|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6197 | ISYS6197003 | ISYS6749052  Business Application Development |
| **Computer Science** | **O252-ISYS6197-LC11702-00** |
| ***Valid on*** *Odd Semester Year 2024/2025* | **Revision 00** |

1. Kelompok tidak diperkenankan untuk:

*Members of the group are prohibited from:*

* + - Melihat sebagian atau seluruh jawaban kelompok lain,

*Seeing a part or the whole answer from other groups,*

* + - Menyadur sebagian atau seluruh jawaban dari buku, catatan, video, dan jenis referensi lainnya,

*Retell a part or the whole answer from books, notes, videos, and other references,*

* + - Menyadur sebagian atau seluruh jawaban dari internet,

*Retell a part or the whole answer from the internet,*

* + - Mengumpulkan jawaban yang tidak sesuai dengan tema soal,

*Submitting an answer with a different theme from the given case,*

* + - Melakukan tindakan yang menyebabkan jawaban dicontek oleh orang lain atau kelompok lain, baik disengaja maupun tidak disengaja,

*Doing action that could result the answer being copied by someone or other groups, intentionally or unintentionally,*

* + - Melakukan tindakan kecurangan lainnya.

*Committing other dishonest actions.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dicantumkan pada butir ke-1, maka nilai mahasiswa dan/atau kelompok yang melakukan kecurangan, baik menyontek atau dicontek, akan dinolkan sesuai dengan peraturan yang berlaku.

*If it has been proven that a group has committed dishonest actions outlined in point 1 above, the whole groups related to the incident, regardless of which one copies or has their answer copied, will be issued a score of zero according to the regulation.*

1. Jawaban yang dapat diterima dan dinilai adalah jawaban yang dikumpulkan sebelum batas waktu yang telah ditentukan.

*The answer must be submitted before the designated deadline to be accepted and graded,*

1. Jawaban akan dinilai berdasarkan teknik atau metode yang diajarkan pada kelas praktikum dengan menggunakan software yang sudah ditentukan.

*The scoring will be based on the materials taught during the practicum classes using the designated software. Using different software than requested may result in your answer not being graded.*

1. Jika Anda tidak membaca peraturan ini, maka Anda dianggap sudah membaca dan menyetujuinya.

*By taking this exam, you agree to these regulations, regardless of whether you have read it or not.*

1. Persentase penilaian untuk matakuliah ini adalah sebagai berikut:

*The score will be distributed as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Perangkat lunak yang digunakan pada matakuliah ini adalah sebagai berikut:

*This course uses the following software:*

|  |
| --- |
| **Software**  *Software* |
| Eclipse 2020.6 R  Java 11.0.18  JavaFX 17.0.7  MySQL Java Connection Library 8.0.24  XAMPP 8.0.7 |

1. Ekstensi file yang harus dikumpulkan untuk matakuliah ini adalah sebagai berikut:

*Your answers must be in the following file extensions:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| JAVA. CLASS | JAVA. CLASS, SQL | - |

1. File yang harus dikumpulkan adalah keseluruhan jawaban beserta dengan aset yang digunakan (gambar, audio, video, dll) dan dokumentasi proyek yang berisikan link referensi aset dan penjelasan mengenai aplikasi yang dibuat (terlampir bersama dengan soal).

*Include other files that can support your project, such as: all files in your project, other files (image, audio, video, etc.) used in your project, \*.doc file (documentation of your project) that contains all pages in your project, reference links of additional files (image, audio, video, etc.) used in your project, the description about how to use your application, etc.*

## Soal

*Case*

**GoGoQuery**

**GoGoQuery** Store is a thriving retail business managed by ElevenSeven Enterprise. Known for its wide range of products and exceptional customer service, **GoGoQuery** Store has seen a significant rise in demand, leading to a bustling store that's often too full. To address these challenges and streamline operations, **GoGoQuery** Store needs a simple application to cater to both customers and staff. The application itself will have two primary roles: **Shopper** and **Manager**. **Shoppers** can browse products, manage their shopping carts, and make purchase. Managers, on the other hand, can oversee inventory management and handle order fulfillment.

As an intermediate programmer, you are asked to develop an online shopping application using **JavaFX** Programming Language and **MySQL** database. Make sure to **not use** **regex, Java FXML, JavaFX Scene Builder, or any library outside the class materials** while making this application.

A diagram of a computer

Description automatically generated

**Figure 1. ERD for GoGoQuery**

* + - 1. **Login Page**

A login screen with text

Description automatically generated

**Figure 2. Login Page**

In the **Login Page**, users can **login** and access the application. To login, users must enter their credentials by entering email and password. The system will execute several **validations** before the user can **successfully login**.

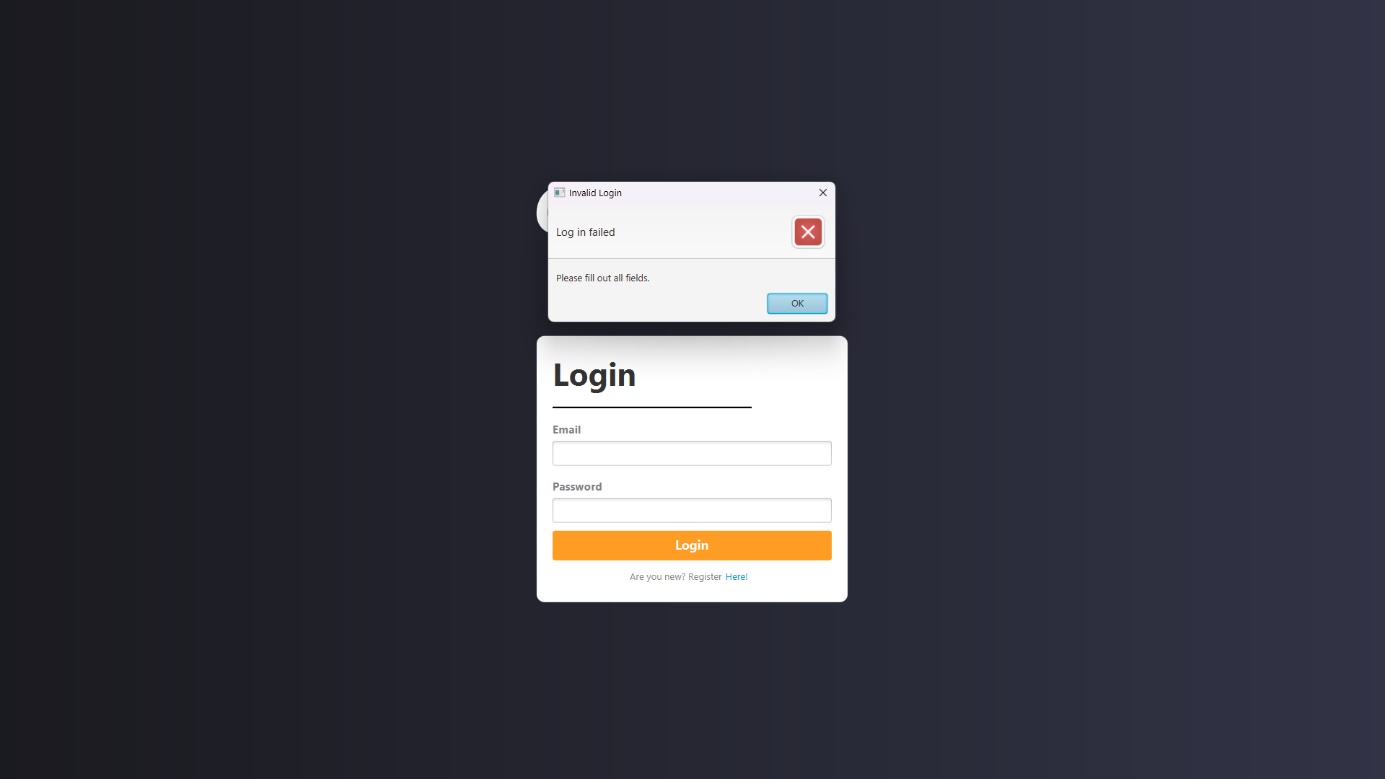
**Validations:**

* Email and Password **cannot be empty**
* The credentials must be **valid**.

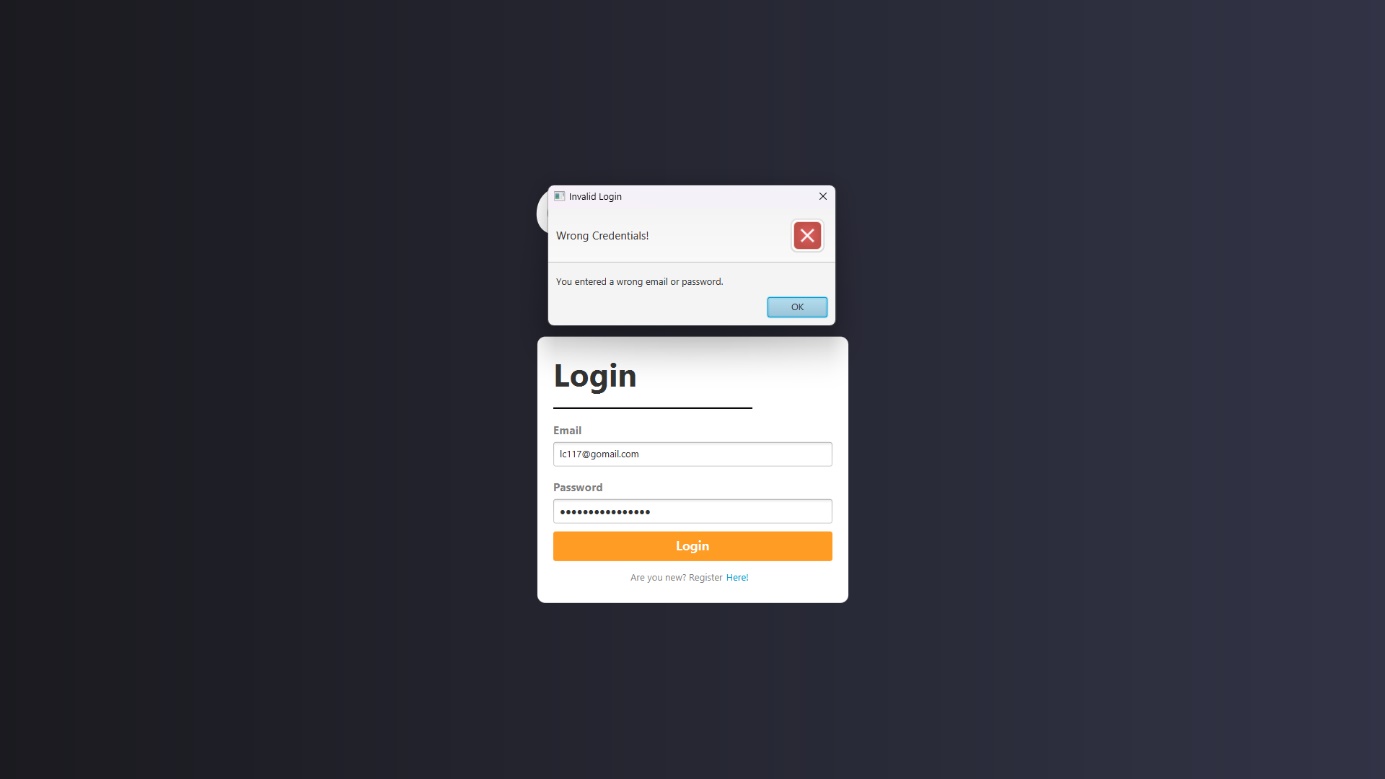
Pressing the **Login Button** or pressing **Enter Key** inside a text field and password field will trigger these validations. Then, if the validation is passed, the user will be **redirected** to the **Home Page**. A **failed login** will display an **Alert**, displaying the type of **error**.

***Components:***

* **Label** for Login label, Email label, Password label, and Register Here
* **Text Field** for Email input
* **Password Field** for Password input
* **Button** for Login button
* **Alert** to display errors

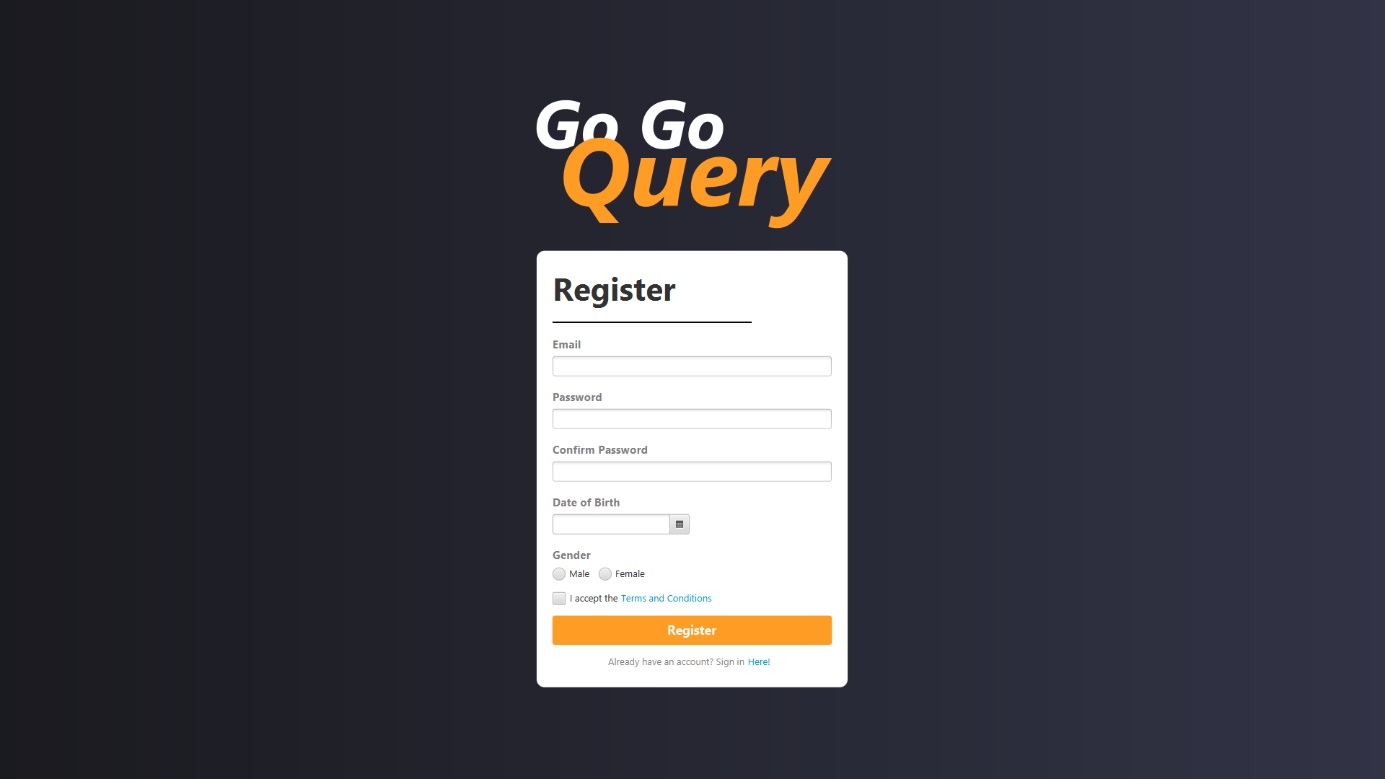


**Figure 3. Login empty field error**



**Figure 4. Wrong credentials error**

1. **Register Page**



**Figure 5. Register Page**

In the **Register Page**, users can **register** and make a new account. To make a new account, users must enter their **email, password,** and **date of birth**. A **successful registration** must pass several **validations**.

