

# Inventory Management System App for Grocery Stores

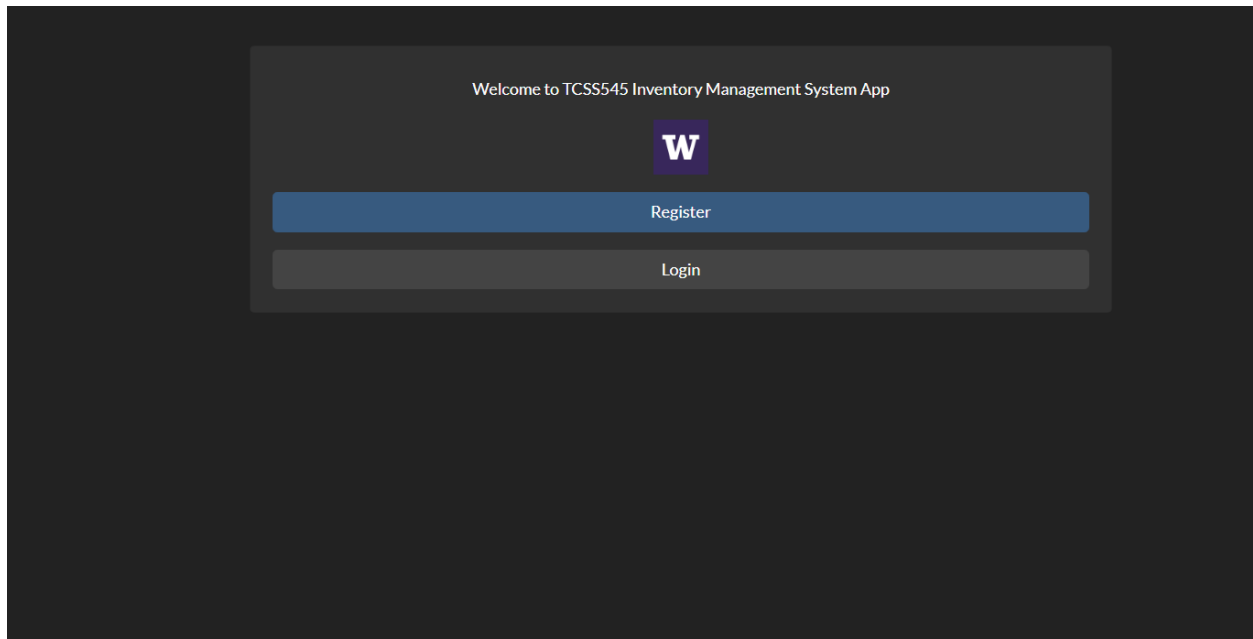
## TCSS545 A - Database Design - Group #7

[adams42@uw.edu](mailto:adams42@uw.edu), [deepthiw@uw.edu](mailto:deepthiw@uw.edu), [sumitr@uw.edu](mailto:sumitr@uw.edu)

### TEST CASES

#### Welcome Page:

Welcome Page is the first page in our application where we can see Login and Register buttons. If a new user wants to access the application, they can click on Register to register themselves to the application. All the existing users can click on the Login option and access the application using username and password.



