EVENT FINDER

Group Members: Robbie Boccard, Abhiram Kasu, Alex Schwab, Joshua Shih

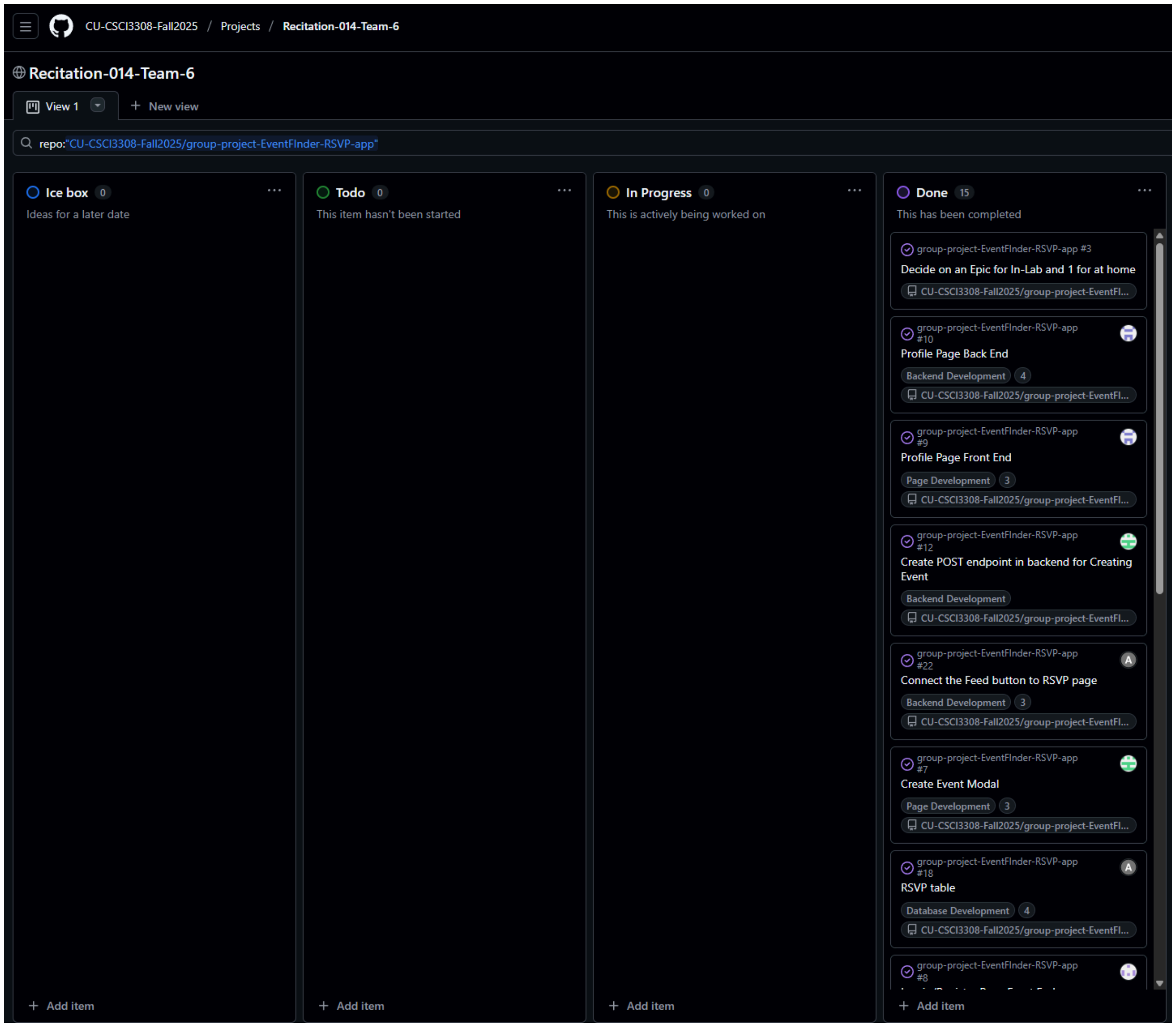
Summary:

Event Finder is a full-stack web application designed to help users easily discover and participate in local events. This app serves as a centralized platform for finding concerts, meetups, community gatherings, and other local activities. Users can register and log in to accounts, browse upcoming events on the feed page, and then RSVP for said events through the app. Each user also has access to a personalized profile page that displays their event attendance history and created events.