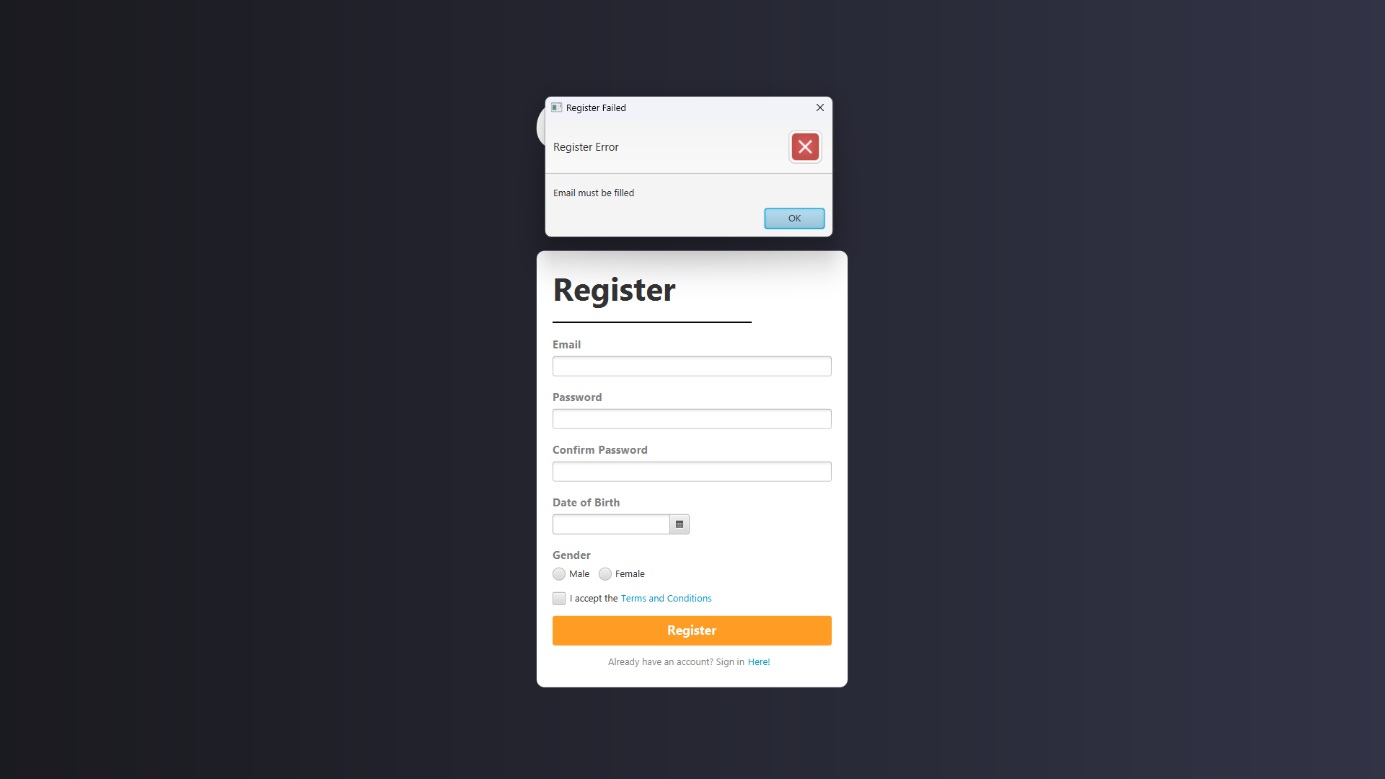
***Components:***

* **Label** for Register label, Email label, Password label, Date of Birth label, and Login Here
* **Text Field** for Email input
* **Password Field** for Password and Confirm Password input
* **Date Picker** for Date of Birth input
* **Radio Button** for Gender input
* **Alert** to display errors

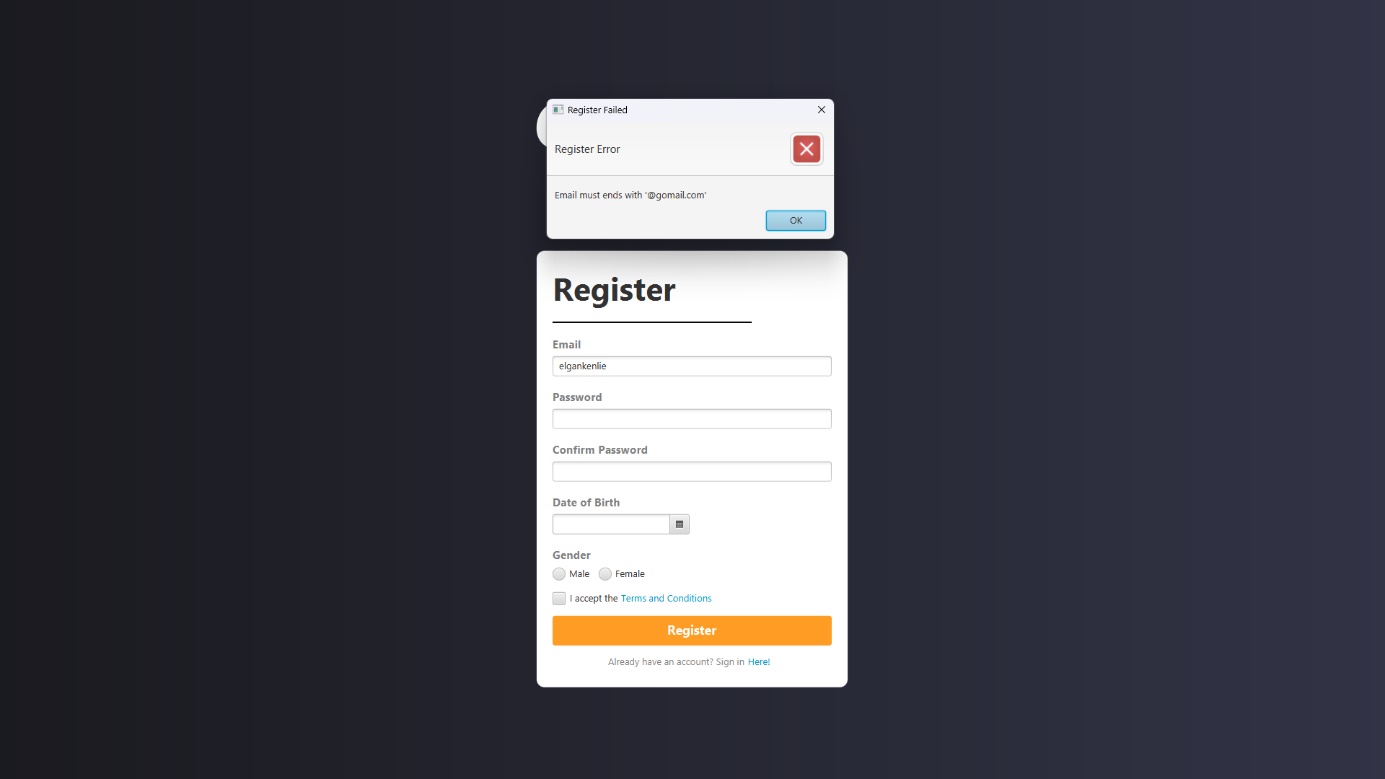
**Validations:**

* **Email**, **Password**, **Confirm Password**, **Date of Birth**, **Gende**r,and **Terms and Condition Checkbox** cannot be **empty.**
* **Email** must be in email format, such as:
* Email must **ends with ‘@gomail.com’**
* Email must **not contain special characters**, except for ‘**@**’, ‘**\_**’, or ‘**.**’.
* **Email** must be unique (not used by other user).
* **Password** must be **alphanumeric.**
* **Password** and **Confirm Password** must match.
* **Date of Birth** must be older than 17 years old

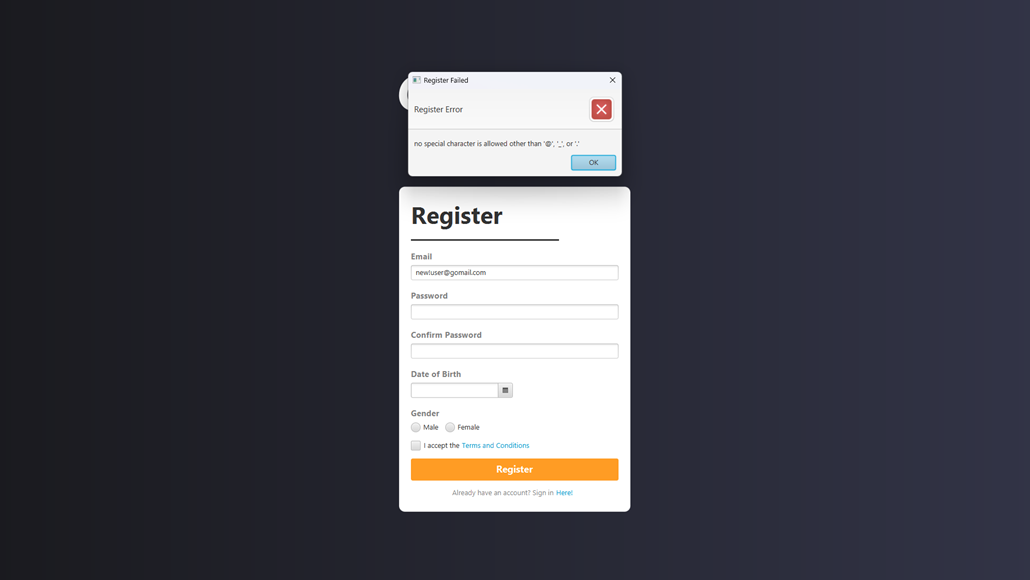
Pressing the **Register Button** or pressing **Enter Key** inside any field will trigger these validations. If the **registration** is **successful**, the user will be redirected to **Select Role Page**. A **failed registration** will display an **Alert**, displaying the type of error.



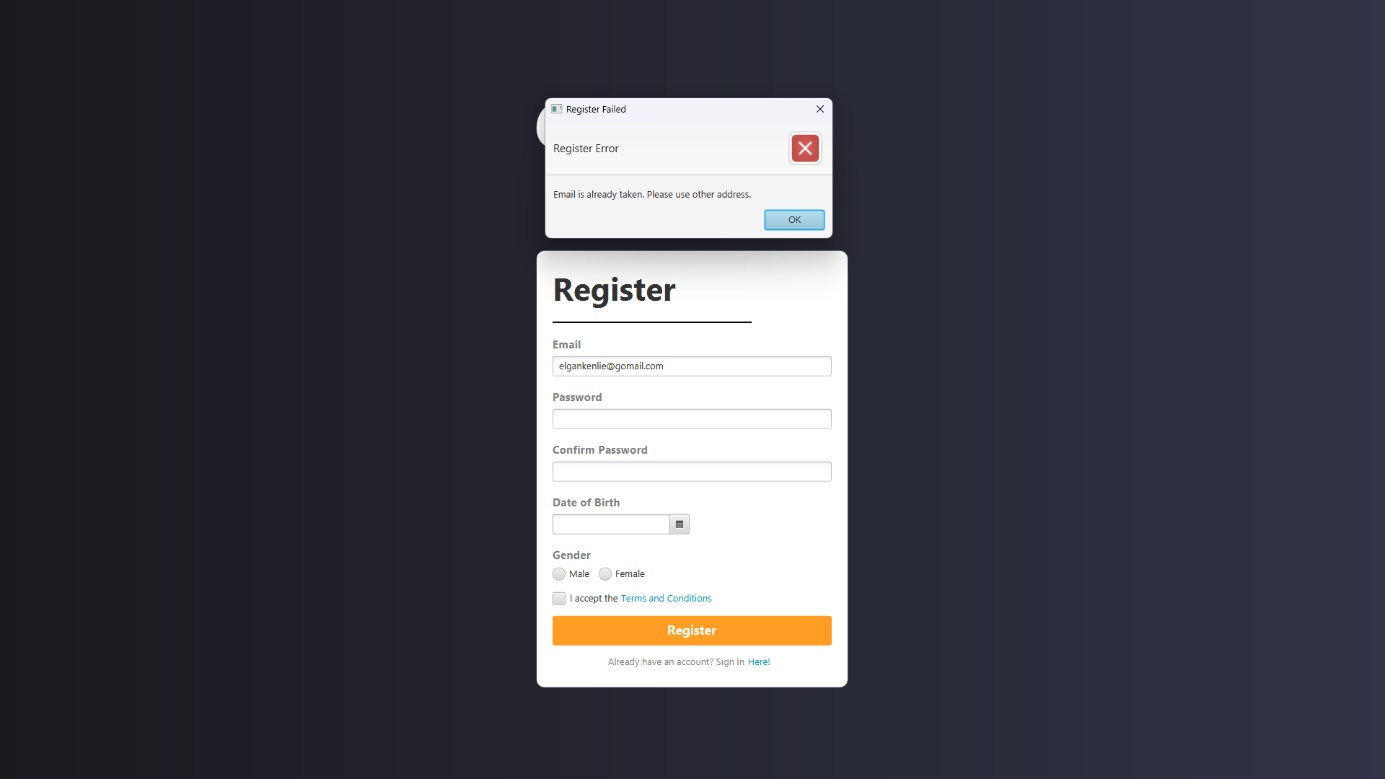
**Figure 6. Empty email error**



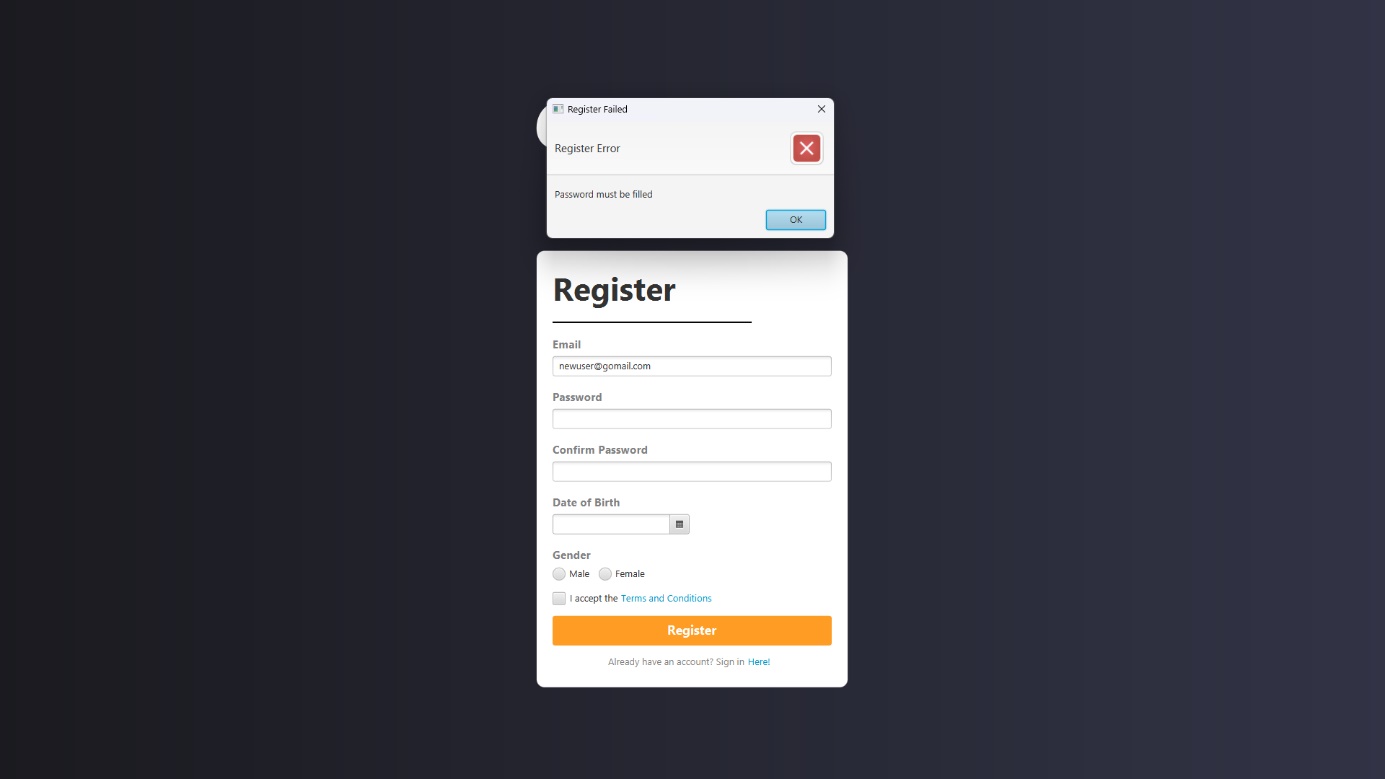
**Figure 7. Wrong email format error**



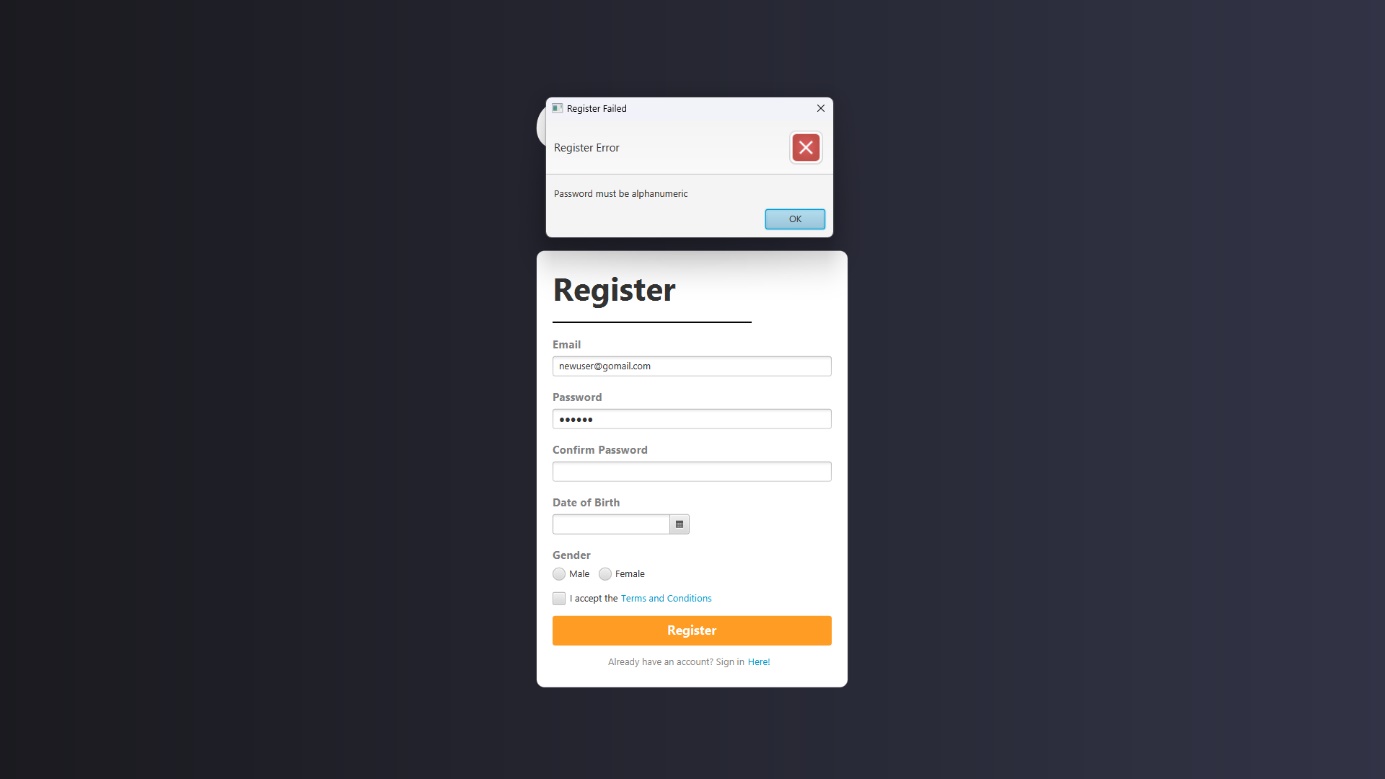
**Figure 8. Invalid email character error**



