

Towson Marketplace

Sprint 3 Documentation

Note you will need a .env file in order to use this app. You will need:

REACT_APP_URI = mongodb database link

EMAIL_ADDRESS = address to send verification code

EMAIL_PASSWORD = google authentication password for nodemailer

JWT_SECRET = special JSON web token

CLOUD_NAME = Cloudinary name

CLOUD_KEY = Cloudinary key

CLOUD_KEY_SECRET = Cloudinary secret key

REACT_APP_AI_KEY = ChatGPT ai key

NOTE If you have issues with email verification, you can use my username and password to login

rkinde1@students.towson.edu

Password: rr

INSTRUCTIONS:

How to Use Our Website:

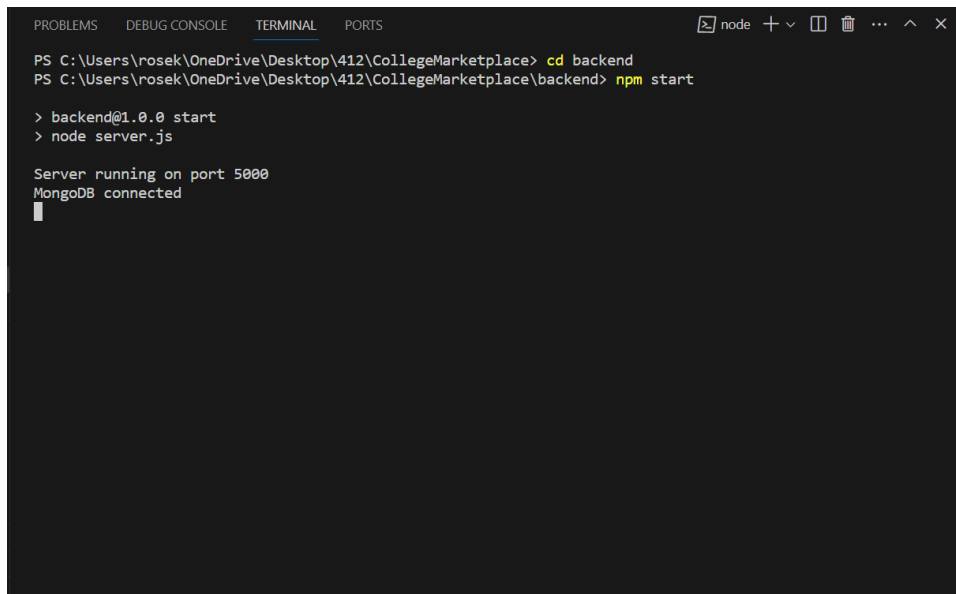
You must have Node.js to run our site. From there, you will unzip the folder and then open it up in VSCode. We have the dependencies split up in frontend and backend so you will have to run **npm i** in the terminal for both backend and frontend:

cd backend

npm i

npm start

If it is working correctly, you should see this in the terminal:



```
PROBLEMS  DEBUG CONSOLE  TERMINAL  PORTS
node + - [ ] [ ] ... ^ x

PS C:\Users\rosek\OneDrive\Desktop\412\CollegeMarketplace> cd backend
PS C:\Users\rosek\OneDrive\Desktop\412\CollegeMarketplace\backend> npm start

> backend@1.0.0 start
> node server.js

Server running on port 5000
MongoDB connected
█
```

Now you want to open another terminal and run the commands below:

cd frontend

npm i

npm start

After this the website should open automatically and you will be able to use our website (unless the .env is not attached). Below is where you should be directed and is a visual of version 1 of our homepage, this is where users will be directed to once they click on our page, and where they will be redirected to when they click the “Home” button.

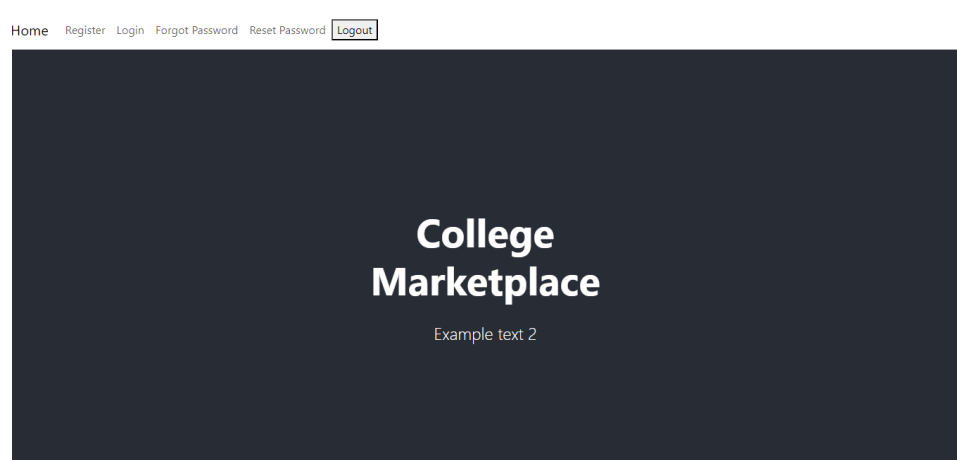


Figure 2.2 above shows our new homepage

The screenshot shows a web browser window with the address bar set to 'localhost:3000'. The page title is 'Sign Up'. The navigation bar includes links for 'Home', 'Register', 'Login', 'Forgot Password', and 'Reset Password'. The main content area is a dark blue rectangle containing a white 'Sign Up' form. The form has four input fields: 'First Name:' (containing 'rr'), 'Last Name:' (containing 'r'), 'Email:' (containing 'rkinde1@students.towson.edu'), and 'Password:' (containing '..'). A 'Sign Up' button is located below the password field. The browser's taskbar at the bottom shows the system clock as 10:34 PM on 10/16/2023.

Figure 2.3 above shows our Signup page

The screenshot shows a white error message dialog box with a thin grey border. The text inside reads 'localhost:3000 says' in bold, followed by 'User already exists' in a lighter font. A blue 'OK' button is positioned in the bottom right corner of the dialog.

Figure 2.4 shows the error message that occurs when a user inputs an email that is already saved into the database on the sign-up page (User Story)

The screenshot shows a white error message dialog box with a thin grey border. The text inside reads 'localhost:3000 says' in bold, followed by 'Please use a Towson email' in a lighter font. A blue 'OK' button is positioned in the bottom right corner of the dialog.

Figure 2.5 above shows the error message that occurs when a user inputs an email that is not towson.edu or students.towson.edu (User story, added towson.edu so you can test it out with your

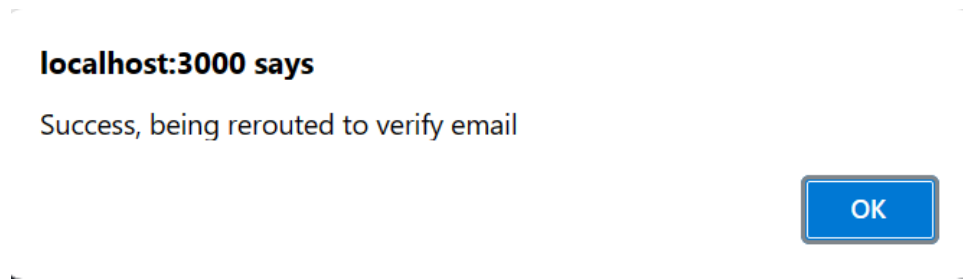


Figure 2.6 above shows what happens after successful registration.

Verify Email

After the success message, the user will receive an email like below and will be redirected to a verification page where they must enter their OTP

