**POSHAN ABHIYAN**

*Theme-Based Project Report Submitted in partial fulfilment of the*

*Requirements for the completion of Mini Project-III*

**BACHELOR OF ENGINEERING**

**VI Semester**

IN

**INFORMATION TECHNOLOGY**

By

**Cheedella Samiksha 1602-20-737-037**

**R.Thirupathi 1602-20-737-031**

**Y.Ajay** **Kumar 1602-20-737-002**

**Under the guidance of**

**K.Srinivasa Chakravarthy**

**Associate Professor**



**Department of Information Technology Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31**

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030**

**Department of Information Technology**



**DECLARATION BY CANDIDATES**

We, **C.SAMIKSHA, R.THIRUPATHI, Y.AJAY KUMAR,** bearing hall ticket number,**1602-20-737-037,1602-20-737-031, 1602-20-737-002,** hereby declare that the project report entitled  **”POSHAN ABHIYAN”** under the guidance of **K.Srinivas Chakravathy**, Associate Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the completion of the MiniProject-III, VI semester, **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report has not been submitted to any institutes.

**C.SAMIKSHA : 1602-20-737-037**

**R.THIRUPATHI : 1602-20-737-031**

**Y.AJAY KUMAR : 1602-20-737-002**

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY) HYDERABAD - 500 030**

**Department of Information Technology**



**BONAFIDE CERTIFICATE**

This is to certify that the project entitled “**POSHAN ABHIYAN**” being submitted by **C.Samiksha, R.Thirupathi, Y.Ajay Kumar** bearing **1602-20-737-037, 1602-20-737-031, 1602-20-737-002,** in partial fulfillment of the requirements for the completion of MINI PROJECT-III, VI Semester of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

|  |  |  |
| --- | --- | --- |
| **Signature of the guide** | **Signature of the external examiner** | **Signature of the HOD** |
| **K.Srinivasa Chakravarthy** |  | **Dr. K. Ram Mohan Rao** |
| **Associate Professor** |  | **HOD, IT** |

## ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to K.Srinivasa Chakravarthy Associate Professor under whom we executed this project. We are grateful to her guidance, and constructive suggestions that helped us in the preparation of this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. We would like to thank all faculty members and staff of the Department of Information Technology for their generous help in various ways for the completion of this project. Finally, yet importantly, We would like to express our heartfelt thanks to our Head of the Department **Dr. K. Ram Mohan Rao** Sir.

### TABLE OF CONTENTS:

|  |  |
| --- | --- |
| **ACKNOWLEDGEMENT ABSTRACT**  **LIST OF FIGURES**  **1. INTRODUCTION** |  |

* 1. OVERVIEW
  2. PROBLEM STATEMENT
  3. MOTIVATION
  4. AIM OF THE PROJECT

1. LITERATURE SURVEY
2. SOFTWARE REQUIREMENTS
3. PROPOSED WORK
   1. ARCHITECTURE
   2. IMPLEMENTATION
4. RESULTS
5. CONCLUSION AND FUTURE WORK
6. REFERENCES

**ABSTRACT**

* While india’s malnutrition rates have dropped dramatically,the country is still home to the largest number of children who are still striving for food.
* India’s malnutrition rates have dropped dramatically, the govt. of India has approved the centrally sponsored scheme ‘Pradhan mantra Poshan Shakti Nirman(PM POSHAN)’ for providing one hot cooked meal in government and government aided schools from 2021-22 to 2025-26.
* The primary goal of our project is to monitor if the necessary number of calories have been delivered to students in schools. We will also monitor each student's attendance depending on the meals that have been provided for them.
* This AI-based tracker captures a picture of the food being consumed and determines how many calories are in it. If the required number of calories are consumed, the child's attendance will be recorded.

**CHAPTER 1 INTRODUCTION**

**What is a Poshan Abhiyan?**

PM POSHAN is a mid-day meal programme that provides kids in classes 1 through 8 in all public and government-aided schools with one hot, wholesome meal. From 2022 to 2023, students from the Balvatika pre-primary class will also be involved. During the school day, these meals are served once daily. Each kid is required to consume 450 calories and 12 grammes of protein at the primary level and 700 calories and 20 grammes of protein at the higher primary level each day. This initiative has a beneficial effect on keeping pupils in schools, according to independent review. Each child's participation in school and use of POSHAN must be monitored, as well as how this affects each child's physical health and overall wellbeing.

* 1. **OVERVIEW**

The focus of Abhiyaan is to lay emphasis on nutritional status of adolescent girls, pregnant women, lactating mothers and children from 0-6 years age.

**1.2 PROBLEM STATEMENT**

PM POSHAN is a mid-day meal programme for giving one hot cooked and nutritious meal to students of class 1 to 8 in all Government and Government-aided schools. Students from pre-primary class (Balvatika) will also be included from the year 2022-23. These meals are given once every day in school during school days. It is mandated that each child should get 450 calories and 12 grams protein at primary level and 700 calories and 20 grams protein at upper primary level per day. Third party evaluation has shown that this programme has a positive impact in retention of students in schools We need to track each child attendance and taking of POSHAN and the consequent impact on physical health and well-being as this has a direct co-relation to cognitive growth of the child We would also like to know the weekly Menu of each school, so that we are clear that the requisite calories are being served to each child We will perhaps need random photographs of meals being served The tracking at school, district, State and national level will be needed with the help of AI based tracker App-based solution.

**1.3 MOTIVATION**

India’s malnutrition rates have dropped dramatically, the govt. of India has approved the centrally sponsored scheme. The main aim is to generate an application that ensures that each student gets the required amount of nutrients. It aims at helping the government aided schools to manage the food served to each student according to each students nutrition factors. Poshan abhiyan helps to the government to track the meals served for the students.

**1.4 AIM OF THE PROJECT**

The application provides logins for students and the respective in-charges.

This helps to identify the food items and also to calculate the calories present in the image of food item.

The attendance will be given to students after meals are served to them.

