

Kartz

by

Christopher Meyer, Nicholas Ruth, Abele Abdi, Prayag Pandit, and Alex Mezcua

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

© Copyright 2022 Abdi, Meyer, Mezcua, Pandit, Ruth

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

 <small>Christopher Meyer (Apr 24, 2022 14:13 EDT)</small>	4/22/2022
Chris Meyer	Date
 <small>Alex Mezcua (Apr 24, 2022 14:27 EDT)</small>	4/22/2022
Alex Mezcua	Date
 <small>Nicholas Ruth (Apr 24, 2022 11:55 EDT)</small>	4/22/2022
Nick Ruth	Date
 <small>Abele Abdi (Apr 24, 2022 11:15 EDT)</small>	4/24/2022
Abele Abdi	Date
 <small>Prayag Pandit (Apr 24, 2022 14:21 EDT)</small>	4/24/2022
Prayag Pandit	Date
 Rebekah Michael, Faculty Advisor	4/22/2022 Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

Table of Contents	2
Table of Figures	3
Abstract	4
Introduction	5
Project Summary	5
Problem Statement:	5
Solution:	6
Project Source:	6
Discussion	7
Project Objectives/Goals	7
Project Scope:	7
Project Timeline:	7
Technologies Used:	9
Technical Architecture Diagram:	9
User Personas:	10
Use Cases:	12
Use Case Diagram:	14
Testing Plan:	15
Overview	15
Methodology.....	15
Scope.....	15
Objectives.....	15
User Acceptance Testing	16
Testing Review	23
Change Management Plan:	23
Budget:	23
Problems Encountered and Analysis of Problems Solved:	24
Conclusion	26
References	27

Table of Figures

Project Summary:	5
Figure 1.0: Project Timeline	7
Figure 2.0 : Technical Architecture Diagram	10
Figure 3.0: User Personas	10
Educated Consumer	10
Professional	11
Teachers	11
Students	12
Organization	13
Navigation	13
Item Modification	13
Search-ability	14
Figure 4.0: Use Case Diagram	15
Figure 6: User Acceptance Testing	16
Test #1	16
Test #2	18
Test #3	19
Test #4	21
Figure 6.0: Budget	23

Abstract

Due to changes to the technology and the ability to have most items delivered to their houses or have the items ready for pickup shortly after purchasing, millions of people take advantage of online shopping services. The problem with having so many online marketplaces is that the end user does not have the convenience of organizing their items. The Kartz application helps the end users by creating a place where they can keep track of the items. The application allows the user to add, edit, and delete items in the Items list. Kartz impacts the online shopping industry by making the experience seamless and organized.

Introduction

Due to the advancement of science and technology, people have been changing their shopping practices over the past decade and a half (Digital Silk, 2021). “The attraction of online shopping is significantly growing because people can utilize their valuable time for their family and personal care. People can also save their travel time as well as the time to decide on selecting the preferred products while staying at home (Digital Silk, 2021)”. In this scenario, Kartz is created to allow users to gather the items that they want to purchase and keep track of the history of items that they will be purchased from. Kartz will encourage a mother, teacher, student, business professional, or any regular online shoppers who have passion as well as a habit of procuring items from various websites and would like to organize in a specific way. On the one hand, for fast and interactive user interfaces, the react app is used as frontend within a script library. On the other hand, flask app is contributing as a backend application framework for scaling up the application’s complexity. Likewise, Microsoft Azure is utilized in the web application to host an SQL database and the database is applied to hold user profile data as well as the user’s cart items. According to Digital Silk, “In 2021, one of the challenges of online shopping is shopping cart abandonment. Digital Silk states that online shoppers abandon their shopping carts 68% of the time and some stores can see the rates as high as 80% (Digital Silk, 2021)”. So, the challenges are not only how to cut down on cart abandonment but also keeping track of each purchase made through the application.

Project Summary:

Kartz allows users to gather the items that they want to buy and keeps track of where the items can be purchased. It is intended to be designed in a way that makes the online shopping experience more organized. Kartz is a web application that makes the users able to access their list of online items. The users have their own account that is secured in a database in the cloud. Azure is used to store all our user data and is hosted with on premise infrastructure. The site lists all the items that the user intends to purchase as well as what they have purchased. There is a login page where the user can enter their credentials, or they can make a new account. After this login page they are then be able to access their list of items. Kartz makes the online shopping experience much more organized and easier for anyone that utilizes the internet to shop daily.