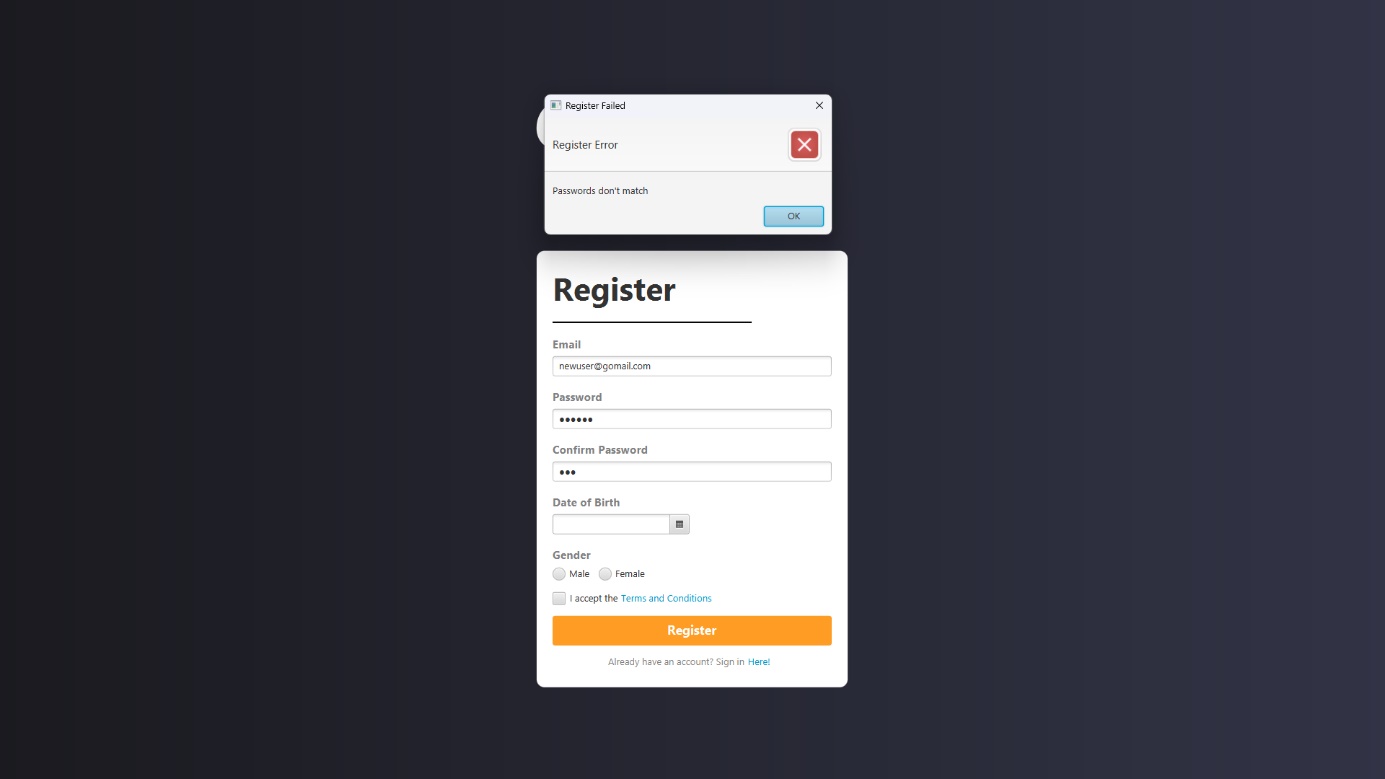
**Figure 9. Email taken error**



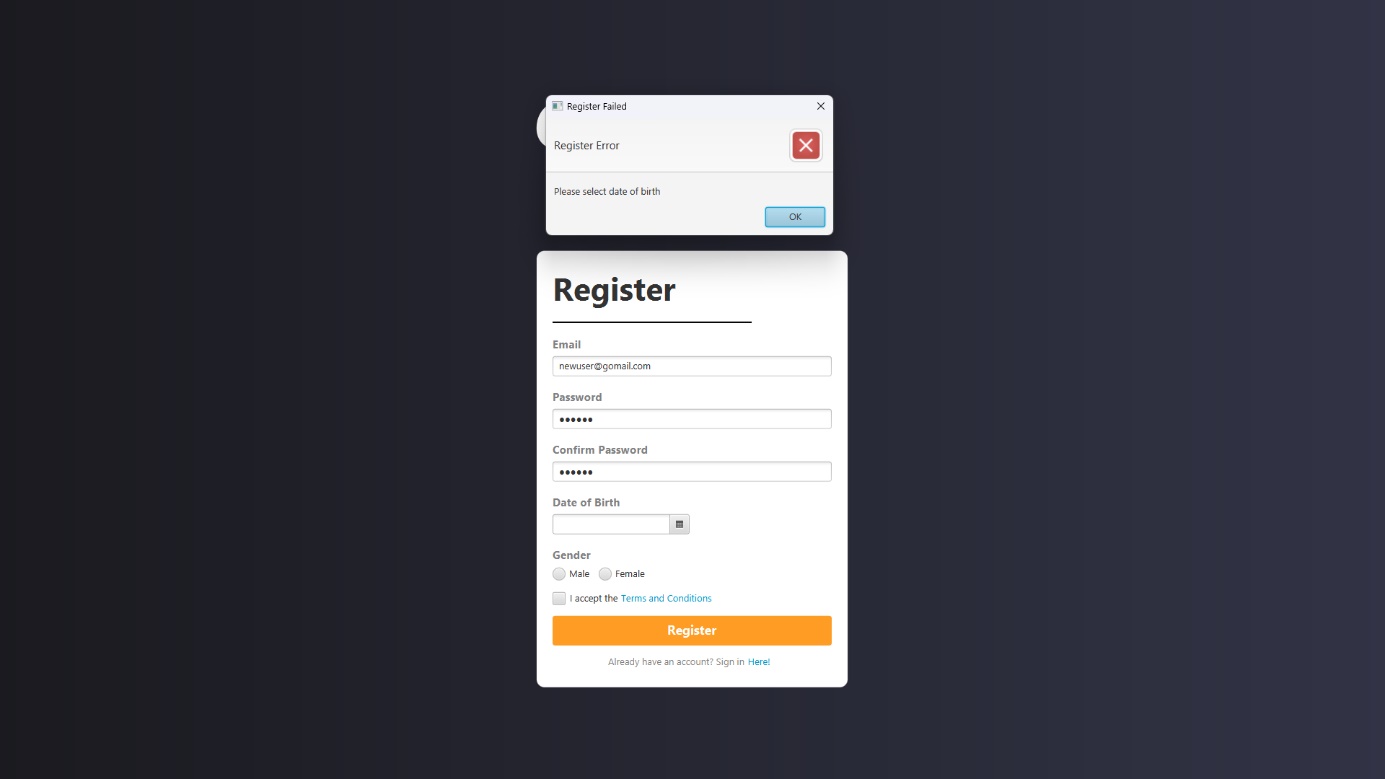
**Figure 10. Empty password error**



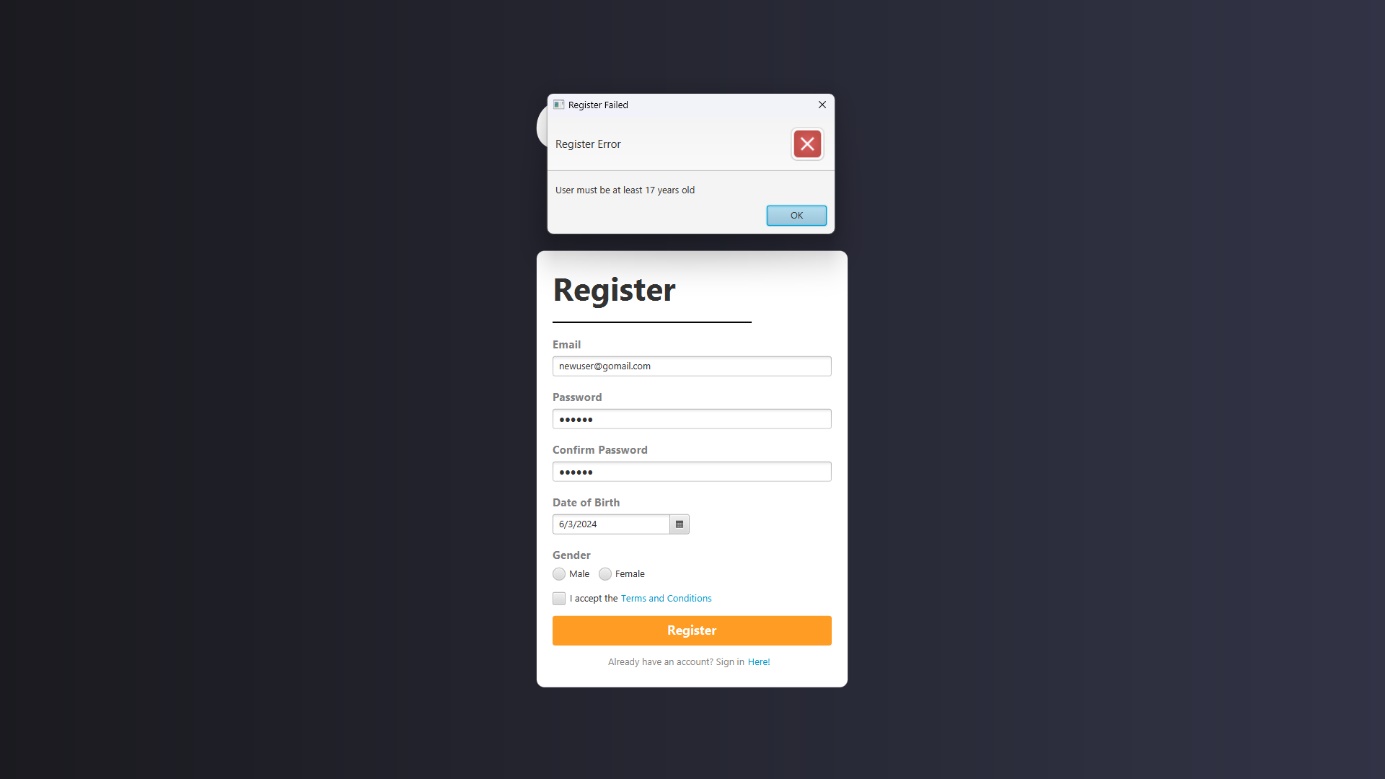
**Figure 11. Password not alphanumeric error**



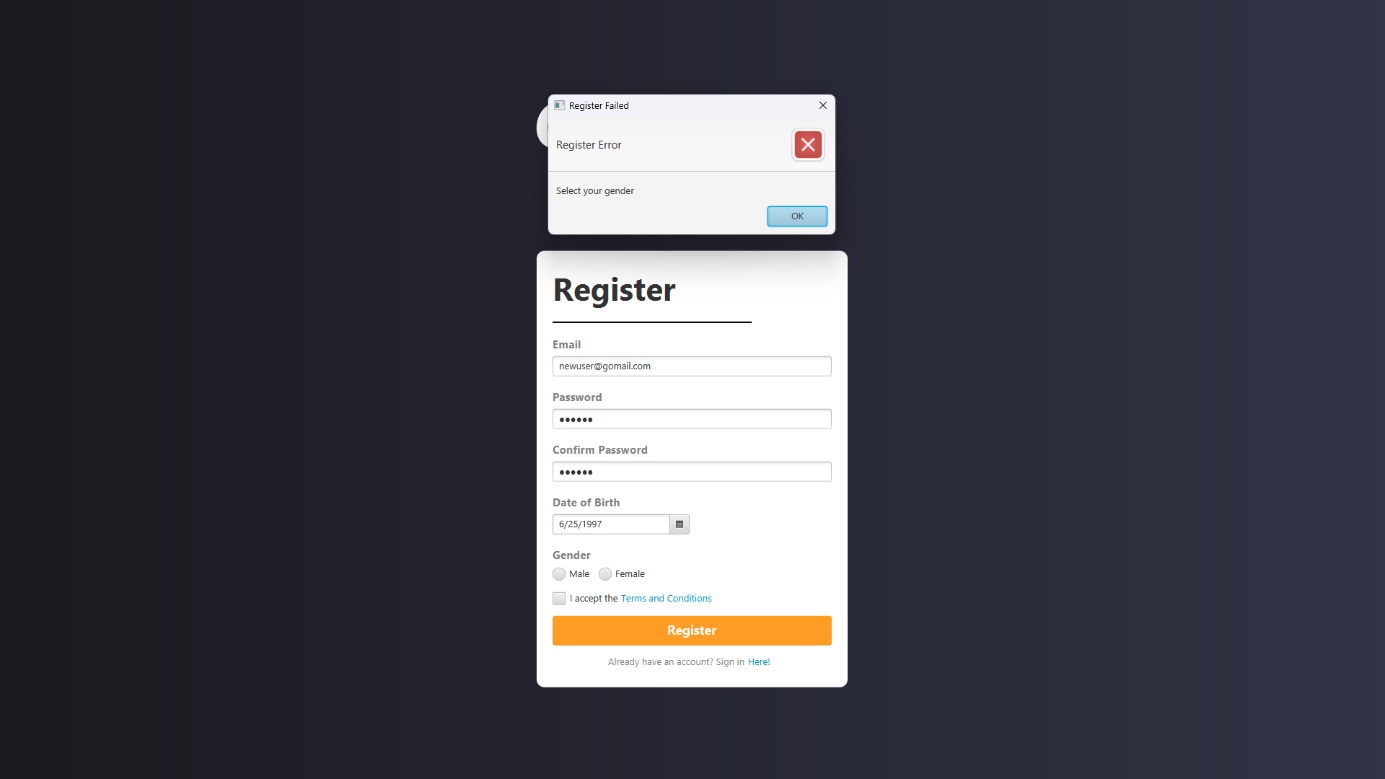
**Figure 12. Password mismatch error**



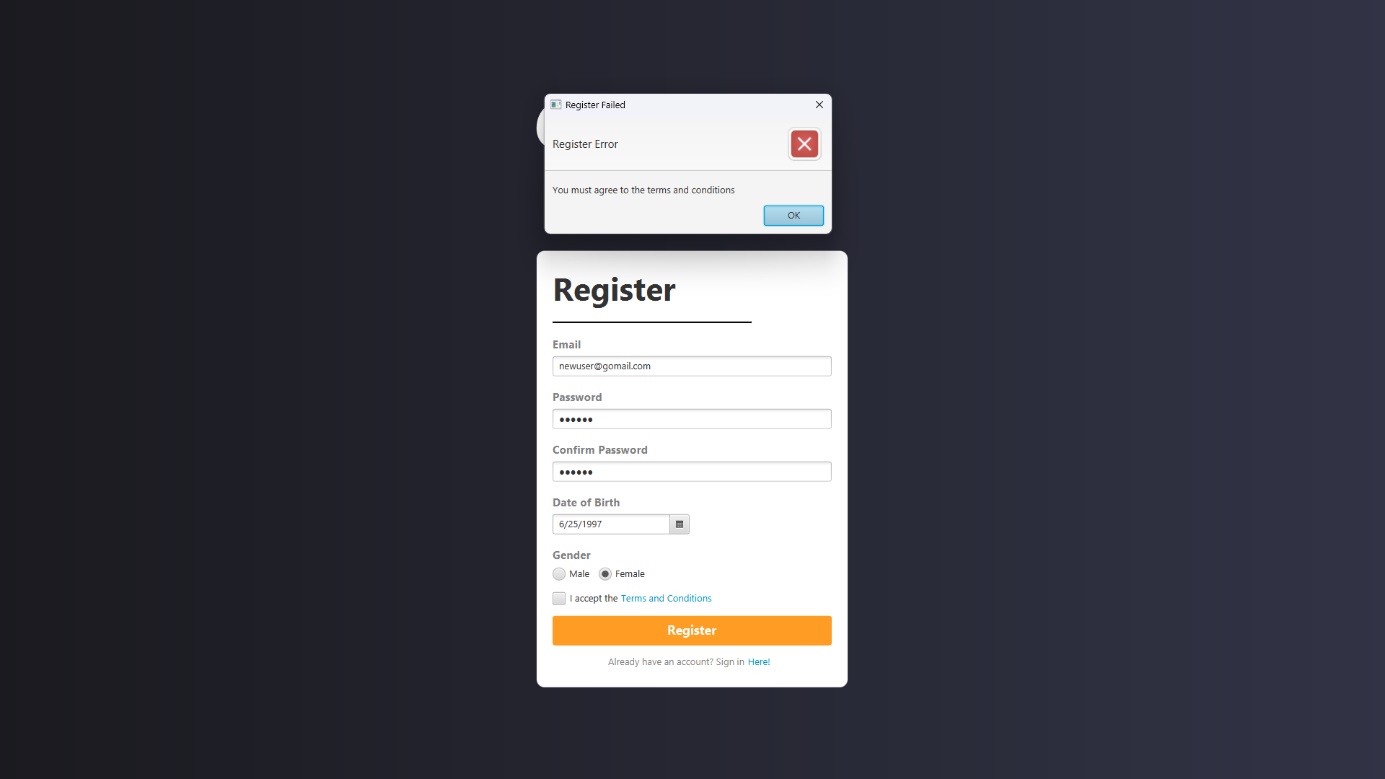
**Figure 13. Empty Date of Birth error**



**Figure 14. Invalid Date of Birth error**

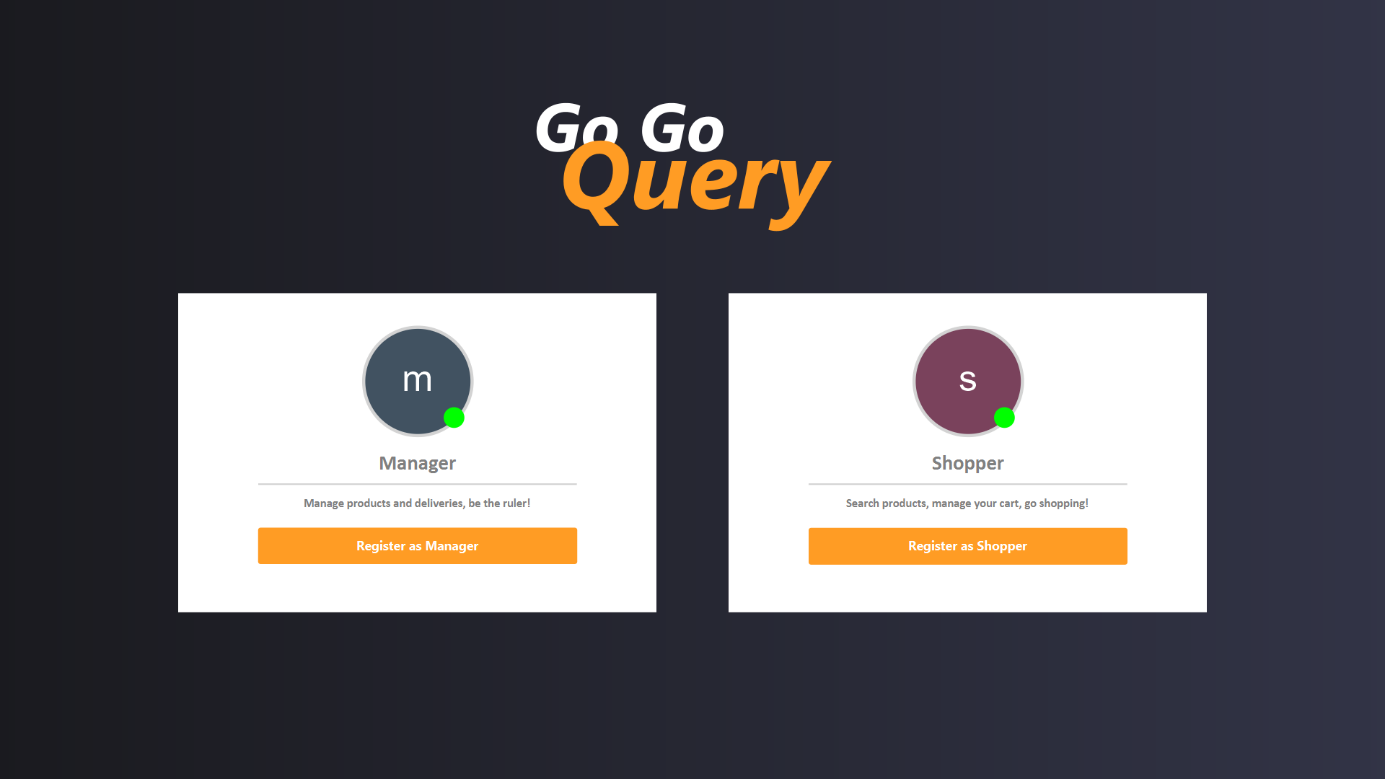


**Figure 15. Empty gender error**



**Figure 16. Empty checkbox error**

1. **Select Role Page**



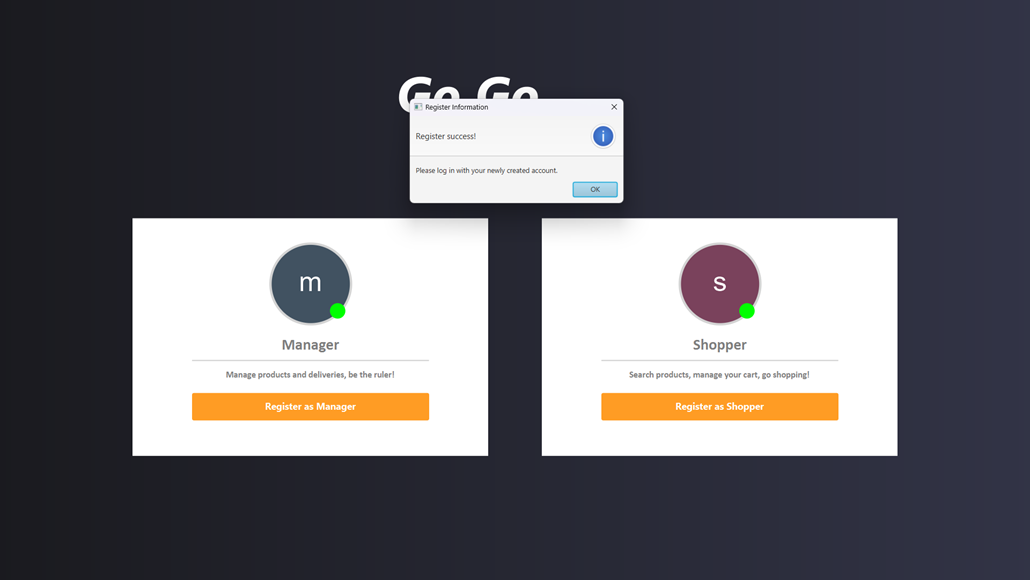
**Figure 17. Select Role Page**

In the **Select Role page**, users can choose to register as either a **Shopper** or a **Manager**. To make a selection, users can simply press one of the two buttons displayed on the scene. Pressing a button will trigger the system to insert a new account using the input from the **Register Page** and the **selected role** as the data in the database, followed by an **alert** displaying a success message.

***Components:***

* **Label** for Manager label and Shopper label
* **Button** for Register as Manager button and Register as Shopper button
* **Alert** for success message

Since **GoGoQuery** has already implemented the Auto Increment feature in its database, you don't need to generate the **UserID**.



**Figure 18. Register success alert**

1. **Shopper Navigation Bar**

A blue and black stripe

Description automatically generated with medium confidence

**Figure 19. Shopper navigation bar**

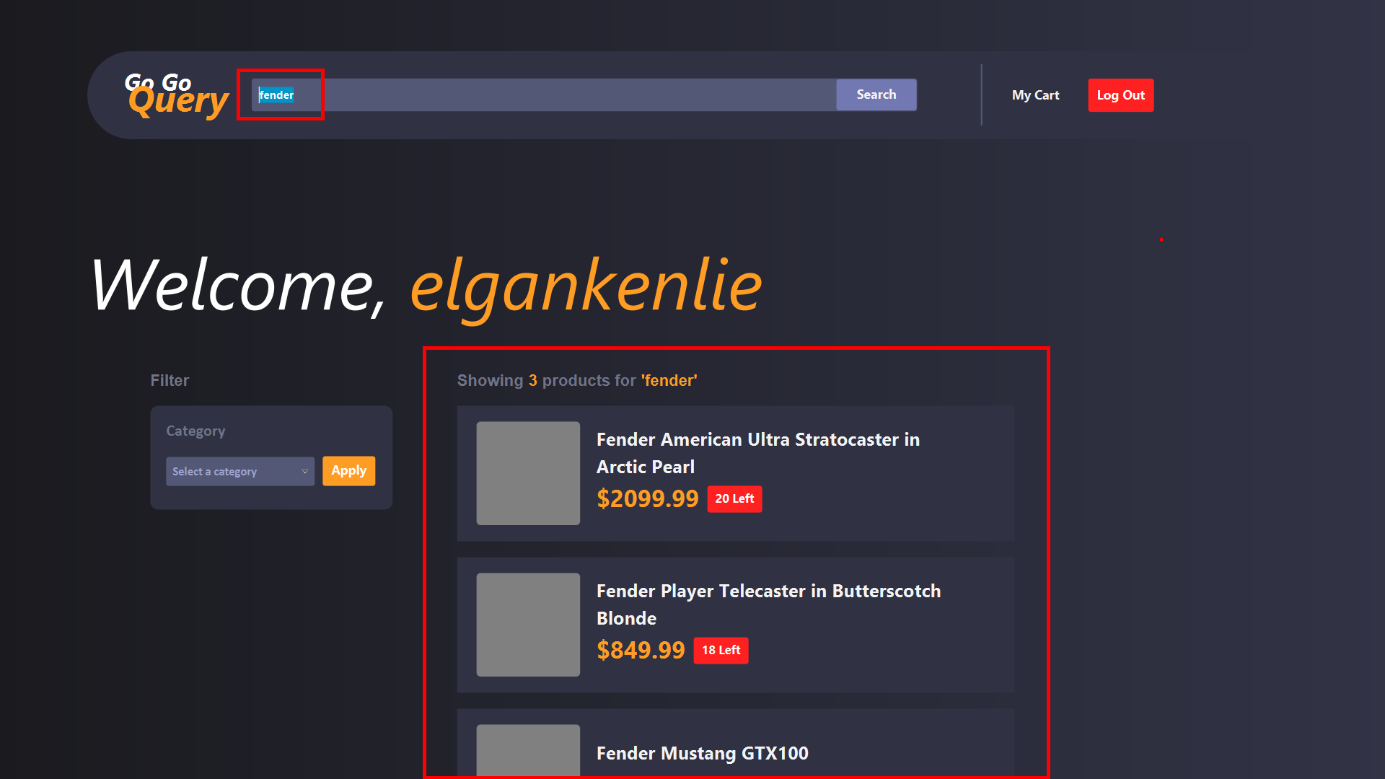
The **Shopper Navigation Bar** is available only for **Shoppers**. It contains a text field, home button, search button, cart button, and logout button.

***Components:***

* **Text Field** for searchbar input
* **Button** for Home button, Search button, Cart button, and Log Out button

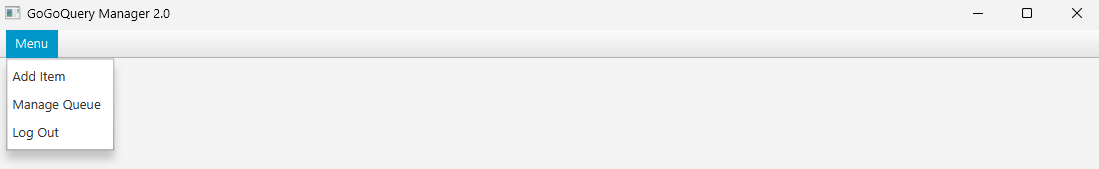
**Descriptions:**

* Pressing the **Home** button will redirect the user to the **Home Page**
* Pressing the **Search** button will filter the item name based on the item name and the text inside the text field.
* Pressing the **My Cart** button will redirect the user to the **Cart Page**
* Pressing the **Log Out** menu will redirect the user to the **Login Page**



**Figure 20. Filtering item name using searchbar**

1. **Manager Navigation Bar**



**Figure 21. Manager navigation bar**

The **Manager Navigation Bar** is available only for **Managers**. It contains a menu that is expandable. When the menu is clicked, it will display the **Add Item, Manage Queue,** and **Log Out** option.