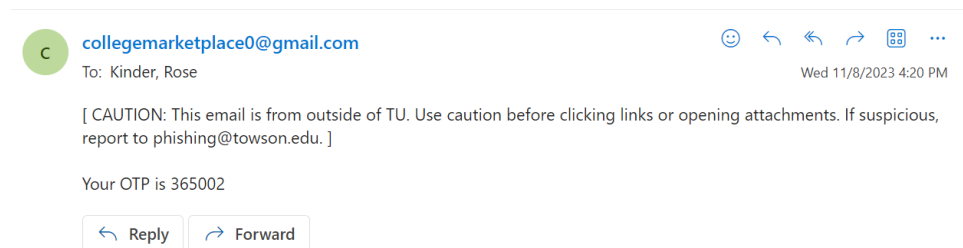


Figure 2.7

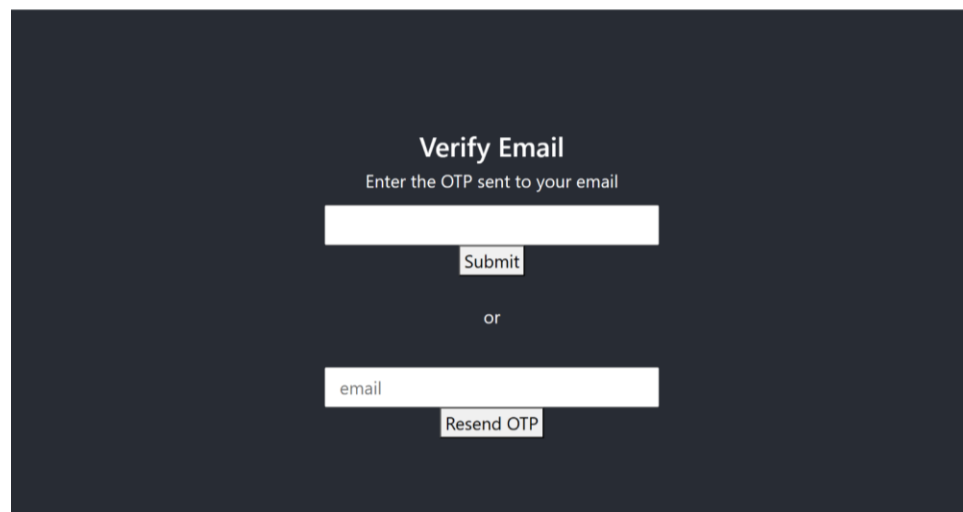
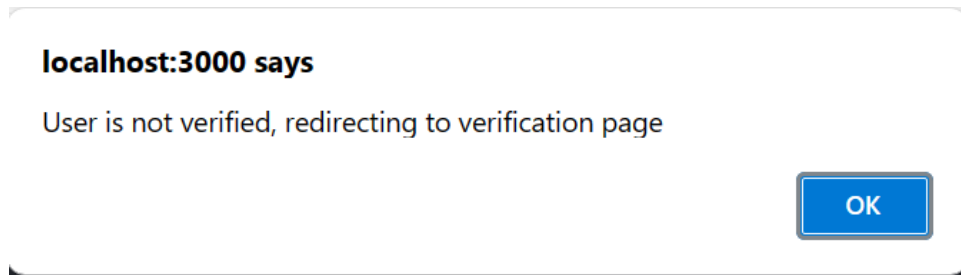


Figure 2.8

If the user attempts to login with that account and they haven't verified their email, they will be redirected back to the verify email page above and will not be able to sign in

Figure below shows error message when trying to login without being verified



The user will also be able to navigate to the verify link on the navbar in order to verify their account as well.

Figure 3.1 below shows what the login page looks like

A screenshot of a web application's login page. The top navigation bar (navbar) is light gray and contains links: "Home", "Register", "Login", "Forgot Password", "Reset Password", and "Logout". The main content area has a dark gray background. In the center, the word "Login" is displayed in white. Below it are two white input fields, one for "Email" and one for "Password". A white "Submit" button is positioned below the password field. At the bottom of the main content area, the word "or" is centered, followed by a blue link that says "Signup Page".

Figure 3.2 below shows what message the user is shown when they input the incorrect password to an email that is saved in the database

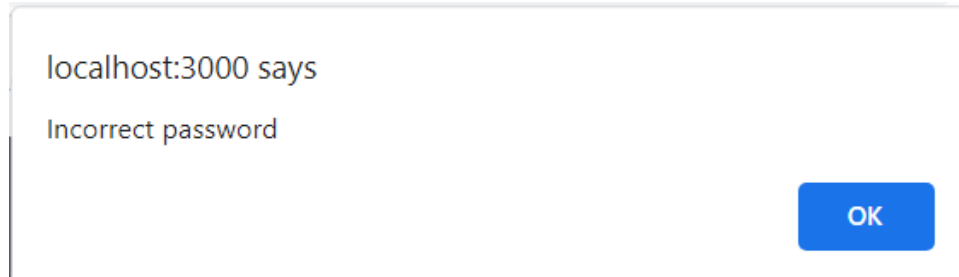


Figure 3.3 below shows the message when a user inputs the correct email and password after signing up (user story)



Figure 3.4 below shows the message the user is shown when they input an email that is not registered with the site (user story)



Forgot Password/Reset Password

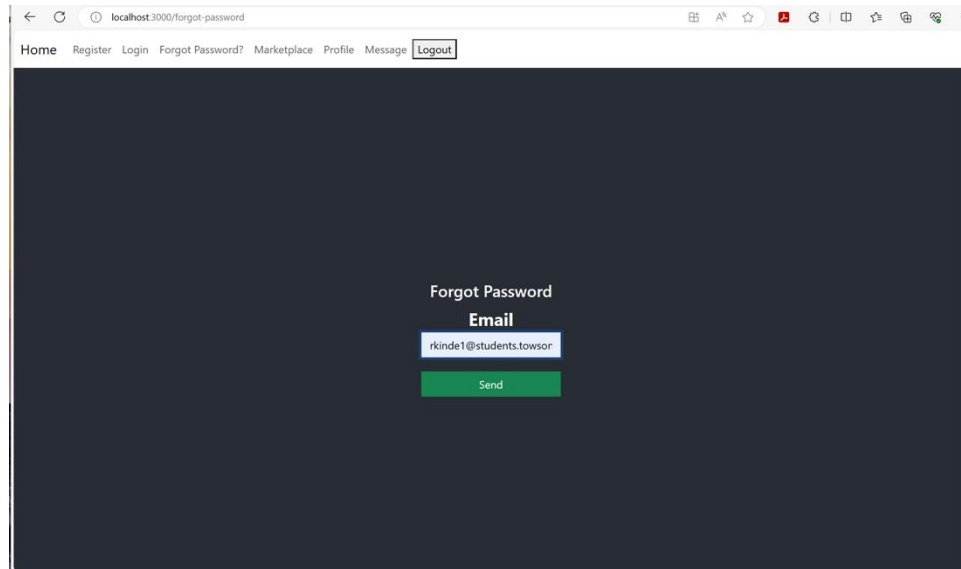
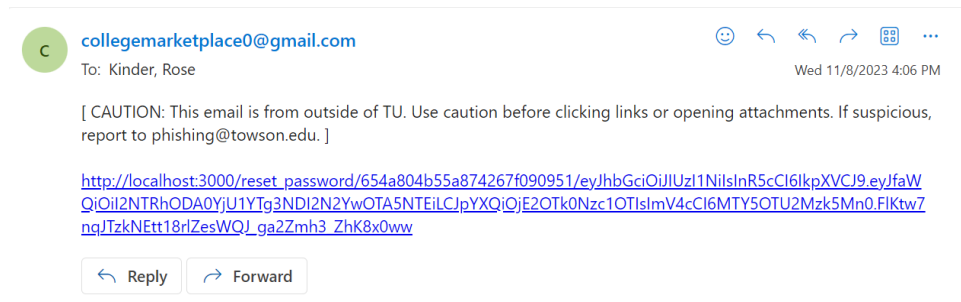


Figure 4.1 above shows what the forgotten password page looks like when you enter your email. When you click send, you will receive a notification saying “success”. Go log into that email and you will find an email with a link:



Upon clicking this link, the user will be rerouted to the reset password below where they can enter their new password. After clicking update, the user will be able to login with the new password.

