

K. J. Somaiya School of Engineering
Department of Computer Engineering

Batch: C1 Roll No.: 16010122323

Experiment No : 05

Group No: 4

Title: Chapter No:05 Prototype Implementation for the Mini-Project.

Expected Outcome of Experiment:

CO3: Implement and prototype creation for the specified application.

Books/ Journals/ Websites referred:

[Students can mention websites/ books used in their project implementation]

Chapter 5

Implementation and Development of the Prototype

This chapter details the implementation process of the proposed prototype/application, outlining the development environment, tools, and technologies utilized. It describes the step-by-step approach taken to transform the conceptual design into a functional system. The implementation methodology, including coding standards, frameworks, and system architecture, is discussed in detail. Key challenges encountered during development and their corresponding solutions are also highlighted. Finally, the chapter concludes with an overview of the system's functionality and its readiness for testing and evaluation.

Introduction:

System Implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system

The implementation and prototyping document should be presented with description of following steps.

1. Modules Description:

Write input-output, properties, scenarios of important modules in the code in the given format.

Module	Name, Definition, purpose
<i>Food Image Recognition</i>	<i>Uses a deep learning model to classify food images using the Food101 dataset. Predicts the food name and corresponding calories.</i>
<i>Manual Food Entry Module</i>	<i>Allows users to input meals (e.g., "apple for breakfast") manually. This is stored and used to adjust the calorie budget.</i>
<i>Calorie Tracking System</i>	<i>Deducts calories consumed throughout the day from the user's set daily budget. Tracks macronutrients (carbs, proteins, fats).</i>
<i>Diet Plan Generator</i>	<i>Based on user data (age, gender, BMR, allergies, goals), generates meal recommendations to use remaining calories efficiently.</i>
<i>Database Management Layer</i>	<i>Handles storage and retrieval of food entries, calorie logs, diet plans, and user profiles using MongoDB.</i>

2. Integration:

Integration was done modularly with clear separation between ML logic (hosted on Google Colab/Python), frontend (React.js), backend (Node.js/Express), and database (MongoDB Atlas). Model ↔ Backend: Python-trained models are triggered via API calls from Node.js using Flask or FastAPI, with JSON payloads. Backend ↔ Frontend: Express routes serve REST APIs, which the React frontend consumes using Axios. Frontend ↔ User: Users interact via the web interface to upload images, enter meals, and view recommendations. Versioning: GitHub branches maintained for separate

features; merged into main after review.

Dependencies:

TensorFlow (image classification)
Express.js
Mongoose
Python (ML logic)
React.js (frontend)

<https://docs.google.com/spreadsheets/d/1upjOT2Gdog4xi-S2fexvRKYDyNzH-rzRmFGYjilG32E/edit?gid=1424227028#gid=1424227028>

1	food_id	A	B	C	D	E	F	G	H	I	J	K	L	M
		food_name	cuisine_type	region	country	diet_type	meal_type	serving_size_g	calories	protein_g	fat_g	carbs_g	fiber_g	
2	1	Escalivada	Spanish	European	Spain	Vegetarian	Breakfast	100	293.6	19.3	19.6	38	2.1	
3	2	Tteokbokki	Korean	Asian	South Korea	Vegetarian	Dessert	100	380.3	7.4	10.4	58.9	4.2	
4	3	Mole Poblano	Mexican	Latin American	Mexico	Non-Vegetarian	Breakfast	100	276.1	12.9	13.6	28.7	5.3	
5	4	Kibbeh Nayeh	Lebanese	Middle Eastern	Lebanon	Non-Vegetarian	Beverage	100	264	30.7	12.4	20.4	3.6	
6	5	Mushroom Tom Y Thai	Asian	Thailand		Vegetarian	Dinner	100	116.3	14.4	19.4	42.3	4.4	
7	6	Bean Tostadas	Mexican	Latin American	Mexico	Vegan	Dinner	100	128.1	7	2.6	51.4	5.3	
8	7	Salt and Pepper Chinese	Asian	China		Non-Vegetarian	Lunch	100	286.7	18.7	29.3	28.3	5.6	
9	8	Stir-Fried Green Chinese	Asian	China		Vegan	Lunch	100	120.4	4	2.7	28	6.5	
10	9	Champonones al Spanish	Spanish	European	Spain	Vegan	Dinner	100	164.6	7.4	4.3	47.1	3.7	
11	10	Boquerones en Espanol	Spanish	European	Spain	Non-Vegetarian	Beverage	100	151.4	31.3	23.6	26.9	4.6	
12	11	Coniglio alla Caciocotta	Italian	European	Italy	Non-Vegetarian	Dinner	100	365.8	27.5	15.3	6.9	1.9	
13	12	Fava	Greek	European	Greece	Vegan	Lunch	100	304.1	6.7	3.9	43.5	9.8	
14	13	Melitzanosalata	Greek	European	Greece	Vegetarian	Lunch	100	230.7	8.1	3.4	14.3	2.3	
15	14	Vegetable Tom Y Thai	Asian	Thailand		Vegetarian	Dessert	100	472.3	3.2	18.2	51.4	1.1	
16	15	Calamares a la F	Spanish	European	Spain	Non-Vegetarian	Dinner	100	382.4	31.2	21.9	31.1	4.8	
17	16	Rajas con Crema Mexican	Mexican	Latin American	Mexico	Vegetarian	Lunch	100	301.9	13.8	8.4	14.4	3.8	
18	17	Lasagna alle Vei Italian	Italian	European	Italy	Vegetarian	Lunch	100	101.7	9.1	10.1	18.9	3	

