UNIVERSITY Of MISSOURI - KANSAS CITY

Cloud Computing

FINAL PROJECT REPORT

Authors:

Rahul Karthik Arunachalam Usharani

Raghul Selvakumar

Ramya Koganti

Rajaram Yadav

Contents

•Macronutrient breakdowns (carbs, fats, protein) 17

•The system helps users understand what they are eating and how it aligns with their goals. 17

3. Health Goal Monitoring: 17

- **Users can define daily calorie limits or macronutrient targets based on their goals. 17**
- **As meals are logged, the system calculates cumulative intake. 17**
- **The UI provides real-time progress tracking (e.g., "You've consumed 80% of your protein goal"). 17**
- **Graphs and status indicators visually show how users are progressing day by day or week by week. 17**

4. AI Chatbot Integration (OpenAI API): 17

- **A key feature is the interactive chatbot, powered by OpenAI's GPT-3.5/4 API. 17**
- **Users can: 17**
- **Log meals through conversation (e.g., "I had rice and chicken for lunch") 17**
- **Ask questions ("Is avocado healthy?") 18**
- **Get real-time advice ("What should I eat post workout?") 18**
- **Behind the scenes, the chatbot passes the user message to a Lambda function, which calls the OpenAI API. 18**
- **The response is formatted and displayed in the chat window for a seamless experience. 18**

- **The chatbot makes the platform conversational, accessible, and user-friendly, especially for users less comfortable with manual data entry. 18**

5. Personalized Recipe Recommendations: 18

- **Based on users' dietary logs and goals, the system suggests healthy recipes. 18**
- **Uses external APIs like Spoonacular or Edamam to fetch recipes filtered by: 18**
- **Caloric needs 18**
- **Macronutrient balance 18**
- **Dietary restrictions (e.g., vegan, gluten-free) 18**
- **Recommends 3–5 meals that match the user's target for the day or week. 18**
- **Helps bridge the gap between nutritional planning and actionable meal ideas. 18**

6. Data Visualization & Nutrition Insights: 18

- **Presents user data in a visually engaging format using charts, graphs, and summaries. 18**
- **Displays: 18**

Calorie trends over time, 18

Macronutrient ratios, 18

Meal timing patterns. 18

- **Helps users understand their habits and adjust their behavior accordingly. 18**
- **Built using React with UI libraries like Material UI and charting tools (e.g.,**

1. Team Introduction

The team has been working collaboratively by dividing tasks based on individual strengths and responsibilities.

- **Rahul Karthik Arunachalam Usharani**

  Role: UI & Visualization

  Developed the user interface using React and configured the project with AWS Amplify for hosting and deployment. Build and manage interactive components for user login/signup, meal tracking, goal setting and dashboard views. Implement data Visualization tools(graphs,charts) to show trends in calorie intake, nutrition distribution, and goal progress. Embed the chatbot UI widget and enable real-time interaction using API responses.

- **Raghul Selvakumar**

  Role: Back End – API, Business logic & Data Handling

  Develop and maintain AWS Lambda functions to process frontend and chatbot requests. Implement business logic for: Meal logging, Nutrition calculations, Health goal tracking, Chatbot query routing. Design and expose secure REST and GraphQL APIs via API Gateway and AppSync. Handle input validation, error handling, and asynchronous task execution within Lambda. Integrate with Amazon DynamoDB to: Design optimized, scalable table schemas for users, meals, goals, and chat logs. Use DynamoDB Indexes (GSI/LSI) for efficient querying and filtering of user data. Implement conditional writes, batch operations, and pagination for performance. Write data modeling logic to minimize read/write costs and latency. Perform unit testing and debugging using tools like Postman, SAM CLI, and CloudWatch Logs.

- **Ramya Koganti**

  Role: Configuration, Monitoring and Visualisation

  Set up and configure core AWS services: Amplify, Cognito, Lambda, DynamoDB, AppSync, and API Gateway. Manage IAM roles, policies, and service permissions to enforce secure access control. Establish user authentication flows with Amazon Cognito, including MFA and password policies. Handle Amplify deployment pipeline, domain setup, and hosting configuration. Configure GraphQL schema in AppSync and define resolvers for DynamoDB

and Lambda. Monitor system performance and logs using CloudWatch and X-Ray. Optimize infrastructure for scalability, fault tolerance, and cost-efficiency using serverless best practices. Ensure secure environment variables, secret management, and encryption of sensitive data.

- **Rajaram Yadav**

  Role: Chatbot Integration and Recipe Recommendation

  Integrate the OpenAI API (e.g., GPT-3.5/GPT-4) to enable conversational intelligence within the web application. Design and test effective system prompts to guide the model in providing nutrition-focused and accurate responses. Route user inputs through the frontend to a Lambda function that formats the payload and communicates with OpenAI. Implement context-aware prompts by retrieving user-specific data (meals, goals, habits) from DynamoDB. Format and sanitize the OpenAI response before returning it to the user. Enable chatbot actions like: Meal logging through conversation .Nutrition tips and explanations. Real-time health advice based on goals. Recipe suggestions matching dietary constraints. Apply rate limiting, error handling, and fallback responses to ensure reliability and safety. Continuously refine chatbot output using feedback loops, analytics, and user testing.

How We Worked Together

Our team adopted a collaborative approach leveraging each member's individual strengths to ensure efficient progress across all areas of the project. From the beginning, we established clear roles and responsibilities.

We held regular team meetings both virtually and in person to:

Review Progress,

Share Updates,

Resolving issues,

And plan the next stages of development

We used version control tools like GitHub to manage code contributions, enabling seamless collaboration and integration across front-end, back-end, and chatbot components.

2. Motivation and Purpose

Maintaining a healthy diet is increasingly difficult in today's busy world. People often lack the time, knowledge, or tools to make informed food choices. With rising lifestyle-related health issues, the need for smart nutrition support is urgent. This project aims to offer a personalized, tech-driven solution to promote better eating habits.

Purpose

The Smart Nutrition Assistant is designed to serve as an intelligent, interactive web application that helps users manage their dietary habits through personalized insights, real-time recommendations, and automated tracking. The purpose of this project is to enhance user engagement with their nutrition through a streamlined platform supported by a conversational chatbot, goal-based tracking, and recipe suggestions.

By leveraging AWS services such as Cognito for authentication, Lambda for serverless computation, and DynamoDB for secure data storage, the system ensures scalable, efficient, and cost-effective performance. Additionally, the chatbot module facilitates real-time interaction, allowing users to log meals, receive tailored advice, and get guidance without navigating complex interfaces.

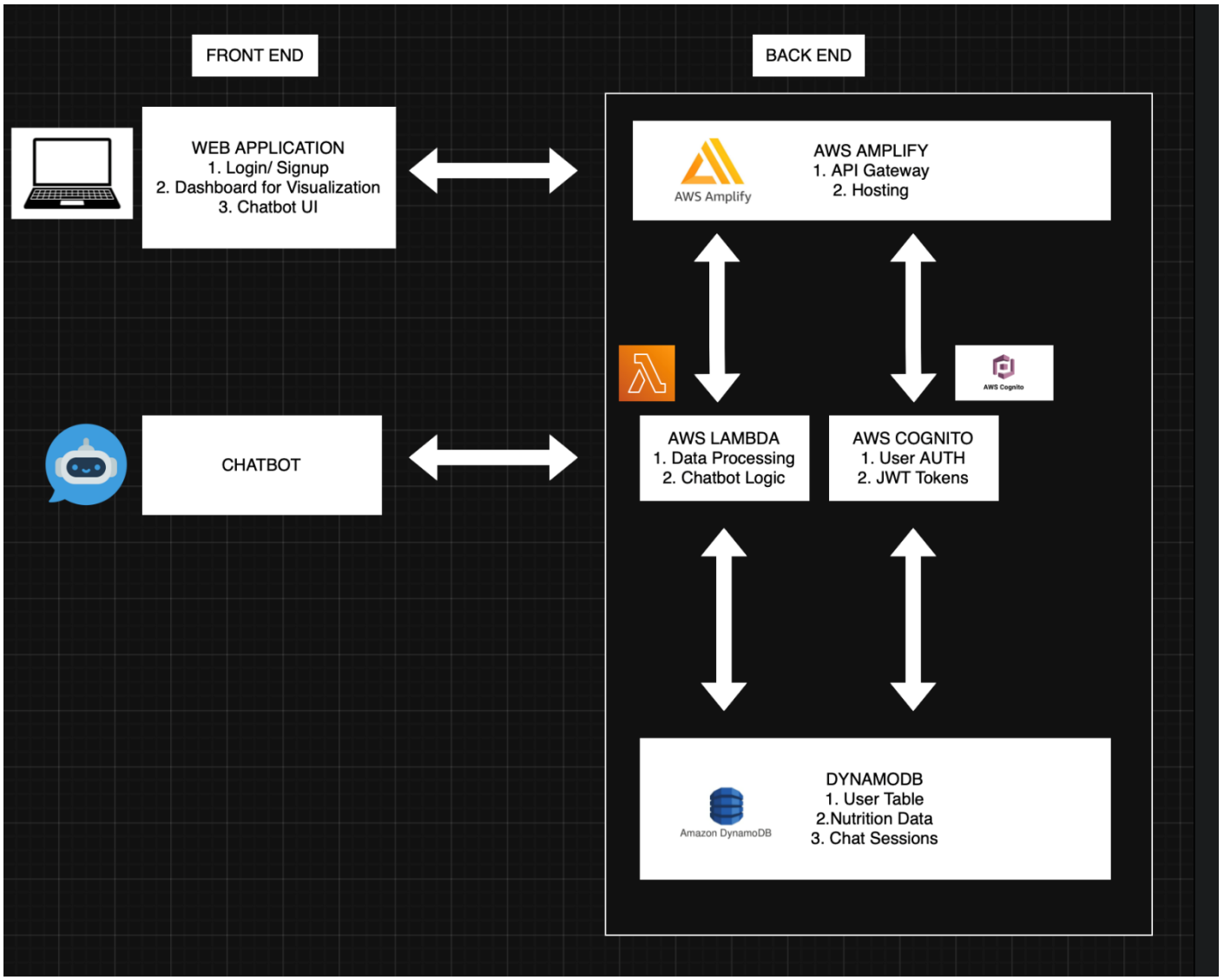
Motivation

In today's fast-paced lifestyle, maintaining a balanced and healthy diet has become increasingly difficult. Many individuals struggle to make informed nutritional decisions due to limited time, lack of dietary knowledge, and minimal access to trustworthy health resources. With rising global concerns around obesity, diabetes, and other lifestyle-related diseases, there is a growing demand for smart, accessible, and personalized digital solutions that promote healthy living.

