University of Colorado Boulder

CSCI 3308 Project Report:

EventEdu

William, Axel, Alen, Luke, Nabil, & Jacob

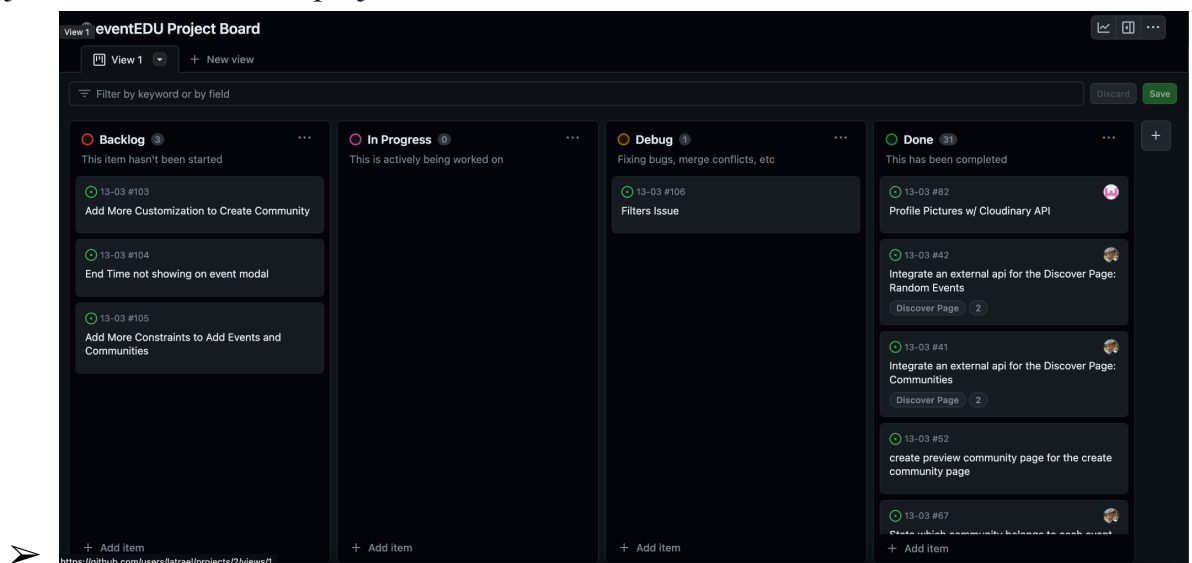Software Development and Methods

Sreesha Nath

Dec 11, 2023

❖ Project Description
  ➢ For this project, our goal was to create an application that includes the following features: a login page, registration page, home page, A server that connects the database and UI, hashed passwords, and session management, and the application had to be built with Docker containers. Our idea was to create a college-based community event tracker. Users would be able to join and create communities that pique their interests. Then They could view or create events that communities have and add friends to see what communities and events they are a part of. There is a Discover page that allows users to search for communities and events. There are multiple ways to filter and search for these, so the user can find what they want. There is a home page that displays all of the user's communities that they are a part of as well as the events that they are attending. There is a Create a Community page that allows users to create their own community. This would give them access to the create event page for their community where they can set up events that other users could join. Finally, there is a profile page where users can view their general information.
❖ Project Tracker - GitHub project board:
  ➢

  ➢ Link: https://github.com/users/latrael/projects/2/views/1

❖ Github Repository:
  ➢ https://github.com/latrael/13-03.git
❖ Video
  ➢ ▶ EventEdu Demo
  ➢ https://www.youtube.com/watch?v=u78yB7r7DYU

- ❖ Contributions:
  - ➢ Jacob
    - ■ Everything on the Friends page was done by me. I implemented a friend request functionality, you can search users on the friends page based on their username. There is a list of all the communities your friends are in. All of your friends are also listed, if you have friend requests they will be at the top of your list, then your current friends, beneath that will be friend requests you sent, and then suggested users at the very bottom. There is a calendar of the events your friends are in. You can view the events and join or leave the event. I then added the same calendar to the community page, along with the ability to leave the community. Finally I distinguished between joining/leaving an event or community on the discover page
  - ➢ Luke
    - ■ Login, Register, Logout, and Account Information pages. Users can create profiles with secure passwords hashed using Bcrypt. The profile page gives users a personalized view of their connections on the website: personal info, their communities and events, and their friends. I also added the ability to view friends' profiles like Facebook. Front end utilized HTML and BootStrap, backend utilized JavaScript, EJS and PostgreSQL. I used an external API (Cloudinary) to implement profile pictures (user-uploaded and stored in the cloud), then rendered on profile/friend profile pages. I also helped debug issues with the Discover page and modify our database structure.

➢ Axel
- The Create Community page. This page allowed for the creation of communities that users could join. Also added the preview community feature in that page. The database tables. This is where all the variable information was stored. The header, footer and menu ejs files. The add event page/functionality, this allowed users to create events for their communities if they were the ones to create said community. CSS.