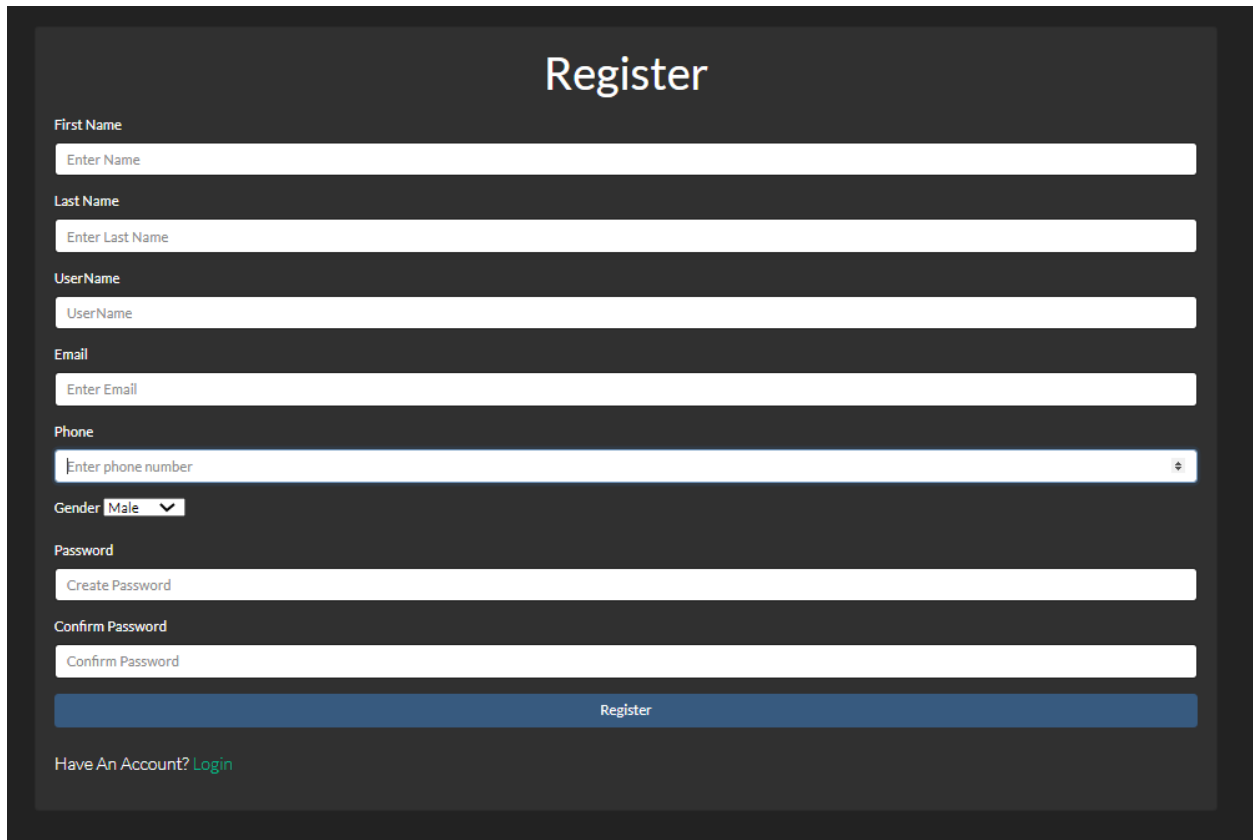
#### Register Page:

Register page helps users to register themselves to access the application. This page validates input data from the user and allows inside the application only if the user meets the required conditions. The conditions are,

1. All the fields in the register page should be filled.
2. Password and confirm password should be the same.

3. Should not enter incorrect details to the field. For example, expected input for the phone field is numbers, application prevents users from entering alphabets in the phone number field.

We can see the screenshot of the Register page below,



The screenshot shows a 'Register' page with a dark background. The title 'Register' is centered at the top in a large, white, sans-serif font. Below the title, there are several input fields, each with a label above it: 'First Name' (with placeholder 'Enter Name'), 'Last Name' (with placeholder 'Enter Last Name'), 'UserName' (with placeholder 'UserName'), 'Email' (with placeholder 'Enter Email'), and 'Phone' (with placeholder 'Enter phone number'). The 'Phone' field has a small icon on the right side. Below the 'Phone' field is a 'Gender' dropdown menu with 'Male' selected. Below the 'Gender' dropdown are two more input fields: 'Password' (with placeholder 'Create Password') and 'Confirm Password' (with placeholder 'Confirm Password'). At the bottom of the form is a blue button labeled 'Register'. Below the button, there is a link that says 'Have An Account? Login'.

Let us see the messages the user gets when failing to satisfy the above-mentioned conditions.

1. Expected outcome when user failed to satisfy the first condition:

## Register

First Name  
Sumitha ✓  
Valid.

Last Name  
Ravindran ✓  
Valid.

UserName  
UserName ⓘ  
Please fill out this field.

Email  
Enter Email ⓘ  
Please fill out this field.

Phone  
Enter phone number ⓘ  
Please fill out this field.

Gender Male ▾

Password  
Create Password ⓘ  
Please fill out this field.

Confirm Password  
Confirm Password ⓘ  
Please fill out this field.

Register

Have An Account? [Login](#)

2. Expected outcome when user failed to satisfy the second condition:  
The application will through a message saying that “The password and confirm password do not match”

## Register

The password and the confirm password do not match

3. Expected outcome when user failed to satisfy the third condition:

## Register

The password and the confirm password do not match

First Name

Valid.

Last Name

Valid.

UserName

Valid.

Email

Valid.

Phone

Please fill out this field.

Gender Female

Password

Valid.

Confirm Password

Valid.

Register

Have An Account? [Login](#)

When a user enters correct inputs, then the application allows the user to register and redirects to the login page with “Manager added successfully. Please login now” message.

## Login

Manager added successfully. Please login now.

Username

Password

Login

No Account? [Register](#)

If an existing user tries to register again, then he/she will be redirected to the login page with “Manager already exists in DB” message.

The screenshot shows a login page with a dark background. At the top, the word "Login" is displayed in a large, white, sans-serif font. Below it, a red error message reads "Manager already exists in DB!". There are two input fields: "Username" with the placeholder text "Enter Username" and "Password" with the placeholder text "Enter Password". A blue "Login" button is positioned below the password field. At the bottom, there is a link that says "No Account? Register" in green text.

