Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

Privacy-Preserving Financial Transaction Analytics System

[1]Mahi Panjwani, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]AMITY UNIVERSITY, CHHATTISGARH
[1]mahipanjwani76@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

In today's digital world, financial institutions deal with a very large amount of transaction data on a daily basis. This data often contains sensitive information related to users, such as spending patterns and transaction details. While analyzing this data is important for tasks like fraud detection and understanding user behavior, it also creates serious privacy concerns.

In this project, we try to build a system that can analyze financial transaction data without directly exposing sensitive information. The system uses simple privacy techniques like anonymization and adding small noise to data so that user identity is not revealed. At the same time, the data still remains useful for analysis.

A full-stack system is developed where the backend is built using Flask, the frontend provides a simple interface for interaction, and SQLite is used for storing the data. The system also performs clustering to group users based on their behavior.

Overall, the project shows that it is possible to get useful insights from financial data while still maintaining user privacy to a certain level.

Keywords- Privacy-Preserving Analytics, Financial Transactions, Data Anonymization, Behavioral Clustering, K-Means Clustering, Differential Privacy, Flask, SQLite, User Behavior Analysis, Data Security

I. INTRODUCTION

In the present time, digital financial transactions have become a very common part of daily life. People use online banking, UPI, credit cards, and mobile payment applications for almost every activity such as shopping, travel booking, bill payments, and money transfers. Because of this, a huge amount of financial transaction data is being generated every day.

This data is very useful. Banks and financial companies use it to understand how users spend money, detect fraud activities, and improve their services. But at the same time, this data is also very sensitive. It contains information like transaction amounts, account details, time of transactions, and user behavior patterns. Even if names are removed, the patterns in the data can still reveal information about a person.

This creates a major problem. On one side, companies need to analyze this data to get useful insights. On the other side, they cannot expose user data because of privacy risks. If data is directly used without protection, it can lead to misuse, identity exposure, and trust issues.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

Because of this, there is a need for a system which can analyze financial data but still keep user information safe.

In this project, we are trying to build a Privacy-Preserving Financial Transaction Analytics System. The main idea is simple: instead of directly using raw sensitive data, we first transform it into a safer version and then perform analysis on it.

The system first generates raw transaction data which includes user details, account numbers, transaction values, and categories. After that, a privacy pipeline is applied where sensitive information is removed or modified. User identity is replaced with anonymous IDs, account numbers are masked, and small noise is added to transaction amounts so that exact values cannot be traced.

After this transformation, the system performs clustering to group users based on their spending behavior. Instead of analyzing each transaction separately, the system focuses on user-level behavior such as average spending, frequency of transactions, and type of expenses. This helps in identifying patterns like low spenders, high spenders, or users who spend more on food or travel.

The system is implemented using a full-stack approach. The frontend is used to display the data and results, the backend handles processing and APIs, and the database stores the transformed data.

Overall, the goal of this project is to show that it is possible to perform meaningful financial data analysis without directly exposing sensitive information. It tries to balance both data usefulness and privacy, which is very important in real-world financial systems.

II. PROBLEM STATEMENT

In today's financial systems, a lot of transaction data is generated, but using this data directly is risky. The main problem is that this data contains sensitive information like transaction amounts, account details, and spending patterns. Even if names are removed, the data can still reveal user behavior.

At the same time, banks and companies still need this data for analysis, like detecting fraud or understanding customer habits. So there is a conflict — data is useful, but it is also sensitive.

Another issue is that many systems either ignore privacy or apply very basic protection like simple masking, which is not enough. Also, most analysis is done directly on raw data, which increases the risk.

So the main problem is to design a system where:

1. Data can be analyzed properly
2. But user privacy is not compromised

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

The system should hide sensitive details, still keep useful patterns, and allow meaningful analysis like clustering of user behavior.

