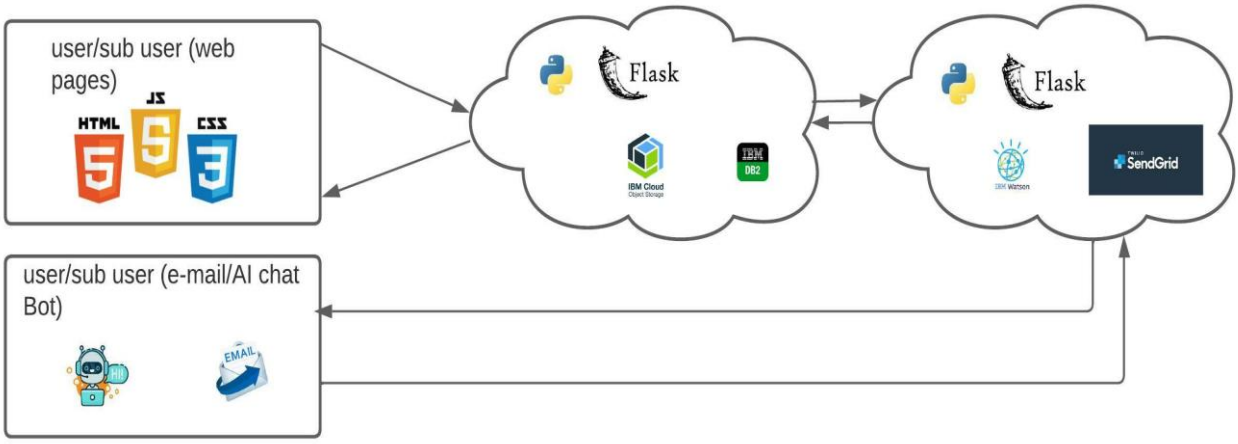
PROJECT DESIGN PHASE-II

TECHNOLOGY STACK (Architecture & Stack)

DATE	18 October 2022
TEAM ID	PNT2022TMID07720
PROJECT NAME	INVENTORY MANAGEMENT SYSTEM FOR RETAILORS
MAXIMUM MARKS	4 MARKS

TECHNICAL ARCHITECTURE:



S.No	Component	Description	Technology
1.	User Interface	User can make use of a web browser for using our web application the entire process is going to be on a web browser.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Login/registration page in this page user can create an account for their inventory	Python /flask
3.	Application Logic-2	In this page user can able to view the main dash board, Here user can able to maintain their entire process	Python/flask
4.	Application Logic-3	In this page user can able to give constrain access to other sub users	Python/flask
5.	Database	User login credentials and sub user login credentials, user policy and about their stock details	MySQL, NoSQL, etc.

6.	Cloud Database	Cloud data base will be used for storing the details about the user as well as their stock details.	IBM DB2, IBM Cloudant.
7.	File Storage	It will be used to store the product image, profile picture of the users and the sub users	IBM cloud object storage
8.	External API-1	It will be used to insert data into IBM DB using python. It also used to insert file to IBM cloud object storage using python. It is used to send email to the user using python.	IBM_DB, IBM_COS_SDK Sendgrid
9.	External API-2	It is used to integrate IBM WATSON AI CHAT BOT with our project	IBM WATSON API
10.	Infrastructure (Server / Cloud)	Containers are used to achieve microservice architecture and kubernetes is a container orchestration.	Container registry, docker, Kubernetes.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask is micro web frame work it contains all business login .It has a development time server by default and it also use a template engine called jinja.	flask
2.	Security Implementations	Two step verification using email by sending an one time password to the user. while login to their account.	sendgrid
3.	Scalable Architecture	In microservice architecture each and every module is loosely coupled so it is easy to maintain it. One module will not affect other module.	Containerization, orchestration.
4.	Availability	In this project failure of one module will not affect other so the availability can be maintained.	Containerization
5.	Performance	we are going to develop this project using microservice architecture to enhance the user performance.	Application Modernization