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

Date	15 October 2022
Team ID	PNT2022TMID48338
Project Name	Smart Fashion Recommender 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

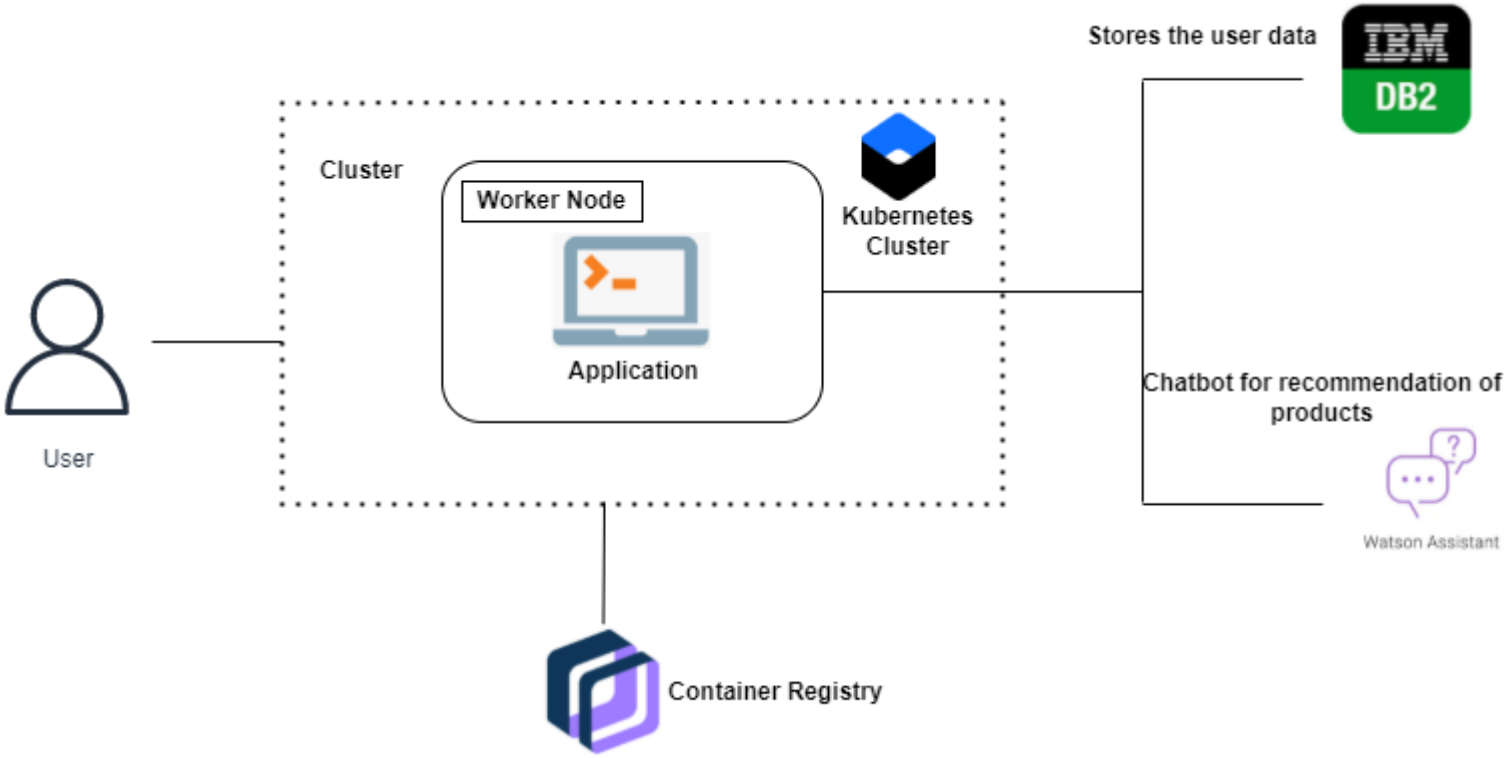

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with application through Web UI	HTML, CSS, Bootstrap, JavaScript, React Js.
2.	User login to the e-commerce website	User will login to the e-commerce website to search and order the required products.	Python, Flask, HTML, CSS, JavaScript, IBM DB2, SendGrid.
3.	Communicates with Chatbot	The user communicates with the chatbot in either textual form or by speaking directly to the chatbot.	IBM Watson STT service
4.	Chatbot recommends products	Chatbot processes the user's input and recommends products based on their needs and interests.	IBM Watson Assistant
5.	Database	Stores the user's name, e-mail, address and his interests information in the database. Data types: Integer, String, Float, Varchar, etc.,	MySQL, SQLite, POSTgre SQL.
6.	Cloud Database	Once the application is deployed in cloud, The information will be stored at cloud	IBM DB2, IBM Cloudant.
7.	File Storage	Necessary files for the application should be maintained at easily accessible place	IBM Object Storage, IBM Block Storage or Local Filesystem
8.	External API-1	To send emails to the user	SendGrid
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration	Local, Cloud Foundry, Kubernetes.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask, Bootstrap(CSS framework), React Js.
2.	Security Implementations	Each user is provided with unique ID & password. Assures all the data inside the system will be protected against malware attacks or unauthorized access	Encryptions, IAM Controls, SendGrid.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	By using Docker and Kubernetes we make our application scalable without making major changes to it
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Usage of Cloud environment like object storage and DB2 and clustering using Docker & Kubernetes makes the available all time
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	IBM Cloud, Kubernetes cluster, Container Registry.