Real-Time Smart Bank Data Streaming Capture

GROUP - 03

Sindhu Nagesha (017419987)
Prayag Nikul Purani (017416737)
Syed Faraaz Ahmed (017428619)
Sai Vivek Chunduri (017435301)

Department of Applied Data Science
San Jose State University
DATA 228: Big Data Tech and Applications
Submitted to: Vishnu S Pendyala
April 28, 2024

Abstract:

In the rapidly evolving landscape of digital banking, the demand for real-time data processing has become paramount in delivering personalized and responsive services to customers. This project presents the development and implementation of a real-time data analytics pipeline tailored specifically for the banking sector. The data is collected from various sources including sensors, transactions, and applications. A robust technological stack, comprising Docker for containerization, Kafka for data streaming, and Apache Spark for real-time data processing, forms the foundation of the pipeline, ensuring efficient data ingestion and analysis. Implementation of IOT and timestream serves as a streamlined process flow for Sensor data. For transactional and application data Apache Kafka leverages its capabilities, the streamed data is processed swiftly, enabling prompt analysis and the derivation of actionable insights. Integration with Flask, and Elastic search facilitates instantaneous visuals providing notifications and alerts, empowering banking personnel to promptly address critical events such as fraud detection or transaction anomalies. Through the deployment of this intelligent bank data pipeline, organizations stand to gain invaluable insights into customer behavior, promptly identify fraudulent activities in real time, and deliver tailored banking experiences. This project underscores the transformative potential of real-time data analytics in revolutionizing the banking sector, ultimately enhancing customer satisfaction and loyalty levels.

Introduction:

In today's digital era, data stands as the cornerstone of modern banking operations, driving efficiency and innovation. The convergence of rapid technological advancements and evolving customer expectations has propelled financial institutions into a realm where real-time access to actionable insights is indispensable. This project marks a pivotal journey to redefine how banks leverage data. We will be implementing data in two workflows: the first flow shows the sensor data which is connected to the Amazon IOT core where the connected devices securely interact with the cloud applications and other devices. Timestream automates the data from the memory store to the magnetic store by connecting through the API and AWS Lambda. Here a logic is implemented based on which fraud transactions are captured. After this, it is integrated with Python where conversion of data, and use of streaming algorithms is implemented and a

connection to Grafana is made that can be viewed as visuals in the application. The second workflow is implemented using application and transactional data where the data is sent and the connection is made through Python before streaming the data into Kafka where fraud detection logic is implemented. Apache Kafka sessions undergo tokenization and streaming algorithms like K-Anonymity, and LSH algorithms are implemented, which then get stored in MongoDB and visualized in Grafana. Alternatively, data is directly visualized through Flask, where the user location is checked and compared with the transaction location. The ability to capture, process, and leverage data instantaneously empowers banks to detect patterns, anticipate trends, and make precise decisions swiftly. Moreover, real-time smart bank data streaming capture holds the potential to transform customer experiences, enabling hyper-personalized services and enhancing engagement and loyalty. By proactively identifying and addressing issues such as fraud or disruptions, banks can cultivate trust and long-term relationships with customers. This project signifies more than a technological upgrade; it signifies a strategic shift towards a data-centric banking approach to reshape the industry landscape. Embracing real-time smart bank data streaming capture opens new avenues for innovation, competitiveness, and sustainable growth in a dynamic marketplace.

Literature Survey:

1. **Apache Spark: A Big Data Processing Engine**

Eman Shaikh, Iman Ahmed Mohiuddin, Yasmeen Alufaisan and Irum Nahvi (2019). Apache Spark: A Big Data Processing Engine. [online] ResearchGate. doi https://www.researchgate.net/publication/339176824_Apache_Spark_A_Big_Data_Processing_Engine.

In this paper, we learn how Big data refers to an excessively large amount of datasets that are used to computationally reveal patterns and trends. To analyze and find knowledge from this bulk of data, a processing framework is required. There are various types of commonly used big data frameworks such as Apache Hadoop, Apache Storm, Apache Spark, Apache Flink, etc. In this paper, we learn about Apache Spark's batch processing and stream processing abilities, use cases, ecosystem, architecture, multi-threading, and concurrency capabilities, and lastly the use of Spark in emerging technologies.

2. Information Security in Big Data: Privacy and Data Mining

Xu, L., Jiang, C., Wang, J., Yuan, J. and Ren, Y. (2014). Information Security in Big Data: Privacy and Data Mining. IEEE Access, 2(2), pp.1149–1176. doi:https://doi.org/10.1109/access.2014.2362522

Data mining can extract valuable knowledge and patterns from large datasets but also raises privacy concerns about sensitive personal information being disclosed. Privacy-Preserving Data Mining (PPDM) is a research area focused on modifying data to enable effective data mining while protecting sensitive information. Most PPDM work has looked at reducing privacy risks in the data mining operations phase. However, privacy threats can arise in other phases like data collection, publishing, and delivery of mining results. The paper takes a broader perspective, identifying four types of users involved in data mining applications: data provider, data collector, data miner, and decision maker.

For each user type, the paper discusses their privacy concerns and methods to protect sensitive information throughout the knowledge discovery process. In Addition to reviewing privacy-preserving approaches per user role, the paper also covers game theoretical approaches that analyze interactions and valuations of sensitive information among different users. The goal is to provide insights into PPDM by differentiating the privacy responsibilities of different user roles in safeguarding sensitive information throughout the data mining pipeline.

3. Beyond Batch Processing: Towards Real-Time and Streaming Big Data

Shahrivari, S. (2014). Beyond Batch Processing: Towards Real-Time and Streaming Big Data. *Computers*, [online] 3(4), pp.117–129, doi: https://doi.org/10.3390/computers3040117

This paper examines the limitations of traditional batch processing systems like Hadoop MapReduce in handling real-time queries, interactive jobs, and continuous data streams. It reviews emerging solutions for real-time processing, such as in-memory computing platforms and real-time query engines, as well as dedicated stream processing frameworks like Storm and S4. Through experimental results, the paper demonstrates the performance advantages of these new solutions over Hadoop for real-time and streaming workloads. The author concludes that

while batch processing with Hadoop is mature, in-memory computing approaches like Spark are becoming essential to meet the growing real-time and streaming needs of big data applications.

4. **Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges**

F. Gürcan and M. Berigel, "Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2018, pp. 1-6, doi: 10.1109/ISMSIT.2018.8567061.

This paper outlines the importance of real-time processing in today's technology landscape, especially in big data applications. It introduces a lifecycle for real-time big data processing, covering phases like data ingestion, storage, stream processing, analytical data storage, and analysis/reporting. The document explores various tools such as Flume, Kafka, Nifi, Storm, Spark Streaming, S4, Flink, Samza, HBase, Hive, Cassandra, Splunk, and Sap Hana, associating them with different lifecycle stages. Additionally, it addresses challenges like handling large and diverse data, ensuring consistency, scalability, real-time processing, data visualization, skill requirements, and privacy/security. The paper aims to provide insights into the lifecycle, tools, and challenges of real-time big data processing.

5. **KAFKA: The modern platform for data management and analysis in big data domain**

R. Shree, T. Choudhury, S. C. Gupta, and P. Kumar, "KAFKA: The modern platform for data management and analysis in the big data domain," 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, India, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343593.

The paper talks about how dealing with real-time data nowadays is quite complicated, with many technologies that need to work together. It suggests using Apache Kafka, which is like a smart system for handling streams of data. It acts as a messaging system and is good at storing data reliably. Kafka is useful for two main things: moving data between systems in real time and creating applications that work with live data streams. It works on multiple servers, storing data in categories called topics, each having a key, a value, and a timestamp. The paper explains

Kafka's structure and gives examples of how it helps solve problems in the Big Data era by using streaming solutions.

6. **Event-Based Sensor Data Scheduling: Trade-Off Between Communication Rate and Estimation Quality**

J. Wu, Q. -S. Jia, K. H. Johansson and L. Shi, "Event-Based Sensor Data Scheduling: Trade-Off Between Communication Rate and Estimation Quality," in IEEE Transactions on Automatic Control, vol. 58, no. 4, pp. 1041-1046, April 2013, doi: 10.1109/TAC.2012.2215253.

In this paper, the focus lies on addressing the challenge of sensor data scheduling for remote state estimation within networked control systems. Given the constraints of limited communication energy and bandwidth, the paper proposes an event-based sensor data scheduler tailored for linear systems. The objective is to enable sensors to make informed decisions on whether to transmit measurements to a remote estimator for further processing, striking a balance between communication rate and estimation quality. Through the derivation of a minimum mean-squared error (MMSE) estimator, the paper outlines a methodology for achieving this balance. By selecting appropriate event-triggering thresholds, the proposed scheduler aims to optimize the trade-off between sensor-to-estimator communication rate and remote estimation quality. The paper provides simulation examples to validate the proposed approach, demonstrating its effectiveness in practical scenarios. Additionally, the paper contributes to the broader field of sensor scheduling and remote estimation under communication constraints, building upon existing research while introducing novel insights and methodologies for improving system performance in resource-constrained environments.

System Architecture:

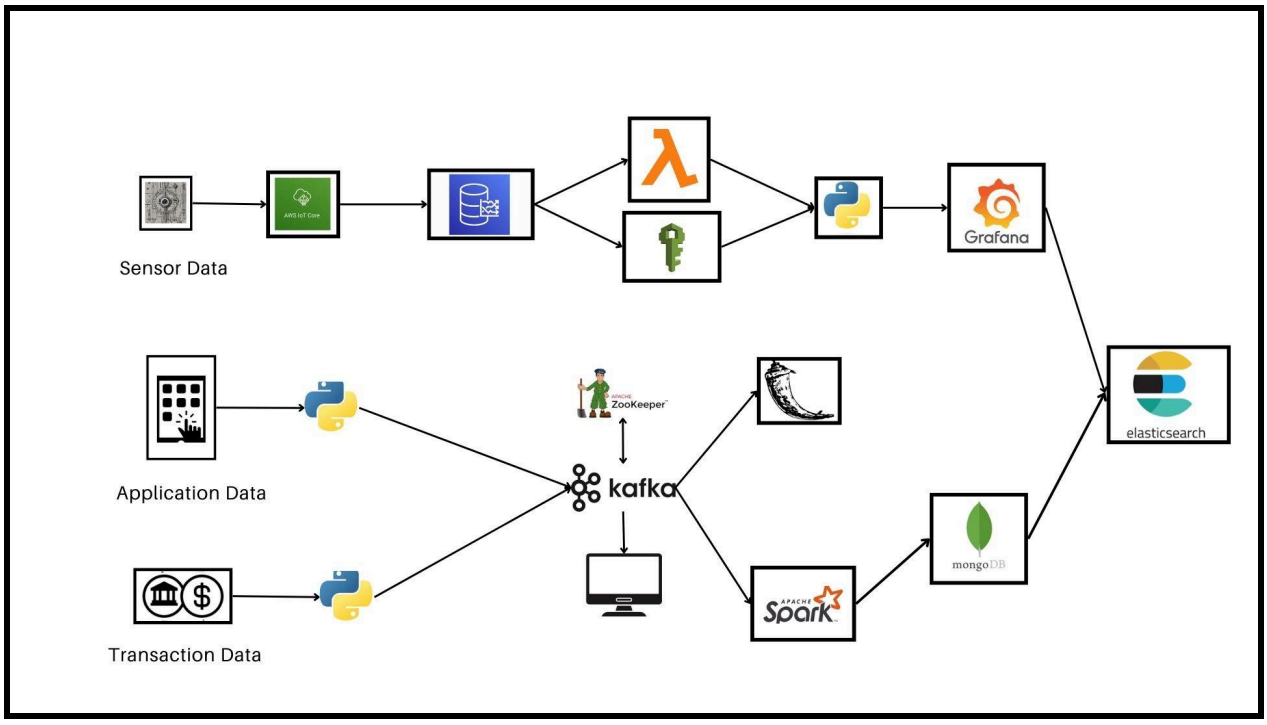


Fig 1: Overall project workflow

Workflow 1 (Sensor)

