# Project Documentation — Customer-Journey GraphRAG

## 1) What this solves

Massive clickstream Events and lightweight Users data are processed without overloading Neo4j. This project compresses user journeys, clusters them, stores minimal metadata in Neo4j, and enables GraphRAG retrieval with natural language queries.

## 2) Repo structure

config.py — Paths and constants  
requirements.txt — Dependencies  
build\_features.py — Stream Parquet → per-user features  
cluster\_and\_summarize.py — Clustering + summary text generation  
load\_neo4j.py — Load clusters & samples into Neo4j  
ask.py — CLI GraphRAG retrieval

server.py — (upcoming) FastAPI /ask endpoint  
3\_build\_category\_vocab.py — (upcoming) Learn category vocabulary  
data/ — Intermediate files  
examples/ — Example outputs

## 3) Data in/out

Input: events/\*.parquet, users/\*.parquet  
Outputs: features.parquet, cluster\_members.parquet, clusters.parquet  
Neo4j: :Cluster, :Customer (sampled), :Journey (optional)  
Raw events remain outside Neo4j.

## 4) How to run

Install: pip install -r requirements.txt  
Run steps:  
python 0\_build\_features.py  
python 1\_cluster\_and\_summarize.py  
python 2\_load\_neo4j.py  
CLI: python ask.py "query here"  
Examples saved in examples/ folder.

## 5) File-by-file details

config.py — Config vars  
build\_features.py — Builds features  
cluster\_and\_summarize.py — Clustering + summaries  
load\_neo4j.py — Loads to Neo4j  
ask.py — Parses filters, graph query, semantic ranking  
server.py — FastAPI wrapper  
build\_category\_vocab.py — Learns categories from data

## 6) Troubleshooting

Neo4j map property errors → flatten dicts  
Empty results → lower thresholds in ask.py  
Windows/Py3.13 safe.

## 7) Coverage matrix

Use full data: ✅ build\_features.py  
Raw events outside Neo4j: ✅  
Minimal Neo4j schema: ✅  
Cluster summaries: ✅  
Graph filters + semantic: ✅  
Ask & Show pipeline: ✅  
Submission deliverables met: ✅

## 8) Upcoming features

1. HTTP API (FastAPI)  
2. Auto-learn categories  
3. Stricter funnel semantics  
4. Threshold controls  
5. Explainability  
6. Neo4j vector index  
7. Enriched :Customer nodes