PROJECT DOCUMENTATION

ON

NUTRITION INTAKE TRACKER USING DOCUMENT DATABASES

NAME: BHAVYA SIROBHUSHANAM

CWID: A20518887

OVERVIEW

The Nutrition Intake Tracker Application is designed to provide users with a tool to manage their daily nutritional intake. The application is built using Python and leverages AWS's DynamoDB as the backend database. DynamoDB is a fully managed, NoSQL database service that provides fast and predictable performance, making it an ideal choice for the application.

The application provides several functionalities, including adding, updating, and deleting nutritional intake records, setting and getting daily nutritional goals, comparing daily intake with goals, and searching for food items by name. By providing these functionalities, users can easily track their nutritional intake and make informed decisions about their dietary habits.

The application's core technology, DynamoDB, is a NoSQL database service that allows for efficient storage and retrieval of data. The use of DynamoDB ensures high performance and scalability, making it suitable for use in applications that require low latency and high throughput. In addition, the application uses the boto3 Python SDK to interact with DynamoDB, allowing for seamless integration between the Python script and the database.

The Nutrition Intake Tracker Application is relevant to big data in several ways. It showcases how NoSQL databases like DynamoDB can be used to store and manage large volumes of data efficiently. It also demonstrates how cloud-based services like AWS can facilitate the processing and storage of big data. Finally, the application showcases how big data can be utilized to provide valuable insights into users' dietary habits and help them make informed decisions about their health.

INTRODUCTION

In today's world, managing daily nutrition intake has become a vital aspect of a healthy lifestyle. People often struggle to keep track of what they eat, and this leads to various health problems in the long run. To address this issue, the Nutrition Intake Tracker Application was developed as a Python-based system that helps users manage their daily nutritional intake.

The application is built on Amazon Web Services' (AWS) DynamoDB, a NoSQL database service that provides fast and predictable performance. DynamoDB is ideal for use cases requiring low latency and high throughput, making it a suitable choice for this application. By leveraging AWS's services, the application ensures high performance, scalability, and reliability.

The Nutrition Intake Tracker Application allows users to track their daily nutritional intake by adding, updating, and deleting nutritional intake records. Users can also set and get daily nutritional goals, compare daily intake with goals, and search for food items by name. This functionality allows users to easily monitor their food consumption and make informed decisions about their dietary habits.

For example, if a user wants to track their daily protein intake, they can set a goal for the amount of protein they want to consume per day. The application then helps them track their protein intake by recording the amount of protein they consume with each meal. Users can view their progress towards their daily goal and adjust their food intake accordingly.

In addition, the application can provide valuable insights into users' dietary habits. By analyzing the data collected, the application can identify patterns and trends in users' food consumption. This information can help users identify areas where they need to make changes in their diet to achieve their health goals.

# **Design**

The design of the Nutrition Intake Tracker Application is centered around the use of Amazon Web Services' (AWS) DynamoDB, a fully managed NoSQL database service, and the boto3 Python SDK. The application utilizes DynamoDB to store and retrieve user nutritional data, and boto3 is used to interact with DynamoDB, allowing seamless integration between the Python script and the database.

The DynamoDB table for the Nutrition Intake Tracker Application has the following structure: UserID, Date, FoodID, Name, Calories, Protein, Carbohydrates, and Fat. This table structure ensures efficient storage and retrieval of data, as it allows the application to easily query the data based on UserID, Date, and FoodID.

The application provides several functionalities for users, including adding, updating, and deleting nutritional intake records, setting and getting daily nutritional goals, comparing daily intake with goals, and searching for food items by name. Users can easily add nutritional intake records by entering the relevant information such as the date, the type of food consumed, and the macronutrient content of the food. The application then stores this information in the DynamoDB table.

The application also allows users to set daily nutritional goals based on their individual dietary needs. For example, a user may want to set a goal of consuming 2000 calories per day, with a specific macronutrient distribution. The application allows the user to set these goals and provides feedback on their progress towards meeting these goals.

To facilitate searching for food items by name, the application uses a simple search algorithm that matches the user input with food names stored in the DynamoDB table. This feature allows users to easily find the nutritional information for a specific food item, even if they do not have the exact nutritional information

# **Architecture**

The web UI of the Nutrition Intake Tracker Application is built using Django, a high-level Python web framework that enables rapid development and clean, pragmatic design. Django follows the Model-View-Controller (MVC) architectural pattern and emphasizes reusability and pluggability of components, making it a popular choice for web development.