The motivation behind this project stems from the need to bridge the gap between nutritional science and everyday food choices using modern, scalable technology. By leveraging the power of cloud computing and intelligent interfaces, we aim to offer users a reliable, real-time nutrition assistant that adapts to their goals and habits—empowering them to take control of their health with confidence.

Ultimately, this project aims to empower users with the tools and knowledge they need to make healthier food choices, develop sustainable habits, and improve their overall well-being through an intelligent, cloud-based solution.

3. System Architecture



Overview:

The Smart Nutrition Assistant is a cloud-native, serverless web application designed to support users in tracking and improving their nutritional habits. Built on top of Amazon Web Services (AWS), the system ensures scalability, high availability, and secure data handling. It integrates a responsive front-end web interface with intelligent back-end processing, real-time interaction through a chatbot, and personalized data visualization.

- **Frontend:**

  The user interacts with the application through the UI, which is responsive and provides easy navigation for logging meals, viewing recommendations, and setting goals. This component

communicates with the back-end via API requests.

- **Backend:**

  Responsible for handling requests from the front-end, processing data, and interacting with databases and external services(like recipe API's, AWS services). User authentication, meal-logging, and real time advice generation happen here.

- **Authentication:**

  Manages user authentication and authorization. It handles user registration, login, password management, and ensures secure access to personalized nutrition data.

- **Real-time chatbot:**

  Provides users with real-time interaction through a conversational chatbot. The chatbot can log meals, offer personalized nutrition advice, and recommend recipes based on user preferences.

- **Data Storage & Monitoring:**

  User data is stored securely in AWS DynamoDB for easy access, while AWS CloudWatch monitors system performance and QuickSight provides insights into user progress.

How It Works

1. **User Access & Authentication:**

- Users register and authenticate securely using AWS Cognito.

- Cognito also integrates with federated identity providers if needed.

2. **Data collection & Input:**

- User log meals and set preferences via the React.js interface, storing data in AWS DynamoDB.
- The chatbot interface enables users to input data via conversational interaction, improving ease of use.
- This data is captured at the frontend and sent back end via API calls handled by Amplify or AppSync.

3. **API handling via Amplify and AppSync:**

- AWS Amplify manages REST-based API's , handling general data exchange and configuration.
- AWS AppSync, which supports GraphQL, is used to efficiently fetch and update user data with fine-grained control.
- This dual-approach improves both performance and flexibility of data interaction.

4. **Backend Processing with Lambda:**

- AWS Lambda functions process requests triggered by API calls.
- They execute key business logic including:

  Parsing meal data

  Calculating nutrition metrics

  Processing chatbot queries

  Generating health insights and recommendations

5. **Data Storage in DynamoDB:**

- Amazon DynamoDB stores all applications data such as:

  User profiles and goals

  Meal logs and nutrition breakdowns

  Chatbot histories

- DynamoDB offers near real-time read/write speeds, ideal for responsive apps.
- The system can scale automatically to handle a growing number of users.

6. **Chatbot Interaction:**

- The chatbot is embedded in the web UI and communicates with Lambda.
- It interprets user input, fetches relevant responses, and guides users through:

  Logging meals

  Asking health related questions

  Receiving daily tips or suggestions

- The chatbot makes the app more interactive and less form-based, reducing friction.

7. **Visualization and Recommendations:**

- The frontend retrieves historical user data from DynamoDB via AppSync.
- Data is rendered using charts and dashboards(calorie intake trends, protein-carb-fat distribution).
- Personalized recipe suggestions are fetched using external API's, matched against the user's nutritional goals.

Key Technologies

AWS Cognito – Handles secure user authentication and management.

AWS Amplify – Handles deployment, API integration and manages frontend and backend connections. Simplifies configuration of hosting, authentication, and CI/CD pipelines.

React.js – Powers the front-end interface for a dynamic user experience.

Node.js – Used for backend development and API handling.

AWS AppSync – Provides a GraphQL interface to interact with DynamoDB and other services. Reduces API complexity compared to multiple REST endpoints.

AWS Lambda – Provides serverless computing for efficient processing. Ensures cost-effectiveness by running code only when triggered.

AWS DynamoDB – Stores users data and meal logs in a database. Easily integrates with Lambda and AppSync for real-time applications.

4. Features

1. **Secure User Authentication & Profile Management:**

- The system uses Amazon Cognito for secure sign-up, login, and session management.
- Each user has a personalized profile that includes:
- Dietary preferences
- Health goals (e.g., lose weight, gain muscle, maintain balance)
- Historical data (meal logs, progress)
- Cognito ensures secure access to user data and protects against unauthorized access using JWT tokens and role-based access control.

2. **Meal Logging & Food Tracking:**

•Users can manually log their daily meals, including:

•Food names, quantities, and meal types (breakfast, lunch, etc.)

•This data is stored in Amazon DynamoDB, structured by user and timestamp.

•AWS Lambda functions process the input to calculate:

•Caloric values

•Macronutrient breakdowns (carbs, fats, protein)

•The system helps users understand what they are eating and how it aligns with their goals.

3. **Health Goal Monitoring:**

- Users can define daily calorie limits or macronutrient targets based on their goals.
- As meals are logged, the system calculates cumulative intake.
- The UI provides real-time progress tracking (e.g., “You’ve consumed 80% of your protein goal”).
- Graphs and status indicators visually show how users are progressing day by day or week by week.

4. **AI Chatbot Integration (OpenAI API):**

- A key feature is the interactive chatbot, powered by OpenAI’s GPT-3.5/4 API.
- Users can:
- Log meals through conversation (e.g., “I had rice and chicken for lunch”)

- Ask questions ("Is avocado healthy?")
- Get real-time advice ("What should I eat post workout?")
- Behind the scenes, the chatbot passes the user message to a Lambda function, which calls the OpenAI API.
- The response is formatted and displayed in the chat window for a seamless experience.
- The chatbot makes the platform conversational, accessible, and user-friendly, especially for users less comfortable with manual data entry.

5. **Personalized Recipe Recommendations:**

- Based on users' dietary logs and goals, the system suggests healthy recipes.
- Uses external APIs like Spoonacular or Edamam to fetch recipes filtered by:
- Caloric needs
- Macronutrient balance
- Dietary restrictions (e.g., vegan, gluten-free)
- Recommends 3–5 meals that match the user's target for the day or week.
- Helps bridge the gap between nutritional planning and actionable meal ideas.

6. **Data Visualization & Nutrition Insights:**

- Presents user data in a visually engaging format using charts, graphs, and summaries.
- Displays:

  Calorie trends over time,

  Macronutrient ratios,

  Meal timing patterns.
- Helps users understand their habits and adjust their behavior accordingly.
- Built using React with UI libraries like Material UI and charting tools (e.g., Chart.js or Recharts).

7**. Serverless & Scalable Backend:**

- Entire backend is built using AWS Lambda (serverless functions), which:
- Processes user inputs
- Handles chatbot interactions

- Communicates with DynamoDB
- API Gateway or AWS AppSync exposes these functions securely to the frontend.
- The infrastructure scales automatically, ensuring cost efficiency and high availability, even as users increase.

8. **Real-Time & Secure Data Storage (DynamoDB):**

- Amazon DynamoDB is used to store all user data:

  Meal logs

  User profiles

  Chatbot conversations

  Goal settings

- It supports:

  Fast, low-latency queries

  Flexible, scalable storage

  Real-time updates

- Includes use of Global Secondary Indexes (GSIs) to allow complex queries, like filtering meals by date or user.

Future Enhancements:

Barcode Scanning & Image Recognition - for logging meals via camera or food labels.

Sync Devices - for automatic activity and calorie tracking(Fitbit, Apple health).

Push Notifications & Reminders - for meal logging, hydration, and health tips.

Voice Assistant Integration - Allow users to speak to the chatbot or app using voice commands. Integrate with services like Amazon Alexa, Google Assistant, or Speech-to-Text APIs for a hands-free experience.

5. Technical Merits

Serverless Architecture:

- Leveraging AWS Lambda allows backend logic to run without managing servers, reducing operational overhead.
- Automatically scales based on demand, ensuring consistent performance under variable user loads.
- Enables event-driven processing of user inputs and chatbot interactions with minimal latency.

Scalable and Resilient Cloud Infrastructure:

- Built entirely on AWS, the system benefits from global scalability, fault tolerance, and high availability.
- Uses Amazon DynamoDB for fast and reliable NoSQL data storage with seamless horizontal scaling.
- Services like API Gateway, AppSync, and Amplify ensure secure, high-throughput communication across components.

Secure User Management:

- Amazon Cognito is used for user authentication and authorization, providing multi-factor authentication (MFA), encrypted token management, and secure identity handling.
- Ensures each user's nutrition data is protected and access-controlled.

Real-Time Interaction & Data Handling:

- The use of AppSync (GraphQL API) enables fine-grained data queries and real-time updates to the frontend, improving performance and reducing data transfer overhead.
- Users can view immediate feedback and insights as they log their meals or interact with the chatbot.

Integration with AI (OpenAI API):

- Chatbot functionality is powered by OpenAI's GPT models, allowing natural language interaction and dynamic, personalized guidance.
- Serverless integration ensures secure and efficient request handling between the user, Lambda functions, and OpenAI.

