Project Test Plan

GiftPal – The Ultimate Gift Exchange and Reminder App

Submitted in fulfillment

Of the requirements of

CMSC 495 Current Trends and Projects in Computer Science

(Group Project)

Bazz Khurshid

Graham Primm

Kalala Kalamba

Nicholas Sink

|  |  |
| --- | --- |
| **Document ID** | GiftPal PTPv2.1 |
| **Version Number** | 2.1 |
| **Due Date** | May 07, 2023 |
| **Classification** | Public |
| **Instructor** | Hung Dao |
| **This submission is by** | Bazz Khurshid, Graham Primm, Kalala Kalamba, Nicholas Sink |

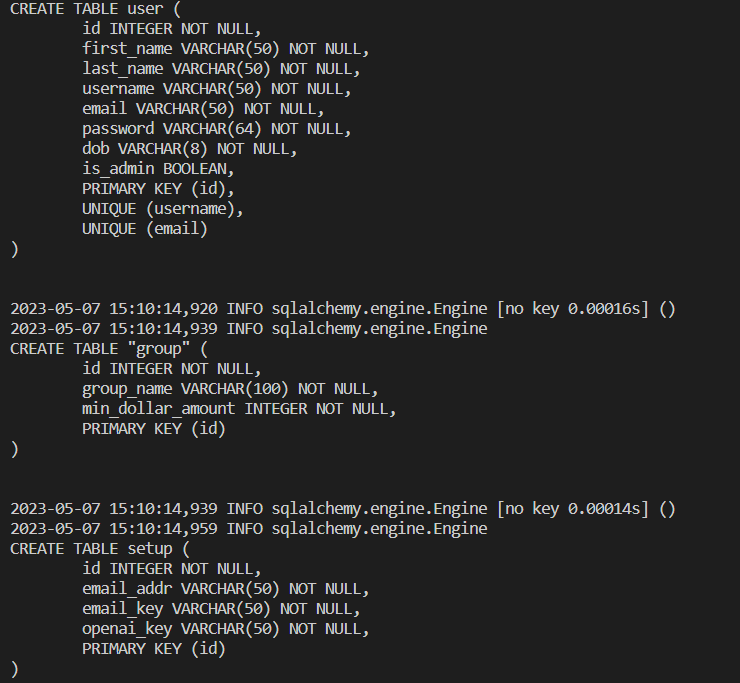
**Revision History**

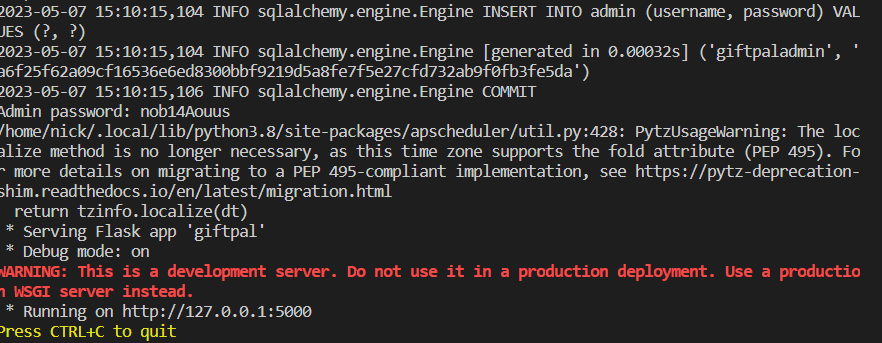
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author (s)** |
| 04/16/2023 | 1.0.0 | Draft Version | Nicholas Sink |
| 04/16/2023 | 1.0.1 | Added Test Description and Expected Result for Normal Operations Cases | Bazz Khurshid |
| 04/16/2023 | 1.0.2 | Mapped Test Cases to Functional Requirements | Nicholas Sink |
| 04/18/2023 | 1.1 | Fully Revised | Nicholas Sink,  Bazz Khurshid |
| 05/07/2023 | 2.0 | Test Plan updated to reflect final application & test results reported | Nicholas Sink |
| 05/07/2023 | 2.1 | Updated error handling cases & completed final review | Graham Primm |

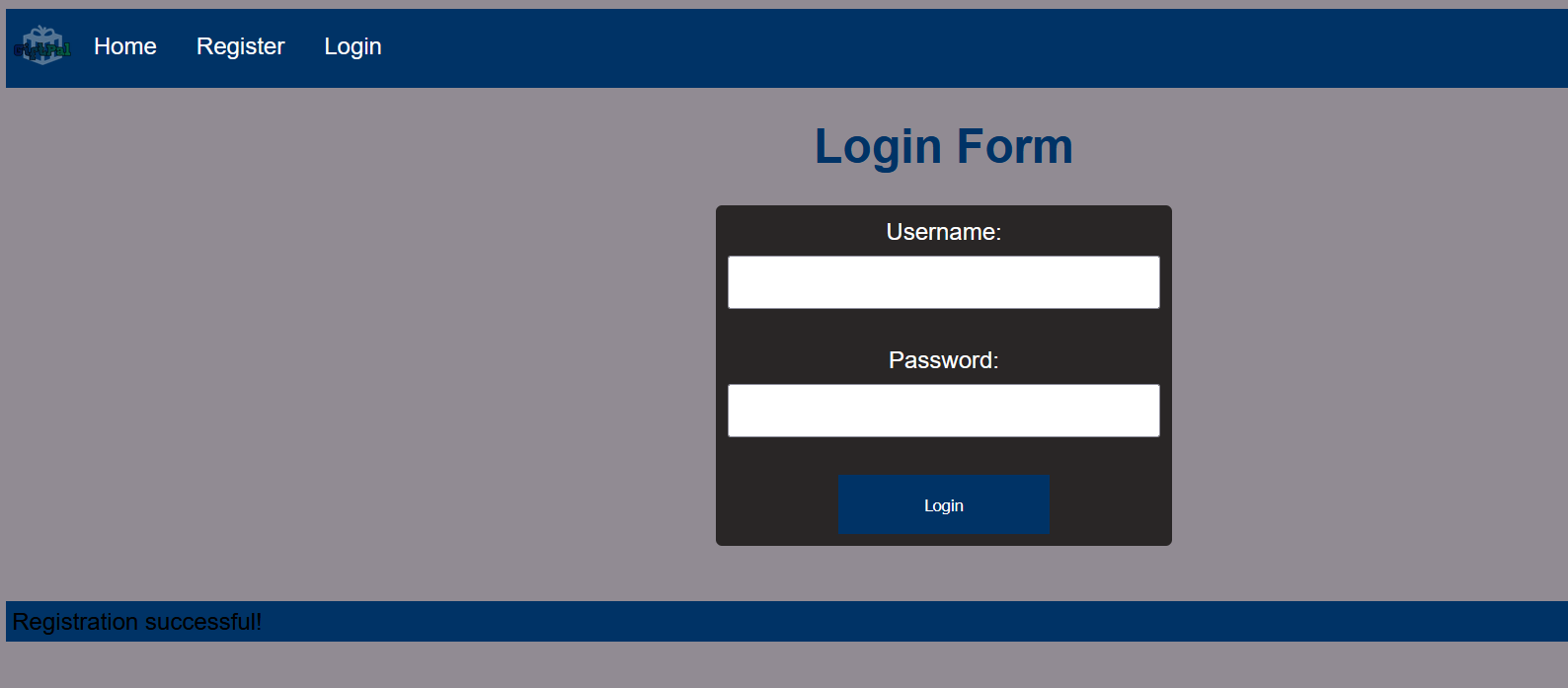
**Project Test Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Number** | **Requirement Number** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| 1 | 1, 2, 7, 8, 9, 10, 11, 12, 13, 17, 19, 20, 21, 22 | *Start-Up Scenario:* The application on startup will create the database file and initialize all database tables so they may be utilized to store data. | The console should print the SQL “CREATE TABLE” commands for each table. | The console prints the SQL “CREATE TABLE” commands for each table  **Figure 1** | Pass |
| 2 | 27 | *System Admin Account Creation & Password Generation:* The system administrator account “giftpaladmin” is created on application startup and the password is randomly set and displayed to the console. | The console should print the SQL ”INSERT INTO admin (username, password) VALUES (?, ?)” and the random admin password should be displayed to the console. | The console prints the SQL ”INSERT INTO admin (username, password) VALUES (?, ?)” and the random admin password is displayed to the console.  **Figure 2** | Pass |
| 3 | 1, 4, 7, 8, 9, 17 | *User Account Creation*: The application displays a registration form to the user. The user will enter their first name, last name, username, email, password, confirm the password, date of birth (DOB), and select “Register” | The user should be redirected to the login page and a message that states “Registration successful!” should be displayed to the user | The user is redirected to the login page and a message that states “Registration successful!” is displayed to the user  **Figure 3** | Pass |
| 4 | 1, 17 | *User Login*: The application displays a login form to the user. The user will enter their username and password and select “Login” | The user should be logged in and redirected to their profile page that states “Welcome, {username}!” | The user is logged in and redirected to their profile page that states “Welcome, {username}!”  **Figure 4** | Pass |
| 5 | 3, 4, 5, 9, 10 | *Loved One’s & Date Addition*: The application displays an Add Event form to the user. The user enters the name, date, and type of the event, and selects “Add Event” | The user should be redirected to their events page and a message that states “Event added successfully!” should be displayed | The user is redirected to their events page and a message that states “Event added successfully!” is displayed  **Figure 5** | Pass |
| 6 | 3, 4, 5, 9, 10 | *Loved One’s & Date Display*: The application displays a page containing all the user’s events that have been successfully stored in the Event database table | A table containing all the user’s events should be displayed to the user | A table containing all the user’s events are displayed to the user  **Figure 6** | Pass |
| 7 | 3, 4, 5, 9, 10, 18 | *Loved One’s & Date Email Reminder*: The application sends a reminder email at midnight on the day of the event to the address provided during user registration, notifying them of their upcoming event. | A email should be received by the user that states “Reminder! Today is the {event.name} {event.type} event. Don't forget to prepare your gift!” | A email is received by the user that states “Reminder! Today is the {event.name} {event.type} event. Don't forget to prepare your gift!”  **Figure 7** | Pass |
| 8 | 6 | *Gift Preference Addition*: The application displays an add wish form to the user. The user will enter the name of their wish and select “Add Wish” | The user should be redirected to their wishlist and a message that states “Wish added successfully!” should be displayed | The user is redirected to their wishlist and a message that states “Wish added successfully!” is displayed  **Figure 8** | Pass |
| 9 | 12 | *Gift Preference Display*: The application displays a page containing all the user’s gift preferences that have been successfully stored in the Wishlist database table | A table containing all the user’s wishes should be displayed to the user | A table containing all the user’s wishes are displayed to the user  **Figure 9** | Pass |
| 10 | 6 | *Gift Preference Deletion:* The application displays a page containing all of the user’s gift preferences and a delete button to the right. The user selects the delete button and the user’s gift preference is deleted. | The page should be reloaded, and the user’s gift preference should be deleted. | The page is reloaded, and the user’s gift preference has been deleted.  **Figure 10** | Pass |
| 11 | 2, 13, 14, 19, 20, 21, 22 | *Group Registration*: The application displays a group registration form to the user. The user enters the group name and sets the minimum dollar amount and selects “Register Group” at the bottom. | The user should be redirected to the Groups page an the message “Group Registration successful!” should be displayed to the user. | The user is redirected to the Groups page an the message “Group Registration successful!” is displayed to the user.  **Figure 11** | Pass |