3. Implementation Details

The system was architected with a **modular full-stack approach**, integrating deep learning models with a modern web technology stack. The design ensures maintainability, scalability, and a clean separation of concerns across components.

Frontend

- **Framework:** React.js
- **Styling:** Tailwind CSS
- **Role:** The frontend serves as the **user interface layer**, enabling users to interact with the application through intuitive components and responsive design. Users can:
 - Upload food images
 - View food classifications and calorie estimates
 - Input dietary preferences
 - Receive personalized diet recommendations
- **Communication:** Uses **Axios** to interact with the backend via RESTful APIs.

Backend

- **Environment:** Node.js
 - **Framework:** Express.js
 - **API Design:** RESTful architecture
 - **Role:** The backend acts as the **middleware layer**, managing communication between the frontend, machine learning models, and database. Key responsibilities include:
 - Routing and request handling
 - Triggering ML model inferences via HTTP requests
 - Performing CRUD operations with the database
 - Ensuring secure and structured data flow
-

Machine Learning Models

Both models were developed and trained using Python on Google Colab with GPU acceleration.

Model 1: Food Image Classifier

- **Type:** Convolutional Neural Network (CNN)
- **Dataset:** Trained on the **Food-101** dataset, comprising 101 categories of common foods.
- **Output:** Predicts the most probable food class for a given image along with confidence scores.

Model 2: Diet Recommender

- **Type:** Hybrid model (Rule-based + NLP + Optimization)
 - **Logic:**
 - **Rule-Based Filtering:** Based on dietary constraints (e.g., vegetarian, vegan).
 - **KNN and Random Forest:** Created clusters based on food's nutritional profiles and categorizes it into Breakfast, Lunch and Dinner.
 - **Calorie Optimization:** Balances meals to meet caloric needs using a basic nutritional database and heuristic scoring.
 - **Seasonal Recommendations:** Recommends food suited to the season
 - **Model Serving:** Models are exposed as API endpoints using **Flask** or **FastAPI**, and are triggered from the backend using HTTP POST requests with JSON payloads.
-