Data Visualization & UX Optimization:

- The frontend, built using React.js and enhanced with Material UI and charting libraries, provides an

interactive, responsive user experience.

- Clear dashboards, meal timelines, and goal progress charts improve data interpretation and engagement.

CI/CD and Deployment with AWS Amplify:

- AWS Amplify simplifies deployment, versioning, and continuous integration.
- Ensures consistent updates and testing workflows without manual intervention.
- Also supports environment separation for development, testing, and production stages.

Cost Efficiency:

- Using a fully serverless model reduces infrastructure costs by charging only for what is used (Lambda, DynamoDB on-demand, AppSync per call).
- No need to provision or manage servers, databases, or compute resources manually.

Modularity and Extensibility:

- The architecture is modular, making it easy to:
- Add new features (e.g., reminders, fitness tracker integration)
- Integrate third-party APIs
- Expand to mobile platforms

6. Challenges and Solutions

Issue 1: Handling secure user authentication while managing login, signup, and session control was initially complex.

Solution: Integrated Amazon Cognito, which provides built-in user pool management, secure token-based authentication, and role-based access control. This offloaded security concerns and ensured compliance with best practices (e.g., encryption, password policy, and MFA support).

Issue 2: Presenting complex nutritional insights (macros, trends, goals) in a user-friendly way was challenging.

Solution: Used Material UI and charting libraries (e.g., Recharts) to create interactive, visually intuitive dashboards. Provided summary widgets and graphical breakdowns of food intake to improve user understanding and engagement.

Issue 3: As the application scaled, ensuring performance under load (especially database read/writes) and keeping costs low became a concern.

Solution: Used DynamoDB with On-Demand Capacity Mode for automatic scaling. Optimized data models using

Partition Keys and Global Secondary Indexes (GSI) to support efficient queries without performance loss.Lessons Learned

- Choosing the right tools early on such as serverless architecture and GraphQL helps avoid rework and technical debt.
- Frequent testing, both unit and user-level, helped us catch logic issues early and refine chatbot behavior based on real-world inputs.
- Clear team roles and version control (GitHub) enabled smooth collaboration and reduced conflicts during development.

7. Conclusion

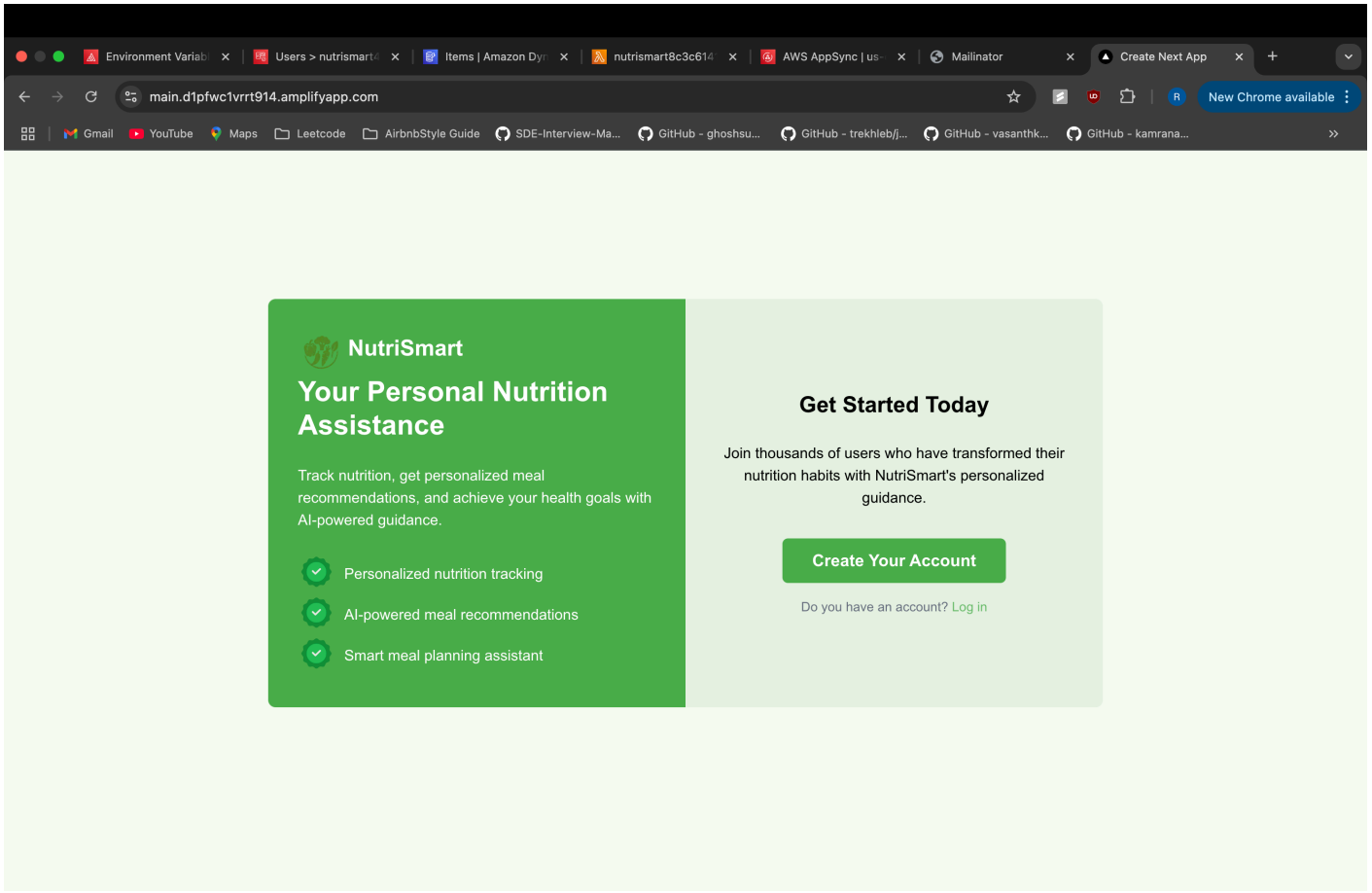
The Smart Nutrition Assistant project successfully demonstrates how modern cloud technologies, artificial intelligence, and intuitive user interfaces can be combined to promote healthier lifestyles through smart dietary tracking and personalized nutrition guidance. By leveraging AWS services such as Cognito, Lambda, DynamoDB, AppSync, and Amplify, the system ensures scalability, security, and seamless data flow from frontend to backend.

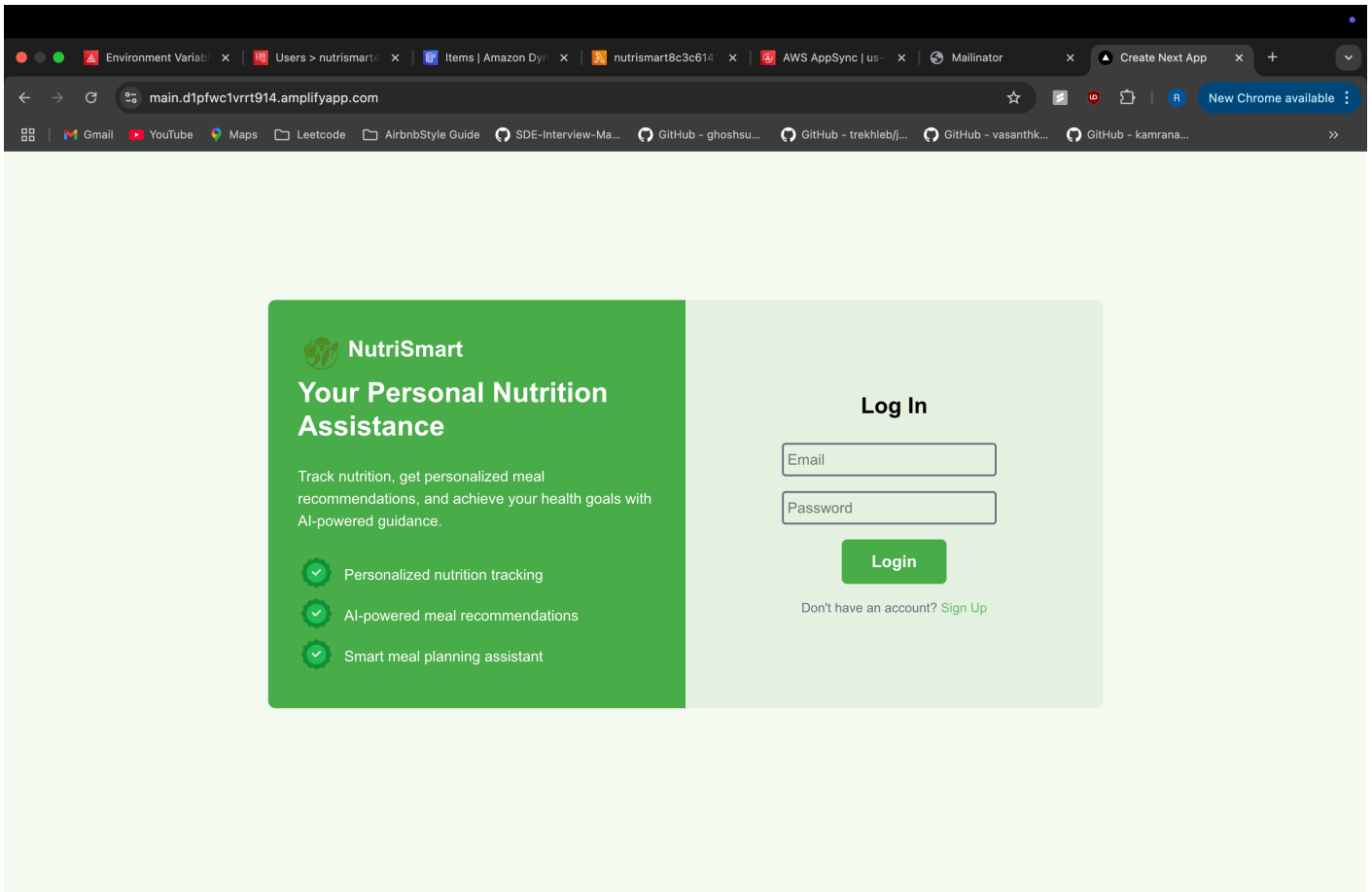
The integration of a conversational chatbot powered by OpenAI further enhances user engagement by allowing natural, real-time interactions for logging meals, receiving advice, and tracking health goals. The front-end experience, built using React.js and enhanced with visualizations, makes complex nutritional data easily understandable for users of all backgrounds.