➢ Nabil
- The home page with calendar UI. We envisioned the home page as somewhere the user can glance at important information about their events communities, combined in one place. I used a javascript framework to create and display a responsive calendar that can switch between months, and display events with times. The events are pulled through the API that is called when the home page is

loaded, and all the events that the user specifically joined are displayed. Also revamped the events database to be able to correctly display event information in a clean format.

> Alen
  - The Home page. This page displays all the users' communities and allows them to visit the specific community page they want to visit and if the user is not a part of any communities it prompts them to join or create a community. Alongside the homepage I worked on the community pages where it displays the members of the community page and the description of the community alongside the calendar for that community. Also helped debug issues of the calendar not displaying events properly on the home page.
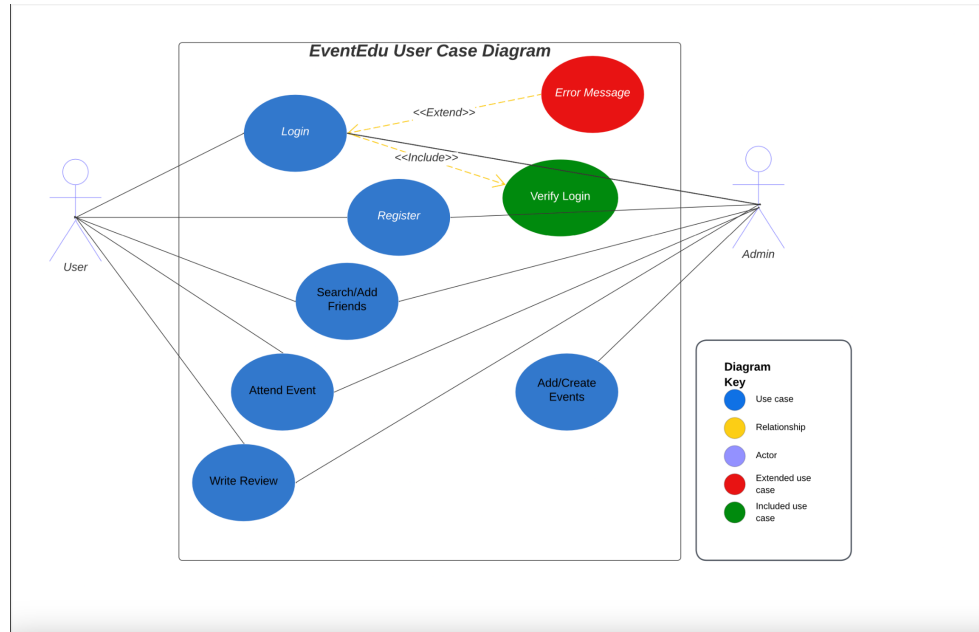
> William
  - The Discover Page. This page allows the users to search the database of both communities and events, where if they click a button, they join the community/event. Along with this, there are a bunch of functionalities that go along with it like searching by name or searching communities by filter.
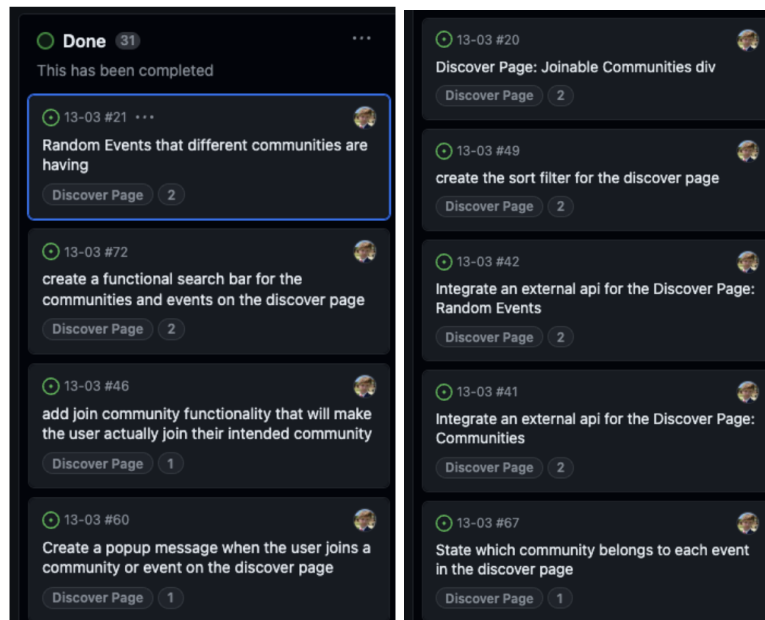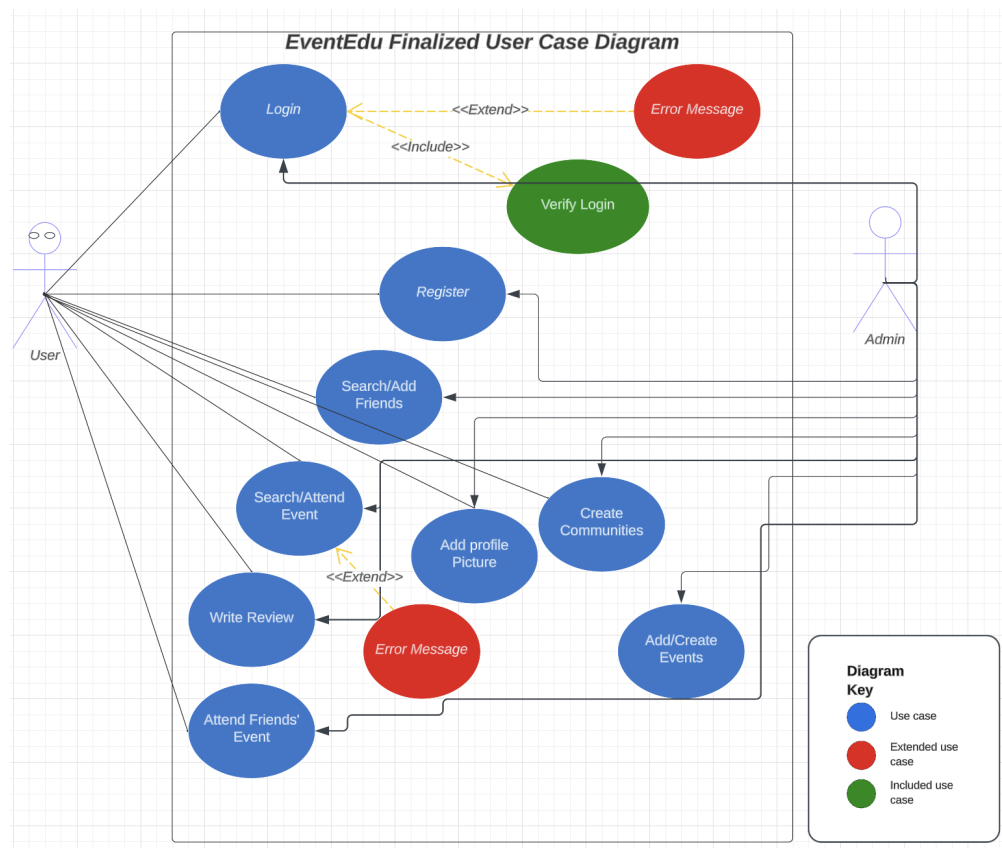

❖ Use Case Diagram:
  > Old Use Case Diagram:

➢ New Use Case Diagram:

❖ Test results:

We had three test plans that were carried out. They were "user joining a community", "user creating a community", and "user creating an event". The results of each are listed below:

"User joining a community":

There were two test cases for this plan:

1) User attempts to join a community they are not a part of
   a) As stated in the plan, a new user was created, that wasn't a part of any communities. The user successfully passed this test, as they joined a community, and was added to the list of members.
2) User attempts to join a community they are already a part of
   a) This test case couldn't be tested, as once the user went back to the discover page, the "join" button was replaced with a "leave" button on the community they were a part of. We consider this a pass.

"User creating a community":

There were two test cases for this plan:

1) User attempts to create a community with valid information
   a) The user entered valid information for the creation of a community, and it was successfully created and listed in the Discover page, and the user that created the community automatically joined that community. This test case passed.
2) User attempts to create a community with invalid field information
   a) The user attempted to create a community with a blank community name, and once the "Create Community" button was pressed, a div popped up prompting the user to fill in the blank field. The same thing happened when the description of the community was left blank. This test case passed.

"User creating an event":

There were four test cases for this plan:

1) User attempts to create an event for a community that they own
   a) The user pressed "New event" on their communities, and entered valid information for the event. Once they pressed "Create event", the event was created, successfully tied to the community, and the user automatically joined their own event. This test case passed.
2) User attempts to create an event for a community they are an "admin" of
   a) This test case is not applicable, as due to time constraints an "admin" functionality wasn't implemented.
3) User attempts to create an event for a community they are a member of

a) The user went to the community page of the community they are a member of, and there was no button available to create an event. This test case passed.
4) User attempts to create an event for a community as an admin or owner with invalid information
   a) The user went to the community page as the owner of the community(admin functionality not implemented), and pressed create event. The user attempted to create an event with a blank title, which the website prevented with a div popup alerting the user to fill the empty field. This happened for every field that was left empty. This test case passed.

❖ Deployment:
  ➢ Link: http://recitation-13-team-03.eastus.cloudapp.azure.com:3000/