The front-end of the application is designed using HTML, CSS, and JavaScript, which are rendered by the Django server. Django provides a templating engine that enables developers to separate the presentation logic from the business logic, resulting in clean and maintainable code. The templates define the structure of the HTML pages, while the views handle the business logic and serve the templates to the client.

The Django server interacts with the DynamoDB table through the Python script, using the boto3 SDK. The server receives requests from the client, processes them, and retrieves data from the DynamoDB table as needed. The server then sends the response back to the client, which updates the UI accordingly.

# **Work Environment**

This application is built using Visual studio code platform. VS Code is easy to use, and has extensive customization options, and its ability to support a wide range of programming languages and frameworks. It also has a robust plugin ecosystem, allowing users to add functionality for things like debugging, version control, and language support. Additionally, VS Code has a built-in terminal and debugging capabilities, making it a versatile tool for developers of all levels. Overall, VS Code is a powerful and user-friendly code editor that is widely used by developers around the world.

A virtual environment is utilized in vs code. It is an isolated Python environment that allows developers to install and manage packages separately from other projects on their system. In VS Code, virtual environments can be created and managed using extensions like Python Extension for Visual Studio Code or Anaconda Extension.

Virtual environments allow you to isolate packages and dependencies for specific projects, ensuring that the project runs smoothly without interference from other dependencies on the system. By using virtual environments, you can ensure that the same package versions are used across different development environments, ensuring that the code behaves consistently. They can also help prevent any security issues that could arise from using outdated packages or dependencies.

**OUTPUT**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, table

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated

**CONCLUSION**

The Nutrition Intake Tracker Application is a highly efficient and scalable solution for users to manage their daily nutritional intake. With the use of Amazon Web Services' DynamoDB, the application provides a fast and predictable performance that can handle large volumes of data. The boto3 Python SDK enables seamless integration between the Python script and the database, allowing for a smooth user experience. The various functionalities provided by the application, such as adding, updating, and deleting nutritional intake records, setting and getting daily nutritional goals, comparing daily intake with goals, and searching for food items by name, make it a valuable asset for users who want to make informed decisions about their health. Moreover, the application showcases the relevance of big data in providing insights into users' dietary habits, which can be leveraged to promote healthy lifestyles. Overall, the Nutrition Intake Tracker Application is an innovative solution that combines the power of cloud-based services with cutting-edge technologies to provide users with a robust and efficient tool to manage their dietary habits.

**REFERENCES**

1. "Using AWS DynamoDB with Python and Boto3" by AWS: <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.Python.html>
2. "Building a Web Application with Python, Django, and DigitalOcean" by DigitalOcean: <https://www.digitalocean.com/community/tutorials/building-a-web-application-with-python-django-and-digitalocean>
3. B. Patil and S. Rajgure, "A Nutrition Tracking System using Big Data and Machine Learning Techniques," 2018 IEEE 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2018, pp. 375-380, doi: 10.1109/CONFLUENCE.2018.8442607.
4. T. Xu, X. Liu, J. Liu and Y. Zhang, "Design and implementation of a nutrition management system based on the Internet of Things," 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 2019, pp. 413-418, doi: 10.1109/ICCCBDA.2019.8725696.
5. Y. Zhou, S. Tan and Z. Cheng, "Design and Implementation of Nutrition Intake Monitoring System Based on Wearable Devices," 2018 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, China, 2018, pp. 475-479, doi: 10.1109/CSE-EUC-DCABES.2018.00116.
6. J. Xu and Q. Zhang, "Design and Implementation of Personalized Nutrition Tracking and Analysis System Based on Machine Learning," 2019 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 2019, pp. 1645-1650, doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00265.
7. A. D. D'Angelo, G. Gualtieri, A. Liccardo, F. Palmieri and M. Pota, "A Platform for Food Intake Monitoring Using Wearable Devices and Machine Learning Techniques," in IEEE Access, vol. 6, pp. 4605-4614, 2018, doi: 10.1109/ACCESS.2017.2784782