The minimum prerequisite calories has in order for each student to have proper nutrition.

A feedback form regarding the food items which were served is provided, which helps make relative changes in the future.

**CHAPTER 2 LITERATURE SURVEY –**

* **Estimation Using Various Models of Machine Learning**

For the last few decades, it has been the popular trend in China that We created a few image-based calorie estimation models based on the food photographs and previously established calorie data in the hopes that they would correctly recognise the names of Chinese and Western dishes, provide information on their calorie content and recipe, and, lastly, make suggestions for meal plans for various groups. People are putting more attention on improving their healthiness and regulating calorie intake for every meal, so that we build a model for calorie estimation of Chinese food. In an attempt to express our concerns on this issue, and with our great interests, we used object detection to estimate the calories count of some famous Chinese dishes as well as that of Western dishes. Based on the food images and previously defined calorie data, we built some image-based calorie estimation models, which we hoped can accurately identify the name of the Chinese and Western foods and provide their calorie intake and recipe, and finally offer meal plan advice for different groups of people. people are putting more attention on improving their healthiness and regulating calorie intake for every meal, so that we build a model for calorie estimation of Chinese food. In an attempt to express our concerns on this issue, and with our great interests, we used object detection to estimate the calories count of some famous Chinese dishes as well as that of Western dishes. Based on the food images and previously defined calorie data, we built some image-based calorie estimation models, which we hoped can accurately identify the name of the Chinese and Western foods and provide their calorie intake and recipe, and finally offer meal plan advice for different groups of people. We used the SSD to identify the dish (Single Shot MultiBox Detector) for processing object categorization and detection in real-time. Additionally, we employed a software programme named "labelImg" to Our dishes are individually labelled with their corresponding original dish titles. In comparison to their traditional ways, customers may quickly and simply compute the calorie consumption of their selected items by capturing images using our models.

* **Automatic Attendance System using CNN -based Face Recognition**

We are living in the 21 st century which is the era of modern technology. Many traditional problems are being solved using new innovative technologies. Taking attendance daily is an indispensable part of educational institutions as well as offices. It is both exhausting and time consuming if done manually. Biometric attendance systems through voice, iris, and fingerprint recognition require complex and expensive hardware support. An auto attendance system using face recognition, which is another biometric trait, can resolve all these problems. This paper represents the development of a face recognition based automatic student attendance system using Convolutional Neural Networks which includes data entry, dataset training, face recognition and attendance entry. The system can detect and recognize multiple person's face from video stream and automatically record daily attendance. The proposed system achieved an average recognition accuracy of about 92 %. Using this system, daily attendance can be recorded effortlessly avoiding the risk of human error.

* **Attendance System based on face recognition using Har-cascade and SVM**

Automated Face Recognition technology (AFR) has led to numerous advancements in the ever-changing world. Smart Attendance utilizes real-time Face Recognition as an actual-world solution that is integrated with everyday activities that help manage students' attendance. For this Automatic attendance system. LBPH and Haar-cascade are used to recognize and detect the image with low-rate error accuracy and for fast calculation of the image recognition. In this process the images were captured from the surveillance and recognized with low error rate.

* **A Study of Calorie Estimation in Pictures of Food**

Software designed to accurately estimate food calories from still images could help users and health professionals identify dietary patterns and food choices associated with health and health risks more effectively. However, calorie estimation from images is difficult, and no publicly available software can do so accurately while minimizing the burden associated with data collection and analysis. Objective: The aim of this study was to determine the accuracy of crowdsourced annotations of calorie content in food images and to identify and quantify sources of bias and noise as a function of respondent characteristics and food qualities

**CHAPTER 3 SOFTWATE REQUIREMENTS –**

**Front-end :**

**HTML**

HTML is the standard markup language used for creating the structure and content of web pages. It provides a set of predefined elements and tags that define the different components of a web page, such as headings, paragraphs, images, links, tables, forms, and more. HTML uses a hierarchical structure of elements, where nested elements define the relationships and organization of the content. By using HTML, web developers can define the structure and semantics of a web page, enabling browsers to interpret and render the content correctly.

**CSS**

CSS is a style sheet language used to describe the presentation and visual appearance of HTML documents. It provides a set of rules, selectors, and properties that define how HTML elements should be displayed on a web page. With CSS, you can control various aspects of a web page's layout, typography, colors, spacing, and more. CSS works by associating styles with HTML elements or classes, allowing you to create consistent and visually appealing designs across multiple web pages. It also supports inheritance and cascading, where styles can be applied globally or selectively overridden for specific elements.

By using CSS, you can customize the appearance of HTML elements, change colors, adjust layouts, add animations, and create responsive designs that adapt to different screen sizes. CSS provides flexibility and separation of concerns, allowing you to modify the visual aspects of a web page without altering its underlying structure.

**Back-end :**

**Flask**

Flask is a popular Python web framework that allows you to build web applications and APIs. It is lightweight, flexible, and easy to use, making it a popular choice for developing web-based projects. Flask follows the model-view-controller (MVC) architectural pattern, where you define routes, views, and templates to handle different HTTP requests and generate dynamic content. It also integrates well with other Python libraries and tools, making it versatile for various web development needs. Flask can be used to create a web application or API that incorporates YOLOv8 for object detection tasks.

Flask handles the routing, request handling, and response generation, while YOLOv8 takes care of the object detection process.

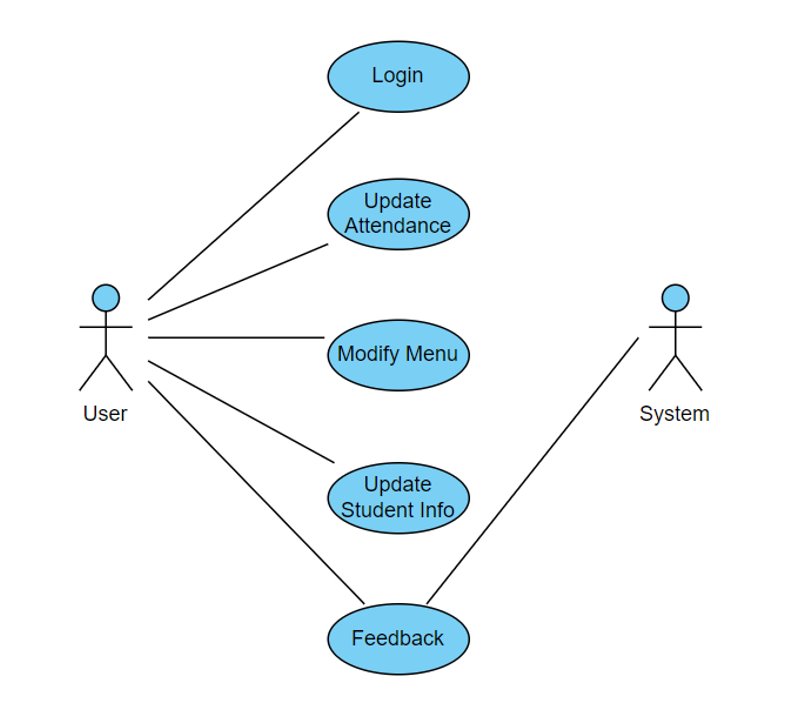
**YOLOv8**

YOLOv8 is an object detection algorithm that stands for "You Only Look Once version 8." It is part of the YOLO (You Only Look Once) family of models, which are known for their real-time object detection capabilities. YOLOv8 builds upon its predecessors (YOLOv1, YOLOv2, YOLOv3) and incorporates improvements in terms of accuracy and speed. It is based on a deep convolutional neural network (CNN) and utilizes a single pass over the input image to predict bounding boxes and class probabilities for detected objects. To integrate Flask and YOLOv8, you would need to develop the necessary endpoints or routes in Flask to handle user requests and serve the YOLOv8 model's predictions. This typically involves receiving the image data, preprocessing it (e.g., resizing, normalization), passing it through the YOLOv8 model, post-processing the model's output (e.g., filtering detections, applying non-maximum suppression), and returning the results to the user.

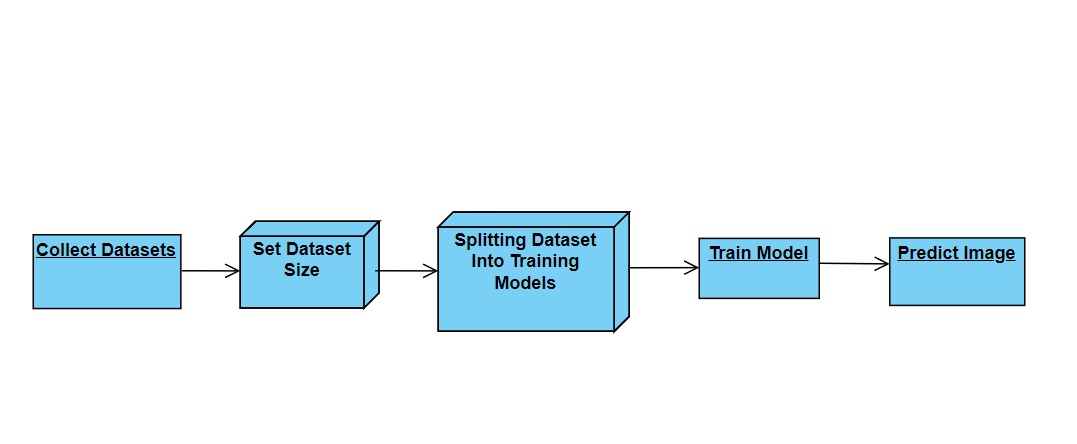
**CHAPTER 4 PROPOSED WORK –**

**4.1 :**

**Use Case Diagram**



**Design Flow Diagram**



**4.2 : Implementation**

**App.py**

from fileinput import filename

from tabnanny import filename\_only

from flask import Flask,flash, render\_template, send\_file, redirect, url\_for, request, Response

from data import data

import urllib.request

from stu\_data import stu\_data

from werkzeug.utils import secure\_filename, send\_from\_directory

from classdata import classdata

from PIL import Image

from keras\_preprocessing.image import load\_img,img\_to\_array

import numpy as np

from keras.models import load\_model

import os

import urllib

import torch

import ultralytics

from ultralytics import YOLO

import torchvision.transforms as transforms

from torchvision import transforms

#import argparse

import io

#import datetime

import cv2

import tensorflow as tf

#from re import DEBUG, sub

import subprocess

from subprocess import Popen

import re

import requests

import shutil

import time

import glob

os.environ['TF\_CPP\_MIN\_LOG\_LEVEL'] = '2'

#model = load\_model(r'C:/Users/Samiksha/Desktop/PM-Poshan-main/yolov8n.pt')

#model = torch.load(r'C:/Users/Samiksha/Desktop/PM-Poshan-main/best.pt')

model = YOLO('best.pt')

labels = {0: 'alooparatha', 1: 'rasgulla', 2: 'biryani',3: 'chickentikka', 4: 'palakpaneer', 5: 'poha',6: 'khichdi',7: 'omelette', 8: 'plainrice', 9: 'chapati'}

def processed\_img(img\_path):

img = cv2.imread(img\_path)

frame = cv2.imencode('.jpg', cv2.UMat(img))[1].tobytes()

image = Image.open(io.BytesIO(frame))

model = YOLO('best.pt')

res = model.predict(image)

a = res[0].boxes.cls

print(a)

s = []

c = 0

for i in a:

print(int(i.item()))

x = int(i.item())

if(x == 0):

s.append('alooparatha')

c += 177

if(x == 1):

s.append('rasgulla')

c += 106

if(x == 2):

s.append('biryani')

c += 360

if(x == 3):

s.append('chickentikka')

c += 260

if(x == 4):

s.append('palakpaneer')

c += 83.86

if(x == 5):

