GROUP H

Team members:

Nguyen, Trung Minh Nguyen, Justin Oberhelman, Andres William Opial, Blake Pardo, Devin Jakob

Github link:

https://github.com/trungnguyen10/CSC4330ProjectGroupH

Members' contribution:

- Trung Nguyen: User Stories, System Architecture, Use Case Diagrams, Database schema, Skeleton Codes, APIs for User, Wishlist, Authentication
- Justin Nguyen: User Stories, Features Description, Frontend(Log In, Homepage)
- Andres Oberhelman: User Stories, Design documents, Frontend(Sign Up, Profile)
- Blake Opial: User Stories, Features Description, APIs for Listing
- Devin Pardo: User Stories, Features Description, Test Cases, Test Code

am members:	
Github link:	1
Members' contribution:	1
User Stories	3
System Architecture	9
Design Documents Activity Diagram Signup and Login Context Model	10 10 11
Sequence Diagram Use Case	12 13
Feature Description Homepage Sign Up Login Profile Page ListingPage	14 14 14 14 15 16
Columbus List Database Schema	17
Test Cases Sign Up Log In Testing Program Background Test Instruction:	18 18 19 21 21
APIs Documentation User and Authorization Wishlist Listing	23 23 25 26
Front End Login Page Signup page Homepage	29 29 30 31

User Stories

Trung Nguyen User Stories:

- As a student, I want to be able to sign up for the system using my student email and a
 password so that I can buy can sell items in the system with other students in the
 university
- As a registered user, I want to be able to login to the system using my student email and my password.
- 3. As a registered user, I want to be able to see recently posted listings for sale after I log in. Every listing needs to have a title, an image, and a price.
- 4. As a registered user, from the homepage, I want to browse as many listings as I want by browsing page by page.
- 5. As a registered user, I want to be able to make a search within the home page. I want to be able to use one or more words to search for items which I want. Those words can be either the title of the listing or the tag of the listing. From the results, I want to be able to filter the result by different categories such as price, date listed, tags
- 6. As a registered user, I want to be able to filter the listings in the home page by price
- 7. As a registered user, from the home page, I want to be able to create a new listing to sell a product.
- 8. As a registered user who want to sell an item, I want to be able to make a new listing with following information: title, images, price, contact information, and tags
- 9. As a registered user who is creating a new listing, I want to be able to quickly tag the listing by a set of predefined tags such as: appliances, books, computers, electronics, video gaming, clothes, others, ...
- 10. As a registered user, I want to be able to access to my profile page where I can find my information, and my listings
- 11. As a registered user who is selling items on the system, I want to be able to edit the price of a listing from my profile page.
- 12. As a registered user who is selling items on the system, I want to be able to remove a listing from my profile page when the listing is sold or no longer available
- 13. As a registered user, whenever I see a listing whether in the home page, or search result, I want to be able to add that listing to my wishlist so that I can check it back any time when I decide to buy it.
- 14. As a registered user, whenever the price of a listing which is in my wishlist changes, I want to be notified of this change. This notification should be an alert so it can catch my attention.

Justin User User Stories:

- 1. Signup: A new user enters the website for the first time unaware of what the website is like, but aware of the services provided. They browse the front screen and see a highlighted button in the corner prompting them to sign up. After clicking the signup button the user fills out the form for signing up.
- 2. Login: On this website, there are multiple ways to login to an account. The first way is on the front page of the website. There is a login button on the top right next to the signup button. The user wants to use a service, already has an account but has forgotten to login. When trying to access the service, they are redirected to a login page.
- 3. Home Page: The user enters the website via a link through a search engine. The link brings them to the home page. On the home page, there is a section with newly posted items from sellers on the website. The section is a revolving window, periodically switching to another item. If the user clicks on the section, it brings them to the desired item.
- 4. Home Page Browse: When the user enters the home page, there is a catalog of suggested items. This particular user has no account so the catalog recommends popular items and vendors to the user. There are several other catalogs with suggested items based on a genre; tech, home goods, etc. The user clicks on an item in this catalog. They aren't interested in buying the item so they hit the back button. When they do, they are returned to the catalog where they were at, the same position, and the same items in the catalog from before.
- 5. Home Page Search: The user wants to search for a genre of item. At the top of the webpage, there is a section with a search icon and area where you can type in your search. The user types in a vague search term and the database does its best to create a suggestion based on what the user has typed. The user sees a suggestion and clicks on it. If the term fits a tag, the search pulls up items listed under the tag.
- 6. Homepage Filter: The user wants to exclude certain types of items from the catalog on the homepage. The user already has an account and is signed in. On the top left of the catalogs of items is a dropdown menu with the name "filter by." The user clicks on this menu and is presented with a check mark box with various criteria they want to filter by. When the user clicks on a check mark, the catalog seamlessly refreshes in the background with items that fit the filtering criteria. When the user logs off, the filters are not saved.
- 7. Homepage New Listing: The user logs in to their account wanting to sell an old couch. After logging in they are directed to the homepage. On the home page, there is a drop down menu linked to the user's profile icon. When the drop down is activated, there is an option to add a listing. The user clicks on "add listing" and is redirected to a form they have to fill out going over the details of what they are selling, how many, and how they want to deliver the product. The user fills out all required documents and hits the continue button. The listing is added to the catalog and can be viewed by others.
- 8. New Listing: When a new listing is created there are many items on the form that need to be filled out. The Item title is what is displayed in the catalog. There is an optional section for item description that is displayed when the item is opened in the catalog.

- There is an optional check box section for item tags that the listing will go under. Price, contact information, and title are required for the listing to be posted.
- 9. Predefined tags: The user wants to search items based on predefined tags created by the website. The user opens up the search bar. The search bar is a drop down menu with suggested terms but has a button to search by tag. The user presses search by tag, they are redirected to a page with a list of tags organized in alphabetical order. At the top of the page, there is a search bar to search for specific tags. The user presses the search bar and selects multiple tags. The tags are pushed into a list of tags the user has selected. The user presses enter to pull a catalog of items that match their search tags.
- 10. The user wants to edit the price of their listing. They go to the account section of the home page and select the option called "my listings." The user is redirected to a page with a list of their current listings in their cataloged forms. On each catalog there are multiple option buttons for the function "edit listing." On the edit listing drop down, the user is presented with several options.
 - a. Edit listing: The user is redirected to a form where they can change the details of the listing. At the bottom of the page there is an option to cancel or save the changes.
 - b. Remove listing: The user can remove the listing with the press of a button. After the button is pressed, a prompt pops up asking if the user wants to delete the item. The user can cancel or confirm.
- 11. When the user is browsing, there is an option on every item to add the item to a wishlist. If the user does not have an account, they are redirected to the sign up page. If the user does have an account, the item is added to a wishlist section located in the users account section.
- 12. If the user wants to receive a notification of when an item changes price, they can select an option on the item page "notify me when the price changes." If the user has an account an email will be sent to the email tied to the account. If the user has no account, they receive a prompt to sign up.

Andres Oberhelman User Stories

- As a student of Colombian State University, Sarah wants to sign up to access the online marketplace. Sarah applies by submitting her email and creating a password for her to use. The system then verifies her email.
- 2. Kenedy wants to login into her student profile on the market place. She enters the website and puts in her email and password in the designated slots and logins.
- 3. Anthony looks through the recently posted items to see if he can find a bookbag to use for class.
- 4. Michael browses through the items trying to find school materials for his next semester classes.
- 5. Alexis needs a new phone charge so she looks up electronics in the search area and goes to the tag that says electronics and searches through the listings with the search term "charger" and finds the one she needs.
- 6. Annie wants to buy a notebook for her art class but could not afford an expensive one so she filtered the price to make the listed only show ones that were less than \$30.
- 7. Mark wants to sell his laptop on the market place so he creates a new listing on the homepage for his 2014 Macbook Pro.
- 8. Mark creates a new listing by adding a title, description, image, price, tags, and contact information into the site and his listing is then created and is shown on the website.
- 9. Annalise needs a highlighter to take notes in her science class. She adds the school supplies tag and the drawing tag and it shows her all the items that fit those tags.
- 10. Meghan is selling a handbag on the marketplace but notices she put the wrong price on the item. She clicks her profile and goes to edit the price of the listing and changes it.
- 11. Andy added three textbooks to the listings on the website but decided to keep one of them so he clicks on his profile icon and goes to remove listing and removes one of his listings.
- 12. Cindy wants to buy a certain handbag but cannot afford it at the moment. She clicks on the listing and adds the item to her wishlist so she can go back and buy it when she can afford it.
- 13. Abbey is searching through the listings for a certain textbook and after a while a notification appears on screen letting her know that the same textbook she added to her wishlist has gone down in price. Knowing this Abbey goes and buys the textbook.

