Verification Checklist:**
   - Business name matches proof photo.
   - Phone reachable
   - Address verifiable via Google Maps street view match.
   - Registration number checked
   - Services listed realistic (e.g., no "exotic surgery" without proof).
   - At least 2 high-proof items (photo + registration or Google Business Profile).
   - No red flags from above list.
   - Community cross-check (send link to known rescuers: "Is this vet reliable?").
3. **Handling Conflicts:**
   - Create linked "versions" in Airtable (one master, child variants).
   - **Prioritize:** Most recent + highest-proof version as primary.
   - Flag as "Conflicting Data - Warning" on publish.
   - Audit trail logs the change
4. **Audit Trail Design:**
   - **Use Airtable's built-in activity log + custom fields:**
     - Verified By (person name/email)
     - Verified On (timestamp)
     - Proof Links/Attachments
     - Changes Log (Airtable revision history or separate "History" linked table)
     - Status Notes (e.g., "Changed phone after owner confirmation call")
   - Export monthly CSV of changes for founder review.
5. **Escalation Rules:**
   - **Reject:** <2 proofs, multiple red flags, unreachable phone, deemed fake
   - **Re-check:** Conflicts, medium-proof only, negative flag
   - **Publish with Warning:** High-proof but minor inconsistency

- **Publish:** All checks pass, high-proof