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

Date	16 October 2022
Team ID	PNT2022TMID36032
Project Name	Personal Expense Tracker Application
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

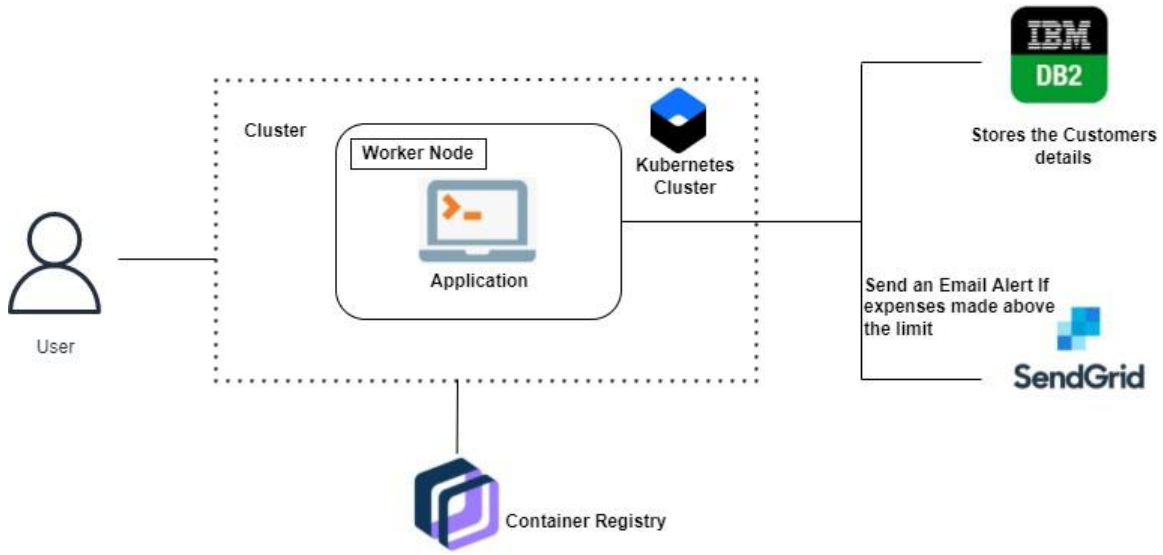

Table 1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user can interact with the application with the use of a Chatbot.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	The application contains the sign-in/sign-up where the user will log in to the main dashboard.	Java / Python/ Node , ExpressJS
3.	Application Logic-2	The dashboard contains the fields like Add income, Complete Tasks, and Track Expenses	IBM Watson STT service, Java/Python
4.	Application Logic-3	The user will get to perform transactions or save the money based on their choices to get rewards	IBM Watson Assistant, Java, JavaScript
4.	Application Logic-3	The user will get the expense report in graph form and also get alerts if the expense limit exceeds.	IBM Watson Assistant, Java/ Python
5.	Database	The Income and Expense data are permanently stored in the database.	MySQL, NoSQL, etc.

6.	Cloud Database	With the use of Database Service on the Cloud, the User data are stored in a well-secured Manner with various features for scalability	IBM DB2, IBM Cloud Services etc.
7.	File Storage	IBM Block Storage is used to store the Financial data of the user in proper structure.	IBM Block Storage or IBM Object Storage or Other Storage Service or Local Filesystem
8.	External API-1	It exposes a business's internal resources to outside users or applications.	IBM Weather API, REST API etc.

Table 2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask Framework in Python is used to implement this Application.	Python-Flask, BootStrap, React
2.	Security Implementations	This Application Provides high security to the user's financial data and expenses. It can be done by using the Container Registry in the IBM cloud.	Container Registry, Kubernetes , Docker, Cluster, IBM DB2
3.	Scalable Architecture	Meet up the high demands of the user , as the number of users increases the application should be able to meet the requirements.	Kubernetes Cluster, Docker, Cloud Services
4.	Availability	This application will be available to the user all the time. The services will be open 24/7.Kubernetes services, the crudest form of load balancing traffic.	Docker, Kubernetes Cluster
5.	Performance	Can handle a large number of requests per second.	Kubernetes Cluster, Docker, Container Registry