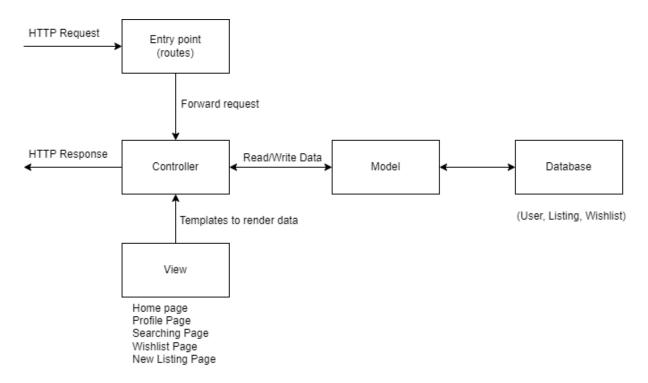
Blake Opial User Stories

- 1. As a student, I want to sign up to use ColumbusList with a verified school email and password.
- 2. As a user, I want to be able to log into ColumbusList using the same email and password I created my account with
- 3. As a user, I would like to see all of the most recent listings.
- 4. As a user, I would like to be able to browse various listings on a homepage.
- 5. As a user, I'd like to be able to search for specific items under predefined tags
- 6. As a user, I'd like to filter all listings by categories(price, alphabetical, etc.)
- 7. As a user, I'd like quick access to add my own listing from the homepage
- 8. As a user selling an item, I'd like to create my own listings with a description, price, and images that I determine.
- 9. As a user selling an item, I'd like to be able to give my listing specific tags to be found easier from the search feature.
- 10. As a user selling an item, I'd like to be able to adjust the price of my listings from my account page at any time.
- 11. As a user selling an item, I'd like to remove my item from being listed whenever I finally sell it to someone.
- 12. As a potential buyer, I'd like to be able to save any listings I'd like to come back to in a wishlist.
- 13. As a potential buyer, I'd like to receive a notification if any listings in my wishlist adjust their prices.

Devin Pardo User Stories:

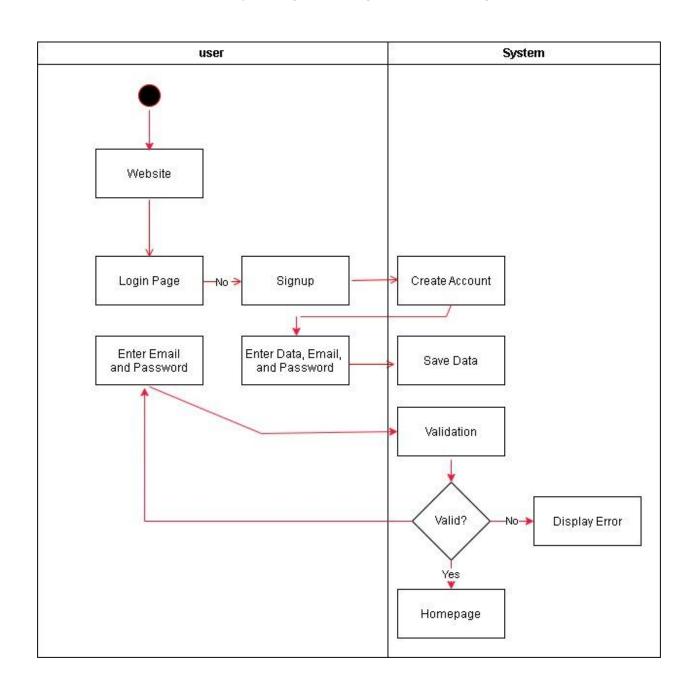
- 1. Signup: As a student, I would like to sign up with a username, password, email, and phone number.
- 2. Login: As a student, I would like to sign in with my username or email and password.
- 3. Homepage-
- 4. Recently posted: As a student, I hope to see new items posted and constantly update when I open and refresh the site.
- 5. Browse: As a student, I hope there will be multiple ways to look at miscellaneous or specific items while I scroll through the homepage.
- 6. Search/Querying Tag: As a student, I want to be able to search for items by typing a specific name or classification tag.
- 7. Filter: As a student, I would like to have a filter that will go through items by name and price and be able to go from increasing or decreasing order.
- 8. Make a new listing: As both a student and seller, I would like to be able to sell an item by simply tapping a button on the homepage.
- 9. New listing: title, description, images, price, contact information, tags: As a student and seller, I would like to type a title, description, price, and contact information for a listing while being able to put as many pictures I need for it.
- 10. Pre-defined tag: As a student, I would like all tags to be predefined so that it would be easier to look for or classify an item quickly and easily.
- 11. Edit price of a listing from the profile page: As a student and seller, I would hope to change the price of my listings within my profile page.
- 12. Remove listing from the profile page: As a student and seller, I would like to delete a listing on my profile page and have a caution message so that you can truly make this decision final or not.
- 13. Wishlist: As a student, I would like to make a Wishlist in order to save items I want to get at a later date.
- 14. Notification alert when price changes: As a student, I want there to be a notification that tells me when a price has increased or decreased in my Wishlist.

