Best Seat

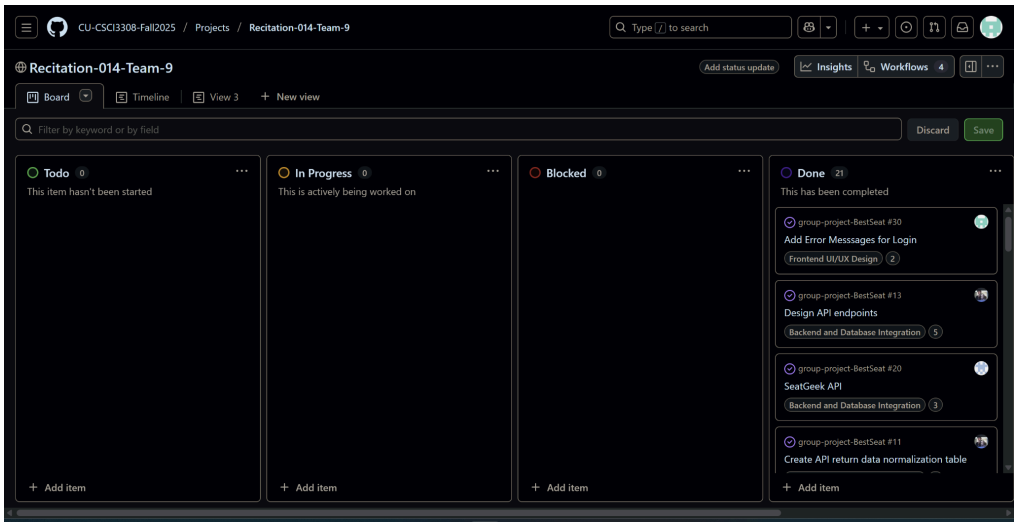
Team Members (GitHub username): Tate Bullinger (TMB101), Eyal Lahat (EyaLahaT), Blake Huhn (ItsAltus), Kyle Behnken (kylelbehnken)

Description:
This application creates a hub for finding the best tickets for concerts or events. Users would sign on, search for the event they are interested in, and see a variety of vendors/websites to find their best ticket.

Users will have their own account, which will have its own login and registration page, respectively. User data will include a hashed password and username/email, which will be stored in a database. Users will be able to change their password anytime they are logged in through the profile page. Users can log out at any time, destroying the session in the profile page

From the login page, Users are able to search and select the event they are searching for using a variety of filters, ranging from date, genre, keyword, and location. Once selected, they will be shown a list of available tickets from different vendors. Links and images of each vendor are present for the selected event, and the user is shown the event's time, location, and name on this comparisons page.

Project Tracker:



Video:

VCS:

Contributions:

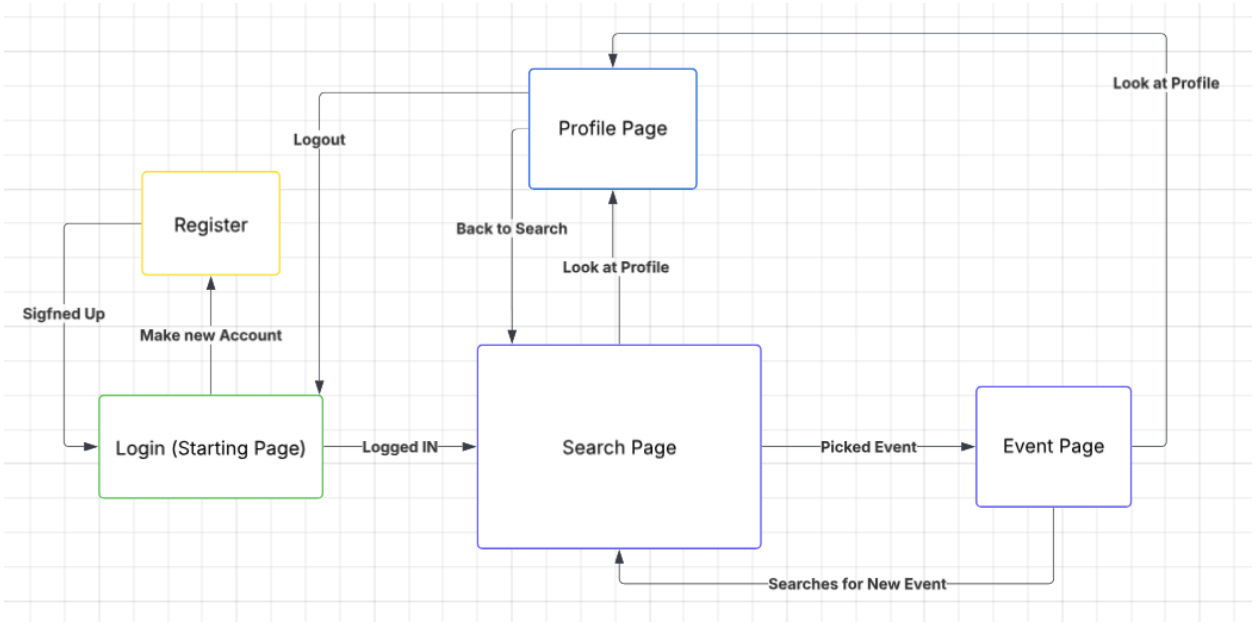
Tate: I implemented most of the front end handle bars files including the login, comparisons, profile, and search. I also integrated the passed in data from the back end to show up properly on the comparisons and search page through the index.js and hbs files. I added features such as the change password, logout, and error messages.

Eyal: I built most of the back end, including the user database, register, login, and search routes using the Ticketmaster API. I also researched the different API possibilities for getting ticket information and built the route for the comparisons page to get the vendor information, using Rapid API.

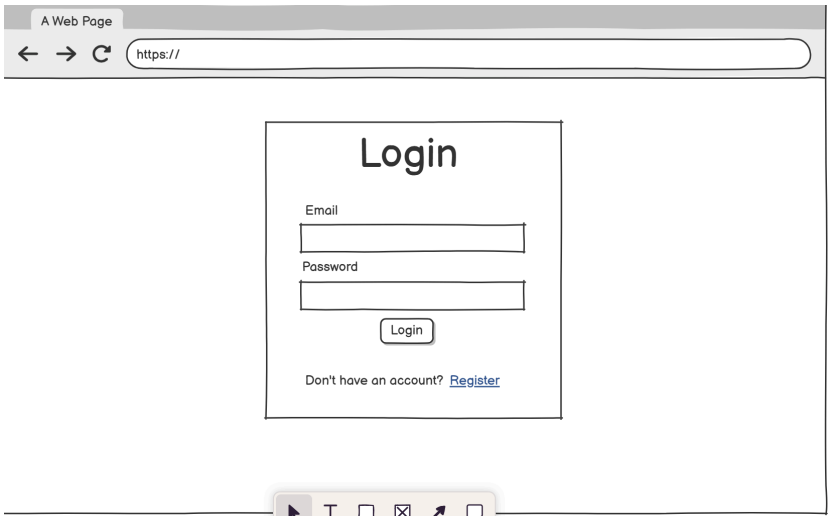
Blake: I worked on the back end and created our custom data structure for storing event details, as well as the search filter functionality, which allows events to be displayed on the search and comparisons pages. I also worked on the template for the search routes before we expanded out from just Ticketmaster.

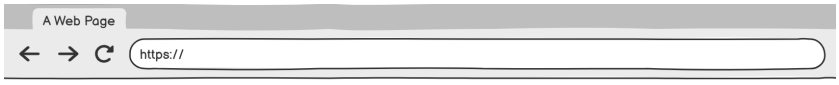
Kyle: I constructed the appearance of our pages and header to accurately match the wireframes designs we drew up. I designed and constructed the search page's sidebar filter. I added functionality to the "save display name" button. I also recorded our website's login system to store the user's information on login (email, account name).

Use Case Diagram:

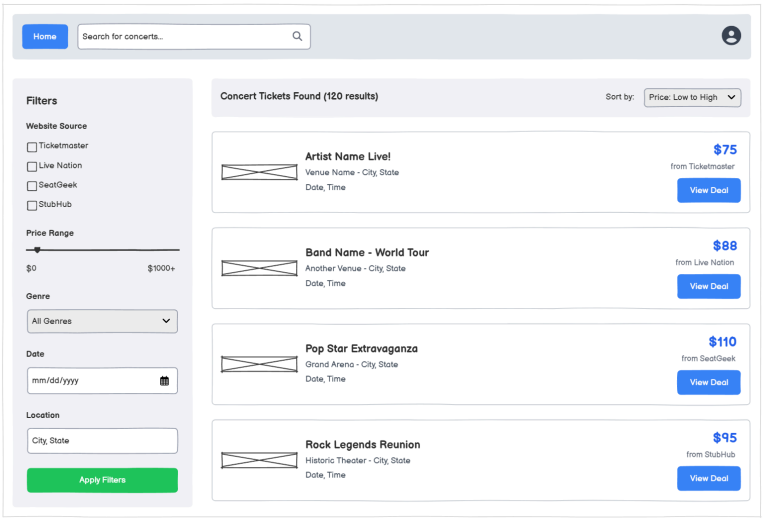


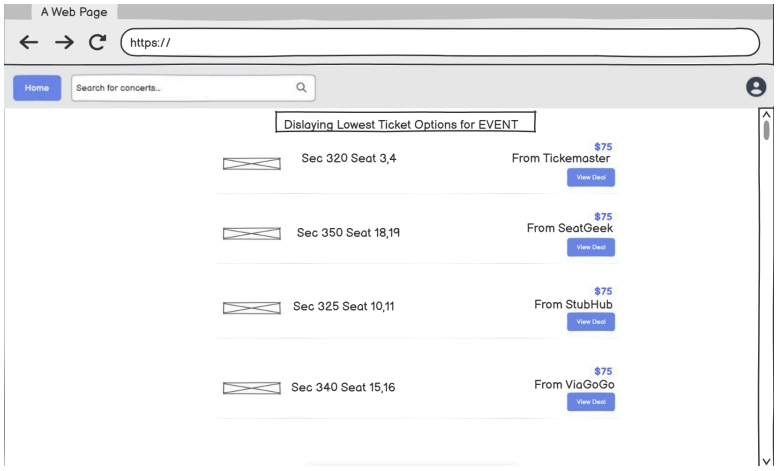
Wireframes:

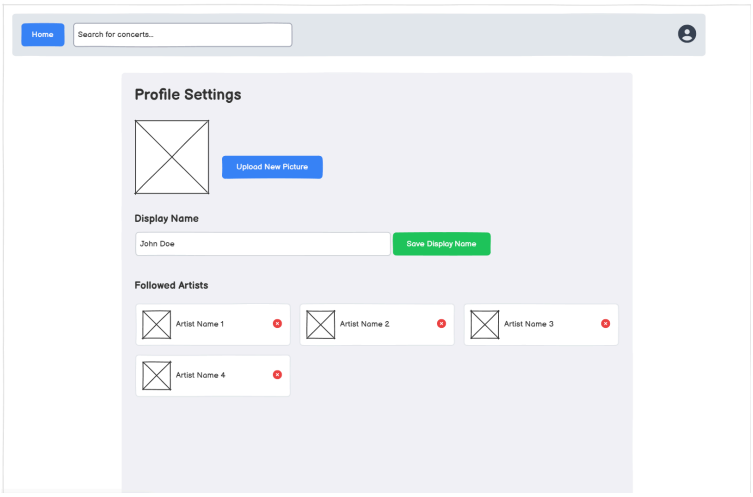

A Web Page
https://
Login
Email
Password
Login
Don't have an account? Register


A Web Page
https://


Register
Email
Password
Register
Already have an account? Login


Home
Search for concerts...
Filters
Website Source
Ticketmaster
Live Nation
SeatGeek
StubHub
Price Range
$0
$1000+
Genre
All Genres
Date
mm/dd/yyyy
Location
City, State
Apply Filters
Concert Tickets Found (120 results)
Sort by:
Price: Low to High
Artist Name Live!
Venue Name - City, State
Date, Time
$75
from Ticketmaster
View Deal
Band Name - World Tour
Another Venue - City, State
Date, Time
$88
from Live Nation
View Deal
Pop Star Extravaganza
Grand Arena - City, State
Date, Time
$110
from SeatGeek
View Deal
Rock Legends Reunion
Historic Theater - City, State
Date, Time
$95
from StubHub
View Deal





Test Results:

UAT Testing:

Observations:

When the user was interacting with the app, they followed the expected workflow for all three tests. Registering, logging in, and testing the search results all followed the exact actions of the UAT plan. Their reasoning was straightforward as they completed each test using the provided test credentials to ensure that all testing actions were as expected written out in the UAT plan. Their behavior matched the expected use cases since they navigated straight to the right pages, entered the required inputs/actions, and validated the results with little to no confusion or struggle. They had to be reminded of some actions for opening up to the terminal to connect to the db and check that the user was present since they don't have a CS background but all other checks were seamless and had the reasoning of performing all of these tests properly.

When observing these tests we noticed that there were no error messages for if the testing user reached a negative case. Even though we did not test a negative case in this UAT test, we wanted to make sure that information was communicated to the user. So, we added more error messages to the login and register page to help confusion with users.

Deployment: