**Documentation for D2C Produce Warehouse Management Application**

Github link:

**1. Introduction**

The D2C (Direct-to-Consumer) Produce Warehouse Management Application is designed to manage various aspects of warehouse operations, including inventory management, temperature control, shelf life prediction, traceability, and adherence to packaging standards. This document provides a comprehensive overview of the sub-modules within the application, their integration, process flows, AI/ML techniques used, and deployment strategies.

**2. Description of Sub-Modules**

**2.1. Inventory Management**

The Inventory Management sub-module handles the addition, updating, and retrieval of inventory items. It includes functionalities to add new products, update quantities, and fetch current inventory levels. Users can input product details and quantities, which are stored in a database for further processing. This module ensures that inventory records are up-to-date and accurately reflect the current stock.

**2.2. Temperature Control**

The Temperature Control module monitors and adjusts the temperature within the warehouse to ensure that products are stored under optimal conditions. This sub-module employs machine learning techniques to forecast temperature trends and detect anomalies. Real-time data from temperature sensors are analyzed to maintain appropriate storage conditions and prevent spoilage.

**2.3. Shelf Life Prediction**

This module predicts the remaining shelf life of products based on various factors such as initial quality, storage conditions, and historical data. Machine learning models, including regression algorithms, are used to forecast shelf life, providing insights into product longevity and helping in inventory management and planning.

**2.4. Traceability**

The Traceability module tracks the origin and journey of products throughout the supply chain. It records details about product sourcing, handling, and distribution, enabling users to trace products back to their origins. This module is crucial for ensuring transparency and compliance with regulatory standards.

**2.5. Packaging Standards**

The Packaging Standards module ensures that products meet the required biodegradable packaging criteria. It includes functionalities to verify packaging materials, assess environmental impact, and ensure compliance with industry standards. This module supports sustainability initiatives by promoting the use of eco-friendly packaging solutions.

**3. Architecture/Design Diagram**

The architecture of the application is designed to ensure seamless integration between sub-modules and efficient data flow. Below is a high-level design diagram showcasing the integration:

**[Insert Architecture/Design Diagram Here]**

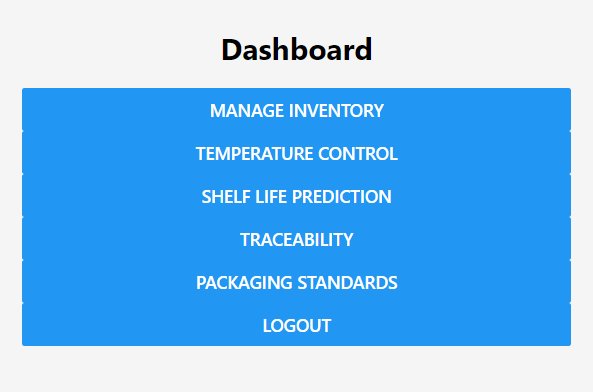
* **User Interface (UI):** Provides interfaces for Inventory Management, Temperature Control, Shelf Life Prediction, Traceability, and Packaging Standards.
* **Backend Services:** Handles business logic, data processing, and interactions between sub-modules.
* **Database:** Stores inventory data, temperature logs, shelf life predictions, traceability records, and packaging standards.
* **AI/ML Models:** Includes models for shelf life prediction and temperature control.

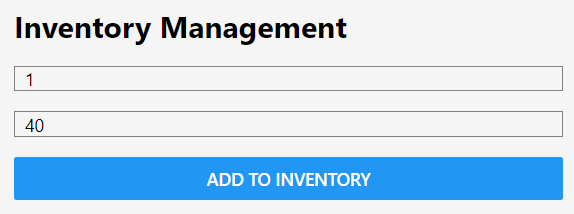
**4. Process Flow Map**

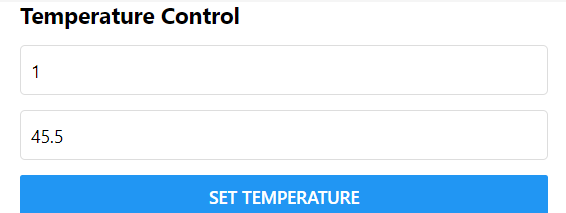
The process flow for users interacting with the application involves the following steps:

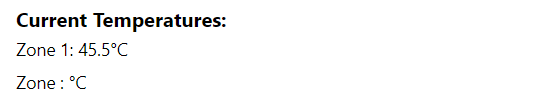
1. **Login:** Users access the application via a secure login screen.
2. **Dashboard:** After logging in, users are directed to the dashboard, which provides an overview of inventory status, temperature conditions, and other key metrics.
3. **Inventory Management:**
   * **Add/Update Inventory:** Users can add new products or update existing inventory.
   * **View Inventory:** Users can view current inventory levels and product details.
4. **Temperature Control:**
   * **Monitor Temperature:** Real-time temperature data is displayed.
   * **Adjust Settings:** Users can adjust temperature settings based on predictions and alerts.
5. **Shelf Life Prediction:**
   * **Predict Shelf Life:** Users can input product details to receive shelf life predictions.
   * **View Predictions:** Display of predicted shelf life for various products.
6. **Traceability:**
   * **Track Products:** Users can track the origin and journey of products.
   * **View Records:** Detailed records of product traceability are available.
7. **Packaging Standards:**
   * **Verify Packaging:** Users can check if packaging meets biodegradable standards.
   * **Review Compliance:** Review compliance reports and packaging details.

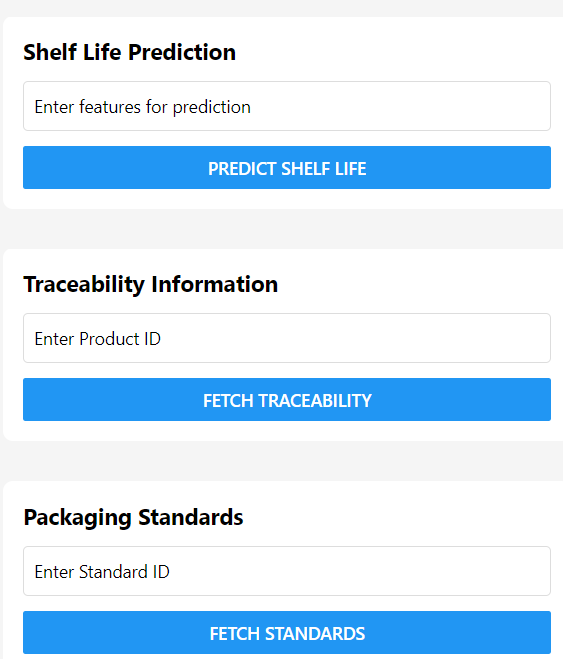
**Output:**









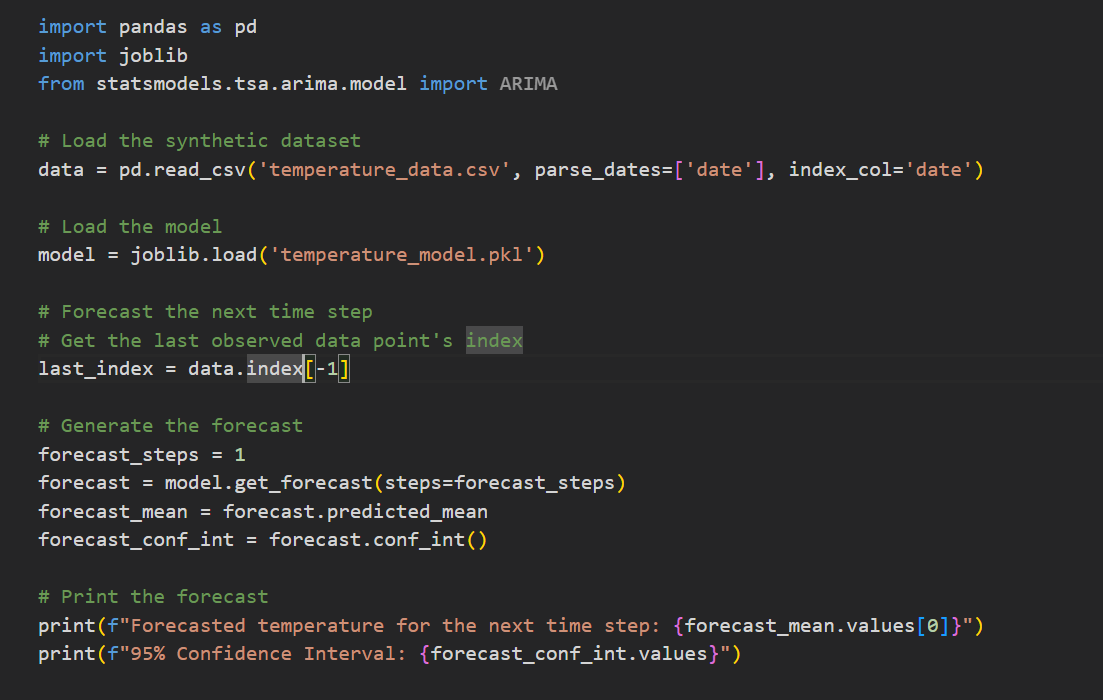


**5. Details of AI/ML Sub-Modules**

**5.1. Temperature Control**

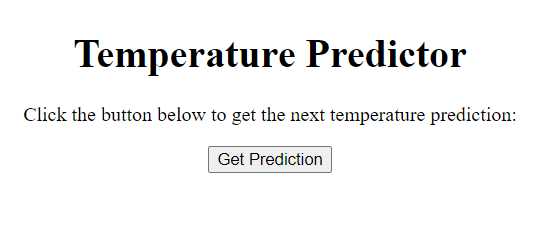
Temperature Control uses Time Series Forecasting and Anomaly Detection algorithms. Time Series Forecasting (e.g., ARIMA) predicts future temperature trends based on historical data, while Anomaly Detection identifies deviations from normal patterns. These models are integrated into the backend system, interfacing with warehouse management systems for real-time temperature adjustments.

