# Golang MID-2
# PROGRESS REPORT

Student: Aslan Tasmagambetov
Id: 190103012

The following functions have been added to the project:

1) Filtering items based on price, rating
2) Giving rating for items
3) Commenting items

"Item" contain fields for the item's ID, name, price, and rating:

```
type Item struct {
    ID     int
    Name   string
    Price  float64
    Rating float64
}
```

The "AddRaitong()" method updates the item's rating based on a new rating given by the user, and the "AddReview()" method adds a review to the database for the item:

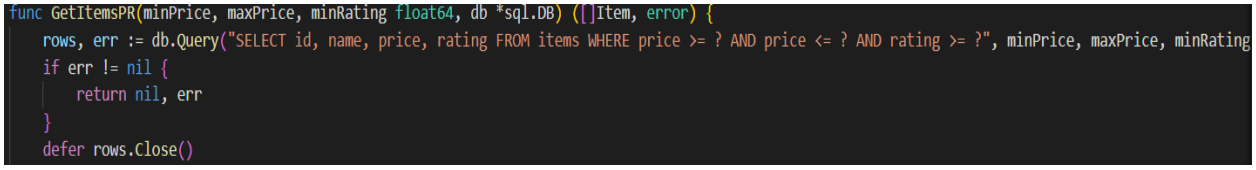
```
func (i *Item) AddRating(rating float64, db *sql.DB) error {
    row := db.QueryRow("SELECT rating, num_ratings FROM items WHERE id = ?", i.ID)
    var currentRating float64
    var numR int
    err := row.Scan(&currentRating, &numR)
    if err != nil {
        return err
    }
```

The "AddRating()" method first queries the database to get the item's current rating and number of ratings. It then calculates the new rating based on the current rating, number of ratings, and the user's new rating. Finally, it updates the database with the new rating and number of ratings, and updates the "Item" struct's "Rating" field.

The "AddReview()" method simply adds a new review to the database for the item.

The “GetItemsPR()” function takes four parameters: minPrice, maxPrice, minRating, and db.

minPrice and maxPrice represent the minimum and maximum prices for the items to be returned. minRating represents the minimum rating for the items to be returned.



```go
func GetItemsPR(minPrice, maxPrice, minRating float64, db *sql.DB) ([]Item, error) {
    rows, err := db.Query("SELECT id, name, price, rating FROM items WHERE price >= ? AND price <= ? AND rating >= ?", minPrice, maxPrice, minRating
    if err != nil {
        return nil, err
    }
    defer rows.Close()
```