System Architecture

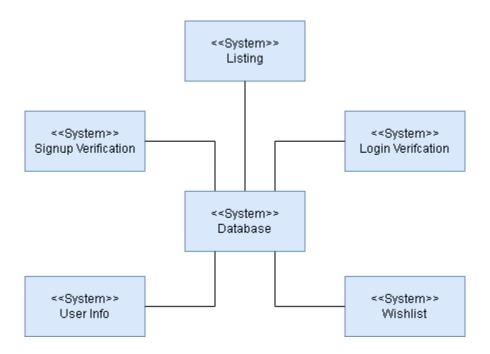


Design Documents

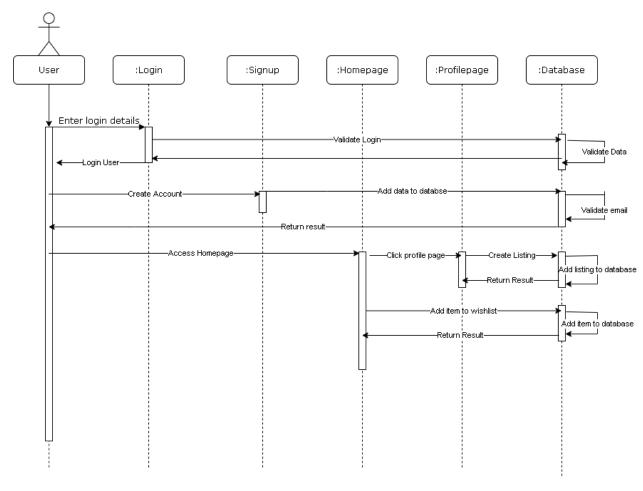
Activity Diagram Signup and Login



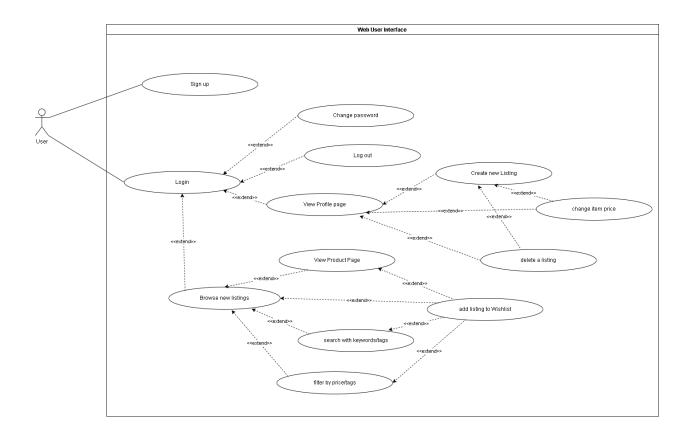
Context Model



Sequence Diagram



Use Case



Feature Description

Homepage

Description

- 1. As a registered user, I want to be able to see recently posted listings for sale after I log in. Every listing needs to have a title, an image, and a price.
- 2. As a registered user, from the homepage, I want to browse as many listings as I want by browsing page by page.
- 3. As a registered user, I want to be able to make a search within the home page. I want to be able to use one or more words to search for items which I want. Those words can be either the title of the listing or the tag of the listing. From the results, I want to be able to filter the result by different categories such as price, date listed, tags
- 4. As a registered user, I want to be able to filter the listings in the home page by price

Constraints: The Homepage must efficiently load multiple item listings at once. Allow for easy searching through filtering or text searching.

Comments: Based on Group H's scenarios.

Sign Up

Description

- As a student, I want to be able to sign up for the system using my student email and a password so that I can buy and sell items in the system with other students in the university
- 2. A new user enters the website for the first time unaware of what the website is like, but aware of the services provided. They browse the front screen and see a highlighted button in the corner prompting them to sign up. After clicking the signup button the user fills out the form for signing up.
- 3. As a student, I want to sign up to use ColumbusList with a verified school email and password.

Constraints: The user must sign up for the website through the homepage, and must present a valid school email address.

Comments: Based on Group H's scenarios.

Login

Description

- 1. As a registered user, I want to be able to login to the system using my student email and my password.
- 2. Kenedy wants to login into her student profile on the market place. She enters the website and puts in her email and password in the designated slots and logins.
- 3. Login: On this website, there are multiple ways to login to an account. The first way is on the front page of the website. There is a login button on the top right next to the signup button. The user wants to use a service, already has an account but has forgotten to login. When trying to access the service, they are redirected to a login page.
- 4. As a student, I would like to sign in with my username or email and password.

Constraints: Access must be restricted to users with a verified school email address or username that is saved in the User Database.

Comments: Based on Group H's scenarios.

Profile Page

Description

- 1. As a registered user, I want to be able to access to my profile page where I can find my information, and my listings
- 2. As a registered user who is selling items on the system, I want to be able to edit the price of a listing from my profile page.
- 3. As a registered user who is selling items on the system, I want to be able to remove a listing from my profile page when the listing is sold or no longer available
- 4. As a user selling an item, I'd like to be able to give my listing specific tags to be found easier from the search feature.
- 5. As a user selling an item, I'd like to be able to adjust the price of my listings from my account page at any time.
- As a registered user who is selling items on the system, I want to be able to remove a listing from my profile page when the listing is sold or no longer available
- 7. If the user wants to receive a notification of when an item changes price, they can select an option on the item page "notify me when the price changes." If the user has an account an email will be sent to the email tied to the account. If the user has no account, they receive a prompt to sign up.

8. As a registered user, whenever the price of a listing which is in my wishlist changes, I want to be notified of this change. This notification should be an alert so it can catch my attention.

Constraints: Changes like item's price must be reflected in the database in a short amount of time then notify users who added that item in their wishlist. Adding/deleting listings must be reflected on the page.

Comments: Based on Group H's scenarios.

ListingPage

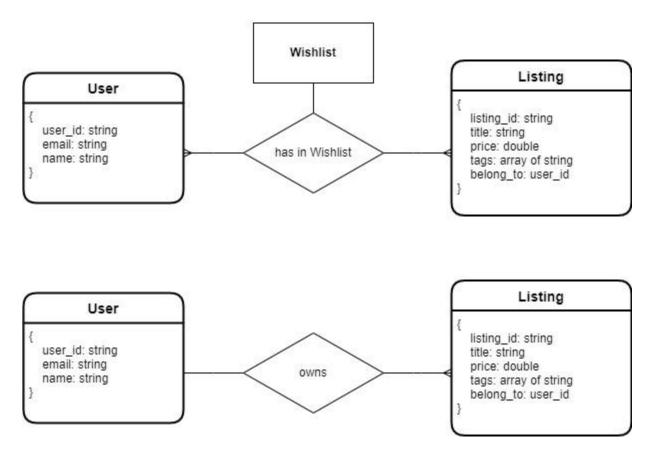
Description

1. As a registered user, whenever I see a listing whether in the home page, or search result, I want to be able to add that listing to my wishlist so that I can check it back any time when I decide to buy it.

Constraints: Only the owner of the listing may edit its details. Potential buyers may save the listing to a wishlist for a later date.

Comments: Based on Group H's scenarios.

Columbus List Database Schema



We use MongoDB as a database management system for our website. MongoDB is a NoSQL database program where data is stored in collections and documents. Each collection contains documents. Each document is an instance of the object and is described in JSON-like syntax. In the application, an user is uniquely identified by a user_id; a listing is uniquely identified by a listing id.

Users and listings have 2 different relationships.

- An user can own 0 or more listings. A listing can only be owned by exactly one user.
- A listing can be in the wishlist of 0 or more users. An user has 0 or more listings in his wishlist

The relationships between users and listings are reflected in belongTo properties of the listing object and Wishlist database model.

Test Cases

Sign Up

1. Test case description: As a student, I want to be able to sign up for the system using my student email and a password so that I can buy and sell items in the system with other students in the university

Input: input type: text/text/email/password/password

Insert placeholder: First, Last, email, password, confirm

Input button: submit

Condition or function under test: The user must sign up for the website through the homepage and must end the email with @columbus.edu. Also, the password cannot be over 8 characters-long.

Expected Output: It will accept your first and last name, email, and password. You will re-enter the password so that you can confirm the same format with the original inputted password. Then, you can submit and have the account created.

Output: First and last name, email, and password are automatically encrypted, and the response (if successful) would have the success message and a token for authentication.

2. Test case description: If the names, email, password, and/or confirmed password are invalid or empty, then the user will be notified to correct what they wrote or come up with a better option.

Input: input type: first/last/email/password/confirmed password

Insert: invalid first/last/email/password/confirmed password

Input button: submit

Condition or function under test: Using an email that the website doesn't recognize or accept, exceeding the 8 character limit, and not having the confirmed password match up with the original one.

Expected Output: The user will be notified with a message saying that the passwords don't match and the email will not be accepted.

Output: "Please tell me your name", "A user must have an email", and "Please provide a password" will be highlighted in their respective fields. If the confirmed password does not match, then it will state "Password does not match."

3. Test case description: If I end up in the signup window but already have an account, I could see a message saying, "Already have an account?" and click the "Login here" button on the page that brings me to the sign-in page.

Input: Access file: signup

Login here button: selected

Access file: login

Condition or function under test: The user must have an account made beforehand and have the button bring them to the login.

Expected Output: After clicking the button, the user will be brought to the login page so that the person could type in their previously made email and password.

Output: Clicking on "Login here" brings you to the login page.

Log In

1. Test case description: As a registered user, I want to be able to login to the system using my student email and my password.

Input: input type: email/password

Insert placeholder: email/password

Input button: submit

Condition or function under test: Access must be restricted to users with a verified school email address and password that are saved in the User Database.