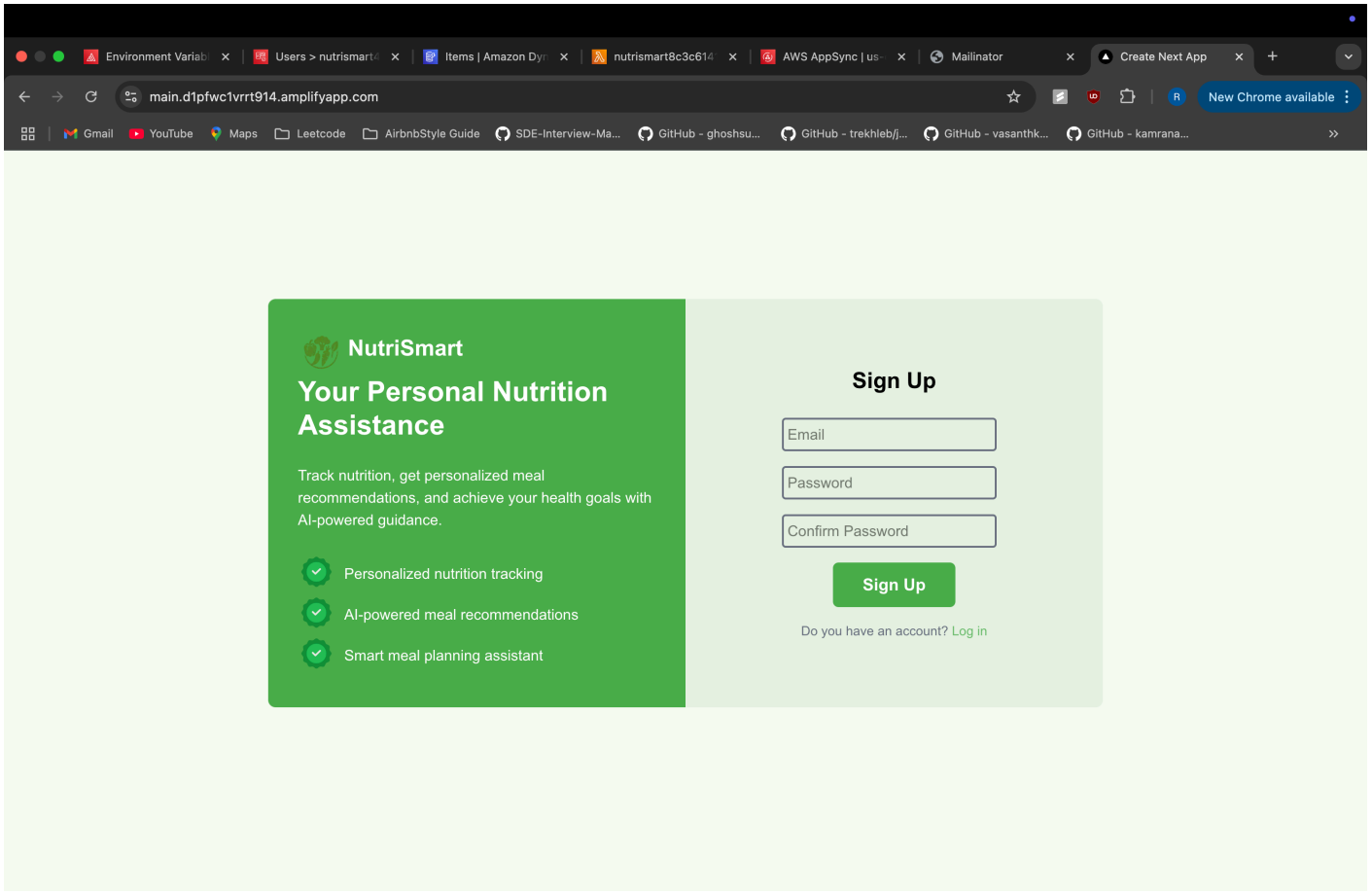
Throughout the development process, the team overcame significant challenges related to secure authentication, backend processing, and data synchronization. The resulting solution is a robust, serverless, and cost-efficient application that meets the growing demand for accessible, personalized nutrition tools.

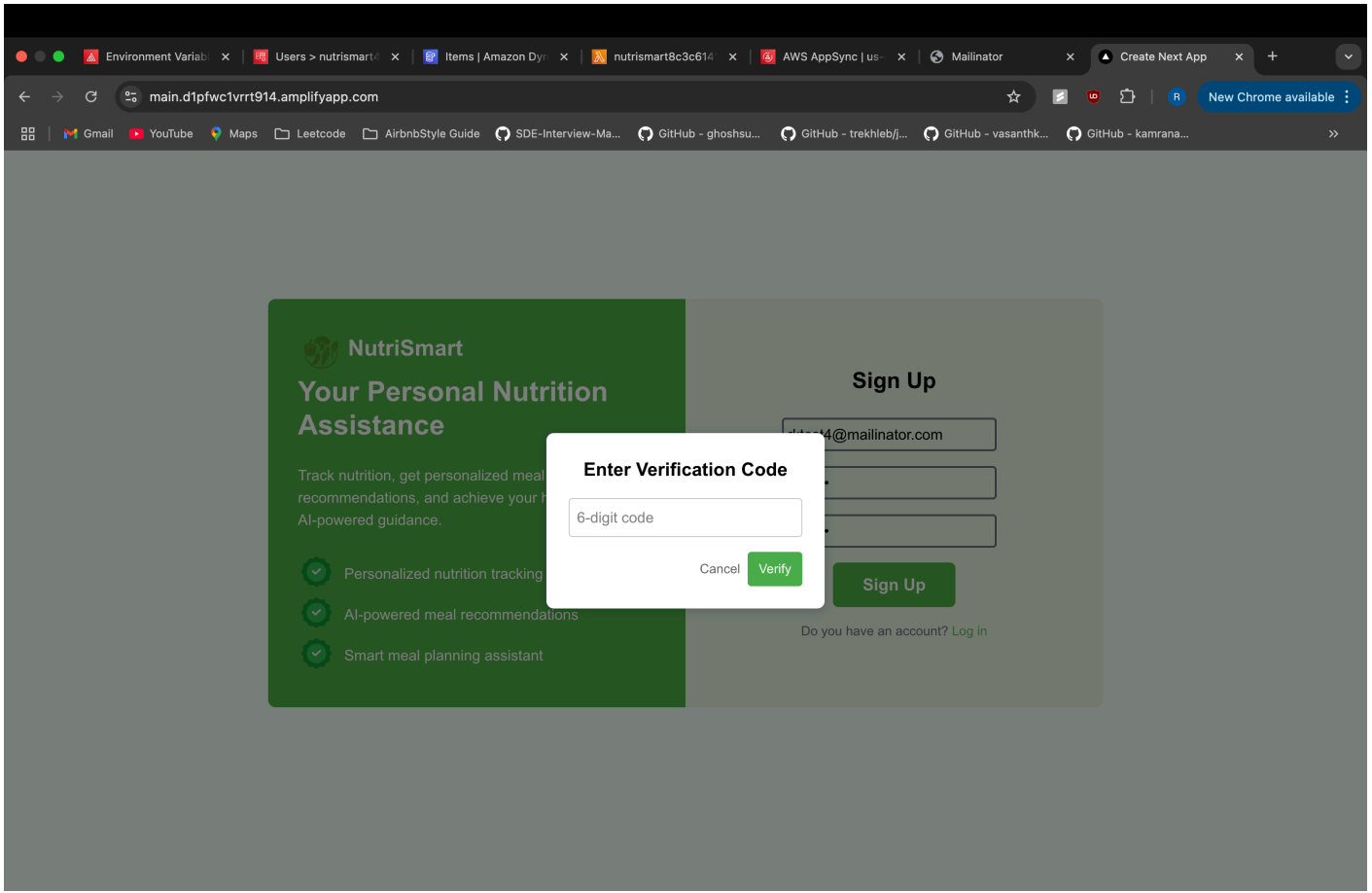
As a complete system, the Smart Nutrition Assistant provides a solid foundation for further innovation—whether through mobile app expansion, wearable integration, or advanced health analytics. It not only delivers on its original purpose but also paves the way for future enhancements that can make nutrition tracking smarter, more accessible, and more impactful in users' daily lives.