III. OBJECTIVES

The main goal of this project is to build a system that can analyze financial transaction data without exposing sensitive user information. Along with that, some specific objectives are:

1. To generate a dataset that simulates real financial transactions with multiple users and different spending patterns.
2. To design a privacy pipeline that removes or hides sensitive details like user identity and account numbers.
3. To apply simple noise to transaction values so exact data cannot be traced back.
4. To perform clustering based on user behavior instead of individual transactions.
5. To group users into meaningful categories like low spenders, high spenders, or category-based users.
6. To build a backend system using APIs for processing and data handling.
7. To create a frontend interface that clearly shows the processed data and clustering results.
8. To demonstrate that useful insights can still be obtained even after applying privacy techniques.

IV. SYSTEM ARCHITECTURE

The system follows a simple three-layer structure where data flows step by step from raw generation to final visualization.

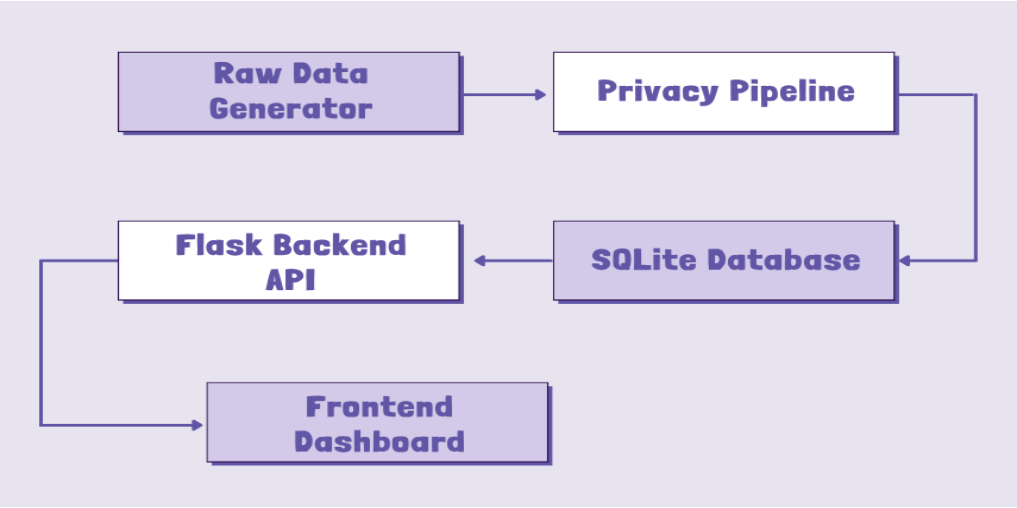


Figure IV. A Architecture of the System

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

4.1 Data Generation & Privacy Layer

In this part, raw transaction data is created for multiple users. Each user has multiple transactions based on a specific behavior (like food, travel, high spending, etc.).

This raw data contains sensitive details such as:

1. user name
2. account number
3. transaction amount

After generating the data, a privacy pipeline is applied:

1. names are removed
2. anonymous IDs are assigned
3. account numbers are masked
4. small noise is added to transaction amounts

This makes the data safer before storing or analyzing it.

4.2 Backend (Processing Layer)

The backend is built using Flask. It handles all the main logic of the system.

Main functions:

1. fetching data from database
2. running clustering algorithm
3. updating cluster values
4. sending data to frontend using APIs

Important APIs:

1. /api/transactions → load data
2. /api/analyze → run clustering

Clustering is done at user level, not transaction level, using features like:

1. average amount
2. number of transactions

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

4.3 Database Layer

SQLite is used as the database.

It stores:

1. privacy-preserved transactions
2. anonymous user IDs
3. cluster labels

The database acts as a middle layer between backend and frontend.

4.4 Frontend (User Interface)

The frontend is used to display the results.

Functions:

1. load transactions
2. run clustering
3. display cluster category

It shows:

1. anonymous ID
2. noisy transaction amount
3. category
4. cluster label

V. METHODOLOGY

The system is built in a step-by-step flow so that data is first prepared, then protected, and finally analyzed.

First, raw transaction data is generated for multiple users. Each user is assigned a behavior type like low spender, food user, traveler, or high spender. Based on this, multiple transactions are created so that the data is not random but follows some pattern.

After this, the privacy pipeline is applied. In this step, sensitive details are removed or modified. Names are removed, anonymous IDs are created, and account numbers are masked. Also, a small amount of noise is added to transaction values so that exact values cannot be traced.