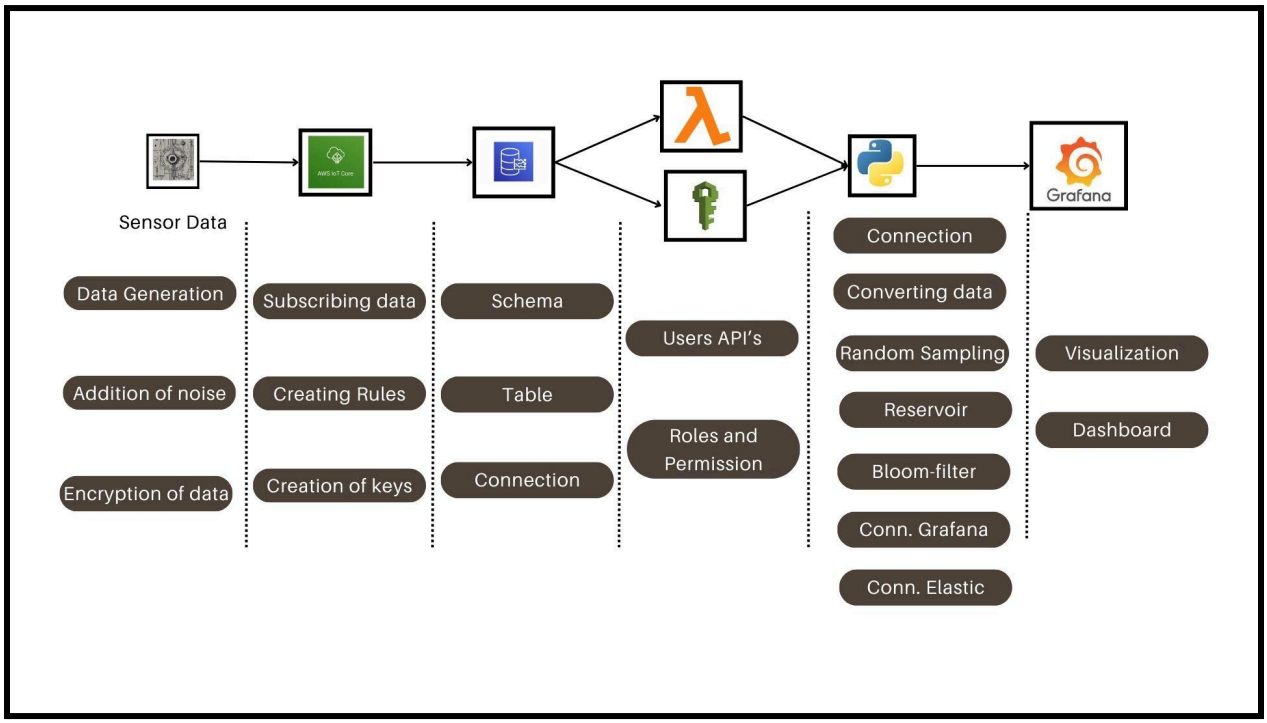


Fig 2: Workflow 1 architecture

There are two workflows that we have implemented in our project for efficiently detecting potential fraudulent transactions within a banking environment. The first workflow (Fig 2) leverages Sensor data where we aim to collect the data from laptop sensors and transmit it to AWS IoT services for further analysis and visualization. To enhance the dataset's utility, we introduced synthesized or 'fake' data alongside the original sensor data. This approach enriched the dataset, making it more representative and providing additional data points for analysis. We utilized techniques such as Bloom filters to efficiently handle large volumes of data and enable fast queries. The setup involved creating an AWS environment, setting up AWS IoT connections, and creating a database and table in AWS Timestream for schema definition. The storage time for the data was set to default, and the IAM user with appropriate permissions was created to establish a connection between Python and AWS Timestream. To interact with the data in AWS Timestream, we developed a function to fetch data in column form and decrypt it to obtain the original data. Additionally, we implemented a random sampling technique, specifically reservoir sampling, to create a sample for input into the Bloom filter. Parameters for reservoir sampling were specified, with potential tuning using different algorithms. In summary, our project workflow involved collecting sensor data, augmenting it with synthesized data, storing it in AWS Timestream, and then analyzing it using Python. Techniques such as Bloom filters and random sampling were employed to enhance data analysis and visualization capabilities, ultimately facilitating informed decision-making and driving innovation in the project.

Workflow 2 (Transaction & Application)

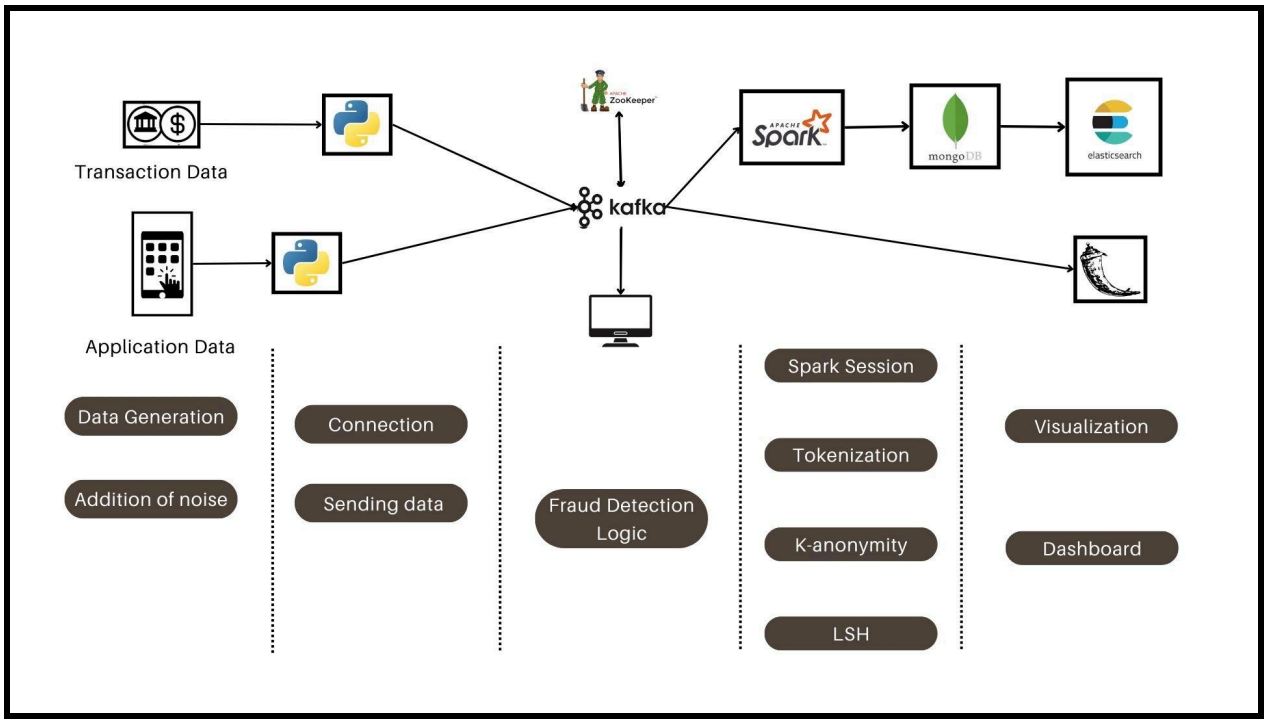


Fig 3: Workflow 2 architecture

In the next workflow (Fig 3) we leverage transaction and application data to mainly identify the potential fraud transaction. Firstly, the transaction data which consists of various key attributes related to transactions, such as TransactionID, UserID, DeviceID, Name, Amount, CardNumber, Merchant, Latitude, and Longitude, are collected and then pushed to Confluent Kafka, which is a cloud-based distributed streaming platform. In other words, the transaction data containing multiple important attributes is gathered and sent to Confluent Kafka, a cloud service that handles real-time data streams in a distributed manner. This Kafka acts as a buffer, enabling real-time data processing and facilitating data transfer between different systems. This ensures that the data is efficiently ingested and available for further processing.

Application data containing attributes like latitude, longitude, time, altitude, speed, and device ID is collected from a mobile app called Sensor Logger. This Sensor Logger application collects all the real-time location details of the mobile by using mobile location. This application data undergoes preprocessing in Python before being ingested into Confluent Kafka for further processing. Once in Confluent Kafka, the data goes through real-time streaming between systems, enabling seamless communication and data flow. At this stage, fraud logic is

implemented. The distance between the latitude and longitude coordinates from both the application and transaction data is calculated. Additionally, the time difference between the application and transaction data is determined. Using this information, the speed is calculated using the formula speed = distance/time. This calculated speed is then compared with the reported speed. If the calculated speed exceeds the reported speed, the transaction is flagged as a 'Potential fraud transaction' (Fig 4). Based on this logic, when a certain transaction is marked as potential fraud it is redirected back to a new Confluent Kafka topic 'Potential_fraud_transactions'. Then we use KSqlDb from the Confluent to perform some real-time analysis on those flagged transactions which helps to minimize latency in fraud detection. Subsequently, Flask and Dash (Fig 8) are used to generate visualizations that help understand the user's location details based on the application data.

Next, the data is read using PySpark, a powerful analytics engine for large-scale data processing in Python. PySpark's distributed computing capabilities allow for the efficient handling of big data sets. Once the data is loaded, a function is written to perform reservoir sampling. Reservoir sampling is a technique used to randomly select a subset of data points while preserving the overall distribution characteristics of the original data set. This sampling approach is crucial for implementing a Bloom filter with good accuracy. Data anonymization techniques are applied to protect user privacy while preserving the integrity of the data. Anonymization involves removing or encrypting personally identifiable information from the data set, ensuring that individual identities cannot be traced back to the underlying transactions.

Once the data is anonymized, Locality-Sensitive Hashing (LSH) is performed. LSH is a technique used for approximate nearest neighbor search, which helps to identify similar data points or patterns within the data set. By employing LSH, the data can be clustered based on similarity, enabling efficient fraud detection and identification of anomalous transactions.

After applying the necessary processing steps, the data is permanently stored in MongoDB using Python. Before storing the data in MongoDB, privacy techniques are employed to adhere to security protocols, such as masking the card numbers and user IDs. Subsequently, the data in MongoDB is connected to Elastic Kibana, which enables extracting meaningful insights and monitoring real-time data movement (Fig 9).

