Here is an update on the current status of our project

1. Data cleaning and Integration

2. Rendering Data on the UI

3. Anomaly Detection.

Project Status Update

1. Data Cleaning and Integration: (Shashank and team)

1.1. **Task**: (i) Upload Safesforce data into PostgreSQL, (ii) Google Analytics (GA4) and Salesforce (SF) data into MongoDB.

1.1.1. **PostgreSQL**: Salesforce data has been uploaded to the PostgreSQL database, which is integral for the NL2SQL Sales Assist to respond to queries. However, the latest data dump has not yet been pushed to the PostgreSQL database. GA data must also be uploaded to the PostgreSQL database so that the data in the snapshot dashboard and the Sales Assist queries answers are correlated.

1.1.2. **MongoDB**:

- A sample of Salesforce data has been successfully uploaded to MongoDB for schema validation. The schema for Google Analytics data is still being defined.
- Full integration of GA and SF data is pending due to changes in the Google Analytics schema.
- Complete automation scripts need to be written to facilitate regular data dumps into MongoDB. This will enable continuous data rendering on the UI.

Database	Current Progress	Current Issues	Next Steps

PostgreSQL	Four year Salesforce data is uploaded (with no link to GA4);	-	Latest data dump yet to be pushed to the database. GA data must be uploaded to PostgreSQL database.
MongoDB	Sample of Salesforce data uploaded for schema validation;	Schema for Google Analytics still being defined. Access to AWS Glue is not yet provided to run the jobs.	Full integration of Google Analytics and Salesforce data pending which will facilitate continuous data rendering on the UI.

2. Rendering ShopLC Data on the UI: (Dev Team)

2.1. **Task**: Ensure real-time rendering and updating of ShopLC data on the UI from various data sources, improving the accuracy and responsiveness on the UI.

2.1.1. **Snapshot Page (Sources - Google Analytics)**: Waiting for data to be pushed to MongoDB. Once the data is available, it will automatically be rendered on the UI.

2.1.2. **Sales Assist (NL2SQL)**:

a) Currently, the system can answer 13 out of 20 potential questions. Remaining 7 questions can be answered when the GA data is pushed to the PostgreSQL database.

b) We are currently awaiting the latest Salesforce data update and the integration of Google Analytics data into PostgreSQL. This integration is crucial for ensuring consistency in the data displayed in both the snapshot and the Sales Assist queries.
c) Next Steps: Ensure that the latest Salesforce and Google Analytics data are uploaded to their respective databases to enable full functionality.

Component	Current Status	Current Issues	Next Steps
Snapshot Page	Renders mock data on the Banavo snapshot UI.	Waiting for GA data to be pushed to MongoDB.	-
Sales Assist (NL2SQL)	Can answer 13 out of 20 potential questions.	Awaiting the latest Salesforce and Google Analytics data to be dumped into PostgreSQL database.	Once the GA and the SF data are linked and uploaded to their respective database, the questions related to the GA data can also be answered.

1. Anomaly Detection: (Shylaja AI Team)

1.1. **Task**: Detects sales anomalies using data from MongoDB sales collections and store results in the AnomalyDetected Collection.

1.1.1. The anomaly detection script is ready and functional. However, the model currently operates with only sample Salesforce data;

complete Salesforce data has not yet been uploaded to MongoDB.

1.1.2. Once the complete Salesforce data is uploaded to MongoDB, the anomaly detection model will be fully operational when initiated from the admin console on the Banavo UI. This setup will enable the model to start displaying relevant sales anomalies on the UI.

Task	Current Status	Current Issues	Next Steps
Anomaly Detection	Anomaly Detection script is ready and functional.	Operating with only sample Salesforce data.	Push complete Salesforce data to MongoDB to fully enable the anomaly detection model to render the anomalies on the UI. Data must be tested thoroughly on the UI to ensure the data rendered is displayed accurately and making sure the data is consistent across the UI.