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

Date	03 October 2022
Team ID	PNT2022TMID47422
Project Name	Nutrition Assistant 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.

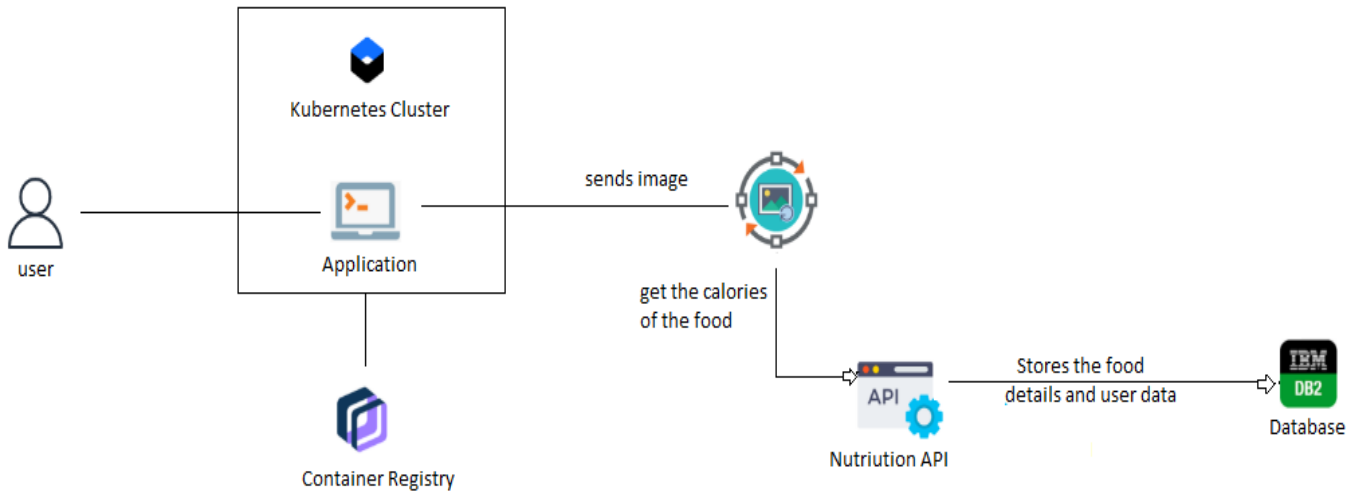

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
•	User Interface	Web UI	HTML, CSS, JavaScript .
•	To get the food nutrition and calorie value	The user will upload the food picture,Then the user will see the food nutrition value and the process will compute.	Python , Flask (web Framework), HTML, CSS , JavaScript.
•	Database	Get the user's name , mail	MySQL or PostgreSQL.

		and stores the food calories value. Data types:integer,string,Float Number and etc..	
•	Cloud Deployment	Through is the application will compose to the internet.	Kubernetes,Docker.
•	External API-1	To predict the image that user will upload in the upload image page.	Clarifai's AI- driven Food detection Model API.
•	External API-2	Food API's for the nutritional value of the identified food.	Food API.
•	Infrastructure(Server / Cloud)	Application Deployment on Local System / Cloud. local and cloud server configurations .	Local , Cloud Foundry , Kubernetes , etc. Docker.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
•	Open-Source Frameworks	We are using both front and back end here to runs the web application.	Flask(Microweb framework) Vue.js
•	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
•	Scalable Architecture	Justify the scalability of architecture (3 - tier , Micro - services)	Presentation tier - HTML / CSS / JavaScript. Application tier-Python(API) Data tier - MySQL , PostgreSQL
•	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Working to reduce the severity and likelihood of problems , closely monitoring applications and infrastructure , keeping technical debt in check , automating recovering mechanisms , and regularly putting those recovery mechanisms to

			the test.
•	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Optimize image sizes , use a content delivery network , use website caching and adopt cloud-based website monitoring.