Event information is retrieved dynamically through the Ticketmaster API, ensuring up-to-date listings in the database. An optional interface allows individuals to create, edit events as well, allowing for custom events. Events created locally and then will appear in the database after creation. The app allows you to see after creation who RSVP'd for your event and your created events.

Event Finder runs on Node.js, uses Express as its core library, and employs Handlebars for server-side templating. Its data is stored in a PostgreSQL database and hosted on Render. Ultimately, the application aims to allow members of a community to discover and connect through meaningful events and social experiences with ease.

Github Project Board:
https://github.com/orgs/CU-CSCI3308-Fall2025/projects/18



CU-CSCI3308-Fall2025 / Projects / Recitation-014-Team-6

Recitation-014-Team-6

repo:"CU-CSCI3308-Fall2025/group-project-EventFInder-RSVP-app"

Ice box 0 — Ideas for a later date

Todo 0 — This item hasn't been started

In Progress 0 — This is actively being worked on

Done 15 — This has been completed

- group-project-EventFInder-RSVP-app #3 — Decide on an Epic for In-Lab and 1 for at home — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #10 — Profile Page Back End — Backend Development 4 — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #9 — Profile Page Front End — Page Development 3 — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #12 — Create POST endpoint in backend for Creating Event — Backend Development — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #22 — Connect the Feed button to RSVP page — Backend Development 3 — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #7 — Create Event Modal — Page Development 3 — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #18 — RSVP table — Database Development 4 — CU-CSCI3308-Fall2025/group-project-EventFI...
- group-project-EventFInder-RSVP-app #8

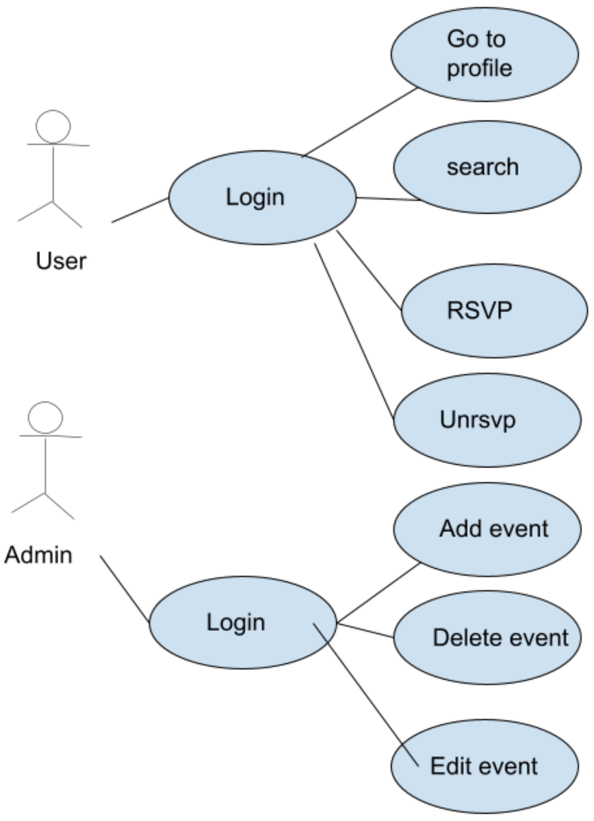
Video demo: [Video_Demo]

Git Repository: https://github.com/CU-CSCI3308-Fall2025/group-project-EventFInder-RSVP-app