Results:

```
C:\Users\saivi\Desktop\Bigdata Project>python check_logic.py
Device ID 0d95505e-cd02-4eb2-9170-802d3fd46645 not found in location data.
Device ID 8b384fec-f203-48d6-9d01-81807a97edfb not found in location data.
Device ID 1682eab8-0749-4173-a39a-3cb74731f2cf not found in location data.
Device ID 1682eab8-0749-4173-a39a-3cb74731f2cf not found in location data.
Device ID a59b5b91-124f-4172-91d0-4feb82ce7631 not found in location data.
Device ID 58e62a06-0cbf-4ca5-8ee1-8517c5e6a19c not found in location data.
Potential Fraud Detected for Transaction ID: 99f8bfea-e947-40ce-a80f-80f476ebe452
Transaction Details: User ID: DEAD975D, Card Number: 62358672243501927, Amount: 481.84
Transaction Time: 2024-04-27 14:27:43.107526, Location Time: 2024-04-27 14:27:39, Time Difference: 0.06845876666666667 minutes
Transaction Location: ('37.335816', '-121.895130'), Device Location: (37.3359243, -121.8953148), Distance: 20.314204038788223 meters
Reported Speed at Location: 0.3325222134590149, Calculated Speed: 4.94560570980883
Device ID b4acb24f-f887-4d02-afb2-a7ceb362e586 not found in location data.
Device ID 77c5615a-bada-4e3f-aea5-391b8d1041f4 not found in location data.
Device ID c404a97d-aaff-4b7c-b9f4-9c0f16aa33b7 not found in location data.
Device ID 3185c75b-c64e-485f-86c3-e4a2dae7239a not found in location data.
Device ID 4c87fdfb-6a9f-4793-8ba7-00072749a73a not found in location data.
Device ID a47e643f-d8ff-4a4a-b7b2-b9914c64ae0d not found in location data.
Device ID a4d3e01e-2301-4e21-bee8-5a851950cb58 not found in location data.
Device ID 78ca5a2d-c6b2-4266-a204-307c4f8c93b9 not found in location data.
Device ID 1c58584c-8e5c-4047-843b-5bed73327692 not found in location data.
No Fraud Detected for Transaction ID: ec188137-34d7-4cee-9942-ecc1c9602c24
Transaction Details: User ID: DEAD975D, Card Number: 4124731290123316, Amount: 346.87
Transaction Time: 2024-04-27 14:27:33.122382, Location Time: 2024-04-27 14:28:30, Time Difference: 0.9479603 minutes
Transaction Location: ('37.335603', '-121.894946'), Device Location: (37.3359035, -121.8952828), Distance: 44.75599406926141 meters
Reported Speed at Location: 0.08960304409265518, Calculated Speed: 0.786882356241807
Device ID bd8bbd24-8764-489b-80db-d8deae31e843 not found in location data.
Device ID 51b91fc3-ca1f-417f-ade9-82b10ed6a9e5 not found in location data.
Device ID 410ad96d-660c-4f7f-a3fb-fe9277b7118b not found in location data.
Device ID 09d9c01a-3848-4af6-865c-81ba845a1fb4 not found in location data.
Device ID f446dba9-fe1d-4ef1-b5bc-74e503033a8e not found in location data.
Device ID ba39935c-d91d-4a63-a2e1-3081ea6c22a1 not found in location data.
Device ID 3ebb9b00-702f-4c97-bb51-3d5244d4e47d not found in location data.
```

Fig 4: Implementing the fraud detection logic for Workflow 2

Fig 5: Grafana Dashboard for Workflow1

The output of the sensor data is passed to Grafana and this is a real-time dashboard that has all the parameters plotted in a line graph. The spikes in the data indicate a sudden change in the

user's location, which is considered unacceptable because it would require the user to be traveling at an unrealistically high speed. To detect such anomalies, a threshold speed of 0.2 (units unspecified) has been chosen. However, this threshold may vary depending on the institution's infrastructure. Due to a limitation in Grafana, where it accepts only three integer digits, it becomes challenging to implement the intended logic for data visualization. Transporting the data from Python to Grafana results in a loss of precision as Grafana will not display digits after the three decimal places. Consequently, only the data that can be easily visualized and interpreted in terms of location changes is passed to Grafana. In summary, while spikes in the data suggest potential fraudulent activity based on unrealistic travel speeds, the visualization in Grafana is limited by its precision constraints, leading to a trade-off where only the data necessary to understand location changes is displayed.

The graph below shows the process of determining the optimal k value for the project. The best k value will be selected based on the data characteristics. However, to select the optimal value, we need to develop a model for this step. Once the model is created, we will use the chosen optimal k value to anonymize the data.

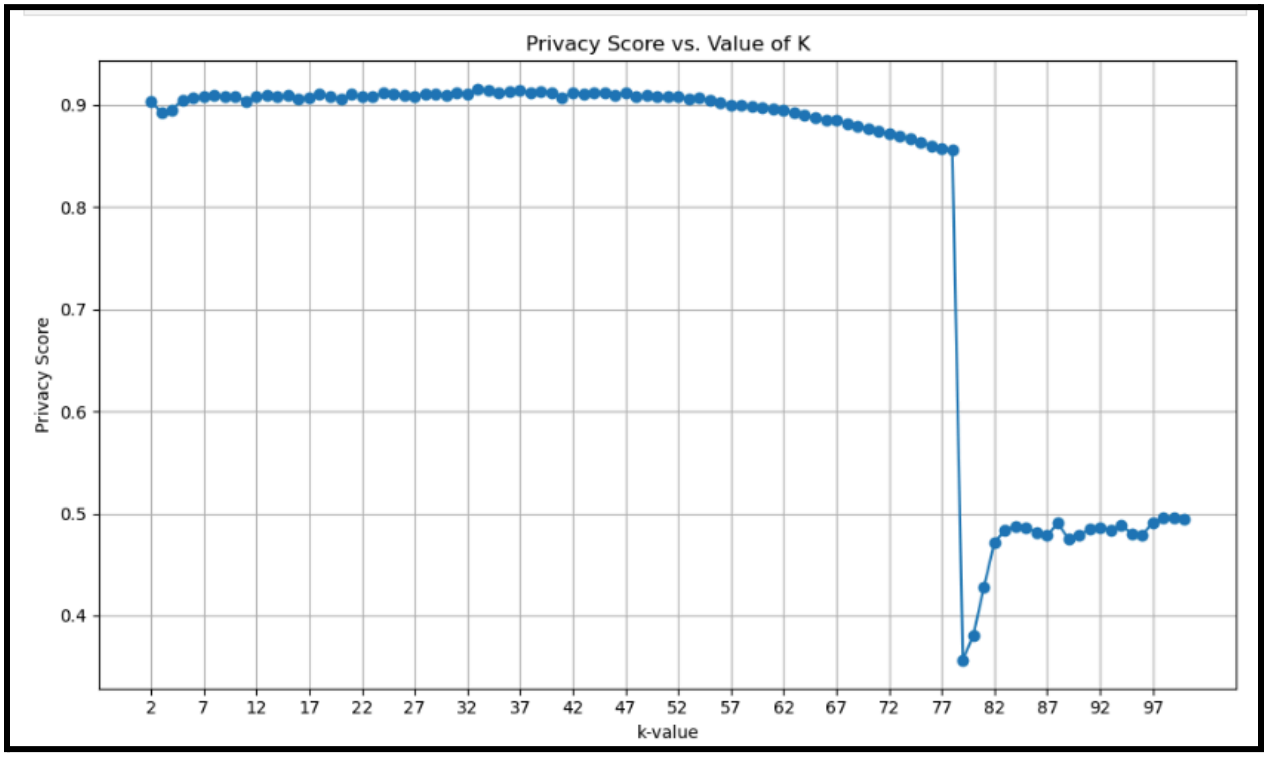


Fig 6: Privacy Score vs Value of K

The below graph shows how the data will be marked as clusters and the representation of the cluster is shown below.

