PEel Rescue

26 April 021 to 14 May 2021

# Overview

## Project Background and Description

|  |  |
| --- | --- |
|  | *Created by Nitu Balwani and Pallavi Limaye* |

Peel Rescue is an e-commerce shopping portal for buying/selling fruits peels to help reduce food wastage, and recover the abundance of nutrients present in fruit peel and put it to good use.

This project is built upon the concepts learnt at GoSchool during the period January - April 2021.

The theme of the project is to promote sustainability and reduce carbon footprint by choosing to recycle and reuse products such as a fruit peel that is normally just discarded without a second thought. This is one small step towards reducing the effects on our environment and climate change due to wastage of food.

The primary users of this online website are fruit peel generators such as fruit juice shop owners who can sell leftover fruit peels to several fruit peel consumers which are industries that are looking for high value nutrients to be used in their products.

## Project Scope

|  |  |
| --- | --- |
|  | *The Peel Rescue project aims to demonstrate our understanding of the concepts learnt during the modules:*  *Go Basic, Go Advanced, Go In Action I, Go In Action II, Microservices I and Microservices II* |

**Build an e-commerce portal to demonstrate the concepts learnt at GoSchool:**

* **Data structures**: Linked list for organizing cart items for buyer
* **Microservice**: Created a REST API with JSON for storing seller’s item information
* **Security**:
  + SSL/TLS Protocol for Application, and REST API
  + Login/Signup validation
  + Concurrent login detection
  + API keys
  + UUID for cookies
  + Hashed password, constant time verification of password
* **Concurrency, mutex** : Concurrency used for session management, and mutex for linked list
* **Error handling**: Reporting errors/warnings/info to log file
* **Containerization and database** : Docker and mySQL
* **Testing** : Basic DB testing, and Client Console application for testing REST API
* **Basic Knowledge of QA Test Cases**: Third party package Data-Dog SQL mock has been used to mock up database functionalities

## Diagram, schematic Description automatically generatedArchitecture

|  |  |
| --- | --- |
|  | *Technologies used: REST API with JSON, Docker and MySQL, Golang Templates with HTML,CSS* |

## Diagram Description automatically generatedDetailed Structure

|  |  |
| --- | --- |
|  | *Primary components of the project are the main application, and the REST API for storing seller items* |

## Application Setup Requirements and Deliverables

|  |  |
| --- | --- |
|  | *System Requirements: Microsoft Windows OS PC/ Command prompt console* |

### Procedure to execute the application in a Windows environment:

1. Copy the application folder “projectgolive” to a folder in PC’s Golang path e.g. D:/Golang/src/projectgolive
2. To execute them main application : Execute the application using source code with the command
3. Change directory to application
4. “go run main.go”
5. To execute them Seller REST API : Execute the API using source code with the command
6. Change directory to sellerAPI
7. “go run .”
8. Go source code files for main application: /application
9. main.go – The main file for Peel Rescue to start the application
10. config/\* – This package declares all constants, connects to user DB, read admin credentials from env file, declares common functions, create template handle, and handle for error log file, and info/warning/trace logs
11. start/\* - This package declares all handler functions using gorilla mux router and Listen and Serve using TLS
12. buyer/\* – This package contains the linked list data structure for storing shopping cart and handlers for all buyer related functions.
13. seller/\* – This package contains all handlers for all seller related functions
14. admin/\* – This package contains all handlers for all admin related functions
15. apiclient/\* – This package contains all functions used to communicate with REST API using JSON
16. server/\* – This package contains all handler functions used for registration, user/admin login and logout. It also contains session management functions, cookie creation and deletion, input validation and sanitization.
17. user\_db/\* - This package contains all functions to interact with database storing user data
18. templates/\* - This folder contains all templates used in the main application
19. assets/\* - This folder contains all the assets used in the main application
20. cert – This folder stores SSL certificate(cert.pem) and private key(key.pem)
21. log – This folder stores log file for error log
22. Go source code files for REST API: /sellerAPI
23. sellerAPI.go – The file contains all functions to handler HTTP requests such as POST/GET/PUT AND DELETE
24. sellerAPIdb.go – This file contains functions to interface with DB maintaining seller items
25. Docker Setup

**To create new database for recycle\_db : Single table for both buyers and sellers**

1. Open cmd terminal, and run the following commands:
2. docker stop recycle\_db
3. docker rm recycle\_db
4. docker run --name recycle\_db -p 33062:3306 -e MYSQL\_ROOT\_PASSWORD=password -d mysql:latest
5. mysql -P 33062 --protocol=tcp -u root -p
6. when prompted: Enter password: Enter ‘password’
7. On reaching mysql> prompt type the following commands:
8. CREATE database recycle\_db;
9. USE recycle\_db;
10. DROP TABLE IF EXISTS userdetails;
11. CREATE TABLE userdetails (Username VARCHAR(64) NOT NULL PRIMARY KEY, Password VARCHAR(128), Fullname VARCHAR(64), Isbuyer BOOLEAN, Phone VARCHAR(64), Address VARCHAR(128), Email VARCHAR(64));
12. SELECT \* FROM userdetails;

**To create new database for sellerAPI\_db : For storing seller items**

1. Open docker desktop , and run the following commands:
2. docker stop sellerAPIdb
3. docker rm sellerAPIdb
4. docker run --name sellerAPIdb -p 33061:3306 -e MYSQL\_ROOT\_PASSWORD=password -d mysql:latest
5. Open cmd terminal, and run the following commands:
6. mysql -P 33061 --protocol=tcp -u root -p
7. when prompted: Enter password: Enter ‘password’
8. On reaching mysql> prompt type the following commands:
9. USE sellerAPIdb;
10. DROP TABLE IF EXISTS `itemsdetails`;
11. CREATE TABLE `itemsdetails` (`Item` varchar(64) NOT NULL, `Quantity` int DEFAULT NULL, `Cost` float DEFAULT NULL, `Username` varchar(64) DEFAULT NULL;