s.append('poha')

c += 110

if(x == 6):

s.append('khichdi')

c += 120

if(x == 7):

s.append('omelette')

c += 154

if(x == 8):

s.append('plainrice')

c += 130

if(x == 9):

s.append('chapati')

c += 297

s = list(set(s))

s2 = ""

for i in range(len(s)):

s2 += s[i]

s2 += ", "

s2 = s2[:len(s2)-1]

return s2.capitalize(),c

def processed\_face(img\_path):

img = cv2.imread(img\_path)

frame = cv2.imencode('.jpg', cv2.UMat(img))[1].tobytes()

image = Image.open(io.BytesIO(frame))

model = YOLO('new\_best.pt')

res = model.predict(image)

a = res[0].boxes.cls

print(a)

s = []

for i in a:

print(int(i.item()))

x = int(i.item())

if(x == 0):

s.append('Chris Evans')

if(x == 1):

s.append('Evangeline Lilly')

if(x == 2):

s.append('Robert Downey Jr')

if(x == 3):

s.append('Scarlett Johansson')

if(x == 4):

s.append('Tom Holland')

print(s)

return s

UPLOAD\_FOLDER = 'static/uploads/'

app = Flask(\_\_name\_\_)

app.secret\_key = "secret key"

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

app.config['MAX\_CONTENT\_LENGTH'] = 16 \* 1024 \* 1024

file\_arr = []

ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

@app.route("/")

def home():

return render\_template("home.html")

@app.route("/student\_dash", methods=["POST", "GET"])

def student\_dash():

if request.method == "POST":

print(request.form)

return render\_template("student/student\_success.html")

else:

return render\_template("student/student.html")

@app.route("/get\_class\_data")

def get\_class\_data():

return classdata

@app.route("/performance/get\_stu\_data")

def get\_stu\_data():

return stu\_data

@app.route("/performance/<school>/get\_school\_graph\_data")

def school\_wise\_graph(school):

data = stu\_data[school]

return {

"girls": {

"height": data[0],

"weight": data[1],

},

"boys": {"height": data[2], "weight": data[3]},

}

@app.route("/school\_dash")

def school\_dash():

return render\_template("school/school\_dash.html")

@app.route("/upload\_menu", methods=["POST", "GET"])

def upload\_menu():

if request.method == "POST":

print(request.form)

return render\_template("school/upload\_menu\_success.html")

else:

return render\_template("school/upload\_menu.html")

@app.route("/reg\_student", methods=["POST", "GET"])

def reg\_student():

if request.method == "POST":

print(request.form)

return render\_template("school/reg\_student\_success.html")

else:

return render\_template("school/reg\_student.html")

@app.route("/upload\_hdata", methods=["POST", "GET"])

def upload\_data():

if request.method == "POST":

print(request.form)

return render\_template("school/upload\_hdata\_success.html")

else:

return render\_template("school/upload\_hdata.html")

@app.route("/upload\_attendance", methods=["POST","GET"])

def upload\_attendance():

if request.method == "POST":

if 'file1' not in request.files:

flash('No file part')

return redirect(request.url)

file1 = request.files['file1']

file2 = request.files['file2']

file3 = request.files['file3']

if file1 and allowed\_file(file1.filename):

filename1 = secure\_filename(file1.filename)

file1.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename1))

f1 = UPLOAD\_FOLDER + filename1

a, b = processed\_img(f1)

if file2 and allowed\_file(file2.filename):

filename2 = secure\_filename(file1.filename)

file2.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename2))

f2 = UPLOAD\_FOLDER + filename2

c, d = processed\_img(f2)

if file3 and allowed\_file(file3.filename):

filename3 = secure\_filename(file3.filename)

file3.save(os.path.join(app.config['UPLOAD\_FOLDER'], filename3))

f3 = UPLOAD\_FOLDER + filename3

e, f = processed\_img(f3)

return render\_template("school/upload\_attendance\_success.html",item=a,cal=b,item1=c,cal1=d,item2=e,cal2=f,total=(float(b)+float(d)+float(f)))

else:

return render\_template("school/upload\_attendance.html")

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

app.run(debug=True)

**Front End**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PM Poshan</title>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static', filename='styles/main.css') }}">

<script src="{{url\_for('static', filename='scripts/home/home.js')}}" defer></script>

<script type="text/javascript" src="//translate.google.com/translate\_a/element.js?cb=googleTranslateElementInit"></script>

</head>

<body>

<div class = "navbar">

<div>

<img src = "{{ url\_for('static', filename='assets/pmposhan.png') }}" height="45rem">

</div>

<div>

<div id="google\_translate\_element" style="margin:1em;"></div>

</div>

</div>

<img src = "{{ url\_for('static', filename='assets/thumb.jpg') }}" width="100%" height="60%">

<div class = "dashbar1">

<h1 style="color: rgb(0, 0, 255);">PM POSHAN</h1>

</div>

<div class = "dashbar2">

<div class = "loginfield">

<h3>Sign in to your account</h3>

<br>

<select style="width: 12rem; border-radius:0.5rem; border-width: 1px; height: 1.2rem;" id="dashselect">

<option value="0">School</option>

<option value="2">Student</option>

</select>

<br><br>

<label>Username:</label>

<input id = "email" type = "email" placeholder="abc@gmail.com" style="border-radius:0.5rem; border-width: 1px; height: 1.2rem;"><br><br>

<label>Password:</label>

<input id = "password" type = "password" placeholder="password" style="border-radius:0.5rem; border-width: 1px; height: 1.2rem;">

<br><br>

<button class="loginbtn" id="loginbtn">Login</button>

<button class="loginbtn" id="forgotpwdbtn">Forgot Password</button>

</div>

</div>

<br><br>

<div style="text-align: center;">

<script type="text/javascript">

function googleTranslateElementInit() {

new google.translate.TranslateElement({pageLanguage: 'en', includedLanguages: 'hi,ta,te,kn,en,ml,mr,bn,gu', layout: google.translate.TranslateElement.InlineLayout.SIMPLE}, 'google\_translate\_element');

}

</script>

</body>

</html>

**Attendance**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PM Poshan</title>

<link rel= "stylesheet" type= "text/css" href= "{{ url\_for('static', filename='styles/main.css') }}">

</head>

<body>

<div class = "navbar">

<div>

<img src = "{{ url\_for('static', filename='assets/pmposhan.png') }}" height="45rem">

</div>

<div>

<button class = "loginbtn" onclick="location.href='/'">Logout</button>

</div>

</div>

<div style="text-align:center; margin:6rem 0 14rem">

<h1>Upload Attendance</h1><br>

<p id = "current\_date"></p><br>

<h1>Upload Attendance</h1><br>

<p id="current\_date"></p><br>

<form action="/upload\_attendance" method="post" target="\_self" enctype="multipart/form-data">

<label for="imageUpload">Upload Image:</label>

<input type="file" id="imageUpload" name="imageUpload" accept="image/\*">

<br><br>

<h1 style="font-size:50px;"></h1>

<body background="{{ url\_for('static', filename='uploads/bg1.jpeg') }}">

<p><h1 align="center" style="color:white;"></h1></p>

<div class="container">

<div class="row">

<body background="{{ url\_for('static', filename='uploads/bg1.jpeg') }}">

<p ><h1 align="center" style="color:white;"></h1></p>

<div class="container">

<div class="row">

<form method="post" action="/" enctype="multipart/form-data">

<div style="display: flex; justify-content: space-between;">

<div>

<h2>Upload food item 1</h2>

<br>

<input type="file" name="file1" class="form-control" autocomplete="off" required

onchange="loadFile1(event)" />

<p><img id="output1" width="200" /></p>

<script>

var loadFile1 = function(event) {

var image1 = document.getElementById('output1');

image1.src = URL.createObjectURL(event.target.files[0]);

};

</script>

</div>

<div>

<h2>Upload food item 2</h2>

<br>

<input type="file" name="file2" class="form-control" autocomplete="off" required

onchange="loadFile2(event)" />

<p><img id="output2" width="200" /></p>

<script>

var loadFile2 = function(event) {

var image2 = document.getElementById('output2');

image2.src = URL.createObjectURL(event.target.files[0]);

};

</script>

</div>

<div>

<h2>Upload food item 3</h2>

<br>

<input type="file" name="file3" class="form-control" autocomplete="off" required

onchange="loadFile3(event)" />

<p><img id="output3" width="200" /></p>

<script>

var loadFile3 = function(event) {

var image3 = document.getElementById('output3');

image3.src = URL.createObjectURL(event.target.files[0]);

};

</script>

</div>

</div>

<br><br>

<button type="reset" class="loginbtn">Reset All Fields</button>

<button type="submit" class="loginbtn">Submit</button>

</form>

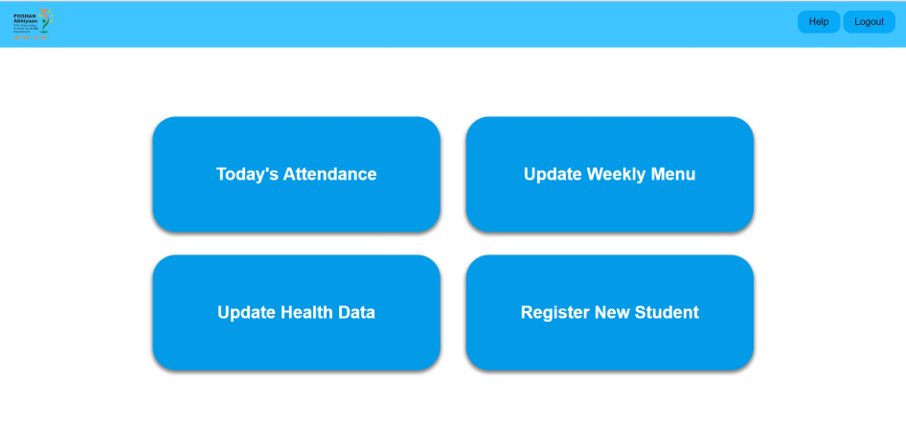
</div>

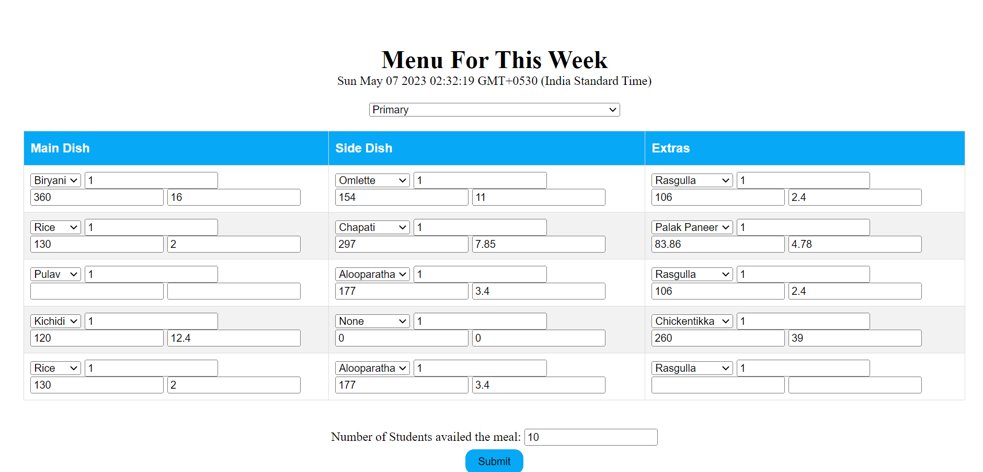
</body>

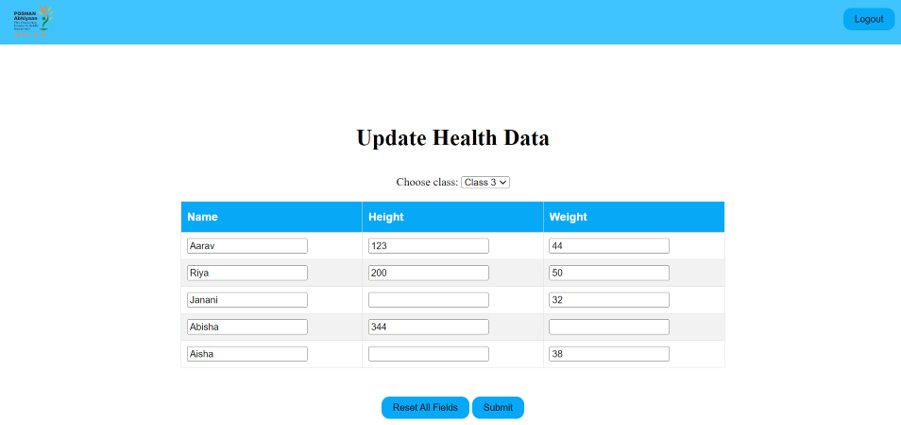
</html>

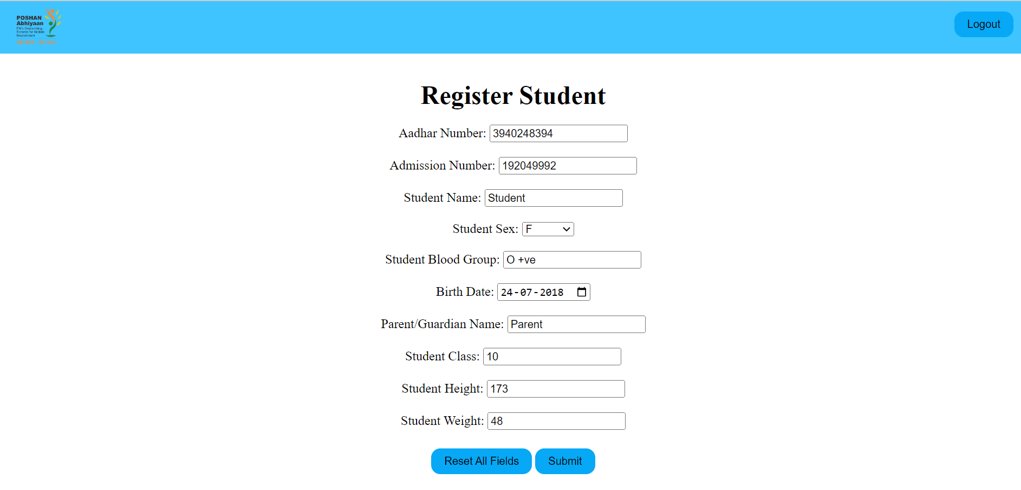
**5 RESULTS**

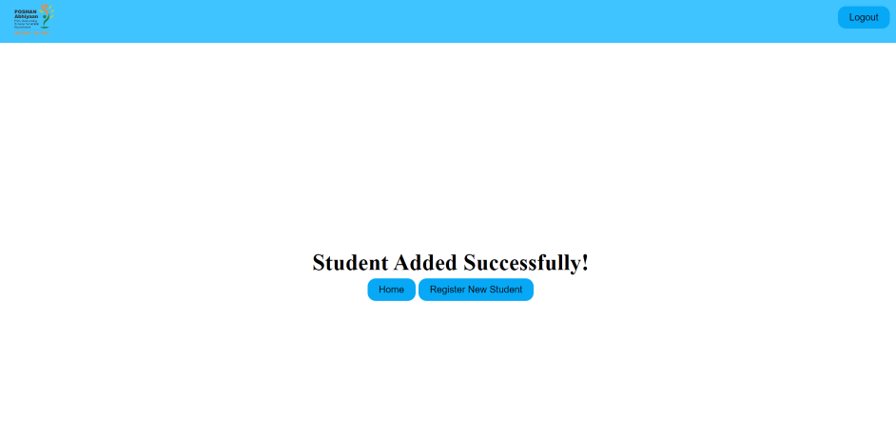


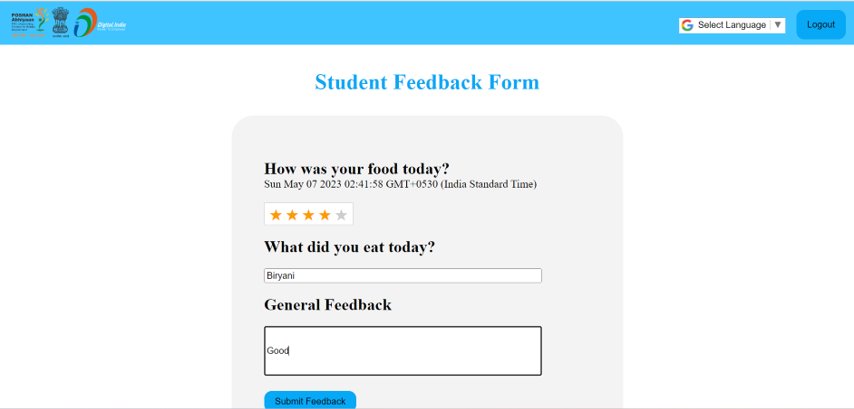


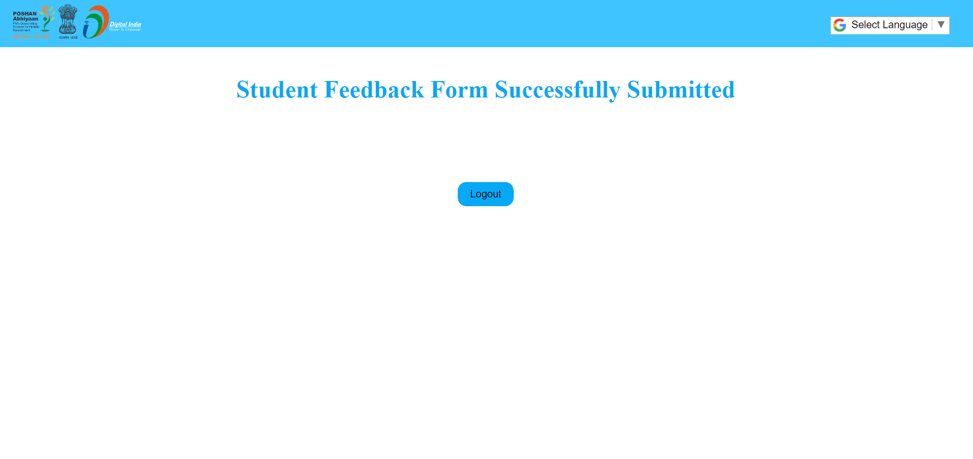




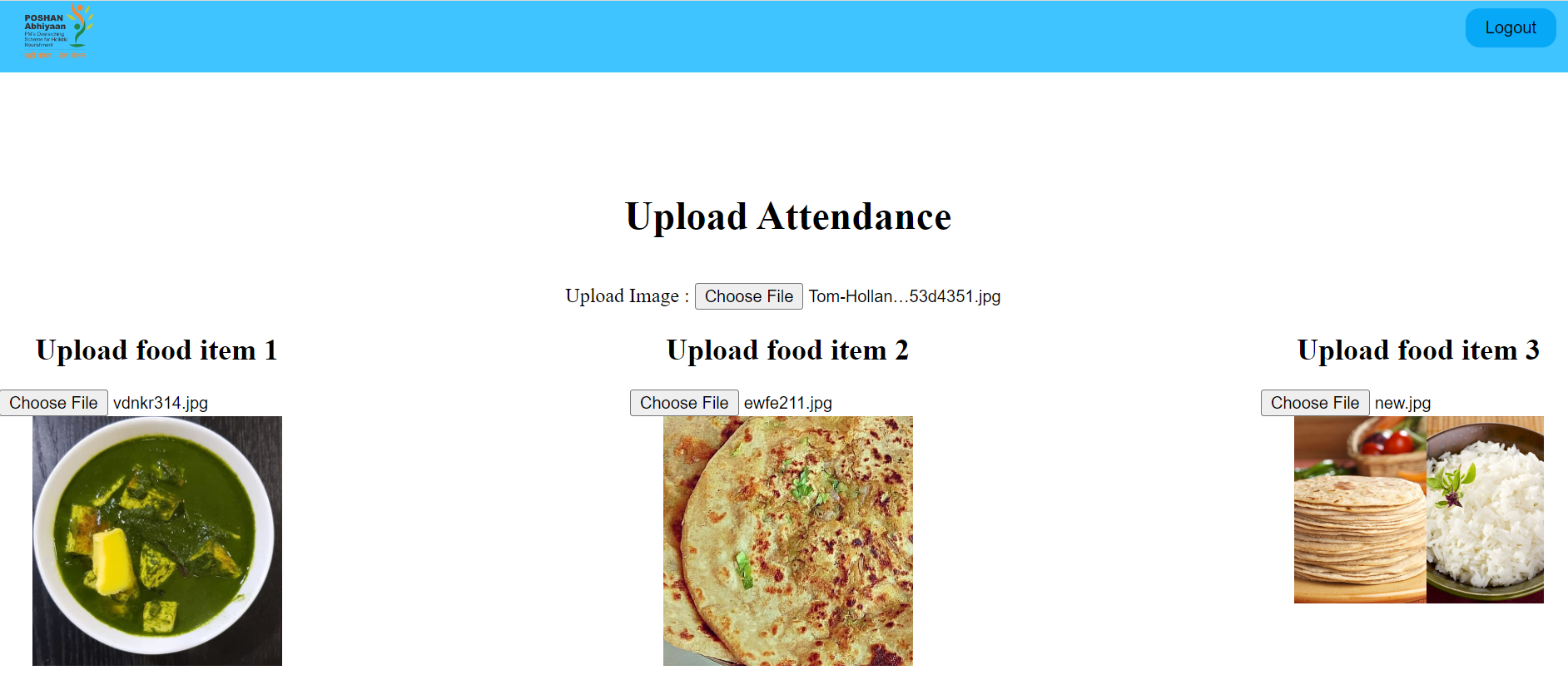


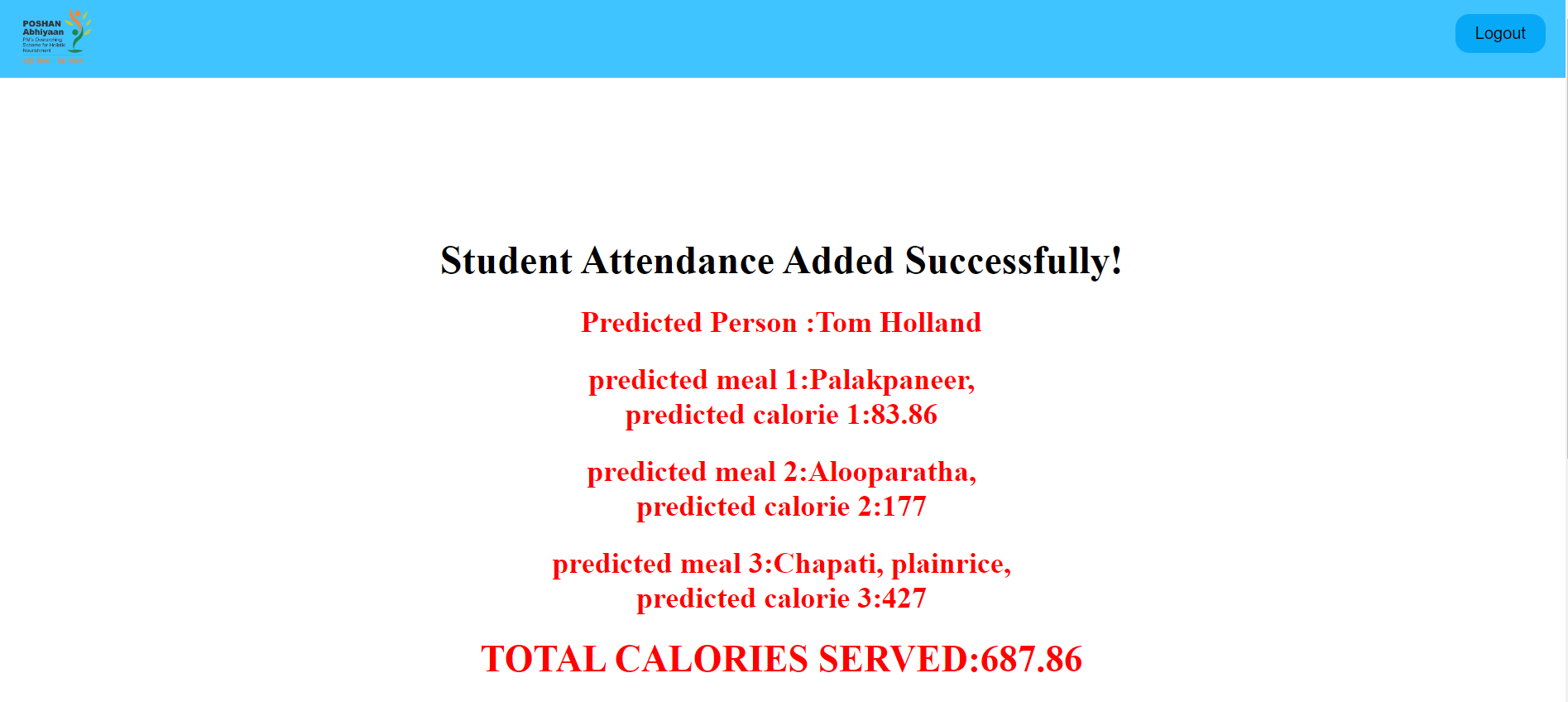


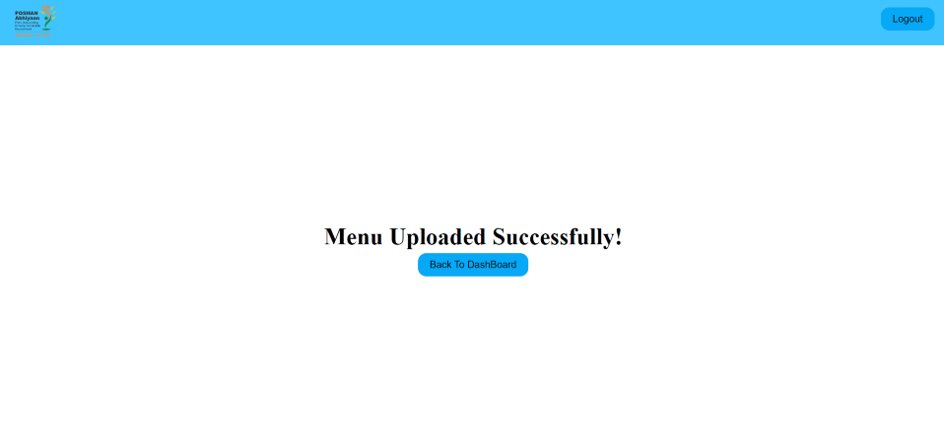


