Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	8 November 2023
Team ID	Team- 592065
Project Name	Vitamin Detection using Deep learning
Maximum Marks	4 Marks

Technical Architecture:

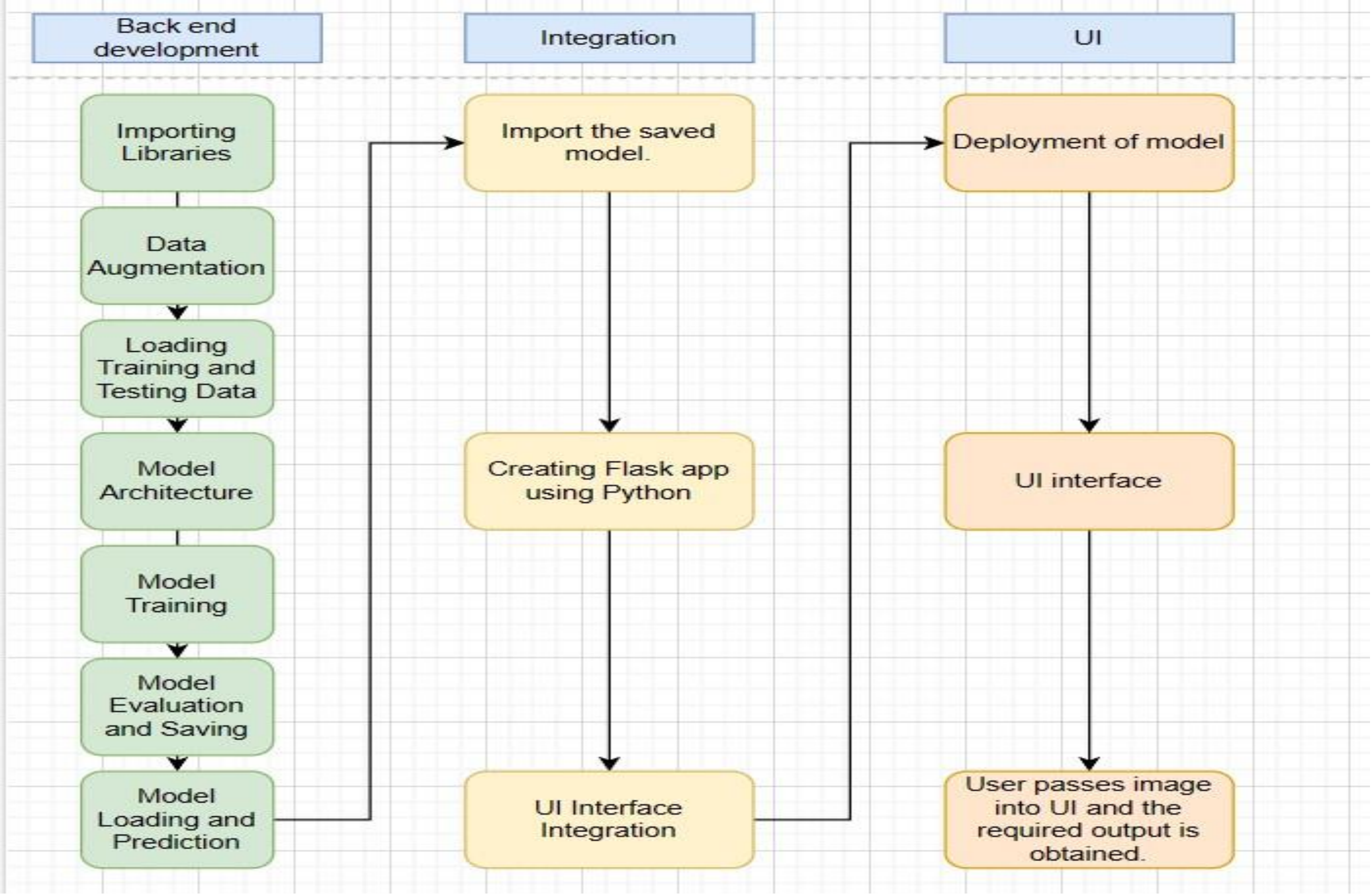
Back end development
Integration
UI
Importing Libraries
Data Augmentation
Loading Training and Testing Data
Model Architecture
Model Training
Model Evaluation and Saving
Model Loading and Prediction
Import the saved model.
Creating Flask app using Python
UI Interface Integration
Deployment of model
UI interface
User passes image into UI and the required output is obtained.

Table-1: Components & Technologies:

S No:	Component	Description	Technology
1.	User Interface	How user interacts with application Example: Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript.
2.	Application Logic-1 (Data collection and Image Preprocessing)	To implement the model by utilizing the dataset provided.	Python
3.	Application Logic-2	To build the model and test the model to get the predictions required.	Tensor flow, Keras
4.	Application Logic-3 (UI Interface and deployment)	To implement the web application and deploy resulting in the required output.	Flask
5.	Database	Collecting the required images for the project. We have collected images of fruits, vegetables, and various food items.	Local system, Kaggle, Google.
6.	File Storage	File storage requirements	Local File system, Google Drive.
7.	Frame work	Used to integrate the web application.	Python Flask
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local system.

Table-2: Application Characteristics:

S No:	Component	Description	Technology
1.	Image Classification	The primary task of the application is to classify images containing food items or supplements into different categories based on the presence of specific vitamins.	Python
2.	Deep Learning Model	Utilizes deep learning models, such as convolutional neural networks (CNNs), for feature extraction and learning hierarchical representations from input images.	Python
3.	Training data	It has diverse dataset of images representing various food items or products containing different vitamins. The model learns from this dataset during the training phase.	Python
4.	Vitamin specific cases	The model is trained to recognize specific classes related to different vitamins. For example, classes may include "Vitamin A," "Vitamin B," "Vitamin C," and so on.	Python
5.	Preprocessing Techniques	Incorporates image preprocessing techniques, such as resizing, normalization, and augmentation, to enhance the model's ability to generalize and improve performance.	Python
6.	UI interface	May include a user-friendly web interface for users to interact with the application. Users can upload images, and the application provides predictions.	HTML, CSS, Javascript
7.	Evaluation metrics	Evaluates model performance using accuracy metrics and potentially other metrics relevant to the application.	Python