BSMASH HatHub

- Contributors:

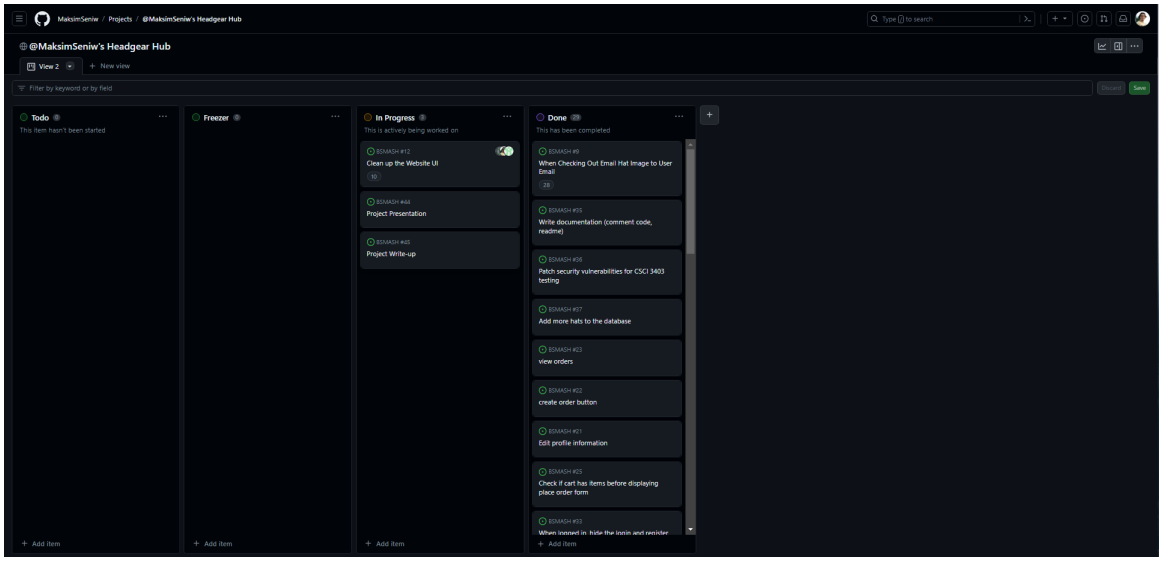
Maksim Seniw, Blake Raphael, Saul Drantch, Streck Salmon, Abdullah Yassine, Hyder Baig

- Project Description:

For our project, we decided to create an online store because it allowed us to be flexible with things we wanted to add while being able to fulfill all of the project requirements. In this online store, a user will be able to purchase hats with a given amount of funds. Before they do so, they must register a new account and enter information such as their name, favorite type of hat, email, username, and password. Once the user correctly logs in with their username and password, they will have access to the store with a set amount of funds. They may also edit their account to add more funds, and change their email, name, password, or username. After they have decided on what hats (and how many of each hat) they would like to purchase, they can add them to their cart where they will be able to see their selections, as well as information they will need to enter to deliver the hats. There is also an option to save their choices for later if they don't want to purchase the hats immediately. Once they place their order, they will receive a confirmation email from HatHub and can find their order in the orders tab on the website. Their funds will also be adjusted accordingly after the order is placed. The user can finally log out once they have completed their shopping.

- Project Tracker - GitHub project board:

https://github.com/users/MaksimSeniw/projects/2/views/2



- Video:

  https://cuboulder.zoom.us/rec/share/nbuQH7jjPakcsIw9zqIrRDjeYCfONNnm6T7oCkYk-xqeJZT1XCIMINgjf_IBV9il.V8kXQgbHTHb-EnfE?startTime=1701900243000

  Passcode: 3.D6XvE$

- VCS:

  https://github.com/MaksimSeniw/BSMASH.git

- Contributions:

Saul Drantch - Throughout the course of this project, I jumped between a couple of different things. In the beginning, I added code for the framework of the project, including a functional registration page, login page, and home page. This included the index.js file which had all of the apis needed to get the register and login pages working. After that, I focussed on getting the test cases working in the server.spec.js file and debugging the site while my teammates added pages like the hats, profile, and cart tab. Once the test cases were finished and working, I helped work on the UI styling for the site in all of the ejs files. Finally, I added the email functionality with an api in the index.js file using a service called mailjet.

Maksim Seniw - The main features I worked on were the overall UI for the website and some minor backend adjustments for the UI. Every page that is shown in the website I made the UI for; using CSS and sources I found online. Further, I created the logo for the website and added an additional social media page (And an edit profile page; my version of the edit profile page was very buggy and Blake was the one who actively fixed it and made it functionable) that displayed the individuals who worked on this website. Additionally, I edited a few of the SQL tables, in order for the UI I created to not crash or bug out the website, while also creating a few new routes/pages that would help flesh out the website. Overall, my contribution towards this project was largely focused on the look and feel, along with some of the routing, of the website while contributing slightly to the backend.

Blake Raphael - The main features I worked on were the database structure and the functionality of the items page, cart page, and orders page along with patching some

security holes. My main technologies were PostgreSQL, Node.js, Chai, EJS, and HTML. I also used LucidCharts to create the Entity Relationship Diagram for our database. My contributions towards the database were mainly setting up the structure of the database along with writing SQL and helping debug teammates' SQL statements. My contributions towards the pages and security were mainly through Node.js by ensuring routes were added correctly, making sure these routes were tested correctly, and ensuring they utilized the data from both the client and server side correctly.

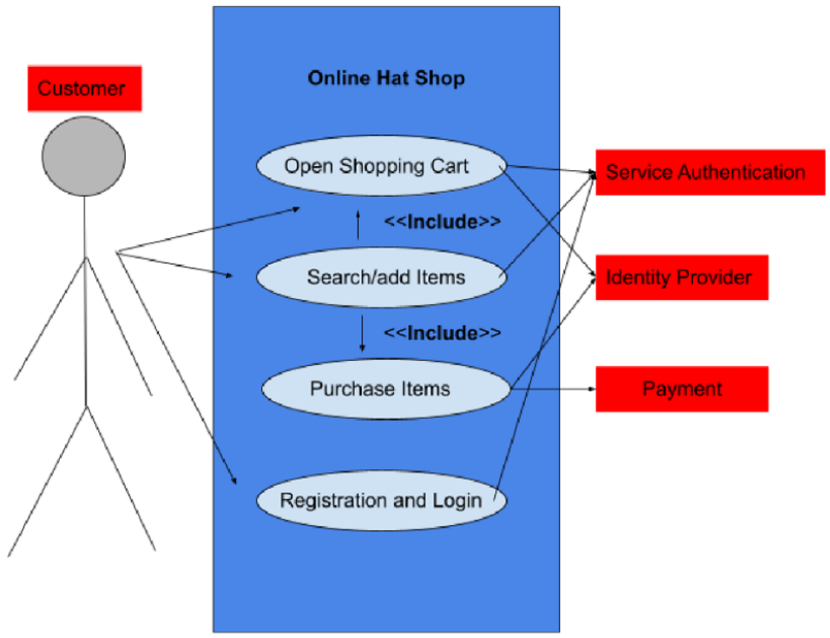
Streck Salmon - The main features I worked on were the cart functionality, database, login/registration, and I also had minor additions to the website's UI. Most of the work contributed over the course of the project as on the backend, enabling the website to function smoothly through clean routing and proper handling of a multitude of cases. A lot of instances where I debugged were in handling unexpected inputs from the user, allowing the website to operate more smoothly and not crash amongst an unexpected input whether it be in login, registration, shipping, or cart. The work on cart mostly consisted of ensuring items were properly tracked and the database was in sync with what was presented to the user. The main technologies I used were PostgreSQL, Node.js, Chai, EJS, and HTML. Most of the work I contributed was within Node.js, EJS, and Chai.

Abdullah Yassine - In the beginning of the project, I worked on discover.ejs, which makes sure that the sign-in page works fine when you log in instead of getting you an error when you log in. Later on, I also worked on the functionality of the cart page. Specifically, I created the Saved For Later feature which allows users to add items into this part of the cart in case they wanted to buy those items in the future and not in the current moment, similar to many online retail stores that provide this feature. Also, near the end of the project, I added a few more hats in the database allowing for more flexible choices when it comes to choosing hats to buy.

Hyder Baig - The main features that I worked on were the database and the images for the products. The main technology that I used was PostgreSQL, and I also used a bit of EJS. I created several databases for the website, and added a lot of the hats for the website, as well as descriptions and pictures for each of them. I also did a bit of debugging. I worked on the functionality on items.ejs, and made sure to fix some errors that were present in the code.

- Use Case Diagram:

Use Case Diagram:



- Test Results

The user was able to successfully register a new account and log in with that account. They registered all on their own and it was consistent with the use case diagram. The user attempted to log in incorrectly and the page displayed an error message correctly and aligned with the use case diagram. The user successfully added hats to their cart due to the buttons on the items page intuitively leading them that way. The user was able to successfully access their cart by seeing that it was an option in the navigation bar. They saved one of their hats in their cart for later because they saw the UI element stating “saved for later”. The user commented positively that they enjoyed the UI colors. The user successfully placed an order and then subsequently canceled that order by intuitively clicking on the orders tab in the navigation bar.. The user successfully deleted the hat from their saved for later. The user was able to successfully place another order. The user successfully checked and edited their profile changing their first name and adding funds. All of these actions fell within the use case diagram. One error that we ran into was double registration. The user mistakenly re-registered instead of logged in. This

was a simple mistake because they misclicked on the wrong navigation bar tab. This presented a bug that we had not accounted for and were able to resolve with a check. Overall the user seemed to use the website intuitively and how we as a team drew it up in our use case diagram. The website seemed to guide them well enough into using it without major roadblocks or issues. In this test we are able to test the following: registration, login, adding hats to cart, adding hats to saved for later, ordering hats, canceling orders, deleting from cart, deleting from saved for later, checking profile, editing profile, and logging out.

- Deployment:

http://recitation-15-team-05.eastus.cloudapp.azure.com:3000/