BXSAVARAJESWARI GROUP OF INSTITUTIONS

Ballari Institute of Technology & Management

AUTONOMOUS INSTITUTE UNDER VISVESVARAYA ITCHNOLOGICAL UNIVERSITYJNANA SANGAMA,

RELAGAVI 590018

INTERNSHIP Report On

SPORTS NUTRITION DATABASE

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

CSE IN DATA SCIENCE

Submitted by

J.Jayalakshmi

3BR23CD030

Internship Carried Out By

EZ TRAININGS & TECHNOLOGIES PVT.LTD

HYDERABAD

|  |  |
| --- | --- |
| Internal Guide | External Guide |
| Mrs. Parvathi | Vishal Kumar |
| Asst. prof .CSE-DS | Technical Trainer |

Mrs. Kavyashree

Asst. prof, CSE-DS

BALLARI INSTITUTE OF TECHNOLOGY &MANAGEMENT

NACC Accredited Institution

(Recognized by Govt. o f Karnataka, approved by AIC"IE New Delhi & Affiliated to

Visvesvaraya Technological University, Belagavi„)

"Jnana Gangotri" Campus, No.87Y2, Ballari- Hospet Road, Allipur,

Ballar1-583 104 (Karnataka) (India)

Ph: 08392 - 237100 / 237190, Fax: 08392 - 237197

2024-25

BXSAVARA.JESWARI GROUP OF INSTITUI'IONS

# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITY.JNANA SANGAMA,

BELAGAVI 590018

NACC Accredited Institution\*

(Recognized by Govt. of Karnataka, approved by AICTL New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

' Jnana Gangotri ' Campus,N0.873/2,Ballari - Hospet Road, Allipur,

Ballar1-583 104 (Karnataka) ( India)

Ph: 08392 - 237100 / 237190, Fax: 08392 -237197

DEPARTMENT OF DATA SCIENCE

# CERTIFICATE

This is to certify that the Internship entitled “SPORTS NUTRITION DATABASE” has been successfully completed by J.Jayalakshmi bearing USN 3BR23CD030a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirements for the Bachelor's Degree in CSE-DS of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Be1agavi during

|  |  |
| --- | --- |
| Signature of Internship Coordinator | Signature Of HOD  Dr. Aradhana.D |
| Mrs. Parvathi | HOD of Data |
| Asst. prof. CSE-DS | Science |
| Mrs. Kavyashree  Asst. prof .CSE-DS | Department |

the academic year 2024-2025

## DECLARATION

I,J.Jayalakshmi , second year student of Computer Science and Engineering(Data Science), Ballari

Institute of Technology, Ballari, declare that Internship entitled" SPORTS NUTRITION DATABASE" is a part of Internship Training successfully carried out by EZ TECHNOLOGIES & TRAININGS PVT. LTD, Hyderabad at "BITM, BALLARI". This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

Date: 28/09/2024

Place : Ballari Signature of the Student

### ACKNOWLEDGEMENT

The satisfactions that a company the successful completion of my internship on "SPORTS NUTRITION DATABASE" would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator "Mrs. Parvathi ( asst. prof ,CSE-DS) , Mrs. Kavyashree ( asst. prof

CSE-DS lab)" for their noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

1 also thank Dr. Aradhana.D, H.O.D. Department of CSE -DS for extending all his valuable support and encouragement.

## Table of Contents

|  |  |  |
| --- | --- | --- |
| Chapter No. | Chapter Name | Page No. |
| 1 | Day to day activity(student diary extract) | 01-02 |
| 2 | Company Profile | 03 |
| 3 | Introduction | 04 |
| 4 | Problem Statement | 05 |
| 5 | Description | 06-07 |
| 6 | Algorithm | 08-09 |
| 7 | Source Code | 10-16 |
| 8 | Output | 17 |
| 9 | Conclusion | 18 |
| 10 | Reference | 19 |

Sports nutrition database

Introduction

1 . A sports nutrition database is a comprehensive collection of nutritional information tailored for Athletics.

2 . It serves as a vital resource for optimizing performance through dietary choices.

3 . Understanding this database can help athletes make informed decisions regarding their nutrition.

4 . In the world of sports ,a well planned nutrition strategy is essential for optimizing Athletic performance.

Problem statement:

**Sports Nutrition Database POC:**

CRUD : Nutrition data

provide\_dietary\_guidance(athlete\_id): provide dietary guidance and nutrition plans for athletes .

Track\_nutritional\_intake(intake\_data): Track and analyze Athlete’s nutritional intake.

Description:-

Classes outline:

* 1.**Nutrient**: Represents a single type of nutrient.
* 2.**Food**: Represents a food item containing various nutrients.
* 3.**Athlete**: Contains athlete-specific information.
* 4.**Nutrition** **Database**: Handles CRUD operations for nutrients and foods.
* 5.**Athlete** **Manager**: Manages athlete dietary guidance and tracking nutritional intake.

Output menu:

* 1.Create a Nutritional Database :
* 2. Read Nutritional Database:
* 3.Update Nutritional Database:
* 4.Delete Nutritional Database:

Real-life example:

1.CRUD OPERATIONS:

* **Create**:-The nutritionist adds a new food item with nutritional details: e.g:- 4g protein,22g carbs , and 120 calories.
* **Read** :- The nutritionist views the existing nutrient data to ensure that each athlete’s meal plans are well-balanced.
* **Update**:- The nutritionist updates the protein content of chicken breast after obtaining a new nutritional analysis.
* **Delete**:-Old or obsolete enteries , such as rarely used foods ,are removed from the database to keep the data current.

2.Provide dietary guidance:

**Provide\_dietary\_guidance:**

* Each athlete requires a tailored dietary plans based on their individual needs, preferences and training goals . The system provides personalized dietary guidance to optimize nutrition and performance.
* John Doe ,a weightlifter , aiming to increase his muscle mass ,receives a dietary plan high in protein . the system recommends meals such as grilled chicken, brown rice and steamed broccoli , with an emphasis on post-workout protein intake through shakes or bars.

3.Track nutritional intake:

1. The system allows athletes to log their daily food intake , helping track nutritional consumption against their dietary plans. Nutritionist can analyze this data to make informed adjustments.

Eg : John Doe logs his meals over a week, indicating that he consistently meets his protein goals but struggles to consume enough vegetables. The system highlights this trend, prompting the nutritionist to adjust his meal plan to include more colorful vegetables and suggest easy ways to incorporate them into his meals.

Algorithm

**Nutrition** **database** **management**

1. Create Nutrient objects.

- Input: name, unit, daily value.

- Output: Nutrient object.

2. Create Food objects.

- Input: name.

- Output: Food object.

3. Update Food objects with nutrient content.

- Input: food name, nutrient object, amount.

- Output: Updated Food object.

**Athlete management**

* 1. Create Athlete objects.
* - Input: ID, name, sport, weight, height, age, activity level.
* - Output: Athlete object.
* 2. Track nutritional intake for athletes.
* - Input: athlete ID, food object.
* - Output: Updated Athlete object.

**Dietary Guidance and Intake Display**

* 1. Provide dietary guidance.
* - Input: athlete ID.
* - Output: Dietary guidance string.
* 2. Calculate daily nutritional intake.
* - Input: athlete ID.
* - Output: Daily nutritional intake dictionary.
* 3. Display daily nutritional intake.
* - Input: athlete ID.
* - Output: Formatted daily nutritional intake string.

Source Code

class Nutrient:

def \_init\_(self, name, unit, daily\_value):

self.name = name

self.unit = unit

self.daily\_value = daily\_value

def repr(self):

return f"{self.name} ({self.unit}) - Daily Value: {self.daily\_value}{self.unit}"

class Food:

def \_init\_(self, name):

self.name = name

self.nutrients = {}

def add\_nutrient(self, nutrient, amount):

self.nutrients[nutrient.name] = {"amount": amount, "unit": nutrient.unit}

def get\_nutritional\_info(self):

return ', '.join([f"{nutrient\_name}: {details['amount']}{details['unit']}"

for nutrient\_name, details in self.nutrients.items()])

def repr(self):

return f"Food: {self.name} | Nutrients: {self.get\_nutritional\_info()}"

class Athlete:

def \_init\_(self, athlete\_id, name, sport, weight, height, age, activity\_level):

self.athlete\_id = athlete\_id

self.name = name

self.sport = sport

self.weight = weight

self.height = height

self.age = age

self.activity\_level = activity\_level

self.daily\_intake = []

def track\_nutritional\_intake(self, food):

self.daily\_intake.append(food)

def calculate\_daily\_nutrition(self):

daily\_totals = {}

for food in self.daily\_intake:

for nutrient, details in food.nutrients.items():

if nutrient not in daily\_totals:

daily\_totals[nutrient] = 0

daily\_totals[nutrient] += details['amount']

return daily\_totals

def repr(self):

return f"Athlete: {self.name} | Activity Level: {self.activity\_level}"

class NutritionDatabase:

def \_init\_(self):

self.nutrients = {}

self.foods = {}

def create\_nutrient(self, name, unit, daily\_value):

if name not in self.nutrients:

nutrient = Nutrient(name, unit, daily\_value)

self.nutrients[name] = nutrient

else:

print(f"Nutrient {name} already exists.")