| 12 | 2, 21 | *Group Display:* The application displays a page containing all the groups that have been successfully stored in the Group database table. | A table containing all the groups should be displayed to the user | A table containing all the groups are displayed to the user  **Figure 12** | Pass |
| 13 | 2 | *Group Member Addition:* The application displays a modify group form to the user. The user enters the username of the group member, selects “Add” and selects “Save Changes” | The new member should be added to the group and the message “User successfully added to group!” should be displayed to the user | The new member is added to the group and the message “User successfully added to group!” is displayed to the user  **Figure 13** | Pass |
| 14 | 2 | *Group Member Deletion:* The application displays a modify group form to the user. The user enters the username of the group member, selects “Delete” and selects “Save Changes” | The member should be deleted from the group and the message “User successfully deleted from group!” should be displayed to the user | The member is deleted from the group and the message “User successfully deleted from group!” is displayed to the user  **Figure 14** | Pass |
| 15 | 19, 20, 22 | *Group Admin Registration:* The application displays a modify group form to the user. The user enters the username of the group member, selects “Make Admin” and selects “Save Changes” | The user should be made an admin of the group and the message “User successfully made admin of group!” should be displayed to the user | The user is made an admin of the group and the message “User successfully made admin of group!” is displayed to the user  **Figure 15, 16** | Pass |
| 16 | 11, 14, 15, 16, 18, 23, 24, 25, 26 | *Gift Exchange Initiation*: The application displays the gift exchange group page which displays all members in the group. The user selects “Initiate Gift Exchange” and gift pairs are formed and displayed in a table to the user. An email to the email addresses provided during user registration is also sent to each user notifying them of their gift exchange pair. | A table containing gift exchange pairs should be displayed to the user. An email containing the name of the gift exchange group and gift pair should be received by the user. | A table containing gift exchange pairs is displayed to the user. An email containing the name of the gift exchange group and gift pair is received by the user.  **Figure 17, 18, 19** | Pass |
| 17 | 1 | *Shut-Down Scenario*: The shut down option will occur through the terminal. The systems administrator will end the program and verify all processes have stopped. The program termination can be verified by running the command “ps -a | grep python3” | The application output should no longer be printed to the console. The process command should result in no output. | The application output will no longer be printed to the console. The process command results in no output.  **Figure 20, 21, 22** | Pass |
| 18 | 1 | *Error-Handling Scenario (Invalid User Password)*: The user will enter their information, including password, into the registration page and select “Register”. The account will not be created, the user is redirected back to the registration page with a message stating that the password requirements were not met. | The Application should flash a message to the user stating " Password must be at least 8 characters long!" | The Application flashes a message to the user stating " Password must be at least 8 characters long!"  **Figure 23, 24** | Pass |
| 19 | 1, 2 | *Error-Handling Scenario (Duplicate Username)*: The user will enter their information, including username, into the registration page and select “Register”. The account will not be created, the user is redirected back to the registration page with a message stating that the username specified is not allowed. | The Application should flash a message to the user stating "This username is already taken!" | The Application flashes a message to the user stating "This username is already taken!"  **Figure 25, 26** | Pass |
| 20 | 1, 7 | *Error-Handling Scenario (Duplicate Email)*: The user will enter their information, including email, into the registration page and select “Register”. The account will not be created, the user is redirected back to the registration page with a message stating that the email specified is not allowed. | The Application should flash a message to the user stating “An account with this email already exists!” | The Application flashes a message to the user stating “An account with this email already exists!”.  **Figure 27, 28** | Pass |
| 21 | 2, 13 | *Error-Handling Scenario (Duplicate Group Name)*: After the user enters their information into the group registration form, the group will not be created, the user is redirected back to the group registration page with a message stating that the group name specified is not allowed. | The Application should flash a message to the user stating that " This group name is already taken!” | The Application flashes a message to the user stating that " This group name is already taken!”  **Figure 29, 30, 31** | Pass |
| 22 | 2, 13, 20 | *Error-Handling Scenario (Nonexistent user for group member addition)*: The application displays the modify group form. The admin enters the username of a nonexistent user to and selects “Save Changes" | The application should display a message that states “User does not exist!  ” to the admin | The application displays a message that states “User does not exist!  ” to the admin.  **Figure 32, 33** | Pass |
| 23 | 2, 13, 19, 20 | *Error-Handling Scenario (User attempts to modify a group they are not the admin for)*: A regular user selects the “Modify Group” button from the groups page of a group they are not the admin for. | The application should display a message that states “You are not an admin of this group!  ” to the user. | The application displays a message that states “You are not an admin of this group!  ” to the user.  **Figure 34** | Pass |

