

# Best Seat

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

## 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:

The screenshot shows a project management interface with a dark theme. At the top, there are navigation links for 'CU-CSCI3308-Fall2025 / Projects / Recitation-014-Team-9'. Below the header, there are buttons for 'Board' (selected), 'Timeline', 'View 3', 'New view', and a search bar. On the right side of the header, there are buttons for 'Add status update', 'Insights', 'Workflows 4', and a three-dot menu. A 'Discard' button is visible near the search bar, and a 'Save' button is at the bottom right of the header area. The main area features four columns: 'Todo' (0 items), 'In Progress' (0 items), 'Blocked' (0 items), and 'Done' (21 items). Each column has a descriptive message below it: 'This item hasn't been started', 'This is actively being worked on', '...', and 'This has been completed'. The 'Done' column contains four cards, each with a circular icon, a title, and a brief description. The titles are: 'group-project-BestSeat #30 Add Error Messages for Login (Frontend UI/UX Design) (2)', 'group-project-BestSeat #13 Design API endpoints (Backend and Database Integration) (5)', 'group-project-BestSeat #20 SeatGeek API (Backend and Database Integration) (3)', and 'group-project-BestSeat #11 Create API return data normalization table (Backend and Database Integration) (1)'. Each card also has a 'Edit' icon and a 'Delete' icon. At the bottom of each column, there is a '+ Add item' button.

## Video:

## VCS:

## **Contributions:**

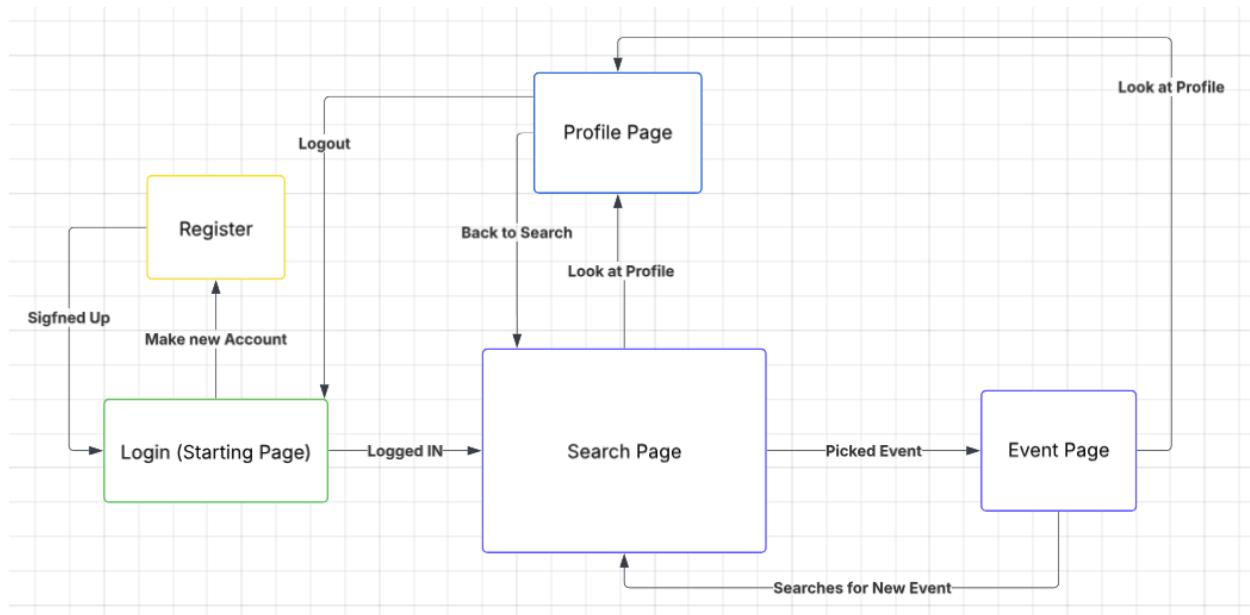
Tate: I implemented most of the front end handle bars files including the login, comparisons, profile, and search. I also integrated the passed in data from the back end to show up properly on the comparisons and search page through the index.js and hbs files. I added features such as the change password,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

Don't have an account? [Register](#)

■ □ T ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

A Web Page

← → ⌂ https://

### Register

Email

Password

Already have an account? [Login](#)

■ □ T ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

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

Concert Tickets Found (120 results)

Sort by: Price: Low to High

Event Name	Venue Name - City, State	Price	Source	Action
Artist Name Live!	Venue Name - City, State	\$75	from Ticketmaster	<input type="button" value="View Deal"/>
Band Name - World Tour	Another Venue - City, State	\$88	from Live Nation	<input type="button" value="View Deal"/>
Pop Star Extravaganza	Grand Arena - City, State	\$110	from SeatGeek	<input type="button" value="View Deal"/>
Rock Legends Reunion	Historic Theater - City, State	\$95	from StubHub	<input type="button" value="View Deal"/>

The top screenshot displays a search results page for concert tickets. The header includes a back arrow, forward arrow, refresh button, and a URL bar showing <https://>. Below the header is a navigation bar with 'Home' and a search bar containing 'Search for concerts...'. A magnifying glass icon is next to the search bar. The main content area is titled 'Displaying Lowest Ticket Options for EVENT'. It lists four ticket options:

- Sec 320 Seat 3,4 From Tickemaster \$75 [View Deal](#)
- Sec 350 Seat 18,19 From SeatGeek \$75 [View Deal](#)
- Sec 325 Seat 10,11 From StubHub \$75 [View Deal](#)
- Sec 340 Seat 15,16 From ViaGoGo \$75 [View Deal](#)

The bottom screenshot shows a 'Profile Settings' page. The header includes a back arrow, forward arrow, refresh button, and a URL bar showing <https://>. Below the header is a navigation bar with 'Home' and a search bar containing 'Search for concerts...'. A user icon is next to the search bar. The main content area is titled 'Profile Settings'. It includes:

- A placeholder for a profile picture with a 'X' icon and a 'Upload New Picture' button.
- A 'Display Name' field containing 'John Doe' with a 'Save Display Name' button.
- A 'Followed Artists' section showing four artist names: 'Artist Name 1', 'Artist Name 2', 'Artist Name 3', and 'Artist Name 4'. Each name is preceded by a checkbox and followed by a red circular icon.

## 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:**