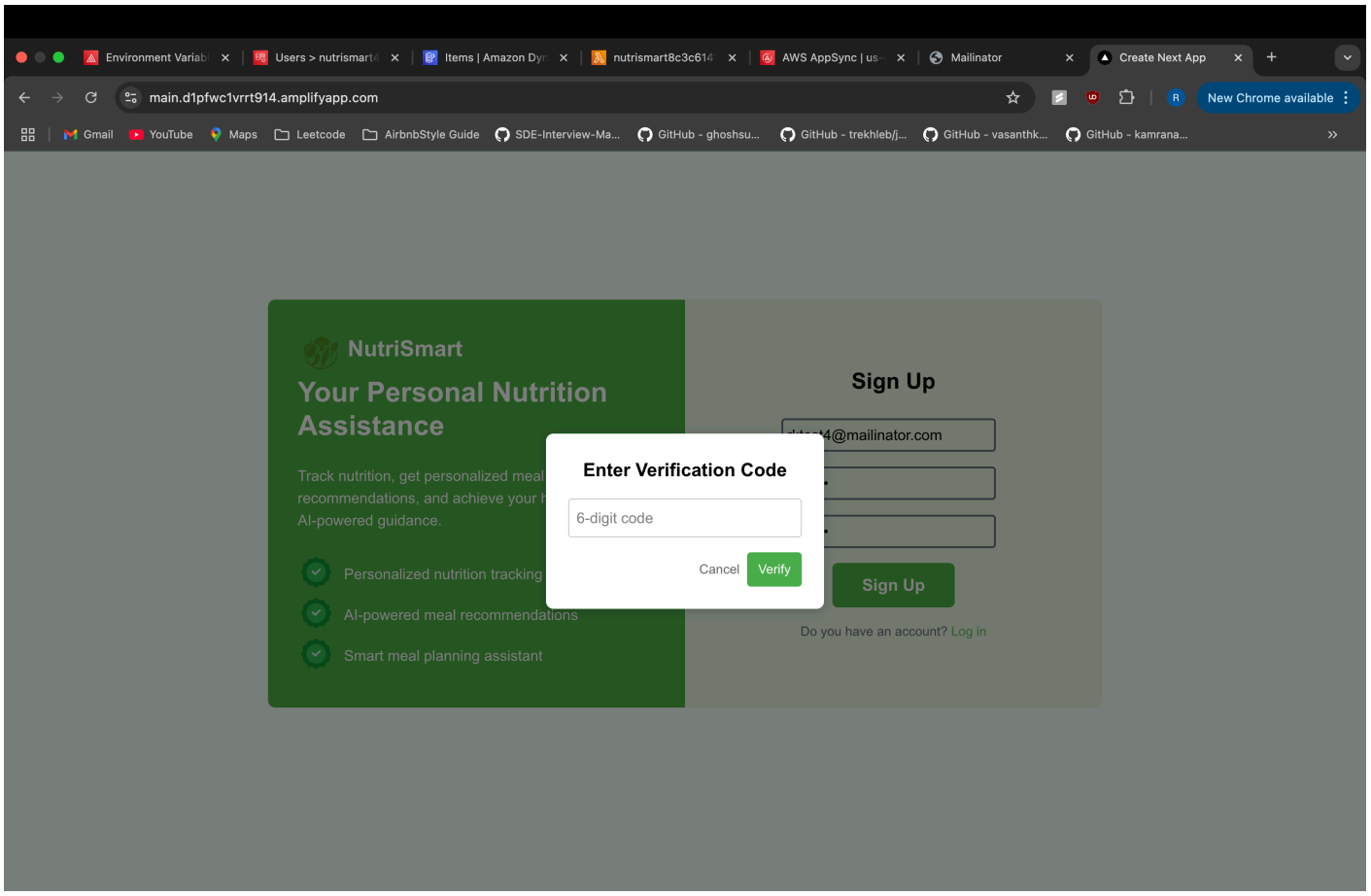
8. Screenshots:

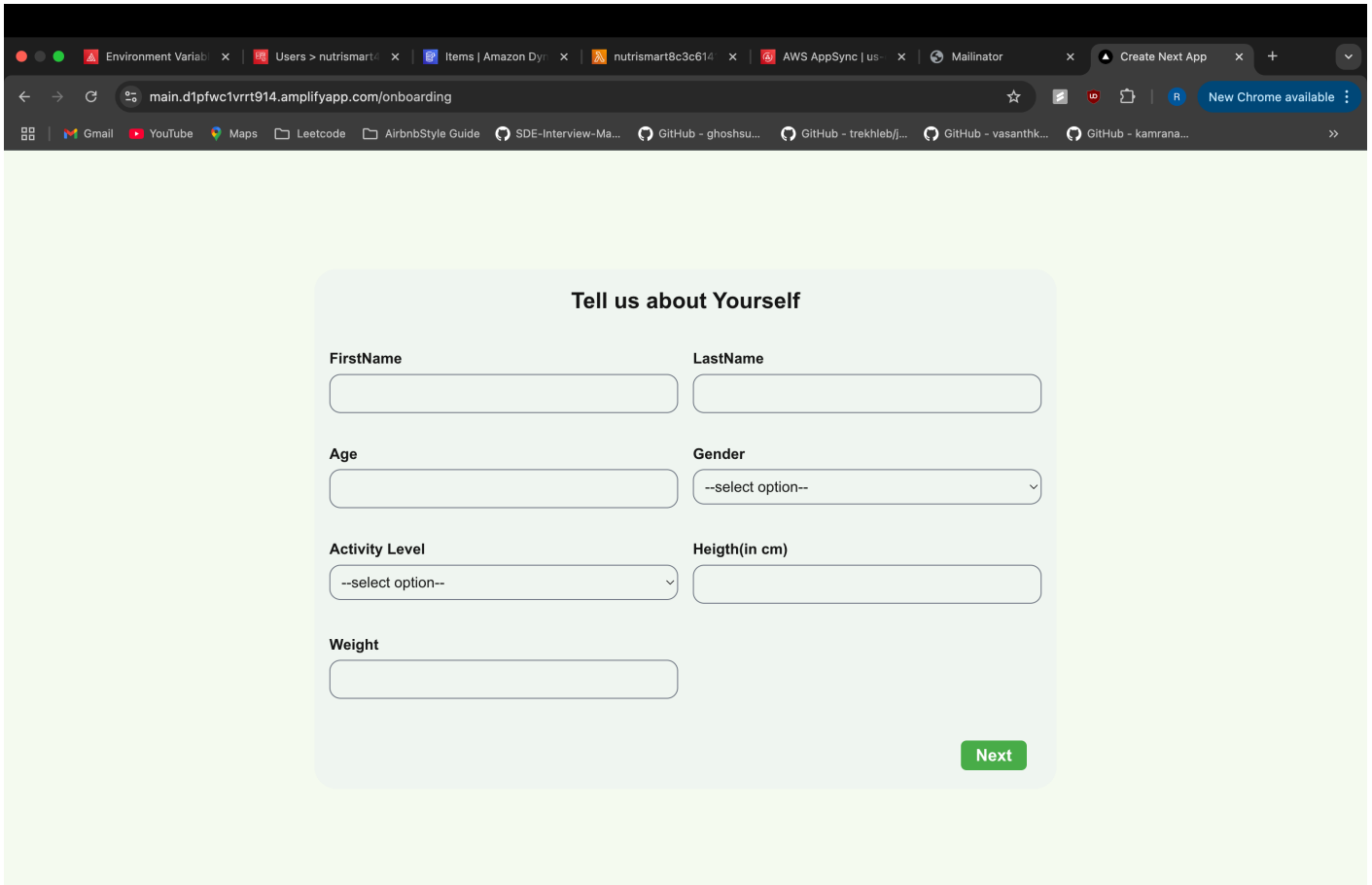

main.d1pfwc1vrrt914.amplifyapp.com
New Chrome available
NutriSmart
Your Personal Nutrition Assistance
Track nutrition, get personalized meal recommendations, and achieve your health goals with AI-powered guidance.
Personalized nutrition tracking
AI-powered meal recommendations
Smart meal planning assistant
Get Started Today
Join thousands of users who have transformed their nutrition habits with NutriSmart's personalized guidance.
Create Your Account
Do you have an account? Log in

