# **Working of Food Recommendation project.**

In this documentation, I’m going to explain about how my food recommendation web application works and pre-installations required to run it successfully.

So, before running the project make sure to install the below applications or files:

**Installations:**

1. **Docker application:**

**Installing Docker on Windows**

**Requirements:**

Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later)

Hyper-V and Containers Windows features enabled

**Steps:**

* Go to the Docker Hub and download Docker Desktop for Windows.
* Run the installer and follow the prompts to install Docker Desktop.
* Start Docker Desktop.

Once installed, Docker Desktop should start automatically. You'll see the Docker icon in the system tray once it's running.

**Installing Docker on macOS**

**Requirements:**

macOS 10.13 or newer

**Steps:**

* Visit the Docker Hub and download Docker Desktop for Mac.
* Open the downloaded .dmg file and drag the Docker icon to your Applications folder.
* Start Docker Desktop.

1. **Python:**Install python from the python website according to the system OS and CPU bits and after installing run the exe file.

**Flow of Project**

**Frontend:**

First the user registers his details in "register.html" file and then he gets logged in using "login.html" and it validates the user and allow him into main "index.html" file where he enters the details regarding the nutritious values and gets the recommendations.

**Database:**

Here the tables are configured and integrated with the MySQL server, with the help of NodeJS. And the user details are stored and validated and in further it also stores the names of the dishes which the user want to save under his email in a table format and also fetches the details from the database by using MySQL queries..

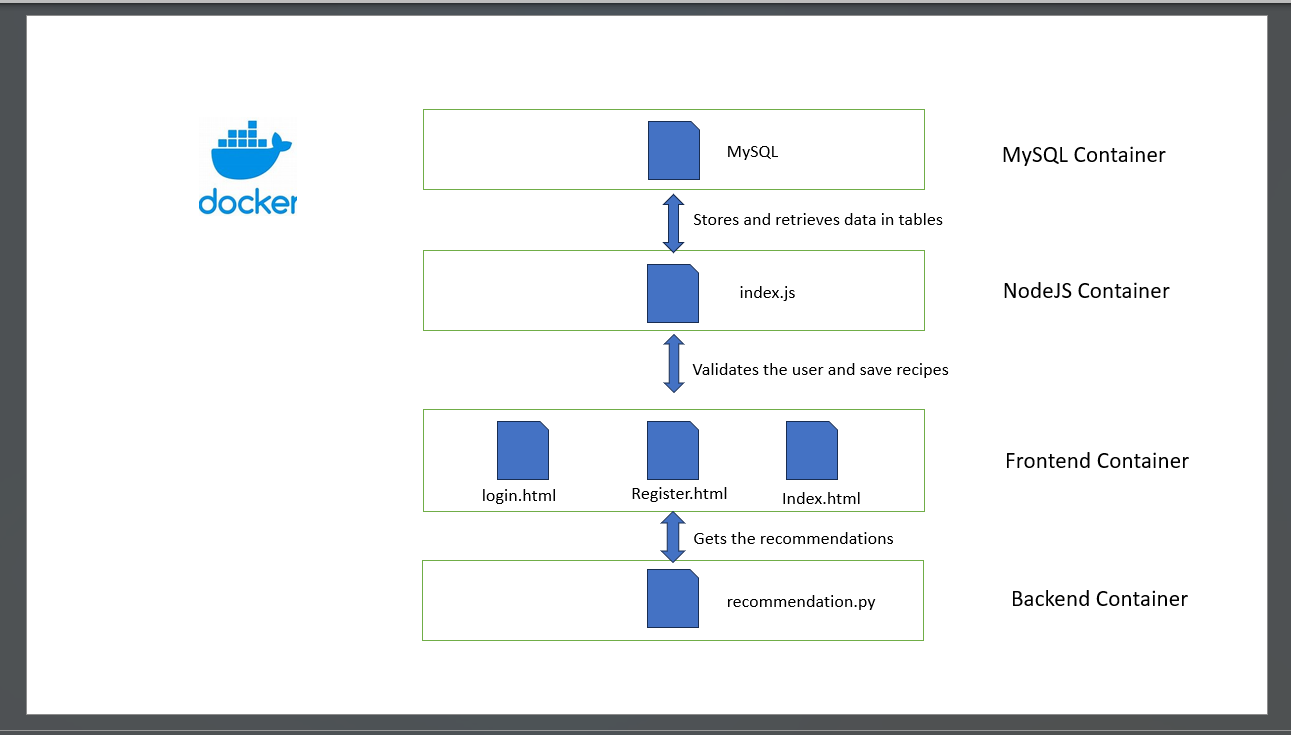
**MySQL:**

In the docker-compose file we pulled an official image of MySQL from the docker hub, and created a database by configuring it with the help of node js, which is present in the database folder. And in this we are going to create a “user” table and “recipes'' table and store the details.

