

# **Healthy Food Suggetion With Calorie InTake**

**Team: Easy Life**

## Submitted By:



**Afzal Hossain Raju**

**0242310005341031**



**Mohammad Abu Obaida Hamim**

**242310005341468**



**Mohammad Nasim**

**242310005341101**

## Submitted To:

**Mr. Md. Abdul Hannan**

**Lecturer**



**Department of Software Engineering  
Daffodil International University**

# **Healthy Food Suggestion With Calorie Intake**

Introducing our capstone project on healthy food suggestion with calorie intake tracking developed using C programming.

1. Introduction.
2. Registration and Login
3. Function and Structure
4. File
5. Our Service
6. Main Function
7. Our Limitation
8. Conclusion



## Registration And Login:

## Login

```

        if (File.Exists(Path.Combine(Path.GetTempPath(), TempName)))
        {
            File.Delete(Path.Combine(Path.GetTempPath(), TempName));
        }
    }

    private void GetTempName(String TempName)
    {
        Random rnd = new Random();
        int rndInt = rnd.Next(1, 1000);
        TempName = rndInt.ToString();
    }
}

```

## **Registration**

```

    if (current == null) {
        new current = new Node(data);
        new current.next = null;
        new current.prev = null;
        return current;
    }
    else {
        Node previous = current;
        current = current.next;
        previous.next = null;
        previous.prev = null;
        return previous;
    }
}
else {
    Node previous = current.prev;
    current = current.next;
    previous.next = null;
    previous.prev = null;
    return previous;
}
}

public void insert(int data) {
    Node current = head;
    while (current != null) {
        if (current.data == data) {
            System.out.println("Element already exists in list");
            return;
        }
        current = current.next;
    }
    insertAtEnd(data);
}

public void delete(int data) {
    Node current = head;
    while (current != null) {
        if (current.data == data) {
            System.out.println("Element found in list");
            if (current == head) {
                head = current.next;
            }
            else {
                current.prev.next = current.next;
                current.next.prev = current.prev;
            }
            return;
        }
        current = current.next;
    }
    System.out.println("Element not found in list");
}

public void display() {
    Node current = head;
    while (current != null) {
        System.out.print(current.data + " ");
        current = current.next;
    }
}
}

```



```

int registerUser() {
    char username[MAX_USERNAME_LEN];
    char password[MAX_PASSWORD_LEN];
    int isTaken;
    do {
        printf("\nEnter a new username: ");
        scanf("%s", username);

        isTaken = isUsernameTaken(username);
        if (isTaken) {
            printf("Username is already taken. Please choose another one.\n");
        }
    } while (isTaken);

    while (1) {
        printf("Enter a password (at least 6 characters, including uppercase, lowercase, digit, and special character): ");
        scanf("%s", password);

        if (strlen(password) >= 6 && isStrongPassword(password)) {
            break;
        } else {
            printf("Password too weak. Please try again.\n\n");
        }
    }

    FILE *fp = fopen("users.txt", "a");
    if (fp == NULL) {
        printf("Error opening file for writing.\n");
        return 1;
    }

    fprintf(fp, "%s %s\n", username, password);
    fclose(fp);

    printf("Registration successful!\n");
    return 0;
}

```

SE133project.c x FoodSuggestions.txt  
SE133project.c > registerUser()  
X SE133project.c 488 int registerUser() {  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Welcome to Easy\_Life  
=====

1. Register  
2. Login  
3. Exit  
Enter your choice:

Welcome to Easy\_Life  
=====

1. Register  
2. Login  
3. Exit  
Enter your choice: 1  
Enter a new username: raju  
Username is already taken. Please choose another one.  
Enter a new username: raju

Welcome to Easy\_Life  
=====

1. Register  
2. Login  
3. Exit  
Enter your choice: 1  
Enter a new username: raju  
Username is already taken. Please choose another one.  
Enter a new username: raju  
Enter a password (at least 6 characters, including uppercase, lowercase, digit, and special character): polaris  
Password too weak. Please try again.  
Enter a password (at least 6 characters, including uppercase, lowercase, digit, and special character):

X SE133project.c 488 int registerUser() {  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
Welcome to Easy\_Life  
=====

1. Register  
2. Login  
3. Exit  
Enter your choice: 1

```

int loginUser() {
    char username[MAX_USERNAME_LEN];
    char password[MAX_PASSWORD_LEN];
    char file_username[MAX_USERNAME_LEN];
    char file_password[MAX_PASSWORD_LEN];

    printf("\nEnter your username: ");
    scanf("%s", username);

    printf("Enter your password: ");
    scanf("%s", password);
    FILE *fp = fopen("users.txt", "r");
    if (fp == NULL) {
        printf("Error opening file for reading.\n");
        return 1;
    }
}

```

```

while (fscanf(fp, "%s %s", file_username, file_password) != EOF) {

    if (strcmp(username, file_username) == 0 && strcmp(password, file_password) == 0) {
        fclose(fp);
        printf("Login successful!\n");
        isLoggedIn=true;
        return isLoggedIn;
    }
}

fclose(fp);
printf("Invalid username or password. Please try again.\n");
return 1;

```

```

SE133project.c 446 int loginUser() {
FoodSuggestions.txt 462
SE133 PROJECT 463

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

1. Register
2. Login
0. Exit
Enter your choice: 1

```

```

SE133 PROJECT 463

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Enter Your Weight(in Kg): 75
Enter your Height(in Feet):
First enter feet: 5
Now enter inch: 9
Enter your Age: 23
Enter Gender:
1.Male
2.Female
1

Choose your activity level:
1. Sedentary (little to no exercise)
2. Lightly active (light exercise/sports 1-3 days/week)
3. Moderately active (moderate exercise/sports 3-5 days/week)
4. Very active (hard exercise/sports 6-7 days a week)
5. Extra active (very hard exercise/sports & physical job or 2x training)
Enter your choice: 3

```

# Project Details



# Functionality Overview

## Function



## All functions of the program

```

int registerUser();
int loginUser();
int isUsernameTaken(const char *username);
int isStrongPassword(char *password);
void logoutUser();

int CalculateHealthWeight( float height, int age, int gender, float activityFactor );
float BMI( float weight, float height, FILE *fIn);

void suggestFood1(Food1 *food1, int size, float calorieNeeds, FILE *fIn);
void suggestFood2(Food2 *food2, int size, float calorieNeeds, FILE *fIn);
void suggestFood3(Food3 *food3, int size, float calorieNeeds, FILE *fIn);
void suggestFood4(Food4 *food4, int size, float calorieNeeds, FILE *fIn);

```

```
activityFactor);  
  
FILE *file);
```

## Structure

## Struct Declaration

Data entry to the variable food1 of type struct Food1

```
19 import java.util.ArrayList;
20 import java.util.List;
21 import java.util.Map;
22 import java.util.Set;
23 
24 import org.junit.Assert;
25 import org.junit.Test;
26 import org.junit.runner.RunWith;
27 import org.junit.runners.JUnit4;
28 import org.junit.runners.Parameterized;
29 import org.junit.runners.Parameterized.Parameters;
30 
31 import com.google.common.collect.ImmutableList;
32 import com.google.common.collect.ImmutableMap;
33 import com.google.common.collect.ImmutableSet;
34 import com.google.common.collect.Lists;
35 import com.google.common.collect.Maps;
36 import com.google.common.collect.Sets;
37 
38 import static org.junit.Assert.*;
39 import static org.junit.Assert.assertEquals;
40 import static org.junit.Assert.assertTrue;
```

**Table 2** Results of the study of the effect of the addition of 10% of the following materials on the properties of the concrete  
1) Gypsum, 2) Lime, 3) Lime-gypsum, 4) Lime-silica, 5) Lime-silica-gypsum, 6) Lime-silica-gypsum-ash, 7) Lime-silica-gypsum-ash-slag, 8) Lime-silica-gypsum-ash-slag-cement, 9) Lime-silica-gypsum-ash-slag-cement-sand, 10) Lime-silica-gypsum-ash-slag-cement-sand-cement, 11) Lime-silica-gypsum-ash-slag-cement-sand-cement-sand, 12) Lime-silica-gypsum-ash-slag-cement-sand-cement-sand-cement, 13) Lime-silica-gypsum-ash-slag-cement-sand-cement-sand-cement-sand, 14) Lime-silica-gypsum-ash-slag-cement-sand-cement-sand-cement-sand-cement.

# Struct Declaration

```
12  
13  
14     typedef struct Food1{  
15         char name[100];  
16         int calories;  
17         float grams;  
18     }Food1;  
19     typedef struct Food2{  
20         char name[100];  
21         int calories;  
22         float grams;  
23     }Food2;  
24     typedef struct Food3{  
25         char name[100];  
26         int calories;  
27         float grams;  
28     }Food3;  
29     typedef struct FoodE{  
30         char name[100];  
31         int calories;  
32         float grams;  
33     }FoodE;
```

# Data entry to the variable food1 of type struct Food1

```
59     Food1 foods1[] = {  
60         {"Plain Water",00, 1000.0},  
61         {"4 Piece Red Wheat Rooti/2 Piece Nun Rooti",240, 200.0},  
62         {"Apple slices with peanut butter", 150, 200.0},  
63         {"2 Piece Banana", 210, 240.0},  
64         {"Cucumber salad", 40, 150.0},  
65         {"Boiled Egg", 54, 48.0},  
66         {"Cabbage / Tamatoes/ Gourd", 43, 100.0},  
67         {"Khichuri (Rice and Lentil Porridge)", 300, 200.0},  
68         {"Paratha with Egg Curry", 350, 150.0},  
69         {"Khichuri (Rice and Lentil Porridge)", 300, 200.0},  
70         {"Panta Bhat(Fermented Rice) with Fried Hilsa Fish", 400, 250.0},  
71         {"Chapati with Mixed Vegetable Curry", 250, 180.0},  
72         {"Chirer Polao (Flattened Rice Pilaf)", 270, 200.0},  
73         {"Aloo Bhaji(Potato Fry) with Roti", 300, 190.0},  
74         {"Shemai (Vermicelli Pudding)", 280, 150.0},  
75         {"Dal Puri (Lentil-Stuffed Bread)", 320, 180.0},  
76         {"Singara (Samosa) with Chutney", 220, 120.0},  
77         {"Beguni (Eggplant Fritters) with Rice", 350, 200.0},  
78         {"Rice Flour Pancakes with Jaggery", 300, 160.0},  
79         {"Luchi with Alur Dom (Puffy Bread with Potato Curry)", 450, 200.0},  
80         {"Fried Rice with Chicken", 400, 250.0},  
81         {"Green tea", 1, 100.0},  
82         {"Doi Chira (Flattened Rice with Yogurt)", 250, 150.0},  
83         {"Puffed Rice with Jaggery and Coconut", 200, 100.0},
```



