

Name= **Subham Ranjan Konar**

Team_name= **subhamranjankonar.21ds**

college= **sai vidya institute of technology**

Purpose:

The purpose of the project is to make the process of inspection of the goods and items during the packaging which involves item classification, item quantity analysis and quality inspection

Structure:

1. Project Overview:

Project Name: Billing System with Quantity and Inventory Verification

Objective:

The goal of this project is to build a robust web-based billing system that ensures the accuracy of item quantities between what is expected (input from the user) and what is present and detected by the trained model . The system validates that:

1. Items listed by users match the items present in the database.
2. Quantity discrepancies between the input and the inventory are detected.
3. Users are informed about missing items, incorrect quantities, and any extra items that don't match the database.
4. The system triggers alert if there is any defective , expired and unwanted item in the basket which was going to be delivered.
5. To make a system which of industrial standards and integrate into the existing well built system seamlessly.

Key Features:

1. Input Verification:

- Users can input item names along with their quantities.
- The system checks whether these items exist in the things that the model has recognised by the camera input while scanning and whether the provided quantities match the stored inventory quantities.

2. Quantity Validation:

- If there are discrepancies in item quantities, users are notified about the excess or shortage.
- The system highlights both the item name and the difference in expected vs. actual quantities.

3. Database Integration:

- The application connects with a database to fetch item details such as names and quantities, expiry dates and checks if it is expired or not.
- The system ensures that all items and their quantities match between user input and the database.

4. Error Reporting:

- Users are notified of missing items, incorrect quantities, and extra items through detailed error messages.
- The system provides statistics like the total number of missing or extra items for clarity.
- The system takes care of every exception and disruptions which may occur and ensures proper error handling and process continuation

5. Real-time Testing & Feedback:

- The project includes a live web interface with real time notification push to facilitate real-time testing and feedback for the users.
- Stakeholders can access the application online, providing input and evaluating results.
- The software can be integrated with apis which can be connected with flipkarts server and fetch the customer order and evaluate the order and give its result right onto the frontend screen.

6. Quality Validation

- If there are discrepancies in item quality , no matter it be damaged or it being expired, users are notified about the condition of the items.
- The system highlights both the item name and the defect name which will help user decide and analyse the condition of the items.

Use Cases:

1. Can be used in flipkart store house and godowns:

- Validate that the inventory accurately reflects customer input and detect discrepancies in item quantities.
- Ensure stock consistency to streamline billing and reduce errors during transactions.

2. Administrators:

- Monitor the system's performance and accuracy by reviewing missing or extra items.
- Manage inventory adjustments effectively based on validation feedback.
- An admin can check the amount of mistakes done and the defective products received from the manufacture and can improve the whole operation.

Outcome:

By implementing this system, users gain confidence that their billing data aligns with the actual inventory. This ensures accuracy, quality, reduces errors, and streamlines the overall inventory management and billing process. The system aims to enhance reliability and provide actionable insights for better decision-making.

40 mini

•

2. Codebase Structure:

steps to run the code and the run the code locally

- Go to the official yolov7 model code and clone the repository .
Clone this repo: <https://github.com/WongKinYiu/yolov7>
 - The go my git repo and clone my git too.
Clone my repo at: https://github.com/Subhamkonar84/flipkart_grid6_robotics
 - Replace detect.py in yolov7 with detect2.py from my folder.
 - Copy the best3.py from my repository and place it in the yolov7 folder.
 - Open terminal and install the requirements in requirements.txt in yolov7
use the code ``pip install -r requirements.txt`` to install all libraries for running
 - Have a mysql database and config your database username and password for storing the data and manually add few data you will be working on with.
 - start the live video detection from camera input using the following command
``python3 detect2.py --weights best3.pt --conf 0.9 --img-size 640 --source 0``
 - The above code will help you start the camera and start detection.
 - To run the gui go to the billing_app folder in my repository and open your terminal.
 - Ensure to download all the django framework libraries for server and run the migration of the models present in the folder(if not you could use my database).
 - Make the server run my running the following command
``python manage.py runserver``
 - This will create an link ``http://127.0.0.1:8000/`` this is a localhost link and can be accessed by any programme running on the device.
 - For testing of the items for quality and quantity add 'name:quantity,name:quantity' in the input text bar and press enter or click check_bill button.
 - Now your programme is running and you could use that to detect and analyse the products.
-

2. Detailed Database Schema and Configurations Template:

Purpose:

Ensure the database schema is well-documented, with clear instructions on how to set up, configure, and interact with the database.