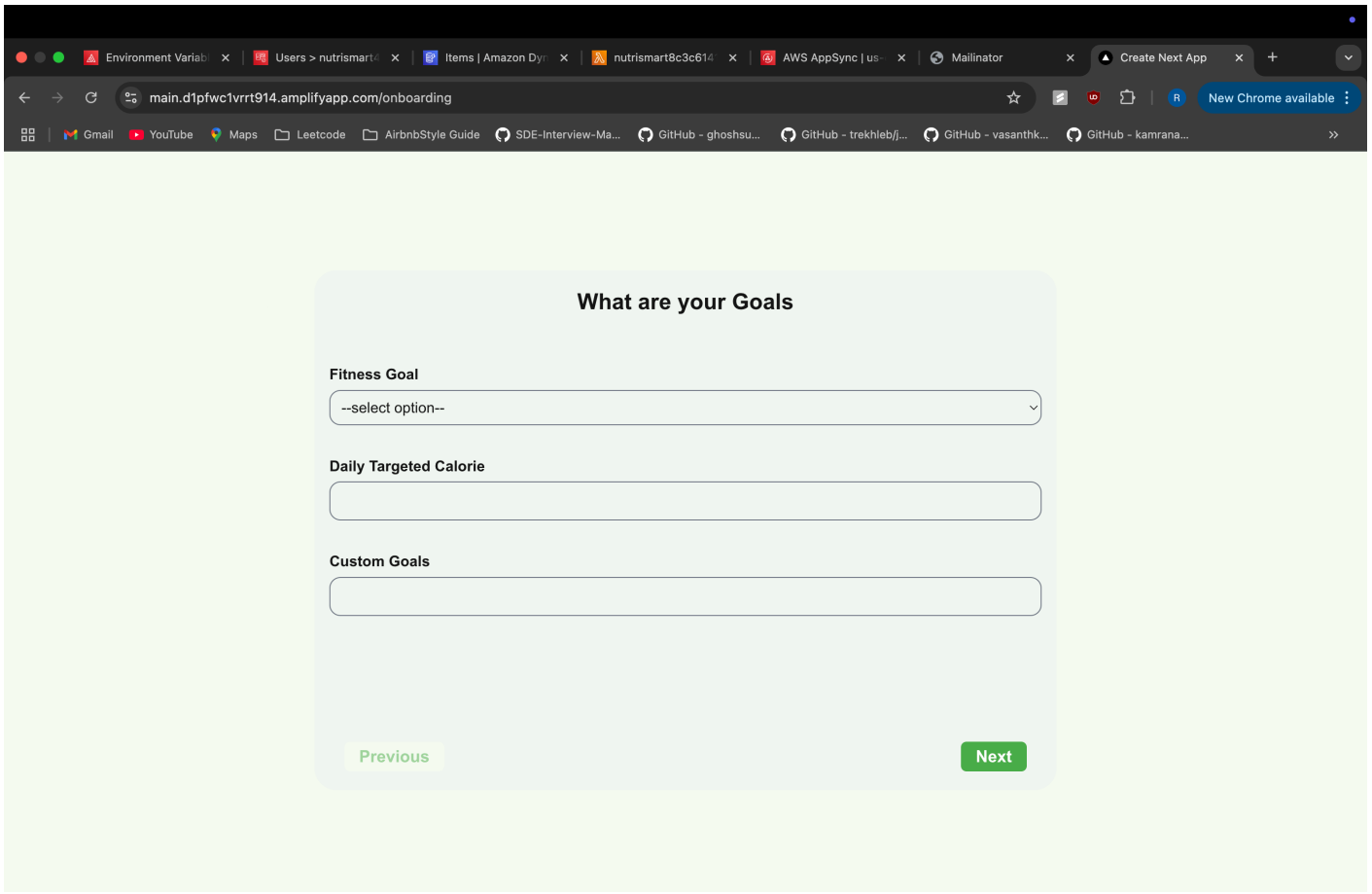

main.d1pfwc1vrrt914.amplifyapp.com
New Chrome available
NutriSmart
Your Personal Nutrition Assistance
Track nutrition, get personalized meal recommendations, and achieve your health goals with AI-powered guidance.
Personalized nutrition tracking
AI-powered meal recommendations
Smart meal planning assistant
Log In
Email
Password
Login
Don't have an account? Sign Up

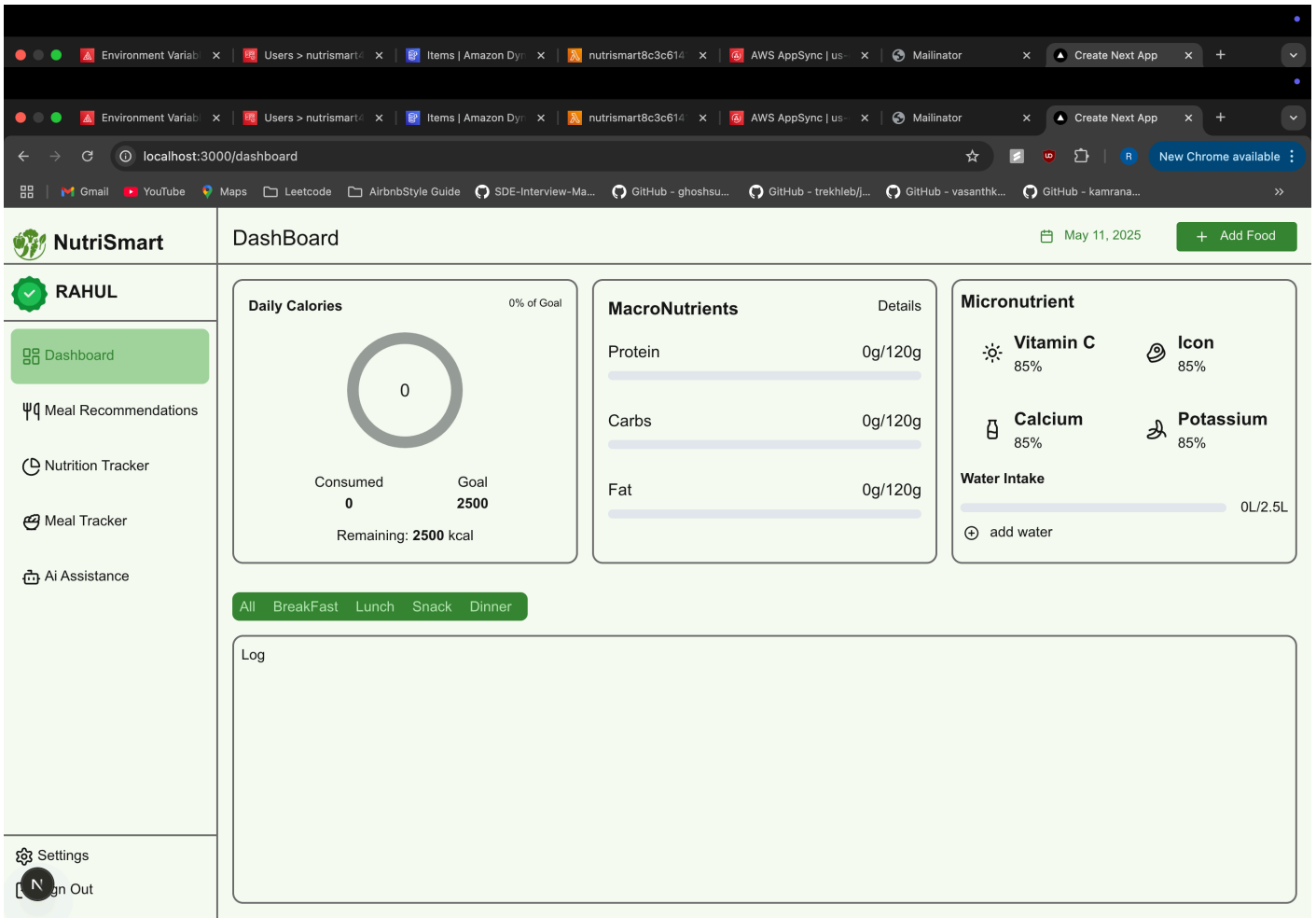

main.d1pfwc1vrrt914.amplifyapp.com
New Chrome available
NutriSmart
Your Personal Nutrition Assistance
Track nutrition, get personalized meal recommendations, and achieve your health goals with AI-powered guidance.
Personalized nutrition tracking
AI-powered meal recommendations
Smart meal planning assistant
Sign Up
Email
Password
Confirm Password
Sign Up
Do you have an account? Log in

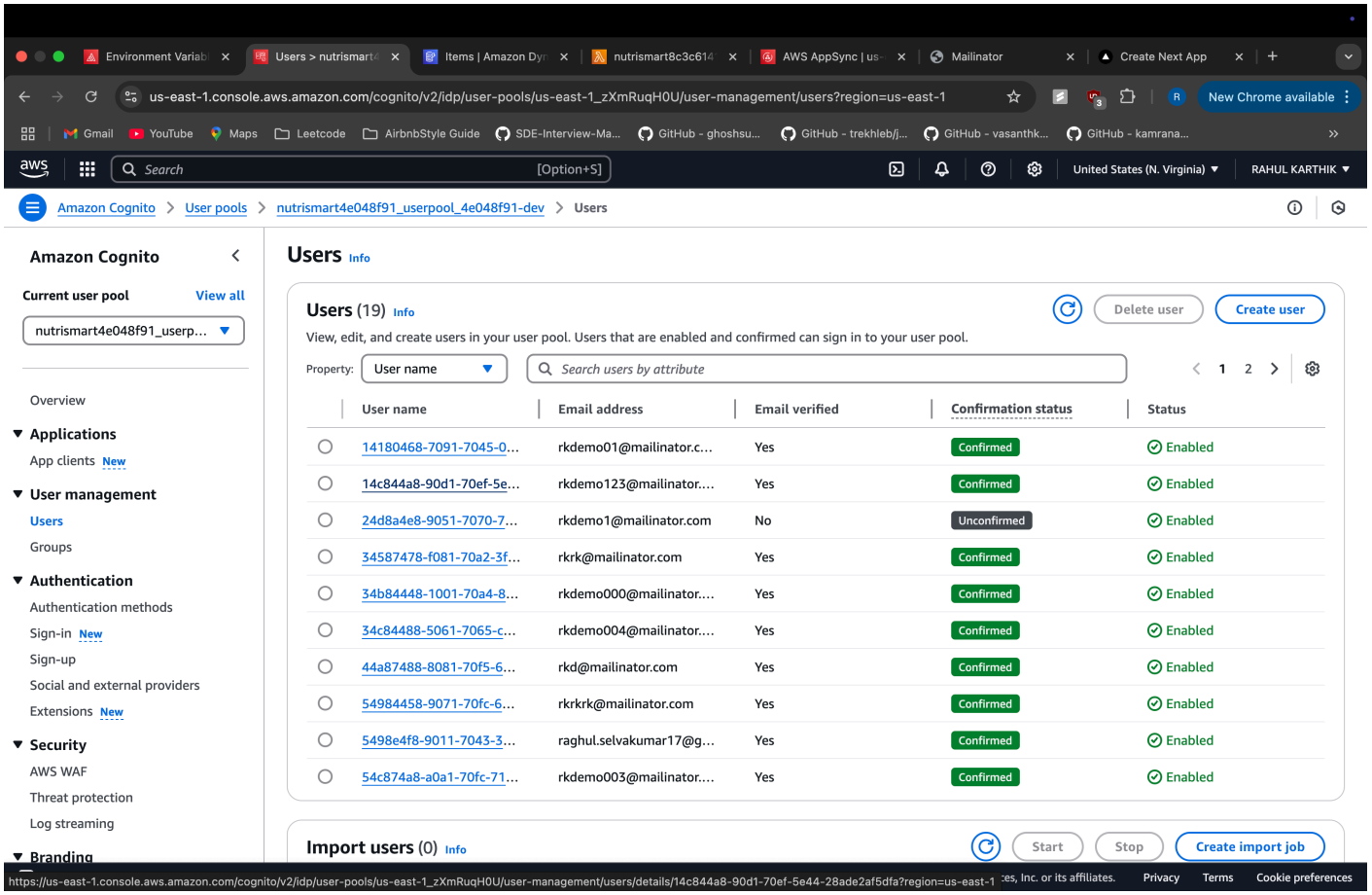

main.d1pfwc1vrrt914.amplifyapp.com
New Chrome available
NutriSmart
Your Personal Nutrition Assistance
Personalized nutrition tracking
AI-powered meal recommendations
Smart meal planning assistant
Sign Up
Do you have an account? Log in
Enter Verification Code
6-digit code
Cancel
Verify