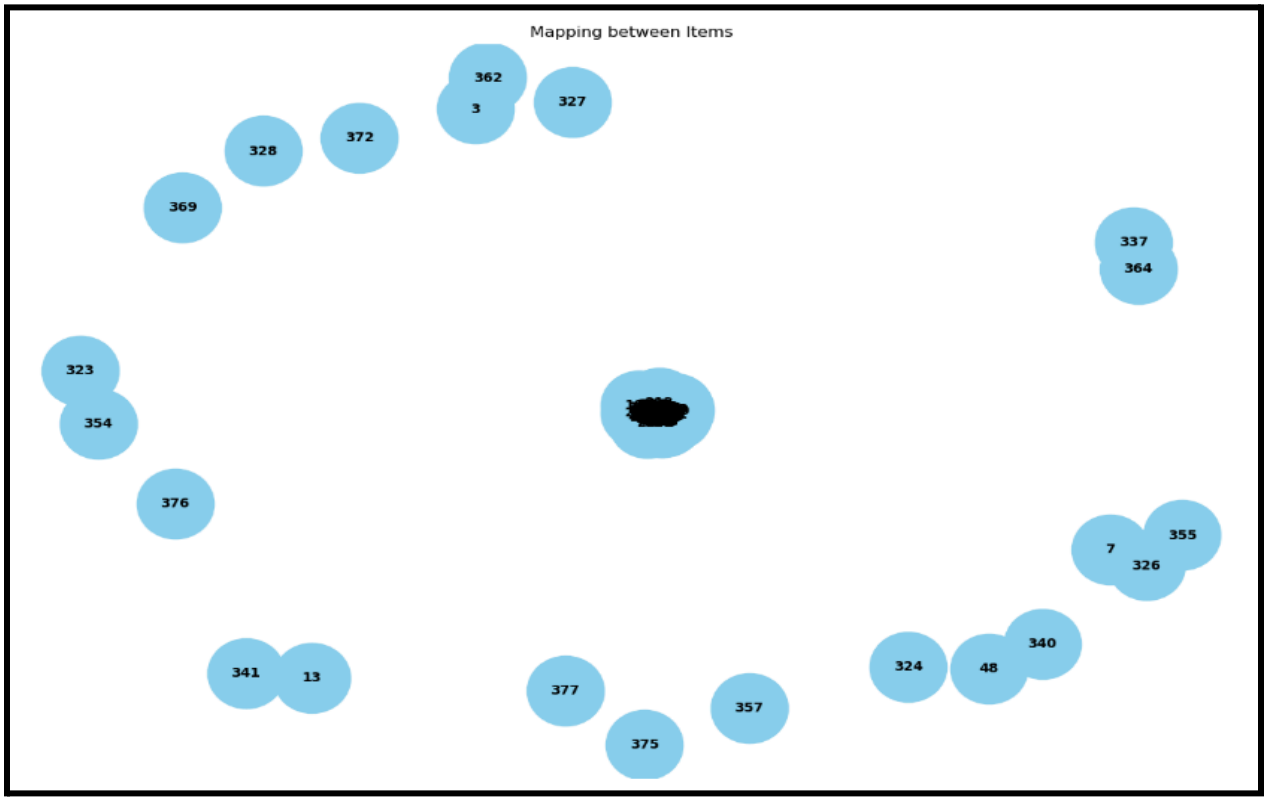


Fig 7: Mapping between items using LSH

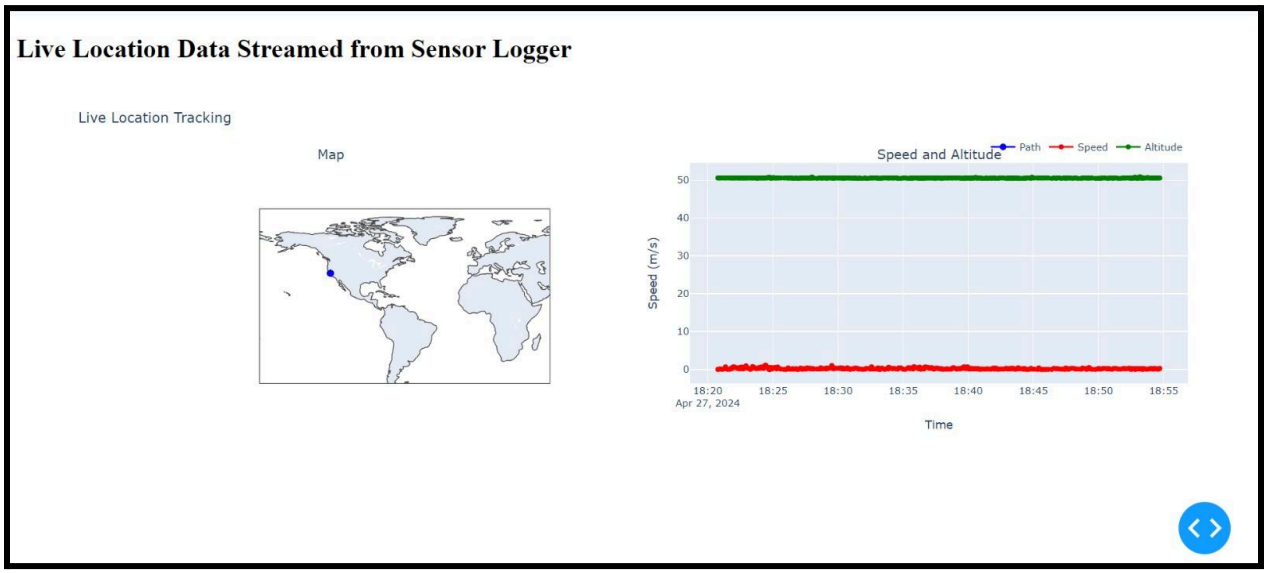


Fig 8: Live Location Data Streamed from Sensor Logger

Fig 9:Dashboard from Elastic Kibana for Workflow 2

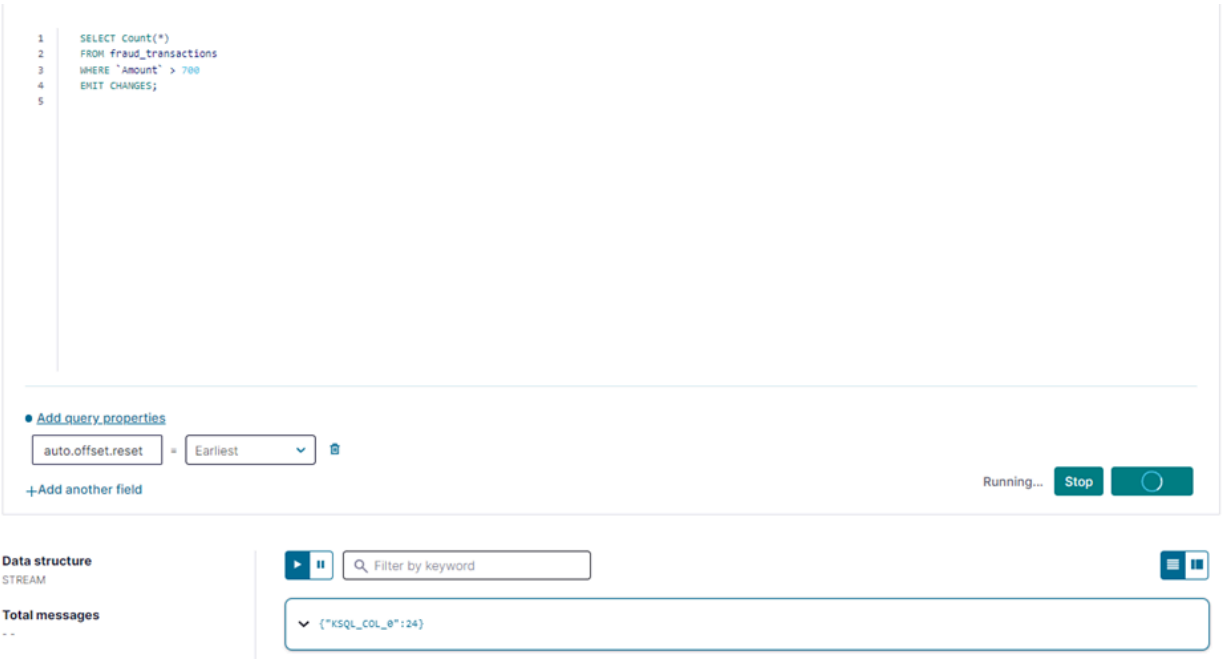


Fig 10: Querying in KsqlDb for detecting the number of people with high value transactions in the Potentially fraud transactions.

Key Learnings:

- Synthesizing Data: Introducing synthesized or 'fake' data alongside original sensor data can enrich datasets, making them more representative and providing additional data points for analysis.
- Efficient Data Handling: Techniques like Bloom filters are employed to efficiently handle large volumes of data and enable fast queries, enhancing data analysis capabilities.
- AWS Integration: Leveraging AWS IoT services and AWS Timestream for data collection, storage, and analysis provides a scalable and reliable infrastructure for real-time analytics.
- Python Integration: Python is utilized for data manipulation, analysis, and visualization, showcasing its versatility in integrating with various data processing technologies.
- Reservoir Sampling: Employing reservoir sampling techniques ensures the creation of representative data subsets, which are vital for accurate analysis and the implementation of Bloom filters.
- Data Anonymization: Implementing data anonymization techniques safeguards user privacy while preserving data integrity, which is essential for compliance with privacy regulations.
- Locality-Sensitive Hashing (LSH): Utilizing LSH facilitates efficient clustering of data based on similarity, enabling effective fraud detection and identification of anomalous transactions.
- Real-time Streaming: Leveraging Kafka for real-time data streaming enables efficient communication and data flow between different systems, facilitating seamless processing and analysis.
- Web Application Development: Flask in association with Dash is utilized for developing web applications and APIs, enabling users to interact with processed data and access analytical insights.
- Analytics with Elasticsearch: Elasticsearch provides robust analytics functionalities, empowering users to perform diverse analyses and visualize data, enhancing decision-making and operational efficiency.

Conclusion:

In conclusion, the implementation of a real-time data analytics pipeline tailored for the banking sector represents a significant leap forward in meeting the evolving demands of digital banking. By leveraging a robust technological stack, including Docker, Kafka, and Apache Spark, alongside innovative integrations like IOT and Timestream for sensor data, and Python for transactional and application data, this project enables efficient data processing and analysis.

The deployment of Flask, Elastic search, and Grafana further enhances the capabilities of the pipeline, empowering banking personnel with instantaneous visuals, notifications, and alerts for prompt action, particularly in critical scenarios such as fraud detection or transaction anomalies.

Through the intelligent utilization of real-time data analytics, organizations can gain invaluable insights into customer behavior, promptly identify fraudulent activities, and deliver tailored banking experiences, ultimately leading to enhanced customer satisfaction and loyalty.

This project underscores the transformative potential of real-time data analytics in revolutionizing the banking sector, emphasizing its role as a strategic shift towards a data-centric approach, poised to reshape the industry landscape and drive innovation, competitiveness, and sustainable growth in the dynamic marketplace of today and tomorrow.

Future Work:

Future works for this project could involve enhancing the fraud detection algorithms by incorporating machine learning models, such as anomaly detection algorithms or deep learning architectures, to improve the accuracy of identifying fraudulent activities. Additionally, exploring advanced data anonymization techniques, such as differential privacy or homomorphic encryption, can further bolster data security and privacy while preserving the integrity of the data. Integration of real-time monitoring and alerting systems can be implemented to provide immediate notifications of suspicious transactions or anomalies, enabling proactive fraud prevention measures. Furthermore, exploring the potential of integrating external data sources, such as social media or third-party APIs, can enrich the dataset and provide additional context for fraud detection and customer profiling. Overall, continuous optimization and refinement of

the data analytics pipeline, along with staying ahead of emerging technologies and industry trends, will be crucial.

References:

https://aws.amazon.com/iot/

https://aws.amazon.com/iot-core/

https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

https://aws.amazon.com/iot-core/features/

https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html

https://aws.amazon.com/iot-core/getting-started/

https://aws.amazon.com/timestream/

https://docs.aws.amazon.com/timestream/latest/developerguide/what-is-timestream.html

https://aws.amazon.com/timestream/pricing/

https://k21academy.com/amazon-web-services/amazon-timestream/

https://github.com/awslabs/amazon-timestream-tools

https://docs.aws.amazon.com/general/latest/gr/timestream.html

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/timestream-write.html

https://docs.aws.amazon.com/timestream/latest/developerguide/getting-started.python.html

https://docs.aws.amazon.com/timestream/latest/developerguide/sample-apps.html

https://www.youtube.com/watch?v=9snyPffdNRc

https://www.youtube.com/watch?v=zkbCswmr2ZQ

https://aws-dojo.com/excercises/excercise24/

https://docs.aws.amazon.com/timestream/latest/developerguide/getting-started.db-w-sample-data.html

https://docs.aws.amazon.com/iot/latest/developerguide/timestream-rule-action.html

https://docs.aws.amazon.com/timestream/latest/developerguide/IOT-Core.html

https://www.youtube.com/watch?v=00Wersoz2Q4

https://www.youtube.com/watch?v=zkbCswmr2ZQ

https://www.youtube.com/watch?v=z8T4hAERuOg

https://aws.amazon.com/blogs/database/patterns-for-aws-iot-time-series-data-ingestion-with-amazon-timestream/
https://www.youtube.com/watch?v=Byik_DSPymA&t=6576s
https://www.youtube.com/watch?v=pilkz645cs4&t=1s
https://grafana.com/docs/pyroscope/latest/configure-client/language-sdks/python/
https://grafana.com/docs/grafana-cloud/monitor-applications/application-observability/setup/instrument/python/
https://grafana.com/docs/grafana-cloud/monitor-applications/application-observability/setup/quickstart/python/
https://medium.com/swlh/create-grafana-dashboards-with-python-14a6962eb06c
https://www.reddit.com/r/kubernetes/comments/14xbht2/write_grafana_dashboards_in_python_without_losing/
https://stackoverflow.com/questions/70813724/programmatically-using-python-import-grafana-dashboard-data-from-grafana-websit
https://community.grafana.com/t/invoking-python-script-from-dashboard/8051
https://community.grafana.com/t/oss-tool-write-grafana-pre-build-dashboards-in-python/91342
https://pypi.org/project/grafana-api/

Appendix

1. Technical Difficulties:

Some of the difficulties we faced while implementing this project include:

- First, the transaction data was not available and for that reason, we need to generate our data. Also, the data was available but it was not in the condition that we wanted and this is one more reason for generating the data.
- The next part is when we are using Debezium and Postgres we were not able to connect them and so, because of this reason we have to drop that plan.
- After that, during the implementation of sensor data, connection with Grafana was also difficult and the main problem was the SQL query that was required for the creation of the graphs.

- The other thing is the use of paid technologies like confluent which expires in one month and also, elastic which gives a free trial for only 7 days so need to create a different account to complete the project.

2. Innovation:

Real-time data analytics presents significant opportunities for improving operational efficiency and customer satisfaction in the banking sector.

- The main and important innovation is the utilization of longitude, latitude, and altitude data from sensors, combined with speed and directional data (x, y, z) from the application.
- Dynamic customer segmentation models that utilize real-time data to precisely tailor services and marketing strategies, ensuring personalized experiences.
- Predictive maintenance initiatives combine sensor data and predictive analytics to preemptively address potential system failures, minimizing downtime and enhancing reliability.
- Continuous compliance monitoring systems analyze real-time data streams to ensure regulatory adherence and promptly detect anomalies, mitigating compliance risks.