Problem Statement:

According to IBISWorld, the estimated value of *e-commerce* sales for 2021 is \$742.33 billion (about \$2,300 per person in the US) dollars (Mohsin, 2021). According to an Oberlo article by Maryam Mohsin, “In 2021, an estimated 2.14 billion people worldwide are expected to purchase goods online (Mohsin, 2021).” Mohsin also states, “Around half of the online shoppers (46 percent) have failed to complete a purchase online because there were too many options to choose from (Mohsin, 2021)”. Mohsin’s research indicates that there is a consumer issue with organizing and executing e-commerce sales. More specifically, there is a problem in keeping track of online carts and desired items that consumers want to purchase. If there was a way to organize and manage all online transactions on a separate site this would mitigate the number of

unfulfilled carts that have been left filled without purchasing. This would not only help the consumer with following up with the purchases they may have intended to make but also help the seller to be able to increase their revenue coming from carts that were left filled. Kartz is the key to filling that void that so many online shoppers experience by helping them track purchases, organize their items, and make the online shopping experience better.

Solution:

Kartz is a website that allows users to save products they want to buy in the future and store items that they have bought in the past. The software would assist them in keeping track of their chosen items from online shopping in an organized manner. Products can be added, updated, and removed from a user's shopping cart. The user should have no issue locating the website where the item will be purchased; Kartz enables users to swiftly locate those goods and purchase them when they are ready.

Project Source:

The group produced this because online shopping has become a big part of our lives. People who shop online frequently deal with organizational issues and often lose track of what they intend to buy. The idea seemed like a solution that could be useful in many different situations.

Discussion

Project Objectives/Goals:

The goals of this application are to give the user a way to store their items in an organized place. The team implemented functionality to allow the user to signup, login, navigate to the various browsers, select images of items, search for any items they wish, gather items of their choice, and store them in the carts.

The objectives and high-level goals of the project are listed below.

- Create updates and delete items from their items list
- Keep a history of items the user has purchased
- Redirect to the site they plan to purchase the item from
- Provide a secure place to keep their info
- Make data recoverable between devices

Project Scope:

The project scope creates a functioning application that allows users to overcome online shopping organization problems and save their selected items. From their own list, the user can add, update, and delete items from the cart. The list is also be able to track whether a user purchased an item in case they want to return or repurchase it.

Project Timeline:

The estimated timeline for the major milestones of the project is listed in the table below. The readers see the tasks, task names, duration, start date, and end date of the project so that it will be easier for the readers to know when the type of tasks commenced and ended as well as how long they took to be completed.

Figure 1.0: Project Timeline

Task	Task Name	Duration	Start Date	End Date
Establish application and development environments	Establish application	1 week	09/19/21	09/26/21
Record desired requirements/functionality	Requirements	1 week	09/19/21	09/26/21
Implement front end and backend environments	Implement flask and react environment	2 weeks	09/26/21	10/8/21
Develop and design database	Develop and design an SQL database using Microsoft Azure	1 Week	10/10/21	10/16/21

Implement list of items	Implement list of items as a flask component	1 week	10/17/21	10/23/21
Develop CRUD Operations	Develop Create, Update, and Delete functionality using flask and react.	5 weeks	10/17/21	11/15/21
Testing	Testing for final presentation	1 week	11/15/21	11/22/21(final presentation)
Review and enhance project based on feedback from final presentation and review	Review feedback	3 weeks	12/3/21	12/24/21
Implement image functionality	Implement functionality to let user select an image for the item in the list	6 weeks	12/24/21	2/1/22
Implement and design Landing Page	Implement landing page for user sign up and sign in	5 weeks	1/16/22	2/05/22
Develop User access	Implement user access and sessions	3 weeks	1/16/22	2/05/22
Develop bought item functionality	Develop functionality to enable a user to label an item as bought	1 week	1/25/22	2/2/22
Design and implement history page	Design page to house items previously in cart	1 week	2/3/22	2/10/22
Testing application for IT Expo	Testing/review	2 weeks	3/4/22	3/18/22
Finalize project	Finalize all features	2 weeks	3/19/22	4/3/22

Proofread paper	Paper maintenance	1.5 weeks	4/4/22	4/11/22
Prepare for IT expo	Expo prep	3 days	4/9/22	4/12/22

Technologies Used:

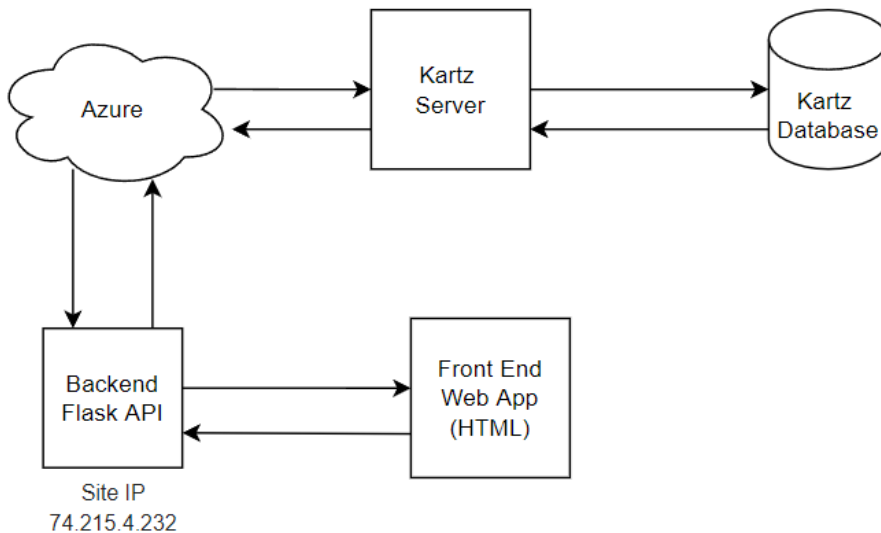
The team uses the following technologies to give the user of the application the best user experience. Flask was chosen because the framework's language, Python, is the most known to the team and is also simple to learn in a short length of time. Flask is used to administer the project's back end. ReactJS was chosen since one of our team members has prior experience with the framework and it is also a widely used and simple to use front-end framework. Yarn is a JavaScript package manager that ties together our flask server and our ReactJS server and manages the initialization of the project when it is running. Also, Azure is used to house the database and our database server. We are using SerpApi to pull images from google for the users to select to attach to their items.

Technical Architecture Diagram:

The technical architecture diagram is a visual representation that was created to walkthrough how Kartz functions, the way data flows, and label the assets.

This diagram shows how the infrastructure of the Kartz application works. It starts with the user accessing the Frontend and traverses through the backend API and then to the Azure database.

Figure 2.0 : Technical Architecture Diagram




User Personas:

Kartz looks to influence any ordinary users, professionals, students, and teachers who might be interested in visiting their favorite sites and shopping the products and brands of that nature. The common groups of users have the following personalities.


Figure 3.0: User Personas

Educated Consumer


<p>Educated Consumer: Educated shoppers would like to know most of the products' information in detail prior to purchasing. They may compare the products' price, brands, quality, and to see how popular and reliable they are on average. They may also go over reviews from other customers before deciding what to buy online.</p>	
	Educated shopper
	Teddy Field
	18
	Male
Behavior	Attentive

Pain	Spends hours of research for online shopping while comparing websites, products, and brands.
Needs & Goals	Wants to get the best deal ever, excellent quality items, and save travel time.

Professional

Professionals stress the significance of learning each type of item to design shopping services that are appealing to their staff.	
	Business Procurement Manager
	John Smith
	42
	Male
Behavior	Professional
Pain	In need of organization and inventory
Needs & Goals	Goals are to create and maintain an organization that uses the products bought from online sites/services


Teachers

Teachers would like to look for items that are required for their class. They would see the items with uniqueness and creativity.	
	Teachers
	Jane Heard Ariana Lopez Jennifer Berry Wiggins Andrew
	Age range 22- 36
	Female
Behavior	Challenging

Pain	In need of ways to organize shopping cart when they are ordering supplies for events at school
Needs & Goals	Their goal is to create a way to organize a shopping cart while buying supplies ahead of time.

Students

Students would like to visit online stores repeatedly, particularly looking for loyalty programs that would give rewards or points for purchase. They might take part in a text message marketing campaign that would qualify on available sales and specials.

	Students
	Antoine Fisher Paul Jordy David Musk Gabriel Smith Lena Jones Keisha Lopez
	Age range 16-30
	Male and Female
Behavior	Repeated shoppers
Pain	Confused due to many brands exist.
Needs & Goals	Need to focus on while choosing between the items and prefer current and future deals or promotions to support their study.

Use Cases:

The users of the Kartz application will commonly perform various tasks on the application in the following ways. The application responds as what the users requests for. For instance, if the user wants to add an item, edit an item, or delete an item, it prompts the action and behave accordingly.

Organization

The table given below has the collection of items or services that would pique the interest of the user. The average shopper might be hesitant to purchase the items, however, can transact and then manage the preferred purchases.

