**PROJECT:- Nutrition Assistant**

**Team Members**

Surya Pratap Singh(19BCG10064)

Apoorv Shrivastava(19BCG10038)

Poornima Raghuwanshi(19BCE10444)

Shourya Shekhar Srivastava(20BCE10830)

App link:-<https://nutrition-demo.herokuapp.com/>

1. **INTRODUCTION**

**1.1 Overview**

The name of this web application is Nutrition Assistant as the name suggest we made it to allow people to have a healthy diet and to let them know what they are eating , what it is made of , what are it’s ingredients and so on . We have use Java Spring Boot , MySQL , Rapid API , Heroku and git to developed this project from scratch.

The main aim of this web application is to ensure that a user have a healthy diet.

We have use RapidAPI to get different amount of data we need to calculate the calories or to get the recipe.

The website was designed taking into account the following: - To be simple and light - More informative as possible - Maintenance easiness & content update

We have a fun section too in the website named as chillax it contains two main parts first is a joke sections and the other is trivia section.

The joke section gives the user a joke related to food when click on the appropriate button.

The trivia Section gives user a specific about different food or recipe.

We can get any recipe made from a specific ingredient.

We can calculate what specific amount of nutritional value a particular dish contains let it be carbohydrates , fat or etc.

We can get any specific recipe with number of servings a user has to intake to get that particular calories count with the link to make that particular dish.

* 1. **Purpose**

With the help of this project user can calculate are they getting proper nutrition value or not , or how much calories a specific food dish or an item contains.

With the help of our project we can automatically calculate the nutritional information for any recipe, analyse recipe costs, visualize ingredient lists, find recipes for what's in your fridge, find recipes based on special diets, nutritional requirements, or favourite ingredient.

To make the web app more interactive we have tried to make a fun section too named as chillax in this section user can learn about some trivia which is very specific and not very well known among people or he may have a laugh with our joke section.

This project it can be used in many different fields of the real world like in gym or some nutrition base company as an integration to their own software to get the basic information as soon as possible.

Our project is very useful for common people who want to have diet and want to calculate their daily intake of calories but don’t want to buy expensive apple watch to remember it for them they can use this web app to make a list and get their total calories intake.

1. **THEORITICAL ANALYSIS**

**2.1 Block diagram**

**Diagram

Description automatically generated**

* 1. **Hardware / Software designing**

**SOFTWARE RQUIREMENTS:-**Graphical user interface, application, table

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

**Application

Description automatically generated with low confidence**

1. **THEORITICAL ANALYSIS**

**Diagram

Description automatically generated**

1. **RESULT**

So our project has 3 main sections:-

Graphical user interface, application, website

Description automatically generated

* + - * **Quick Browse**

With the help of quick browse 3 more menu side options open up , they are as follows:-

A picture containing timeline

Description automatically generated

* + - * + Recipe by calories:- With the help of this feature user can get specific recipes of dishes according to the number of calories he enters and number of calories her needs to intake to get that specific calories.

For ex. If user need 55 calories and click get the recipes he will get some recipes which can satisfy the user needs with the number of servings he need to take with a URL to make recipe.

Graphical user interface, application

Description automatically generated

Nutrients

With the help of this feature user can calculate the exact amount of calories carbohydrates protein and fats it contains.

A picture containing text

Description automatically generated

* + - * **Quick Answer**

Quick Answer option helps the user to tell him how much kilo calories are present in 1 serving of a specific item and how much percentage 1 serving of that item meets the daily quota he needs of it. With a URL for user to see that item .For ex in the below pic we can see how much kilo calories are present in flour with a URL.

**A picture containing box and whisker chart

Description automatically generated**

* + - * **Chillax**

Chillax consist of mainly 2 sections:-

A picture containing graphical user interface

Description automatically generated

Joke Section :- When you click it you see a really funny joke something related to food or dishes.

Trivia Section: -When you click this button you see a very specific trivia about food which is rare knowledge.

1. **Advantages and Disadvantages**
   * + - Advantages
2. Easy to use.
3. User Friendly UI
4. No need to sign up use the site.
5. Fast access.
6. Web Link Management: Providing link resources for site users is simple and you can sort them into categories, even count every click.
7. New changes can be made easily in the web application even live changes.
8. Docker Image of the above application can easily be made to make it possible to run on various systems.
9. User have fast access to all the links of the recipe they need not only names so they can easily made the dish.
10. Lots of options for user to explore on the web app.
    * + - Disadvantages
11. There are less features available right now.
12. No login to save user data currently.
13. Not very secure.
14. **Applications**

The main application of our web app are as follows:-

* Gyms:- Gym trainers can just open up our web application to check for the nutrients a particular food item contains to recommend it to the people who come their to gym , just simple people they don’t even have to see the extra information just the important related to the item or dish.
* Restaurants:- If a customer ask for a dish with some specific nutritional value the cook can always open our web app go into quick browse “Recipe by calories “ and find the adequate dishes for that person.
* For persons who are very into keeping their body fit they can always check with some few clicks how much calories their food item contains and they can intake the food accordingly.
* Homes:- Many times mom’s are very concern about their children health they want their food to contain every bit of nutritional value so they can always go to our web app search about “Recipe by calories” and they can get different amount of recipes to cook with the URL.
* Hospitals:- In hospitals patients are often told to eat food with some specific amount of nutritional value ,not less and not more nurse can easily use our website to get that information.
* Students:- Students who love to be fit and cook can use our trivia section to gain knowledge while at the same time getting healthy recipes.
* We will be adding new features soon so it will be really easy to search any recipe according to item that contains , carbohydrate it

1. **Conclusion**

To conclude our project we can say our web application is mainly aim at right now to find a healthy diet for people and make them knowledgeable about different food items and dishes about their nutritional value, how much calories , carbohydrates , protein etc. is present in them , we use the technology of spring boot to make this web application so in future it will be easy to add new features as microservices , we have MySQL as the database MySQL delivers the ease of use, scalability, and high performance, our web app project is pretty much platform independent after making the docker image of it. We have use git and Heroku to deploy it as it is now. Our entire web application consists of one main point that is nutritarian value which we can easily find the menu of it with 3 main option to explore quick browse, quick answer and chillax , our aim is to provide a safe and secure platform easy to use so that anyone can make whatever healthy food they want with the proper knowledge technique they required which they will be getting from our web application ,we have additionally used RapidAPI to get data about various food , dishes etc , we will be using this and many other more API’s to extend our project further , our mail languages used in the project were java ,html ,css.

1. **FUTURE SCOPE**

Our future scope is as follows:-

* Introducing Login page connected with an independent database and with Gmail ,Facebook to save users search data so they can easily look back if they want to.
* Introducing new features such as:-

Search recipe by dish name

Search recipe by ingredient name

Meal Plan week

To Generate Shopping list for healthy food items by nutritional value enter by user.

Chatbot feature for help and FAQ.

Get food name by image analysis

Search for nearby restaurants which serve a particular type of dish user wants

Search nearby shopping malls , grocery shops to get what user wants.

History about different dishes how they were first made every detail and every relevant source available for curious users.

To generate food items and dishes list according to the user health condition user needs to input what conditions they have and our web application will generate a list of items which will be beneficial for the user to eat.

A specific tab with different food dishes images and on clicking on them user will get all the information about them including a tutorial in video and written format.

1. **Appendix**

**API CALL**

package com.hemlata.app;

import java.io.IOException;

import java.net.URI;

import java.net.URLEncoder;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.net.http.HttpResponse;

public class APICall {

public String data(String query) throws InterruptedException, IOException

{

String encoded=URLEncoder.encode(query,"UTF-8").replace("+", "%20");

String url="https://rapidapi.p.rapidapi.com/recipes/quickAnswer?q="+encoded;

System.out.println("Enecoded query: "+encoded);

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create(url))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

public String Joke() throws IOException, InterruptedException

{

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create("https://rapidapi.p.rapidapi.com/food/jokes/random"))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

public String trivia() throws IOException, InterruptedException

{

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create("https://rapidapi.p.rapidapi.com/food/trivia/random"))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

public String recipeByCal(int cals) throws IOException, InterruptedException

{

String url="https://rapidapi.p.rapidapi.com/recipes/mealplans/generate?targetCalories="+cals+"&timeFrame=day";

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create(url))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

public String convertAmount(String a,String b,String c, float d) throws IOException, InterruptedException

{

String url="https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.com/recipes/convert?ingredientName="+a+"&targetUnit="+b+"&sourceUnit="+c+"&sourceAmount="+d;

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create(url))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

public String nutByDish(String name) throws IOException, InterruptedException

{

String encoded=URLEncoder.encode(name,"UTF-8").replace("+", "%20");

String url="https://spoonacular-recipe-food-nutrition-v1.p.rapidapi.com/recipes/guessNutrition?title="+encoded;

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create(url))

.header("x-rapidapi-key", "4e96ca0c82msh7acc5424070fda1p1c7d82jsn9ec25a33485e")

.header("x-rapidapi-host", "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com")

.method("GET", HttpRequest.BodyPublishers.noBody())

.build();

HttpResponse<String> response = HttpClient.newHttpClient().send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

return response.body();

}

}