

Firebase-Based Recipe Analytics Pipeline

Technical Report

Executive Summary

This report documents a complete data engineering pipeline built on Firebase Firestore for recipe data extraction, transformation, validation, and analytics. The system processes an authentic Maharashtrian cuisine dataset containing 20+ recipes, demonstrating modern ETL practices with NoSQL databases.

1. Introduction

1.1 Project Overview

The Recipe Analytics ETL Pipeline is designed to extract recipe data from Firebase Firestore, transform it into normalized relational formats, validate data quality, and generate actionable insights through analytics and visualizations.

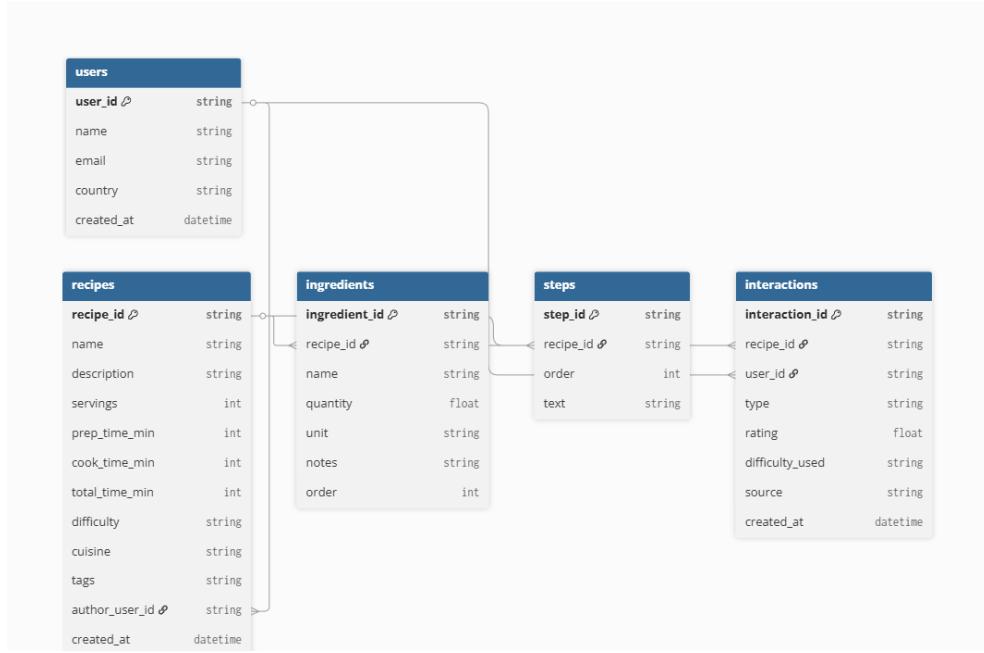
1.2 Technology Stack

Component	Technology
Database	Firebase Firestore
Programming Language	Python 3.8+
Data Processing	Pandas, NumPy
Visualization	Matplotlib
Firebase SDK	firebase-admin