Use Case ID	1
Use Case Name	Organization
End Objective	Have a collection of items/services that pique interest
User/Actor	The Average Shopper
Trigger	An indecisiveness to buy an item
Frequency of Use	Often
Preconditions	An item a user wishes to purchase exists
Basic Flow	User gets an item on our site to organize all potential transactions and then can manage their purchases
Alternate Flow	User puts items onto site however does not go through with purchasing or deleting items
Postconditions	Purchasing of item, deleted item, list of items exist on site for management

Navigation

The following table shows that the user is directed to the store page for the products and might find the items looking for after being stored.

Use Case ID	2
Use Case Name	Navigation
End Objective	Be directed to the store page for an item
User/Actor	Teachers
Trigger	A decision to buy an item
Frequency of Use	Often
Preconditions	An item a user wishes to purchase exists
Basic Flow	User clicks button from the list taking them to the site where they found the item
Alternate Flow	User goes to page, but does not buy the item
Postconditions	User deletes item

Item Modification

The table below shows that the business inventory manager might change on the site page and click on the buttons for update or any desired and required changes.

Use Case ID	3
-------------	---

Use Case Name	Item modification
End Objective	User changes item attribute
User/Actor	Business inventory manager
Trigger	Item changes on site page and the user wants to be accurate
Frequency of Use	Often
Preconditions	An item a user wishes to purchase exists
Basic Flow	User clicks button to edit the details of the item and changes an attribute
Alternate Flow	User does not save changes to the item
Postconditions	The item is updated based on the user input

Search-ability

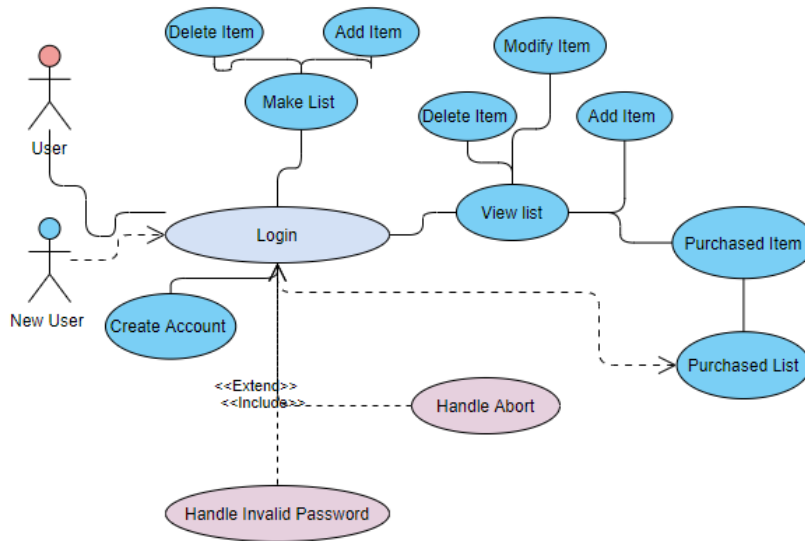
The following table indicates that the students are searching for the products of their need but seem a bit confused due to the numerous brands that exist. Therefore, they are seriously considering developing future shopping strategies.

Use Case ID	4
Use Case Name	Search-ability
End Objective	Choose for the desired products to pile up in the cart
User/Actor	Students
Trigger	A decision to buy an item
Frequency of Use	Often
Preconditions	An item a user wishes to purchase exists
Basic Flow	User clicks button from the list taking them to the site where they found the item
Alternate Flow	User goes to page, but does not buy the item
Postconditions	User deletes item

Use Case Diagram:

The application is organized as the following diagram depicts. The diagram shows the entry point for the user. The established user can login and then follow in the steps of her/his preference, whereas the new user is prompted to create the account before proceeding to the processes. The user is able to utilize the application with these functionalities given below. These use case diagrams provide the process of how a typical shopper makes purchase online. For instance, the process includes Make List, Add Item, purchased Items, Modify Item, View Item, and so forth.

Figure 4.0: Use Case Diagram



Testing Plan:

Overview

The team is using our weekly Sunday meetings to physically test the application on all team members' machines. We will also run unit tests for all essential functions such as the database connection and the export of data. We have also conducted user acceptance testing to get feedback and to find issues that we have not found yet.

Methodology

We have been testing the application physically on our devices. The team has tests to test the list creation function and the ability to test the functionality to delete an item. We plan to test user input and the backend functionality for all use cases.

Scope

The team has to test the connection to the database, retrieving items from the database, editing items in the database, deleting items from the database, and user validation.