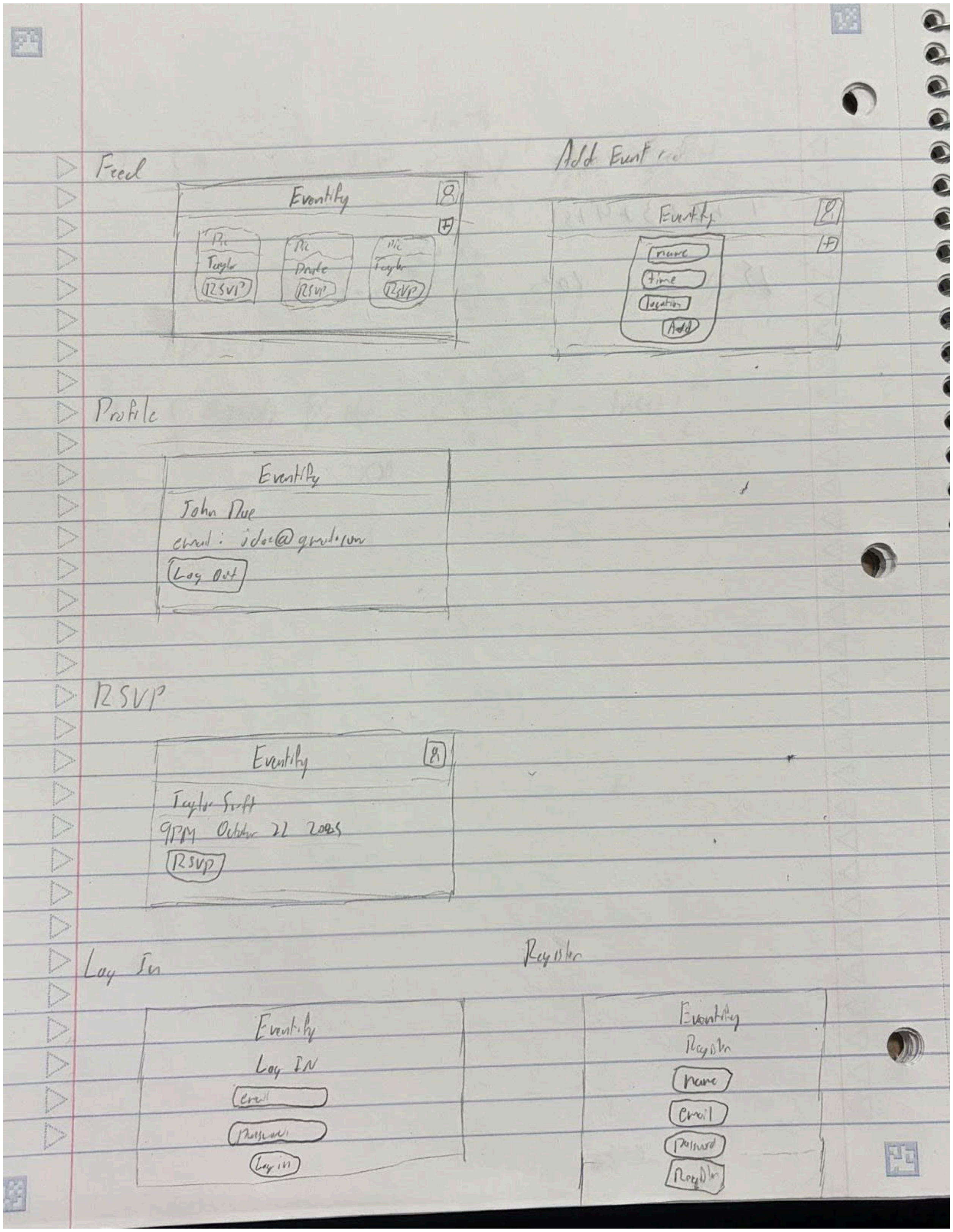
Contributions:

- **Alex:** I mainly worked on the front and back end of the RSVP page. I constructed the webpage design, created functionality with JavaScript, and connected it to the database. Additionally, I also used styling to make the webpage more attractive and consistent with the other pages. I also created the GET/POST for the RSVP page, as well as ensuring the flow of data from the feed page to the RSVP page. I did all of this through a mixture of VSCode and VSCode through CSEL.
- **Robbie:** I worked on the login/register pages, both front end and back end. I used handlebars for consistency with the styling on the front end design, making users have to input their email and confirm their password to successfully register, and then be directed to the log in page. I made the user info be stored in a database, and made get and post routes to the pages as well as testing routes to ensure validity. I did this through VSCode.
- **Josh:** I primarily worked on the profile page, creating a frontend that would display the desired information in an organized and attractive format, as well as the backend that used SQL requests to correctly retrieve the necessary information (name, email, RSVP'd events, and created events). I then secured the page and ensured that the user must be logged in to access it, otherwise they would be redirected to the login page. Additionally, I created a standardized navigation bar with Handlebars to be used for all pages in the site. I did this through VSCode.
- **Abhiram:** I mainly worked on the feed page and backend with ensuring the schema of the database would work for our purposes. I also integrated the Ticketmaster api with the application allowing for events to be pulled from both the local db and the Ticketmaster api by standardizing the data between them. I also created the modal for creating events and the backend api for actually pushing the new event to the database. I also had to make sure that all the buttons from the events on the feed page lead to the correct routes. I also implemented the cookie based authentication across the backend and site making sure that all the routes except for login and register were secure. I also added extensive testing for all the features that I added. I did all of this through VSCode locally on my laptop.

Use Case Diagram:



Wireframes:



Feed
Eventify
Pic
Taylor
RSVP
Pic
Drake
RSVP
Pic
Taylor
RSVP
Add Event
Eventify
name
time
location
Add
Profile
Eventify
John Doe
email: jdoe@gmail.com
Log Out
RSVP
Eventify
Taylor Swift
9PM October 22 2025
RSVP
Log In
Eventify
Log IN
email
password
Log in
Register
Eventify
Register
name
email
password
Register

Tests:

Use Case 1: Viewing Events to RSVP

- Description: The user logs in and navigates to the "Events" page. The system fetches event data from both the Ticketmaster API and the custom_events database table. The user should only see the event list after logging in.
- Expected Result: The event list correctly displays once the user is authenticated. Unauthenticated users cannot access the event list.

Use Case 2: RSVP Functionality

- Description: The user selects an event from the list and submits an RSVP. The system verifies the user's login status and ensures the user has not already RSVPed. Upon success, RSVP information is recorded in the user_rsvp database table.
- Expected Result: The RSVP completes successfully for a logged-in user. Repeated RSVP attempts for the same event are denied.

Use Case 3: Creating an Event

- Description: The user accesses the "Create Event" page after logging in. The system validates that all mandatory fields (event name, description, location, start time, and end time) are filled before submission.
- Expected Result: The event is successfully created and stored in the custom_events table when all fields are valid. Unauthenticated users or incomplete forms are rejected.

Use Case 4: Access Restriction for Unauthenticated Users

- Description: A user attempts to view the event list or RSVP without logging in.
- Expected Result: The system prevents access and redirects the user to the login page.

Test Observations + Results:

User Actions:
Participants navigated the interface intuitively. They first attempted to view events while logged out, followed login prompts, and then successfully accessed event data. During the RSVP and event creation steps, they followed logical, step-by-step paths without external guidance.

User Reasoning:
Users interpreted the "Login required" message as a security and access control measure. When creating events, they recognized mandatory field indicators and filled them before submission. This shows a clear understanding of expected workflows.

Behavior Consistency:
User behavior aligned closely with the expected paths in each use case. Each participant

understood that login was a prerequisite for all event-related actions and used the feature as intended.

Deviations:
No notable deviations occurred since all test cases passed. However, one user initially tried to RSVP twice for the same event, confirming that the duplicate prevention logic worked as intended, triggering an appropriate validation message.

Application Adjustments:
No functional changes were required because all test cases succeeded. However, based on user comments, minor usability enhancements could be made:

- Provide a confirmation dialog before submitting an event creation form.

Deployment: https://group-project-eventfinder-rsvp-app.onrender.com/