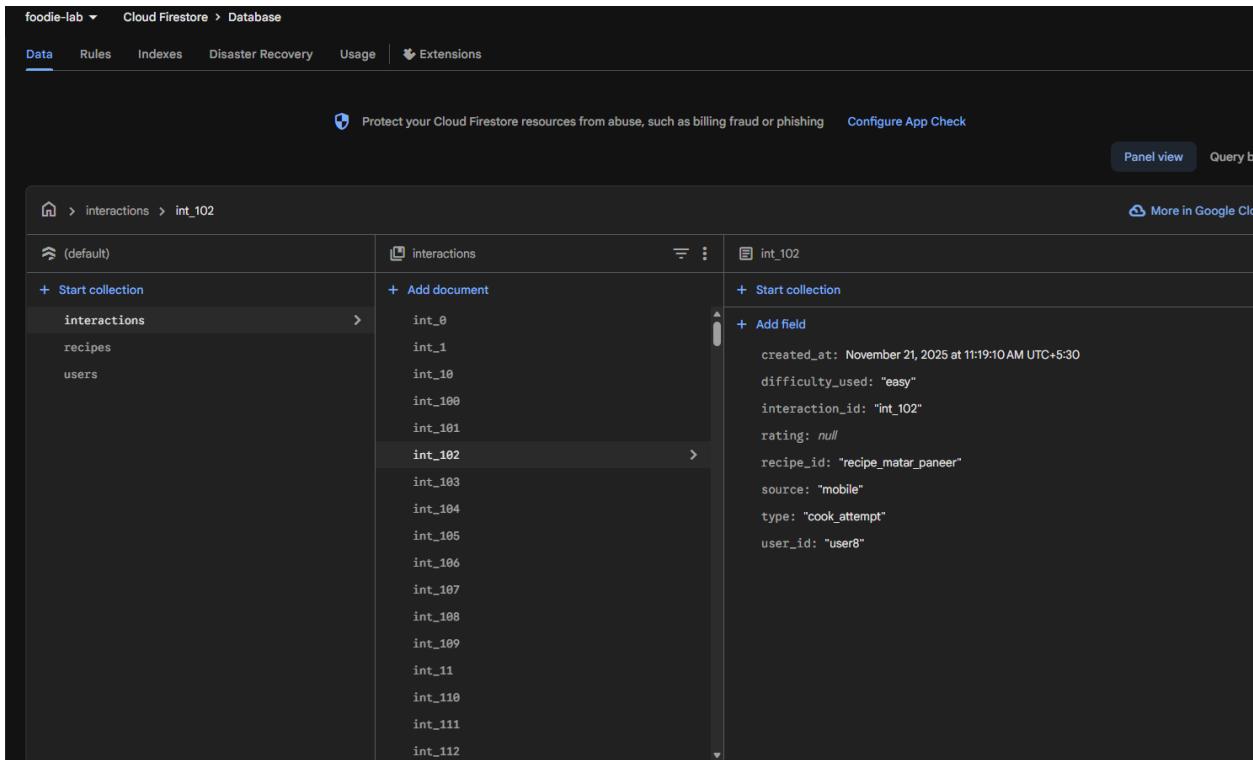
1.3 System Architecture



2. Data Model Design



2.1 Database Architecture



The screenshot shows the Firebase Firestore console interface. At the top, there's a navigation bar with 'foodie-lab' (selected), 'Cloud Firestore', 'Database', and tabs for 'Data', 'Rules', 'Indexes', 'Disaster Recovery', 'Usage', and 'Extensions'. Below the navigation is a security warning: 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' link. On the right, there are 'Panel view' and 'Query builder' buttons.

The main area displays a hierarchical document structure under the 'interactions' collection. The left sidebar shows the database structure: 'recipes' and 'users' collections, with 'interactions' being expanded to show its sub-documents: 'int_0', 'int_1', 'int_10', 'int_100', 'int_101', 'int_102' (selected), 'int_103', 'int_104', 'int_105', 'int_106', 'int_107', 'int_108', 'int_109', 'int_110', 'int_111', and 'int_112'. The right panel shows the detailed structure of the selected document 'int_102', which contains fields like 'created_at', 'difficulty_used', 'interaction_id', 'rating', 'recipe_id', 'source', 'type', and 'user_id'.

The system uses Firebase Firestore with a hierarchical document structure optimized for NoSQL patterns.

2.2 Design Decisions

Design Choice	Reasoning
Interactions under Recipes	Groups recipe activity together; enables fast queries for single recipe analytics
Activities under Users	Tracks user behavior across recipes; enables user-centric analytics
Denormalized author names	Avoids extra reads; Firestore doesn't support JOINs

2.3 Output Schema

The pipeline produces four normalized CSV tables:

recipes.csv - Primary recipe information including title, description, timing, difficulty, category, dietary type, and author details.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
recipe_id	name	descriptor	servings	prep_time	cook_time	total_time	difficulty	cuisine	tags	author_us	created_at		
recipe_alo	Aloo Parat	Synthetic r	4	26	26	52	medium	Indian	synthetic	user2	2025-11-21T05:49:08.988Z		
recipe_chi	Chicken Bi	Synthetic r	2	35	33	68	medium	Indian	synthetic	user3	2025-11-21T05:49:09.738Z		
recipe_dal	Dal Tadka	Synthetic r	3	21	23	44	easy	Indian	synthetic	user1	2025-11-21T05:49:09.476Z		
recipe_egg	Egg Curry	Synthetic r	6	35	46	81	easy	Indian	synthetic	user3	2025-11-21T05:49:09.828Z		
recipe_fish	Fish Curry	Synthetic r	3	33	21	54	easy	Indian	synthetic	user3	2025-11-21T05:49:09.786Z		
recipe_gul	gulab Jam	Synthetic r	5	21	50	71	medium	Indian	synthetic	user3	2025-11-21T05:49:08.903Z		
recipe_idli	Idli Sambh	Synthetic r	3	37	16	53	hard	Indian	synthetic	user1	2025-11-21T05:49:09.684Z		
recipe_jeer	Jeera Rice	Synthetic r	2	25	19	44	easy	Indian	synthetic	user1	2025-11-21T05:49:09.520Z		
recipe_khe	Kheer	Synthetic r	2	20	25	45	medium	Indian	synthetic	user1	2025-11-21T05:49:08.947Z		
recipe_ma	Masala Dc	Synthetic r	2	11	34	45	medium	Indian	synthetic	user1	2025-11-21T05:49:08.757Z		
recipe_ma	Matar Pan	Synthetic r	2	12	16	28	hard	Indian	synthetic	user2	2025-11-21T05:49:08.888Z		
recipe_mi	Misal Pav	Synthetic r	4	35	28	63	easy	Indian	synthetic	user1	2025-11-21T05:49:09.324Z		
recipe_pa	Palak Pane	Synthetic r	2	16	25	41	medium	Indian	synthetic	user1	2025-11-21T05:49:09.557Z		
recipe_pa	Paneer Bu	Synthetic r	3	34	37	71	hard	Indian	synthetic	user3	2025-11-21T05:49:08.804Z		
recipe_po	Poha	Synthetic r	2	11	32	43	hard	Indian	synthetic	user3	2025-11-21T05:49:09.024Z		
recipe_pur	Puran Poli	Traditiona	4	45	30	75	medium	Indian	sweet,fest	user_veda	2025-11-21T05:49:06.147Z		
recipe_sa	Sabudana	Synthetic r	2	29	18	47	medium	Indian	synthetic	user1	2025-11-21T05:49:09.392Z		
recipe_shr	Shrikhand	Synthetic r	4	29	47	76	hard	Indian	synthetic	user3	2025-11-21T05:49:09.431Z		
recipe_upr	Upma	Synthetic r	2	20	29	49	medium	Indian	synthetic	user1	2025-11-21T05:49:09.590Z		
recipe_vac	Vada Pav	Synthetic r	4	14	35	49	easy	Indian	synthetic	user2	2025-11-21T05:49:09.363Z		
recipe_veg	Veg Palao	Synthetic r	5	13	44	57	hard	Indian	synthetic	user2	2025-11-21T05:49:08.859Z		

ingredients.csv - Normalized ingredient data linked to recipes via foreign key, containing name, quantity, unit, and optional flag.

A1	recipe_id	ingredient_name	quantity	unit	notes	order
1	recipe_puri ng1	chana dal	1	cup	soaked 2â·	1
2	recipe_puri ng10	ghee	4	tbsp	for roasting	10
3	recipe_puri ng11	water	1	cup	as required	11
4	recipe_puri ng2	jaggery	1	cup	grated	2
5	recipe_puri ng3	cardamom	0.5	tsp		3
6	recipe_puri ng4	nutmeg powder	0.25	tsp	optional	4
7	recipe_puri ng5	wheat flour	2	cups		5
8	recipe_puri ng6	maida (optional)	0.5	cup		6
9	recipe_puri ng7	turmeric powder	0.25	tsp		7
10	recipe_puri ng8	salt	0.25	tsp		8
11	recipe_puri ng9	oil	2	tbsp	for dough	9
12						
13						

users.csv - Recipe instructions with step numbers, instructions, and duration, linked to parent .

