Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	13 November 2023
Team ID	Team - 592211
Project Name	Understanding Audience: A Machine Learning Approach To Customer Segmentation
Maximum Marks	4 Marks

Technical Architecture:

The technical architecture comprises a modular design integrating data collection, RFM analysis, and machine learning components. Utilizing scalable frameworks, it ensures dynamic adaptability to evolving customer behaviors for optimal marketing precision.

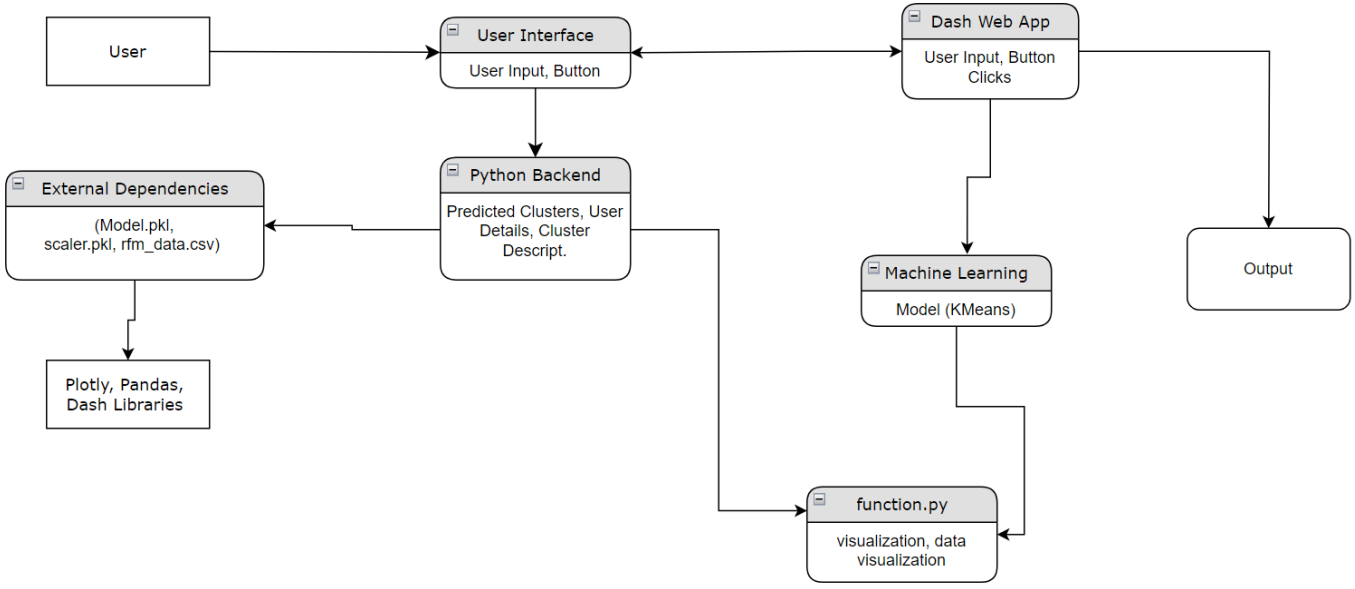

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Collection	Gather customer data from diverse sources, ensuring accuracy and data quality.	Python, Pandas, NumPy, Data APIs
2.	RFM Analysis	Implements Recency, Frequency, and Monetary analysis for customer segmentation.	Python, Scikit-Learn
3.	Machine Learning Model	Develops a predictive model for dynamic customer segmentation.	Python, Scikit-Learn, Pickle
4.	Solution Architecture	Designs a modular and scalable architecture integrating data and analysis components.	Docker
5.	Web Application (Dash)	Transforms the project into an interactive web app for user-friendly access.	Dash by Plotly, HTML, CSS, JavaScript, Dash Bootstrap Components
6.	Data Visualization	Utilizes visualizations to interpret and present insights effectively.	Plotly, Matplotlib, Seaborn
7.	Hosting	Renders the web application for online access	Render

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source frameworks to enhance flexibility and community support.	Dash by Plotly, Scikit-Learn, Matplotlib, Seaborn, NumPy, Pandas
2.	Security Implementations	Implements security measures to protect customer data and ensure data privacy.	SSL for Dash web app, Secure data handling practices
3.	Scalable Architecture	Features a modular and scalable architecture to accommodate varying workloads.	Docker, Design patterns for scalability
4.	Availability	Ensures high availability to provide consistent access to the web application.	Load balancing, Redundancy in hosting infrastructure
5.	Performance	Optimizes performance for efficient data processing and a responsive user experience.	Code optimization, Efficient algorithms, Monitoring for performance improvement

References:

Dash Documentation: https://dash.plotly.com/

Scikit-Learn Documentation: https://scikit-learn.org/stable/documentation.html

Matplotlib Documentation: https://matplotlib.org/stable/contents.htm

Seaborn Documentation: https://seaborn.pydata.org/documentation.html

NumPy Documentation: https://numpy.org/doc/stable/

Pandas Documentation: https://pandas.pydata.org/pandas-docs/stable/