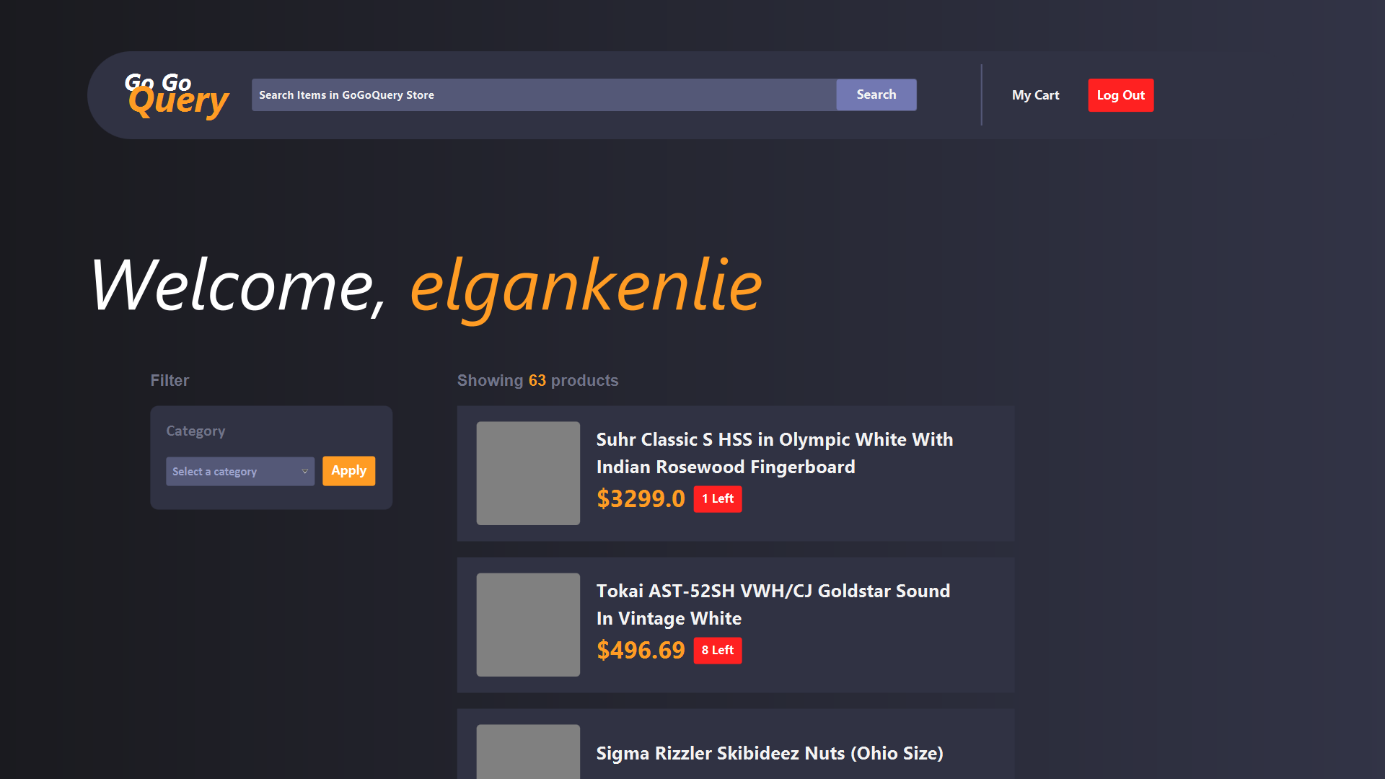
***Components:***

* **Menu Bar** and **Menu**
* **Menu Item** for the menu options (Add Item, Manage Queue, Log Out)

**Descriptions:**

* Pressing the **Add Item** menu will redirect the user to the **Add Item Page**
* Pressing the **Manage Queue** menu will redirect the user to the **Queue Management Page**
* Pressing the **Log Out** menu will redirect the user to the **Login Page**

1. **Shopper Home Page**



**Figure 22. Shopper Home Page**

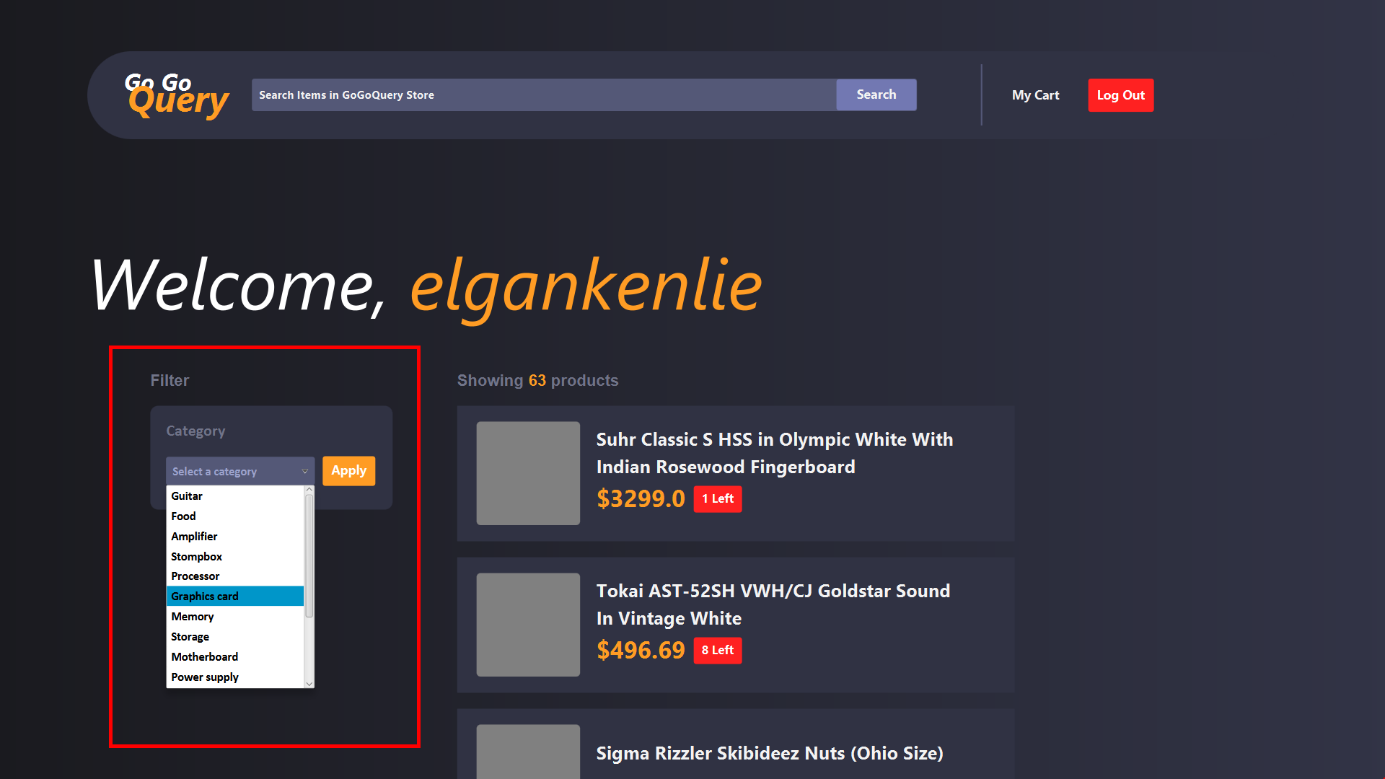
The **Home Page** is the landing page after the user has successfully logged in. It has a welcome message, where the **Username** will change according to the current user, an item category filter, and the list of the products itself.

***Components:***

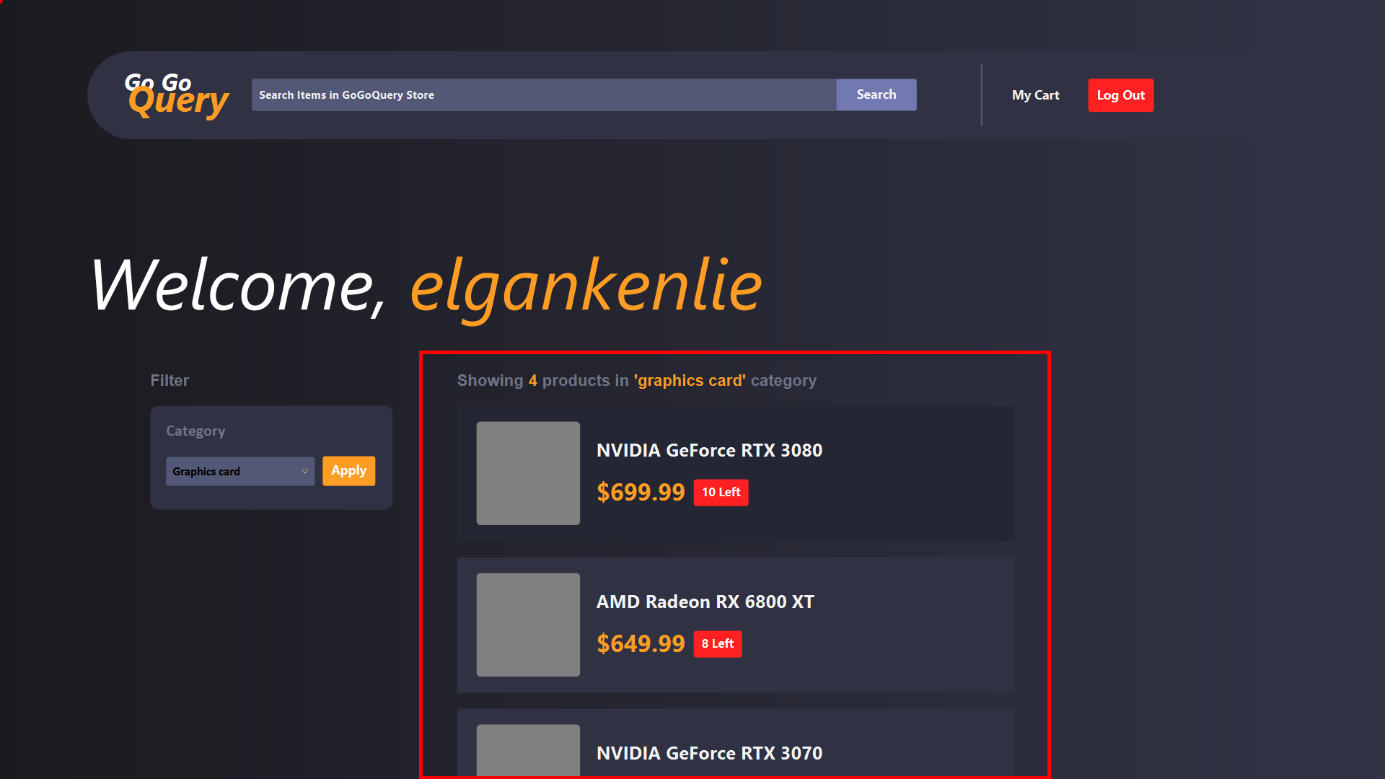
* **Label** for welcome message
* **List View** for displaying products inside the database
* **Button** for Apply Filter button and View Product Detail button
* **Combo Box** for item category selection

**Descriptions:**

* The **Username** is obtained from the current user **Email** before the ‘@’ character.
* Pressing the item will redirect user to **Product Detail Page**
* Pressing the **Apply Filter** button will filter the items based on selected category
* The **List** will only **display all the items** **with available stock** (more than 0).
* The **Combo Box** contains all existing category in the database

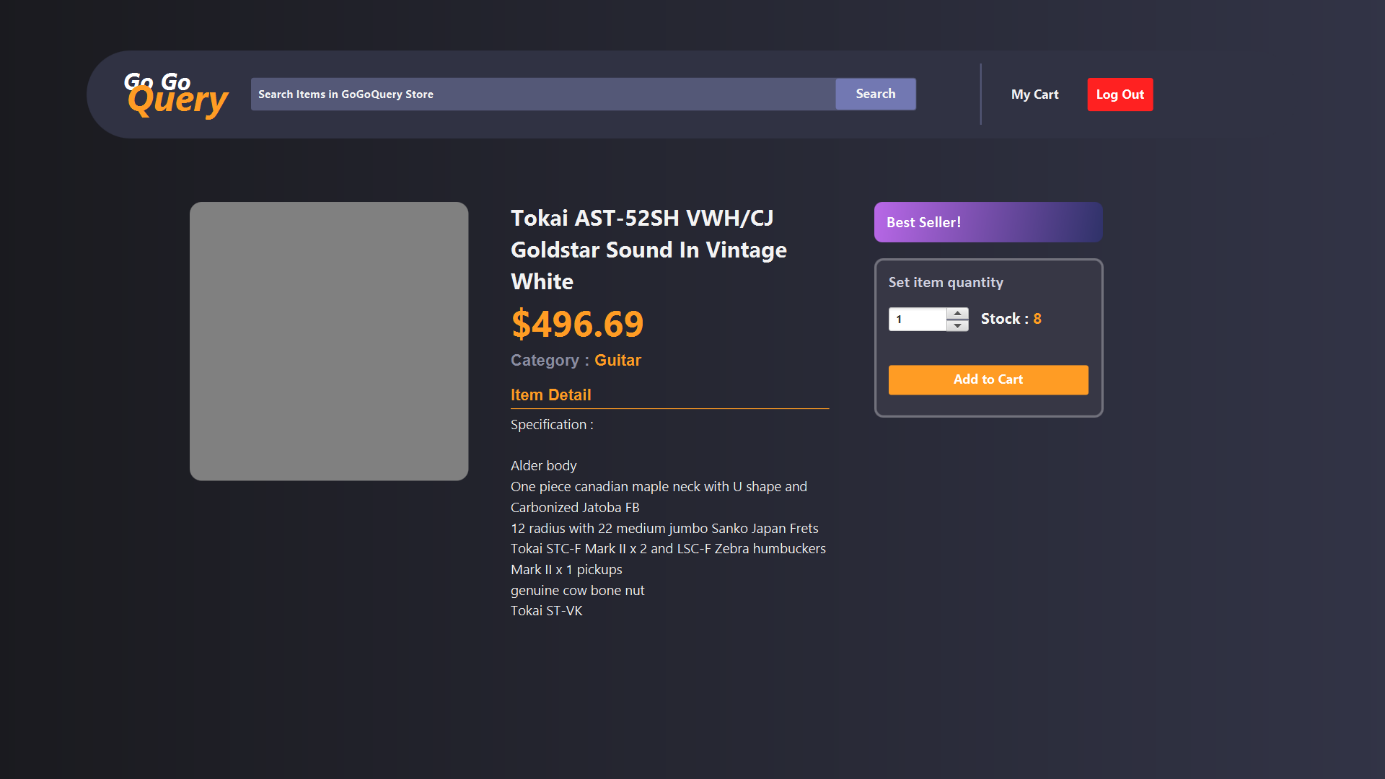


**Figure 23. Selecting item category**



**Figure 24. Filter items based on category**

1. **Product Detail Page**



**Figure 25. Product Detail Page**

Once the user presses an item, they will be redirected to this page where they can see more details about the selected item. The **Product Detail Page** displays the item name, price, category, and details about the selected product. A **spinner** and a **button** are also present to add the selected item to the user's cart with an adjustable quantity. The system will execute some validations before the item can get into user’s cart. After passing the validations, the item will be inserted into database.

***Components:***

* **Label** for Item Name label, Item Price Label, Item Category label, and Item Details label
* **Spinner** for adjusting item quantity
* **Button** for Add to Cart button
* **Alert** for success message and error message

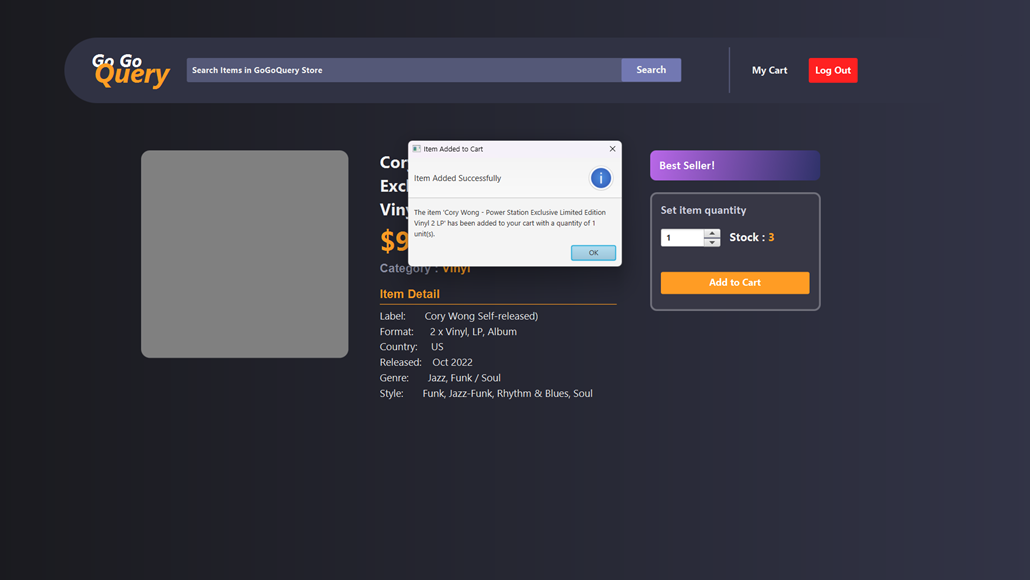
**Validations:**

* Spinner value **must be a positive integer** and **cannot exceed item stock**
* Total quantity in the cart including the current spinner value **cannot exceed item stock**

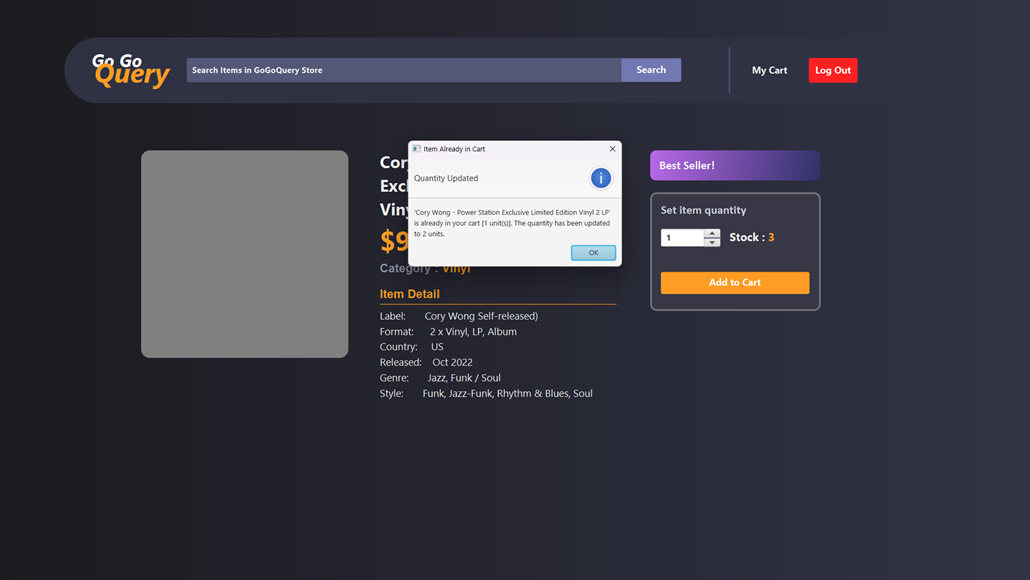
Pressing the **Add to Cart** button will trigger some validations, such as increasing the quantity based on the spinner value if the same item is already in the cart, and ensuring that the total quantity in the cart including the current spinner value cannot exceed the item stock. After passing some validations, the selected items will be added to the cart.