Objectives

- All major features and use cases need to be accounted for
- All bugs must be recorded in the GitHub repository
- All bugs need to be resolved before IT Expo
- Bugs and errors should be resolved within 1 week of discovery

User Acceptance Testing

In our user acceptance testing, we wanted to see how people reacted to using the application. We also used it as a chance to test most of our functionality to identify bugs and check to see if the existing functionality worked correctly while deployed. The test cases walk the user through the essential functionality of the application and then ask the user some questions to get feedback on the application.

Figure 6: User Acceptance Testing

Test #1

Name: Luke Meyer

Age: 16

Use Case: Student

Test Case	Expected Output	Actual Output	Pass/Fail
Sign Up (fill all fields)	Successful login	Successful lo	pass
Sign Up (don't fill all fields)	throws error	Does not throw errors	Pass
Log in to the account you created	Taken to home page	Successful login	pass
Enter random information into the login page	Error thrown	Does not throw error	Pass
Add Item (fill in all fields)	Item shows up in list	Successful	pass
Add Item (leave price and or name blank)	Alert is thrown	Alert is thrown	pass
Edit an item	Successfully updated in list	Success	pass
Edit an item (leave the name or the price blank)	Throws alerts	Does not throw alert if you take price out	fail
Remove an item from list	Item will show in history page	Item will show in history page	pass
Set an item as purchased	Item will show in purchased list	Item will show in purchased list	pass

Go to the history page and delete an item	The item will be gone	The item will be gone	pass
In the history page, return an item to the list	Item shows in main list on the home page	Item shows in main list on the home page	pass
Go to purchased page and set the item as unbought	the item will return to the main list on the home list	the item will return to the main list on the home list	Pass
Search for an item by name	Only Items that contain your search will show	Only Items that contain your search will show	Pass
Go to profile page and change your password	Successfully updated	Successfully updated	Pass
Log out and log back in with new password	Successful entry	Successful entry	Pass
Go to profile page and change your email	Successfully updated	Successfully updated	Pass
Log out and log back in with new email	Successful entry	Successful entry	Pass

What do you think of the Application? He thought it was good. He thought it was good because it helps find the item instead of writing it down.

What do you think could be improved? URL validation (check to see what is on the page before going there)

Would you use this? Yes, it leads you to where you find items from many places, not just one place.

Test #2

Name: Sandra Lopez

Age: 35

Use Case: Business inventory manager

Test Case	Expected Output	Actual Output	Pass/Fail
Sign Up (fill all fields)	Successful login	Successful lo	pass
Sign Up (do not fill all fields)	throws error	Does not throw errors	Pass
Log in to the account you created	Taken to home page	Successful login	pass
Enter random information into the login page	Error thrown	Does not throw error	Pass
Add Item (fill in all fields)	Item shows up in list	Successful	pass
Add Item (leave price and or name blank)	Alert is thrown	Alert is thrown	pass
Edit an item	Successfully updated in list	Success	pass
Edit an item (leave the name or the price blank)	Throws alerts	Does not throw alert if you take price out	fail
Remove an item from list	Item will show in history page	Item will show in history page	pass
Set an item as purchased	Item will show in purchased list	Item will show in purchased list	pass
Go to the history page and delete an item	The item will be gone	The item will be gone	pass
In the history page, return an item to the list	Item shows in main list on the home page	Item shows in main list on the home page	pass

Go to purchased page and set the item as unbought	the item will return to the main list on the home list	the item will return to the main list on the home list	Pass
Search for an item by name	Only Items that contain your search will show	Only Items that contain your search will show	Pass
Go to profile page and change your password	Successfully updated	Successfully updated	Pass
Log out and log back in with new password	Successful entry	Successful entry	Pass
Go to profile page and change your email	Successfully updated	Successfully updated	Pass
Log out and log back in with new email	Successful entry	Successful entry	Pass

What does she think of the Application? She thinks the application is easy to search for items she looks for.

What do you think could be improved? Images need to be seen on the main page, Currently, having issue with the image appearance.

Would you use this? Yes, it helps with finding product items from multiple sites and locations.