	A1	:	X	✓	fx	user_id	
1	A	B	C	D	E	F	G
2	user_id	name	email	country	created_at		
3	user1	User 1	user1@tes IN		2025-11-21T05:49:09.930Z		
4	user10	User 10	user10@tes IN		2025-11-21T05:49:09.931Z		
5	user2	User 2	user2@tes IN		2025-11-21T05:49:09.930Z		
6	user3	User 3	user3@tes IN		2025-11-21T05:49:09.930Z		
7	user4	User 4	user4@tes IN		2025-11-21T05:49:09.930Z		
8	user5	User 5	user5@tes IN		2025-11-21T05:49:09.930Z		
9	user6	User 6	user6@tes IN		2025-11-21T05:49:09.930Z		
10	user7	User 7	user7@tes IN		2025-11-21T05:49:09.931Z		
11	user8	User 8	user8@tes IN		2025-11-21T05:49:09.931Z		
12	user9	User 9	user9@tes IN		2025-11-21T05:49:09.931Z		
13	user_veda	Vedant Ra	vedant@e IN		2025-11-21T05:49:09.930Z		

interactions.csv - User engagement data including ratings, cook notes, and timestamps.

	A1	:	X	✓	fx	interaction_id				
1	A	B	C	D	E	F	G	H	I	J
2	interaction	user_id	recipe_id	type	rating	difficulty	source	created_at		
3	int_0	user8	recipe_fish like		medium	mobile		2025-11-21T05:49:10.140Z		
4	int_1	user10	recipe_gul cook_attempt		medium	mobile		2025-11-21T05:49:10.141Z		
5	int_10	user3	recipe_par like		hard	mobile		2025-11-21T05:49:10.142Z		
6	int_100	user3	recipe_upr rating		5	easy	web	2025-11-21T05:49:10.146Z		
7	int_101	user4	recipe_pol rating		hard	mobile		2025-11-21T05:49:10.146Z		
8	int_102	user8	recipe_ma cook_attempt		easy	mobile		2025-11-21T05:49:10.146Z		
9	int_103	user4	recipe_dal cook_attempt		hard	mobile		2025-11-21T05:49:10.146Z		
10	int_104	user2	recipe_fish rating		5	easy	web	2025-11-21T05:49:10.146Z		
11	int_105	user8	recipe_alo view		easy	mobile		2025-11-21T05:49:10.146Z		
12	int_106	user6	recipe_vac view		easy	web		2025-11-21T05:49:10.146Z		
13	int_107	user9	recipe_fish rating		3	medium	mobile	2025-11-21T05:49:10.147Z		
14	int_108	user8	recipe_chi rating		easy	mobile		2025-11-21T05:49:10.147Z		
15	int_109	user8	recipe_vac cook_attempt		easy	web		2025-11-21T05:49:10.147Z		
16	int_110	user9	recipe_kh rating		medium	mobile		2025-11-21T05:49:10.147Z		
17	int_111	user1	recipe_jee like		medium	mobile		2025-11-21T05:49:10.147Z		
18	int_112	user10	recipe_veg rating		5	medium	web	2025-11-21T05:49:10.147Z		
19	int_113	user2	recipe_pur cook_attempt		easy	web		2025-11-21T05:49:10.147Z		
20	int_114	user1	recipe_idli like		medium	web		2025-11-21T05:49:10.147Z		
21	int_115	user2	recipe_idli rating		5	medium	mobile	2025-11-21T05:49:10.147Z		
22	int_116	user3	recipe_dal like		medium	mobile		2025-11-21T05:49:10.147Z		
23	int_117	user4	recipe_jee view		easy	web		2025-11-21T05:49:10.147Z		
24	int_118	user9	recipe_pal rating		medium	mobile		2025-11-21T05:49:10.147Z		
25	int_119	user6	recipe_fish cook_attempt		easy	mobile		2025-11-21T05:49:10.147Z		
26	int_120	user1	recipe_par like		hard	web		2025-11-21T05:49:10.147Z		
27	int_121	user5	recipe_idli cook_attempt		hard	mobile		2025-11-21T05:49:10.147Z		
28	int_122	user5	recipe_kh like		medium	mobile		2025-11-21T05:49:10.147Z		

3. ETL Process

3.1 Pipeline Architecture

The pipeline follows a four-stage process: Extract → Transform → Validate → Analyze

3.2 Extract Phase

The extraction phase connects to Firebase Firestore using the Admin SDK with streaming for efficient data retrieval. Key operations include authentication via service account credentials, document streaming for large collections, subcollection extraction for interactions, and timestamp conversion to ISO 8601 format.