# All functions of the program

```
int registerUser();
```

```
int loginUser();
```

```
int isUsernameTaken(const char *username);
```

```
int isStrongPassword(char *password);
```

```
void logoutUser();
```

```
int Calorie(float weight, float height, int age, int gender, float activityFactor);
```

```
float BMI(float weight, float height, FILE *file);
```

```
void suggestFood1(Food1 *foods1, int size, float calorieNeeds, FILE *file);
```

```
void suggestFood2(Food2 *foods2, int size, float calorieNeeds, FILE *file);
```

```
void suggestFood3(Food3 *foods3, int size, float calorieNeeds, FILE *file);
```

```
void suggestFoodE(FoodE *foodsE, int size, float calorieNeeds, FILE *file);
```

```
void suggestFood1(Food1 *foods1, int size, float calorieNeeds, FILE *file);
```

```
141 void suggestFood1(Food1 *foods1, int size, float calorieNeeds, FILE *file) {  
142     printf("Food1 suggestion for serving size %d: %s\n", size, foods1->name);  
143     if (size <= 100) {  
144         printf("Calories: %d\n", foods1->calories);  
145     } else if (size > 100) {  
146         printf("Calories: %d\n", foods1->calories * 2);  
147     } else {  
148         printf("Calories: %d\n", foods1->calories / 2);  
149     }  
150 }
```

```
161 void suggestFood2(Food2 *foods2, int size, float calorieNeeds, FILE *file) {  
162     printf("Food2 suggestion for serving size %d: %s\n", size, foods2->name);  
163     if (size <= 100) {  
164         printf("Calories: %d\n", foods2->calories);  
165     } else if (size > 100) {  
166         printf("Calories: %d\n", foods2->calories * 2);  
167     } else {  
168         printf("Calories: %d\n", foods2->calories / 2);  
169     }  
170 }
```

```
float BMI(float weight, float height, FILE *file);
```

```
181 float BMI(float weight, float height, FILE *file) {  
182     float BMI; // Variable to store BMI value  
183     float weightInKg = weight * 0.453592; // Convert weight from lbs to kg  
184     float heightInM = height * 0.0254; // Convert height from inches to meters  
185     BMI = weightInKg / (heightInM * heightInM); // Calculate BMI  
186     if (BMI < 18.5) {  
187         printf("Underweight: %f\n", BMI); // Underweight BMI range  
188     } else if (BMI > 18.5 & BMI < 24.9) {  
189         printf("Normal weight: %f\n", BMI); // Normal weight BMI range  
190     } else if (BMI > 24.9 & BMI < 29.9) {  
191         printf("Overweight: %f\n", BMI); // Overweight BMI range  
192     } else {  
193         printf("Obesity: %f\n", BMI); // Obesity BMI range  
194     }  
195 }
```

```
int Calorie(float weight, float height, int age, int gender, float activityFactor);
```

```
196 int Calorie(float weight, float height, int age, int gender, float activityFactor) {  
197     float Calorie; // Variable to store Calorie value  
198     float weightInKg = weight * 0.453592; // Convert weight from lbs to kg  
199     float heightInM = height * 0.0254; // Convert height from inches to meters  
200     Calorie = (10 * weightInKg) + (6.25 * heightInM) + (5 * age) - (170 * gender) + activityFactor;  
201     if (Calorie < 1000) {  
202         printf("Calorie: %f\n", Calorie); // Low Calorie output  
203     } else if (Calorie > 1000 & Calorie < 1500) {  
204         printf("Calorie: %f\n", Calorie); // Moderate Calorie output  
205     } else {  
206         printf("Calorie: %f\n", Calorie); // High Calorie output  
207     }  
208 }
```

```
int Calorie(float weight, float height, int age, int gender, float activityFactor);
```

```
C SE133project.c > ⚏ Calorie(float, float, int, int, float)
483 }
484
485 int Calorie(float weight, float height, int age, int gender, float activityFactor){
486
487     float BMR;
488     float heightMeter = height * 0.3048;
489
490     printf("=====\\n");
491     printf("Easy_Life\\n");
492     printf("=====\\n");
493     printf("\\n\\n");
494
495     if (gender== 1) {
496         // For males
497         BMR =88.362 + (13.397 * weight) + (4.799 * heightMeter *100) - (5.677 * age);
498     }
499     else if (gender == 2) {
500         // For females
501         BMR = 447.593 + (9.247 * weight) + (3.098 * heightMeter *100) - (4.330 * age);
502     }
503     else {
504         printf("Invalid gender.\\n\\n Please enter '1' for male \\n '2' for female.\\n");
505         BMR = -1; // Return -1 to indicate error
506     }
507
508     calorieNeeds =BMR*activityFactor;
509     //printf("Calories is %f\\n\\n",calorieNeeds);
510
511     return calorieNeeds;
512 }
```

```
float BMI(float weight, float height, FILE *file);
```

```
C SE133project.c > ⚙ suggestFood1(Food1 *, int, float, FILE *)
512 }
513
514 float BMI(float weight, float height, FILE *file){
515     float bmi;
516     float heightMeter = height * 0.3048;
517
518     bmi=weight/(heightMeter*heightMeter);
519
520     printf("Your BMI is: %.2f\n\n",bmi);
521     fprintf(file, "Your BMI is: %.2f\n\n", bmi);
522
523     if (bmi < 16) {
524         printf(" - Very underweight: Consult a doctor for a personalized plan.\n");
525         fprintf(file, " - Very underweight: Consult a doctor for a personalized plan.\n");
526     }
527     else if (bmi >= 16 && bmi < 18.5) {
528         printf(" - Underweight: Focus on calorie-dense foods with healthy fats.\n");
529         fprintf(file, " - Underweight: Focus on calorie-dense foods with healthy fats.\n");
530     }
531     else if (bmi >= 18.5 && bmi <= 24.9) {
532         printf(" - Normal weight: Maintain a balanced diet with all food groups.\n");
533         fprintf(file, " - Normal weight: Maintain a balanced diet with all food groups.\n");
534     }
535     else if (bmi >= 25 && bmi <= 29.9) {
536         printf(" - Overweight: Focus on portion control and nutrient-rich foods.\n");
537         fprintf(file, " - Overweight: Focus on portion control and nutrient-rich foods.\n");
538     }
539     else {
540         printf(" - Obese: Aim for gradual weight loss through a combination of calorie reduction and increased physical activity.\n");
541         fprintf(file, " - Obese: Aim for gradual weight loss through a combination of calorie reduction and increased physical activity.\n");
542     }
543     return 0;
544 }
```



```
void suggestFood1(Food1 *foods1, int size, float calorieNeeds, FILE *file);
```

```
545
546 void suggestFood1(Food1 *foods1, int size, float calorieNeeds, FILE *file) {
547     printf("Your required calorie for morning is: %.2f\n\n",calorieNeeds);
548     fprintf(file, "Your required calorie for morning is: %.2f\n\n",calorieNeeds);
549
550     for (int i = 0; i < size; i++) {
551         if (foods1[i].calories <= calorieNeeds) {
552             printf("%s: .... .... %.2f grams/ml\n", foods1[i].name, foods1[i].grams);
553             fprintf(file, "%s: .... .... %.2f grams/ml\n", foods1[i].name, foods1[i].grams);
554             calorieNeeds -= foods1[i].calories;
555         }
556         if (calorieNeeds <= 0)
557             break;
558     }
559 }
560
```

```
560
561 void suggestFood2(Food2 *foods2, int size, float calorieNeeds, FILE *file) {
562     printf("Your required calorie for noon is: %.2f\n\n",calorieNeeds);
563     fprintf(file, "Your required calorie for noon is: %.2f\n\n",calorieNeeds);
564
565     for (int i = 0; i < size; i++) {
566         if (foods2[i].calories <= calorieNeeds) {
567             printf("%s: .... .... %.2f grams/ml\n", foods2[i].name, foods2[i].grams);
568             fprintf(file,"%s: .... .... %.2f grams/ml\n", foods2[i].name, foods2[i].grams);
569             calorieNeeds -= foods2[i].calories;
570         }
571         if (calorieNeeds <= 0)
572             break;
573     }
574 }
575
```



# File and Our Services

## File

There are total two files  
in the program

1. users.txt
2. FoodSuggestions.txt



## Our Services

### Our Services:

1. Provides BMI
2. Provides activity suggestions based on BMI
3. Provides a healthy diet chart
4. Provides an online prescription



**There are total two files  
in the program**

1. users.txt
  2. FoodSuggestions.txt

users.txt

## **FoodSuggestions.txt**

```
int main(
```

# users.txt

```
int isUsernameTaken(const char *username);
```

```
376 int isUsernameTaken(const char *username) {  
377     char file_username[MAX_USERNAME_LEN];  
378     char file_password[MAX_PASSWORD_LEN];  
379  
380     FILE *fp = fopen("users.txt", "r");  
381     if (fp == NULL) {  
382         printf("Error opening file for reading.\n");  
383         return 0;  
384     }  
385  
386     while (fscanf(fp, "%s %s", file_username, file_password) != EOF) {  
387         if (strcmp(username, file_username) == 0) {  
388             fclose(fp);  
389             return 1; // Username is taken  
390         }  
391     }  
392  
393     fclose(fp);  
394     return 0; // Username is not taken  
395 }
```

```
int registerUser();
```

```
432  
433     FILE *fp = fopen("users.txt", "a");  
434     if (fp == NULL) {  
435         printf("Error opening file for writing.\n");  
436         return 1;  
437     }  
438  
439     fprintf(fp, "%s %s\n", username, password);  
440     fclose(fp);  
441  
442     printf("Registration successful!\n");  
443     return 0;  
444 }
```

```
int loginUser();
```

```
457     FILE *fp = fopen("users.txt", "r");  
458     if (fp == NULL) {  
459         printf("Error opening file for reading.\n");  
460         return 1;  
461     }  
462  
463  
464     while (fscanf(fp, "%s %s", file_username, file_password) != EOF) {  
465  
466         if (strcmp(username, file_username) == 0 && strcmp(password, file_password) == 0) {  
467             fclose(fp);  
468             printf("Login successful!\n");  
469             isLoggedIn=true;  
470             return isLoggedIn;  
471         }  
472     }  
473  
474     fclose(fp);  
475     printf("Invalid username or password. Please try again.\n");  
476     return 1;  
477 }
```

# FoodSuggestions.txt

```
int main();
```

```
308 // Open the file for writing food suggestions
309 FILE *file = fopen("FoodSuggestions.txt", "w");
310 if (file == NULL) {
311     printf("Error opening file for writing.\n");
312     return 1;
313 }
314
315
316 fprintf(file, "=====\\n");
317 fprintf(file, "Easy_Life\\n");
318 fprintf(file, "=====\\n");
319 fprintf(file, "\\n\\n");
320
321 BMI(weight, height, file);
322
323 // Suggest food based on calculated calorie needs
324 printf("\\n\\nFood suggestions based on your calorie requirements:\\n");
325 fprintf(file, "\\n\\n\\nFood suggestions based on your calorie requirements:\\n\\n");
326
327 printf("\\nMorning meal:\\n");
328 fprintf(file, "Morning meal:\\n");
329 suggestFood1(foods1, numFoods1, morningCalories, file);
330 printf("-----\\n\\n");
331 fprintf(file, "-----\\n\\n");
332
```

## **Our Services:**

1. Provides BMI
2. Provides activity suggestions based on BMI
3. Provides a healthy diet chart
4. Provides an online prascription



# Conclusion

- Summarizing the successful development and implementation of our healthy food suggestion project.

## Main Function

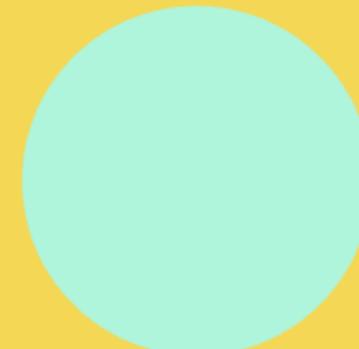


```
1# Main function
2def main():
3    # Your code here
```

1Main\_function

2 HealthyFoodSuggestionWithCalorieIntake

## Limitation



## Reference



The screenshot shows a Windows terminal window with the following content:

```
SE133project.c X FoodSuggestions.txt
SE133project.c > main()

55 int Calorie(float weight, float height, int age, int gender, float activityFactor);
56 float BMI(float weight, float height, FILE *file);
57
58 int main() {
59     Food1 foods1[] = {
60         {"Plain Water", 0, 1000.0},
61         [{"4 Piece Red Wheat Rooti", 240, 200.0}, {"2 Piece Nun Rooti", 240, 200.0}],
62         {"Apple slices with peanut butter", 150, 200.0},
63         {"2 Piece Banana", 210, 240.0},
64         {"Cucumber salad", 40, 150.0},
65         {"Boiled Egg", 54, 48.0},
66         {"Cabbage / Tomatoes/ Gourd", 43, 100.0},
67         {"Khichuri (Rice and Lentil Porridge)", 300, 200.0},
68         {"Paratha with Egg Curry", 350, 150.0},
69         {"Khichuri (Rice and Lentil Porridge)", 300, 200.0},
70         {"Panta Bhat(Fermented Rice) with Fried Hilsa Fish", 400, 250.0},
71         {"Chapati with Mixed Vegetable Curry", 250, 180.0},
72         {"Chirer Polao (Flattened Rice Pilaf)", 270, 200.0},
73         {"Aloo Bhaji(Potato Fry) with Roti", 300, 190.0},
74         {"Shemai (Vermicelli Pudding)", 280, 150.0},
75     };
76 }
```

ads\SE133 Project>

## 1.Main\_function

## 2. HealthyFoodSuggetionWithCalorieIntake



# Reference

1. [https://www.jmeiners.com/efficient-programming-with-components/13\\_searching.html#:~:text=History%20of%20binary%20search,t%20remember%20people%20like%20that.](https://www.jmeiners.com/efficient-programming-with-components/13_searching.html#:~:text=History%20of%20binary%20search,t%20remember%20people%20like%20that.)
2. <https://chatgpt.com/?oai-dm=1>
3. <https://www.javatpoint.com/data-structure-tutorial>
4. <https://www.programiz.com/dsa>
5. <https://www.ittefaq.com.bd/305428/>
6. <https://bangla.hindustantimes.com/pictures/best-dinner-options-as-per-ayurveda-rules-31655699168422.html>



# Thank You

**Our Presentation link:**

[Capstone Project Presentation by Afzal Hossain Raju 0242310005341031 on Prezi](#)