Once the data is transformed, it is stored in the database. This stored data is now safe to use for analysis.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

Next, clustering is performed. Instead of clustering individual transactions, the system groups users based on their overall behavior. For each user, features like average transaction amount and number of transactions are calculated. These features are then used in the clustering algorithm.

The clustering groups users into different categories based on their spending patterns. These categories are then stored back in the database.

Finally, the frontend displays this data. The user can load transactions and run clustering using buttons. The results are shown in a table with cluster labels.

Overall, the methodology ensures that data is first protected and then analyzed, which keeps the system both useful and secure.

VI. WORKING OF THE SYSTEM

The system starts working when the user opens the frontend interface. From there, different actions can be performed using buttons like loading data or running clustering.

First, the raw data is generated using the dataset generator. This creates multiple users and multiple transactions based on different behavior patterns. This data is not directly used.

Next, the privacy pipeline runs. In this step, sensitive details are removed or changed. User names are replaced with anonymous IDs, account numbers are masked, and transaction amounts are slightly modified. This processed data is then stored in the database.

When the user clicks on **Load Transactions**, the frontend sends a request to the backend API. The backend fetches the stored data from the database and sends it back to the frontend. The data is then displayed in a table.

When the user clicks on **Run Clustering**, another API call is made. The backend takes the stored data and performs clustering based on user behavior. It calculates features like average spending and transaction count for each user and then assigns cluster values.

After clustering is completed, the updated cluster values are saved in the database. When the user loads the data again, the table now shows the cluster labels for each transaction.

So the complete flow is:

1. generate data
2. apply privacy
3. store in database
4. run clustering
5. display results

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

This way, the system shows how data can be processed and analyzed without directly exposing sensitive information.

VII. RESULTS

After running the system, the output is shown on the frontend in the form of a table. The table displays the processed transaction data along with cluster labels assigned after analysis.

The data shown is not raw data. It is already passed through the privacy pipeline, so sensitive details are not visible. Instead of real user identity, anonymous IDs are shown. Transaction amounts are slightly modified, and only useful fields like category are kept.

When clustering is applied, users are grouped based on their behavior. These groups are reflected in the table using cluster labels. Different users fall into different clusters depending on their spending pattern.

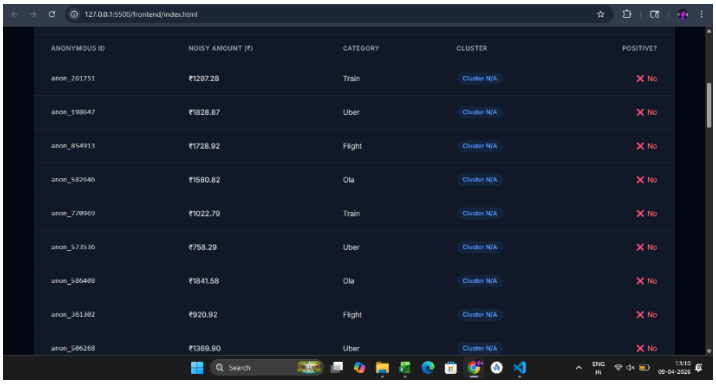
The results show that even after applying privacy techniques, meaningful grouping of users is still possible.

Table VII.A Sample Output

Anonymous ID	Noisy Amount	Category	Cluster
anon_45231	1450.25	Uber	Traveler
anon_78321	320.10	Zomato	Food User
anon_11234	7800.75	Amazon	High Spender
anon_99871	210.40	Grocery	Low Spender

Figure VII.A

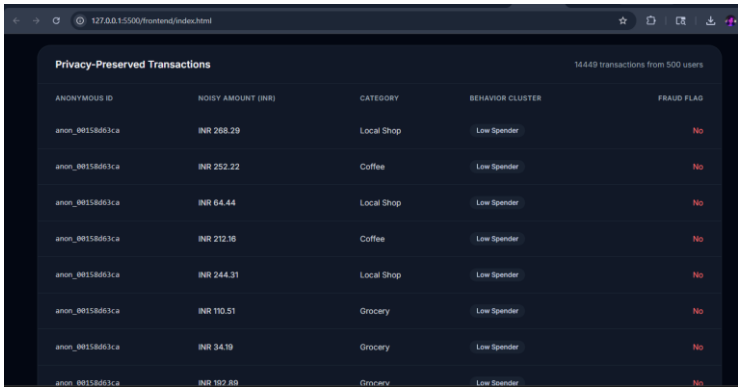
1. Before clustering
2. Table showing data with Cluster = N/A



Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

Figure VII.B

1. After clicking "Run Clustering"
2. Table showing cluster labels (Low Spender, Food User, etc.)



Different users with similar behavior fall into the same cluster, which shows that clustering is working based on patterns and not random values.

