# Technical Architecture:

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

|  |  |
| --- | --- |
| Date | 15 October 2022 |
| Team ID | PNT2022TMID40681 |
| Project Name | Project – Nutrition Assistant Application |
| Maximum Marks | 4 Marks |

USER IBM CLOUD



Register / Login 



Upload meal Image

Display nutritional values

EXTERNAL API



# Guidelines:

1. To use the app the user must register / login.
2. After successful registration/login, the user can upload the meal image.
3. Using Clarify AI- Driven API the name of the meal will be identified.
4. The identified name will be sent to Nutrition API using Flask.
5. Using Nutrition API, the nutritional value of the meal will be obtained and displayed in the UI using Flask.
6. The diet history will be added to the database to track their daily calorie intake.



# Table-1 : Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | User interacts with application Web UI | HTML, CSS, JavaScript |
| 2. | Application Logic-1 | Connection with Database and external API’s | Python Flask |
| 3. | Application Logic-2 | Integration of chatbot with application | IBM Watson Assistant |
| 4. | Database | Data Type, Configurations etc. | MySQL |
| 5. | Cloud Database | Database Service on Cloud – used to store user details for registration and login, and track diet history | IBM DB2 |
| 6. | External API-1 | This API is used to find the name of the food, for which the image has been uploaded | Clarify AI-Driven API |
| 7. | External API-2 | This API is used to find the recipe and Nutritional value present inside the food | Nutrition API ( Rapid API) |
| 8. | Infrastructure | Application Deployment to provide good performance and scalability | Kubernetes |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Flask is used for connecting database and external API’s. | Python flask |
| 2. | Security Implementations | Security is provided for accessing the database. | SSH |
| 3. | Scalable Architecture | Presentation tier: User Interface to login and upload meal image  Application tier: Nutrition API, Clarifai API Database tier: IBM cloud DB2 | HTML, CSS, JavaScript, Flask, Kubernetes, IBM DB2 |
| 4. | Availability | Clustering improves availability. This can be achieved with the help of Kubernetes cluster. | Kubernetes |
| 5. | Performance | By using cache and adding master nodes we can improve performance of the application | Kubernetes |