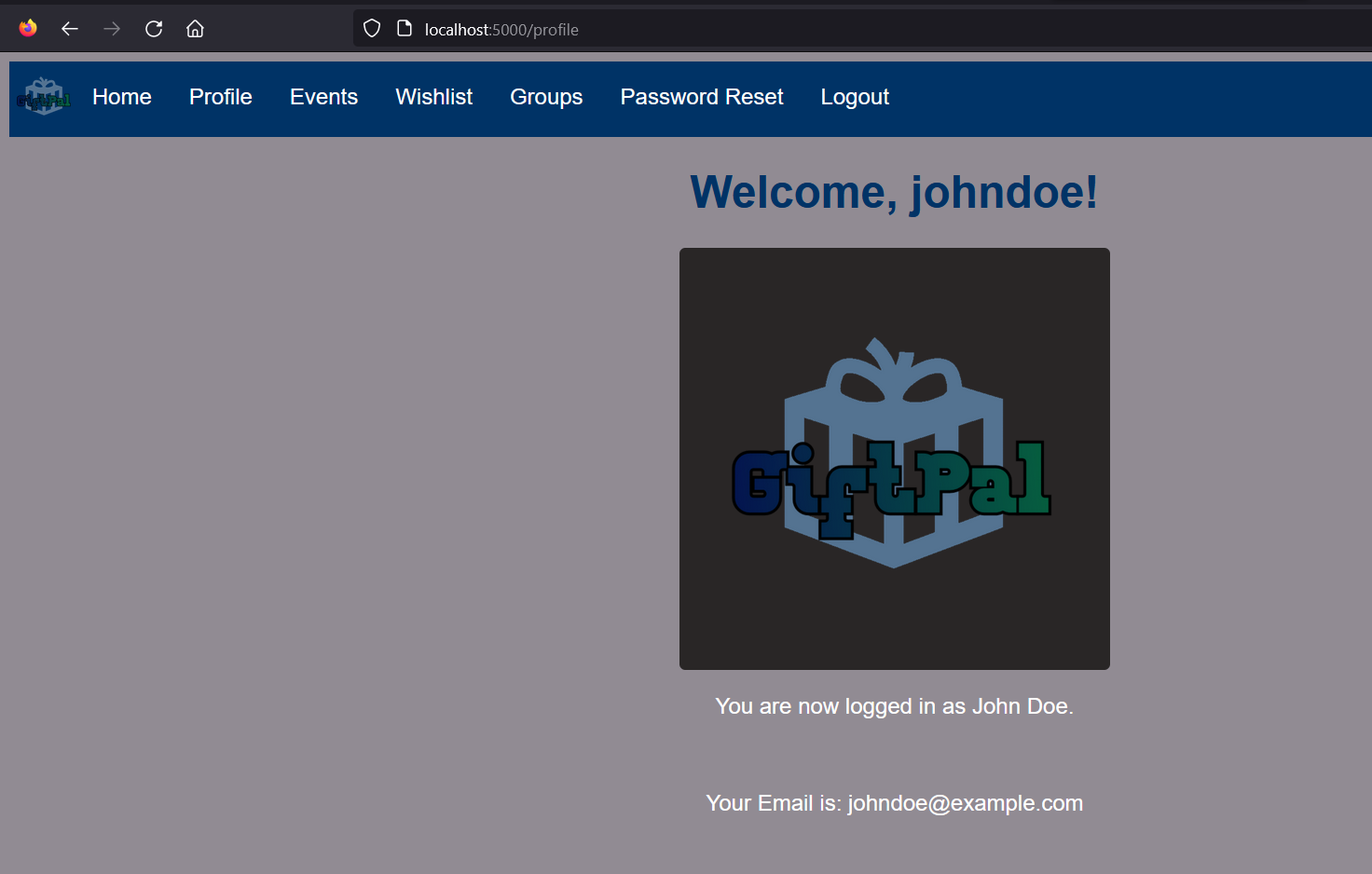
**Figure 1:**



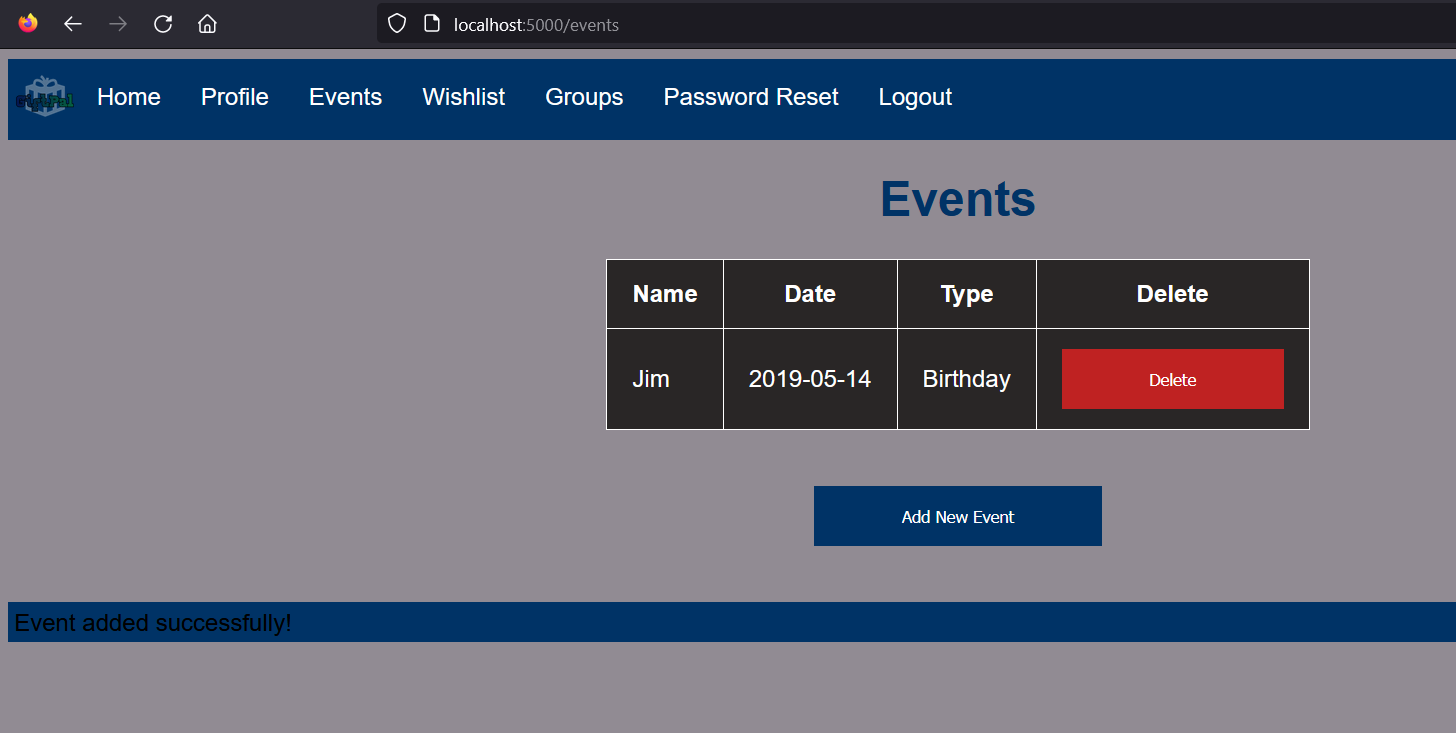
**Figure 2:**  
  
**Figure 3:**



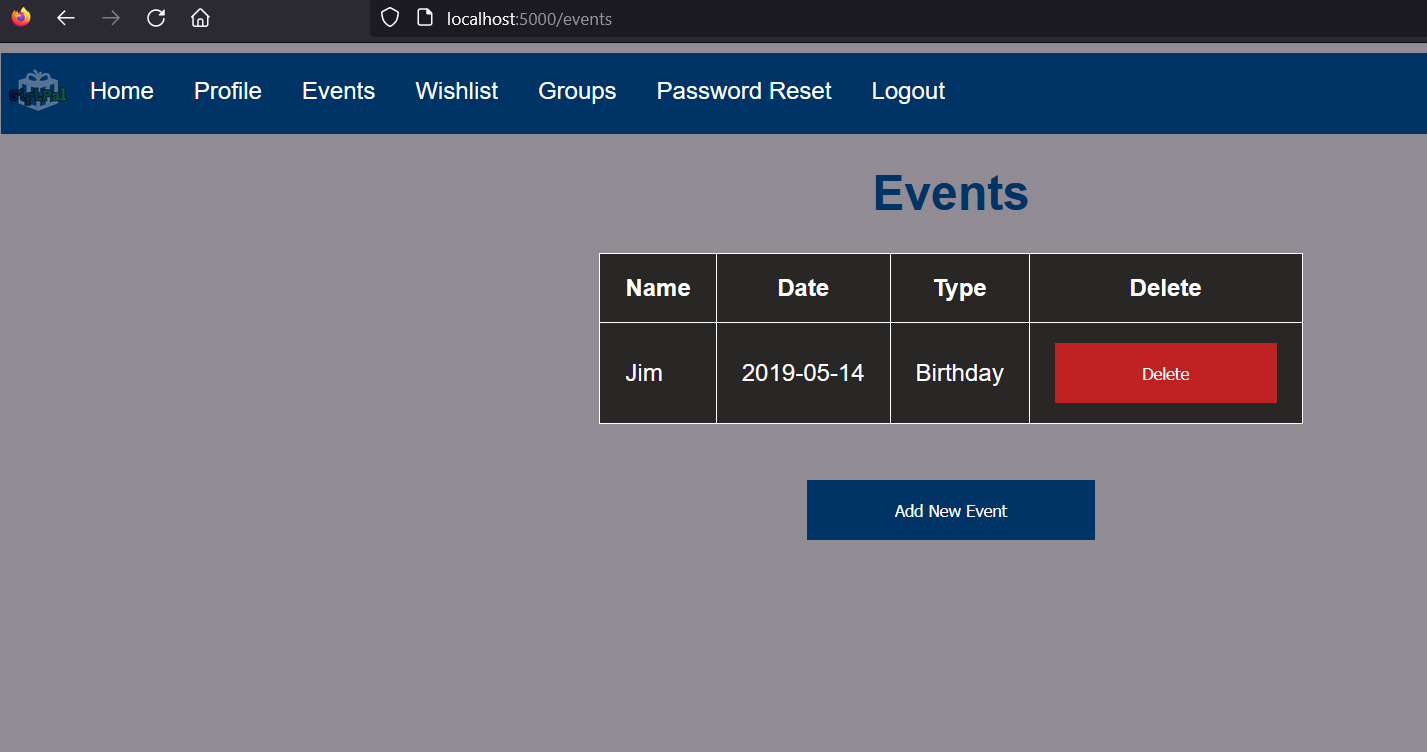
**Figure 4:**



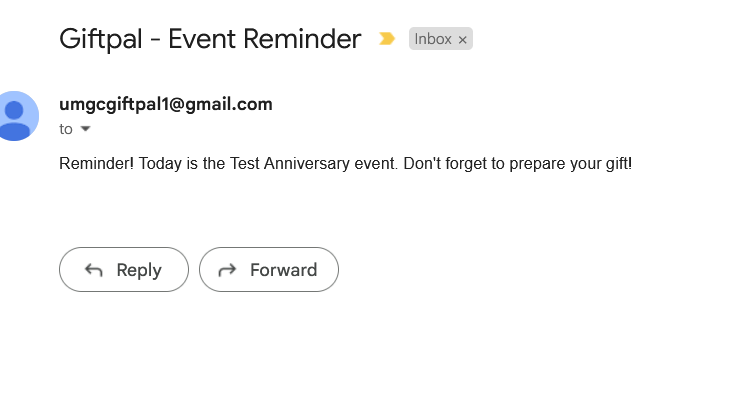
**Figure 5:**



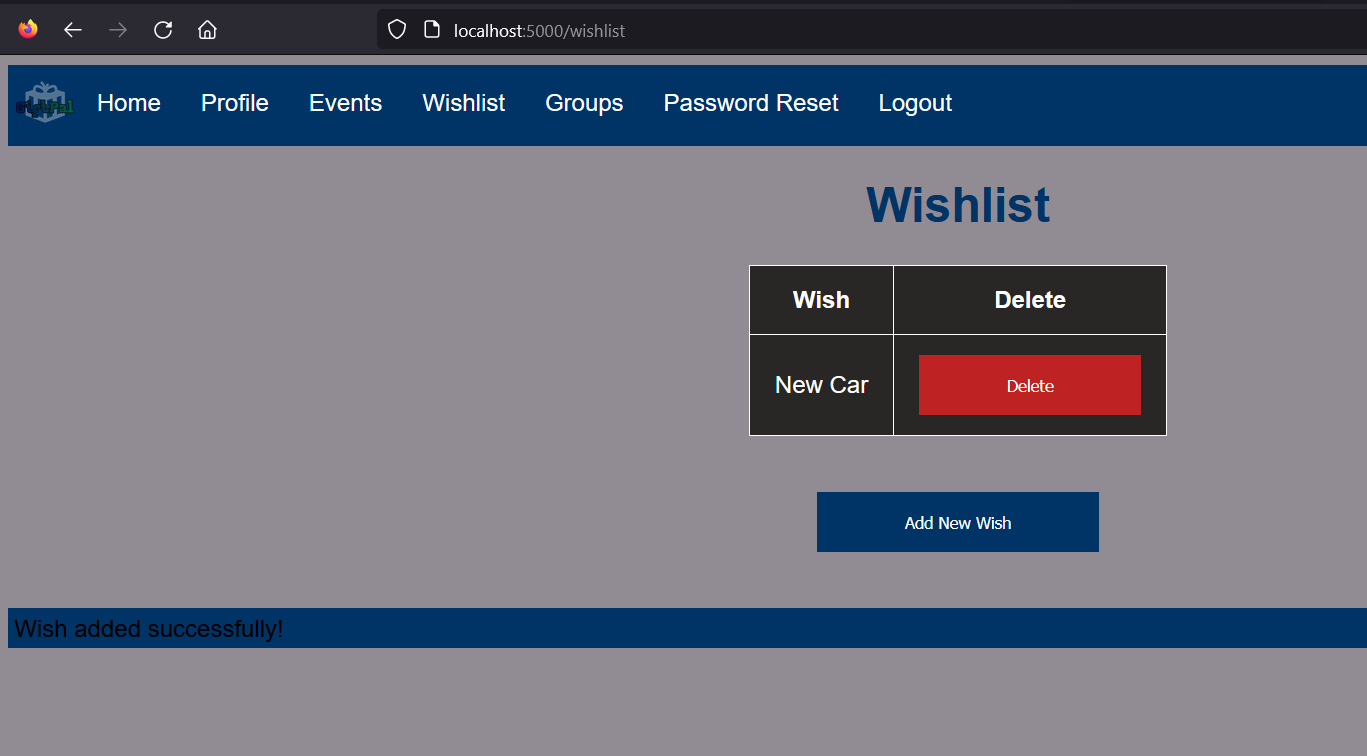
**Figure 6:**



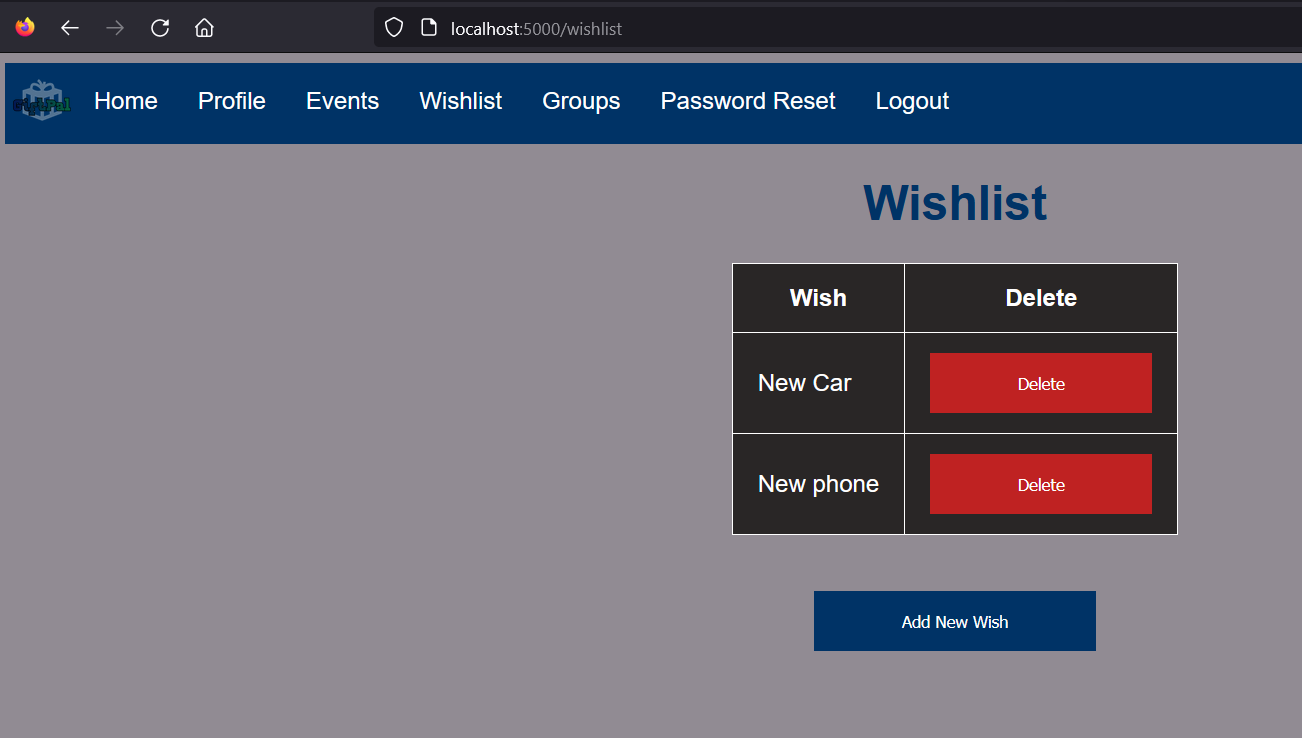
**Figure 7:**



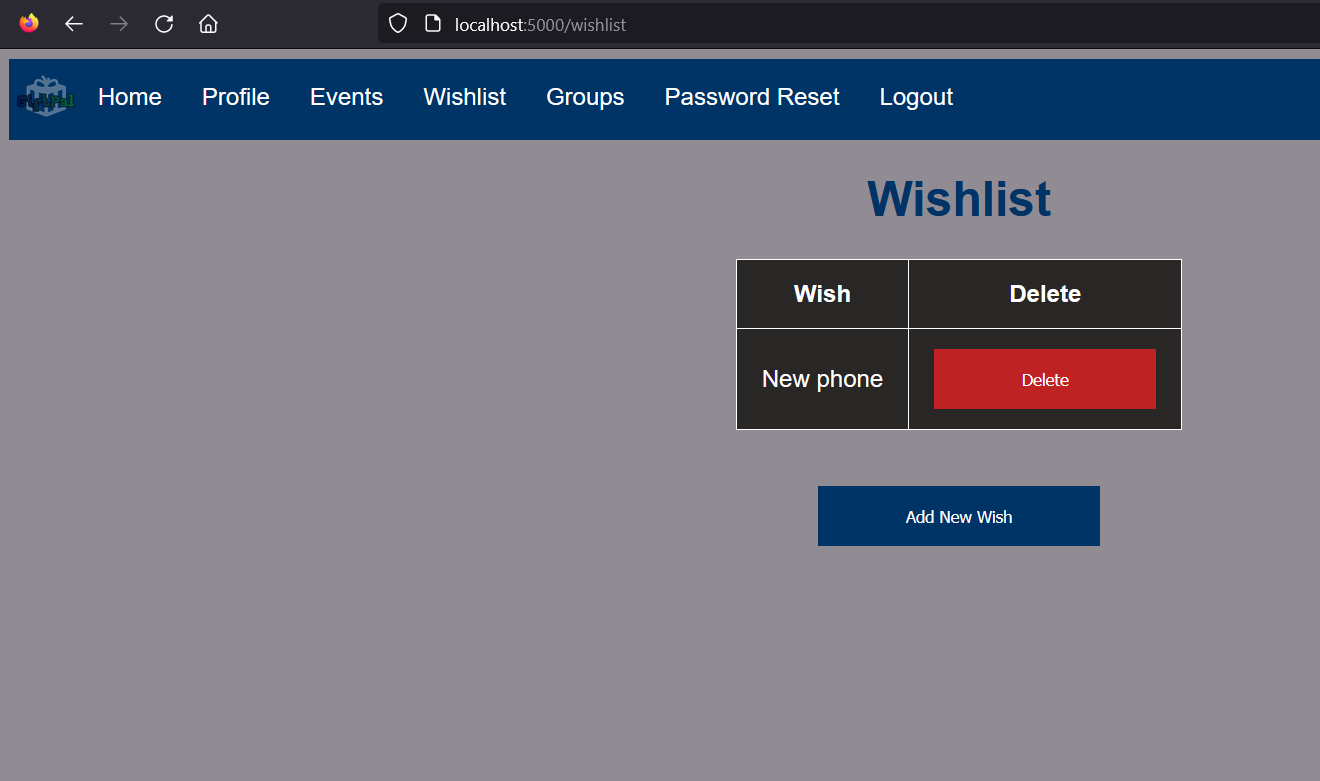
**Figure 8:**



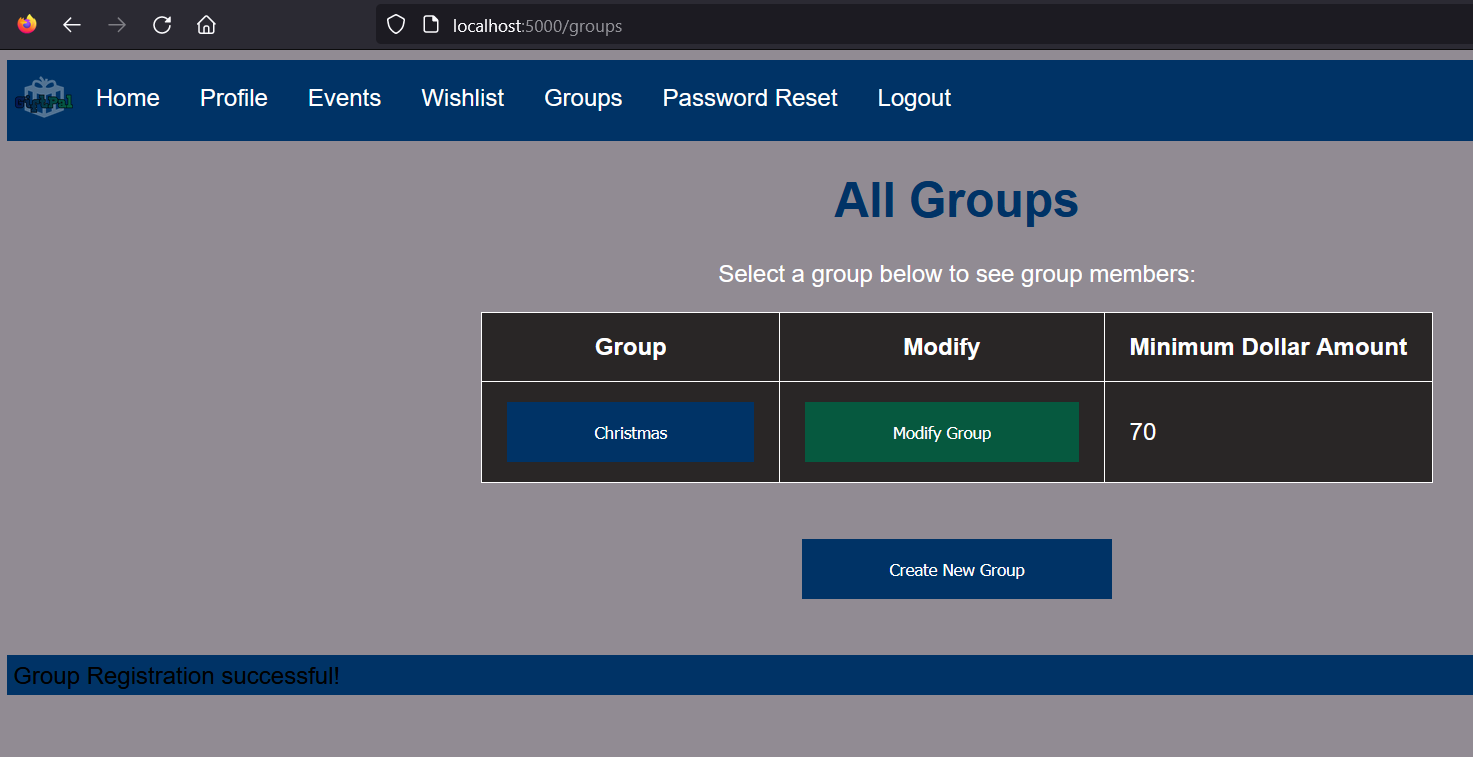
**Figure 9:**



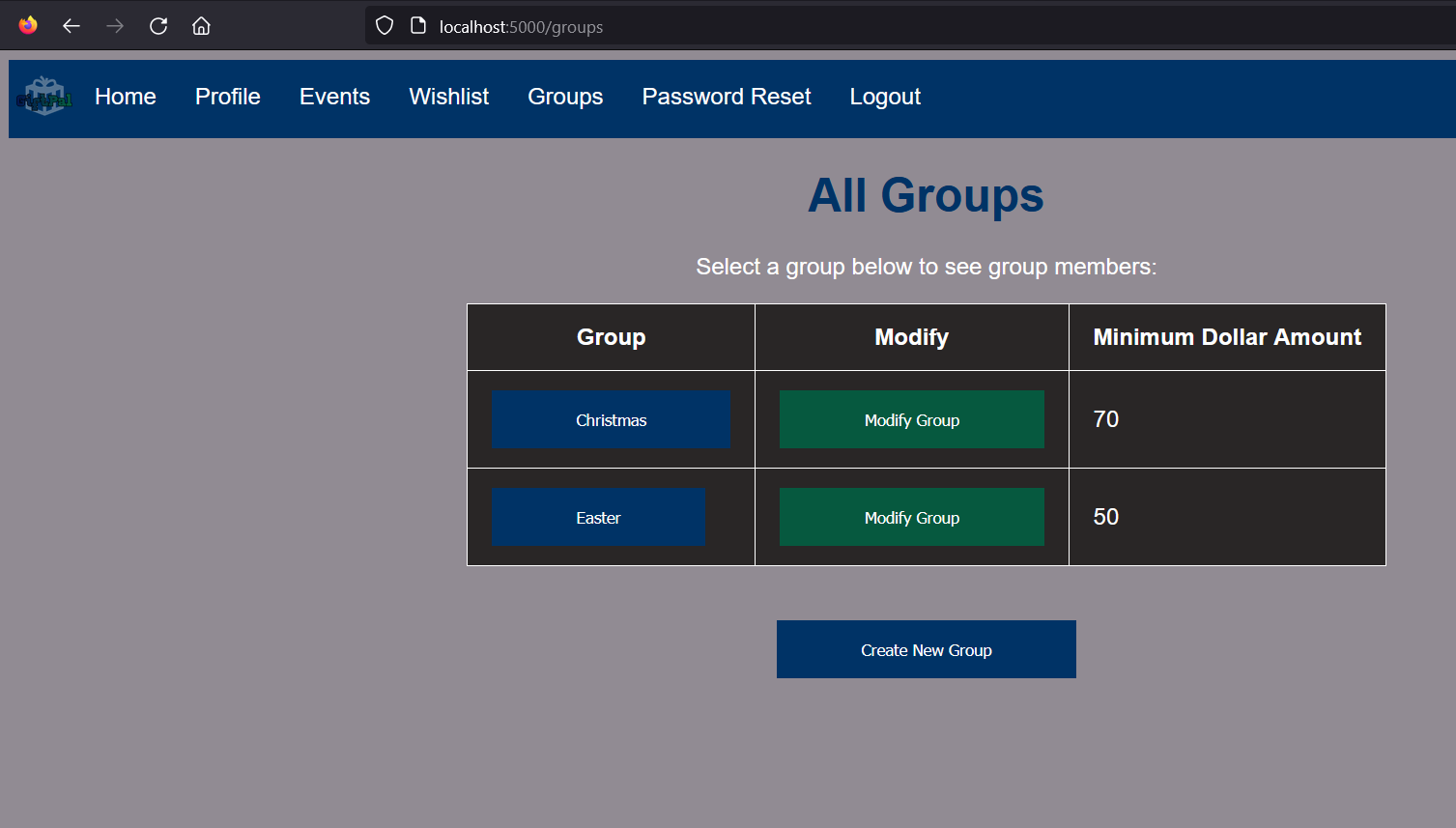
**Figure 10:**



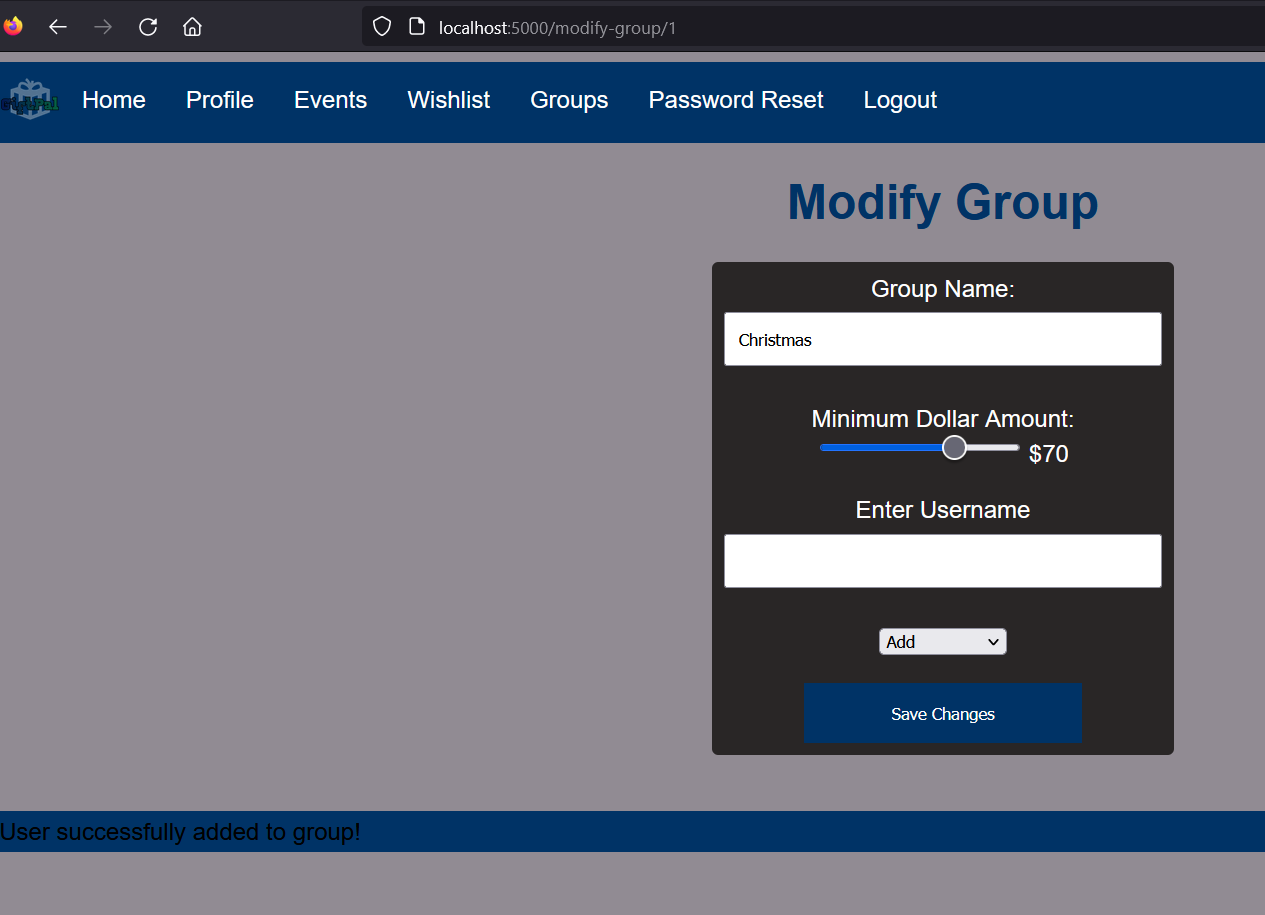
**Figure 11:**



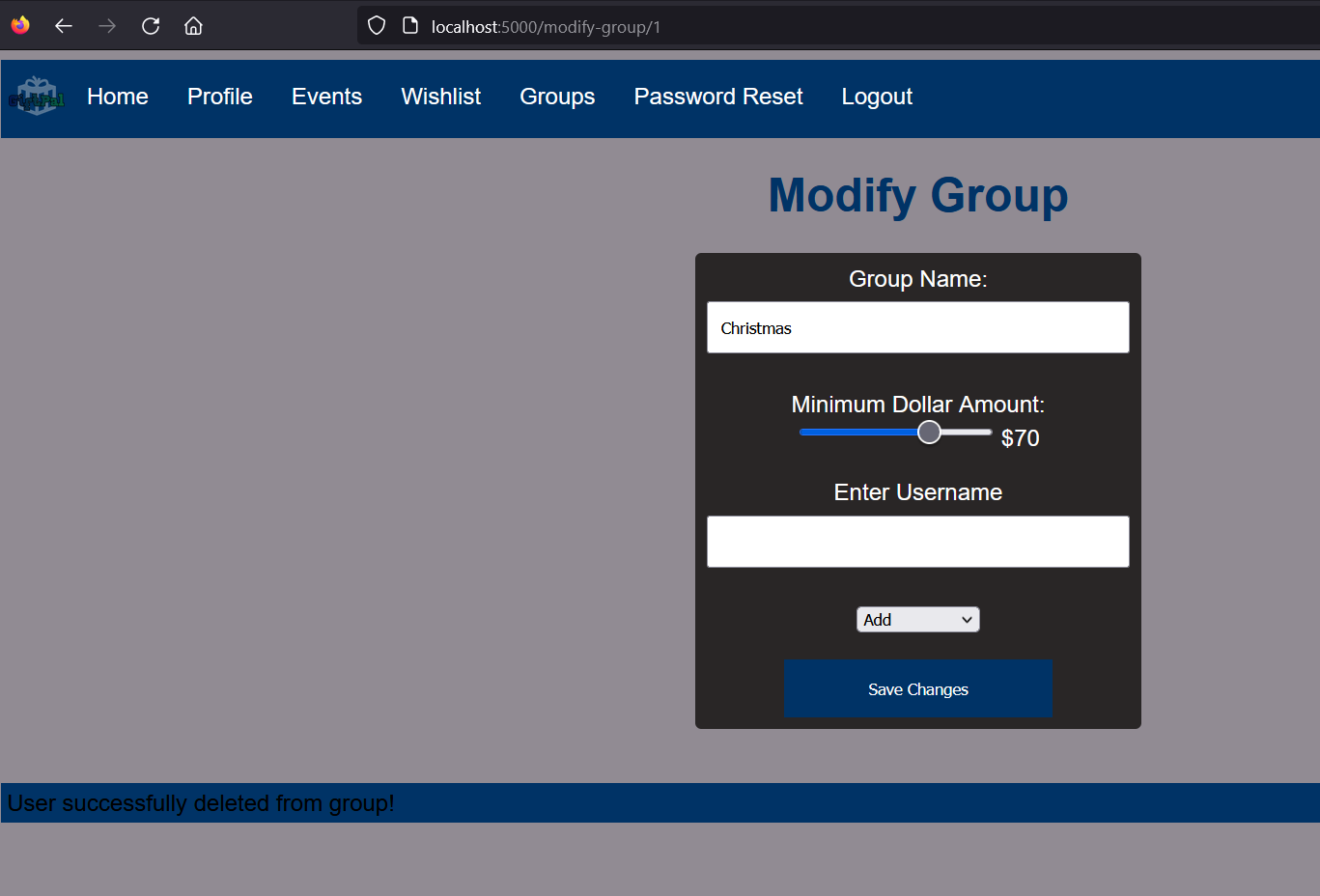
**Figure 12:**



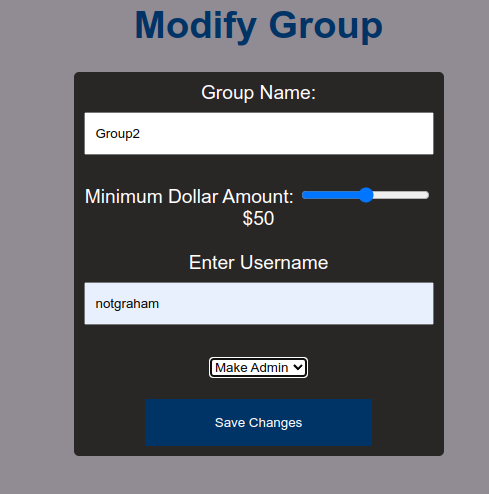
**Figure 13:**



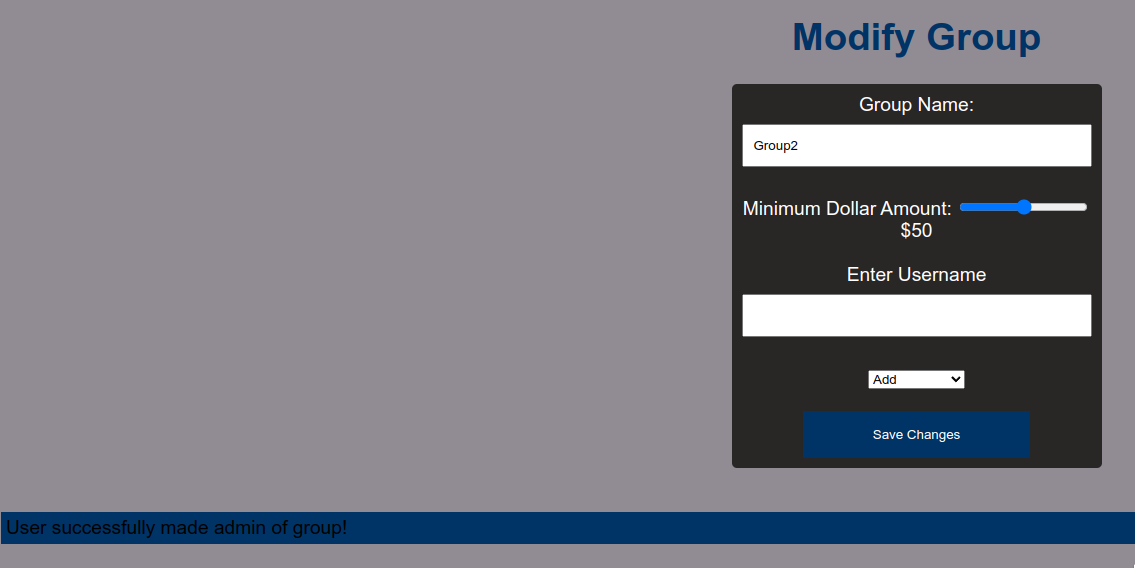
**Figure 14:**



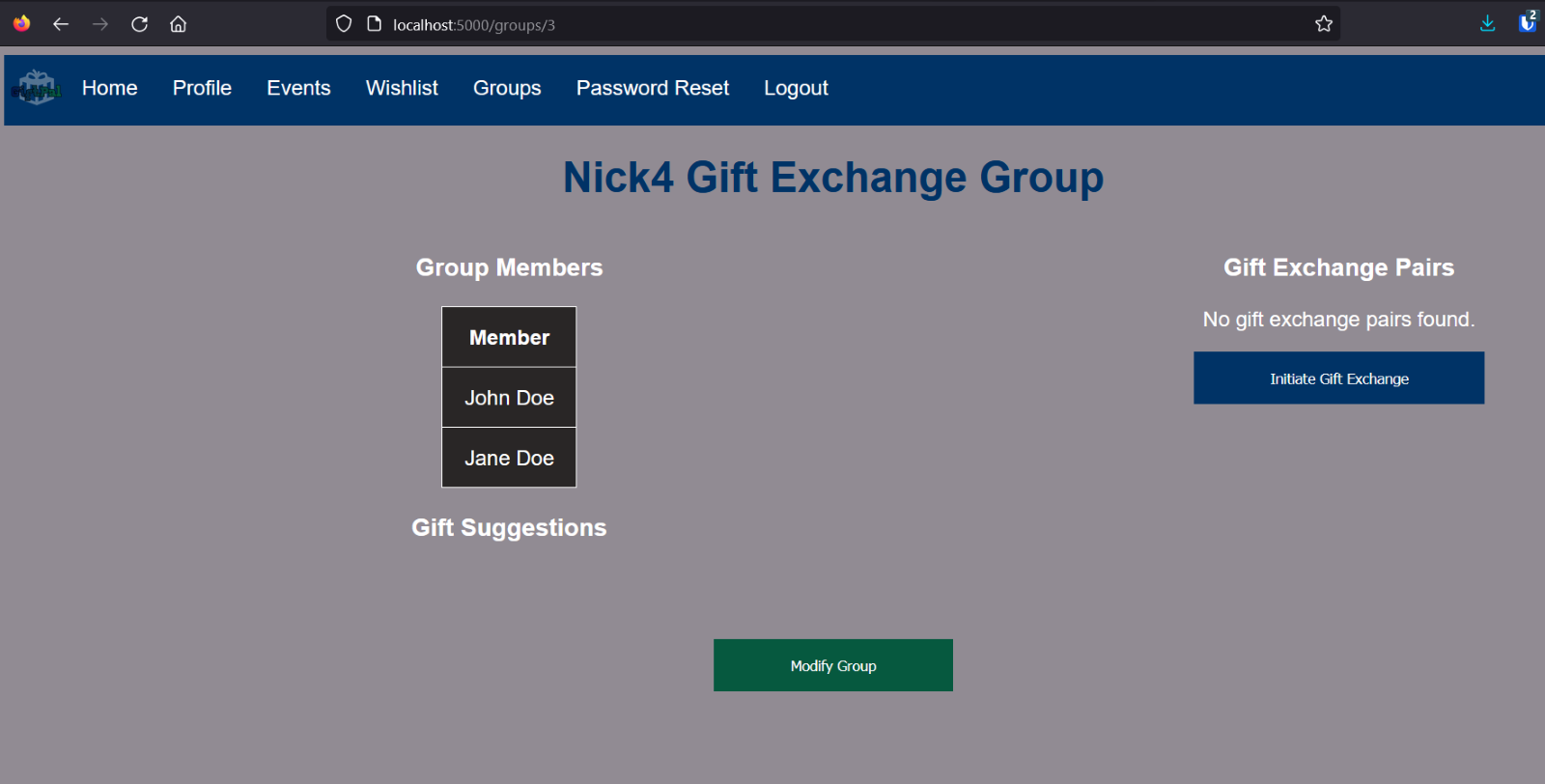
**Figure 15:**



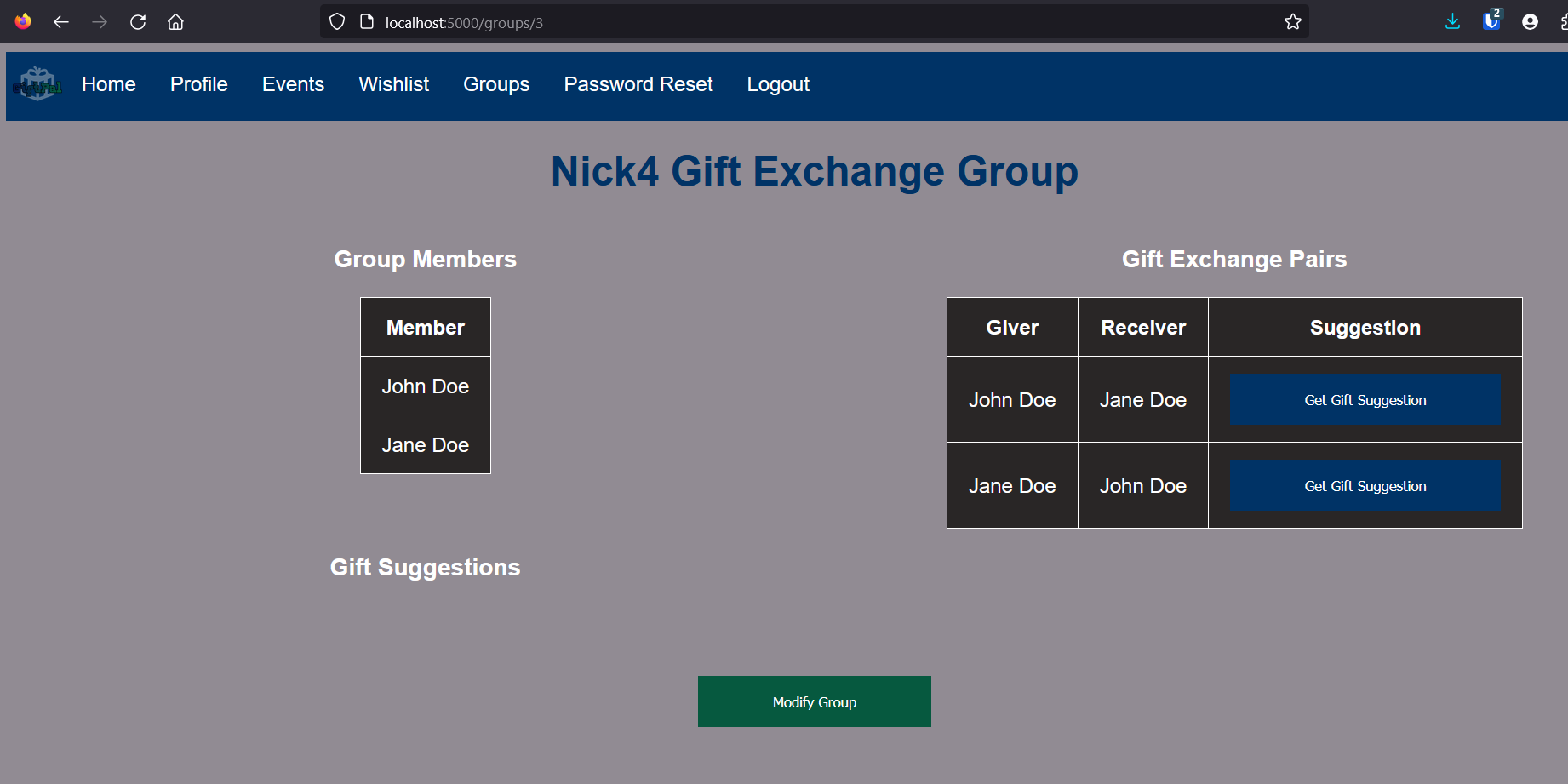
**Figure 16:**



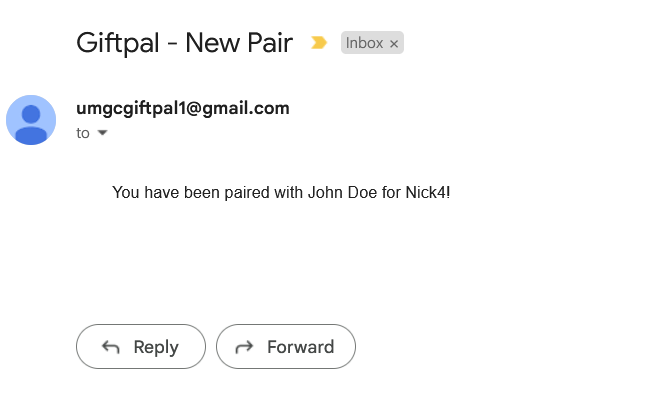
**Figure 17:**



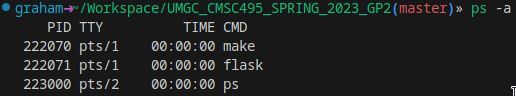
**Figure 18:**



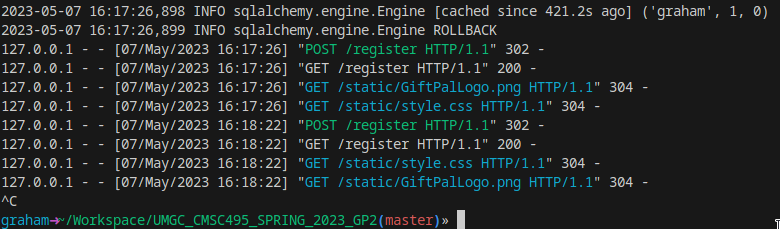
**Figure 19:**



**Figure 20:**



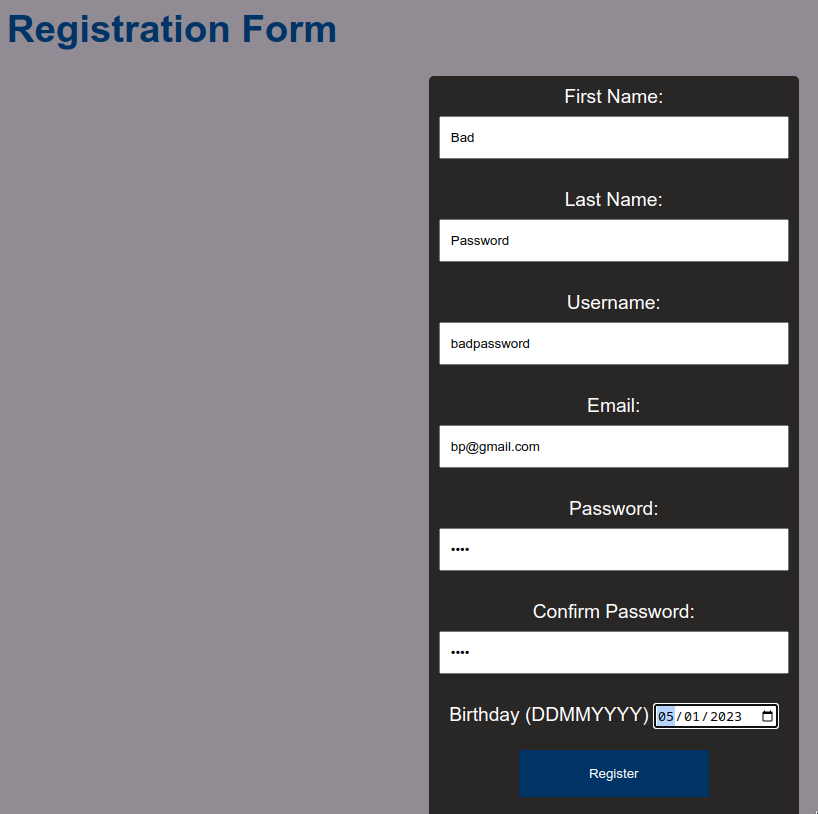
**Figure 21:**



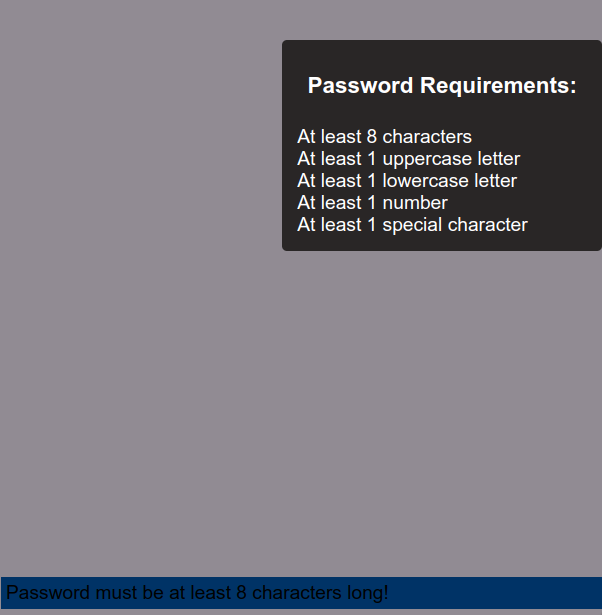
**Figure 22:**



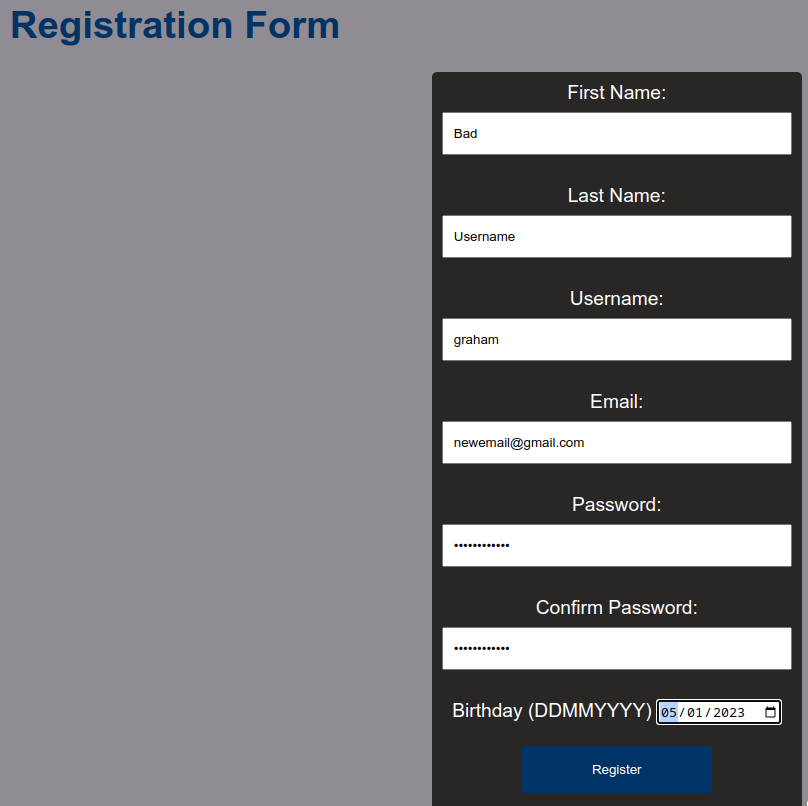
**Figure 23:**



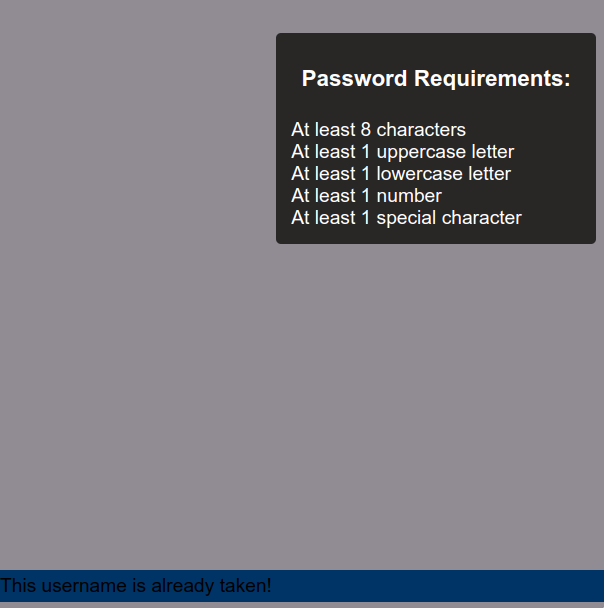
**Figure 24:**



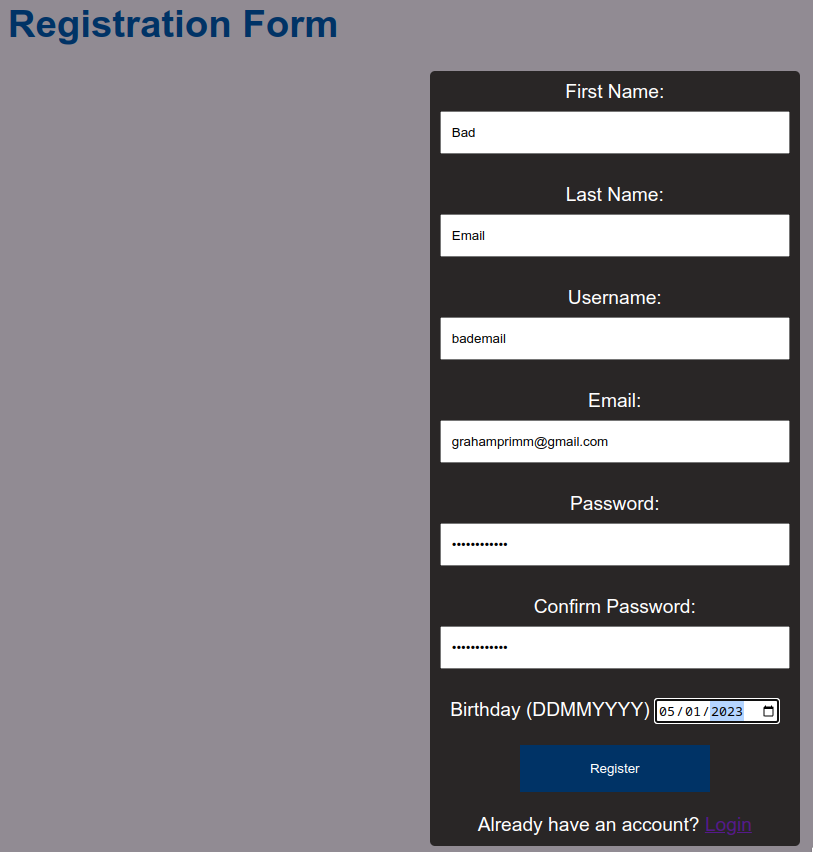
**Figure 25:**

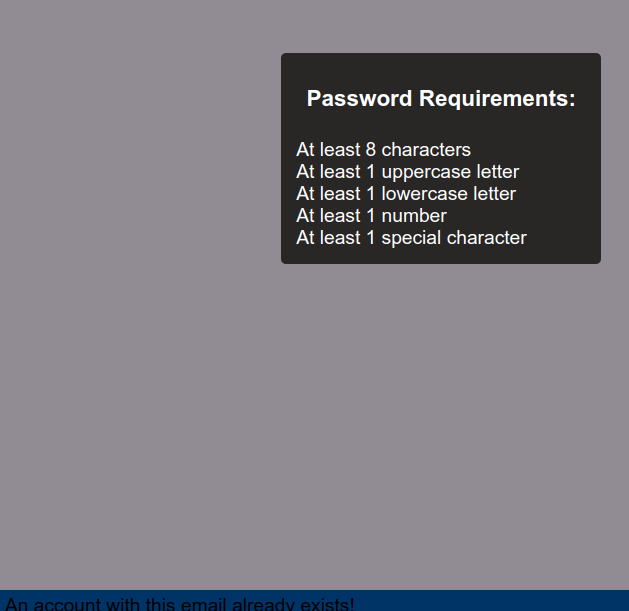


**Figure 26:**

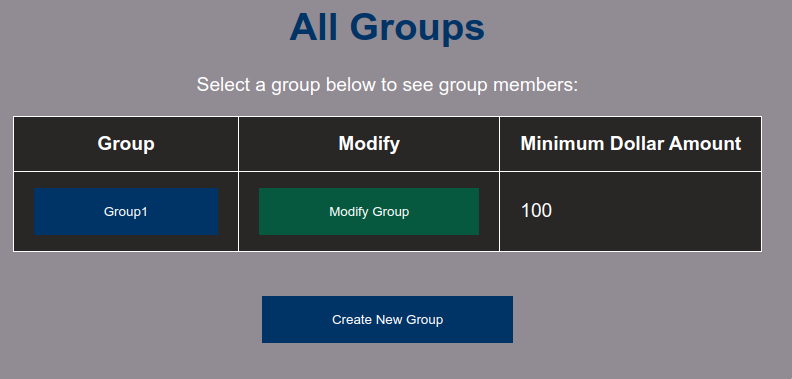


**Figure 27:**

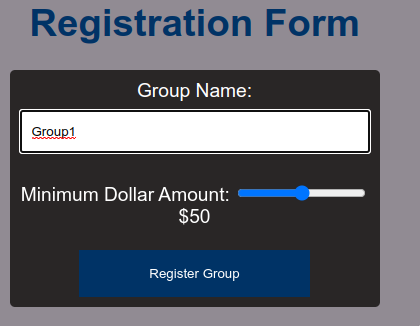


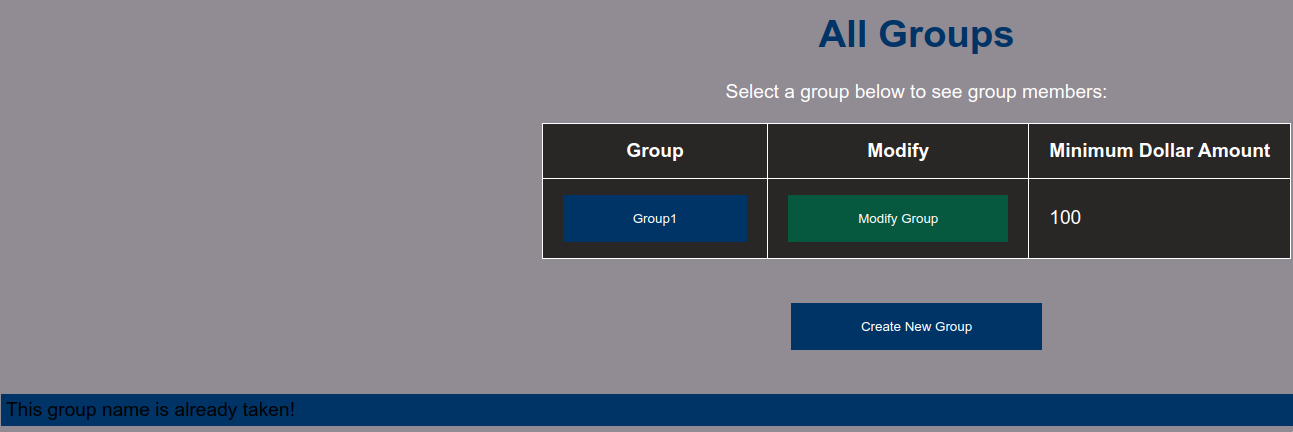
**Figure 28:**  


**Figure 29:**

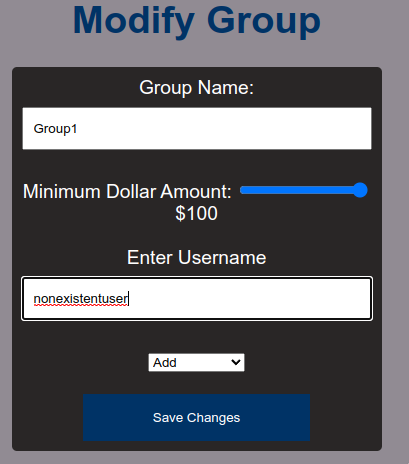


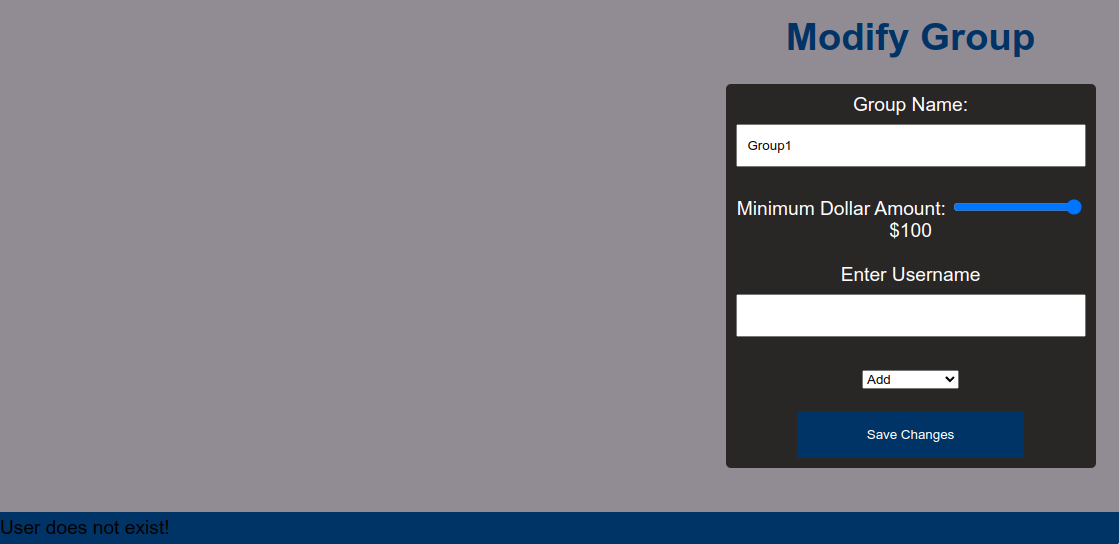
**Figure 30:**



**Figure 31:**  


**Figure 32:**



**Figure 33:**  


**Figure 34:**