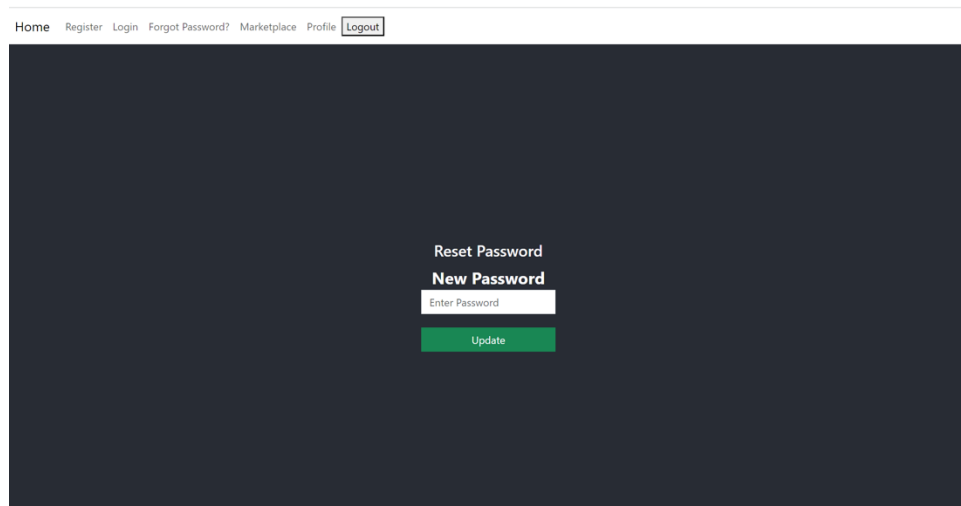


Figure 4.1 above shows what the reset password page looks like

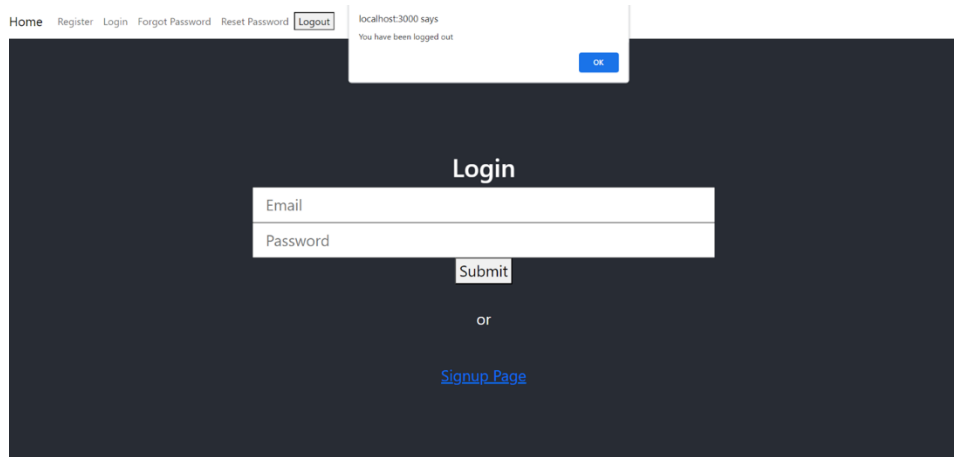


Figure 4.2 features the logout button on the menu and shows the message the user gets)

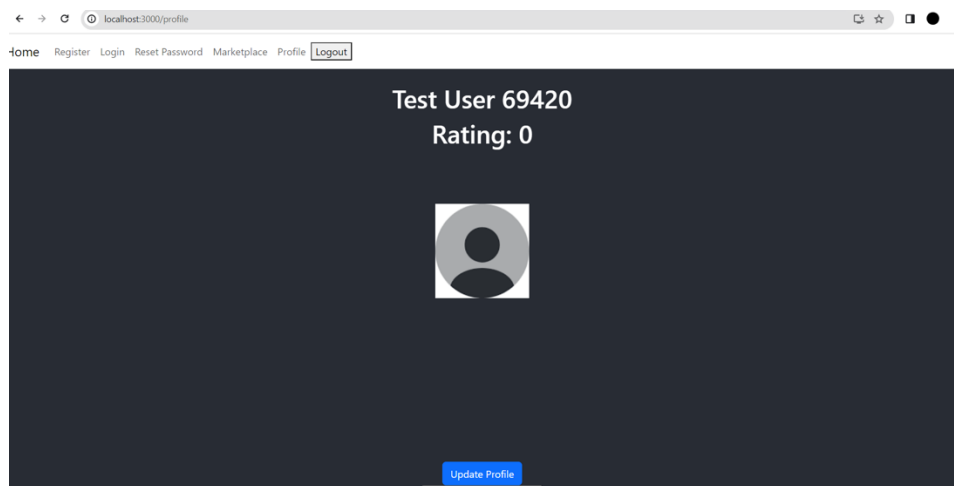


Figure 4.3 shows the default user profile for any account recently made, with no user input. The profile features the users first and last name, default image, graduation year, and overall rating which is set to 0.

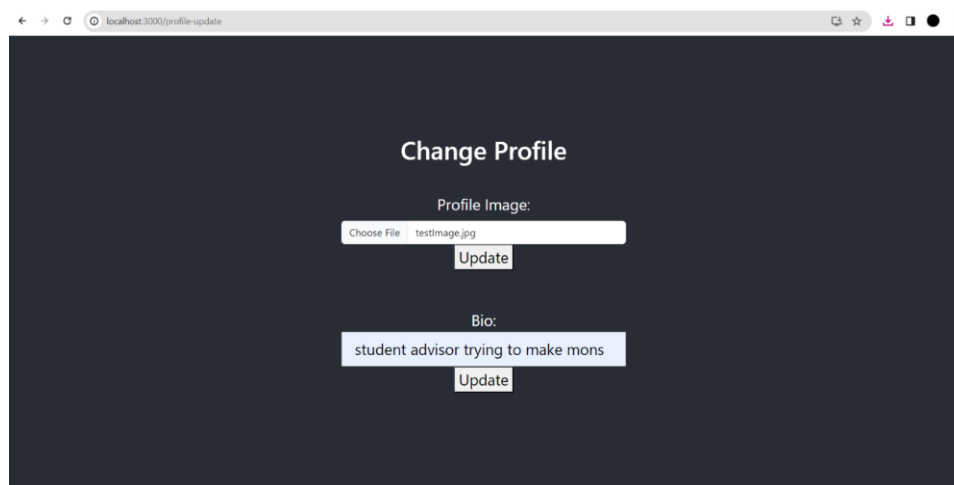


Figure 4.4 shows the updateProfile page that users can get to from the blue button on their profile. Users can upload a profile image and custom bio.

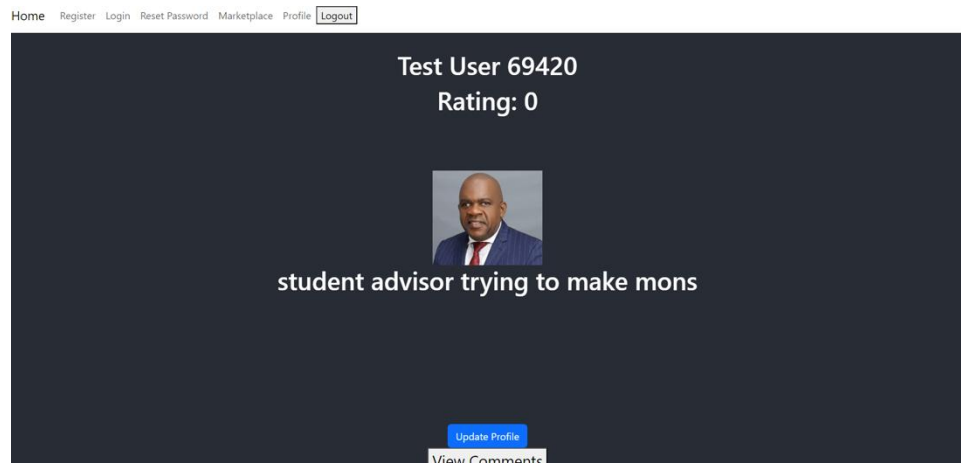


Figure 4.5 shows an updated profile with custom profile image and bio

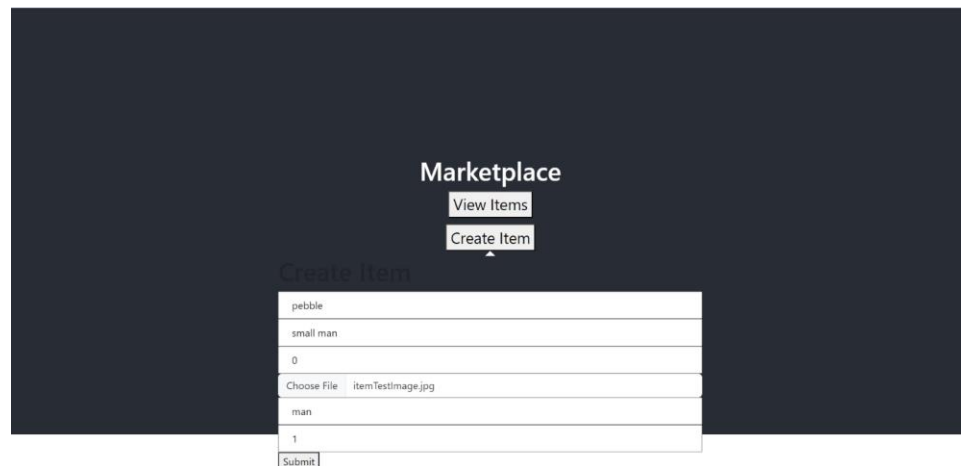


Figure 4.6 shows a new option to upload an image of an item for the marketplace.

