Live

A project by Team 105-4

Colton O'Connor, Ryan Jones, Haifeng Jiang, Reid Pritchard, and Cade Gorman

Description

We aimed to create a web service with the goal of allowing individuals easy access to information about current live events that are happening in the their area. People's taste differ so we would allow individuals to select interest that apply to the, such as music, comedy, politics, et cetera. Given this information our software will be able to suggest the perfect event for a night out, curated to the user.

Tools We used

Node Js

Rating: ★★★★★ (5/5)

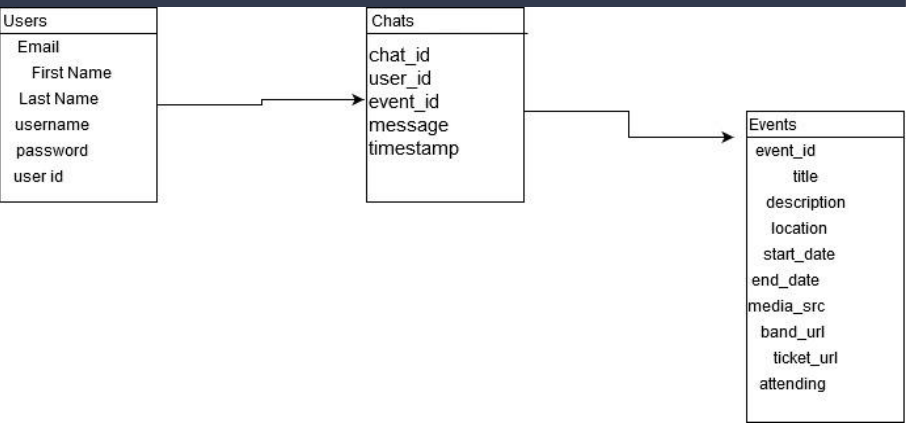
Purpose: The main tool for input and output operations that occured between the frontend and the backend.

Additional Node.js Libraries:

- Bcrypt (5/5)
- Express (5/5)
- BodyParser (5/5)
- Express-sessions (5/5)

All additional libraries provided useful functionality with easy setups

PostgreSQL

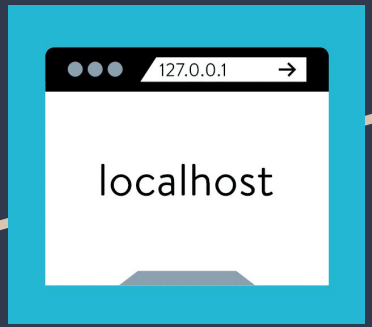


Rating: ★★★★☆ (4/5)

Purpose: Storing all user and event data and serving that data to rendered pages.

Thoughts: Slight challenge getting it setup consistently across multiple systems. But, easy to query and nice support for array types.

Local Host



Rating: ★★★★★ (5/5)

Purpose: Testing and development

Thoughts: Lots of control over whole the full application. Though if we were to switch to a publicly hosted site we may run into issues we weren't able to identify locally.

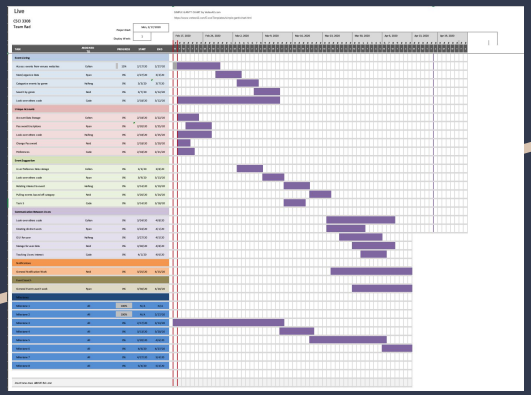
GitHub

Rating: ★★★★☆ (4/5)

Purpose: Managing code base.

Thoughts: Slight learning curve and occasional difficulties merging.

Gantt Chart



Rating: ★★☆☆☆ (2/5)

Purpose: Organization and Task Management

Thoughts: Though a great to allow for early planning and deadlines, with campus closer, irregular scheduled meetings, and project changes, it became difficult to remain on the planned schedule.

