# Tech Saksham

## Final Project Report

# Track Name

## Building your Diet using Artificial Intelligence, with Python

## MIT college of arts and science for women, Musiri

| ROLL NO | NAME |
|---|---|
| CB20S202953 | Harshini. K |
| CB20S202967 | Rithika. S |
| CB20S202969 | Saranya. M |
| CB20S202974 | Soundaraya. S |

| | |
|---|---|
| Mayank Shrivastava | Trainer Name |
| Mayank Shrivastava | Master Trainer |

# INDEX



# CHAPTER 1

# INTRODUCTION

- **Overview**
- **Feature**
- **Advantages**
- **Scope**

**1.5Future Work**

# 1.1 Overview

**Just similar to a human dietitian, this AI Diet Consultant is based on android operating system which will act like your device dietitian. When you go to a doctor of nutrition, than she will ask you your personal details related to body and health such as your age, your height, your weight etc.**

# 1.2 Feature

**Just similar to this doctor, this artificial intelligent diet consultant also asks you similar questions in your device and you have to answer all those questions and then this AI Diet Consultant will also advice you about what should your intake in your diet and what should you ignore in order to keep yourself healthy via your**

**diet.**

**Generally, you have to hire a dietitian in order to get advice. Hiring a nutrition doctor will not only waste your time and efforts for calling them, going to them and so on but also cost you very high as their charges per month are very high. A situation might also arise when they will not available for you and you have to search for some other dietitian urgently.**

## 1.2 Advantage

**The main advantage of using this standalone AI Diet Consultant application is that the time required by the people to travel to the dietitian will be reduced and also it reduces the cost of hiring dietitians for some particular purpose.**

## 1.4 Scope

**Dietitians can utilize this system to ensure what they suggest patients.This system can be very much utilized in clinical universities for educating and rehearsing purposes so understudy can gain from it.**

**This system can likewise be used in rec center**

especially for working out the clients' calories and diet plans.Individual can likewise utilize this product particularly for themselves in home.

## 1.5 Future Work

In the Literature Survey part, we noticed the principal objective of our undertaking which was to be made and begun looking for distributed papers on it which will help us in building the application. We went across numerous IEEE and Bayes Papers and found many papers which was some or the alternate way associated with our task in view of wellbeing. We found many fascinating papers as well as straightforward ones, we accumulated the information from them. In the current medical services framework, the essential necessity and hindrance is actual presence of individual and dietician for each interview. In the current eating regimen advisor framework, you need to employ a dietitian to get guidance. Additionally, there is a high opportunity of confusion of information as well as event of

**mistakes. In addition, it is tedious.**

**With the expansion in volume of patients in the medical care establishments, customary strategy for the board has left stage. Subsequently, a high level Diet Consultant Management System has been the interest of time. A few Systems were constructed straightforwardly for sole motivation behind calories admission and some were Activity reason applications, a few ventures site based and some were versatile application based.**

**Our task was to be based on android so that individuals can get a decent UI and furthermore the application ought to be easy to understand. A portion of the applications were paid-to-utilize and some were free, we needed to assemble our task to be free to all. We began gathering data on the current framework and how it functions and**

**furthermore a genuine dietitian works and computes an eating regimen in view of an individual's subtleties like level, age, weight, orientation and so forth.**

**Indeed, even the web helped us a ton for discovering a few fundamental recipes for computing the eating regimen and absolute calories. An individual's eating routine thoroughly relies on his BMI and BMR values.**

# CHAPTER 2

# SERVICES AND TOOLS REQUIRED

**In the Literature Survey part, we noticed the principal objective of our undertaking which was to be made and begun looking for distributed papers on it which will help us in building the application. We went across numerous IEEE and Bayes Papers and found many papers which was some or the alternate way associated with our task in view of wellbeing. We found many fascinating papers as well as straightforward ones, we accumulated the information from them. In the current medical services framework, the essential necessity and hindrance is actual presence of individual and dietician for each interview. In the current eating regimen advisor framework, you need to employ a dietitian to get guidance. Additionally, there is a high opportunity of confusion of information as well as event of mistakes. In addition, it is tedious.**

**With the expansion in volume of patients in the medical care establishments, customary strategy for the board has left stage. Subsequently, a high level Diet Consultant Management System has been the interest of time. A few Systems were constructed straightforwardly for sole motivation behind calories admission and some were Activity reason applications, a few ventures site based and some were versatile application based.**