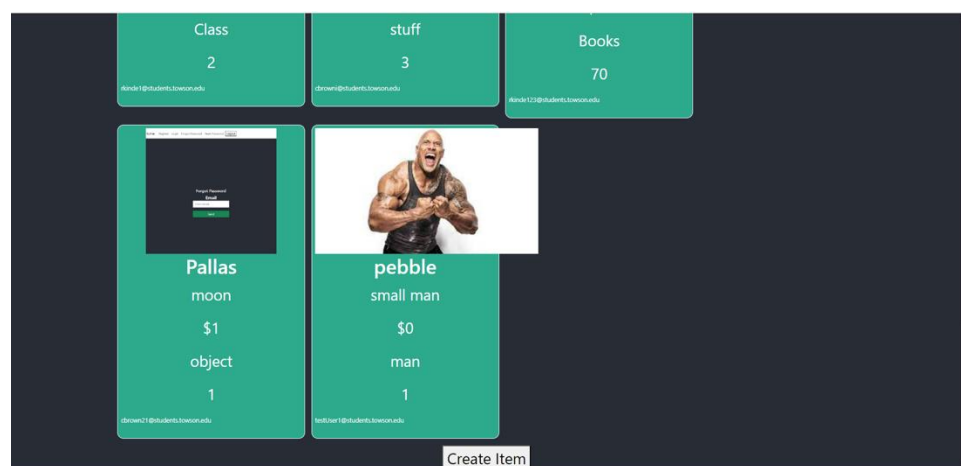


Figure 4.7 shows the images for items on the marketplace after clicking "View Items"

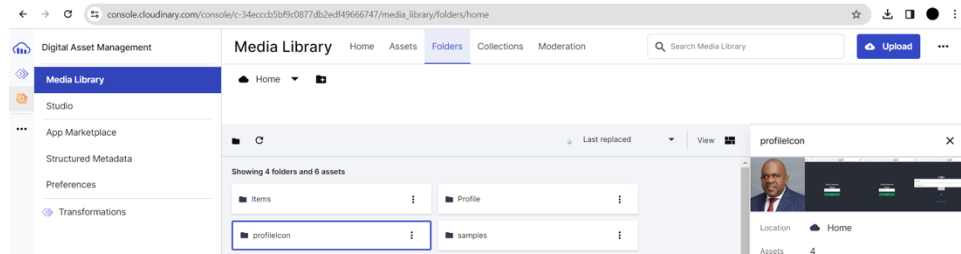


Figure 4.8 shows our cloudinary account. This is how we can save users images. Cloudinary allows us to upload a saved image, then send us a url we can save to the backend for later use.

AI Chat bot and progress bar

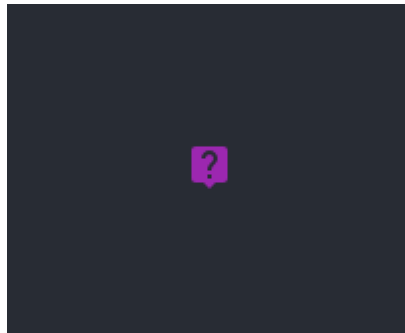


Figure 5.1 shows the support icon/button used to open the chat box. This is located on the navbar next to the logout button

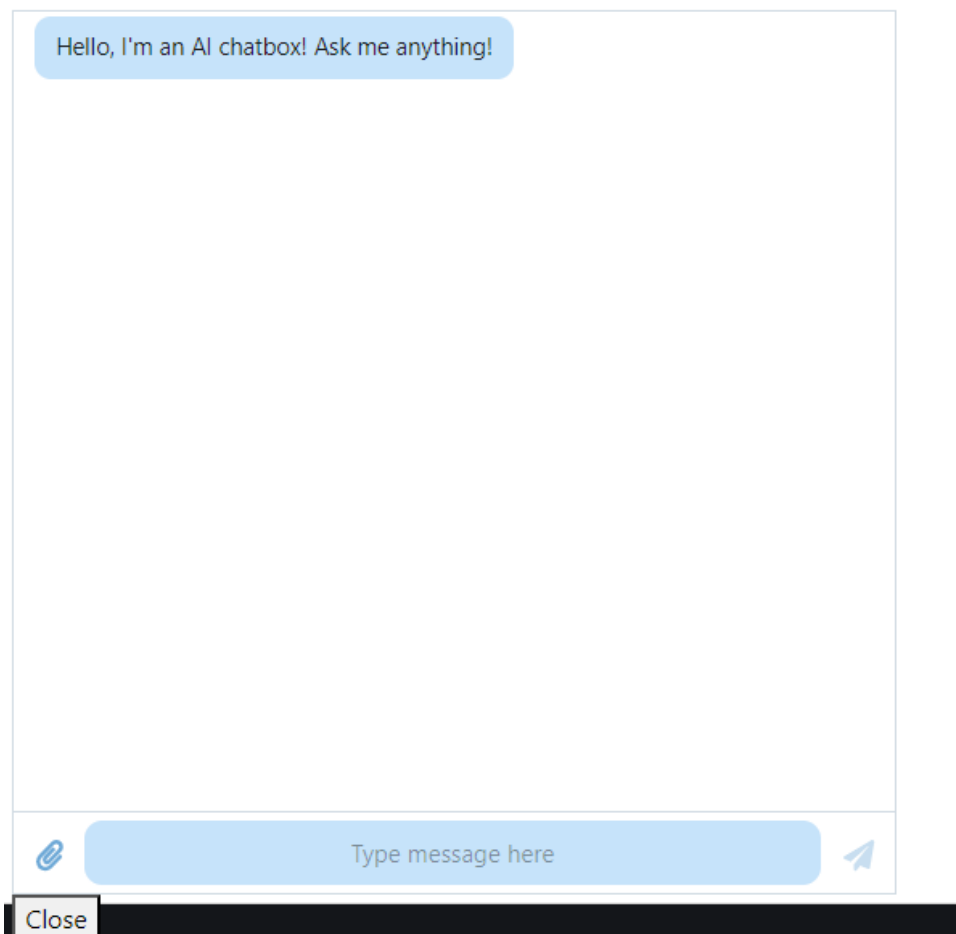
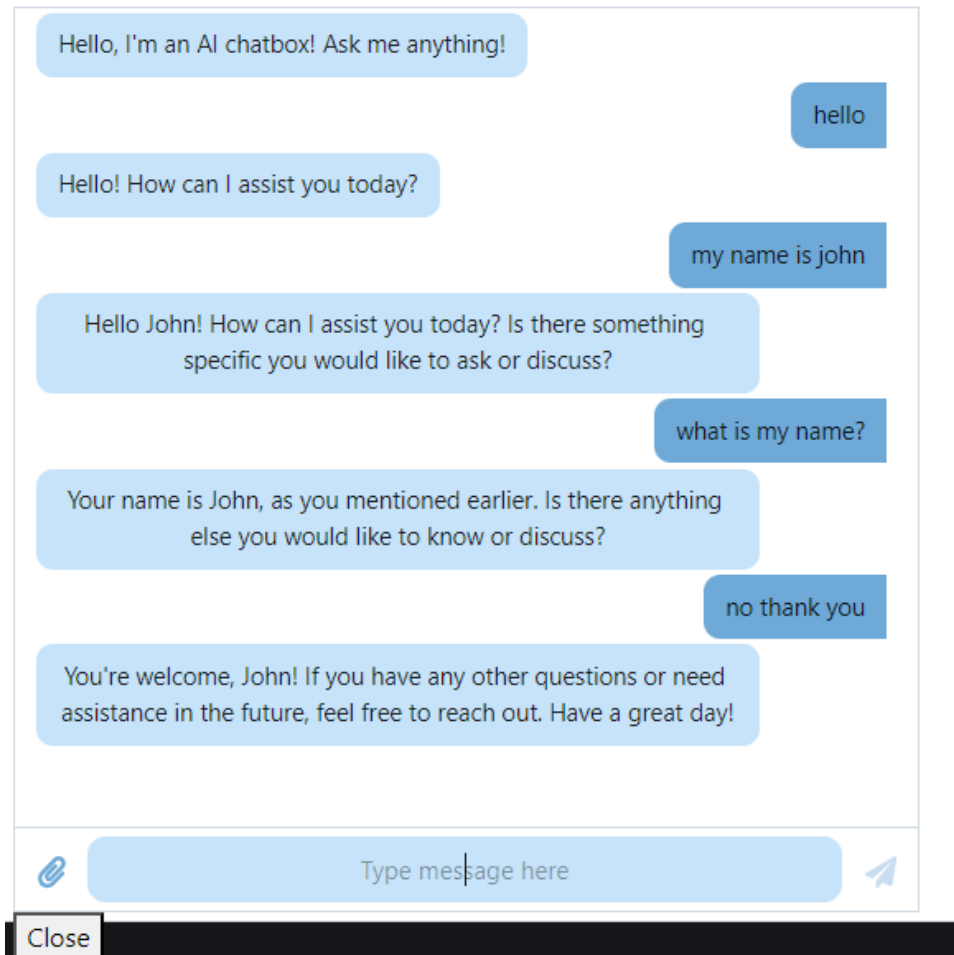


Figure 5.2 shows the chat box a user will see once the support button is pressed.



Search Bar:

On the top left of the navbar is a link to the market. Upon clicking on this, it will redirect to the image in 6.1 below.

The Search Bar feature was an important device we needed to create for the user. We chose a dynamic Search Bar and laid out the three conditions we wanted to test against for all searches. Those three conditions were: Name, Category, and Description. The Search Bar will pass the user input, on keystroke, through those 3 conditions and match any series of letters or words.

Matt B left the Search Bar button in the design, for user familiarity and possible future designs.