Test #3

Name: Kendall Schumacher

Age: 21

Use Case: Student

Test Case	Expected Output	Actual Output	Pass/Fail
Sign Up (fill all fields)	Successful login	Successful lo	pass
Sign Up (do not fill all fields)	throws error	Does not throw errors	Pass

Log in to the account you created	Taken to home page	Successful login	pass
Enter random information into the login page	Error thrown	Does not throw error	Pass
Add Item (fill in all fields)	Item shows up in list	Successful	pass
Add Item (leave price and or name blank)	Alert is thrown	Alert is thrown	pass
Edit an item	Successfully updated in list	Success	pass
Edit an item (leave the name or the price blank)	Throws alerts	Does not throw alert if you take price out	fail
Remove an item from list	Item will show in history page	Item will show in history page	pass
Set an item as purchased	Item will show in purchased list	Item will show in purchased list	pass
Go to the history page and delete an item	The item will be gone	The item will be gone	pass
In the history page, return an item to the list	Item shows in main list on the home page	Item shows in main list on the home page	pass
Go to purchased page and set the item as unbought	the item will return to the main list on the home list	the item will return to the main list on the home list	Pass
Search for an item by name	Only Items that contain your search will show	Only Items that contain your search will show	Pass
Go to profile page and change your password	Successfully updated	Successfully updated	Pass
Log out and log back in with new password	Successful entry	Successful entry	Pass

Go to profile page and change your email	Successfully updated	Successfully updated	Pass
Log out and log back in with new email	Successful entry	Successful entry	Pass

What do you think of the Application? She thinks the application is very well put together and really liked the image function

What do you think could be improved? Design and Color scheme she says could be better as well as maybe some spinners when something is loading like the images

Would you use this? Yes, she would in a well-polished version something like a browser extension for the application

Test #4

Name: Nahum Mulugeta

Age: 16

Use Case: Student

Test Case	Expected Output	Actual Output	Pass/Fail
Sign Up (fill all fields)	Successful login	Successful lo	Pass
Sign Up (do not fill all fields)	throws error	It does not throw an error	Pass
Log in to the account you created	Taken to home page	Taken to the home page	Pass
Enter random information into the login page	Error thrown	It does not throw an error	Pass
Add Item (fill in all fields)	Item shows up in list	Items added to the list	Pass
Add Item (leave price and or name blank)	Alert is thrown	Alert is thrown	Pass
Edit an item	Successfully updated in list	Successfully updated in the list	Pass

Edit an item (leave the name or the price blank)	Throws alerts	It does not throw errors if you take the price out	Fail
Remove an item from list	Item will show in history page	Item will show in the history page	Pass
Set an item as purchased	Item will show in purchased list	Item will show in purchased list	Pass
Go to the history page and delete an item	The item will be gone	The item will be gone	Pass
In the history page, return an item to the list	Item shows in main list on the home page	Items show in the main list on the home page	Pass
Go to purchased page and set the item as unbought	the item will return to the main list on the home list	The items will mainly list on the home page	Pass
Search for an item by name	Only Items that contain your search will show	Only Items that contain your search will show	Pass
Go to profile page and change your password	Successfully updated	Successfully updated	Pass
Log out and log back in with new password	Successful entry	Successful entry	Pass
Go to profile page and change your email	Successfully updated	Successfully updated	Pass
Log out and log back in with new email	Successful entry	Successful entry	Pass

What do you think of the Application? He thinks the application is well organized and easy to use. Also, he likes the image searching feature because it lets you find your items fast.

What do you think could be improved? He thinks the color scheme could be better because we only used three colors, but it's effectively used at the right places.

Would you use this? Yes, he says he would use this when it is more developed and becomes a mobile app.

Testing Review

The testing of this application has been a challenge. We have learned that it is easier to write unit tests before writing actual code rather than writing tests that validate the pre-written code. The procedure of doing testing in the future will change.

Change Management Plan:

Any person on the team can request a change in any circumstances. The group votes on whether to approve the change to the project. The project manager, Chris, can decide if a change is ready to be merged into the Master Branch. When a change does not work as intended or it breaks the application in any way, we will set up time and troubleshoot as a team. If no progress is made with the change in a certain amount of time, we roll back the change and revisit the intended function or feature. We mostly communicate changes to our advisor through our weekly progress reports and our biweekly meetings.

Budget:

This budget is based on how much we are spending on our software and our Work Effort. We have estimated that our IT and developing labor equaled around \$20.00 per hour as a group and work an estimated 20 hours per week on Kartz. We utilize Azure for our database. In total our budget comes to \$400.00 per week and \$1,600 per month. In total this project has been in progress for 3 months, which is \$4,800.

Our application also includes a “Google Images Search API” that charges a fixed amount per number of searches. We currently pay a \$50.00 fee every month for 1000 API searches.