Structure:

1. Database Schema Documentation:

- **Database Design:**
 - will be using 2 tables in the database called flipkart
 - name of the tables: items, fresh
 - items is created by using this following query

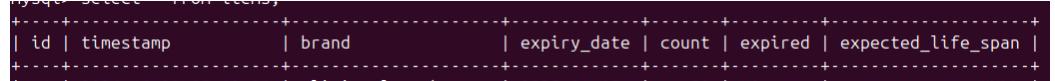
```
``CREATE TABLE items (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    timestamp DATETIME NOT NULL,  
    brand VARCHAR(255) NOT NULL,
```

```

    expiry_date DATE NOT NULL,
    count INT NOT NULL,
    expired ENUM('NA', 'yes') DEFAULT 'NA',
    expected_life_span INT DEFAULT NULL
);`

```

- This will create a table like this



id	timestamp	brand	expiry_date	count	expired	expected_life_span
----	-----------	-------	-------------	-------	---------	--------------------

- This following trigger can be used to make the other details auto fill during data insertion.

```

`DELIMITER //
CREATE TRIGGER update_inventory BEFORE INSERT ON inventory
FOR EACH ROW
BEGIN
    -- Calculate expected_life_span
    SET NEW.expected_life_span = DATEDIFF(NEW.expiry_date,
CURDATE());

```

```

    -- Determine if expired
    IF NEW.expiry_date < CURDATE() THEN
        SET NEW.expired = 'yes';
    ELSE
        SET NEW.expired = 'NA';
    END IF;
END;
//
DELIMITER ;`

```

- This is one example on how you can add an item into the table

```

INSERT INTO items (timestamp, brand, expiry_date, count)
VALUES ('2024-11-29T05:12:01', 'clinic_plus_shampoo', '2025-01-15', 50);`

```
- create the next table using the following sql query

```

`CREATE TABLE fresh (
    id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp DATETIME NOT NULL DEFAULT
CURRENT_TIMESTAMP,
    product VARCHAR(255) NOT NULL,
    freshness INT NOT NULL,
    expected_life_span INT NOT NULL
);`

```
- This query can be used to insert an item into the table

```

`INSERT INTO fresh (product, freshness, expected_life_span)
VALUES ('potato', 5, 12);`

```

2. Database Configuration:

- I have created a user with username proj and password [project@123](#) as password, you can use the same for the project or make your own and update the same in the code .

3.APIs

- link <http://127.0.0.1:8000/> this will return you the main index page.
- /external-update/= used to add items into the table
- /fetch-items/= used to reload and update the table in the frontend.
- /check-notifications/= used to get push down notification for alert for the user of the application .

3. Functional, Live Web Link for Real-Time Application Testing Template:

Structure:

1. Deployment Instructions:

- **Hosting Platform:**
 - Currently is deployed in the local system but not limited to it , can be hosted in any cloud of your own choice.
- **Training platform:**
 - run the training.ipynb with your own custom date either in cloud or in the in house system(ensure that the system has cuda cores access) for training of the model.

2. Live Application URL:

- A clear URL pointing to the live version of the app.
<http://127.0.0.1:8000/>
- live screen shot of detection and analysis

Billing Application

Add New Item

Name:

Quantity:

Expiry date:

Add Item

Item List

Name	Quantity	Expiry Date	Days Until Expiry
clinic_plus	2	2025-02-13	64
exo_bar	5	2025-03-07	86

Clear Table

Check Bill

Format: name:quantity,name:quantity

Check Bill

3. Testing and Feedback:

- The index page has testing for the gui which enables to check user for gui functionality.
- Postman can be used to analyse the api's of the GUI system.
- The detect.py has camera and cmd output system which can help user to test the detection and setting up the model for the best prediction.
- Basically both the frontend and the backend can run individually and can be tested individually

4. Maintenance and Updates:

- need to train the model on new data by using the training.ipynb file and updating the weights of the model in detection part.
-

Featured of the project

- Accuracy of the detection model has more than 92 % confidence at optimal lighting and by meeting the best system requirements .
- Brand detection: detection of the items is always correct and right with less than 1000 milliseconds delay and more than 95% + brand detection rate.
- Expiry date detection : with brand detection comes the expiry date detection , which is correct almost all the time which means around 100 % correct expiry date calculation.it also gives the number of days remaining for expiry for all the products ,even for fresh products.
- The counting of products is 100% accurate and it also covers multiple items, half seen items, partial or side-by-side placement of the items;
- freshness detection = once entered by user it has detection modules which tells the difference between good and bad vegetables/fruits and if fresh it also estimates the number of days it has to get spoilt.
- Alert push down notification which allows user to see if there is any bad,defective or expired item in the cart and change the product immediately.
- Automatic refresh of the items in the table , and billing testing input box for the user to test for the quality and quantity testing of the items.
- Allows user to either give input to add items, or use apis to push in the items into the table, also the user can use the text field to check the items (for quality and quantity), this text field can be attached to the flipkarts api to get user order details for faster and automated use of the software.

Technologies involved

- Django
- Html,css,Bootstrap,javascript
- Resnet(yolov7)

- MySql,SqlLite
- **error handling** = handled all errors for streamline usage of software
- **practicality**= This model and software system can be easily integrated with the existing system as it has support of latest technology and api's support for data exchange.
- **real_world_applicability**= This can be very easily applicable in any store with minute tweaks and turns and is applicable to many problems related to billing and packing in stores.

To reach out to me for any queries

email= subhamkonar84@gmail.com

phone= 8548898977