Methodologies

Methodology: Agile

Description: A methodology that allows for change over the projects creation and deployment, leaning heavily on working code rather than documentation.

Why chosen: With our project, had a large number of ideas and features we wanted to incorporate. Using agile we were able to work together to add these as we went, allowing our project to adapt.

Expected Tools

Javascript Test Framework

Expected Tool: Mocha

Rating: ★★★☆☆ (3/5)

Purpose: Automated testing

Thoughts: While useful, the tests had to be written after the functions was already working. A more TDD style would have worked better for our team.

Project Tracker

Expected Tool: Asana

Rating: ★★☆☆☆ (2/5)

Purpose: Project Tracking

Thoughts: We would want an easy way to communicate and keep track of all the elements needed to produce a working product. GitHub is a good means to hold all the software elements, but products such as Asana helps with scheduling and communication on a project.

Auto-Documenter

Expected Tool: Doxygen

Rating: ★★☆☆☆ (2/5)

Purpose: Project Documentation

Thoughts: Similar to the project tracker, documentation is important to keep a backlog on all elements of the building of software. It is important to keep record on how the projected is adapting from its original build, especially with an agile methodology.

Miscellaneous Tools for Completion and Maintenance

VCS Repository: GitHub - Great way to allow access to all.

Data Base: PostgreSQL - Already established in our project

Testing Tool: PyUnit - Great way to establish unit tests for a project

Deployment Environment: LocalHost/Heroku - As of now the project is hosted locally, but in the future we might look to have it remotely hosted.

Hardware: Server - With the event and user data, we will need to store this on a server for the software to run properly

Project Challenges

COVID-19 and Campus Closure

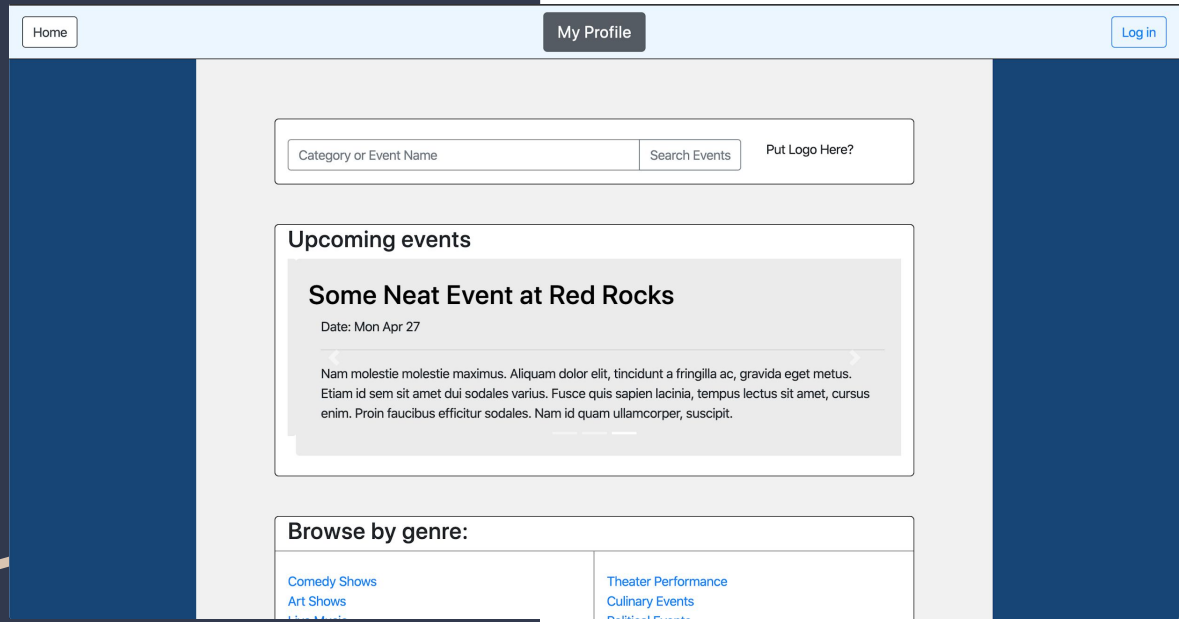
Due to the state of the world, we were no longer allowed us to meet in person and this unfortunately changed the schedule for the project. Communication became much more difficult. In order to fix this we had to rely on github, canvas, and zoom to maintain our communication. We would host meetings on zoom and all work together to reach our final project. This allowed for anyone to ask questions as well as confirming which project aspect was being worked on. Though with our schedules needing rearrangement, we did have to drop some stretch goals of our project, such as communication between users.

