Mini Project Report

Project Title

Hybrid Customer Database Manager

1. Introduction

The Hybrid Customer Database Manager is a console-based system designed to integrate and analyze customer data from two distinct domains: telecom (calls, SMS) and property (leads, listings). Businesses often maintain separate systems for communication and customer engagement, leading to fragmented insights. This project bridges that gap by providing a unified platform for querying and analyzing customer interactions across both domains.

2. Objective

- To build a unified database system that merges telecom and property data.
- To enable console-based analytics for customer engagement and lead tracking.
- To support business decisions through integrated insights from communication and property interest data.

3. Scope of the Project

- Integration of telecom data (call logs, SMS records) with property data (customer leads, property listings).
- Console interface for querying combined analytics.
- Support for queries like:
  - Most contacted leads
  - Listings with highest inquiries
  - Customer activity summaries
- Modular design for future expansion (e.g., CRM integration, dashboard UI).

4. System Features

- Unified Data Model: Combines telecom and property datasets.
- Console Query Engine: Accepts user commands to retrieve analytics.
- Customer Insights: Tracks communication volume and property interest.
- Lead Prioritization: Identifies high-engagement leads.
- Listing Analytics: Highlights popular properties based on inquiries.

5. Technology Stack

Programming Language: Python / Java

Data Storage: SQLite / CSV / JSON

Interface: Console (CLI)

Libraries: datetime, argparse, pandas (Python) or JDBC + Scanner (Java)

Architecture: Object-Oriented Design

6. Sample Console Output

> show top 5 leads by call volume

```
+------------+------------------+------------+
| Lead ID    | Name             | Call Count |
+------------+------------------+------------+
| L102       | Priya Sharma     | 18         |
| L089       | Arjun Mehta      | 15         |
| L054       | Kavya Reddy      | 13         |
| L120       | Rohan Verma      | 12         |
| L033       | Sneha Iyer       | 11         |
```

> show listings with most SMS inquiries

```
+------------+-----------------------+--------+
| Listing ID | Title                 | SMSes  |
+------------+-----------------------+--------+
| P201       | 2BHK in Sholinganallur | 25     |
| P178       | Villa in ECR          | 21     |
| P145       | Studio in Velachery   | 19     |
```

> show customer activity summary for +91-9876543210

Calls: 12

SMS: 8

Leads Contacted: 3

Listings Inquired: 2

7. Learning Outcomes

- Learned how to design and implement a hybrid data model.

- Gained experience in object-oriented programming and modular system design.

- Developed skills in building console-based query engines.
- Understood how to merge and analyze data from different business domains.
- Practiced integrating analytics into real-world business use cases.

8. Module Overview

1. Telecom Data Module: Manages call and SMS records.
2. Property Data Module: Manages property listings and customer leads.
3. Query Engine Module: Accepts and parses console commands.
4. Integration and Analytics Module: Joins telecom and property data on customer identifiers.

9. Architecture Overview

Architectural Style: Console-based application using object-oriented design.

Data Layer: Flat files (CSV/JSON) or SQLite for persistent storage.

Interaction: CLI commands trigger backend logic and data queries.

10. Module-wise Design

Telecom Data Module:

- Features: Store and retrieve call/SMS logs.
- Entities: CallRecord, SMSRecord.

Property Data Module:

- Features: Manage property listings and customer leads.
- Entities: Lead, Listing.

Query Engine Module:

- Features: Accepts commands and maps to backend logic.
- Entities: Query, QueryResult.

Integration and Analytics Module:

- Features: Join telecom and property data.
- Entities: CustomerProfile.

11. Deployment Strategy

Local Deployment:

- Designed for local execution on developer machines.

- No external dependencies or cloud services required.

Steps:

1. Clone/download the project.

2. Run the main script or class from terminal/IDE.

3. Input queries via the console.

12. Database Design

Tables/Structures:

- Customer: customer_id, name, phone

- CallRecord: call_id, customer_id, timestamp, duration

- SMSRecord: sms_id, customer_id, timestamp, message_length

- Lead: lead_id, customer_id, property_id, timestamp

- Listing: property_id, title, location, price

Relationships:

- One customer can have many call/SMS records.

- One customer can inquire about many listings (leads).

- One listing can be linked to many leads.

13. User Interface Design

Console Interface:

- Text-based command input.

- Tabular or summary output.

- Example commands:

  - show top 5 leads by call volume

  - show listings with most SMS inquiries

  - show customer activity summary for <phone_number>

14. Assumptions and Constraints

Assumptions:

- All data is stored locally in structured files or a lightweight database.

- Customer IDs are consistent across telecom and property datasets.

- The system is used by analysts or support staff familiar with CLI.

Constraints:

- No GUI or web interface is provided.
- No real-time data ingestion; batch data only.
- Limited to local execution; no cloud or distributed support.