### Login Page:

This page allows only registered users to login. If a non-registered user or a user enters incorrect username or password and tries to login, the application would not allow them to login.

The screenshot shows the same login page as above, but with a different error message in red: "Invalid username/password. Please enter the right credentials / register if you are a new user." The layout, including the "Login" title, input fields, button, and "Register" link, remains the same.

When a user enters the correct/existing username and password, the application will allow them to access the main page which consists of operations related to inventory management system.

### Main Page:

# Welcome to TCSS-545 Inventory Management System

[View Items](#)[Add Item](#)[Delete an Item](#)[Update ItemPrice](#)[View price hike](#)[Logout](#)

We can perform six different operations using this main page, they are

1. View Items
2. Add item
3. Delete an item
4. Update itemPrice
5. View price hike
6. Log out

## View Items page:

The manager can see all the items in the inventory when View Items Button is clicked.

## Items Available in the Inventory

S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	IPhone	Electronics	75	299.99
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	Grapes	Fruits	200	1.99
6	Orange	Fruits	40	2.99
7	Beetroot	Vegetables	60	6
8	Carrot	Vegetables	100	5
9	Cilantro	Vegetables	150	0.49
10	Green Beans	Vegetables	45	2.99

[Return to the Main Page](#)

## Add item Page:

The manager can add an item to the inventory item list by clicking on the **Add item** button in the main page.

This application allows the user to add a new item to the list.  
For example: If the manager needs to add Cherries to the inventory and it is in an existing category “Fruits”, the manager can do so using the form below. Once clicked on “add” the item gets added to the inventory and gets listed in the View Items Page.

## Add an Item

Item Name

cherries

Valid.

Category

Fruits

Valid.

Quantity

25

Valid.

Price Per Item

4.99

Valid.

Submit

[Return to the Main Page](#)

## Items Available in the Inventory

S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	IPhone	Electronics	75	330
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	cherries	Fruits	25	4.99
6	Grapes	Fruits	200	1.99
7	Orange	Fruits	40	2.99
8	Chocobar	Ice creams	100	15
9	Beetroot	Vegetables	60	6
10	Cilantro	Vegetables	150	0.49

Managers can also enter an item to a **new category** that does not exist until now in the category table. The new category will be added to the category table using the trigger called “**category Trig**”.

As You can see in the screenshot below, I have added a new item “Chocobar” under the new category name “Ice creams”. Once the manager clicks on the submit button, the app will redirect to view items page where the manager can see the item has been added to the list.

## Add an Item

Item Name

Category

Quantity

Price Per Item

[Submit](#)

[Return to the Main Page](#)

## Items Available in the Inventory

S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	iPhone	Electronics	75	299.99
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	Grapes	Fruits	200	1.99
6	Orange	Fruits	40	2.99
7	Chocobar	Ice creams	100	2.99
8	Beetroot	Vegetables	60	6
9	Carrot	Vegetables	100	5
10	Cilantro	Vegetables	150	0.49
11	Green Beans	Vegetables	45	2.99

[Return to the Main Page](#)

When a manager tries to add an existing item into the list, the app would not allow him to do it.



## Item already Exists in the Inventory

[Return to the Main Page](#)

### Delete an item Page:

Manager can also delete an item by clicking on the **Delete an item** button. On a successful delete the app will redirect to view items page to see that the item has been deleted.

For example, let us delete carrots from the list.

## Delete an Item Name

Item Name:

Submit

[Return to the Main Page](#)

We just deleted carrot from the list, if the manager tries to delete the item “carrot” again, the app will throw a “Item does not exist. Please check the item name” message.

## Item doesnot Exist. Please check the Item Name

[Return to the Main Page](#)

### Update item price Page:

Manager can update the price of an item using the **Update itemPrice** button. The updated prices are reflected in the view items page. We have used a trigger called “**ItemPriceTrig**” to check and update “**RipoffItems**” table only if the price of an updated item raised by more than \$5.

For example: If a manager tries to update an existing item “Chocobar” price by \$5 then it will reflect in the view items page but not in RipoffItems table because the raised price is less than \$5, the old price was \$2.99 and the updated price \$5.

## Update the Price of an Item

Item Name

chocobar

Price

\$

Submit

[Return to the Main Page](#)

## Items Available in the Inventory

S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	IPhone	Electronics	75	299.99
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	Grapes	Fruits	200	1.99
6	Orange	Fruits	40	2.99
7	Chocobar	Ice creams	100	5
8	Beetroot	Vegetables	60	6
9	Cilantro	Vegetables	150	0.49
10	Green Beans	Vegetables	45	2.99

[Return to the Main Page](#)

Let us check the RipoffItems table in database,

```
6 • select * from RipoffItems;
```

Result Grid					
Filter Rows:					
Edit:					
Export/Import:					
id	itemName	oldPrice	pricePerItem	categoryName	updatedOn
NULL	NULL	NULL	NULL	NULL	NULL

Now raise the price of “chocobar” to \$15, then it should reflect in both the view items page and Rip Off Items table.