Figure 6.0: Budget

	Rate Per/Hr	Work Effort (Hours)	1 X Costs
Labor – IT	\$20/hr	20/hrs per week	\$400
Labor – External	\$0.00	0.00	\$0.00
Software – External	\$.50/hr	20/hrs per week	\$50.00
Hardware – External	\$0.00	0.00	\$0.00
Misc.	\$0.00	0.00	\$130.00
TOTAL	\$410 per week		Total \$4,800

Problems Encountered and Analysis of Problems Solved:

The Team has had a rough time getting through this project. The problems started with idea changes which caused a couple of pivots. The problem that impacted the team the most was setting up our development environments. Other Problems included connecting to our database, getting our IDs to auto increment in the database, and using unit tests to test functionality.

- Idea changes
 - The team originally had an idea to use Python web-scraping to let the user copy and paste a link and the scraping would pull the data from the link and add the item to a list. Due to security implementation from online marketplaces, that idea was not manageable. The group then tried to implement an Android mobile application that would take more user input. Our group had a lot of problems with getting the environment set up, so we moved to the idea that we are working on currently.
- Environment setup
 - The team had a very hard time with setting up our development environments. We spent a couple of weeks trying to get an Android application set up. After a lot of time spent, the team decided to switch to a web application. One of the members of the team was able to complete the setup and get it all working in a short amount of time. The other members of the team had a really challenging time with getting their environments set up. When setting up the React and Flask environment, the terminal threw many errors from react not having certain node modules. The team had multiple calls to help get everyone's development environment setup. This problem would happen every time someone would clone the repository from GitHub. The package-lock.json and the package.json files were messing up the setup so we added those files to the gitignore file. The ultimate resolution was pulling and rebasing the Master branch to our local branch. The application is now easy to run.
- Database connectivity
 - Most of the team had a problem with the backend of our application not being able to connect to the Azure Database. Chris was able to implement the connection to the database on the first try, but all other members were unable to connect to the database due to a Python module called PYODBC and Wheel not being able to be installed. The resolution to this is related to certain Python versions not having those modules. For example, Chris has a python version 3.9.6 which has those modules available to download. At the beginning of the semester when everyone was installing Python, the others installed Python 3.10.0 which does not have those modules able to install.
- SQL ID auto increment
 - When the team was trying to develop the add functionality, we were unable to do it directly through the database. We resolved this by creating a function in Python to generate the next Item ID for the database. The function queries the database for the IDs in the Items table, finds the highest ID in a list, and returns

one more than the highest ID to use in the Database. In the spring semester, the team made a change to the database that made the workaround unnecessary. We made the ItemID column an Identity site that would increment IDs through the server.

- Unit Testing
 - With the way our project is set up, we had trouble getting unit tests for some of the functions that we should test. This problem has not been solved yet. Unit testing should have been more of a priority. The team has some unit tests, but we should have done more. Our project is setup in such a way that it is hard to write unit tests. The team figured out that it is hard to write unit tests for functionality already written.
- Design
 - Initially, our team had a big struggle with trying to implement our front-end style. Eventually, we were able to implement front-end style. The team spent a lot of time trying to get a design that everyone would be okay with. We think a big cause of this problem was trying to develop something outside of the application and then trying to add it into our code. We have been able to design signup and sign in pages. The first-time user has to sign-up and then sign-in to access the web application. We have utilized CSS, html, and JavaScript to implement the color effects and design. The home button, bought items, items history and search bar are vertically designed and our project, karts, user profile, and logout are designed on top bar of the web application. Finally, we have been able to implement the decent color and styles within the application.
- Image search functionality
 - After our fall semester presentation and technical review, we needed to make the application more user friendly. The first thing that came to mind was adding images to associated with the items. The team decided to implement a feature to search for images from google and let the user add a picture to the item in their list. We think this feature was the hardest feature to implement due to the complexity of how to display the images on the screen. The team spent a long time implementing this feature. The part that made it so hard to implement is figuring out how to get the images to show up when the user searches for images. The challenge of it was taking the image name from the add component and sending it to the child component and then sending the data back to the parent. Chris found a workaround for the problem. The workaround involved having the child call a function when the search images button is clicked.
- Deployment
 - Throughout the year working on the Kartz application all team members worked on the application locally. We needed to figure out how to get this application in a production environment. The group originally wanted to deploy the application on Azure where it would be secured in a cloud environment. However, we ran into many issues with failed deployments and the application not building properly. So instead, I was able to set up the backend of the web application to be hosted with a server and port forwarding. There were still some issues with

hosting using on premise infrastructure, but we smoothed out the issues and now works flawlessly.