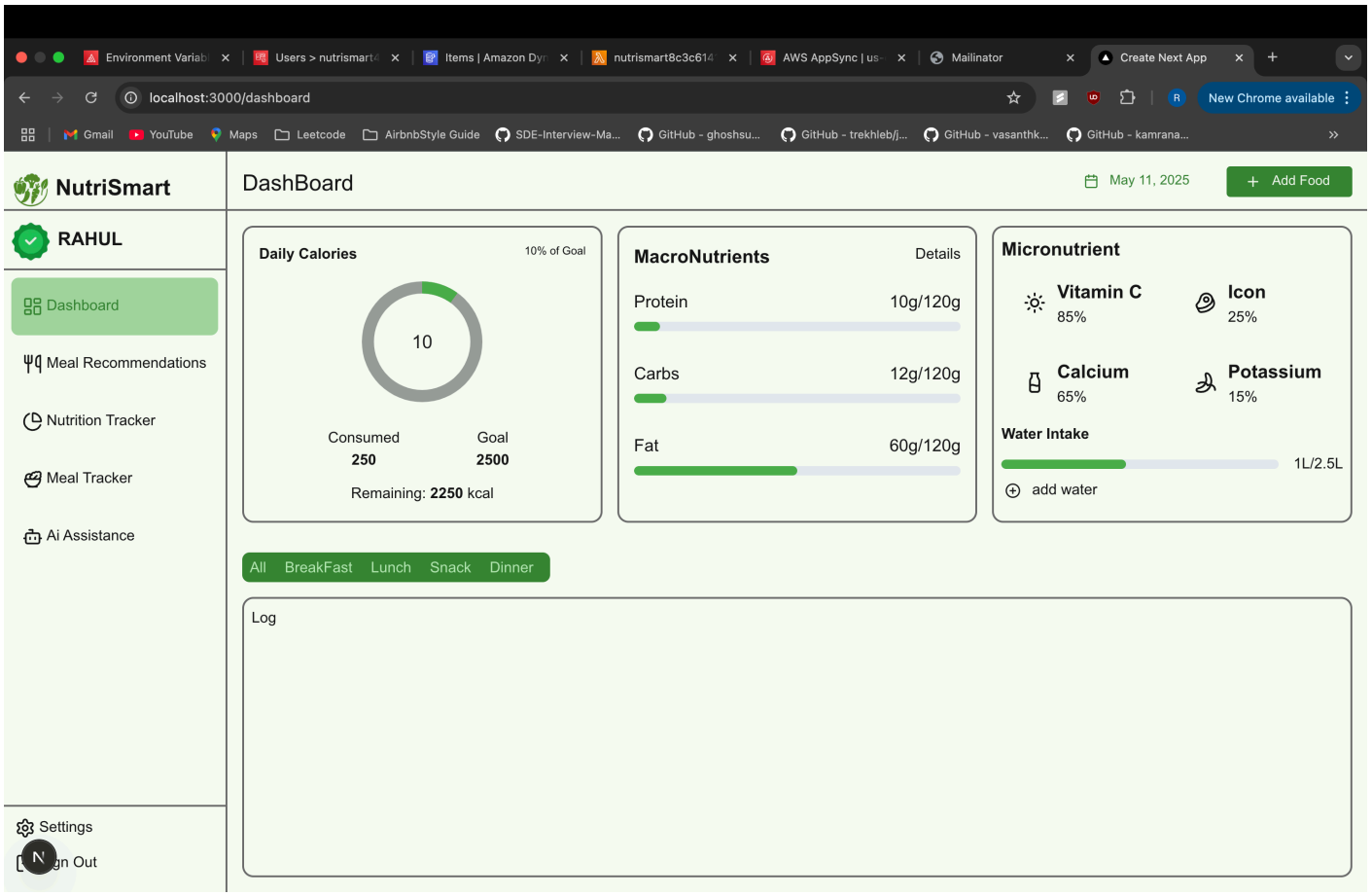

main.d1pfwc1vrrt914.amplifyapp.com
NutriSmart
Your Personal Nutrition Assistance
Track nutrition, get personalized meal recommendations, and achieve your h AI-powered guidance.
Personalized nutrition tracking
AI-powered meal recommendations
Smart meal planning assistant
Sign Up
@mailinator.com
Sign Up
Do you have an account? Log in
Enter Verification Code
6-digit code
Cancel
Verify

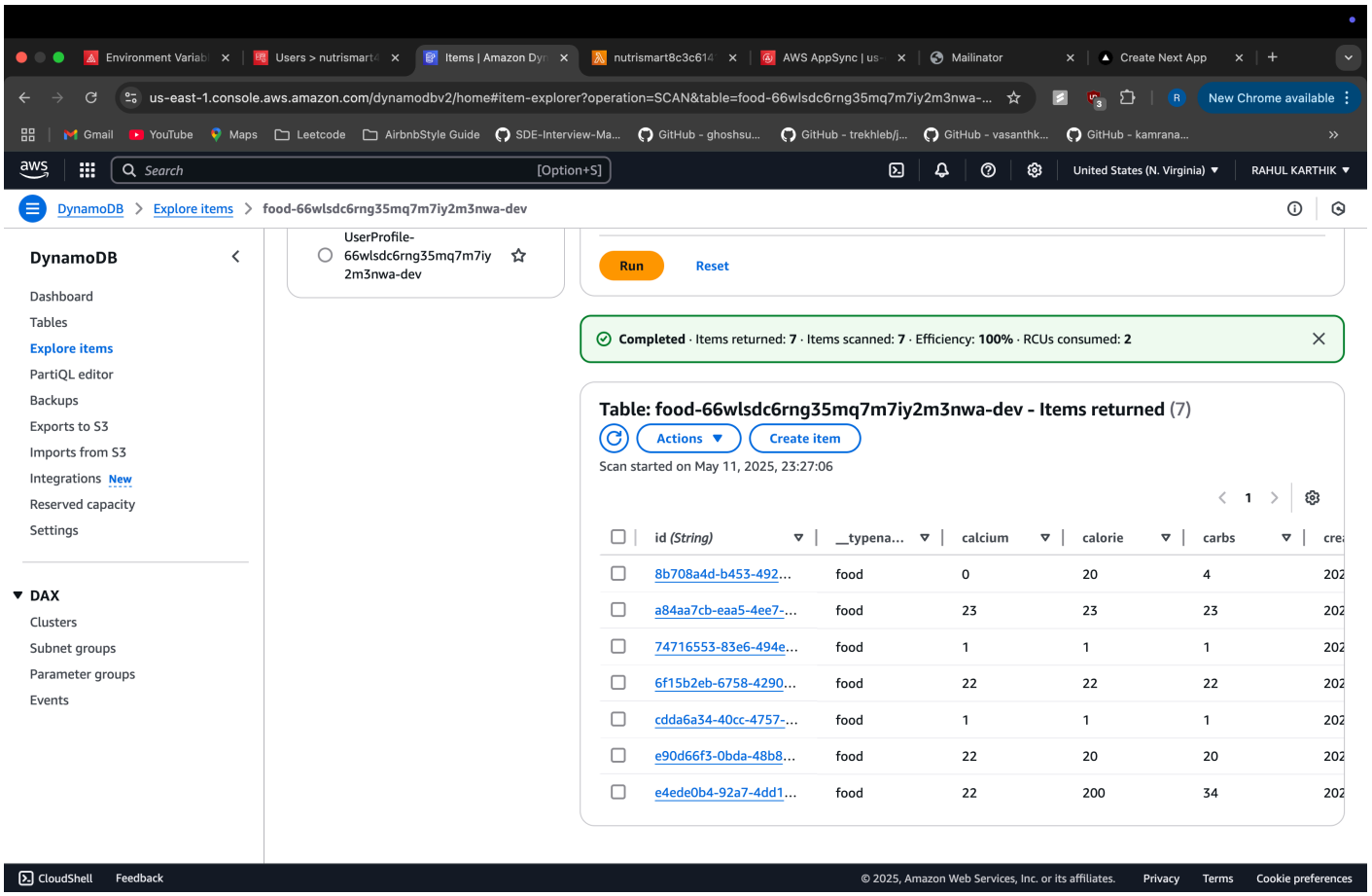

main.d1pfwc1vrrt914.amplifyapp.com/onboarding
Tell us about Yourself
FirstName
LastName
Age
Gender
--select option--
Activity Level
--select option--
Heigth(in cm)
Weight
Next