The pretrained ARIMA model:

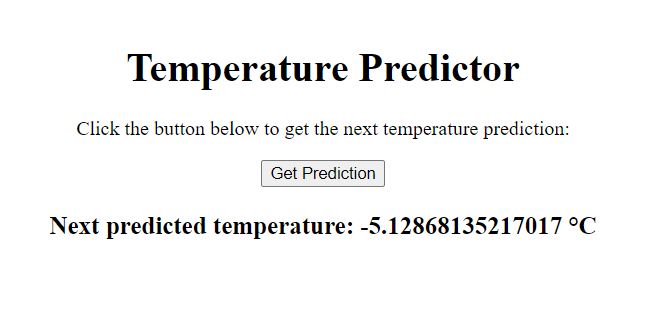


The temperature prediction is done using ARIMA and the next temperature can be displayed

**Before prediction:**



**After Prediction:**



**5.2. Shelf Life Prediction**

Machine learning techniques are used to forecast the remaining shelf life of products. Regression algorithms, such as Linear Regression is employed to analyze historical data and predict future shelf life. The model are trained using data on product characteristics and storage conditions. The trained models are deployed as APIs, allowing real-time predictions based on current data.



**5.3. Deployment Approach**

Models are trained on historical data, validated for accuracy, and deployed as APIs or integrated into the backend. Data pipelines are set up to handle continuous data input and model retraining. This approach ensures that the AI/ML models provide accurate and timely insights for inventory management and temperature control.

The trained models are deployed as pkl dumps and are integrated using Flask

**6. Citations and References**

For further reading and technical references on AI/ML techniques used, consult the following resources:

1. **Regression Techniques:**
   * "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido.
   * "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron.
2. **Time Series Forecasting:**
   * "Time Series Analysis and Its Applications: With R Examples" by Robert H. Shumway and David S. Stoffer.
   * "Deep Learning for Time Series Forecasting" by Jason Brownlee.
3. **Anomaly Detection:**
   * "Anomaly Detection: A Survey" by Chandola, V., et al. (ACM Computing Surveys, 2009).
   * "Practical Machine Learning for Data Analysis Using Python" by Himanshu Singh.