Expected Output: If both your email and password match, then you can access all the pages, information, and advantages of your account.

Output: The response (if success) would have the success message and a token for authentication

2. Test case description: If you insert an incorrect or random email and/or password that doesn't match with the profile, then you will be notified.

Input: input type: email/password

Insert: wrong email/password

Input button: submit

Condition or function under test: The user must have put down an email or password that does not match with the ones used for the created account.

Expected Output: Near the email and password, the user will be reminded to provide the correct email and password.

Output: The user will be told that the inserted email/password are wrong and to input the correct email and password.

3. Test case description: The user wants to use the service, but does not have an account and wants to go to the signup page through the login page for convenience.

Input: Access file: login

Sign up here button: selected

Access file: signup

Condition or function under test: The user must not have an account, and the "Sign up here" button should go to its assigned page.

Expected Output: The person can find a "Need an account?" message and click "Sign up here" to get to that specific page.

Output: Pressing the "Sign up here" button will bring you to the sign up page.

Testing Program Background





Chai: Chai is a BDD/TDD assertion library for node.js and easily works well with other testing frameworks for java code. The BDD styles of "should" and "expect" provide expressive language, and the TDD style of "assert" keeps things simple but still effective.

Mocha: Mocha is another testing framework used within node.js, and it's truly effective at making asynchronous testing easy and simple.

Test Instruction

Assuming the user has access to npm, a coding platform, and node.js you can submit these items under the terminal:

Npm install mocha

Npm install chai

Make sure that mocha, chai, and jest are part of dependencies and other sets in the package.json and package-lock.json

Next, you would have to put "test": "mocha" under the "scripts" section of the .json files Create a test java file within the coding program

Set up the code so that it could be read through chai and mocha in node

Such as bringing in should, expect, and assert as a constant for chai; and describe, it, and other terms that deal with testing for mocha

Now, put "npm test" or "npm run test" in the terminal

There the user will see if their test(s) is/are passing or failed For example, it result in:

- > csc4330projectgrouph@1.0.0 test
- > mocha

Login

- ✓ login should allow you to fill in your created email and password
- ✓ login should return type string
- ✓ login should notify the user when email and/or password are wrong
- ✓ The button at the bottom of the login page should bring the user to the signup

Signup

- ✓ Signup should allow you to fill in name, email, password, passwordConfimed
- ✓ Signup should return type string
- ✓ Signup should have be a student email
- ✔ Password should have 8 characters at most in order to be accepted
- ✓ Confirmed password should completely relate to password
- ✓ The button at the bottom of the signup page should bring the user to the login

10 passing (31ms)

Following that, the user can either add more tests or fix those that failed

APIs Documentation

APIs for Columbus List are documented using a tool called Postman. The details of documentation are at: <u>Columbus List (getpostman.com)</u>

User and Authorization

POST Sign Up

```
{{URL}}api/v1/user/signup
```

User's name is only a field in the document. So if you have first and last name just concatenate them into single string.

BODY raw

```
{
   "name": "blake",
   "email": "blake@columbus.edu",
   "password": "pass1234",
   "passwordConfirm": "pass1234"
}
```

POST Login

{{URL}}api/v1/user/login

BODY raw

```
{
   "email": "trung@columbus.edu",
   "password": "pass1234"
}
```

PATCH Update Password 🗎

```
{{URL}}api/v1/user/updatepassword
```

User must be logged in to change the password. So if you want to test this API, make sure you're already logged in

AUTHORIZATION Bearer Token

Token {{jwt}}

BODY raw

```
{
    "currentPassword" : "pass1234",
    "newPassword": "newpass123",
    "passwordConfirm": "newpass123"
}
```

GET Log Out

{{URL}}api/v1/user/logout

Wishlist

POST Create Wishlist {{URL}}api/v1/wishlist User must be logged in to add a listing to wishlist. The body of the request must have the _id of the listing. Owner cannot add his listing to wishlist. Not valid listing _id will result an error **AUTHORIZATION** Bearer Token Token {{jwt}} **BODY** raw "listing_id":"6266f83dbfce521c3c51ba65" } GET Get User Wishlist {{URL}}api/v1/wishlist User must be logged to get the array of his wishlist listing. **AUTHORIZATION** Bearer Token Token {{jwt}} DEL Delete The Listing in Wishlist 🕯 {{URL}}api/v1/wishlist/6266f83dbfce521c3c51ba6 User must be logged in to delete a listing in his wishlist. Return error for invalid listing _id **AUTHORIZATION** Bearer Token Token {{jwt}}

Listing

Listing

POST Create Listing

```
{{URL}}api/v1/listing
```

When creating a new listing, only two items are required.