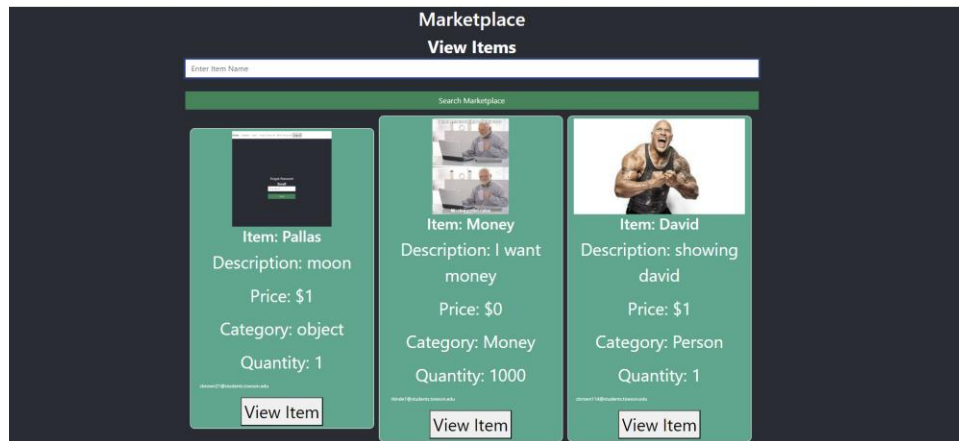


Figure 6.1

Figure 6.1 demonstrates how an empty search returns the entire list of items in the marketplace.

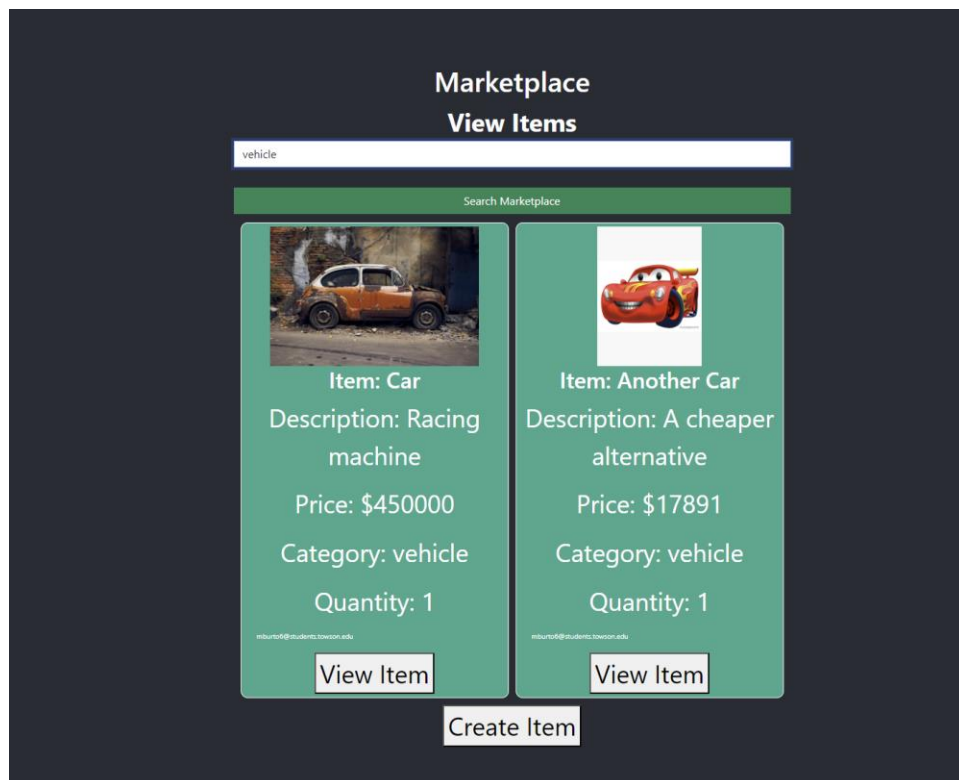


Figure 6.2

Figure 6.2 demonstrates a search where the category is matched.

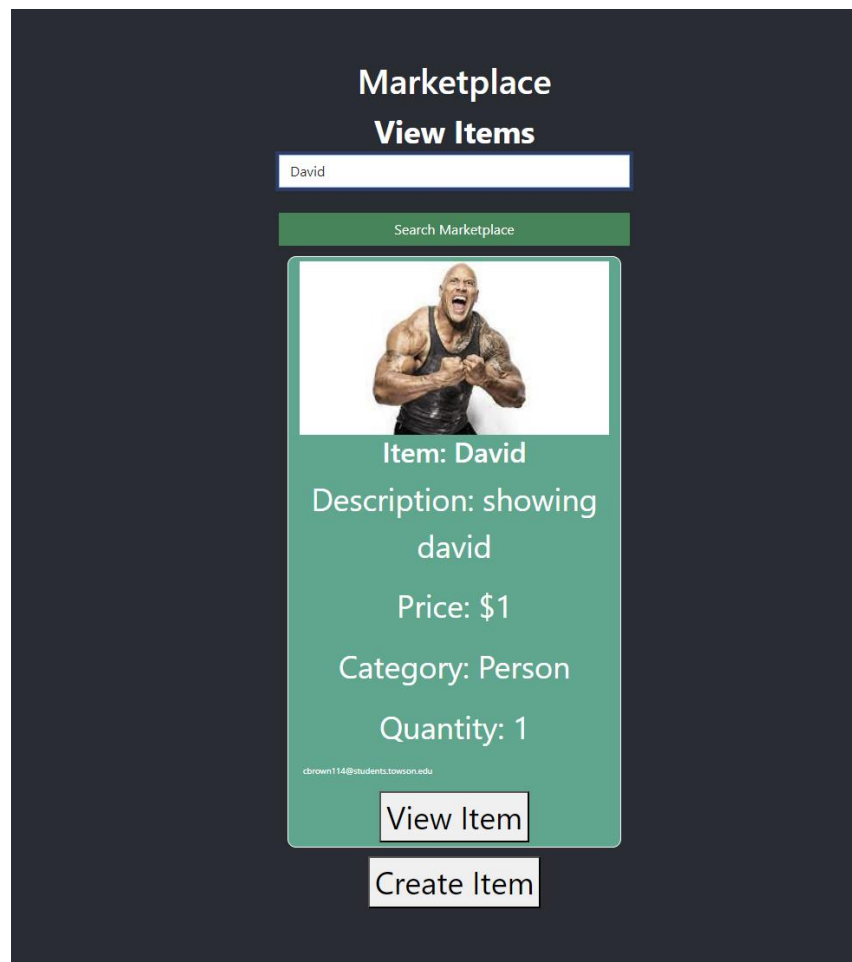


Figure 6.3

Figure 6.3 demonstrates a search where the name is matched.

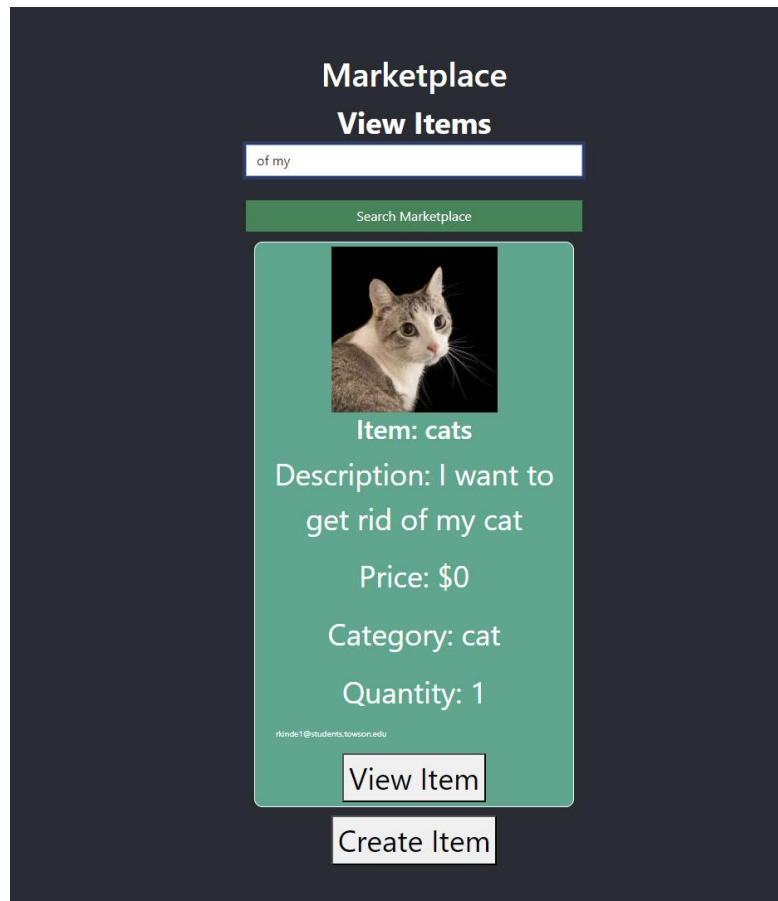


Figure 6.4

Figure 6.4 demonstrates a search where any string of letters is matched.