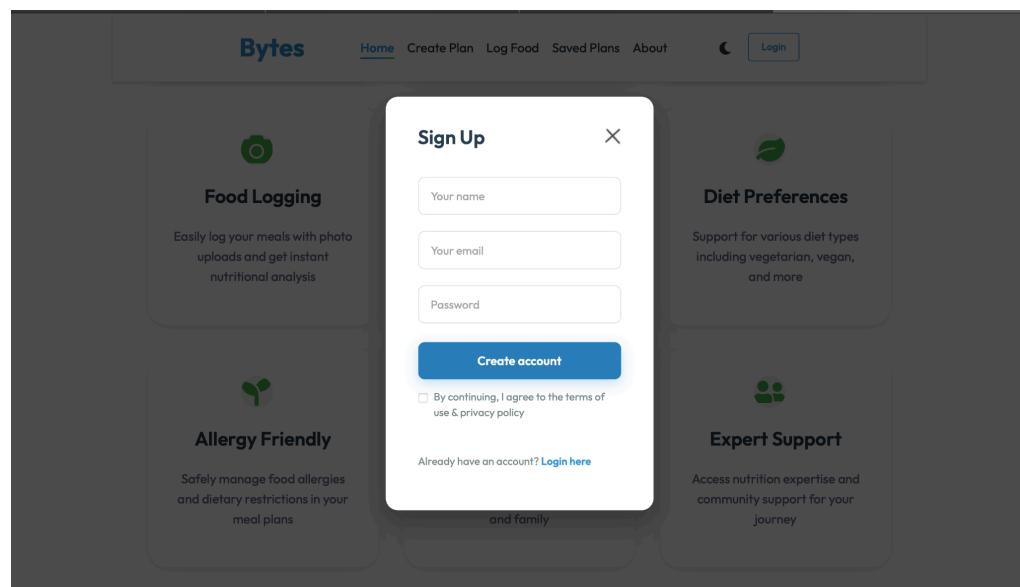
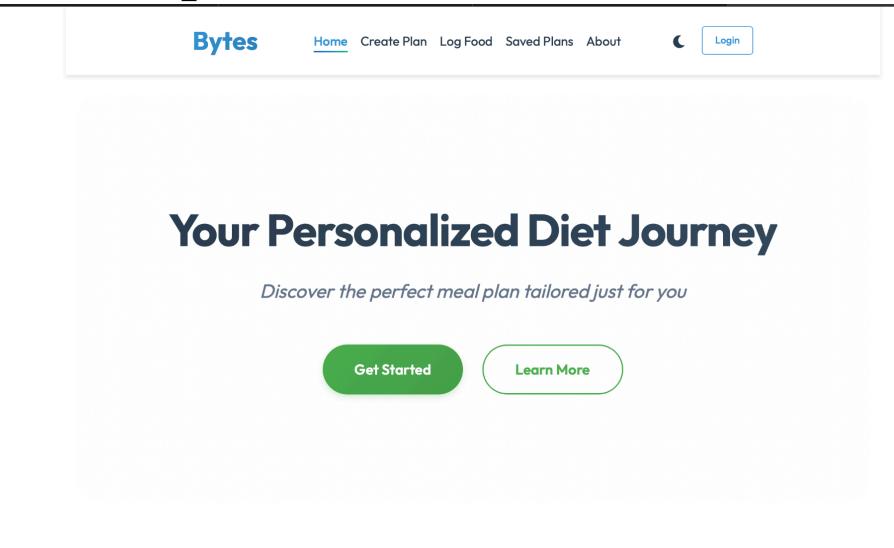
Database

- **Type:** MongoDB (hosted on MongoDB Atlas)
 - **Data Modeling:** Utilizes Mongoose for schema design and database operations.
 - **Collections:**
 - **Users:** Stores user credentials, preferences, and profile details.
 - **Meals:** Logs user-uploaded meals with classification and calorie data.
 - **Foods:** Contains nutritional values of recognized foods.
 - **DietPlans:** Stores personalized meal plans and recommendations.
-

Deployment (Planned)

3. **Frontend:** Hosted on **Vercel**, ensuring fast, scalable, and seamless delivery of the React application.
4. **Backend:** Deployed on **Render**, a cloud application platform capable of handling Node.js services.
5. **Version Control:** All source code is managed using **GitHub**, with branches created for features, bug fixes, and stable releases.
6. **Future Consideration:** Containerization with Docker and CI/CD pipelines with GitHub Actions may be integrated for production-grade deployments.

Implementation Screenshots:



K. J. Somaiya School of Engineering Department of Computer Engineering

Bytes Home [Create Plan](#) Log Food Saved Plans About  [Login](#)

Create Your Meal Plan

Let's get to know you better to personalize your plan

Progress bar: 1 / 3

- 1 Basic Info
- 2 Goals & Diet
- 3 Cuisines

Basic Information

Age  Sex 

Enter your age (in years)  Male  Female 

Weight (kg)  Height (cm) 

Enter your weight in kilograms  Enter your height in centimeters 

Activity Level 

 Sedentary  Light  Moderate  Active  Very Active

Bytes Home [Create Plan](#) Log Food Saved Plans About  [Login](#)

Basic Information

Age  Sex 

Enter your age (in years)  Male  Female 

Weight (kg)  Height (cm) 

Enter your weight in kilograms  Enter your height in centimeters 

Activity Level 

 Sedentary  Light  Moderate  Active  Very Active

[Next →](#)

Cuisine Preferences

Search cuisines...

Surprise me with a balanced variety!