**Our task was to be based on android so that individuals can get a decent UI and furthermore the application ought to be easy to understand. A portion of the applications were paid-to-utilize and some were free, we needed to assemble our task to be free to all. We began gathering data on the current framework and how it functions and furthermore a genuine dietitian works and computes an eating regimen in view of an individual's subtleties like level, age, weight, orientation and so forth.**

**Indeed, even the web helped us a ton for discovering a few fundamental recipes for computing the eating regimen and absolute calories. An individual's eating routine thoroughly relies on his BMI and BMR values.**

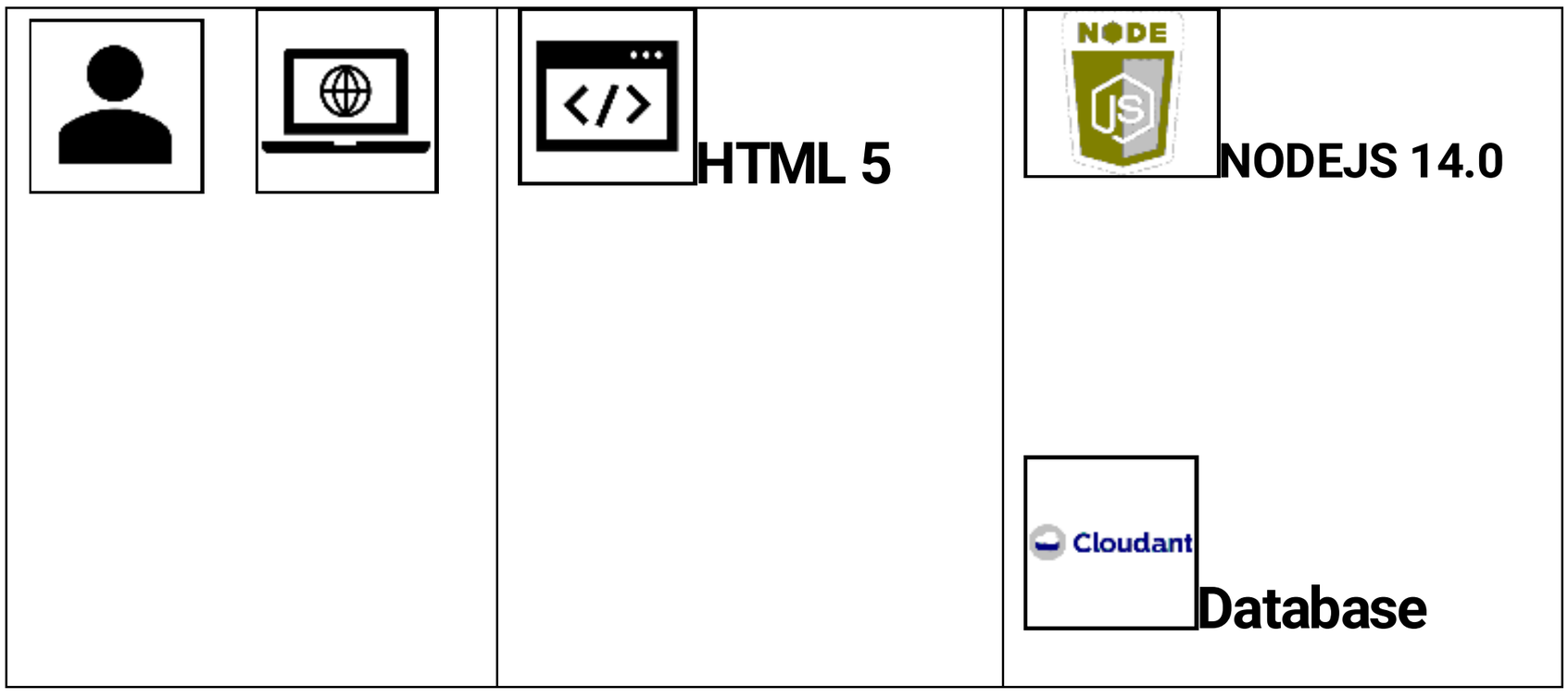
**The complete calories to be consumed ought to be adjusted extent of full scale supplements like Proteins, Carbohydrates and Fats. Additionally, there is a high opportunity of confusion of information as well as event of blunders. Additionally, it is bulky and tedious. With the expansion in volume of patients in the medical care organizations, particularly now after COVID pandemic conventional technique for the board has left stage. Therefore, a high level Health Care Management System has been the interest**

of time.