3. Possible best practices:

For this project involving real-time transaction analysis for fraud detection, adhering to best practices is crucial for robustness, efficiency, and maintainability. Here are some best practices we suggest:

- When working with Docker, it is recommended to delete old instances and install new images to avoid potential port conflicts.
- When dealing with sensitive information such as card numbers, locations, and user IDs, it is crucial to follow privacy techniques like hashing or masking the data before storing it in a database or making it publicly accessible, to ensure compliance with regulations like GDPR.
- When handling real-time data, always verify the formatting of the incoming data to ensure data integrity.

- When using cloud instances, it is advised to pause or shut down the instances when not in use to avoid unnecessary costs.
- When dealing with real-time incoming data, continuously monitor system performance and optimize both the software and infrastructure. This includes tuning Kafka topics, MongoDB collections, and Elasticsearch indices for optimal performance.

4. Lessons Learnt:

The exploration of real-time data analytics within the banking sector has yielded invaluable lessons showing the importance of technological innovation and strategic implementation.

- Firstly, the adoption of a comprehensive technological stack, including Docker, Kafka, and Apache Spark, has demonstrated the significance of selecting robust tools capable of handling complex data streams efficiently.
- This highlights the necessity of investing in cutting-edge technologies to ensure seamless data processing and analysis.
- Secondly, data-capturing techniques have proven pivotal in capturing and analyzing real-time data alterations, emphasizing the importance of staying abreast of emerging methodologies to enhance data capture capabilities.
- Thirdly, the implementation of advanced fraud detection models and dynamic customer segmentation strategies has showcased the transformative potential of real-time analytics in mitigating risks and enhancing customer experiences.
- Overall, the project has emphasized the critical role of real-time data analytics in revolutionizing banking operations and delivering tailored, responsive services to customers, thereby paving the way for future advancements in the field.

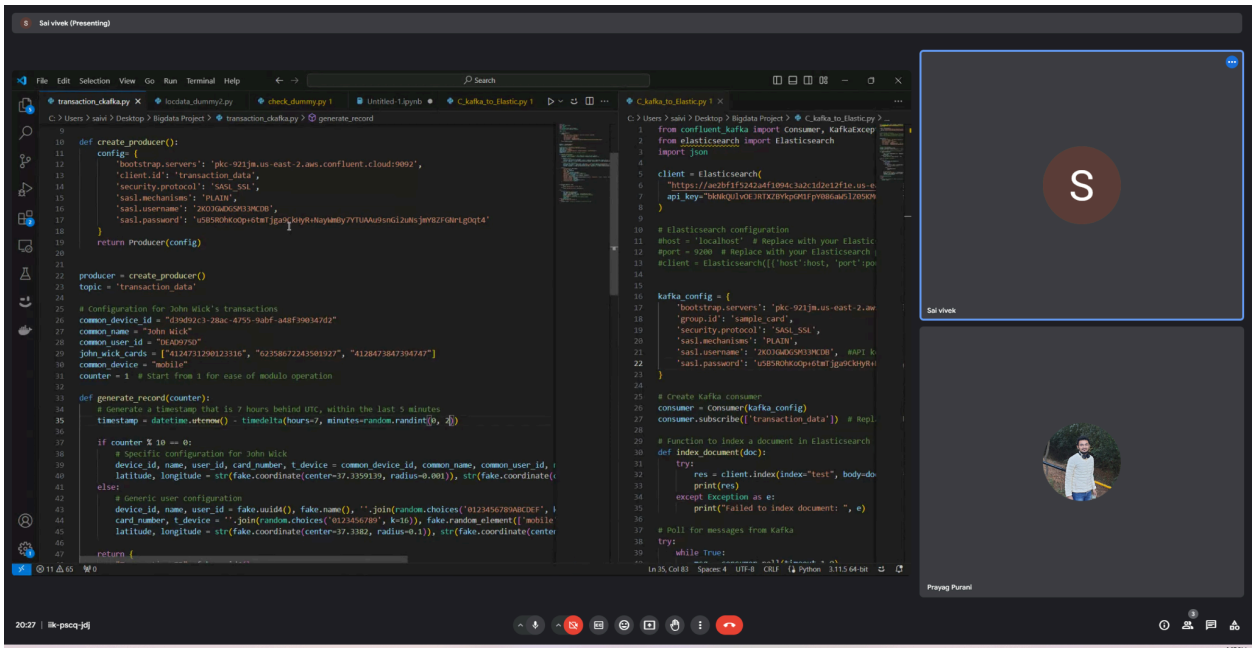
5. Prospects of winning competition/ Publication:

Here's a concise one-paragraph correction: The real-time data analytics project in banking harnesses Apache Spark, Kafka, and Docker for seamless data streaming, processing, and real-time analysis. This robust technological stack enables swift fraud detection, dynamic customer segmentation, and personalized experiences, addressing critical issues while ensuring customer trust. The project prioritizes data privacy and security through anonymization and encryption protocols, complying with CCPA and GDPR while setting benchmarks for

responsible data management practices across industries. Its innovative methodologies and scalable infrastructure hold transformative potential across other sectors like e-commerce, healthcare, and supply chain management, driving operational efficiencies and innovation through real-time insights, proactive risk management, and enhanced customer experiences. Furthermore, the industry-standard fraud detection logic can handle growing data volumes without performance degradation, making it particularly relevant for fields like fintech with increasing data volumes. This combination of advanced technologies, data privacy measures, and scalable fraud detection logic increases the project's potential for publication or competition success.

6. Pair Programming:

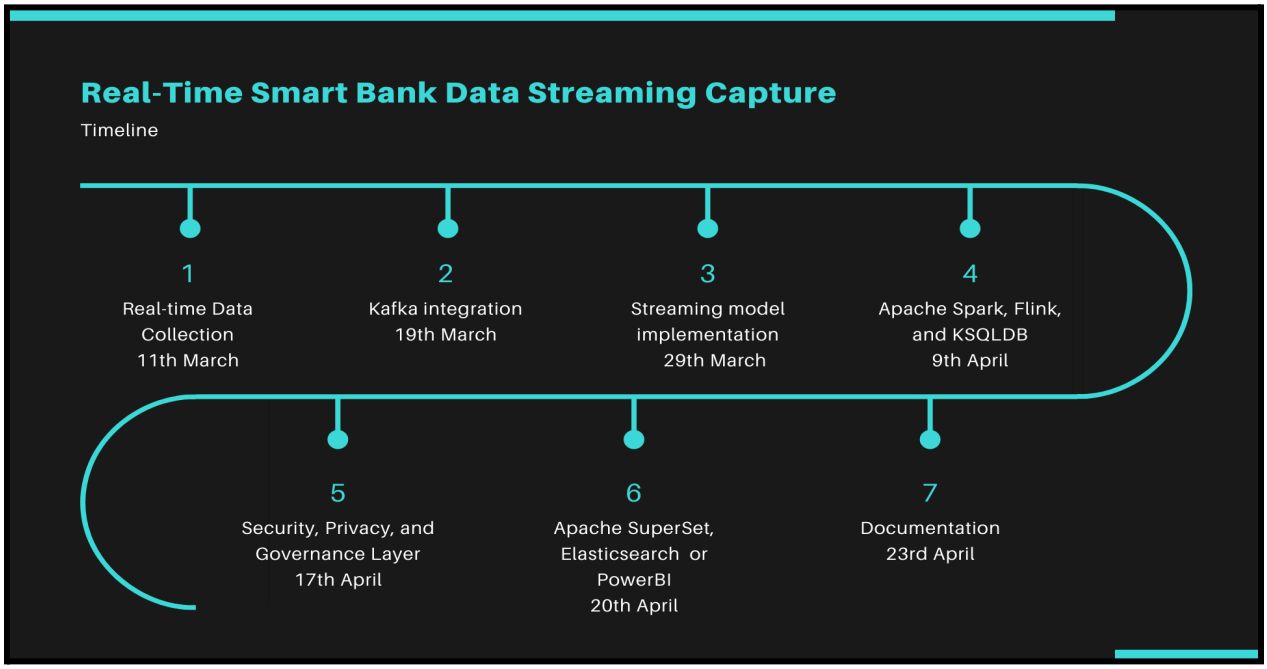
Pair programming can significantly enhance the efficiency and effectiveness of various stages and components within the real-time data analytics project for the banking sector. Firstly, in the initial phase of data preprocessing and analysis, we collaborated to ensure the cleanliness and consistency of raw data collected from banking systems. Together, we employed advanced techniques such as feature engineering and anomaly detection using tools like PySpark and Pandas. Secondly, in the domain of integration and testing, paired up as teams to seamlessly integrate various components of the data pipeline, including Kafka streams, Apache Spark jobs, and Elasticsearch indices. Through collaborative efforts, as a result, we were able to do comprehensive unit tests, integration tests, and end-to-end tests to validate the correctness and reliability of the entire system. Thirdly, in dashboard development, implemented interactive dashboards using tools like Grafana and Flask. Focused on real-time data streaming integration to deliver intuitive and insightful dashboards. Furthermore, in infrastructure setup and deployment, to configure and deploy the necessary infrastructure on cloud platforms like AWS. Collaborating on resource provisioning, networking configuration, and Docker container deployment, ensured scalability, reliability, and security of the system. Lastly, we all collaborated in documentation and knowledge sharing, it is important to document design decisions, implementation details, and best practices. Overall, by implementing pair programming in these critical areas, we were able to achieve collective expertise, foster collaboration and communication, and accelerate the development and deployment of the real-time data analytics solution for the banking sector.



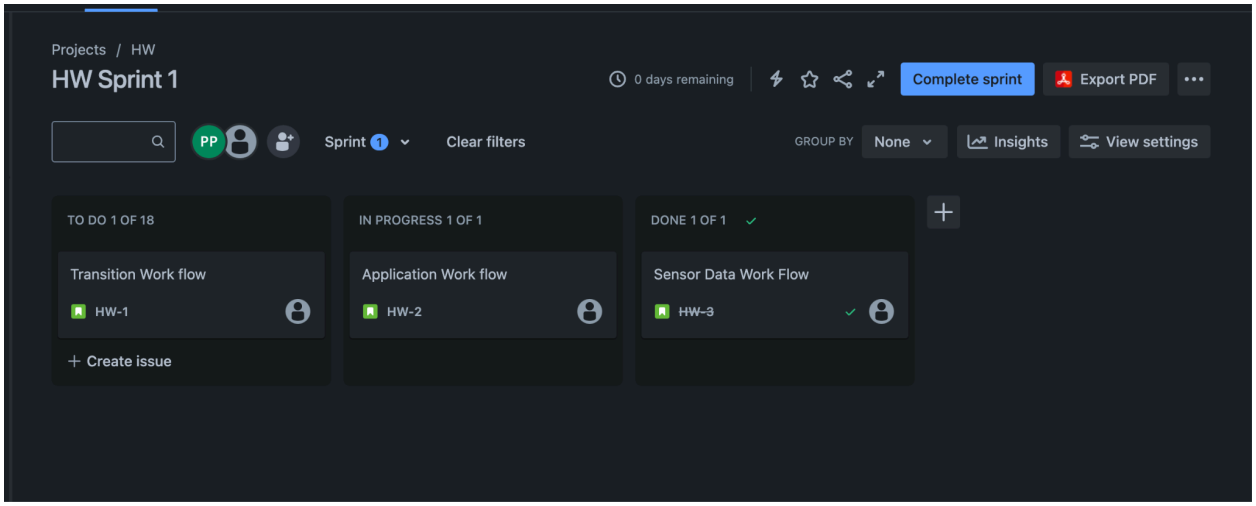
7. Contribution:

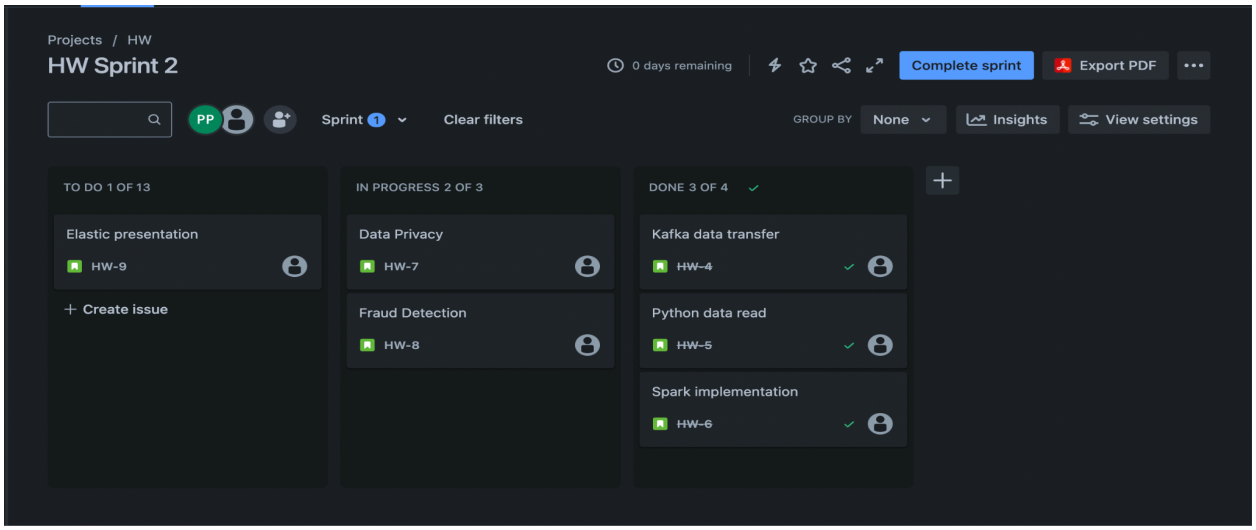
Name	Contribution description
Prayag Nikul Purani	Sensor workflow, Significant paper, Documentation, PPT.
Sai Vivek Chunduri	Transaction workflow, Application workflow, Significant paper, Documentation, PPT.
Sindhu Nagesha	Industry Case Study, Sensor workflow Dashboard, Documentation, PPT.
Syed Faraaz Ahmed	Industry Case Study, Transaction workflow Dashboard, PPT, Documentation.

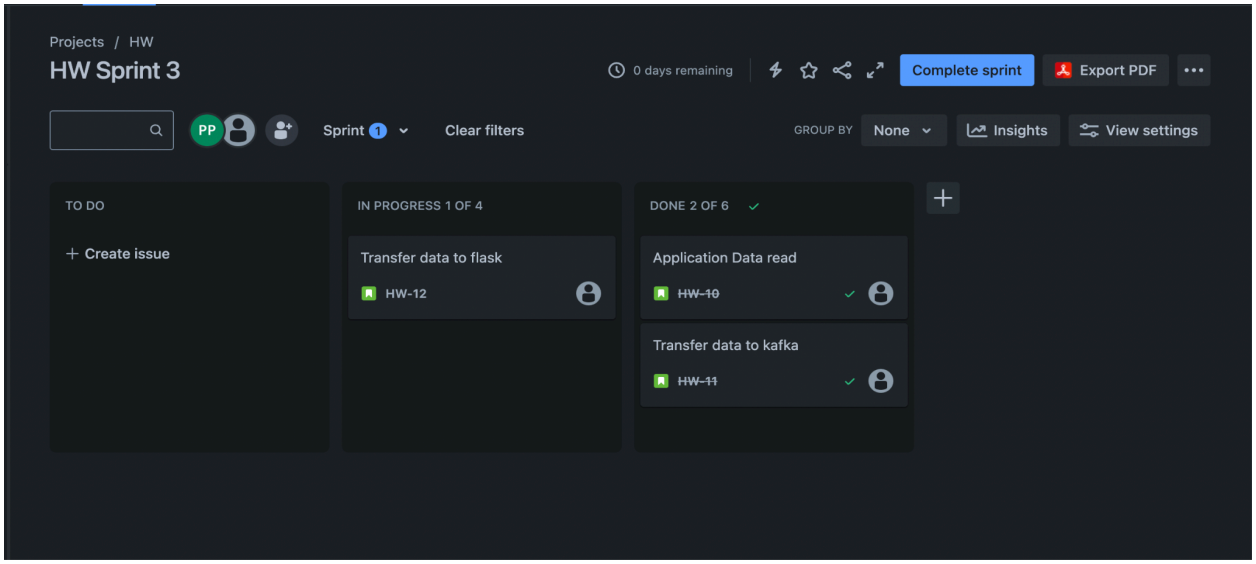
8. Timeline:

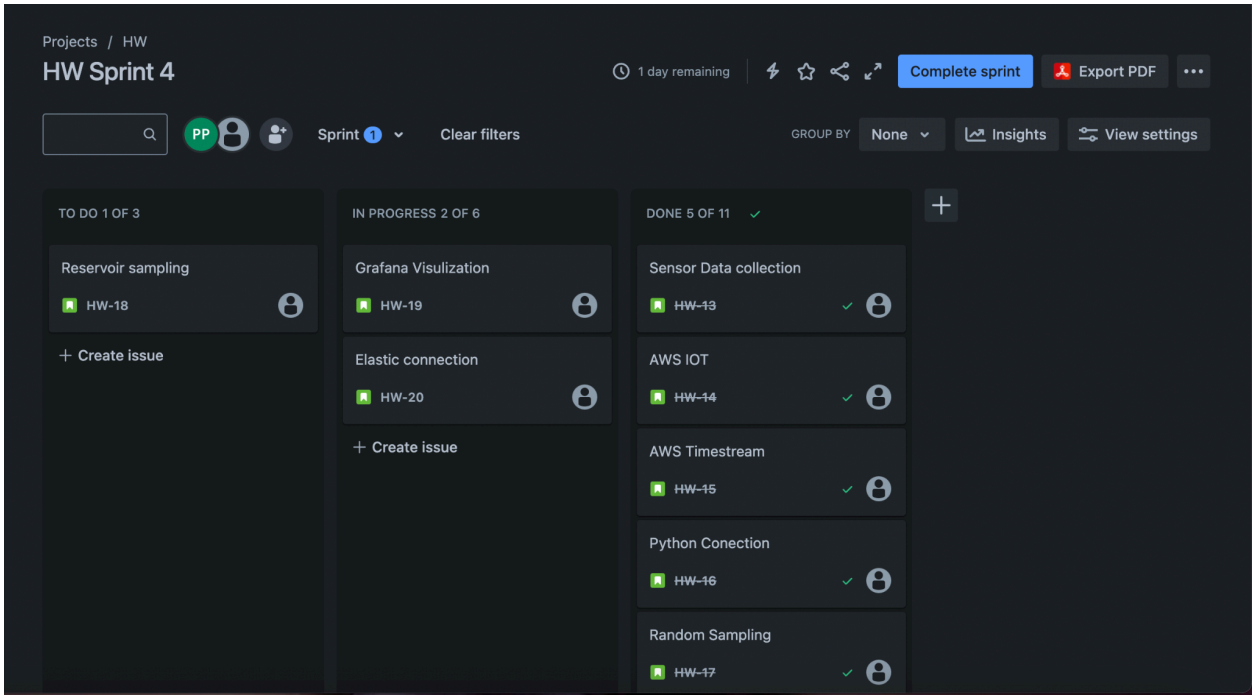


9. Kanban sprint:

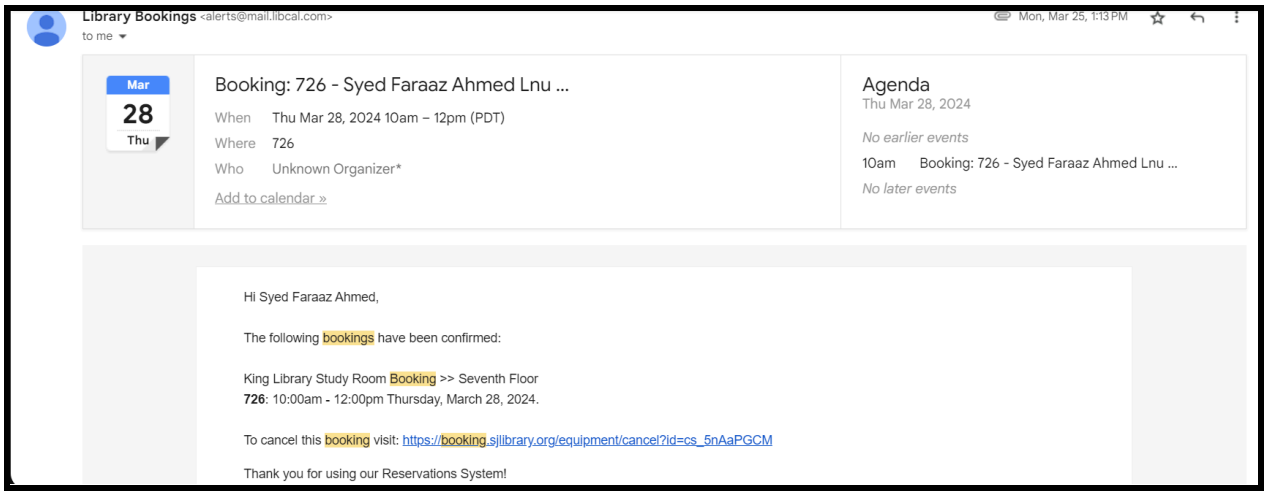

Projects / HW
HW Sprint 1
0 days remaining
Complete sprint
Export PDF
PP
Sprint 1
Clear filters
GROUP BY
None
Insights
View settings
TO DO 1 OF 18
Transition Work flow
HW-1
Create issue
IN PROGRESS 1 OF 1
Application Work flow
HW-2
DONE 1 OF 1
Sensor Data Work Flow
HW-3


Projects / HW
HW Sprint 2
0 days remaining
Complete sprint
Export PDF
PP
Sprint 1
Clear filters
GROUP BY
None
Insights
View settings
TO DO 1 OF 13
Elastic presentation
HW-9
Create issue
IN PROGRESS 2 OF 3
Data Privacy
HW-7
Fraud Detection
HW-8
DONE 3 OF 4
Kafka data transfer
HW-4
Python data read
HW-5
Spark implementation
HW-6





10. Project Planning and execution proof of teamwork: As a team, we have employed each other's skills to pitch in for project completion. The following are some of our library room bookings for our meeting.



Library Bookings <alerts@mail.libcal.com>
to me

Mon, Mar 25, 1:13 PM

Mar
28
Thu

Booking: 726 - Syed Faraaz Ahmed Lnu ...

