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

Date	03 October 2022
Team ID	PNT2022TMID00787
Project Name	Nutrition assistant Application
Maximum Marks	4 Marks

Technical Architecture:

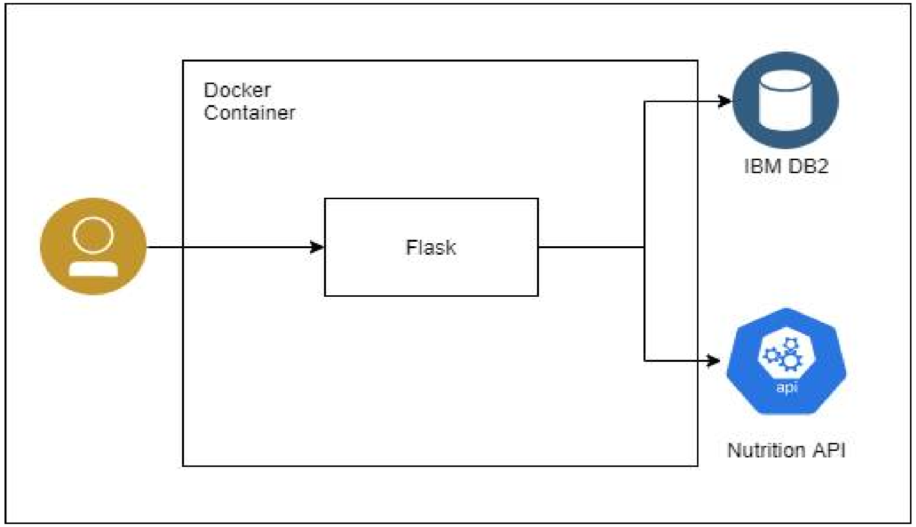

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI	HTML, CSS, JavaScript.
2.	To determine the calorie count and nutrients of food	The uploader will include the food image. The user will then be shown the food's nutritional value.	Python, Flask (web Framework), HTML, CSS, JavaScript
3.	Database	obtains and stores the user's name and email. stores the number of calories in food. Data types: Float Number, String, and Integer	MySQL or PostgreSQL
4.	. Cloud Deployment	This application will allow you to post to the Internet.	Kubernetes, Docker
5.	External API-1	to anticipate the user-uploaded image on the upload image page	Clarifai's AI-driven Food detection Model API
6.	External API-2	Food APIs for the specified food's nutritional value	Food API
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Running the web application using both the front end and the back end	Flask (Microweb framework) Vue.js
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Presentation tier- HTML/ CSS/ JavaScript Application tier- Python (API) Data tier- MySQL, PostgreSQL
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	By attempting to lessen the degree and likelihood of issues, closely monitoring applications and infrastructure, limiting technical debt, automating recovery methods, and routinely testing those recovery mechanisms, as well as by working to reduce the occurrence and severity of issues
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	Adapt cloud-based website monitoring, employ a content delivery network, leverage website caching, and optimise image sizes.