Application Title: funny haha maker

Team Members: (Name, Github Username, Email)

- Alexander Juenemann, axjuenemann, alexander.juenemann@colorado.edu
- Aneesh Kaushik, aneesh27, anka9751@colorado.edu
- Faez Alanazi, omgfayez, Faez.Alanazi@colorado.edu
- Lindsey Trussell, lindsey-trussell, lindsey.trussell@colorado.edu
- Solus Thompson, Solus5, solus.thompson@colorado.edu
- Travis Johnson, tjohnson23, trjo4692@colorado.edu

Project Description

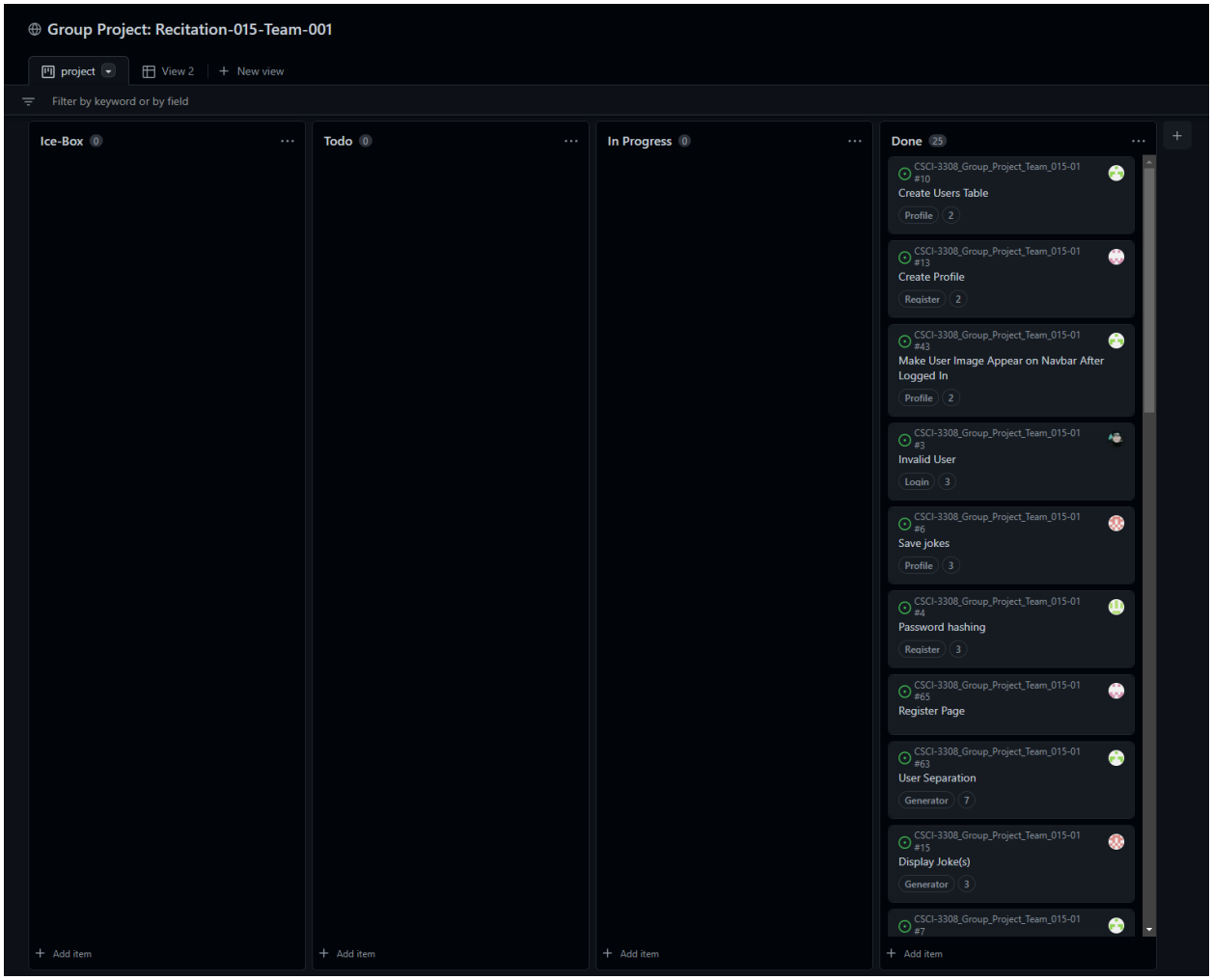
This application utilizes the joke APIs icanhazdadjoke, YOMOMMA, Blauge.xyz, Geek Jokes, and Bread Puns in order to provide swift access to hilarious jokes. Each user can create their own personal account, including the ability to set up a profile picture! Users can generate as many jokes as they want from the mentioned APIs 15 jokes at a time! These jokes are then displayed in an easy-to-read format that features a save function with each joke. Every single joke that is generated can be saved to a user's account, which they can then view later under the saved page. Once a user has gotten all the laughs they can out of a joke, they can unsave jokes that they have previously saved. If in the future the joke once again tickles the funny bone, it can be resaved. Saved jokes are unique to each user, and several users can save and unsave the same jokes as much as they see fit. If a user no longer fancies their username, profile picture, or password, all of these can be edited within the profile page. With the easy-to-use navigation bar, feel free to peruse the application!

Project Tracker
GitHub project board: https://github.com/orgs/CSCI3308-Fall22/projects/53

Group Project: Recitation-015-Team-001

project | View 2 | + New view

	Title	Assignees	Status
1	Create Users Table #10	axjuenemann	Done
2	Create Profile #13	tjohnson23	Done
3	Make User Image Appear on Navbar After Logged In #43	axjuenemann	Done
4	Invalid User #3	omgfayez	Done
5	Save jokes #6	aneesh27	Done
6	Password hashing #4	lindsey-trussell	Done
7	Register Page #65	tjohnson23	Done
8	User Separation #63	axjuenemann	Done
9	Display Joke(s) #15	aneesh27	Done
10	View saved jokes #7	axjuenemann	Done
11	Authentication Barrier #39	axjuenemann	Done
12	Create Jokes Table #8	axjuenemann	Done
13	Create Users_to_jokes Table #9	axjuenemann	Done
14	Access Profile #14	tjohnson23	Done
15	Create Profile Page EJS #19	Solus5	Done
16	Create login page EJS #32	lindsey-trussell	Done
17	Select Specific Filters #16	omgfayez	Done
18	Add Partials #42	aneesh27	Done
19	Joke APIs #47	axjuenemann	Done
20	Save and remove jokes (Joke display page) #17	aneesh27	Done
21	Update Username #20	Solus5	Done
22	Update Avatar #21	Solus5	Done
23	Update Password #22	Solus5	Done
24	No duplicate jokes #59	axjuenemann	Done
25	Password Authentication for Profile Edits #55	Solus5	Done



Video
Link to the demonstration:
https://drive.google.com/file/d/19ReKxJCUT3ps6K_J11bKxAwuGYIfhJ9S/view?usp=sharing

VCS
Repository link: https://github.com/CSCI3308-Fall22/CSCI-3308_Group_Project_Team_015-01

Contributions

AJ
AJ worked on setting up the database for the website as well as all of the joke API returns. This included working with icanhazdadjoke, YOMOMMA API, Blauge.xyz, Geek Joke API, and the Bread Puns API through axios calls. He also worked on helping with several other small features of the website including creating the user session, making sure jokes are saved and unsaved individually for each user, and fixing miscellaneous bugs.

Aneesh

Aneesh initially worked on setting up the partials. Later, he worked on setting up the display jokes page where the jokes retrieved by APIs were displayed as rows on a table with the ability to save each of the displayed jokes. He also made sure no joke can be saved twice by disabling the button once the joke was saved. The saved jokes were also unique to each account and were stored in the database (the database was tweaked in order to incorporate this). Additionally, he worked on displaying the saved jokes with the ability to remove/unsave certain saved jokes.

Lindsey

Lindsey worked primarily on testing for the register, login, joke generating, and joke saving pages using mocha, chai, and chaihttp. Additionally, they worked on the implementation of the login page, including the insertion of user authentication data into the database, and redirecting to the joke generating page.

Faez

Faez worked mostly on the design such as the buttons, the messages, the modals in profile page and the image size that is in the nav-bar and in the profile. then worked on the rogex for register/login page “validation of the password and the image link”. Worked a little bit on the back-end for the username validation. Helped solus with the profile page and was offering some help over the time. Gave the image Idea to the group and it was accepted later.

