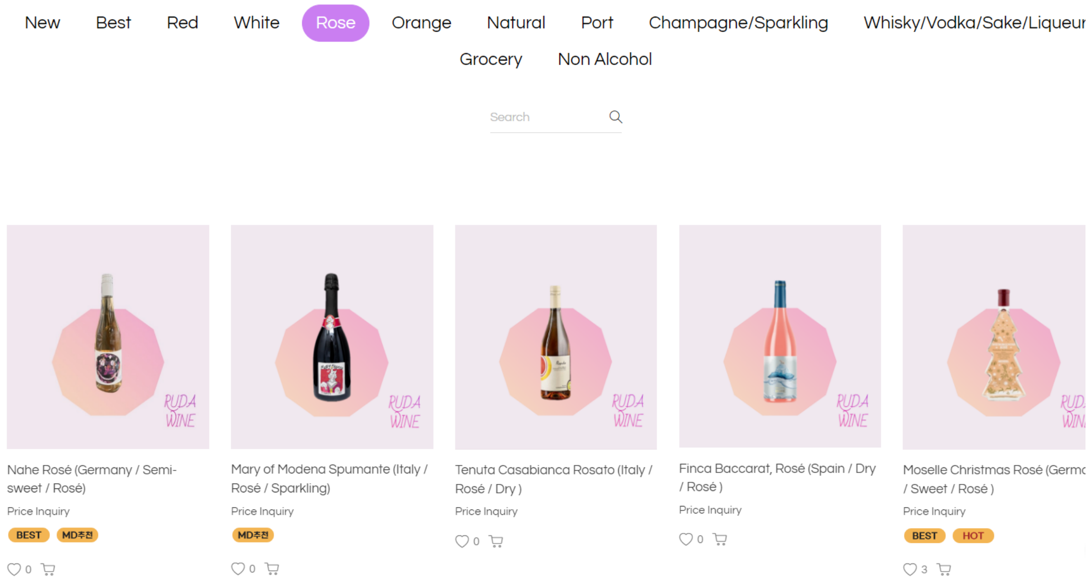
Weekly Progress Report – March 3, 2023 (4th report)

Done for this week: implementation of the wine list.

During this week, the wine list functionality was added to the project. There are 12 different catalogs in the wine list: New, Best, Red, White, Rose, Orange, Natural, Port, Champagne, Whisky/Vodka, Grocery, and Non-Alcohol.



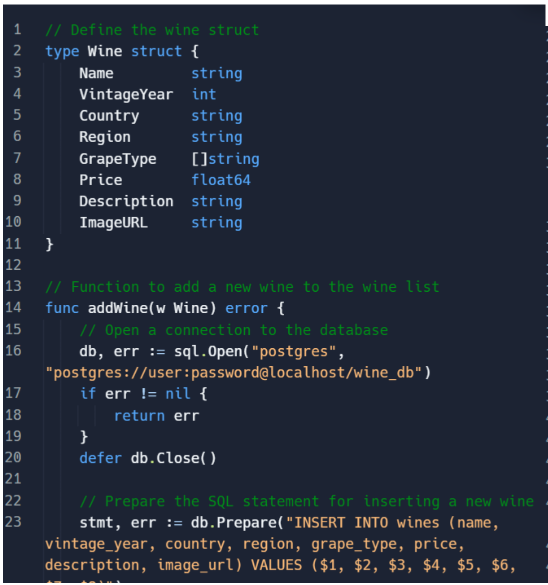
Nahe Rosé (Germany / Semi-sweet / Rosé)
Price Inquiry

Mary of Modena Spumante (Italy / Rosé / Sparkling)
Price Inquiry

Tenuta Casabianca Rosato (Italy / Rosé / Dry)
Price Inquiry

Finca Baccarat, Rosé (Spain / Dry / Rosé)
Price Inquiry

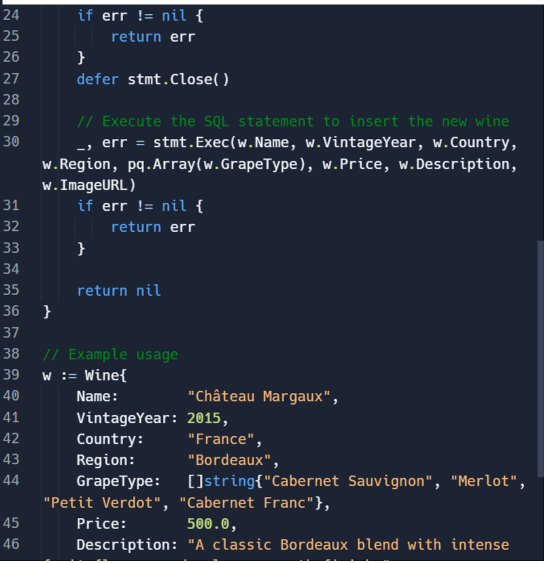
Moselle Christmas Rosé (Germa / Sweet / Rosé)
Price Inquiry



```go
// Define the wine struct
type Wine struct {
    Name        string
    VintageYear int
    Country     string
    Region      string
    GrapeType   []string
    Price       float64
    Description string
    ImageURL    string
}

// Function to add a new wine to the wine list
func addWine(w Wine) error {
    // Open a connection to the database
    db, err := sql.Open("postgres", "postgres://user:password@localhost/wine_db")
    if err != nil {
        return err
    }
    defer db.Close()

    // Prepare the SQL statement for inserting a new wine
    stmt, err := db.Prepare("INSERT INTO wines (name, vintage_year, country, region, grape_type, price, description, image_url) VALUES ($1, $2, $3, $4, $5, $6,
```

