Dimitrios Stathis

Team Member Information

* + stathisd1@mymail.nku.edu

Stage B

Feature 1: Shopping System

The feature I plan to add for the wiki system is a shopping system feature that will include the ability to add items (games) to a cart system. On top of this, a user will be able to apply shopping information for purchasing purposes and can be saved to a database. The user will be able to remove items and add items to a shopping cart and then will be able to proceed to a checkout.

Rationale

***WHY:***The feature is needed so that it gives a user the ability to purchase a game of their choice. It gives the user the option to buy as many games as they want or remove which games they don't want. The database is also needed so that way it can store the users information and shopping history of any purchases they make.

***Technology:***The technology for this feature will be using Python with its Flask and Jinja frameworks to route the user to the cart and purchasing page of the feature. SQL will also be used in this feature to provide the ability to save the users information about purchasing in a database.

Requirements

As a user, I want to be able to pick a game of my choice that I like and be able to purchase it through a purchasing system.

Deliverables

* Design:

A diagram of a computer program

Description automatically generated with medium confidence

As of right now I plan on creating a Shopping cart class that will be inherited by the Shopping Form class. This class will contain information for credit card info, user email, and user purchasing history. I mainly want this Shopping Form class to inherit from FlaskForm and then I can declare an instance of this object in order to link this form to a function called shopping\_cart() which will render a template called shopping.html which will be using Flask and Jinja.

* code:

[GitHubLink-Shopping-SystemLinks to an external site.](https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping_system)

* test:

[GitHubTests-Shopping-SystemLinks to an external site.](https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping_system/Tests)

* documents: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system

Test Plan

I will make tests to make sure a game can be added and removed from the cart. I will also test that information can be saved to an SQL database. I will also test the linking of the add to cart button on each of the game displays so that it functions correctly and directs you to the Cart page with your items listed.

Milestones and Deadline

* + **Deadline**: 3/17/2024
  + Milestone 1
    - Date: 3/17/2024
    - Goal: Completing Stage B Outline and preparation for the Software Feature
  + Milestone 2 **Deadline**: 3/24/2024
    - Date: 3/17/2024
    - Goal: Work on getting the Add to Cart button on the Navigation tab and have it link to a template called shopping\_cart.html and have it display Shopping Form Information. Also have a shopping\_cart function in routes.py that will perform this action.
  + Milestone 3 **Deadline:**3/31/2024
    - Date: 3/24/2024
    - Goal: For this week the main goal is to implement an SQL database so that way the users information is stored and saved in a database.
  + Milestone 4 **Deadline:**4/7/2024
    - Date: 3/31/2024
    - Goal: For this week the main goal is to add a purchasing form that will accept the users payment method information.
  + Milestone 5 **Deadline:** 4/14/2024
    - Date 4/7/2024
    - Goal: The goal for this week is to give the user the ability to add and remove games to and from their designated cart.
  + Milestone 6 **Deadline:** 4/18/2024
    - Date 4/14/2024
    - Goal: The goal for this week is to be able to add and remove multiple games at the same time since now i can add and remove just one game and finish up my shopping system feature. "Fine tuning".

Risk Analysis

1. HW3 (4/10/2024)
2. Midterm 2 (4/10/2024)
3. HW4 (4/24/2024)
4. Interactions with other team members code and seeing if everything will function correctly at the same time. (C-Stage Implementation time period).

Stage C: Progress

Feature Implementation

Week 1

Summary

For this week, I was able to accomplish what I had myself working on for my feature for my milestone. I first was able to create a button that will take you to the cart page. Once in the cart page, personal information input boxes are then prompted for the user and a box that will store the list of items the user wants to purchase was also added. This box will let the user have the ability to see what items are in their cart for them to purchase. To accomplish this i added a new ShoppingInfoForm class that extends FlaskForm to provide the user the ability to add their information. I also added a shopping\_cart function that renders a new template which i also added called shopping\_cart.html which routes you to the cart page. I also added code inside of base.html to be able to create the 'Cart' button.

Milestones or risks in this week

For this week i accomplished my second milestone task of working on getting the Cart button on the Navigation tab and have it link to a template called shopping\_cart.html and have it display Shopping Form Information and have a shopping\_cart function in routes.py that will perform this action. The way i dealt with risks were i started my project early that way i had enough time to take care of what risks i had for this past week. I started to work on my feature 3/19/2024 around 7:00pm.

Mainly Integrated tests:

* Test the 'Cart' button and see if it takes you to the 'Cart' page.
* Test and see if the ShoppingInfoForm adds input functionality in the cart page.
* Test and see if shopping\_cart function in routes.py performs these functionalities.

Design

Again as of right now my progress for my feature has been that I have added a 'Cart' button that once clicked on can take the user to their designated 'Cart' page. I then was able to add user input features for personal information such as name, address, email, phone number, etc. I also added an items field/ box that will store and show the user what items are in their cart.

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Code

Added a new form in forms.py called ShoppingInfoForm. A function called shopping\_cart was added in routes.py which renders a newly created template called shopping\_cart.html. I also added 3 lines of code to base.html for the cart button and 1 line of code for a css file I made that I use for my shopping\_cart.html template.

LoC added in forms.py: **10**, Total LoC: **54**

LoC added in routes.py: **6**, Total LoC: **146**

LoC added in base.html: **4**, Total LoC: **88**

Newly added template shopping\_cart.html total LoC: **49**

Newly added css file called styles.css total LoC: **14**

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Week 2

Summary

For this week, I was able to accomplish what I had myself working on for my feature for my third milestone goal. This week i was able to create and implement an SQL database which will track and save user information when applied inside the cart. The user is prompted for Name, Address, City, State, Zip, Email, and Phone Number. The database I created now is able to save all of that information. I had to install a new package called flask-sqlalchemy which gave me the ability to allow communication between the flask application and the database. This task for this past week took me the longest to create and figure out and test. Mainly Integrated tests.

Milestones or risks in this week

For this week I accomplished my third milestone goal of implementing an SQL database so that way the users information is stored and saved in a database. The database is SQLite. This part of my feature took me the longest to implement and I had many risks as apart of this. I had an exam this week and homework in other classes.

Mainly Integrated tests:

* Tested each input would save to a database.
* Test and see the functionality of the SQL database.

Design

Again as of right now my progress for my feature has been that I have added a functioning database to store user personal information such as Name, Address, City, State, Zip, Email, and Phone Number. Once that information is inputted and submitted by the user it will then be stored inside the database.

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Code

Added three new files in my project for the database and modified my shopping cart function for the database functionality/communication between the ShoppingInfoForm (user input fields) and the and the actual database.

Added another function in routes.py called success\_page which once the submit button in shopping\_cart is pressed by the user all the information will be saved to the database and the user will be redirected to another page with a message popping up displaying "Success" and gives you the option to return to a blank/brand new cart page. A success\_page.html was also added for this functionality.

LoC added in routes.py: **19**, Total LoC: **166**

LoC added in \_\_init\_\_.py in web: **4**, Total LoC: **46**

LoC added in Riki.py in web: **3**, Total LoC:**10**

Newly added template success\_page.html total LoC: **9**

Newly added extensions.py total LoC: **2**

Newly added models.py total LoC: **10**

Newly added shopping\_cart\_db file directory with a shopping\_cart\_db.py total lines of code: **18**

**GitHub Link**: [https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/RikiLinks to an external site.](https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping_system/Riki)

Week 3

Summary

For this week, I was able to accomplish what I had myself working on for my feature for my fourth milestone goal. This week i was able to create another page but this one was to gather the users Payment Information. Once the 'Proceed to Checkout' button is clicked on by the user after inputting their personal information, first their data gets stored into the database i created and then the user is taken to another page where they will have to input and submit their payment information. Once the 'Checkout' button is clicked on the user will have a 'Success' message and page pop up once submission is complete.

Milestones or risks in this week

For this week I accomplished my fourth milestone goal which was to add a purchasing form that will accept the users payment method information. The user has input fields to place their information in and can submit it all. The Risks for this week were working on HW3 and other HW for other classes.

Mainly Integrated tests:

* Test and see if the PurchasingForm adds input functionality in the cart page.
* Tested to see if the 'Proceed to Checkout' button once clicked saves your personal information and routes you to the Page for purchasing info.
* Tested to see if input functionality works for the purchasing info page and tested the 'Checkout' button once clicked would take you to an end page for success of submission with a success message.
* Tested to see if you could route back to the 'Home' page once submission was complete.

Design

For the design i added a new from in forms.py called PurchasingForm which extends FlaskForm. The input fields for the PurchasingForm i had were the card number, name on card, expiration date of card, and cvv of the card and a checkout button. I added a new function in routes.py called purchasing and i modified the shopping\_cart function in routes to redirect you to the purchasing information page once the 'Proceed to Checkout' button is clicked. Modified the success\_page to have a link to redirect you to the Home page once submission is completed.

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Code

Added a new template called purchasing\_form.html and modified routes.py by adding a new function called purchasing which will render the purchasing\_form.html template.

Added another form in forms.py called PurchasingForm.

Added a 'Country' field with a drop down menu to give the user the option to pick which country they are from.

Modified shopping\_cart.html for the Country field.

Can now save Country field in the database.

LoC added in forms.py: **7**, Total LoC: **63**

LoC added in routes.py: **9**, Total LoC: **175**

Newly added template purchasing\_form total LoC: **36**

Modified shopping\_form.html LoC added: **4**, Total LoC: **53**

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Week 4

Summary

For this week, I was able to accomplish what I had myself working on for my fifth milestone goal. This week i was able to give the user the ability to add and remove a game to and from their designated cart. First an 'Add to Cart' button was added for the game that we had on display for the home page. Once that 'Add to Cart' button is clicked on by the user, the user will then be routed to the shopping cart page and will see the game they chose to add to their cart will show up in the 'Items' area of the shopping cart. The user will also be able to remove that game from the items box in the shopping cart page with the click of the 'Remove' button. Also, the user every time they add or remove a game will be notified of their actions each time they do it.

Milestones or risks in this week

For this week I accomplished my fifth milestone goal which was to give the user the ability to add and remove a game to and from their cart. The user has an 'Add to Cart' button under the game they want to choose to add. The user while inspecting their cart can also remove games they don't want in their cart. Risks this week were homework's for other classes and taking Midterm 2 for this class.

Mainly Integrated tests:

* Tested the 'Add to Cart' button so that it can route you to the shopping cart page
* Tested the ability to add a game to the items box area inside of the shopping cart page
* Tested to see if the user would get notified of this action of items being added
* Tested to see if the user can remove a game from inside the items box area of the shopping cart page
* Tested to see if the user would also get notified of this action of an item being removed from the cart.

Design

For the design i added a new from in forms.py called AddToCartForm which extends FlaskForm. All this form has is a field for 'Add to Cart' which will be used as the button for adding a game to the cart from the homepage. Modified home.html template to display the 'Add to Cart' button underneath the game in the homepage. Added a new route function called remove\_from\_cart() which provides the functionality of removing games from the cart. Modified shopping\_cart.html template which puts on display the games that were added to the cart and modified shopping\_cart() function inside of routes which will provide the communication of the games being displayed inside the items box area of the cart.

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Code

Added a new form in forms.py called AddToCartFrom

Modified shopping\_cart() function in routes.py to provided some functionality of games that were added to the cart to be displayed.

Modified shopping\_cart.html template to display games that were added to the cart and if there are no games at all in the cart a message is displayed saying that there are no items in the cart.

Modified home.html template for the 'Add to Cart' button.

Added a new routes function called remove\_from\_cart() which provides the functionality of removing a game from the items box area in the shopping cart page.

LoC added in forms.py: **2**, Total LoC: **65**

LoC added in routes.py: **18**, Total LoC: **189**

Modified shopping\_form.html LoC added: **15**, Total LoC: **68**

Modified home.html LoC added: **7**, Total LoC: **13**

**GitHub Link**: [https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/RikiLinks to an external site.](https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping_system/Riki)

Week 5

Summary

For this week, I was able to accomplish what I had myself working on for my sixth milestone goal.  The goal for this week was to be able to add and remove multiple games at the same time since now i can add and remove just one game and finish up my shopping system feature. "Fine tuning". I also made it to where once a user completes a submission the cart will completely empty out. I also added a 'Total Price' Field which will update every time a user either adds or removes a game. It is displayed underneath the 'items' box area which holds each game.

Milestones or risks in this week

For this week I accomplished my sixth milestone goal which was to be able to add and remove multiple games at the same time since now i can add and remove just one game and finish up my shopping system feature. "Fine tuning". Risks this week homework for other classes and homework 4 for SE and a Math Exam.

Mainly Integrated tests:

* Tested the 'Add to Cart' button so that it can route you to the shopping cart page
* Tested the ability to add multiple games to the items box area inside of the shopping cart page
* Tested to see if the user can remove multiple games from inside the items box area of the shopping cart page
* Tested to see the 'Total Price' field would display underneath the items box area inside of the shopping cart page
* Tested to see if the 'Cart' would empty completely upon submission from the user.

Design

For the design i modified home.html template to display the 'Add to Cart' button underneath the game in the homepage. Modified the route function called remove\_from\_cart() which provides the functionality of removing games from the cart. Modified shopping\_cart.html template which puts on display the games that were added to the cart and modified shopping\_cart() function inside of routes which will provide the communication of the games being displayed inside the items box area of the cart. purchasing() was also modified so that when the submission ends the cart will empty out.

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki

Code

Modified shopping\_cart() function in routes.py to provided some functionality of games that were added to the cart to be displayed.

Modified home.html template for the 'Add to Cart'.

Modified remove\_from\_cart() in routes.py

LoC added in routes.py: **22**, Total LoC: **202**

Modified shopping\_form.html Total LoC: **70**

Modified home.html Total LoC: **23**

Modified purchasing\_form.html Total LoC: **30**

**GitHub Link**: https://github.com/FlamingPyro/CSC440-Wiki-Gaming/tree/shopping\_system/Riki