

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

Date	21 October 2022
Team ID	PNT2022TMID07222
Project Name	Personal Expense Tracker Application
Maximum Marks	4 Marks

Technical Architecture:

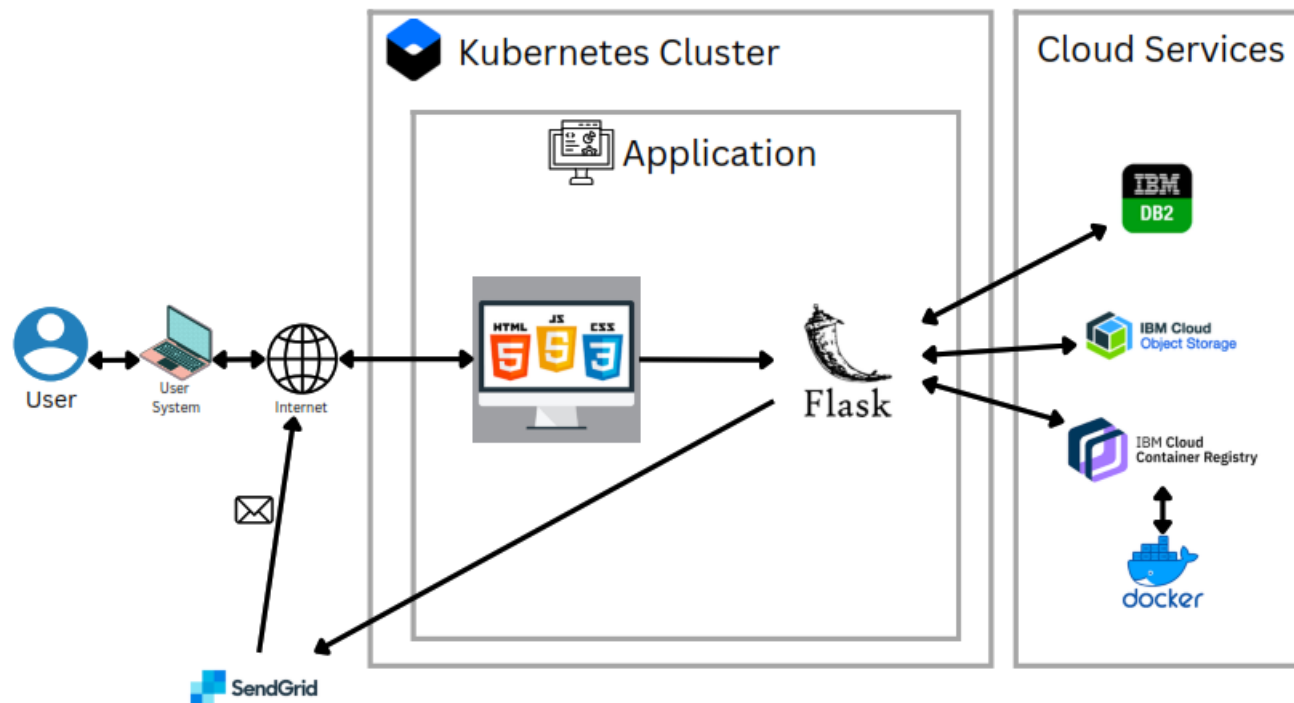


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user can interact with the application with the use of a Chatbot.	HTML, CSS, JavaScript / React Js.
2.	Application Logic-1	The application contains the sign in/sign up where the user will login into the main dashboard.	Flask
3.	Application Logic-2	Dashboard contains the fields like Add income, Add Expenses, Genrate Report, Grpahs.	IBM Watson STT service
4.	Application Logic-3	The user will get the expense report in the graph form and also get alerts if the expense limit exceed	IBM Watson Assistant, SendGrid
5.	Cloud Database	The Income and Expense data are stored in the IBM DB2 database. With use of Database Service on Cloud, the User data are stored in a well secured Manner.	IBM DB2
6.	File Storage	IBM Block Storage used to store the financial data of the user.	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	To alert users when the limit of expense reaches or over used and recurring payment criteria.	SendGrid
8.	Infrastructure (Server / Cloud)	Application Deployment on Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Using flask to implement backend and connect with external services and database	Flask – Python, Kubernetes
2.	Security Implementations	This application provides high security to the user financial data. It can be done by using the container registry in IBM cloud	Container Registry, Docker, Kubernetes Cluster

S.No	Characteristics	Description	Technology
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. Using Microservices Architecture to provide scalable application	Container Registry, Docker, Kubernetes Cluster
4.	Availability	This application will be available to the user at any part of time. Deploying the application with Kubernetes cluster to make application available across the globe on the internet.	Container Registry, Docker, Kubernetes Cluster
5.	Performance	Can handle a large number of requests per second. The performance will be high because there will be no network traffics in the application	Docker, Kubernetes Cluster