# CHAPTER 3

# PROJECT ARCHITECTURE

## 3.1 Architecture



```
import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from pulp import *

import seaborn as sns

data = pd.read_csv('nutrition.csv').drop('Unnamed: 0',axis=1)

data.head()

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 8789 entries, 0 to 8788

Data columns (total 76 columns):

 #   Column                 Non-Null Count  Dtype

---  ------                 --------------  -----

0   name                   8789 non-null   object

1   serving_size           8789 non-null   object

2   calories               8789 non-null   int64

3   total_fat              8789 non-null   object
```

4 saturated_fat 7199 non-null object

5 cholesterol 8789 non-null object

6 sodium 8789 non-null object

7 choline 8789 non-null object

8 folate 8789 non-null object

9 folic_acid 8789 non-null object

10 niacin 8789 non-null object

11 pantothenic_acid 8789 non-null object

12 riboflavin 8789 non-null object

13 thiamin 8789 non-null object

14 vitamin_a 8789 non-null object

15 vitamin_a_rae 8789 non-null object

16 carotene_alpha 8789 non-null object

17 carotene_beta 8789 non-null object

18 cryptoxanthin_beta 8789 non-null object

19 lutein_zeaxanthin 8789 non-null object

20 lucopene 8789 non-null int64

21 vitamin_b12 8789 non-null object

| | | | | |
|---|---|---|---|---|
| 22 | vitamin_b6 | 8789 | non-null | object |
| 23 | vitamin_c | 8789 | non-null | object |
| 24 | vitamin_d | 8789 | non-null | object |
| 25 | vitamin_e | 8789 | non-null | object |
| 26 | tocopherol_alpha | 8789 | non-null | object |
| 27 | vitamin_k | 8789 | non-null | object |
| 28 | calcium | 8789 | non-null | object |
| 29 | copper | 8789 | non-null | object |
| 30 | irom | 8789 | non-null | object |
| 31 | magnesium | 8789 | non-null | object |
| 32 | manganese | 8789 | non-null | object |
| 33 | phosphorous | 8789 | non-null | object |
| 34 | potassium | 8789 | non-null | object |
| 35 | selenium | 8789 | non-null | object |
| 36 | zink | 8789 | non-null | object |
| 37 | protein | 8789 | non-null | object |
| 38 | alanine | 8789 | non-null | object |
| 39 | arginine | 8789 | non-null | object |

40 aspartic_acid 8789 non-null object

41 cystine 8789 non-null object

42 glutamic_acid 8789 non-null object

43 glycine 8789 non-null object

44 histidine 8789 non-null object

45 hydroxyproline 8789 non-null object

46 isoleucine 8789 non-null object

47 leucine 8789 non-null object

48 lysine 8789 non-null object

49 methionine 8789 non-null object

50 phenylalanine 8789 non-null object

51 proline 8789 non-null object

52 serine 8789 non-null object

53 threonine 8789 non-null object

54 tryptophan 8789 non-null object

55 tyrosine 8789 non-null object

56 valine 8789 non-null object

57 carbohydrate 8789 non-null object

58 fiber 8789 non-null object

59 sugars 8789 non-null object

60 fructose 8789 non-null object

61 galactose 8789 non-null object

62 glucose 8789 non-null object

63 lactose 8789 non-null object

64 maltose 8789 non-null object

65 sucrose 8789 non-null object

66 fat 8789 non-null object

67 saturated_fatty_acids 8789 non-null object

68 monounsaturated_fatty_acids 8789 non-null object

69 polyunsaturated_fatty_acids 8789 non-null object

70 fatty_acids_total_trans 8789 non-null object

71 alcohol 8789 non-null object

72 ash 8789 non-null object

73 caffeine 8789 non-null object

74 theobromine 8789 non-null object

75 water 8789 non-null object

```
dtypes: int64(2), object(74)

memory usage: 5.1+ MB

data =

data[['name','serving_size','calories','carbohydrate','total_fat','protein
']]

print(data.info())

data.head()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 8789 entries, 0 to 8788

Data columns (total 6 columns):

#   Column       Non-Null Count  Dtype

---  ------       --------------  -----

 0   name         8789 non-null   object

 1   serving_size  8789 non-null   object

 2   calories     8789 non-null   int64

 3   carbohydrate  8789 non-null   object

 4   total_fat    8789 non-null   object

 5   protein      8789 non-null   object
```

dtypes: int64(1), object(5)

memory usage: 412.1+ KB

None

sns.countplot(data.serving_size)