The fields title and price are required by the BackEnd. If eihter of these two fields are empty, the database will throw an error.

tags are optional, simply pass them through as an array of strings. There currently is no limit to tags.

The fields <code>createdAt</code> and <code>belongTo</code> are generated by the database when all necessary information has been passed to it. They get assigned the current calendar date and the user's ID, respectively.

AUTHORIZATION Bearer Token

Token {{jwt}}

BODY raw

```
{
    "title": "computer",
    "price": 125.99,
    "tag": "tech"
}
```

GET Get All Listing

 $\label{eq:continuous} $$\{URL\}\}$ api/v1/listing/?[text][search]=tech\&price[gt]=300\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=1.000\&sort=-createdAt\&limit=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&page=10\&p$

- Filters: Fields available for query are: price, createdAt

 Use gte, lte, gt, lt for aggregated query. For example / ?price[lt]=300 will query the listings of which price is less than 300.
- Searching: support text search, the result will include all listings that have the matching text in its title or tag. For example, /?[text][search]=tech will result all listings having 'tech' in their title or tag
- Pagination: support pagination the returned query. limit is the number of items per page, page is the page number of the result. e.g., /?page=3&limit=10 will return the page number 3 with 10 items per page.
 The default value is limit=20 and page=1
- Sort: support sorted by the specified fields, use _ for decreasing order. e.g., /?sort=-price, createAt will sort the result in decreasing price order then increasing createAt order. By default, the result is order by createAt in decreasing order.
- Support combination of search, filters, sort, pagination
 e.g. /?[text][search]=tech&price[lt]=300&sort=-createdAt&limit=10&page=2]

AUTHORIZATION Bearer Token

Token	{{jwt}}
PARAMS	
[text][search]	tech
price[gt]	300
sort	-createdAt
limit	10
nage	1

PATCH Update A Listing

{{URL}}api/v1/listing/6261a3ca79cd8f30a846e672

```
api/v1/listing/id
```

when creating a request to update a listing, it's imperative that the specific listing [id] be passed as a parameter.

On the server side, the <code>user_id</code> is compared to the <code>belongTo</code> varible to verify user ownership. If it fails an error is thrown.

Once verified, simply pass whatever variables are being changed (ie, price, title, tags). The database will only update the variables that need to be changed, the rest will be the same.

AUTHORIZATION Bearer Token

Token {{jwt}}

PARAMS

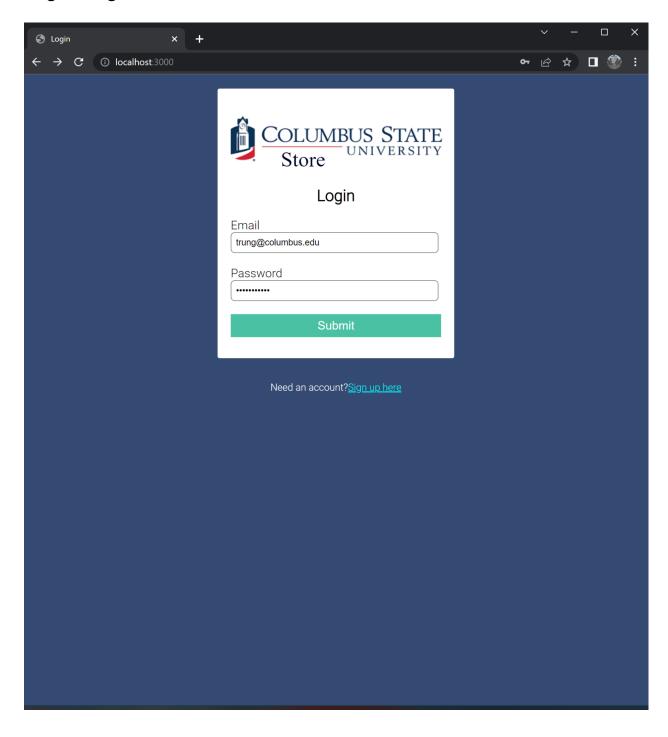
_id The specific listing ID of the document to be changed.