3.3 Transform Phase

Transformation	Description
Flatten Ingredients	Nested array converted to separate CSV with recipe_id FK
Flatten Steps	Nested array converted to separate CSV with recipe_id FK
Extract Subcollections	Firestore subcollection to interactions.csv
Normalize Time	Structured time object to individual minute columns
Handle Missing Data	Default values: "Uncategorized", "Unknown"

4. Data Validation

4.1 Validation Rules

Rule	Field	Criteria
Required Fields	title	Must not be empty
Valid Difficulty	difficulty	Must be: Easy, Medium, Hard, Expert

Rule	Field	Criteria
Prep Time	prep_time_min	Must be > 0
Cook Time	cook_time_min	Must be ≥ 0
Time Logic	total_time_min	Must be $\geq \text{prep_time} + \text{cook_time}$
Ingredient Quantity	quantity	Must be > 0 if numeric
Rating Range	rating	Must be between 0 and 5
Has Steps	steps	At least one step required
Has Ingredients	ingredients	At least one ingredient required

4.2 Validation Output

The validator produces a JSON report containing total recipe count, valid/invalid counts, detailed error messages for invalid records, and list of valid record IDs.

5. Analytics & Insights

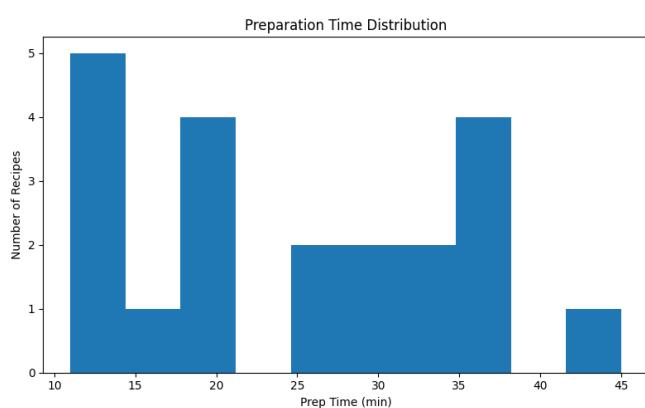
5.1 Generated Insights

The pipeline produces 11 analytical insights:

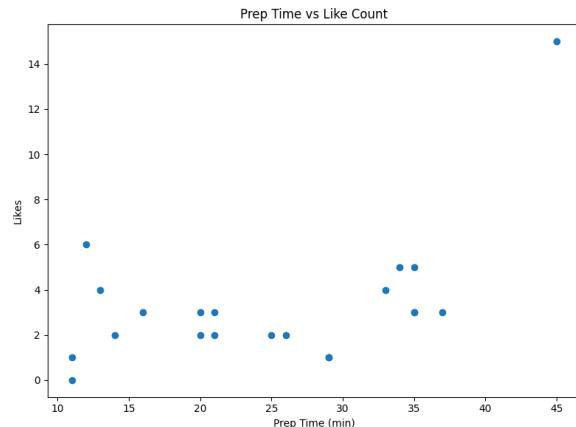
1. **Most Common Ingredients** - Top 20 ingredients by frequency
2. **Average Prep Time** - Mean preparation time in minutes
3. **Average Cook Time** - Mean cooking time in minutes
4. **Difficulty Distribution** - Recipe count per difficulty level
5. **Most Interacted Recipes** - Top 20 by interaction count
6. **Prep vs Rating Correlation** - Statistical correlation analysis

7. **High-Rating Ingredients** - Ingredients appearing in 4+ star recipes
8. **Top Rated Recipes** - Top 10 by average rating
9. **Steps Distribution** - Statistical summary of recipe complexity
10. **Most Commented Recipes** - Top 10 by cook note count
11. **Longest Recipes** - Top 10 by total preparation time