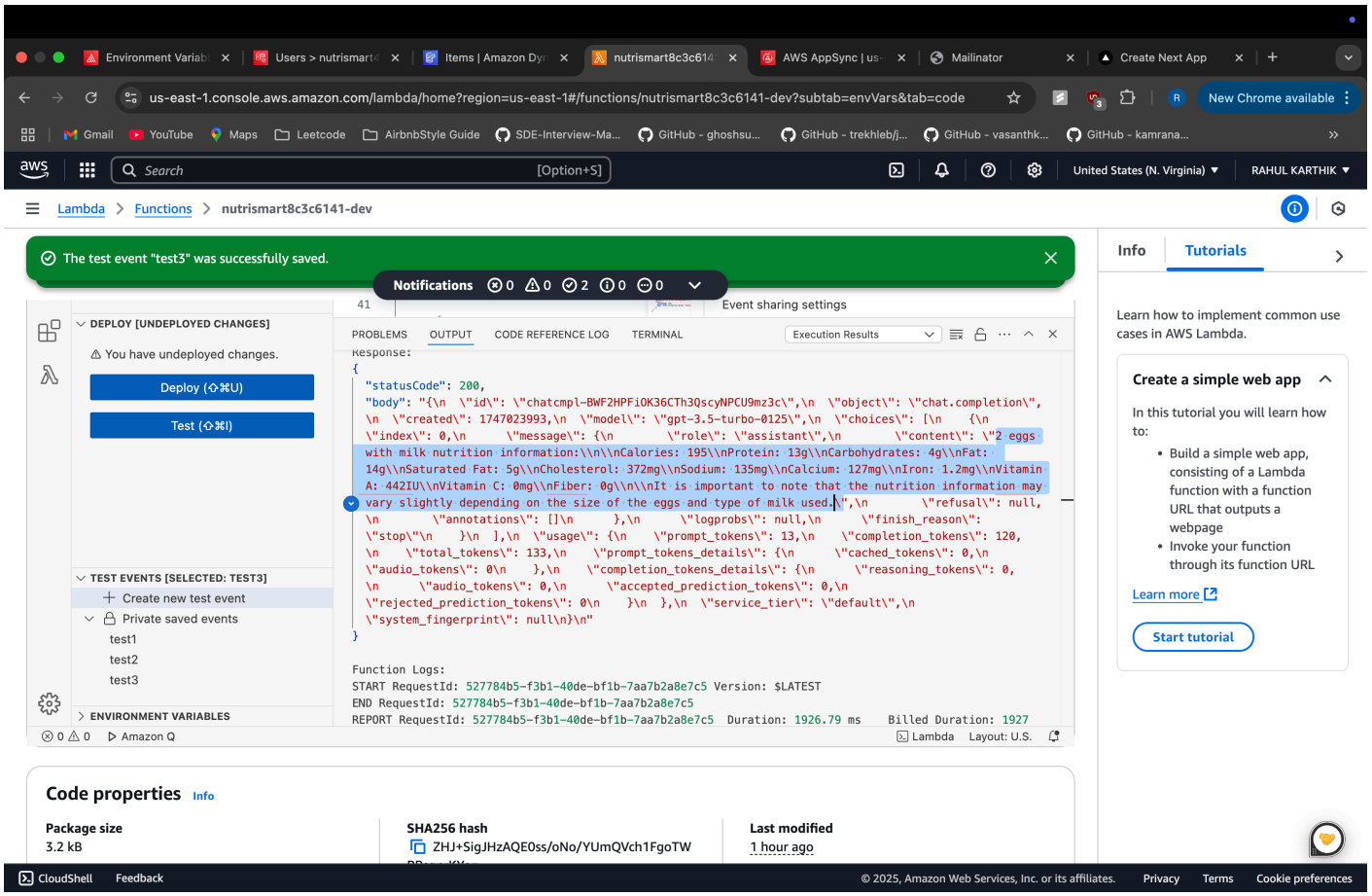

main.d1pfwc1vrrt914.amplifyapp.com/onboarding
What are your Goals
Fitness Goal
--select option--
Daily Targeted Calorie
Custom Goals
Previous
Next


localhost:3000/dashboard
NutriSmart
RAHUL
Dashboard
Meal Recommendations
Nutrition Tracker
Meal Tracker
Ai Assistance
Settings
DashBoard
May 11, 2025
Add Food
Daily Calories
0% of Goal
0
Consumed
0
Goal
2500
Remaining: 2500 kcal
MacroNutrients
Details
Protein
0g/120g
Carbs
0g/120g
Fat
0g/120g
Micronutrient
Vitamin C
85%
Icon
85%
Calcium
85%
Potassium
85%
Water Intake
0L/2.5L
add water
All
BreakFast
Lunch
Snack
Dinner
Log


us-east-1.console.aws.amazon.com/cognito/v2/idp/user-pools/us-east-1_zXmRuqH0U/user-management/users?region=us-east-1
Amazon Cognito > User pools > nutrismart4e048f91_userpool_4e048f91-dev > Users
Amazon Cognito
Current user pool
View all
nutrismart4e048f91_userp...
Overview
Applications
App clients New
User management
Users
Groups
Authentication
Authentication methods
Sign-in New
Sign-up
Social and external providers
Extensions New
Security
AWS WAF
Threat protection
Log streaming
Branding
Users Info
Users (19) Info
View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.
Delete user
Create user
Property: User name
Search users by attribute
User name
Email address
Email verified
Confirmation status
Status
14180468-7091-7045-0...
rkdemo01@mailinator.c...
Yes
Confirmed
Enabled
14c844a8-90d1-70ef-5e...
rkdemo123@mailinator....
Yes
Confirmed
Enabled
24d8a4e8-9051-7070-7...
rkdemo1@mailinator.com
No
Unconfirmed
Enabled
34587478-f081-70a2-3f...
rkrk@mailinator.com
Yes
Confirmed
Enabled
34b84448-1001-70a4-8...
rkdemo000@mailinator....
Yes
Confirmed
Enabled
34c84488-5061-7065-c...
rkdemo004@mailinator....
Yes
Confirmed
Enabled
44a87488-8081-70f5-6...
rkd@mailinator.com
Yes
Confirmed
Enabled
54984458-9071-70fc-6...
rkrkrk@mailinator.com
Yes
Confirmed
Enabled
5498e4f8-9011-7043-3...
raghul.selvakumar17@g...
Yes
Confirmed
Enabled
54c874a8-a0a1-70fc-71...
rkdemo003@mailinator....
Yes
Confirmed
Enabled
Import users (0) Info
Start
Stop
Create import job
Privacy
Terms
Cookie preferences


localhost:3000/dashboard
NutriSmart
RAHUL
Dashboard
Meal Recommendations
Nutrition Tracker
Meal Tracker
Ai Assistance
Settings
DashBoard
May 11, 2025
Add Food
Daily Calories
10% of Goal
10
Consumed
250
Goal
2500
Remaining: 2250 kcal
MacroNutrients
Details
Protein
10g/120g
Carbs
12g/120g
Fat
60g/120g
Micronutrient
Vitamin C
85%
Icon
25%
Calcium
65%
Potassium
15%
Water Intake
1L/2.5L
add water
All
BreakFast
Lunch
Snack
Dinner
Log


us-east-1.console.aws.amazon.com/dynamodbv2/home#item-explorer?operation=SCAN&table=food-66wlsdc6rng35mq7m7iy2m3nwa-...
DynamoDB > Explore items > food-66wlsdc6rng35mq7m7iy2m3nwa-dev
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations New
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
UserProfile-66wlsdc6rng35mq7m7iy2m3nwa-dev
Run
Reset
Completed · Items returned: 7 · Items scanned: 7 · Efficiency: 100% · RCUs consumed: 2
Table: food-66wlsdc6rng35mq7m7iy2m3nwa-dev - Items returned (7)
Actions
Create item
Scan started on May 11, 2025, 23:27:06
id (String)
_typena...
calcium
calorie
carbs
8b708a4d-b453-492...
a84aa7cb-eaa5-4ee7-...
74716553-83e6-494e...
6f15b2eb-6758-4290...
cdda6a34-40cc-4757-...
e90d66f3-0bda-48b8...
e4ede0b4-92a7-4dd1...
food
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/nutrismart8c3c6141-dev?subtab=envVars&tab=code
Lambda > Functions > nutrismart8c3c6141-dev
The test event "test3" was successfully saved.
Notifications
DEPLOY [UNDEPLOYED CHANGES]
You have undeployed changes.
Deploy (⇧⌘U)
Test (⇧⌘I)
TEST EVENTS [SELECTED: TEST3]
Create new test event
Private saved events
test1
test2
test3
ENVIRONMENT VARIABLES
PROBLEMS
OUTPUT
CODE REFERENCE LOG
TERMINAL
Execution Results
Response:
{
"statusCode": 200,
"body": "{\n \"id\": \"chatcmpl-BWF2HPFiOK36CTh3QscyNPCU9mz3c\",\n \"object\": \"chat.completion\",\n \"created\": 1747023993,\n \"model\": \"gpt-3.5-turbo-0125\",\n \"choices\": [\n {\n \"index\": 0,\n \"message\": {\n \"role\": \"assistant\",\n \"content\": \"2 eggs with milk nutrition information:\\n\\nCalories: 195\\nProtein: 13g\\nCarbohydrates: 4g\\nFat: 14g\\nSaturated Fat: 5g\\nCholesterol: 372mg\\nSodium: 135mg\\nCalcium: 127mg\\nIron: 1.2mg\\nVitamin A: 442IU\\nVitamin C: 0mg\\nFiber: 0g\\n\\nIt is important to note that the nutrition information may vary slightly depending on the size of the eggs and type of milk used.\",\n \"refusal\": null,\n \"annotations\": []\n },\n \"logprobs\": null,\n \"finish_reason\": \"stop\"\n }\n],\n \"usage\": {\n \"prompt_tokens\": 13,\n \"completion_tokens\": 120,\n \"total_tokens\": 133,\n \"prompt_tokens_details\": {\n \"cached_tokens\": 0,\n \"audio_tokens\": 0\n },\n \"completion_tokens_details\": {\n \"reasoning_tokens\": 0,\n \"audio_tokens\": 0,\n \"accepted_prediction_tokens\": 0,\n \"rejected_prediction_tokens\": 0\n }\n },\n \"service_tier\": \"default\",\n \"system_fingerprint\": null\n}\n"
}
Function Logs:
START RequestId: 527784b5-f3b1-40de-bf1b-7aa7b2a8e7c5 Version: $LATEST
END RequestId: 527784b5-f3b1-40de-bf1b-7aa7b2a8e7c5
REPORT RequestId: 527784b5-f3b1-40de-bf1b-7aa7b2a8e7c5 Duration: 1926.79 ms Billed Duration: 1927
Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial
Code properties
Info
Package size
3.2 kB
SHA256 hash
ZHJ+SigJHzAQE0ss/oNo/YUmQVch1FgoTW
Last modified
1 hour ago
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

GITHUB LINK : **https://github.com/rahul-karthik-au/smart-nutrition-assistance**

PROJECT LINK : **https://main.d1pfwc1vrrt914.amplifyapp.com/**

PRESENTATION LINK :

https://drive.google.com/drive/folders/1daVxitz9I3_rrhLfqzXD_DeTnp6wJxQa?usp=sharing