As the search bar was being implemented, we also did a quick pass on the formatting to make the list of items look neater.

View Single Item

Once the user clicks on view item on any item, a screen will appear like below with details of the item along with messaging the seller, initiating transaction, and viewing the seller profile by clicking the link. This is also where the poster/seller can delete their item. The red button only shows up for the seller and no one else.

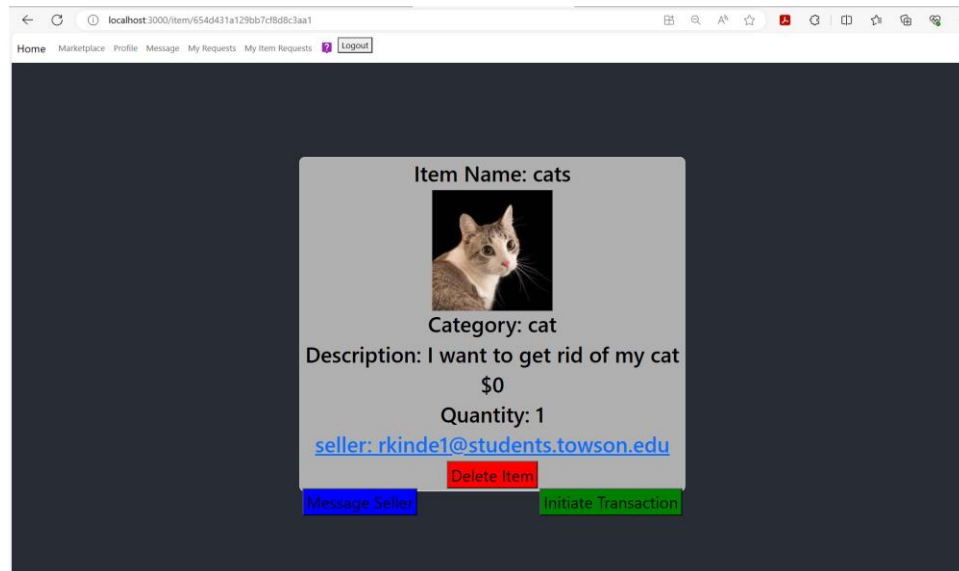


Figure 7.1

The picture below shows what an item looks like when the buyer is looking at another item that isn't their own. Notice the lack of delete item button.

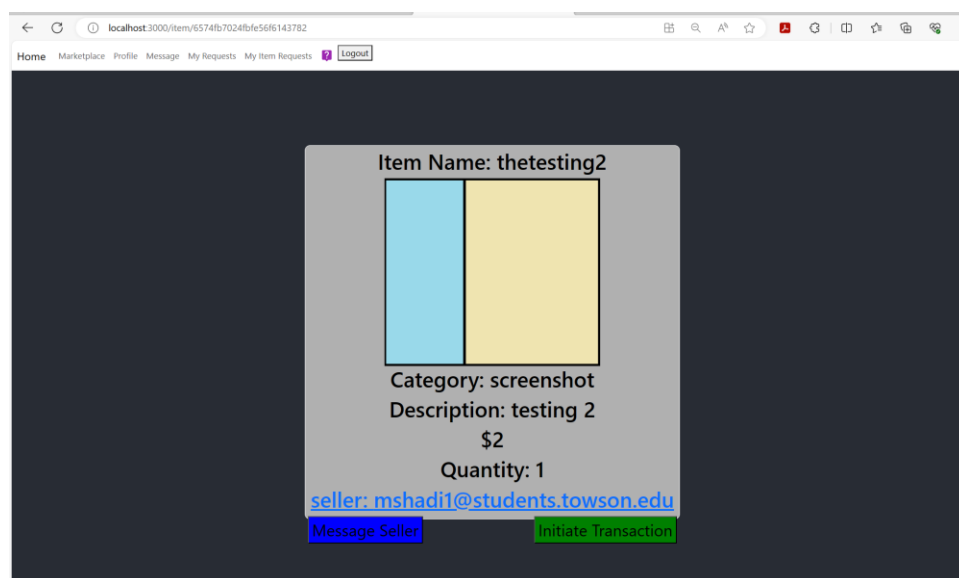
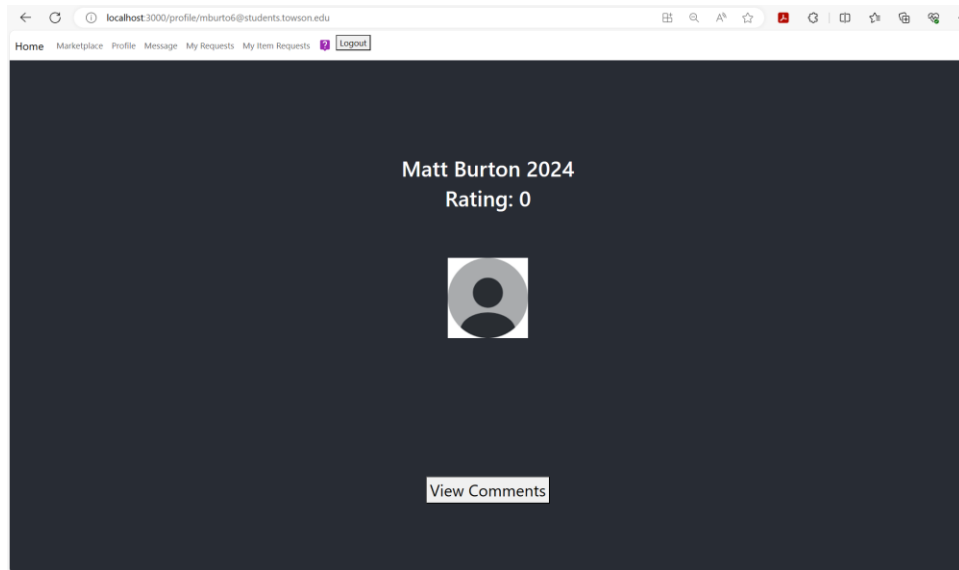
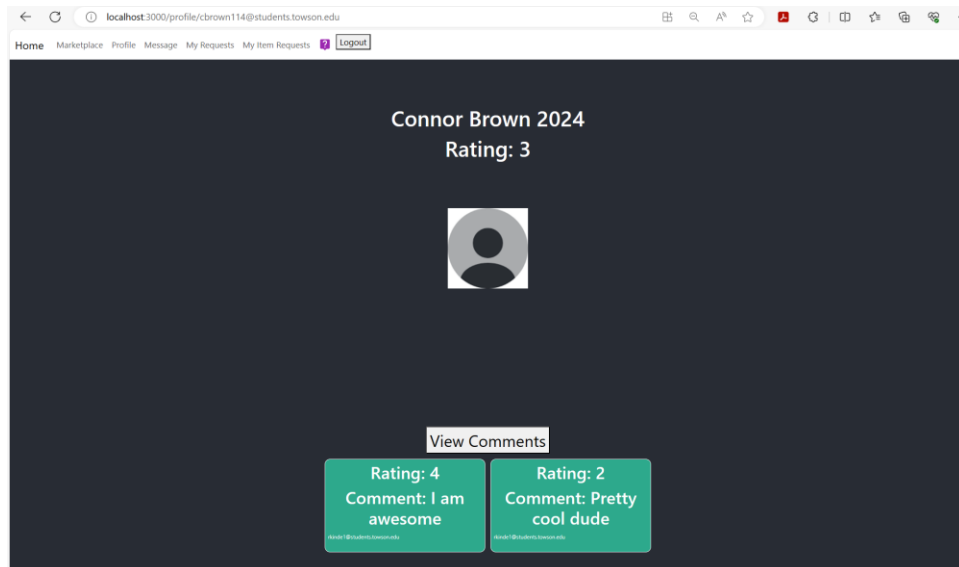


Figure 7.2

If the user clicks on the link with the seller, they will be navigated to the seller's profile which tells them the graduation year, full name, rating, and any comments that other people have made. This profile has no comments.



Here is a profile below that has comments and rating total.