def read\_nutrient(self, name):

return self.nutrients.get(name, f"Nutrient {name} not found.")

def update\_nutrient(self, name, unit=None, daily\_value=None):

if name in self.nutrients:

nutrient = self.nutrients[name]

if unit:

nutrient.unit = unit

if daily\_value:

nutrient.daily\_value = daily\_value

else:

print(f"Nutrient {name} not found.")

def delete\_nutrient(self, name):

if name in self.nutrients:

del self.nutrients[name]

else:

print(f"Nutrient {name} not found.")

def create\_food(self, food\_name):

if food\_name not in self.foods:

food = Food(food\_name)

self.foods[food\_name] = food

else:

print(f"Food {food\_name} already exists.")

def read\_food(self, food\_name):

return self.foods.get(food\_name, f"Food {food\_name} not found.")

def update\_food(self, food\_name, nutrient, amount):

if food\_name in self.foods:

self.foods[food\_name].add\_nutrient(nutrient, amount)

else:

print(f"Food {food\_name} not found.")

def delete\_food(self, food\_name):

if food\_name in self.foods:

del self.foods[food\_name]

else:

print(f"Food {food\_name} not found.")

class AthleteManager:

def \_init\_(self):

self.athletes = {}

def add\_athlete(self, athlete\_id, name, sport, weight, height, age, activity\_level):

if athlete\_id not in self.athletes:

athlete = Athlete(athlete\_id, name, sport, weight, height, age, activity\_level)

self.athletes[athlete\_id] = athlete

else:

print(f"Athlete with ID {athlete\_id} already exists.")

def track\_nutritional\_intake(self, athlete\_id, food):

athlete = self.athletes.get(athlete\_id)

if athlete:

athlete.track\_nutritional\_intake(food)

else:

print(f"Athlete with ID {athlete\_id} not found.")

def provide\_dietary\_guidance(self, athlete\_id):

athlete = self.athletes.get(athlete\_id)

if athlete:

if athlete.activity\_level == "high":

return f"{athlete.name}'s Dietary Guidance: High-protein, high-carbohydrate diet recommended."

elif athlete.activity\_level == "medium":

return f"{athlete.name}'s Dietary Guidance: Balanced diet recommended."

else:

return f"{athlete.name}'s Dietary Guidance: Focus on low-calorie, nutrient-dense foods."

return "Athlete not found."

def display\_athlete\_intake(self, athlete\_id):

athlete = self.athletes.get(athlete\_id)

if athlete:

daily\_totals = athlete.calculate\_daily\_nutrition()

return ', '.join([f"{nutrient}: {amount}" for nutrient, amount in daily\_totals.items()])

return "Athlete not found."

if \_name\_ == "\_main\_":

nutrition\_db = NutritionDatabase()

athlete\_manager = AthleteManager()

nutrition\_db.create\_nutrient("Protein", "g", 50)

nutrition\_db.create\_nutrient("Carbohydrates", "g", 300)

nutrition\_db.create\_nutrient("Fat", "g", 70)

nutrition\_db.create\_food("Chicken Breast")

nutrition\_db.update\_food("Chicken Breast", nutrition\_db.read\_nutrient("Protein"), 30)

nutrition\_db.create\_food("Rice")

nutrition\_db.update\_food("Rice", nutrition\_db.read\_nutrient("Carbohydrates"), 45)

athlete\_manager.add\_athlete(1, "John Doe", "weightlifting", 75, 180, 25, "high")

athlete\_manager.add\_athlete(2, "Jane Smith", "marathonrunning", 60, 165, 30, "medium")

athlete\_manager.track\_nutritional\_intake(1, nutrition\_db.read\_food("Chicken Breast"))

athlete\_manager.track\_nutritional\_intake(1, nutrition\_db.read\_food("Rice"))

print(athlete\_manager.provide\_dietary\_guidance(1))

print(athlete\_manager.provide\_dietary\_guidance(2))

print(athlete\_manager.display\_athlete\_intake(1))

Output

John Doe's Dietary Guidance: High-protein, high-carbohydrate diet recommended.

Jane Smith's Dietary Guidance: Balanced diet recommended.

Protein: 30, Carbohydrates: 45

conclusion:

* Efficiency managing and organizing nutritional data to tailor dietary plans.
* Providing personalized dietary guidance based on athlete-specific needs and goals.
* Tracking and analyzing nutritional intake, enabling dynamic adjustments to meal plans.
* This comprehensive system ensures athletes receive optimal nutrition, helping them enhance performance , support recovery , and maintain overall heath, which is critical in a competitive sports environment.

Reference:

* Google websites
* Chatgpt
* MetaAI