<AxesSubplot:xlabel='serving_size', ylabel='count'>

data = data.drop('serving_size',axis=1)

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 8789 entries, 0 to 8788

Data columns (total 5 columns):

| # | Column | Non-Null Count | Dtype |
|---|---|---|---|
| 0 | name | 8789 non-null | object |
| 1 | calories | 8789 non-null | int64 |
| 2 | carbohydrate | 8789 non-null | object |
| 3 | total_fat | 8789 non-null | object |
| 4 | protein | 8789 non-null | object |

dtypes: int64(1), object(4)

```
memory usage: 343.4+ KB

data['carbohydrate'] = np.array([data['carbohydrate'].tolist()[i].split('
') for i in range(len(data))])[:,0].astype('float')

data['protein'] = np.array([data['protein'].tolist()[i].split(' ') for i in
range(len(data))])[:,0].astype('float')

data['total_fat'] = np.array([data['total_fat'].

week_days =
['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sund
ay']

split_values = np.linspace(0,len(data),8).astype(int)
split_values[-1] = split_values[-1]-1

def random_dataset():

    frac_data =
data.sample(frac=1).reset_index().drop('index',axis=1)

    day_data = []

    for s in range(len(split_values)-1):


day_data.append(frac_data.loc[split_values[s]:split_values[s+1]])

    return dict(zip(week_days,day_data))
```

```
def build_nutritional_values(kg,calories):
    protein_calories = kg*4
    res_calories = calories-protein_calories
    carb_calories = calories/2.
    fat_calories = calories-carb_calories-protein_calories
    res = {'Protein Calories':protein_calories,'Carbohydrates Calories':
carb_calories,'Fat Calories':fat_calories}
    return res
def extract_gram(table):
    protein_grams = table['Protein Calories']/4.
    carbs_grams = table['Carbohydrates Calories']/4.
    fat_grams = table['Fat Calories']/9.
    res = {'Protein Grams':protein_grams, 'Carbohydrates
Grams':carbs_grams,'Fat Grams':fat_grams}
    return res
build_nutritional_values(70,2000)
{'Protein Calories': 280,
 'Carbohydrates Calories': 1000.0,
```

```
 'Fat Calories': 720.0}

print(extract_gram(build_nutritional_values(70,2000)))

{'Protein Grams': 70.0, 'Carbohydrates Grams': 250.0, 'Fat Grams':
80.0}

days_data = random_dataset()

def model(day,kg,calories):

    G = extract_gram(build_nutritional_values(kg,calories))

    E = G['Carbohydrates Grams']

    F = G['Fat Grams']

    P = G['Protein Grams']

    day_data = days_data[day]

    day_data = day_data[day_data.calories!=0]

    food = day_data.name.tolist()

    c  = day_data.calories.tolist()

    x  = pulp.LpVariable.dicts( "x", indices = food, lowBound=0,
upBound=1.5, cat='Continuous', indexStart=[] )

    e = day_data.carbohydrate.tolist()

    f = day_data.total_fat.tolist()
```

```
p = day_data.protein.tolist()

prob  = pulp.LpProblem( "Diet", LpMinimize )

prob += pulp.lpSum( [x[food[i]]*c[i] for i in range(len(food))]  )

prob += pulp.lpSum( [x[food[i]]*e[i] for i in range(len(x)) ] )>=E

prob += pulp.lpSum( [x[food[i]]*f[i] for i in range(len(x)) ] )>=F

prob += pulp.lpSum( [x[food[i]]*p[i] for i in range(len(x)) ] )>=P

prob.solve()

variables = []

values = []

for v in prob.variables():

    variable = v.name

    value = v.varValue

    variables.append(variable)

    values.append(value)

    values = np.array(values).round(2).astype(float)

    sol = pd.DataFrame(np.array([food,values]).T, columns =
['Food','Quantity'])

    sol['Quantity'] = sol.Quantity.astype(float)
```

```
    return sol

sol_monday = model('Monday',70,1500)

Welcome to the CBC MILP Solver

Version: 2.10.3

Build Date: Dec 15 2019


command line - /Users/pieropaialunga/opt/anaconda3/lib/python3.

9/site-packages/pulp/apis/../solverdir/cbc/osx/64/cbc

/var/folders/zy/fwj_m0697v936qlt3c751l780000gn/T/bf42b6f636b

54411abd14e0e153166cc-pulp.mps timeMode elapsed branch

printingOptions all solution

/var/folders/zy/fwj_m0697v936qlt3c751l780000gn/T/bf42b6f636b

54411abd14e0e153166cc-pulp.sol (default strategy 1)

At line 2 NAME          MODEL

At line 3 ROWS

At line 8 COLUMNS

At line 4605 RHS

At line 4609 BOUNDS

At line 5862 ENDATA
```

Problem MODEL has 3 rows, 1252 columns and 3344 elements

Coin0008I MODEL read with 0 errors

Option for timeMode changed from cpu to elapsed

Presolve 3 (0) rows, 1226 (-26) columns and 3294 (-50) elements

0  Obj 0 Primal inf 275.41367 (3)

5  Obj 954.37596

Optimal - objective value 954.37596

After Postsolve, objective 954.37596, infeasibilities - dual 0 (0), primal 0 (0)

Optimal objective 954.3759597 - 5 iterations time 0.002, Presolve 0.00

Option for printingOptions changed from normal to all

Total time (CPU seconds):       0.00   (Wallclock seconds):       0.02

```
sol_monday = sol_monday[sol_monday['Quantity']!=0.0]

sol_monday.Quantity = sol_monday.Quantity*100

sol_monday = sol_monday.rename(columns={'Quantity':'Quantity (g)'})

sol_monday
```

```
def model(prob,day,kg,calories):

    G = extract_gram(build_nutritional_values(kg,calories))

    E = G['Carbohydrates Grams']

    F = G['Fat Grams']

    P = G['Protein Grams']

    day_data = days_data[day]

    day_data = day_data[day_data.calories!=0]

    food = day_data.name.tolist()

    c  = day_data.calories.tolist()

       x  = pulp.LpVariable.dicts( "x", indices = food, lowBound=0,
   upBound=1.5, cat='Continuous', indexStart=[] )

       e = day_data.carbohydrate.tolist()

       f = day_data.total_fat.tolist()

       p = day_data.protein.tolist()

   #    prob  = pulp.LpProblem( "Diet", LpMinimize )

    prob += pulp.lpSum( [x[food[i]]*c[i] for i in range(len(food))]  )

    prob += pulp.lpSum( [x[food[i]]*e[i] for i in range(len(x)) ] )>=E

    prob += pulp.lpSum( [x[food[i]]*f[i] for i in range(len(x)) ] )>=F
```

```
prob += pulp.lpSum( [x[food[i]]*p[i] for i in range(len(x)) ] )>=P

prob.solve()

variables = []

values = []

for v in prob.variables():

    variable = v.name

    value = v.varValue

    variables.append(variable)

    values.append(value)

    values = np.array(values).round(2).astype(float)

    sol = pd.DataFrame(np.array([food,values]).T, columns =
['Food','Quantity'])

    sol['Quantity'] = sol.Quantity.astype(float)

    sol = sol[sol['Quantity']!=0.0]

    sol.Quantity = sol.Quantity*100

    sol = sol.rename(columns={'Quantity':'Quantity (g)'})

    return sol

def total_model(kg,calories):
```

```
result = []

for day in week_days:

    prob  = pulp.LpProblem( "Diet", LpMinimize )

    print('Building a model for day %s \n'%(day))

    result.append(model(prob,day,kg,calories))

return dict(zip(week_days,result))
```

# CHAPTER 4

# ARCHITECTURE BLOCKS DETAIL WORKING

## 4.1 Blocks

## A. Hardware Requirements

## 1.Laptop or PC

- **i3 Processor Based Computer**
- **1GB RAM**
- **5 GB Hard Disk**

**2. Android Phone or Tablet**

**1.2 Quad core Processor or higher.**

- **1 GB RAM**

**B. Software Requirements**

**1.Laptop or PC**

- **Windows 7 or higher.**
- **SQL Server 2008**
- **Java**
- **Android Studio**
- **Azure Data Studio**

**2. Android Phone or Tablet**

- **Android v5.0 or Higher**

## WORKING

**It goes about as an eating regimen specialist like a genuine dietician. This framework acts likewise as that of a dietician. An individual to know his/her eating routine arrangement needs to give a data to the dietician, for example, its body type, weight, level and age. Client's BMI and BMR will be determined and in view of BMR and BMI result, the eating regimen will be created.**

**Comparative way this framework likewise gives the eating regimen plan as per the data entered by the client. The framework asks every one of his information from the client and cycles it to give the eating routine arrangement to the client. Consequently, the client doesn't have to visit any dietician which additionally saves time and the client can get the expected eating regimen plan in only a tick.**

**The framework will give more exact outcomes as it acknowledges the information entered by the client and cycles it relying upon certain measurements definitely known to the application based**

**on which an eating regimen plan is created and inquire as to whether the client acknowledges the eating routine arrangement or need a few changes. On the off chance that client needs transforms, he can put custom solicitation and the dietician will refresh the eating routine arrangement according to the solicitation so the eating regimen is kept up with as well.**

**The BMI (Body Mass Index) is determined by applying the accompanying condition:**

**SI, Metric Units:**

**BMI = 703 * mass(kg)/height^2(m)**

**USC Units:**

**BMI = 703 * mass(lbs)/height^2(in)**

**The BMR is determined utilizing the Mifflin-St Jor Equation,**

**For Men,**

**BMR = 10W + 6.25H - 5A +5**

**For Women,**

**BMR = 10W + 6.25H - 5A - 161**

**Where,**

**A= Age**

**W = Weight**

**H = Height**

## CONCLUSION

**The System is a valuable apparatus for instructing clients on healthful related themes with the assistance of enormous and dependable information base made with help of master dieticians. Many individuals counsel a dietician when needing a legitimate eating regimen to go with their activities. Since, our proposition will assist individuals with the eating routine; they won\'t have to visit dieticians. The clients will get diet conveyed to their screens for them which will save time as well as cash as the administrations given by our undertaking will be liberated from cost, in contrast to different choices available right now. Our application is utilizing man-made reasoning calculation called RETE calculation so every**

single client will get a customized diet as indicated by their need and inclinations.

# REFERENCES

[1] Sneha Sadhwani [1], \"way of life and wellbeing: advantages of utilizing wellness applications\" medindia.net on 22nd June 2019 [2] Chinan Mehta [2], \"Top advantages of building wellbeing and wellness applications: significance of wellbeing and wellness applications\" SolutionAnalysts.com on third walk 2020. [3] Oleksandr Sh [3], \"How to make a wellness application that moves clients\" cleverhead.com on 30th November 2020 [4] Antasia Khomych [4], \"Ten should have highlights for wellbeing and wellness applications\" blog.getsocial.im on 22nd september 2020 [5] Jen-Hao

Hsiao and Henry Chang [5],\"SmartDiet: An individual eating routine specialist for wellbeing feast arranging\" IBM Research Collaboratory, Taiwan [6] Divya Mogaveera and Vedant Mathur [6],\"e-Health Monitoring framework with diet and wellness proposal utilizing AI\", ICICT 2021 [7] Dr. Meera Gandhi and Vishal Kumar Singh [7], \"IntelliDoctor - AI based clinical colleague\", ICONSTEM 2019 [8] Sanchit Kalra, Garvit Arora and Rajat Aggarwal [8], “ Application of Artificial Intelligence for weekly meal planning for children” IIT Delhi [9] Fule Wang, Yuan Yuan, Yu Pan, Nin Hu [9], “ Study on the principles of the intelligent Diet Arrangement System based on Multi Agent” , College of Management Science and Engineering, Nanjing University of Technology, Nanjing [10] Shaikh Saqib, khan Vaqui, Shaikh Mohd Asfaque [10],” Online Dietician Using Artificial Intelligence” , Rizwi College of Engineering [11] Kartik K , Vignesh K, M.Dhurgadevi [11], “ Android based diet consultant using rule pattern based algorithm” , Nehru Institute of Engineering and Technology Coimbatore [12] Prajakta Dadasaheb Jadhav [12], “ AI Dietician” , Modern Education Society’ s College of Engineering Pune,India. [13] Siddharthan Chitra

**Suseendran [13]," Virtual Nutritionist using AI" , IJEAT(2019) [14] Yulong Xu [14]," Research and Implementation of Improved Random Forest Algorithm Based on Spark" ,Hebei University of Technology Tianjin, China (2017) [15] Xiaolong Xu, Wen Chen [15], Implementation and Performance Optimization of Dynamic Random Forest" , IEEE (2017)**

# CODE

## Please Provide Code through Git Hub Repo Link

https://github.com/SoundaryaSenthilkumar/Building-your-Diet-using-Artificial-Intelligence.git