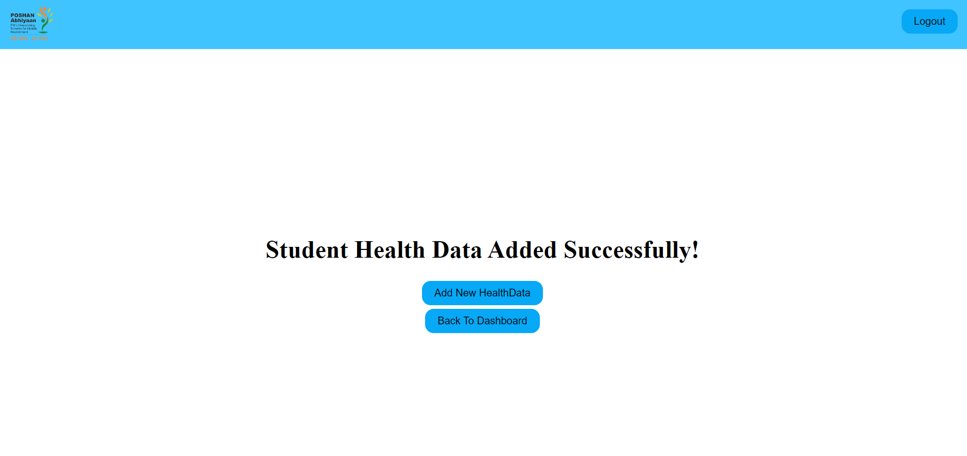












**CHAPTER 6**

**CONSLUSION**

In conclusion, the PM POSHAN programe may monitor attendance, track food distribution, and evaluate the effects on students' physical health and well-being by using this application. This programe may offer insightful information at the school, district, state, and federal levels by merging recognition and nutritional evaluation. This approach guarantees responsibility, openness, and the provision of wholesome meals to kids, supporting their general growth and development as well as their cognitive development.

**FUTURE SCOPE –**

* We will improv calorie estimation using instance segmentation, which will help calculate the precise number of calories present.
* We will include real time database consisting of real time food images and student face recognition dataset
* We will implement it as an android application

**CHAPTER 7 REFERENCES –**

* https://www.w3schools.com/html/
* https://www.w3schools.com/css/default.asp
* https://www.w3schools.com/js/default.asp
* https://www.w3schools.com/python/default.asp
* https://roboflow.com/
* https://www.kaggle.com/
* https://www.tutorialspoint.com/flask/index.htm
* https://docs.ultralytics.com/models/