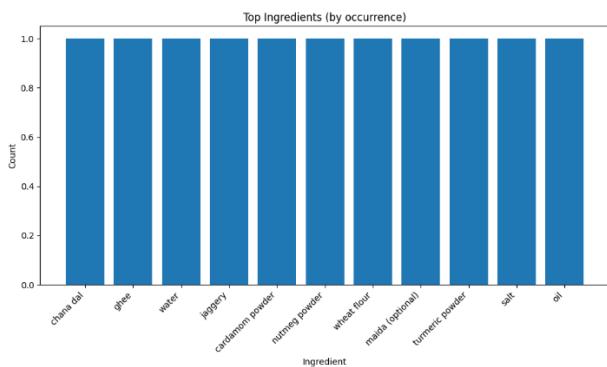
5.2 Visualizations



6.1 Preparation Time Distribution

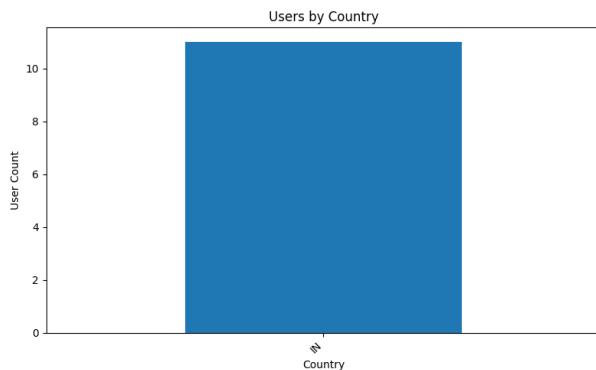


6.2 Prep Time vs Like Count

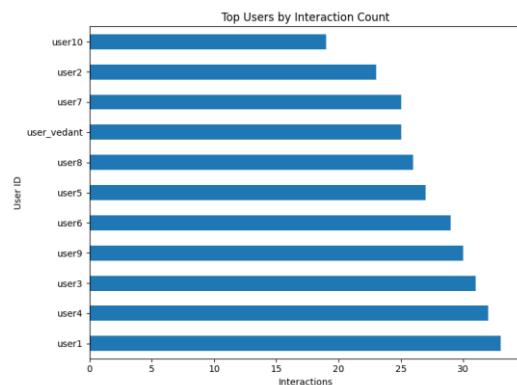


6.3 Top Ingredients (by frequency)

7. User Analytics (Charts)



7.1 Users by Country



7.2 Top Users by Interaction Count

6. Constraints & Limitations

6.1 Firestore Limitations

Constraint	Impact	Mitigation
No native JOINs	Cannot query across collections	Denormalized data; subcollections
Read costs	Each document read is billed	Streaming instead of batch reads
No aggregations	No COUNT/SUM/AVG in queries	Aggregations in Python post-export
Subcollection queries	Cannot query all at once	Iterate per parent document

6.2 Pipeline Constraints

- Sequential execution required (seed → transform → validate → analyze)
- Full export only; no incremental/delta processing
- Memory-bound with pandas DataFrames
- Hardcoded relative paths in some scripts

6.3 Scalability Notes

- Current tested capacity: ~20 recipes, ~200 interactions
 - For 1000+ recipes: implement pagination in export
 - For larger datasets: consider chunked processing or Apache Spark
-

7. Project Structure

RECIPE_ANALYTICS/

```
|── analytics/      # Analytics outputs and charts  
|── config/        # Firebase credentials  
|── data_validation/  # Validation scripts and reports  
|── Firebase_Setup/   # Data seeding scripts  
|── transform_data/    # ETL outputs (CSV files)  
└── README.md
```

8. Installation & Execution

8.1 Prerequisites

- Python 3.8 or higher
- Firebase project with Firestore enabled
- Service account credentials (JSON)

8.2 Execution Steps

1. **Seed Initial Data** - Run seed_data.py to create base recipe and user
2. **Generate Synthetic Data** - Run generate_sythetic.py for 20 Maharashtrian recipes
3. **Transform Data** - Run transform.py to export to CSV

4. **Validate Data** - Run validator.py for quality checks
 5. **Generate Analytics** - Run analytics.py for insights and charts
-

9. Deliverables Summary

Deliverable	Status
Source files for ETL scripts	Complete
Validation script	Complete
Normalized CSV output	Complete
Analytics summary (JSON)	Complete
Documentation	Complete
Visualization charts	Complete

10. Conclusion

The Firebase-Based Recipe Analytics Pipeline successfully demonstrates a complete ETL workflow for NoSQL data, producing normalized relational outputs suitable for further analysis. The system handles the unique challenges of Firestore's document model while maintaining data quality through comprehensive validation.