BODY raw

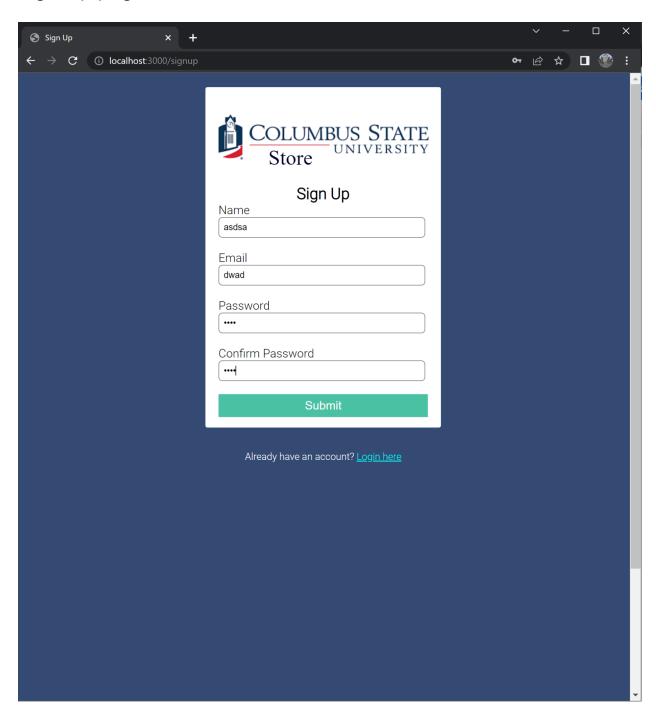
```
{
    "price": 30000
}
```

Front End

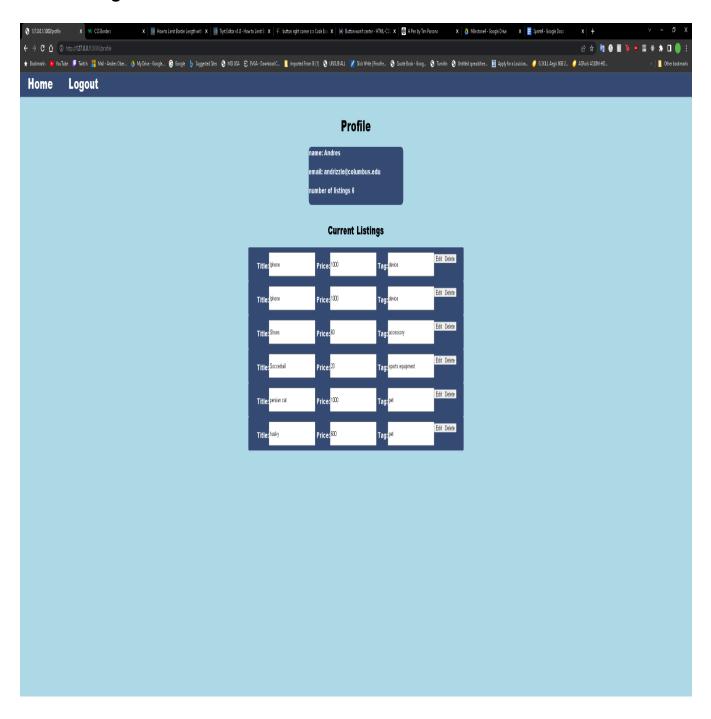
Login Page



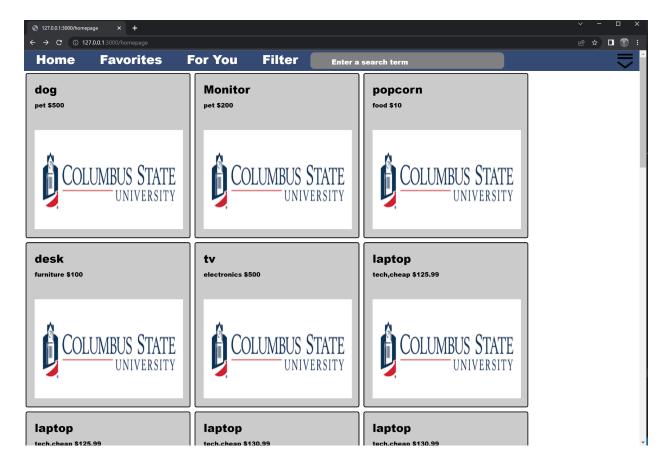
Sign Up page



Profile Page



Homepage



Product Backlog (Updated)

Requirements Specification Document

Home Page/Listing Catalog

- Recently posted items from item database
- Browse items by pagination
- Search item by title
- Search item by tag
- Filter item by price
- Link to Create New Listing page
- Link to Profile page

Sign Up

- Email verification
- Sign up form

Login Page

- Verify Email/Password
- Pull information from User Database
- Redirect to Home Page

Profile Page

- User information
- Change password
- User's current listings
- Edit price of posted listing
- Remove a listing

Listing Page

- Picture gallery of the item
- Basic information (Price, Owner of the object)
- Tags
- Add to Wishlist button
- Add to cart

Listing Database

- Name of Product
- Price
- Owner information
- Associated Tags

User Database

- Email (***@colombus.edu)
- Password
- User created Listings

Refactoring code to improve understandability and performance

Sprint Backlog

APIs for User
APIs for Authentication
APIs for Listing
APIs for Wishlist
Testing
Homepage
Profile page
Login page
Sign Up page