Initiate Transaction

Returning to figure 7.1 from earlier, if the user clicks “initiate transaction”, they will receive an alert which says “success” and will be redirected to the my requests page like below:

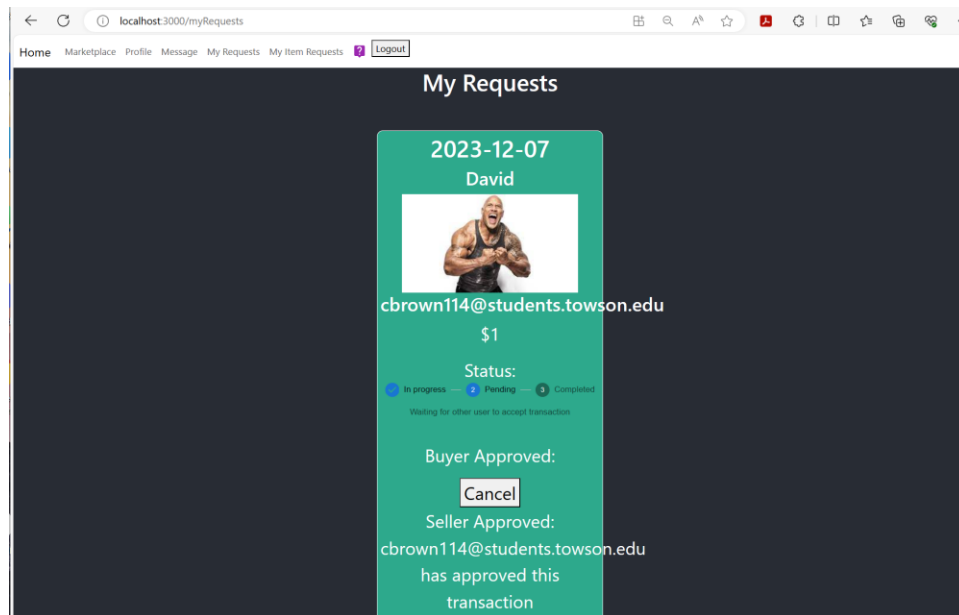


Figure 8.1

The user will then be able to view all of their requests that they have made. From here the buyer will have to approve of the transaction and then wait on the seller to approve the transaction before they can rate each other.

Below shows what the user can do while they are waiting for the seller to approve. They have the option of deleting the transaction in case they decide that they no longer want the item. The status will also stay in progress. If the user deletes the transaction, it will disappear from the page and from the seller's page.

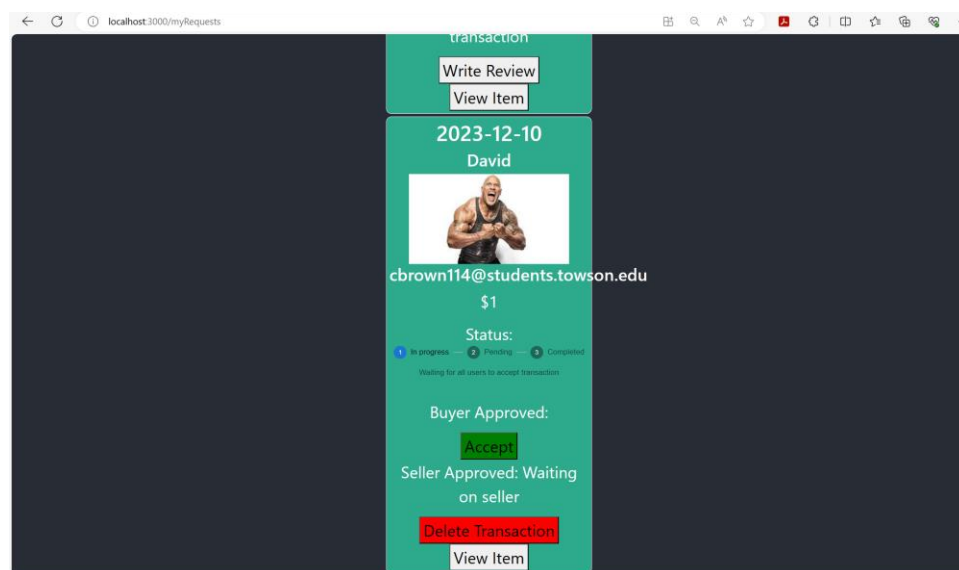


Figure 8.2

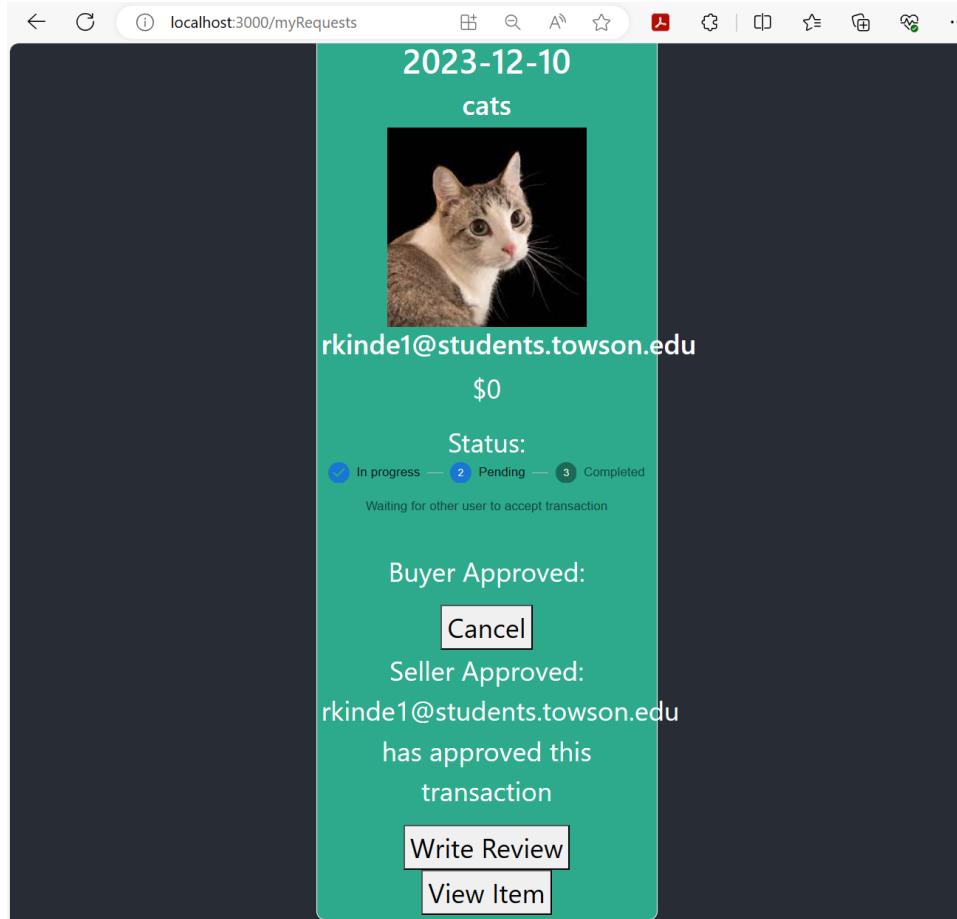


Figure 8.3

Once both parties accept, the status bar changes and the option to delete transaction no longer exists. Instead, the user will have the option to write a review for the seller.

My Item Requests

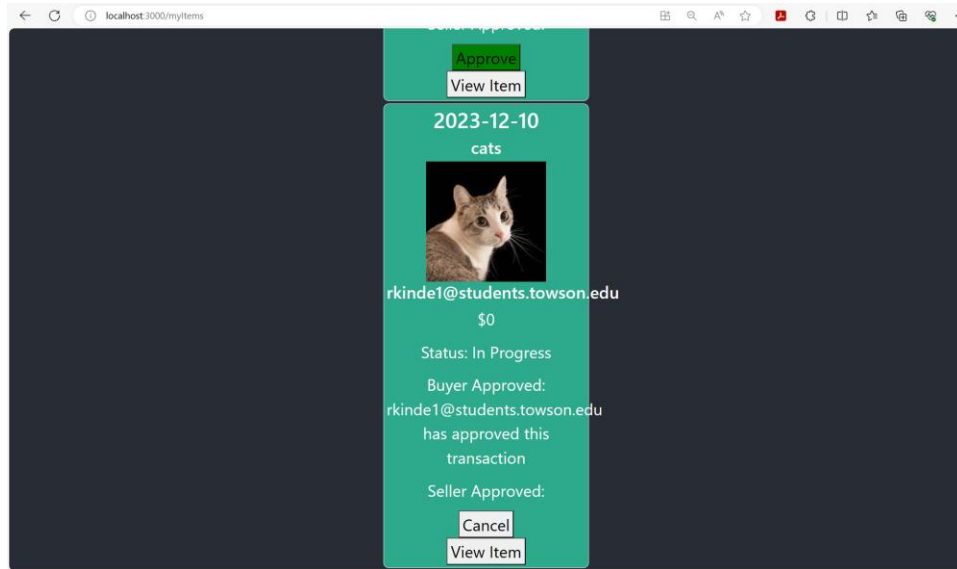


Figure 9.1

My item requests are a page for sellers to look at initiated transactions from other buyers. It resembles the myRequests tab very similarly. The only difference is that they do not have a rate buyer or delete transaction option.