Migration from HTML to .ejs

As we began to work more on our database and the implementation of shared information, we realized that we need a better way to connect the from and back end of our project. Up to this point our front end was entirely in HTML, so we had no way to display our event data that was stored in a SQL database. So we decided to convert from .html to .ejs files and use Node JS to connect the front and back. This took quite a bit of time, but in the end left us with a much more fluid design.

A Look at Our Project

Landing Page



Profile Page

Home | My Profile | Log in

A Short Discription about User

Input Interests

My Favorite Events

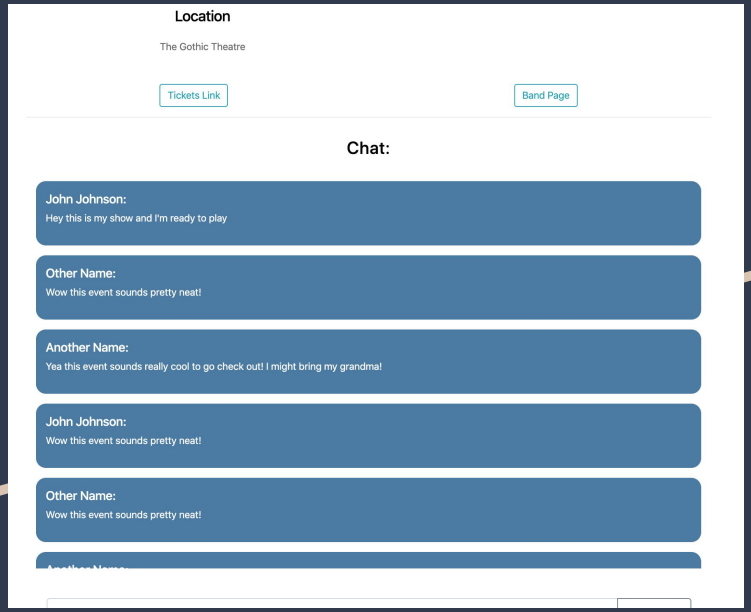
#	Name of Event	Genre	Date
1	Example name	Example Genere	Example Date
2	Example name	Example Genere	Example Date
3	Example name	Example Genere	Example Date

Go to My Events

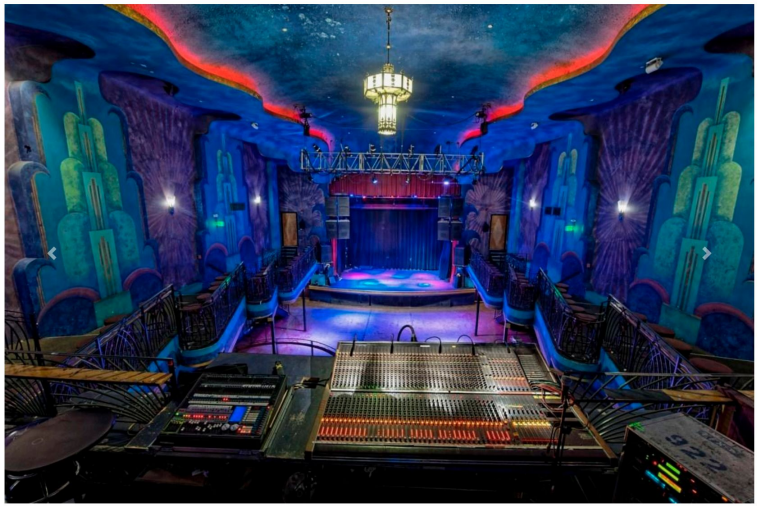
Suggest Event

Suggest Event

Event Page



John Johnson @ the Gothic Theatre



Start Date:

Description: