Sprint 1 Retrospective Document

Team 24

Sakshi Choudhary, Giorgi Khmaladze, Nick Litman, Mihir Somani, Conley Utz

What went well?

In general, we successfully implemented the core of the project and integrated all the technologies to efficiently communicate with each other. We implemented database tables for users and item listings, email verification for new accounts, responsive frontend that works on all types of devices, and an easy to use/understand UI.

User Story #1

As a user, I would like to be able to register for a MyUniMarket account with a purdue.edu email

#	Description	Estimated Time	Owner
1	Create UI panel to display the fields for registration	2 Hrs	Conley
2	Retrieve field values using GET/POST via PHP	1 Hrs	Mihir
3	Validate submitted fields	2 Hrs	Nick
4	Store the validated fields in User table in MySQL database	2 Hrs	Mihir
5	Send confirmation to the given purdue.edu email	4 Hrs (each)	Mihir Nick
6	Test that validation code is successfully delivered to the user's email address.	2 Hrs	Mihir

Completed:

New users can sign-up for the website only if they have a purdue.edu account. They can choose their username and password. New users are unable to log in unless they validate their email address. If the registration is successful, the new user details are stored under the users table in the database.

User Story #2

As a user, I would like to be able to login to my MyUniMarket account

#	Description	Estimated Time	Owner
1	Create UI panel to display the fields for login	2 Hrs	Conley
2	Notify user if they have not completed a required field	2 Hrs	Giorgi
3	Validate login information	1 Hr	Giorgi
4	If correct, redirect to market page. Else, notify user that information is incorrect.	3 Hrs	Giorgi
5	Testing of session variables and redirection	2 Hrs	Mihir

Completed:

The users are able to log into their account if they provide the correct email and password that is associated with the user. After logging in they are redirected to the main market page

User Story #3

As a user, I would like to be able to sign out of my MyUniMarket account

#	Description	Estimated Time	Owner
1	Create UI interface for sign out on each page	2 Hrs	Conley
2	Change Session variables of PHP	4 Hrs	Mihir
3	Display Logout SUCCESS message	2 Hrs	Sakshi
4	Test correct modification of session variables	4 Hrs	Mihir

Completed:

The users are able to sign out of their account from any page associated with the website (excluding the 404 page.) Signing out destroys the session variables, meaning that the user needs to log back in to view their account again.

User Story #4

As a user, I would like to be able to edit my account information

#	Description	Estimated Time	Owner
1	Create UI panel to display the fields for account editing	2 Hrs	Nick
2	Create UI panel which asks user to confirm all their edits	1 Hrs	Nick
3	Edit user details from completed fields then update the database	2 Hrs	Nick
4	Functional and Usability Testing	3 Hrs	Nick

Completed:

The users are able to edit their account details, namely the username and password. During password change the users must verify the change by entering the current password. The new password is also stored as a hash under the users table in the database.

User Story #5

As a user, I would like to be able to delete my account

#	Description	Estimated Time	Owner
1	Create UI panel to display delete button	1 Hr	Conley
2	Create UI panel which asks user to confirm whether they wish to delete their account	2 Hrs	Sakshi
3	Create UI panel for user to enter their password to confirm account deletion	3 Hrs	Sakshi
4	User is automatically signed out of their account	2 Hrs	Nick
5	Remove user's account from database	2 Hrs	Giorgi
6	Testing of database tables and validation	2 Hrs	Nick

Completed:

The users can delete their account if they wish to. After confirming their current username and password, the user is logged out from their account, redirected to the sign in page, and removed from the users table of the database.

User Story #6

As a seller, I would like to list items for sale

#	Description	Estimated Time	Owner
1	Create button to redirect to new listing page.	1 Hr	Conley
2	Create UI panel to display the fields for creation of a listing.	3 Hrs (each)	Conley Nick
3	Retrieve field values using GET/POST via PHP	2 Hrs	Mihir
4	Validate submitted fields	2 Hrs	Nick
5	Store the validated fields inside new entries in Item table in MySQL database	2 Hrs	Nick
6	Usability Testing of Listing page	2 Hrs	Nick

Completed

After pressing the “Create Post” button, the user is brought to the Create Post page. The user has the ability to create a post which contains all necessary information about their item such as the name, price, quality, category, a description, and the location of the user. This information is validated and stored in a MySQL database. We have ensured that the user must input valid data for all fields in order to successfully make a post.

User Story #7

As a seller, I would like to provide the quality of my item for sale

#	Description	Estimated Time	Owner
1	Create UI panel	1 Hr	Conley
2	Add radio buttons with qualities 1 to 5 (Used - Poor, Used - Good, Used - Great, Used - Like New, New)	2 Hrs	Conley
3	Create “quality” class and create string array with the indices linking to **2**	3 Hrs	Sakshi
4	Link quality with listing on submission of post by adding to the database	4 Hrs	Mihir

Completed

When creating a post, the user is able to select the quality of their item via a series of radio buttons. For the purposes of this sprint, we represent quality as an integer in both the front and backend, but this will be updated later with more descriptive titles for the various qualities. This information is successfully represented in the database as a number.

User Story #8

As a seller, I would like to set an asking price for my item for sale

#	Description	Estimated Time	Owner
1	Create UI panel to display asking price field	2 Hrs	Conley
2	Validate that asking price is in correct format	2 Hrs	Giorgi
3	Store asking price in database	1 Hrs	Giorgi

Completed

When creating a post, the user is able to type an asking price for their item. The user is only able to type numbers and the "." key in order to ensure that the price is represented as a number and the database does not receive invalid input.

User Story #9

As a seller, I would like to write a description of my item for sale

#	Description	Estimated Time	Owner
1	Create UI panel to display description field	2 Hrs	Conley
2	Store description in database	2 Hrs	Sakshi
3	Create user from completed fields then store in database	3 Hrs	Sakshi

Completed

When creating a post, the user is able to type a description for their item. We have implemented a 200 character limit on the description to ensure valid input that the database can handle.

User Story #10

As a seller, I would like to categorize my item for sale

#	Description	Estimated Time	Owner
1	Create category class with string array for each category	3 Hrs	Sakshi
2	Create UI panel with dropdown to display various categories that can be selected from **1**	2 Hrs	Giorgi
3	Store selected categories in database on submission	2 Hrs	Giorgi
4	Validate that the category has been stored correctly.	1 Hr	Giorgi

Completed

While creating a new listing the user can attach a specific category to their item. The category is stored in the database under items table. The user is able to select a category via a drop down menu. For the purposes of this sprint, the menu contains placeholder values which will be changed later.

User Story #11

As a seller, I would like to display my location on my posts.

#	Description	Estimated Time	Owner
1	Create UI panel to display the field for location	2 Hrs	Conley
2	Store the location as a string in database	2 Hrs	Giorgi
3	Validate that the location has been stored correctly.	1 Hr	Giorgi

Completed:

While creating a new listing, the user is able to attach a specific location to their item. The location is stored in the database in the items table. To ensure valid input, the field has a limit of 100 characters.

User Story #12

As a user, I would like to easily navigate the website

#	Description	Estimated Time	Owner
1	Find a marketplace design that fits our needs	2 Hrs	Conley
2	Customize the design the ensure a simple, yet thorough experience	4 Hrs (each)	Conley Sakshi
3	Ensure bootstrap functionality is attained	2 Hrs	Conley
4	Create routing functionality for pages	2 Hrs	Conley
5	Interface Testing	2 Hrs	Sakshi

Completed

Our website is able to be viewed from any type of device while maintaining responsiveness and providing a convenient user experience.

User Story #13

As a user, I would like to avoid seeing inappropriate words/phrases on the marketplace

#	Description	Estimated Time	Owner
1	Create list of words and phrases considered ‘inappropriate’	2 Hrs	Giorgi
2	Add list of inappropriate words and phrases to a database	2 Hrs	Giorgi
3	Reject any post which contains any inappropriate word/phrase	2 Hrs	Giorgi

Completed:

All of the text input fields that are public on the website have a filter for inappropriate words. This includes usernames, item names, item descriptions, and location for items. Using javascript, we created a function that checks the fields for inappropriate words. The user is notified if a match is found and the associated form submission process is cancelled.

User Story #14

As a developer, I would like to store passwords as hashes in the MySQL Database

#	Description	Estimated Time	Owner
1	Utilize Hashing library of MySQL	3 Hrs (each)	Mihir Sakshi
2	Store Hashed Password in Database	2 Hrs	Mihir

Completed:

The passwords of all accounts are stored as a hash in the database. Instead of MySQL library for hashing we switched it with a PHP function that fundamentally does the same. We made this decision because it integrated with the code and worked well.

What did not go well?

In general, some of the user stories were challenging. For example, email verification for user story 1 could not be completed within the deadline we set for ourselves. Even though some of the user stories were overdue in terms of our group's deadlines, we still managed to complete all of them before the end of the sprint.

How should we improve?

The main challenge for this sprint was getting familiar with the technologies and integrating them together. Since that specific phase is over and we have successfully implemented the core of the project, this should not pose a problem any further. The major field that we need to improve in our upcoming sprints is setting and following our own deadlines for completing certain tasks. Doing so will ensure much smoother progress on the project and should allow us time to fix any unexpected errors that arise from implementing new features as opposed to scrambling near the end of the sprint to make last minute changes.

Another problem that we encountered was an inefficient developing process. During the first two weeks we struggled with implementing some features because they were dependent on other features to be implemented first. In order to avoid these types of issues in the future, we need to communicate amongst ourselves more effectively to remove development bottlenecks. In summary, we would like to prioritize functionality implementation correctly so we do not have this type of a problem next time.