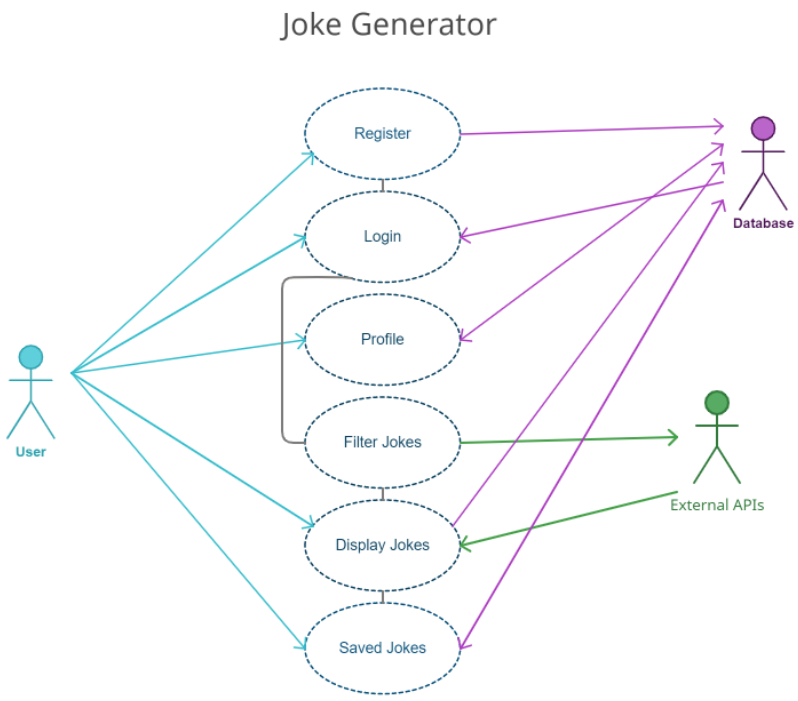
Travis

Travis worked on designing the wireframes for the profile, register, login, and joke generator pages. Additionally, he worked on the implementation of the register page, including the authentication, putting the data in the database, password encryption, and redirecting to the login page after successfully logging in. Lastly, he worked on designing the powerpoint slides, including the content, design, and order of the slides.

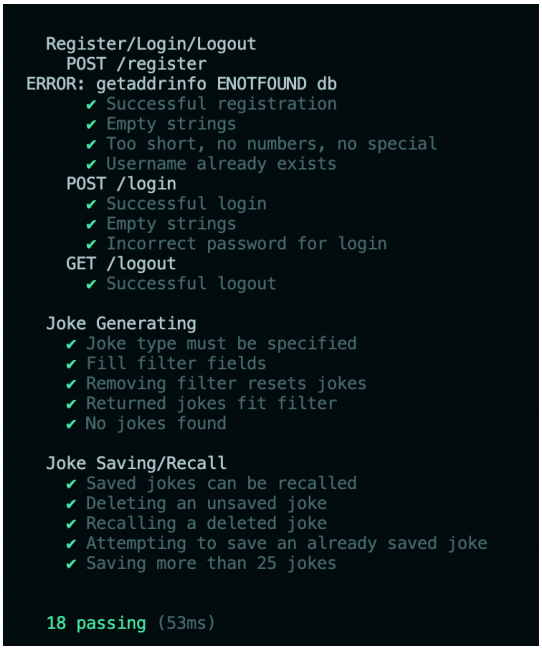
Solus

Solus primarily worked on the profile page, first by writing the .ejs file to display the page, as well as the current user’s most up to date profile information. Solus also wrote the code to connect modals to the javascript file, where they also wrote the APIs to take input, check if the user entered their most up to date password, and if so, update the value of their username, avatar, or password. Additionally, Solus ensured that these profile changes are always reflected right away without needing to refresh. Additionally, Solus assisted with bug fixes and the initial research and selection of which joke APIs to use for joke generation.

Use Case Diagram



Test Results



For unit testing, we used mocha and chai in order to test the ejs and js files. These tests served to make sure that get and post requests were returning the correct statuses and/or errors. There were eighteen tests total, sorted into Register/Login/Logout, Joke Generating, and Joke recall.

These same tests were used for reference when observing user testing. Testers were asked to run through the website as they felt made sense, and ranged from individuals with little to no experience with technology to those with lots of experience. Testing went as expected with testers having no issues navigating the site, apart from errors caused by the APIs themselves going down. Since these errors were caused by external factors, the best fix was to remove the API entirely or find a replacement.

Deployment

Due to issues pertaining to the CU deployment environment provided in class, the application is not deployed (as instructed by TA)