Breakfast Preferences 0/14

Spanish Korean Mexican Lebanese Thai
 Chinese Italian Greek French Vietnamese
 American Japanese Indian Fruit

Lunch Preferences 0/14

Spanish Korean Mexican Lebanese Thai
 Chinese Italian Greek French Vietnamese
 American Japanese Indian Fruit

Dinner Preferences 0/14

Spanish Korean Mexican Lebanese Thai
 Chinese Italian Greek French Vietnamese
 American Japanese Indian Fruit

[← Back](#) [Generate Meal Plan](#)



Create Your Meal Plan

Let's get to know you better to personalize your plan



Goals and Diet Preferences

Your Goal i

Lose Weight Maintain Weight Build muscle mass or gain healthy weight

Gain Weight

Diet Type i

Omnivore Vegetarian Vegan

Allergies i

Type an allergy and press Enter

[← Back](#) [Next →](#)

Log Your Food

Upload a photo of your meal to get instant nutritional analysis

🍴 Log Your Food

Upload Food Image

Choose File no file selected

Mealtime (Optional)

Select mealtime

 Analyze Food

🍴 Log Your Food

Upload Food Image

Choose File Screenshot 2025-04-22 004935.png 



Mealtime (Optional)

Breakfast

 Analyze Food

Analysis Results

Food Name:	beignets	
Detected Food:	beignets	
Calories:	230 kcal	(per piece)
Type:	Measured by piece	
Protein:	15g	
Carbohydrates:	30g	
Fats:	10g	
Meal:	Breakfast	

[Log This Meal](#)
[Try Another](#)

Your Personalized Meal Plan

[Print](#)
[Save Plan](#)

Daily Targets

1848
Daily Calories

80g
Protein

266g
Carbs

51g
Fat

Profile Summary

Diet: Regular

Goal: Maintenance

Current Season: Spring

[Breakfast](#)
[Lunch](#)
[Dinner](#)
[Snacks](#)

Target Calories: 462



Vegetable Pad Thai



Gaeng Som



Tofu Tod

Breakfast [Lunch](#) [Dinner](#) [Snacks](#)

Target Calories: 462

🍴

Vegetable Pad Thai

460.6 cal 8g 54.6g 25.4g
Thai

[Click for details](#)

🍴

Gaeng Som

442.5 cal 9.5g 58.4g 11.4g
Thai

[Click for details](#)

🍴

Tofu Tod

399.1 cal 9.1g 27.9g 18.8g
Thai

[Click for details](#)

Seasonal Recommendations

These foods are in season during spring and match your dietary preferences:

Breakfast

- Pla Rad Prik
- Gai Yang
- Gaeng Pa

Lunch

- Squid Masala
- Peas Pulao
- Sorpotel

Dinner

- Yaknhi Pulao
- Fish Caldine
- Peas Pulao

Snack

- Samkeh Harra
- Bouillabaisse
- Chettinad Chicken

Bytes

[Home](#) [Create Plan](#) [Log Food](#) [Saved Plans](#) [About](#) [Login](#)

Your Saved Meal Plans



No Saved Meal Plans

You haven't saved any meal plans yet.

[Create New Meal Plan](#)

Bytes Diet Planner

Your personalized diet planning companion. Get customized meal plans based on your preferences and goals.

Quick Links

- [Home](#)
- [Create Plan](#)
- [Saved Plans](#)
- [About](#)

Resources

- [Nutrition Guide](#)
- [Healthy Recipes](#)
- [Diet Tips](#)
- [FAQ](#)

Contact Us

Email: support@bytesdiet.com
Phone: (123) 456-7890

© 2025 Bytes Diet Planner. All rights reserved.

Bytes

Home Create Plan Log Food Saved Plans [About](#) [Login](#)

Cuisine preferences for each meal type

Food allergies and sensitivities

How It Works

1 Create Your Profile
Enter your body metrics, activity level, dietary preferences, and goals.

2 Get Recommendations
Our system analyzes your data and generates personalized meal plans.

3 Track & Adjust
Save your meal plans, track your progress, and adjust as needed.

The Science Behind Our Recommendations

BMR Calculation
We use the Mifflin-St Jeor Equation to calculate your Basal Metabolic Rate (BMR), which is the number of calories your body needs to maintain basic functions at rest.

