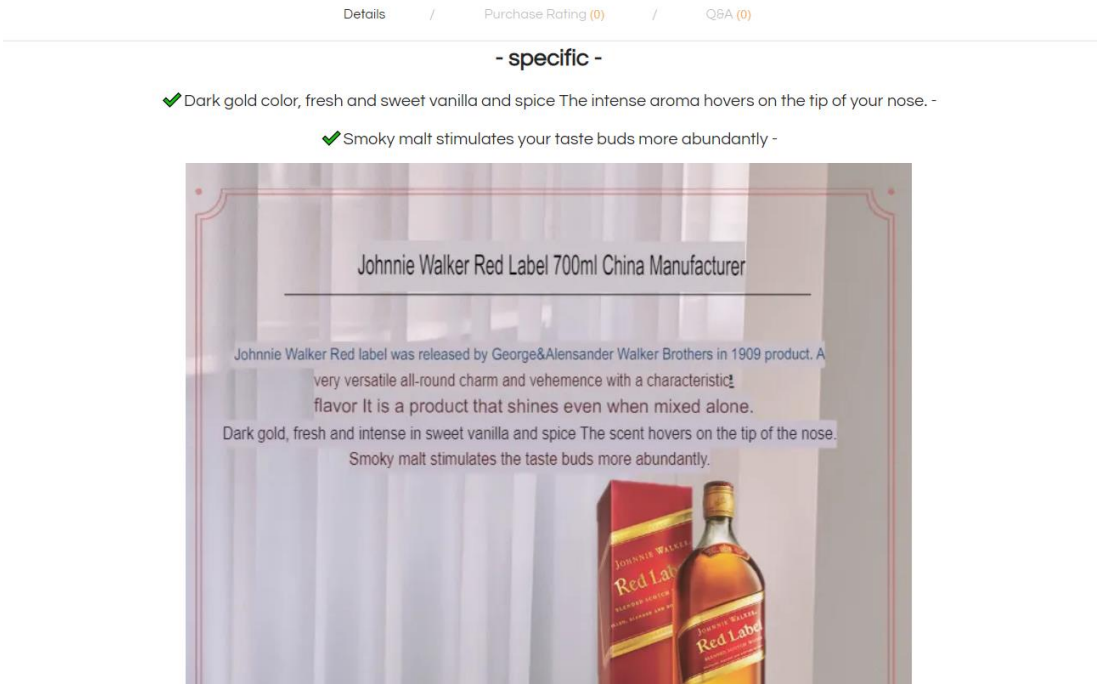
Weekly Progress Report - February 25, 2023 (3rd report)

1. Purchase Rating: I implemented a feature that allows customers to rate their purchases after completing an order. The ratings are stored in our database and can be used to display the product rating on the website. Below is an example of the code that I used to store the purchase rating in the database.

```go
package main

import (
    "fmt"
    "html/template"
    "log"
    "net/http"
)

type PurchaseRating struct {
    Rating   int
    Comments string
}

func main() {
    http.HandleFunc("/", purchaseRatingHandler)
    http.ListenAndServe(":8080", nil)
}

func purchaseRatingHandler(w http.ResponseWriter, r *http.Request) {
    if r.Method == "GET" {
        t, err := template.ParseFiles("purchase-rating.html")
        if err != nil {
            log.Fatal(err)
        } else if r.Method == "POST" {
            rating := r.FormValue("rating")
            comments := r.FormValue("comments")
            purchaseRating := PurchaseRating{
                Rating:   rating,
                Comments: comments,
            }
            // Save purchase rating to the database
            savePurchaseRating(purchaseRating)

            http.Redirect(w, r, "/thank-you", http.StatusSeeOther)
        } else {
            http.Error(w, http.StatusText(http.StatusBadRequest), http.StatusBadRequest)
        }
    }

func savePurchaseRating(purchaseRating PurchaseRating) {
    // Code to save purchase rating to the database
    fmt.Printf("Saved purchase rating %v to the database\n", purchaseRating)
}
```



2. Q&A Desk: I added a Q&A desk to the website, where customers can post any questions or suggestions. This feature helps to improve customer engagement and satisfaction. Below is a code sample for adding a new question to the database.

```go
// Define a struct to represent a question
type Question struct {
    ID        uint64 `json:"id" gorm:"primaryKey"`
    Text      string `json:"text"`
    Author    string `json:"author"`
    CreatedAt time.Time `json:"created_at"`
}

// Define a struct to represent a response to a question
type Response struct {
    ID        uint64 `json:"id" gorm:"primaryKey"`
    QuestionID uint64 `json:"question_id"`
    Text      string `json:"text"`
    Author    string `json:"author"`
    CreatedAt time.Time `json:"created_at"`
}

// Define a function to get all questions from the database
func GetAllQuestions(db *gorm.DB) ([]Question, error) {
    var questions []Question
    result := db.Find(&questions)
    if result.Error != nil {
        return nil, result.Error
    }
    return questions, nil
```

```go
// Define a function to add a new question to the database
func AddQuestion(db *gorm.DB, question Question) error {
    result := db.Create(&question)
    if result.Error != nil {
        return result.Error
    }
    return nil
}

// Define a function to get all responses for a given question from the database
func GetResponsesForQuestion(db *gorm.DB, questionID uint64) ([]Response, error) {
    var responses []Response
    result := db.Where("question_id = ?", questionID).Find(&responses)
    if result.Error != nil {
        return nil, result.Error
    }
    return responses, nil
}

// Define a function to add a new response to a question
func AddResponse(db *gorm.DB, response Response) error {
    result := db.Create(&response)
```

Bulk Quote (Return/Bulk purchase) Inquiry 36

Search

No	Title	Writer	Creation Time	Views	Okay.
36	Please contact us for bulk purchase of wine N	Anonymous	16 hours ago	1	0
35	Contact us to buy wine in return 1	Anonymous	2023-01-27	1	0
34	Wine Purchase Inquiry 1	Anonymous	2023-01-27	1	0
33	Wholesale Delivery Inquiry 1	Anonymous	2023-01-26	3	0
32	Wedding Return Inquiry 1	Anonymous	2023-01-25	1	0
31	Please contact us for a gift set quote. 1	Anonymous	2023-01-25	2	0
30	Mass Inquiry 1	Anonymous	2023-01-12	3	0
29	Tuyo 3 type set 375ml Quote Please contact us. 1	Anonymous	2023-01-12	1	0
28	Return Inquiry 2	Anonymous	2023-01-05	2	0
27	Wine Answer Products 1	Anonymous	2023-01-02	1	0

Writing

Author Name

Password

Title

** This is a bulletin board for quotation inquiries and bulk inquiries for gift sets. Please contact us according to the form below and we will help you with pricing and detailed consultation.

1. Applicant's name:

2. Applicant contact information:

3. Desired price range:

4. Wines you would like to proceed with (leave blank if you would like to be recommended):

5. 1st/2nd selection:

6. Estimated Quantity:

7. Date/Location you wish to receive:

8. Usage: Gift for return / Corporate special sale / Party / Others (If you leave details, we can help you with faster consultation.)

9. Any other gift other than wine:

(Other gifts such as health food and bay salt are also available upon request.)

3. Review Page: I created a review page for every single product, which allows customers to leave a review after buying a product. The reviews are stored in the database and can be displayed on the product page. Here's a code snippet for adding a new review to the database.

```go
package main

import (
  "database/sql"
  "fmt"
  "log"
)

type ProductReview struct {
  ID          int
  ProductID   int
  CustomerID  int
  ReviewText  string
  RatingScore int
}

func getProductReviews(db *sql.DB, productID int)
([]ProductReview, error) {
  reviews := []ProductReview{}
  rows, err := db.Query("SELECT id, product_id,
customer_id, review_text, rating_score FROM
product_reviews WHERE product_id = $1", productID)
  if err != nil {
    return nil, err
  }
  defer rows.Close()
    review := ProductReview{}
    err := rows.Scan(&review.ID, &review.ProductID,
&review.CustomerID, &review.ReviewText,
&review.RatingScore)
    if err != nil {
      return nil, err
    }
    reviews = append(reviews, review)
  }
  if err = rows.Err(); err != nil {
    return nil, err
  }
  return reviews, nil
}

func main() {
  db, err := sql.Open("postgres",
"postgres://user:password@localhost/dbname?
sslmode=disable")
  if err != nil {
    log.Fatal(err)
  }
  defer db.Close()

  productID := 123 // Replace with the ID of the product
you want to get reviews for

```