**Descriptions:**

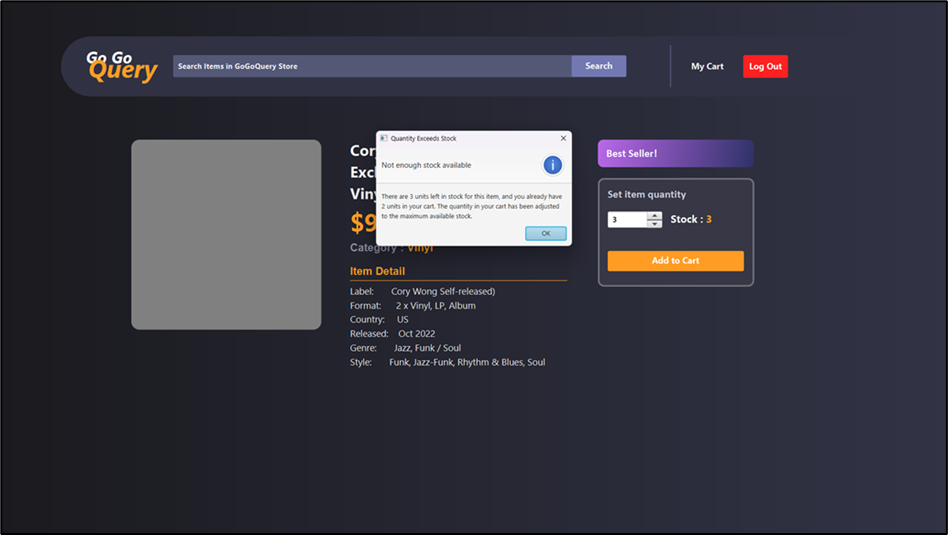
* Adding an item that already exists in cart will **increase its quantity by the spinner value**.
* Adding an item that already exists in the cart, with the combined quantity from the spinner and existing cart quantity **exceeding the available stock**.
* Adding an item will trigger an **Alert** based on these cases:
* Adding an item that does not exist in the cart yet
* Adding an item that already exists in the cart
* Adding existing item that exceeds stock limit



**Figure 26. Adding new item to cart**

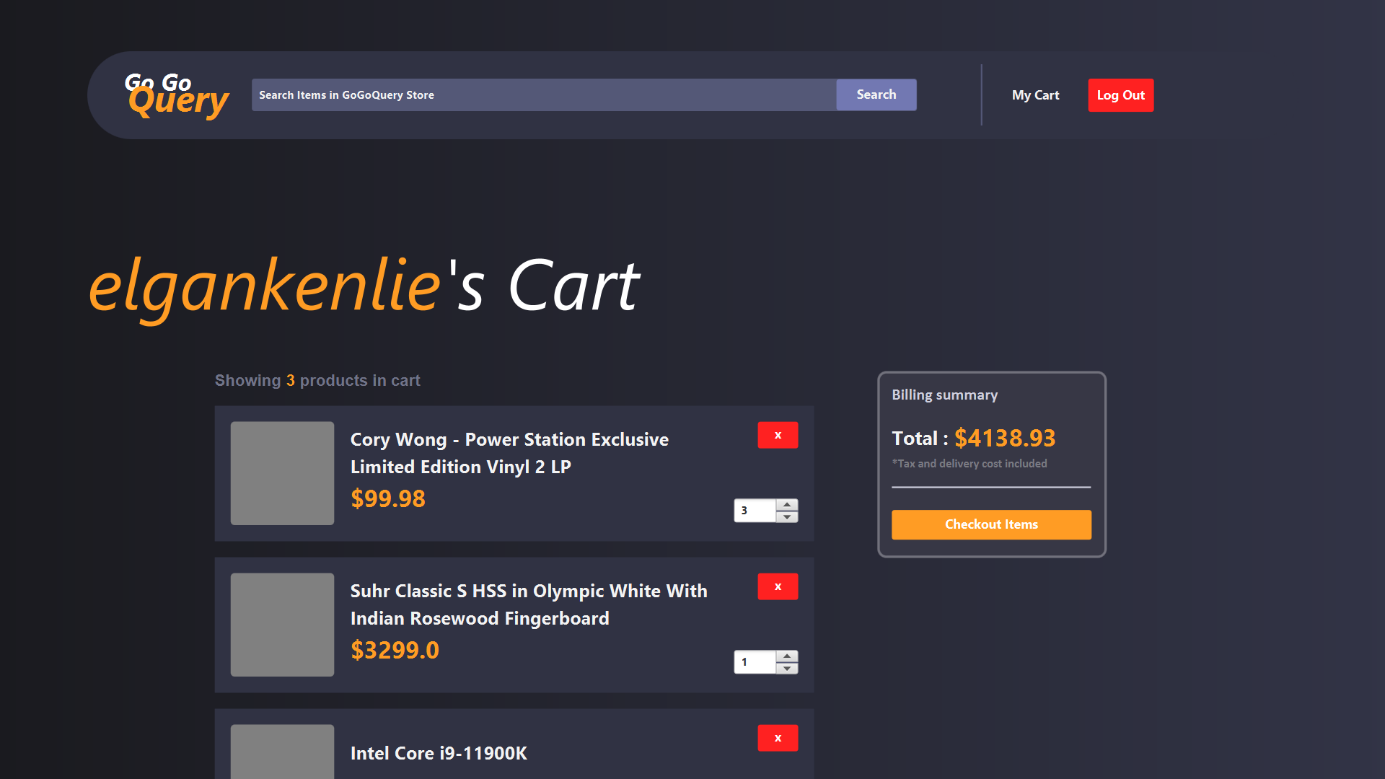


**Figure 27. Adding existing item to cart**



**Figure 28. Adding existing item that exceeds stock limit**

1. **Cart Page**



**Figure 29. Cart Page**

The **Cart Page** displays the name, price, quantity of the items inside the logged in user’s cart, and the grand total price. Users can adjust the desired quantity for each items, remove unwanted items, and make a purchase by clicking the checkout button.

***Components:***

* **Label** for Grand Total Price label
* **List View** for displaying items in the logged in user’s cart
* **Spinner** for adjusting item quantity
* **Button** for Remove Item button and Checkout Items button
* **Alert** for success message and confirmation message

**Validations:**

* Show a message **if there is no item inside the cart**
* Spinner value **cannot exceed item stock**
* Adjusting the spinner value to zero will **remove an item from the cart**

Pressing the **Checkout** button will trigger the system to show a confirmation **Alert** and insert a new transaction to the database with ‘In Queue’ as the transaction status, followed by another **Alert** that shows a success message, while pressing the Remove Item button will trigger a confirmation **Alert** and remove the selected item from cart.

**Descriptions:**

* **Changing the spinner value will update the Grand Total Price** label based according to the following formula :

**Grand Total Price**

* Clicking the **Remove Item Button** will trigger a confirmation **Alert**. Clicking OK on the alert will result in **item removal and database update**.
* Clicking the **Checkout Button** will trigger a confirmation **Alert**. Clicking OK on the alert will result in **cart clearance and a new transaction header**, along **with its transaction details will be created**. The **Item stock** will also be **updated**. Then, a **successful alert** will be displayed.

**Transaction Detail**

**TransactionID**: the same as the TransactionID in the Transaction Header

**ItemID**: using the ItemIDs of the cart items

**Quantity**: using the Quantities of the cart items

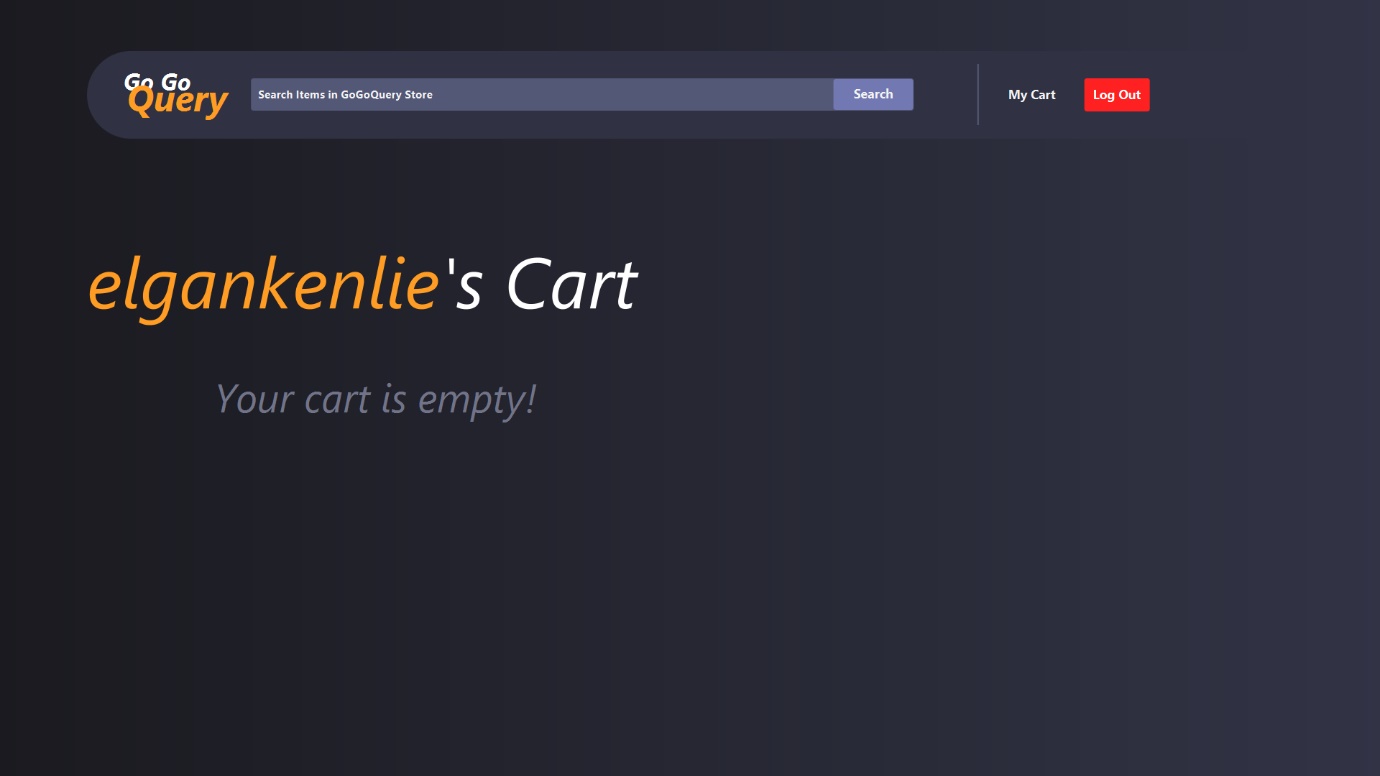
**Transaction Header**

**TransactionID:** AUTO INCREMENT

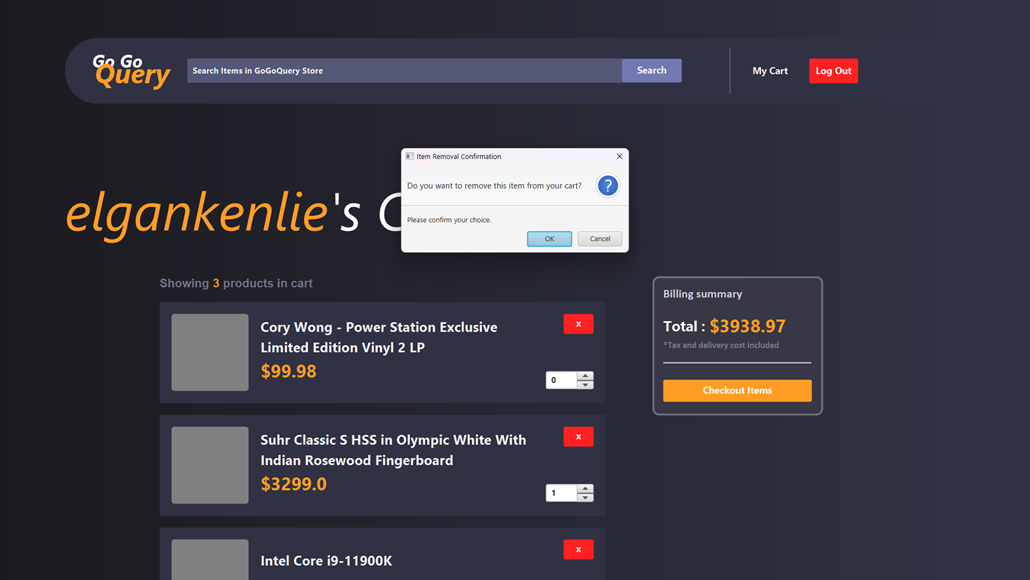
**UserID**: the logged in user’s ID

**DateCreated**: the current date in yyyy-MM-dd format

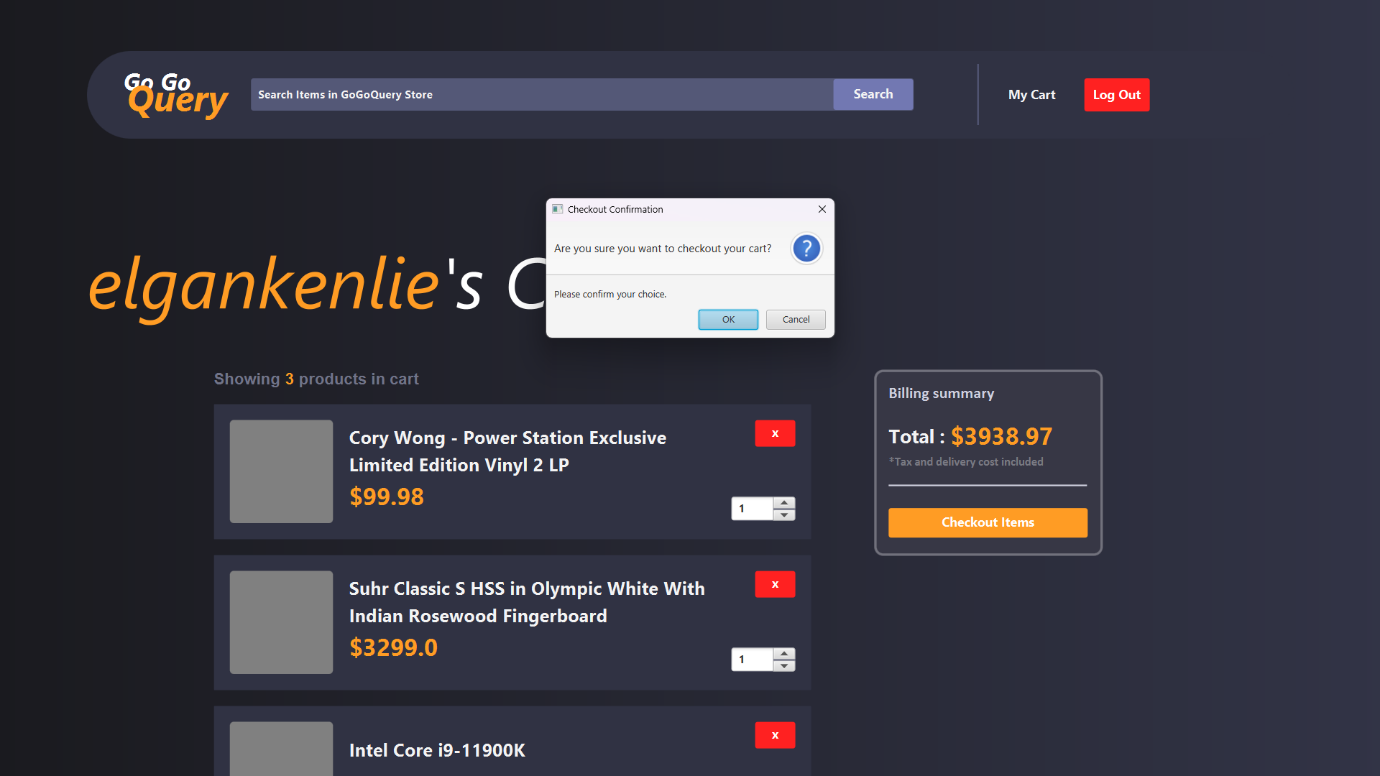
**Status** : ‘In Queue’