Messaging Feature

There are two ways a user can navigate to the messaging feature. They can find it on the navbar or find the blue button on the view item in figure 7.1. After that, the user will be navigated to a page that looks like below:

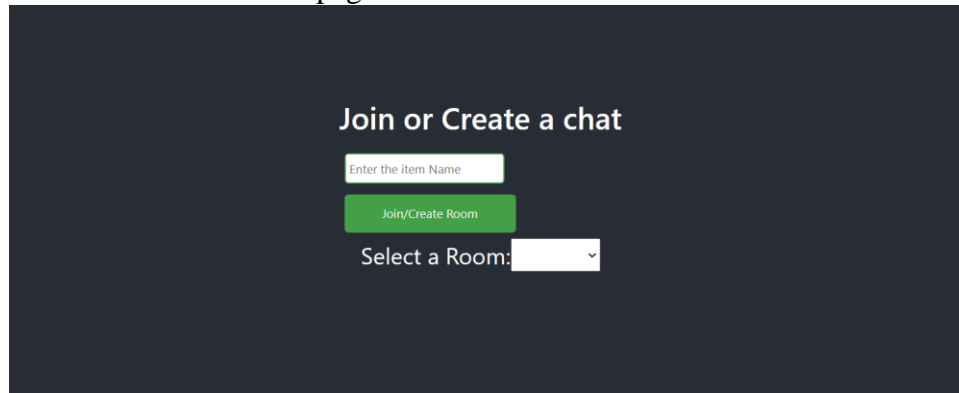


Figure 10.1

The user can enter the item name and create join/create room and enter their message of choice. They can send messages and receive messages from other people with that same item name that will show up as blue. The user's message will show up as green with the time and person's name.

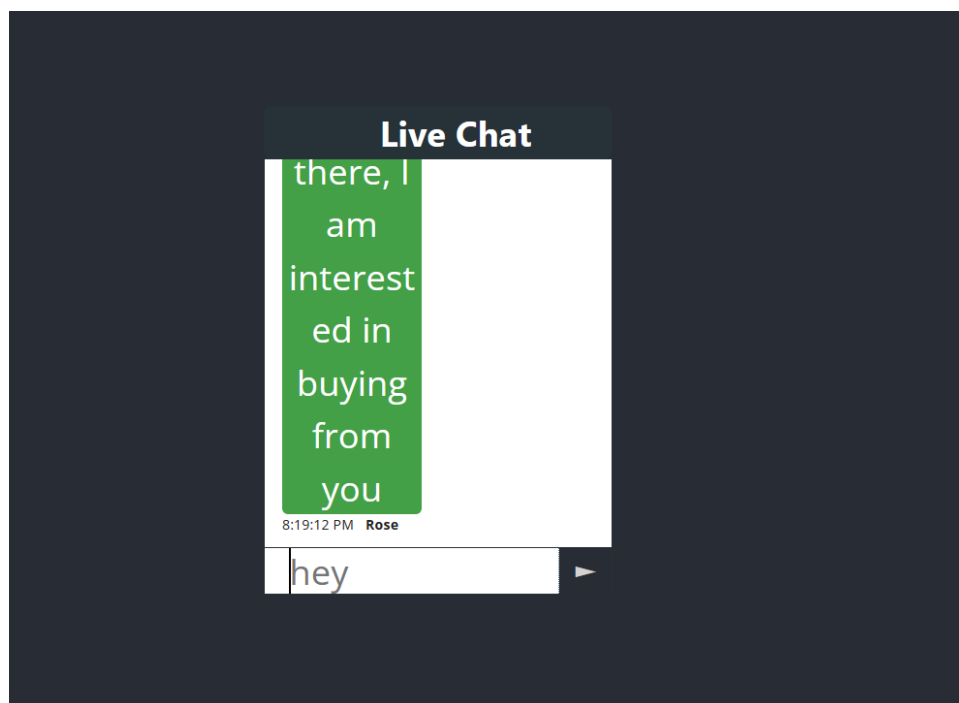


Figure 10.2

If the user exits the chat, they can return to it by selecting the room of the item name they had before shown in figure 10.3 below. If the user has issues with contacting the seller, they can also reach them by email.

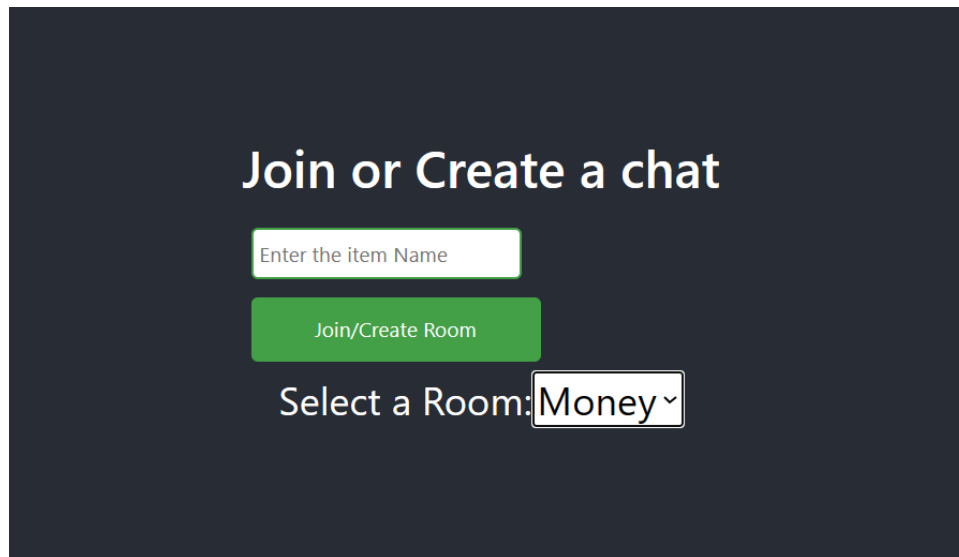
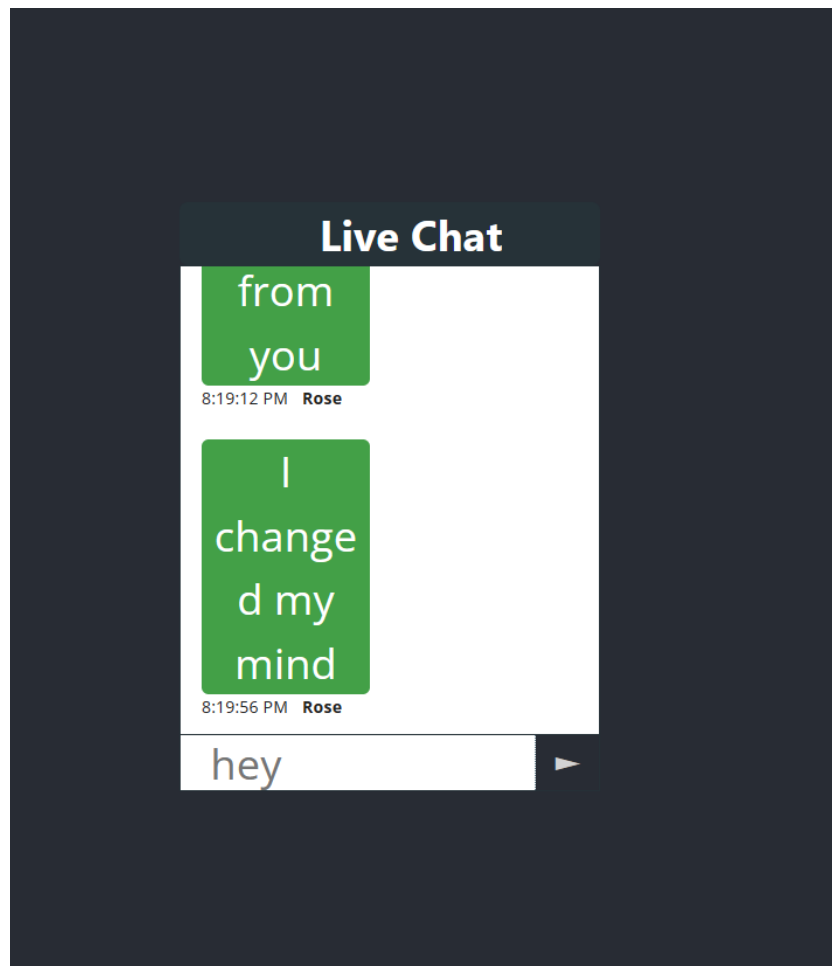


Figure 10.3

Here is showing the continuation of the messages after rejoining the room.



Private Routing:

The next feature on our list was the implementation of private routes. We wanted to ensure that only authorized accounts are able to view the marketplace.

If a user attempts to view the marketplace without an account, they will be rerouted to the login page where they can login or register.