For men: BMR = $(10 \times \text{weight in kg}) + (6.25 \times \text{height in cm}) - (5 \times \text{age in years}) + 5$

For women: BMR = $(10 \times \text{weight in kg}) + (6.25 \times \text{height in cm}) - (5 \times \text{age in years}) - 161$

Macronutrient Distribution
Our system recommends an optimal balance of macronutrients based on your goals:

Protein: 1.6g per kg of body weight

Fat: 25% of total calories

Carbohydrates: Remaining calories after protein and fat

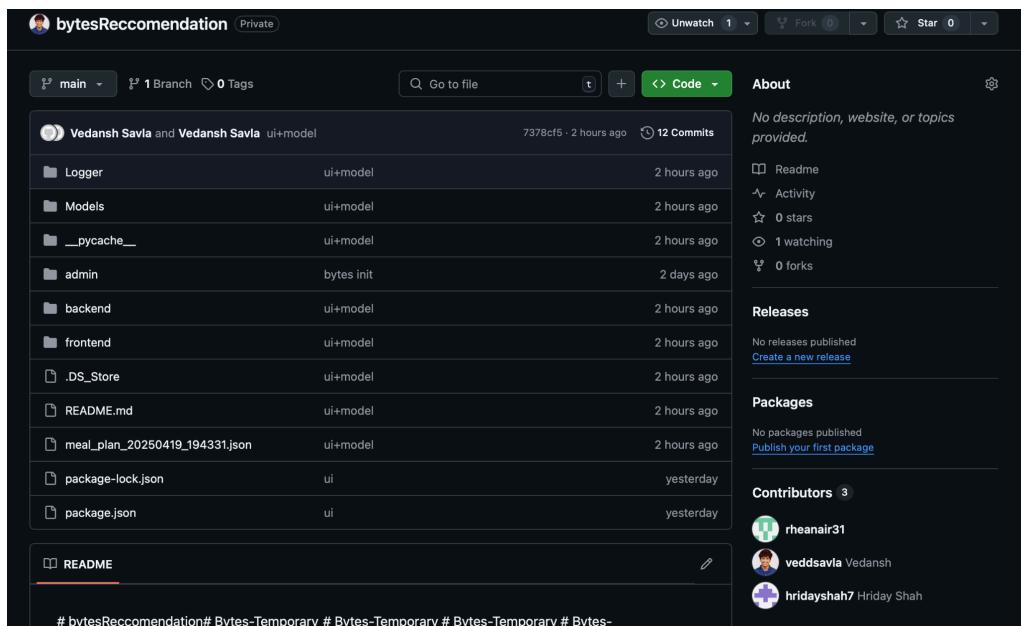
Bytes Diet Planner Your personalized diet planning companion. Get customized meal plans based on your preferences and goals.	Quick Links Home Create Plan Saved Plans About	Resources Nutrition Guide Healthy Recipes Diet Tips FAQ	Contact Us Email: support@bytesdiet.com Phone: (123) 456-7890
--	--	---	--

Github link and contributions:

<https://github.com/veddsavla/bytesReco mendumation>

https://github.com/rheanair31/Bytes_A_Diet_Recommendation_System

<https://github.com/hridayshah7/Bytes2>



 Bytes_A_Diet_Recommendation_System Public

main 1 Branch 0 Tags Go to file Add file Code

rheanair31	Initial commit	d217642 · yesterday	3 Commits
admin	bytes init	2 days ago	
backend	Initial commit	yesterday	
frontend	Initial commit	yesterday	
.DS_Store	bytes init	2 days ago	
README.md	Initial commit	2 days ago	
breakfast_predictor.pkl	bytes init	2 days ago	
diet_recommendation_training.ipynb	bytes init	2 days ago	
diet_recommender.py	bytes init	2 days ago	

 Bytes2 Public

Watch 1

main 1 Branch 0 Tags Go to file + Code

hridayshah7	Re-add model file using Git LFS	4f2b9df · 4 hours ago	5 Commits
Logger	Re-add model file using Git LFS	4 hours ago	
Models	Your commit message describing the update	6 hours ago	
__pycache__	Start from scratch	8 hours ago	
admin	Start from scratch	8 hours ago	
backend	Start from scratch	8 hours ago	
frontend	Food Recognition	5 hours ago	
.DS_Store	Start from scratch	8 hours ago	
.gitattributes	Add model file using Git LFS	5 hours ago	
.gitignore	Start from scratch	8 hours ago	
README.md	Start from scratch	8 hours ago	
meal_plan_20250419_194331.json	Start from scratch	8 hours ago	
package-lock.json	Start from scratch	8 hours ago	
package.json	Start from scratch	8 hours ago	