When Thu Mar 28, 2024 10am – 12pm (PDT)
Where 726
Who Unknown Organizer*

Add to calendar »

Agenda
Thu Mar 28, 2024

No earlier events
10am Booking: 726 - Syed Faraaz Ahmed Lnu ...
No later events

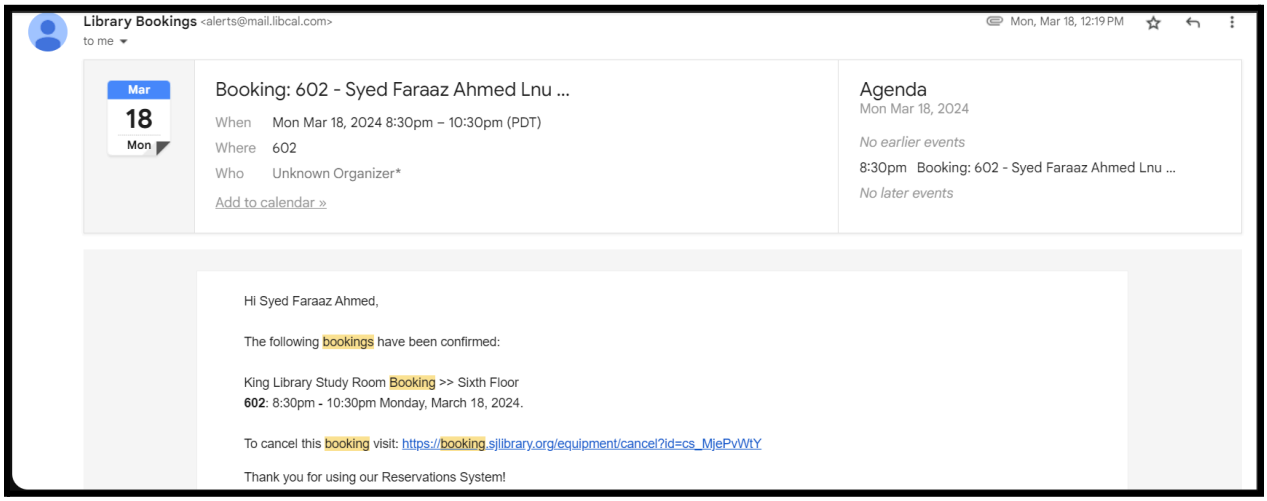
Hi Syed Faraaz Ahmed,

The following bookings have been confirmed:

King Library Study Room Booking >> Seventh Floor
726: 10:00am - 12:00pm Thursday, March 28, 2024.

To cancel this booking visit: https://booking.sjlibrary.org/equipment/cancel?id=cs_5nAaPGCM

Thank you for using our Reservations System!



Library Bookings <alerts@mail.libcal.com>
to me

Mon, Mar 18, 12:19 PM

Mar
18
Mon

Booking: 602 - Syed Faraaz Ahmed Lnu ...

When Mon Mar 18, 2024 8:30pm – 10:30pm (PDT)
Where 602
Who Unknown Organizer*

Add to calendar »

Agenda
Mon Mar 18, 2024

No earlier events
8:30pm Booking: 602 - Syed Faraaz Ahmed Lnu ...
No later events

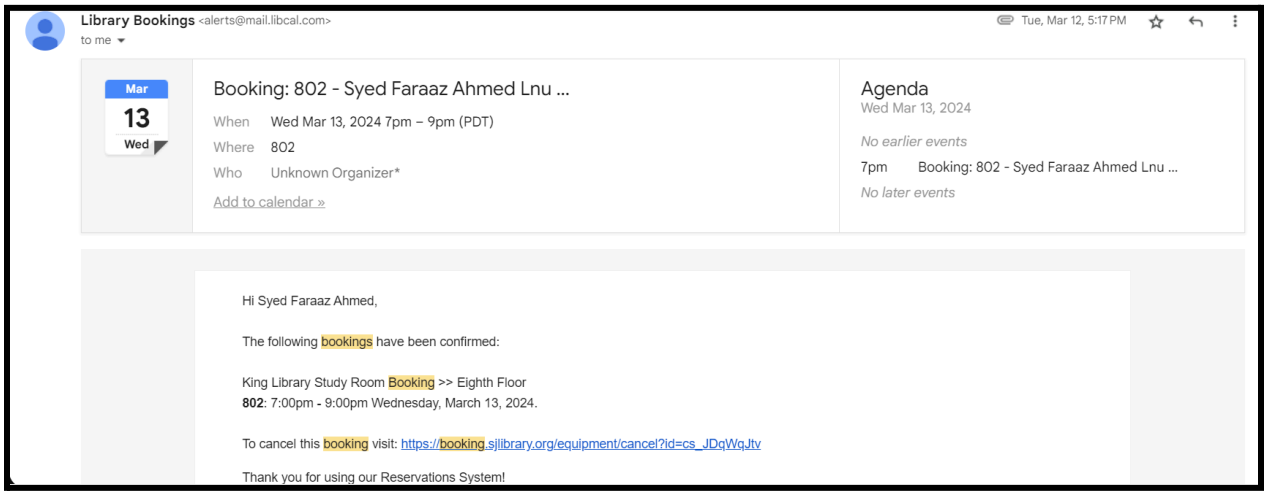
Hi Syed Faraaz Ahmed,

The following bookings have been confirmed:

King Library Study Room Booking >> Sixth Floor
602: 8:30pm - 10:30pm Monday, March 18, 2024.

To cancel this booking visit: https://booking.sjlibrary.org/equipment/cancel?id=cs_MjePvWtY

Thank you for using our Reservations System!



Library Bookings <alerts@mail.libcal.com>
to me

Tue, Mar 12, 5:17 PM

Mar
13
Wed

Booking: 802 - Syed Faraaz Ahmed Lnu ...

When Wed Mar 13, 2024 7pm – 9pm (PDT)
Where 802
Who Unknown Organizer*

Add to calendar »

Agenda
Wed Mar 13, 2024

No earlier events
7pm Booking: 802 - Syed Faraaz Ahmed Lnu ...
No later events

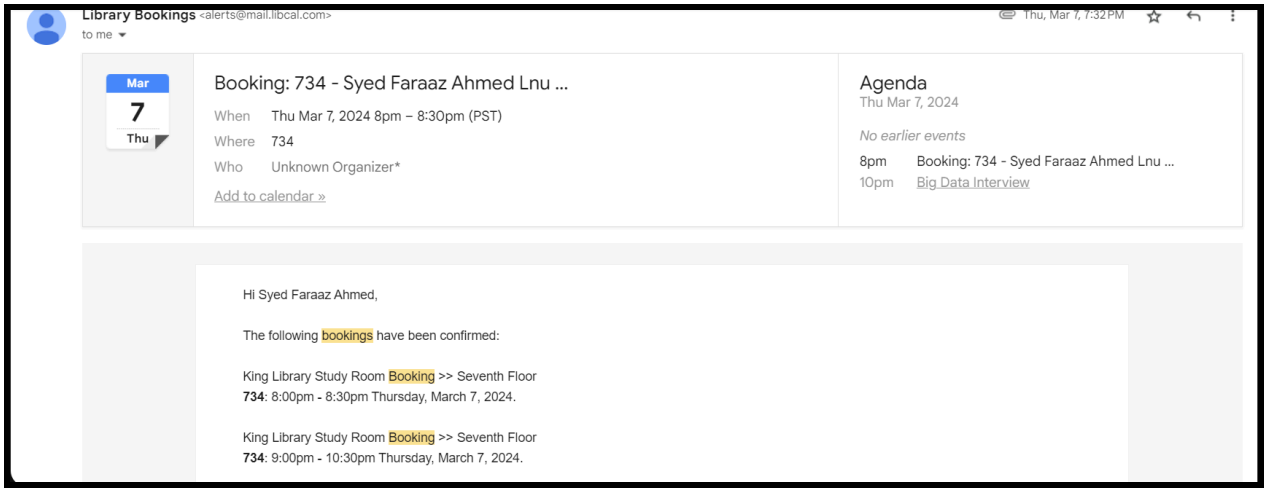
Hi Syed Faraaz Ahmed,

The following bookings have been confirmed:

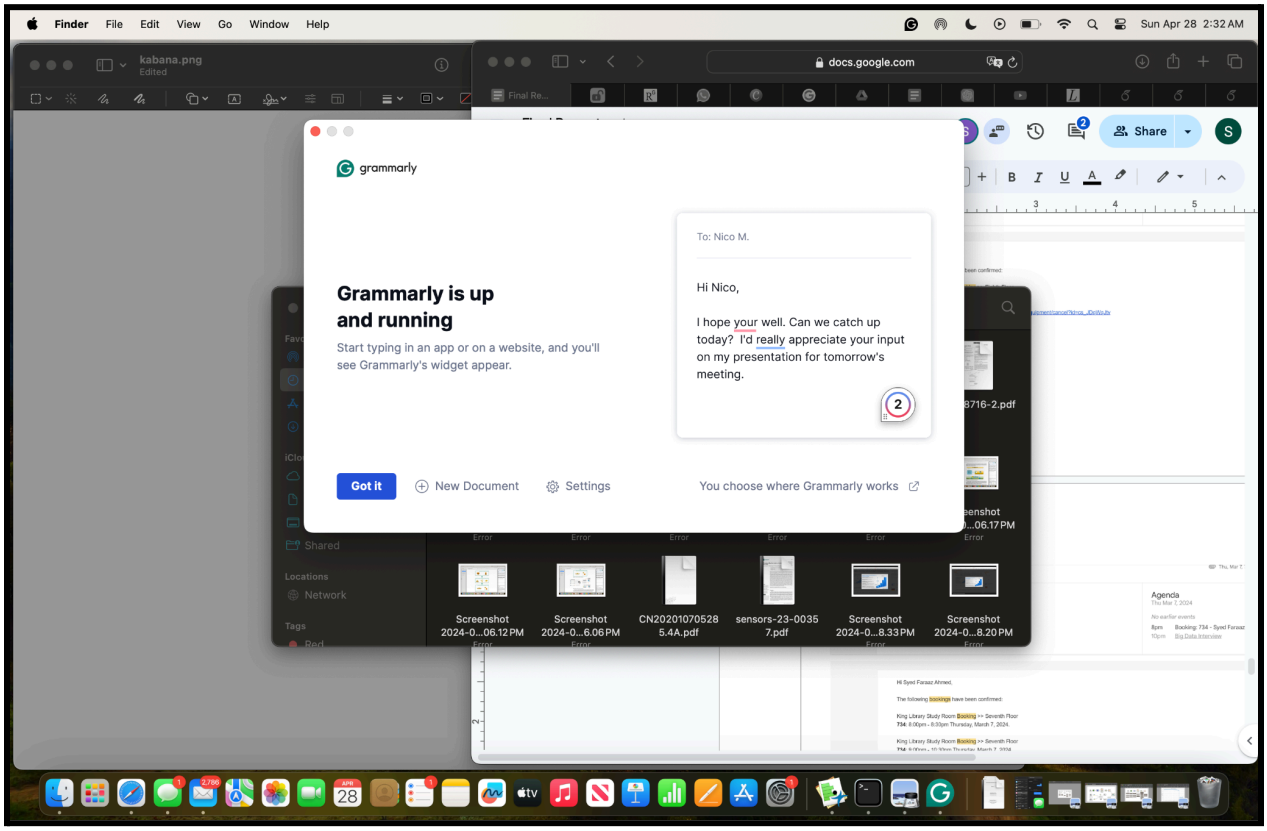
King Library Study Room Booking >> Eighth Floor
802: 7:00pm - 9:00pm Wednesday, March 13, 2024.

To cancel this booking visit: https://booking.sjlibrary.org/equipment/cancel?id=cs_JDqWqJtv

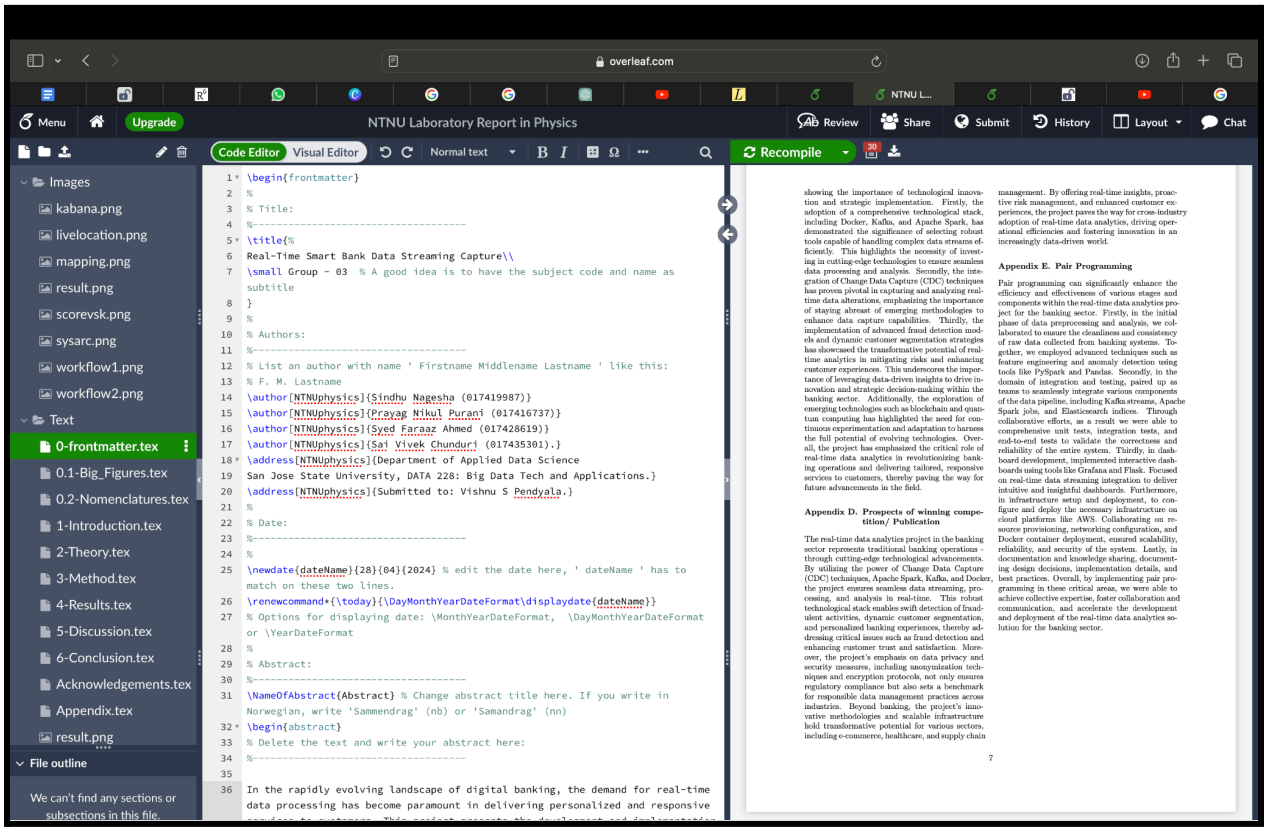
Thank you for using our Reservations System!



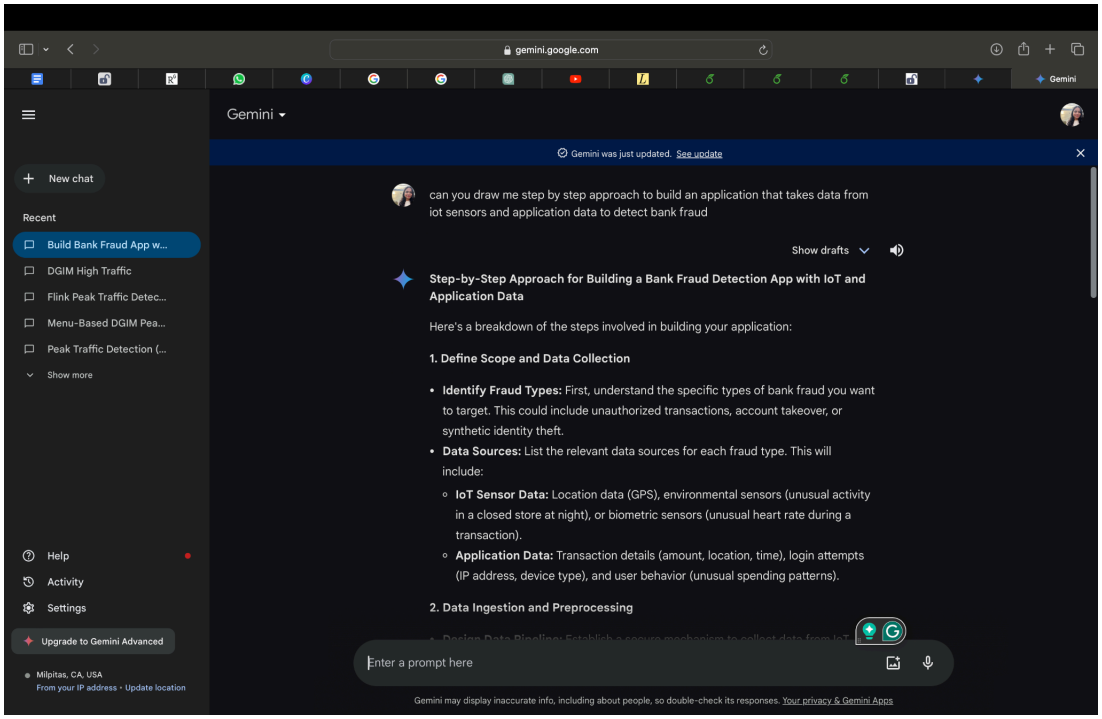
11. Grammarly: Utilization of Grammarly, helped in formatting our report and Latex report.



12. **Used Latex:** proof for using Latex



13. **Generative AI:** Involvement of Generative AI



As big data is a huge field, the chances of us getting lost on our way to the completion of projects were stuck in some of the stages where we were not very familiar with the tools. We took help from Gemini to help in installing the software and its connection.

14. Any other tools and techniques covered in the course not included in the other criteria:

1. **AWS IoT (Amazon Web Services Internet of Things)**:
    - AWS IoT is a managed cloud platform that enables devices to connect securely to the cloud and interact with other devices and AWS services.
    - It provides features such as device provisioning, authentication, and communication protocols for IoT devices.
    - AWS IoT allows you to collect, store, and analyze data from connected devices, enabling real-time monitoring, control, and automation of IoT applications.
2. **AWS Timestream**:
    - AWS Timestream is a fully managed time-series database service provided by Amazon Web Services.
    - It is optimized for handling time-series data at scale, making it well-suited for storing and analyzing data generated by IoT devices, sensors, and other time-sensitive applications.
    - Timestream provides features such as automated data retention, query optimization, and integration with other AWS services for real-time analytics and visualization.
3. Grafana:
    - Grafana is an open-source analytics and monitoring platform that allows you to visualize and analyze data from various sources, including databases, time-series databases, and IoT platforms.
    - It provides customizable dashboards, charts, and alerts for monitoring system performance, tracking key metrics, and troubleshooting issues in real time.
    - Grafana supports integration with a wide range of data sources, making it a popular choice for building monitoring and visualization solutions for IoT applications.
4. Flask:
    - Flask is a lightweight web framework for building web applications and APIs in Python.
    - It provides tools and utilities for routing HTTP requests, handling request/response cycles, and managing application logic.
    - Flask is commonly used for building backend services, RESTful APIs, and web applications that interact with IoT devices, sensors, and cloud services.
5. AWS Lambda:

- AWS Lambda is a serverless computing service provided by Amazon Web Services.
- It allows you to run code in response to events without provisioning or managing servers.
- Lambda functions can be triggered by various AWS services, including AWS IoT, AWS Timestream, API Gateway, and others, making it a flexible and scalable platform for building event-driven IoT applications.

Criteria	**Explanation**
Code Walkthrough	In Class
Presentation Skills Include time management	N/A
Discussion / Q&A	N/A
Demo	In Class and also in the video.
Report Format, completeness, language, plagiarism, whether turnItIn could process it (no unnecessary screenshots), etc	Yes
Version Control Use of Git / GitHub or equivalent; must be publicly accessible	Mentioned in the Links section.
Lessons learned Included in the report and presentation?	Yes
Prospects of winning competition/publication	Yes included in the report
Innovation	The combination of all three inputs from transaction, application, and sensor data from the end user.

Teamwork	Describe above
Technical difficulty	Describe above
Practiced pair programming?	Yes, use of Google Colab, Docs, and Canva.
Practiced agile / scrum (1-week sprints)?	Kanban, also the link is attached below
Used Grammarly / other tools for language?	Yes
Slides	N/A
Used LaTeX	Yes
Used creative presentation techniques The use of Generative AI is ok here. Try animation, effects, newer features such as those offered by Prezi, etc.	Generative AI, from Canva, to get images for PPT
Literature Survey	Yes
Use of streaming algorithms Use of Reservoir Sampling, Bloom Filter, Flajolet-Martin, DGIM, Computing Moments, algorithms for graph streams, etc	Yes (Reservoir Sampling, Bloom Filter)
Use of Stream Processing Frameworks such as Spark, Flink, and Kafka	Yes (Kafka, PySpark and more)
Use of Locality Sensitive Hashing	Yes on the transaction database
Use of Privacy techniques K-Anonymity, L-Diversity, Differential Privacy, etc.	K-Anonymity

Any other tools and techniques covered in the course not included in the other criteria Think Machine Unlearning, Explainable AI; Fairness; Federated Learning; Data Poisoning, Responsible AI, etc	Tokenization
Use of new tool(s) that were not used for any of the HW	AWS IoT. AWS Timestream, Grafana, Flask, AWS Lambda, Flask and Dash.

14. Links:

Description	Links
Agile board	https://sjsu-prayagnikulpurani.atlassian.net/jira/software/projects/HW/boards/3
Project PPT	https://www.canva.com/design/DAGCrro9bbk/7UuZduRUTGIHKNN2flfm5w/edit?utm_content=DAGCrro9bbk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
Significant Slides	https://www.canva.com/design/DAGCFBwy_gQ/9blF5w3tJBkyRAL5zoOkcA/edit?utm_content=DAGCFBwy_gQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
Industry Case Study Slides	https://www.canva.com/design/DAGAwkEmsy4/x_t_Y6xC_zVsgYkWCwSicA/edit?utm_content=DAGAwkEmsy4&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton
GitHub Link	https://github.com/pp11-web/Big_data_project
Significant paper	https://ieeexplore.ieee.org/abstract/document/8879579
Industry case paper	https://www.mdpi.com/1424-8220/23/1/357

Video Pitch	https://www.youtube.com/channel/UCUzsnQpBHc9CFExiXgwBTlQ
Latex	https://www.overleaf.com/project/662dde8aa1e6cc9854d7845c
Grafana Dashboard	https://prayagnikulpurani.grafana.net/d/ddj0kexrpojy8a/big-data?orgId=1&from=1713288741000&to=1713288768000