VIII. TECHNOLOGY USED

The system is built using simple and lightweight technologies so that it is easy to develop and run.

Frontend:
HTML, CSS, JavaScript
(Basic UI for displaying data and interacting with backend)

Backend:
Python, Flask
(Used for handling API requests and processing logic)

Database:
SQLite
(Stores processed transaction data and cluster values)

Libraries / Tools:
Pandas – for data handling
Scikit-learn – for clustering (K-Means)
NumPy – for numerical operations

Development Tools:
VS Code, Git, GitHub

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

IX. APPLICATIONS

This system can be useful in real-world financial and data analysis scenarios where privacy is important.

Banks and financial institutions can use this type of system to study customer spending patterns without exposing sensitive user information. Instead of working on raw data, they can use privacy-preserved data for analysis.

It can also be used in fraud detection systems. By grouping users based on behavior, unusual patterns can be identified without directly accessing personal details.

Fintech companies can use similar systems to understand user habits and improve their services, like suggesting better offers or detecting risky transactions.

This approach can also be useful when data needs to be shared with third parties for research or analytics. Instead of sharing raw data, a privacy-preserved version can be shared safely.

Overall, the system shows how useful insights can still be obtained while maintaining user privacy, which is very important in today's digital systems.

X. ADVANTAGES

1. Helps analyze financial data without exposing sensitive user information
2. Protects user identity by using anonymization and masking
3. Allows meaningful clustering based on user behavior
4. Works with multiple users and realistic transaction patterns
5. Simple system design, easy to understand and implement
6. Can be extended for real-world banking and fintech applications
7. Uses lightweight technologies, so it is easy to run and maintain
8. Shows practical use of privacy concepts in data analysis

XI. LIMITATIONS

1. The system uses simulated data, not real financial data
2. Privacy techniques used are basic and not mathematically strict
3. Clustering is simple and may not capture very complex patterns
4. Limited features are used for analysis (mainly amount and frequency)
5. Not designed for large-scale real-world deployment
6. No real-time data processing is included

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-4 | April-2026 | Page 113-122

XIII. CONCLUSION

In this project, we tried to build a system that can analyze financial transaction data without directly exposing sensitive user information. Instead of using raw data, a privacy pipeline was applied where identity was removed, values were slightly modified, and only useful fields were kept.

After that, clustering was used to group users based on their spending behavior. This showed that even after applying privacy techniques, meaningful patterns can still be identified.

The system is simple but it gives a clear idea of how privacy and data analysis can work together. It shows that it is not always necessary to use raw data for getting insights.

Overall, the project demonstrates a basic but practical approach towards privacy-preserving financial analytics, which can be further improved for real-world use.

References

[1] C. Dwork, "Differential Privacy," Automata, Languages and Programming, Springer, 2006.

[2] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," IEEE Symposium on Security and Privacy, 2008.

[3] T. Li, N. Li, and J. Zhang, "Privacy-Preserving Data Mining," Springer, 2012.

[4] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Elsevier, 2011.

[5] S. R. Sweeney, "k-Anonymity: A Model for Protecting Privacy," International Journal of Uncertainty, 2002.

[6] Scikit-learn Documentation, "K-Means Clustering," [Online]. Available: https://scikit-learn.org

[7] Flask Documentation, "Flask Web Framework," [Online]. Available: https://flask.palletsprojects.com

[8] I. Goodfellow et al., Deep Learning, MIT Press, 2016.

[9] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," ACM SIGMOD, 2000.

[10] SQLite Documentation, "SQLite Database Engine," [Online]. Available: https://www.sqlite.org