Conclusion

The team has developed skills through the problems they faced and continued using those skills throughout the duration of the problem. We learned that setting up development environments is not always as easy as some guy on YouTube makes it look. Through the process, we learned a lot about how to troubleshoot the errors that we were given. The team also learned skills and processes through the overall development of the application. Through development, the team has also learned how to use GitHub more effectively to decrease down time when updating our local branches. The team has learned a lot more about full-stack development while implementing flask and react. In the spring, we developed unit tests for the backend. The team implemented functionality such as information security, user access, user ability to label an item as bought, and implementing a history page, search functionality across all lists, and a way to associate an image with an item. We also implemented a sign-in page and a sign-up page. In the end, the team is happy with the finished product, and the team is planning to make improvements to Kartz after graduating.

References

- IBISWORLD, *E-commerce sales*. (2021). <https://my-ibisworld-com.uc.idm.oclc.org/us/en/business-environment-profiles/f318/business-environment-profile>
- Digital Silk. (2021, October 26). *7 biggest challenges of ecommerce and how to overcome them*. Retrieved November 24, 2021, from <https://www.digitalsilk.com/challenges-of-ecommerce>.
- Mohsin, M. (2021, September 18). *10 online Shopping statistics you need to know in 2021*. Oberlo. Retrieved September 21, 2021, from <https://www.oberlo.com/blog/online-shopping-statistics>.












it Group 8 Final


Final Audit Report

2022-04-24


Created:	2022-04-24
By:	rebekah michael (bekahmarie@gmail.com)
Status:	Signed
Transaction ID:	CBJCHBCAABAA9I0LcZfhINOYPAkqzn_ORedFM70qhPD1

"it Group 8 Final" History


-  Document created by rebekah michael (bekahmarie@gmail.com)
2022-04-24 - 2:52:09 PM GMT- IP address: 50.5.35.125
-  Document emailed to Abele Abdi (abdiam@mail.uc.edu) for signature
2022-04-24 - 2:54:10 PM GMT
-  Email viewed by Abele Abdi (abdiam@mail.uc.edu)
2022-04-24 - 3:05:40 PM GMT- IP address: 104.28.103.85
-  Document e-signed by Abele Abdi (abdiam@mail.uc.edu)
Signature Date: 2022-04-24 - 3:15:42 PM GMT - Time Source: server- IP address: 74.83.212.203
-  Document emailed to Nicholas Ruth (ruthns@mail.uc.edu) for signature
2022-04-24 - 3:15:44 PM GMT
-  Email viewed by Nicholas Ruth (ruthns@mail.uc.edu)
2022-04-24 - 3:54:07 PM GMT- IP address: 104.28.77.26
-  Document e-signed by Nicholas Ruth (ruthns@mail.uc.edu)
Signature Date: 2022-04-24 - 3:55:57 PM GMT - Time Source: server- IP address: 74.215.4.232
-  Document emailed to Christopher Meyer (meyer3cj@mail.uc.edu) for signature
2022-04-24 - 3:55:59 PM GMT
-  Email viewed by Christopher Meyer (meyer3cj@mail.uc.edu)
2022-04-24 - 6:10:15 PM GMT- IP address: 98.28.201.244
-  Document e-signed by Christopher Meyer (meyer3cj@mail.uc.edu)
Signature Date: 2022-04-24 - 6:13:57 PM GMT - Time Source: server- IP address: 98.28.201.244
-  Document emailed to Prayag Pandit (panditpg@mail.uc.edu) for signature
2022-04-24 - 6:13:59 PM GMT

 Email viewed by Prayag Pandit (panditpg@mail.uc.edu)


2022-04-24 - 6:20:41 PM GMT- IP address: 74.215.167.42

 Document e-signed by Prayag Pandit (panditpg@mail.uc.edu)

Signature Date: 2022-04-24 - 6:21:53 PM GMT - Time Source: server- IP address: 74.215.167.42

 Document emailed to Alex Mezcua (mezcuaaj@mail.uc.edu) for signature

2022-04-24 - 6:21:55 PM GMT

 Email viewed by Alex Mezcua (mezcuaaj@mail.uc.edu)

2022-04-24 - 6:27:08 PM GMT- IP address: 74.139.49.94

 Document e-signed by Alex Mezcua (mezcuaaj@mail.uc.edu)

Signature Date: 2022-04-24 - 6:27:30 PM GMT - Time Source: server- IP address: 74.139.49.94

 Agreement completed.

2022-04-24 - 6:27:30 PM GMT