Development Document: Customer Journey Analysis Using Clustering and Dimensionality Reduction

# 1. Development Environment

The development was carried out using Google Colab with Python. Google Colab offers a cloud-based Jupyter notebook interface with pre-installed libraries, making it convenient for machine learning development.

# 2. Step-by-Step Development Process

Step 1: Import Required Libraries  
```python  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.preprocessing import StandardScaler  
from sklearn.decomposition import PCA  
from sklearn.manifold import TSNE  
from sklearn.cluster import KMeans  
```  
  
Step 2: Load and Explore the Dataset  
```python  
df = pd.read\_csv('customer\_journey.csv')  
print(df.head())  
```  
  
Step 3: Data Preprocessing  
```python  
# Handle missing values  
df = df.dropna()  
# Normalize the data  
scaler = StandardScaler()  
scaled\_data = scaler.fit\_transform(df)  
```  
  
Step 4: Dimensionality Reduction  
```python  
pca = PCA(n\_components=2)  
pca\_result = pca.fit\_transform(scaled\_data)  
```  
  
Step 5: Clustering  
```python  
kmeans = KMeans(n\_clusters=3)  
clusters = kmeans.fit\_predict(pca\_result)  
```  
  
Step 6: Visualization  
```python  
plt.figure(figsize=(8, 6))  
sns.scatterplot(x=pca\_result[:, 0], y=pca\_result[:, 1], hue=clusters, palette='viridis')  
plt.title('Customer Clusters')  
plt.show()  
```

# 3. Testing and Validation

The clustering results were validated using silhouette scores and visual inspection of cluster separation. Further analysis was performed to profile each cluster and interpret customer behavior patterns.

# 4. Challenges Faced

- Handling high-dimensional data required careful selection of dimensionality reduction techniques.  
- Choosing the optimal number of clusters involved iterative experimentation.  
- Ensuring interpretability of clusters to derive meaningful insights.