**Backend:**

Here the recommendation is done, that is where all logic works which includes training of the model and flask application to recommend dishes and returns the dishes according to the trained model to the frontend..

**Flowchart**

****

**Running the project in your local machine :**

**Clone the project**

* git clone <https://github.com/civilian-sai/food_recommendation.git>
* Open the food\_recommendation folder.

// cd food\_recommendation

* **Steps to do before running by using docker**

//enter into backend

cd backend

//save the recipes.csv file in this folder by using the above link or download from the google drive provided.

Google drive link: <https://drive.google.com/drive/folders/1y_9G-yE9NlRMDxRsfYg0d-0gQCxEBREC?usp=drive_link>  
  
Kaggle link:  
<https://www.kaggle.com/code/tanishqdublish/diet-recommendation-system-preprocessing/input>

**The below step is just optional to check whether the database is integrated well and for your further**

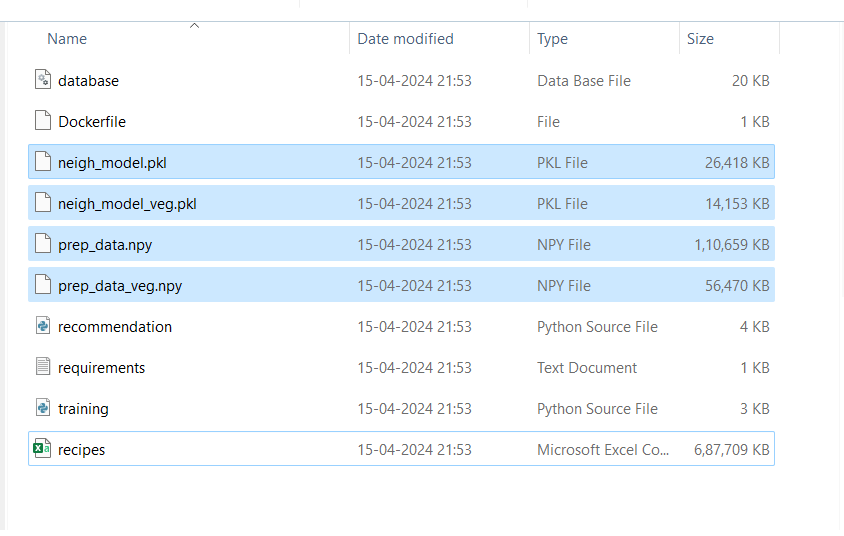
**understanding of the flow of the program.**

//now **run training.py** to get the .pkl, .npy files which stores the training instances objects and trained

data.

* **python training.py**

**You can see the following highlighted files will get included into your backend directory**

****

**Now, with the below steps you can run the project to see the results.**

* Come to the directory where the docker-compose file is present.

**To run with docker (make sure that you are running docker desktop in background)**

* docker-compose up --build

// use the below URL in the browser to get started.

<http://localhost:5500/register.html>

**Problems that may arise**:

**1. Dataset(recipes.csv) not found**:

This error arises because in my git repository, I was not able to upload my dataset because github doesn't allow oversized data. So, I request you guys to kindly download it from the google drive link or kaggle link which was provided in this document or in my readme file. And make sure that this **dataset is saved in the backend folder**.

**The below step is just optional to check the integrity of the dataset and also for further understanding.**

And after downloading it **run the training.py** in the terminal by using the following command to get some files which are needed for further process:

//python training.py (this will generates some .pkl and .npy files)

After this you can run docker-compose up in the directory where the docker-compose file is present.  
  
  
**2. Registration Failed**:

This error arises because of the loading mechanism of containers. Because in my project first the MySQL container should be created and then node application container should be created, and only then it can make connections with the MySQL database but this happens in reverse, where first node application container is created and then MySQL container. (this happens because in docker the containers are created parallelly and low size containers will get created first and node application is low in size). So to rectify this error all you need to do is, in docker desktop app, you need to restart the node application container.

**More details.**  
  
To check the tables in MySQL server use the “**MySQL workbench**” app, where after opening it with the help of our password which is “password” you can access the tables in the database section.