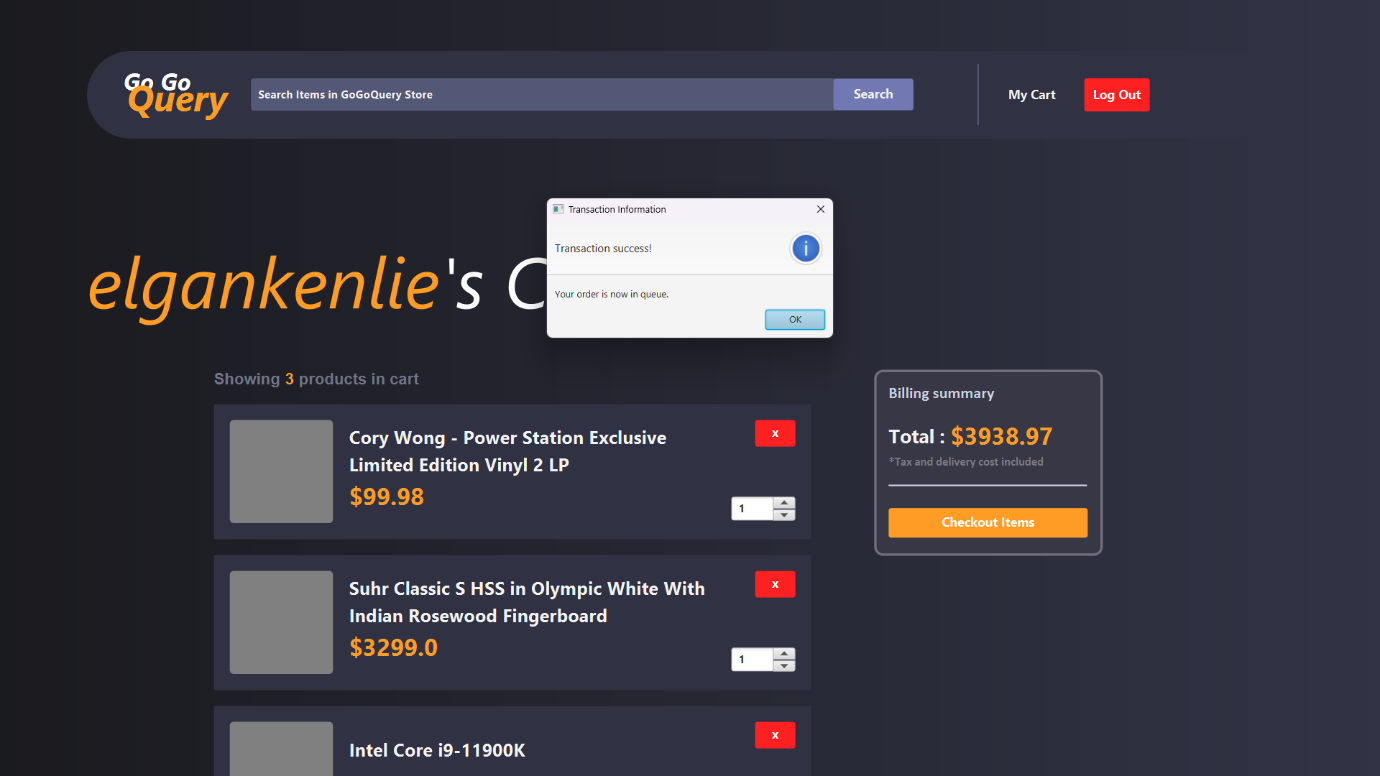
**Figure 30. Empty cart**



**Figure 31. Item removal confirmation alert**

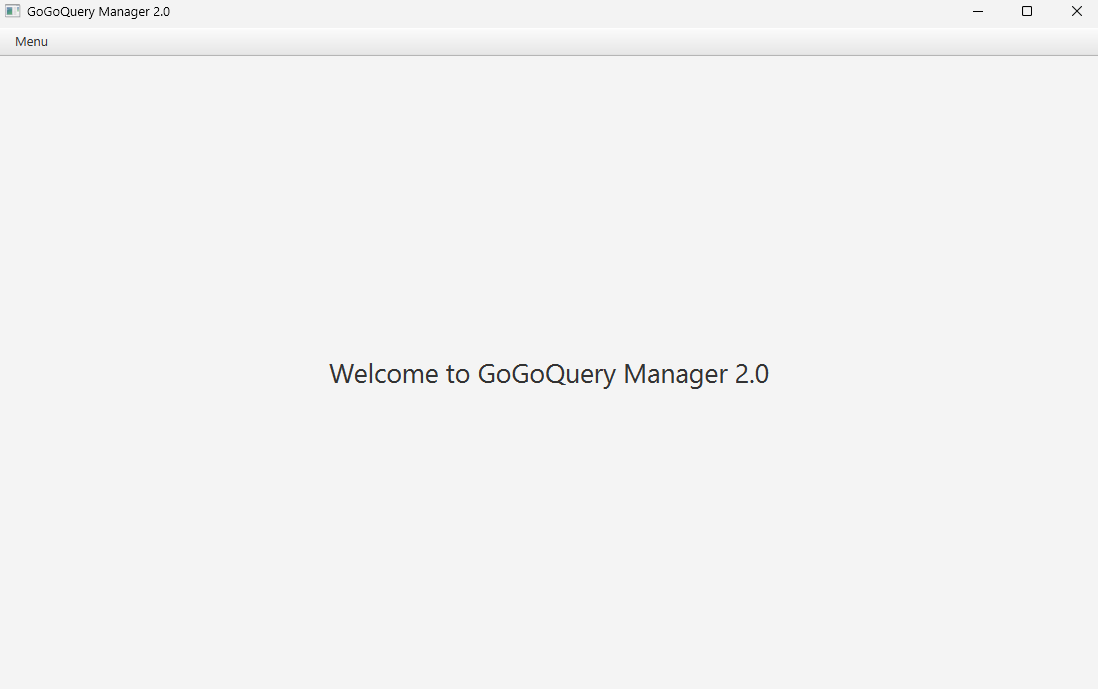


**Figure 32. Checkout confirmation alert**



**Figure 33. Checkout success alert**

1. **Manager Home Page**



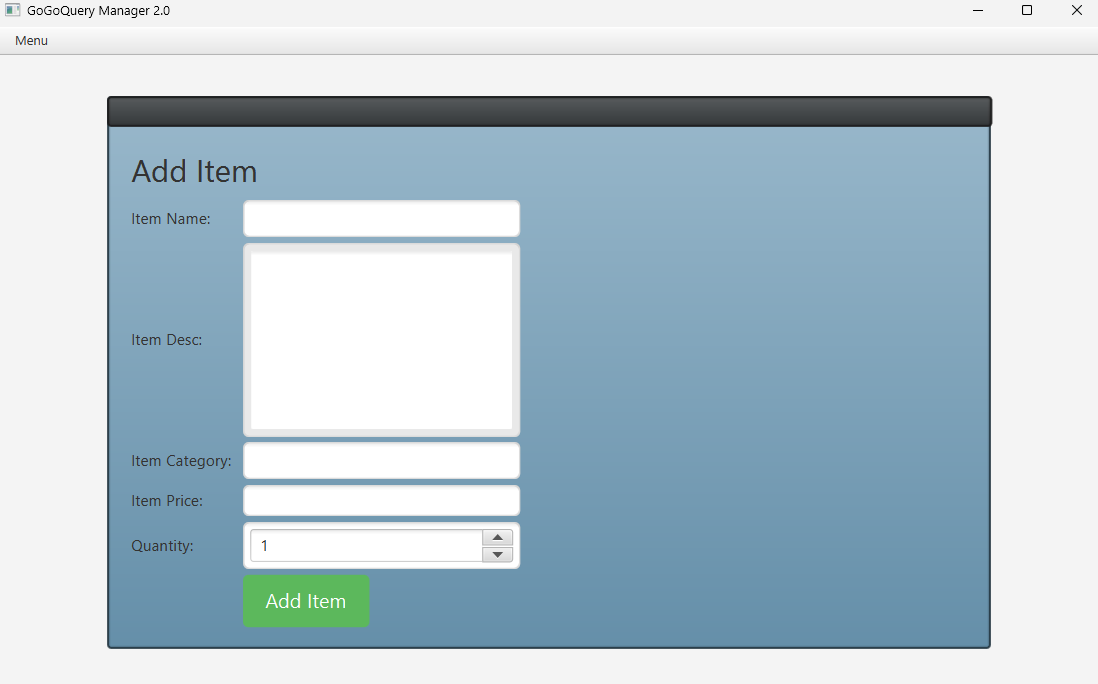
**Figure 34. Manager Home Page**

The **Manager Home Page** is the landing page after the **Manager** has successfully logged in. The **Manager Home Page** only consists of **Manager Navigation Bar** and a welcome message.

***Components:***

* **Label** for welcome message

1. **Add Item Page**



**Figure 35. Add Item Page**

The **Add Item Page** will be displayed on the screen once a user with **Manager** role selected **Add Item** option from the **Manager Navigation Bar.** This page allows Managers to add new items to the store's database by filling out the form shown inside a **Pop-Up Window**. The system will execute multiple validations before the new data can be inserted into the database.

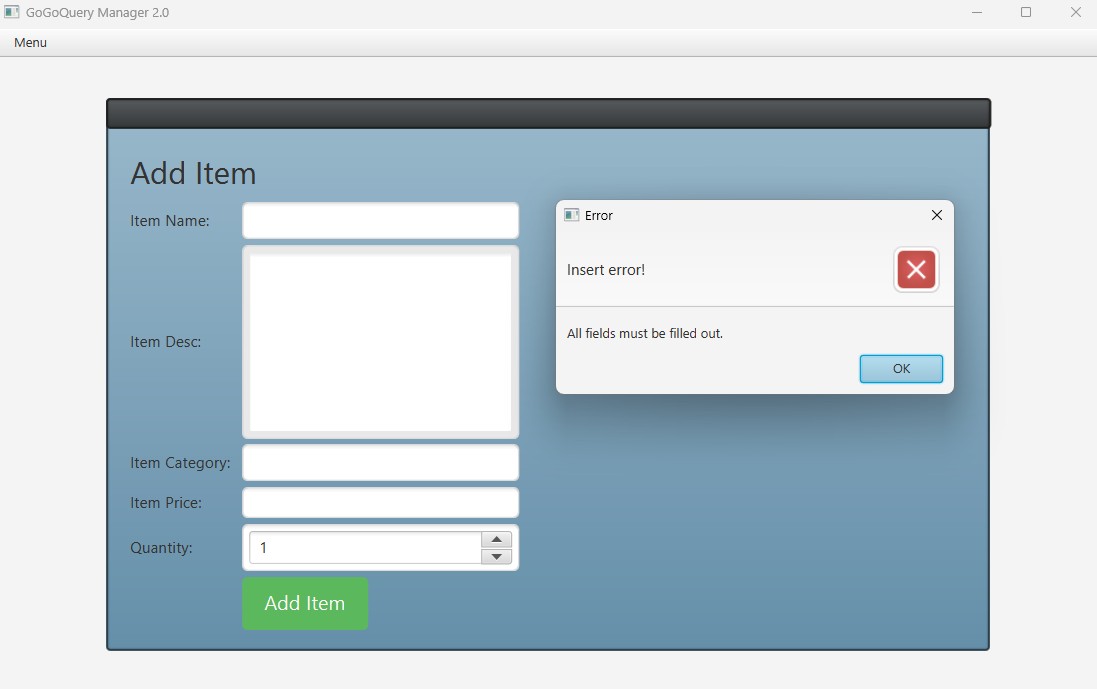
***Components:***

* **Label** for Item Name label, Item Description label, Item Category label, and Item Price label
* **Text Field** for Item Name input, Item Category input, Item Price input
* **Text Area** for Item Description input
* **Spinner** for adjusting item quantity
* **Button** for Add Item button
* **JFXtras Window** for add item form pop up

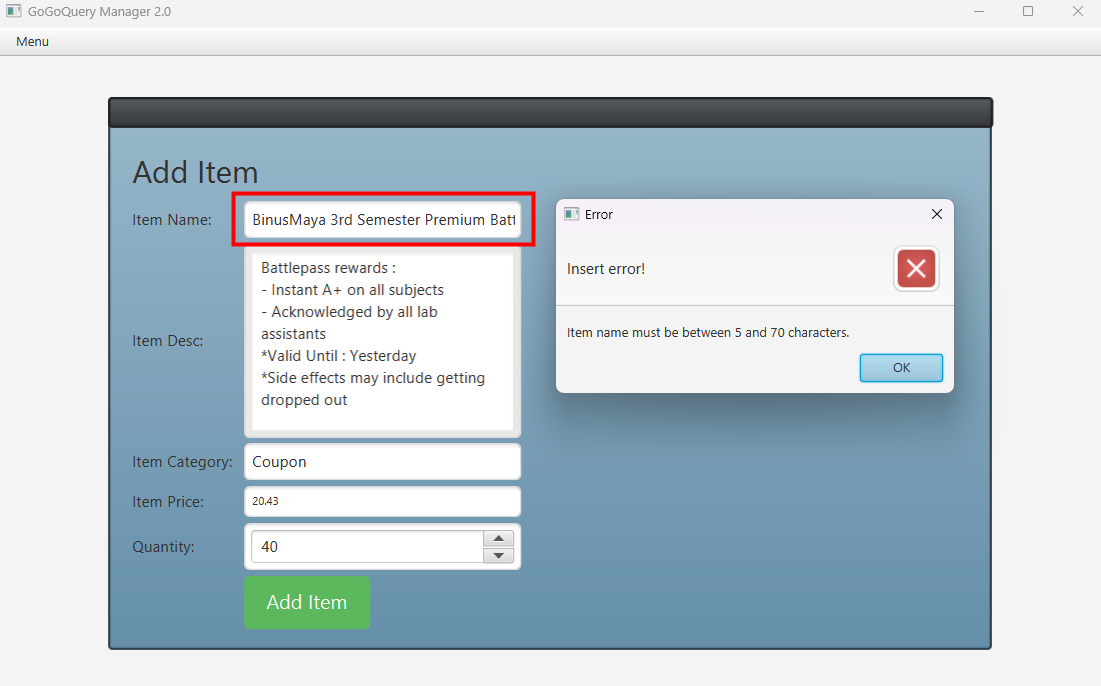
**Validations:**

* All fields **must be filled out**.
* Item name **must be between 5 and 70 characters**.
* Item description **must be between 10 and 255 characters**.
* Quantity **must be a Double and between $0.50 and $900,000**.
* Spinner value **must be a positive integer** and **cannot be more than 300**

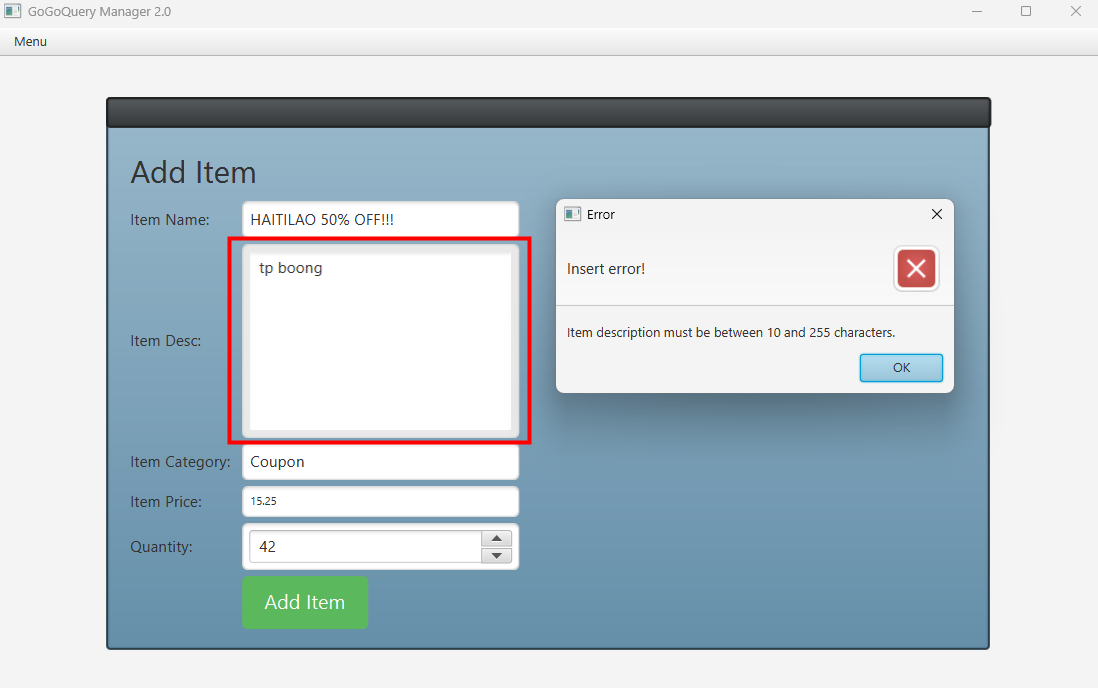
Pressing the **Add Item** button will trigger these validations, any invalid input will cause an **Alert** to show the error message according to the error type.