## Update the Price of an Item

Item Name

Price

Submit

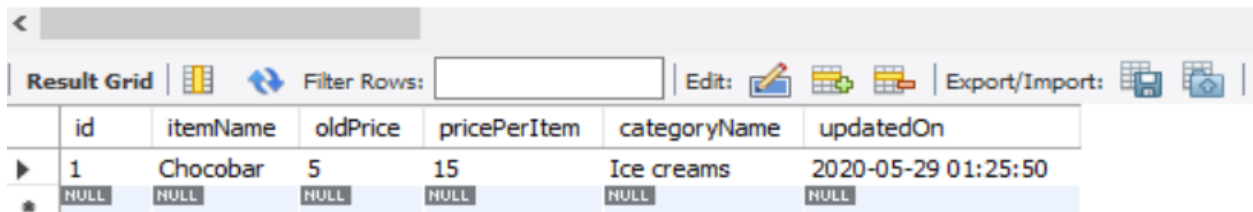
[Return to the Main Page](#)

## Items Available in the Inventory

S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	IPhone	Electronics	75	299.99
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	Grapes	Fruits	200	1.99
6	Orange	Fruits	40	2.99
7	Chocobar	Ice creams	100	15
8	Beetroot	Vegetables	60	6
9	Cilantro	Vegetables	150	0.49
10	Green Beans	Vegetables	45	2.99

[Return to the Main Page](#)

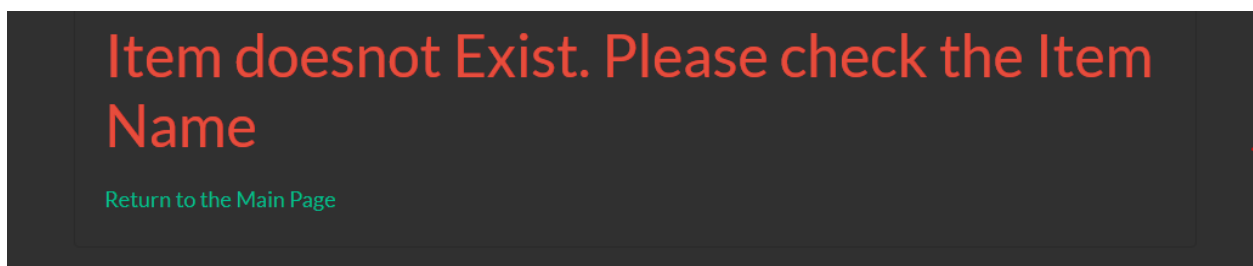
```
6 • select * from RipoffItems;
```



The screenshot shows a database application interface. At the top, there's a toolbar with icons for 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. Below the toolbar is a table with the following columns: id, itemName, oldPrice, pricePerItem, categoryName, and updatedOn. The first row contains the values: 1, Chocobar, 5, 15, Ice creams, and 2020-05-29 01:25:50. The second row contains NULL values for all columns.

id	itemName	oldPrice	pricePerItem	categoryName	updatedOn
1	Chocobar	5	15	Ice creams	2020-05-29 01:25:50
NULL	NULL	NULL	NULL	NULL	NULL

If the manager tries to update an item which is not in the list, the app will throw a “Item does not exist. Please check the item name” message.



### Price hike view Page:

The priceHike View is used to display the items that have had a priceHike of more than 20\$. This would help in further analysis to prevent price gouging. To view price hike view, click on **price hike view** button.



The screenshot shows a form titled 'Update the Price of an Item' in a large, bold, blue font. The form has two input fields: 'Item Name' with the value 'iphone' and 'Price' with the value '330'. Below the input fields is a blue 'Submit' button. At the bottom of the form, there is a green link that says 'Return to the Main Page'.

Items Available in the Inventory				
S.No	Item Name	Category	Quantity	Price Per Item
1	Ipad	Electronics	20	399.99
2	IPhone	Electronics	75	330
3	Apple	Fruits	50	2
4	Banana	Fruits	100	0.99
5	Grapes	Fruits	200	1.99
6	Orange	Fruits	40	2.99
7	Chocobar	Ice creams	100	15
8	Beetroot	Vegetables	60	6
9	Cilantro	Vegetables	150	0.49
10	Green Beans	Vegetables	45	2.99

[Return to the Main Page](#)

Items in price hike view				
id	Item Name	Price Hike	Category Name	Updated On
1	IPhone	30.009999999999999	Electronics	Fri May 29 2020 01:36:02 GMT-0700 (Pacific Daylight Time)

[Return to the Main Page](#)

### Logout button:

Once done with all the changes, the manager can logout from the app using the **logout** button. Then the app will redirect to the “**Welcome**” page.

