Deliverables:

1. Report on Clustering Results:

- **Number of Clusters**: Clearly specify the number of clusters formed, which should be chosen based on analysis (e.g., DB Index or Elbow Method).
- **DB Index Value**: Include the calculated **Davies-Bouldin Index**, which evaluates the compactness and separation of the clusters.
- **Other Metrics**: Discuss additional clustering metrics such as Silhouette Score, Dunn Index, or within-cluster sum of squares (WCSS).

2. Code Submission:

- Provide a **Jupyter Notebook or Python script** containing:
  - Data preprocessing steps.
  - Clustering code (e.g., KMeans, DBSCAN, or Hierarchical Clustering).
  - Visualization logic.
  - Code to calculate and report the DB Index and other metrics.

Evaluation Criteria:

1. Clustering Logic and Metrics:

- Ensure the clustering algorithm is suitable for the dataset and scales properly.
- Use **profile and transaction data** to form meaningful clusters.
- Report and explain the **DB Index** and justify the choice of the number of clusters.

2. Visual Representation of Clusters:

- Use **scatter plots**, **pair plots**, or **3D plots** to visualize clusters in 2D or 3D space.
- Annotate visuals to make clusters and their distinctions easy to interpret.
- Optionally, include heatmaps or cluster centers for added clarity.

Clustering Results Report

1. Number of Clusters Formed

- Based on the analysis (using Elbow Method, DB Index, or Silhouette Scores), we determined the optimal number of clusters is **[number]**.
- Each cluster represents a distinct customer group with shared characteristics based on transaction and profile data.

2. DB Index Value

- The calculated **Davies-Bouldin Index (DB Index)** for the clustering is **[value]**.
    - Lower values indicate better clustering (compact and well-separated clusters).

3. Other Relevant Metrics

- **Silhouette Score**: [value] (indicates how well clusters are separated).
- **Within-Cluster Sum of Squares (WCSS)**: [value].
- Explain the significance of these metrics in validating the clustering performance.

4. Clustering Algorithm

- Algorithm used: [e.g., KMeans, DBSCAN, or Hierarchical Clustering].
- Preprocessing included [e.g., normalization, handling missing values, feature selection].

5. Insights from Clustering

- **Cluster Descriptions**: Describe each cluster based on common traits (e.g., high-value customers, frequent shoppers, regional segmentation).
- **Business Implications**: Highlight how the clusters can help in marketing, sales strategies, or personalized recommendations.

Visual Representations

- **Scatter Plot**: Show clusters in a 2D/3D feature space with clear boundaries.
- **Pair Plot**: Visualize relationships between features within clusters.
- **Cluster Centroids**: Illustrate where the cluster centers are located relative to the data.
- Include appropriate legends, labels, and titles for clarity.

Conclusion

Summarize key findings:

- "The clustering analysis revealed [number] distinct customer groups with meaningful separations based on [features]."
- Highlight potential use cases like customer targeting, retention strategies, or optimizing product recommendations.