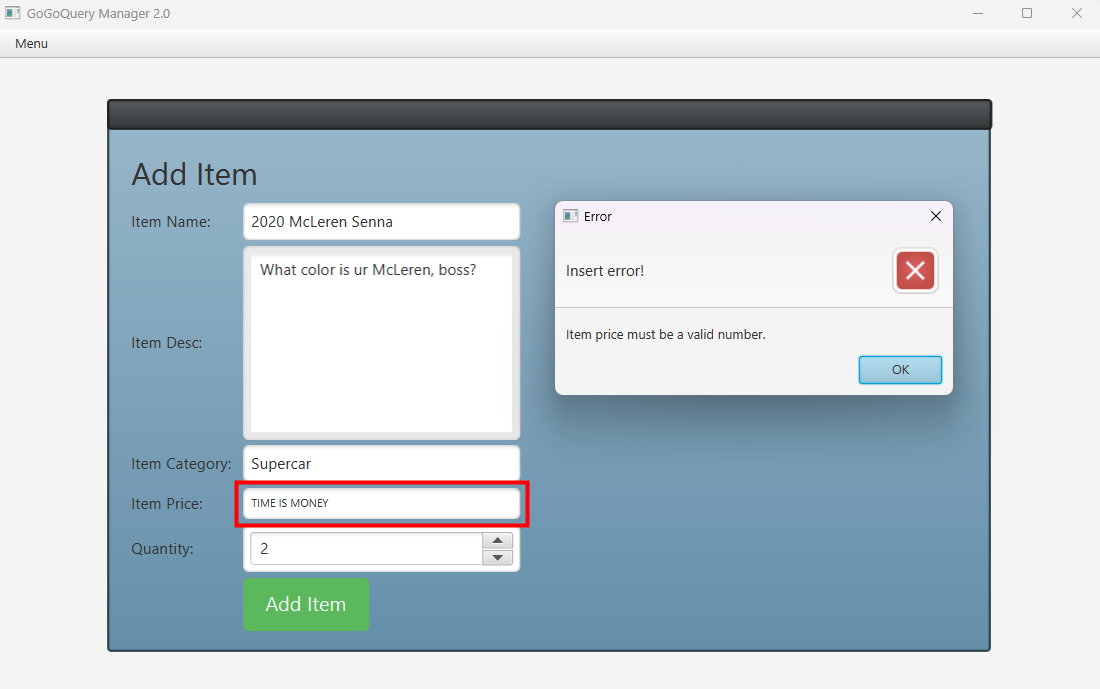
**Figure 36. Add item empty field error**



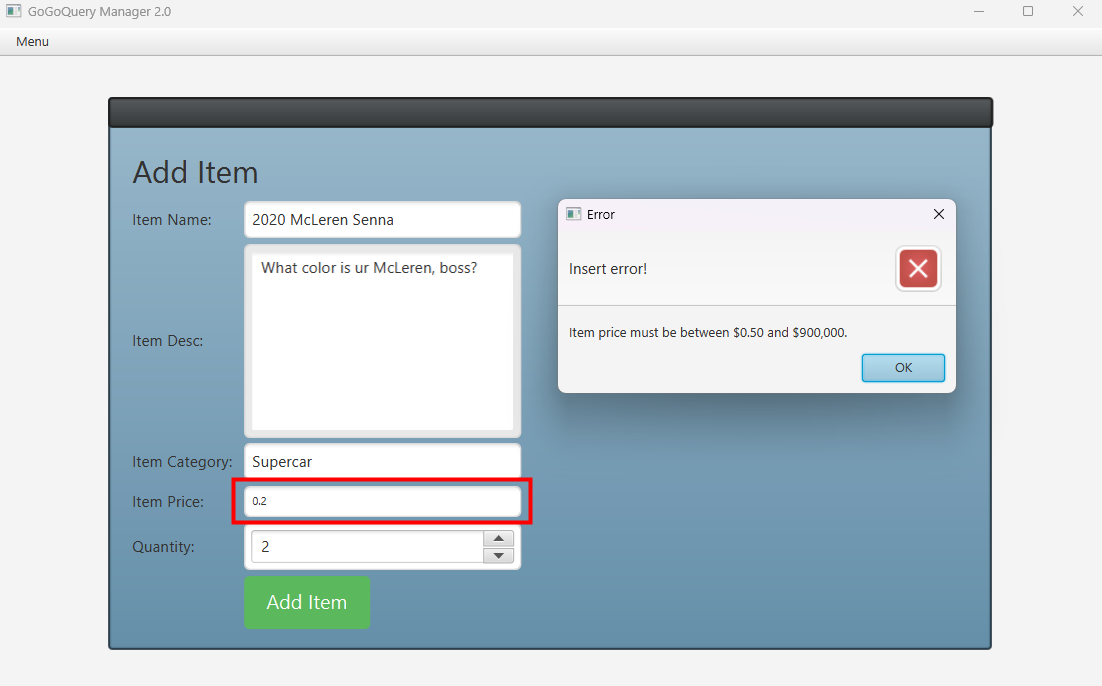
**Figure 37. Invalid item name error**



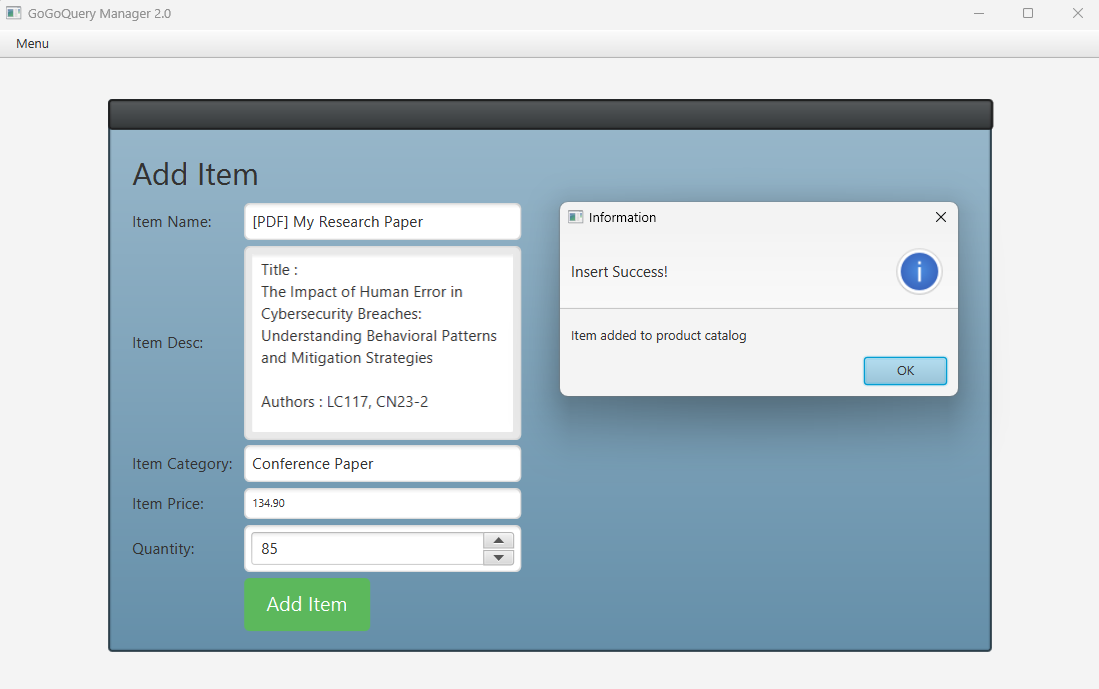
**Figure 38. Invalid item description error**



**Figure 39. Invalid item price datatype error**

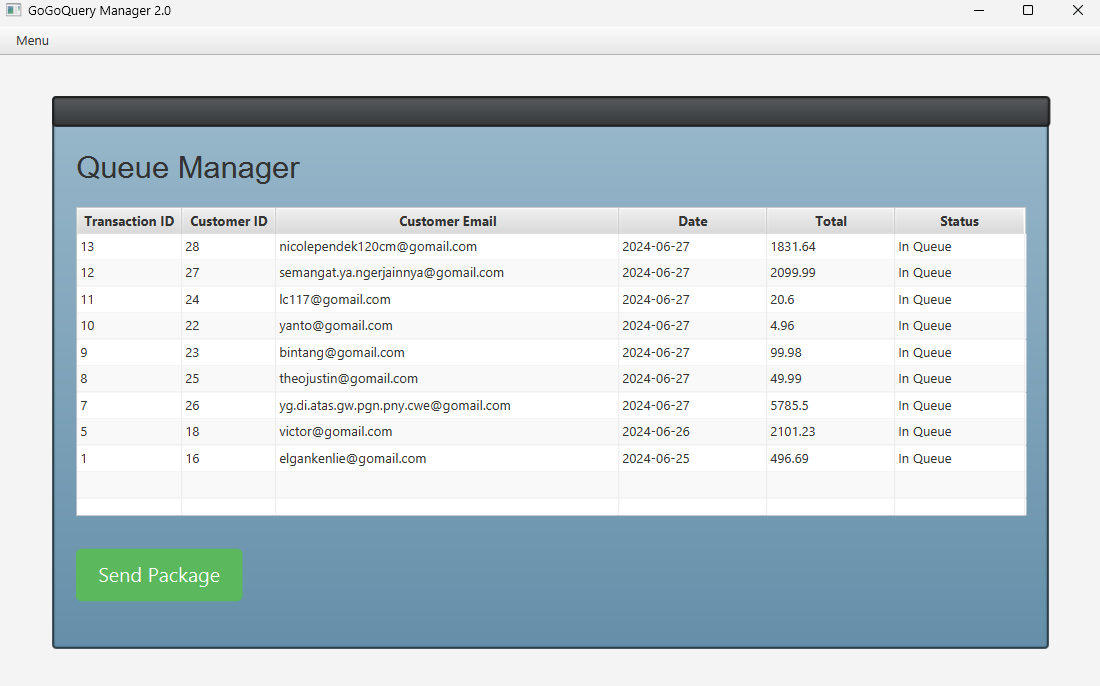


**Figure 40. Invalid item price value error**



**Figure 41. Add item success message**

1. **Queue Management Page**



**Figure 42. Queue Management Page**

The **Queue Management Page** displays multiple data from all transactions via **Table View**. A user with the **Manager** role can access this page to manage transaction queue by prioritizing which transactions to send first via clicking a button.

**Components:**

* **Table View** for displaying transaction data
* **Button** for Send Package button
* **Alert** for displaying error message

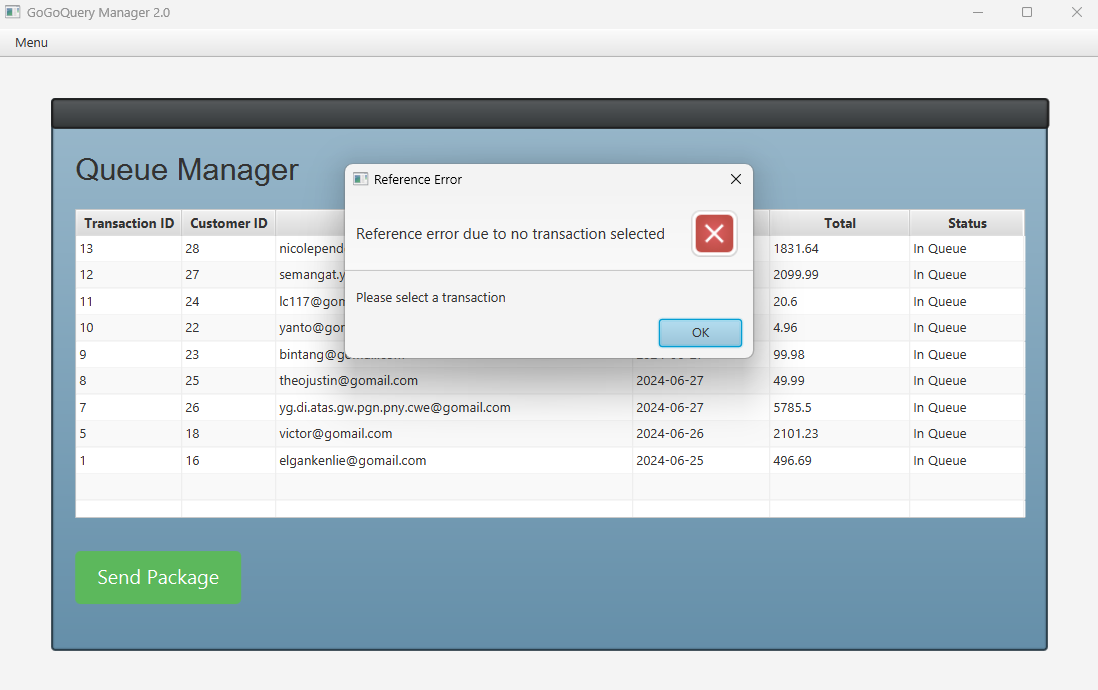
**Validation:**

* **A transaction must be selected** firstbefore managers can send package

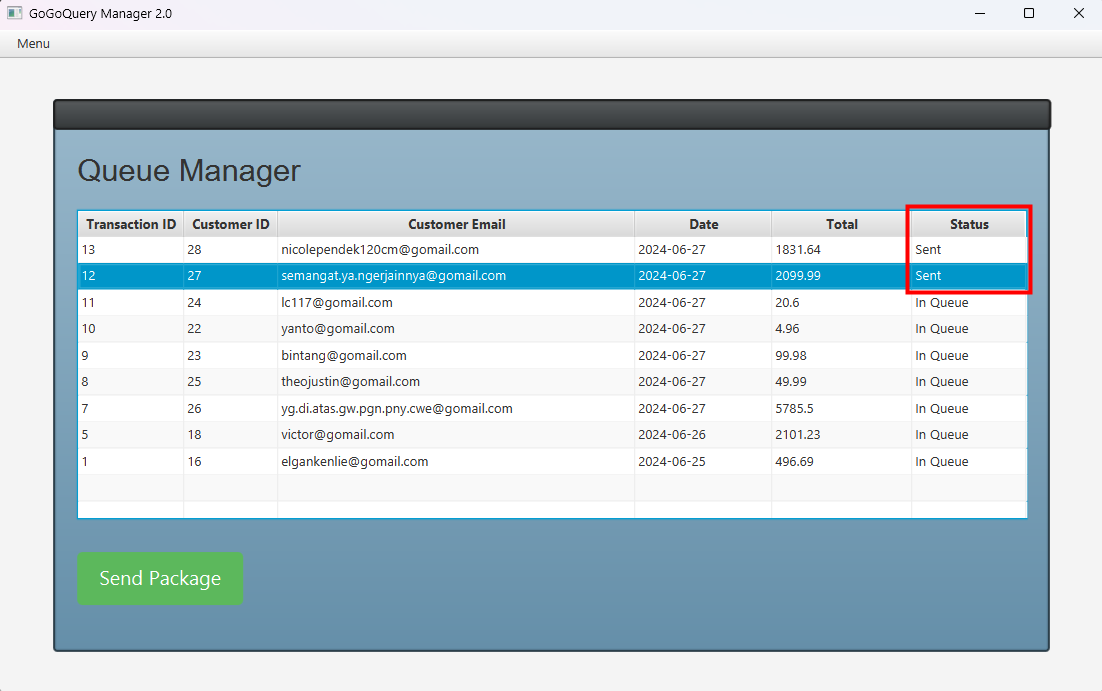
Pressing the **Send Package** button will trigger the system to execute the validation before setting the transaction status from ‘**In Queue**’ to ‘**Sent**’. A failed attempt will trigger an **Alert** to show the error message.

**Descriptions:**

* **Manager** can **update a transaction from database** by **selecting the transaction that they want to send** and **clicking** the **Send Package** button. If the user **has not selected** a transaction to be sent, an **error alert** will be displayed. Otherwise, the database will be updated and the status inside the **Table View** will be set to ‘**Sent**’



**Figure 43. No transaction selected error**



**Figure 44. Successful attempt on sending package**