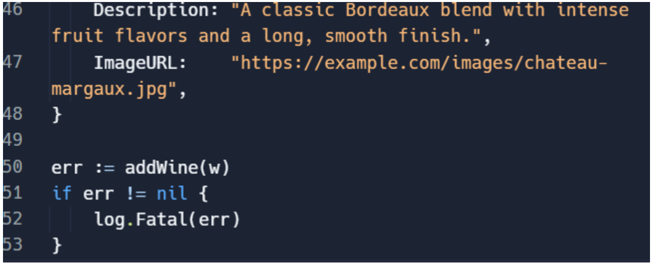


```go
    if err != nil {
        return err
    }
    defer stmt.Close()

    // Execute the SQL statement to insert the new wine
    _, err = stmt.Exec(w.Name, w.VintageYear, w.Country, w.Region, pq.Array(w.GrapeType), w.Price, w.Description, w.ImageURL)
    if err != nil {
        return err
    }

    return nil
}

// Example usage
w := Wine{
    Name:        "Château Margaux",
    VintageYear: 2015,
    Country:     "France",
    Region:      "Bordeaux",
    GrapeType:   []string{"Cabernet Sauvignon", "Merlot", "Petit Verdot", "Cabernet Franc"},
    Price:       500.0,
    Description: "A classic Bordeaux blend with intense
```

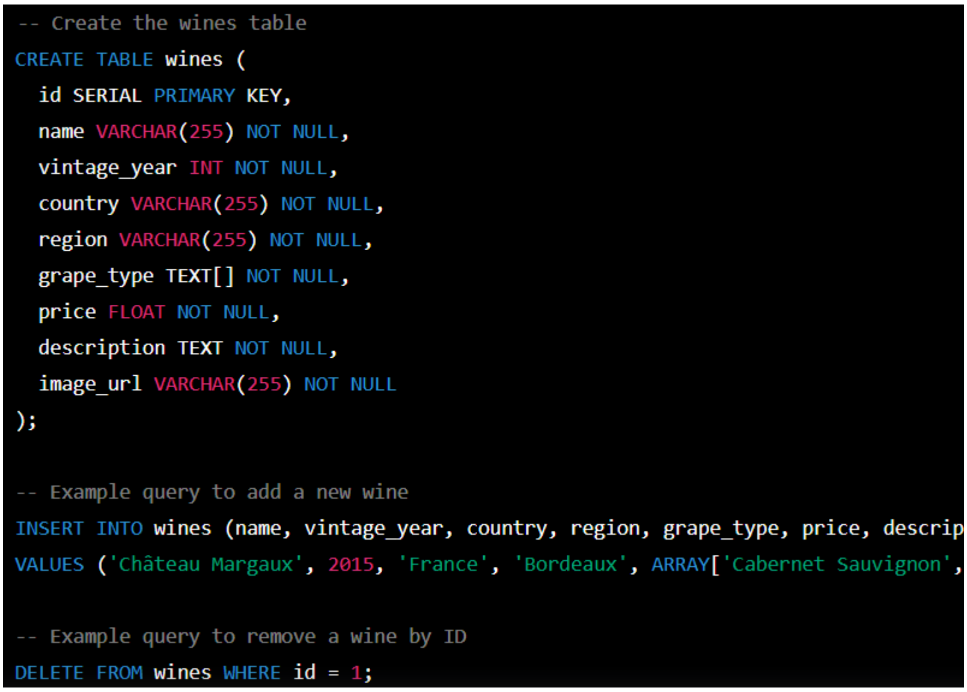


```go
    Description: "A classic Bordeaux blend with intense fruit flavors and a long, smooth finish.",
    ImageURL:    "https://example.com/images/chateau-margaux.jpg",
}

err := addWine(w)
if err != nil {
    log.Fatal(err)
}
```

The implementation of the wine list involved creating the necessary data structures to store the wine information, as well as the necessary database tables to store this information persistently. The necessary endpoints were also implemented to allow users to view the wine list, as well as filter and sort the wines according to various criteria.

The admin panel was updated to include functionality for managing the wine list, including adding, editing, and deleting wines. The necessary database tables and queries were also updated to support this functionality.



```sql
-- Create the wines table
CREATE TABLE wines (
  id SERIAL PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  vintage_year INT NOT NULL,
  country VARCHAR(255) NOT NULL,
  region VARCHAR(255) NOT NULL,
  grape_type TEXT[] NOT NULL,
  price FLOAT NOT NULL,
  description TEXT NOT NULL,
  image_url VARCHAR(255) NOT NULL
);

-- Example query to add a new wine
INSERT INTO wines (name, vintage_year, country, region, grape_type, price, descrip
VALUES ('Château Margaux', 2015, 'France', 'Bordeaux', ARRAY['Cabernet Sauvignon',

-- Example query to remove a wine by ID
DELETE FROM wines WHERE id = 1;
```