Project Title: Online Event Management System

1. Introduction

Briefly introduce the project, outlining its purpose and objectives.

Our Project was to create a website with an interface that allowed users to be created and registered. The website was one where a user could create and plan an event. Users would also be able to view events that have been created and registered. For testing please see Appendix 1 for how to set up the database

2. Project Team

List the names of the team members and their respective roles and responsibilities in the project.

Our team is comprised of 3 members and they are Eric Vo, Thomas Longwell, and Gwendelen Cady. Eric and Thomas's roles and responsibilities were front and back-end coding. Gwendelen's responsibilities were front-end development.

3. Technologies Used

Detail the technologies and tools used in this project, including PHP, JavaScript, CSS, MySQL, Git, etc.

Some of the technologies we used were languages such as PHP, MySQL, CSS, and HTML. The IDE's used with these languages were VS code and PHPStorm. Along with the IDE's and the coding languages used, were Github to store our code, Trello to organize our tasks, and Discord to communicate with one another. Resources that were used with and for our code were Bootstrap, W3 School, StackOverflow, and YouTube for troubleshooting.

4. System Overview

Provide a high-level overview of the Online Event Management System, explaining its main features and functionalities.

Our online event management system allows users to create a profile on our website which will allow them to create events and allows other users to join those events. The

welcome page is the base for all of the functions of the website. It allows users to reset their password or log out as well as allows them to create and view events. The main page is the launching point into our smaller subpages such as our event list page which is a larger more detailed list of each event on the website, this page leads to multiple other pages such as the registration page where users can register to join events or the event information page which allows viewers to see all of the details for events. Due to time constraints, we were unable to implement this function but another proposed function for this page would allow event creators to assign admins as well as access an edit page for their events and manage users who registered for their event either approving or denying their request.

5. Implementation Details

This section should be split into different subsections, each explaining how you implemented different aspects of the system:

5.1 User Authentication and Authorization: Discuss the implementation of user registration, login, and roles.

We implemented user registration, login, and roles by having a create account page on our website, this allows users to create accounts save their information into our database, and allowing them to sign in later which allows their user ID to be tied to actions such as creating or registering for events. We would implement roles by assigning participants to everyone when they create an account then if they create an event they will be assigned the event creator role tied to that event ID allowing them to assign admins for that event as well as edit the event and decide whos registration is accepted or denied.

5.2 Event Creation and Management: Explain how events are created, edited, deleted, and managed.

To create events we have a create event page which allows users to fill in the information for an event and save it to our database this event is then displayed on our event list page, which uses PHP to fill our table with the event information, and other users can register for the event from this page. Events would be edited deleted or managed from this event list page and those options would only be displayed to the

event creator or their selected admins. These functions would also be available from the event information page for the admins or event creators for the event listed.

5.3 Event Registration and Participation: Describe the registration process and how participants interact with events.

The registration process for our website is located on our event list page or event information page for specific events selected from the event list. This takes users to a page that allows them to register for the event, their response is then saved in the registration table in the database that ties their user ID with the event ID allowing us to see who registered for which specific event.

5.4 Administrative Functions: Explain how admins can manage users and events.

Admins could manage users and events by going to the event list page and selecting the options which would then take them to the page to edit the event which would use PHP and the event creation page to allow them to see what's saved in the database about the event then they could edit that data and save it and it would use PHP to update the information for that event ID. they can manage users using the same functions by seeing which users are requesting to be registered to the event and they'll be displayed with an admin approval drop-down menu allowing the admin to approve or decline the user's request which would be added to the database.

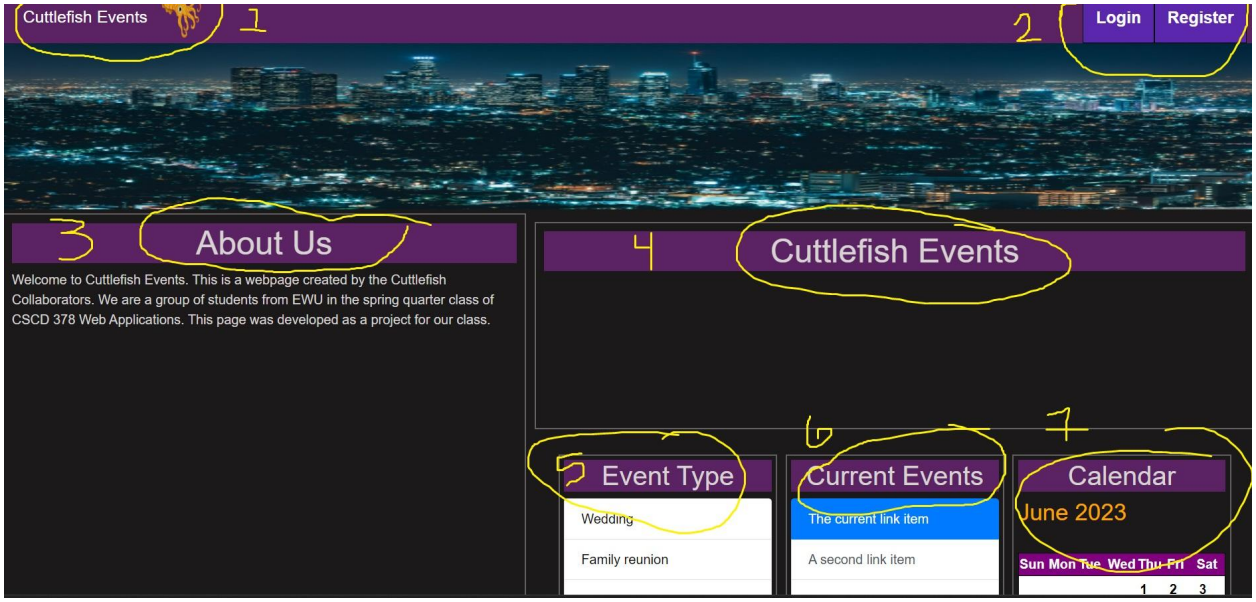
5.5 Cookies and Logging: Detail how you implemented cookies and logging to a database.

Our database uses JavaScript to save users' login information into a temporary cookie allowing them to stay logged in to the website even after they close it, this cookie will be saved for a temporary time then expire and need to be redone by a new sign in with their password in order to have security and require a sign in every so often making sure the user is authenticated.

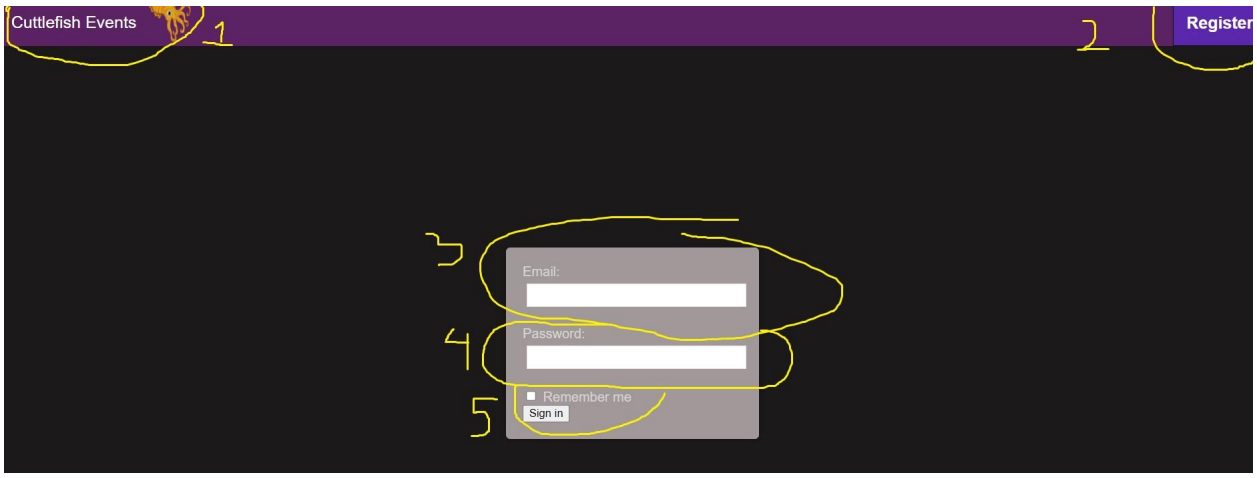
Include snippets of important code blocks and screenshots where necessary to illustrate your explanations.

6. User Interface (Gwen)

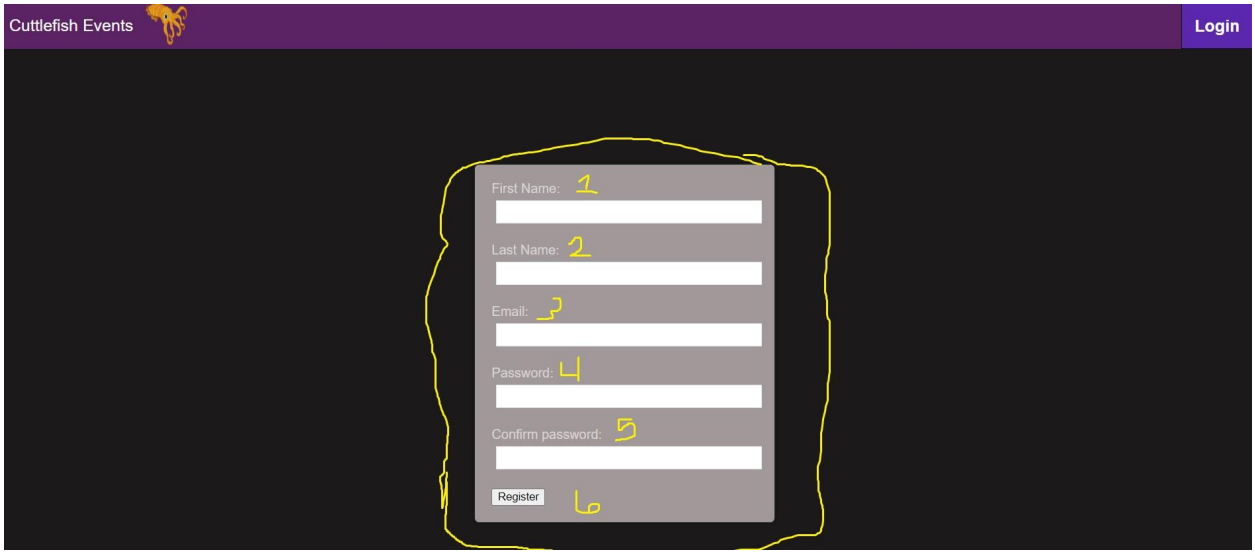
Include screenshots of the application's user interface, highlighting key features and views.



1. Top left corner: The title of our webpage. This links to the home page.
2. Top right corner: The login button allows users to login to the website
3. Middle left: This is where the user can read about the webpage
4. Middle Right: This is the title of the webpage. This box is for adding pictures from events.
5. This is a list of the event types.
6. Current Events is a spot where the user can select events that are upcoming.
7. The calendar: This is a calendar for the user to view.

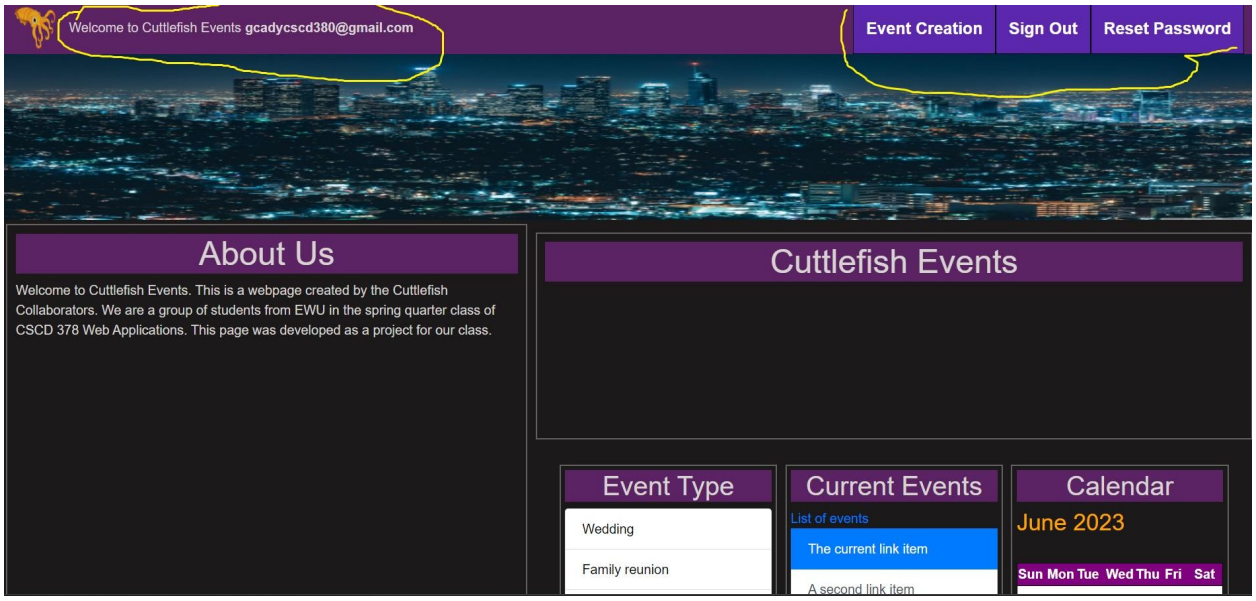


1. Top Left: Title of our website on the login page.
2. Button to register user.
3. Email for logging in.
4. Password for logging in.



What you see above is the registration page.

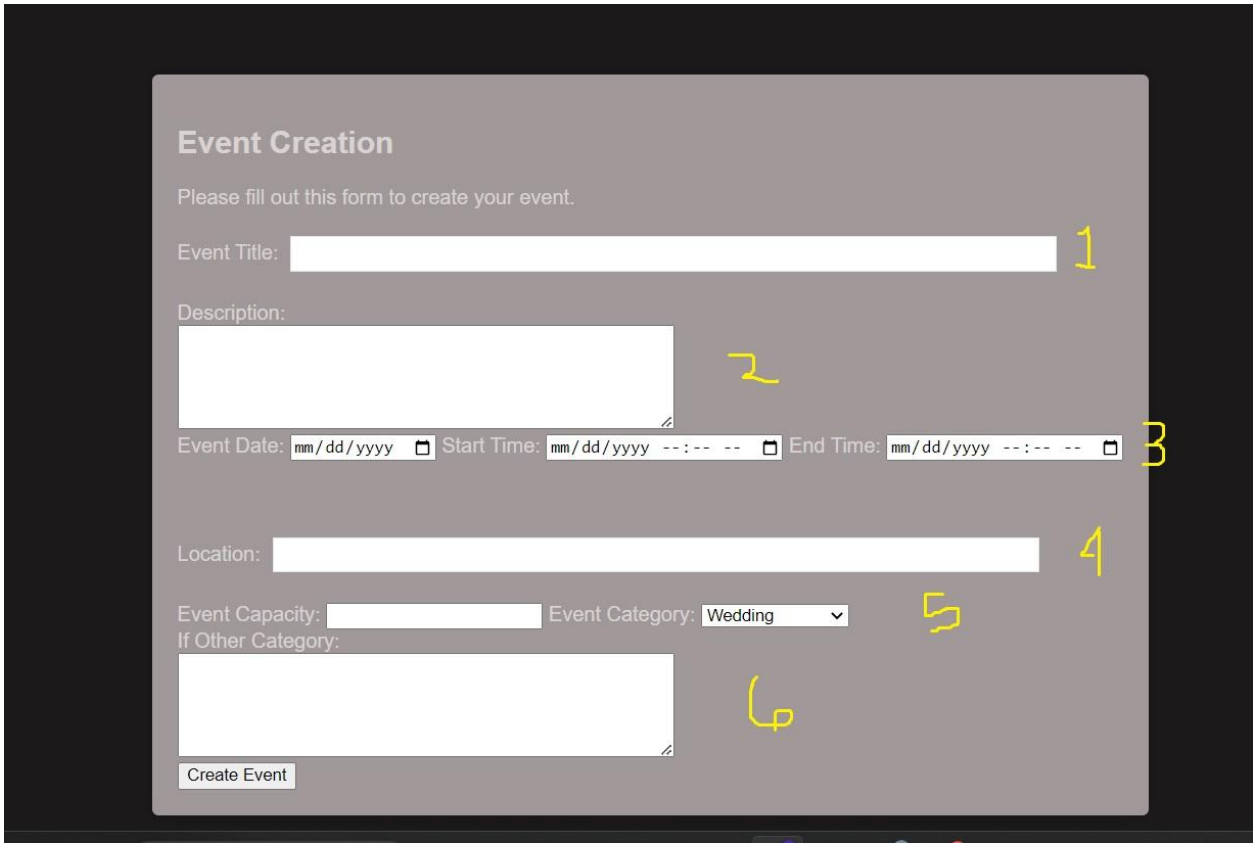
1. User’s first name.
2. User’s last name.
3. The User’s email address.
4. Password.
5. User retypes their password to confirm it.
6. The registration button is the submit button that will save the user's information in the database.



This is the welcome page. This page is very similar to the home page as it contains all the features(1, 3-7) that were mentioned before. The only difference is that the page

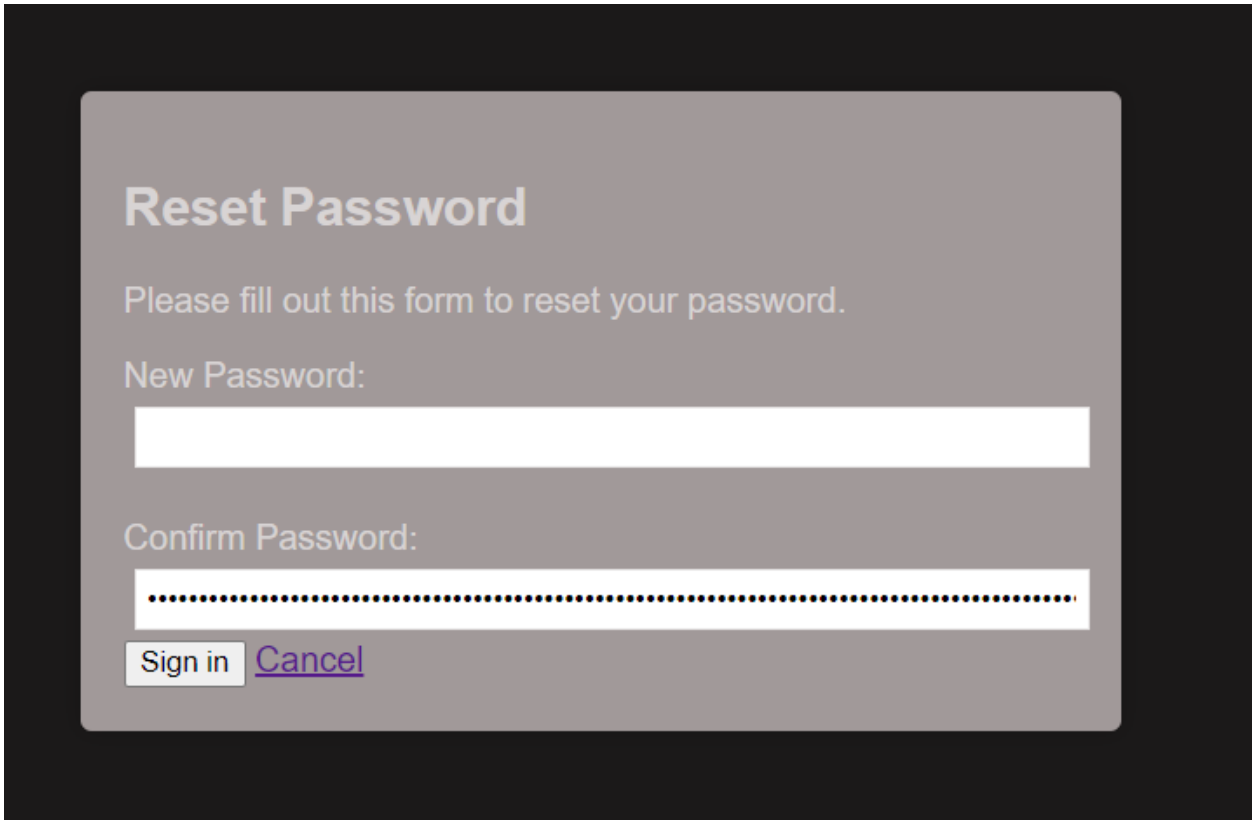
says Welcome to Cuttlefish Events user email (Top Left). In the top right corner are three buttons. From left to right, they are as follows:

1. Event creation: This brings the user to the event creation page where they can create an event that will be saved into the database.
2. Signout: This signs the user out of their account.
3. Reset password: this brings the user to the reset password page where they can change their password.



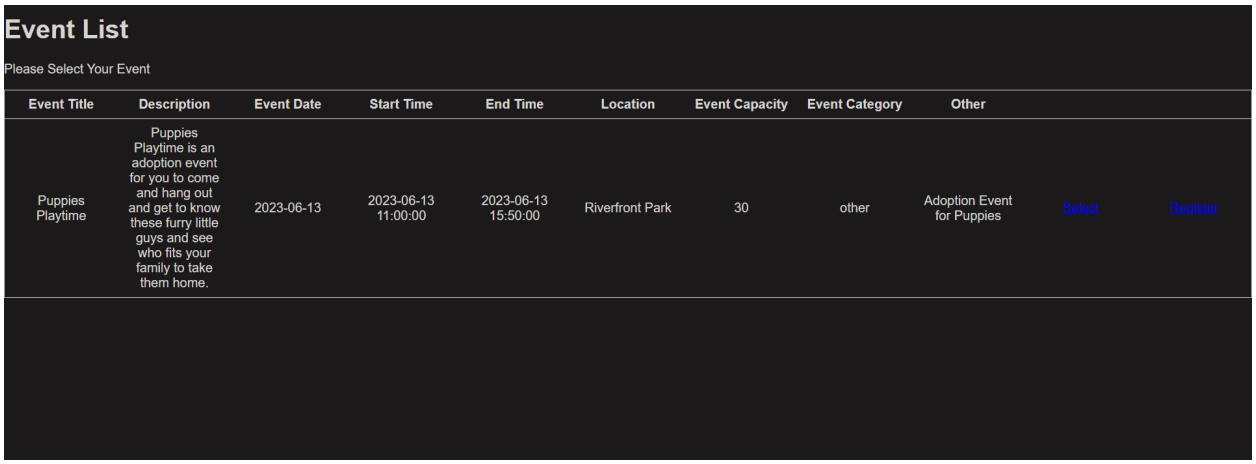
This is the event creation page. On this page the user can create their events.

1. This is where the user can put the event name/title.
2. This is where the user types in the description of the event.
3. The event date can be selected on this row (left). The start time can be selected here (middle) and the end date can also be selected(right).
4. The location of the event can be input on this row.
5. Event capacity can be selected and the event can be categorized.
6. If the category is other you can further give a label/identifier to the event type.



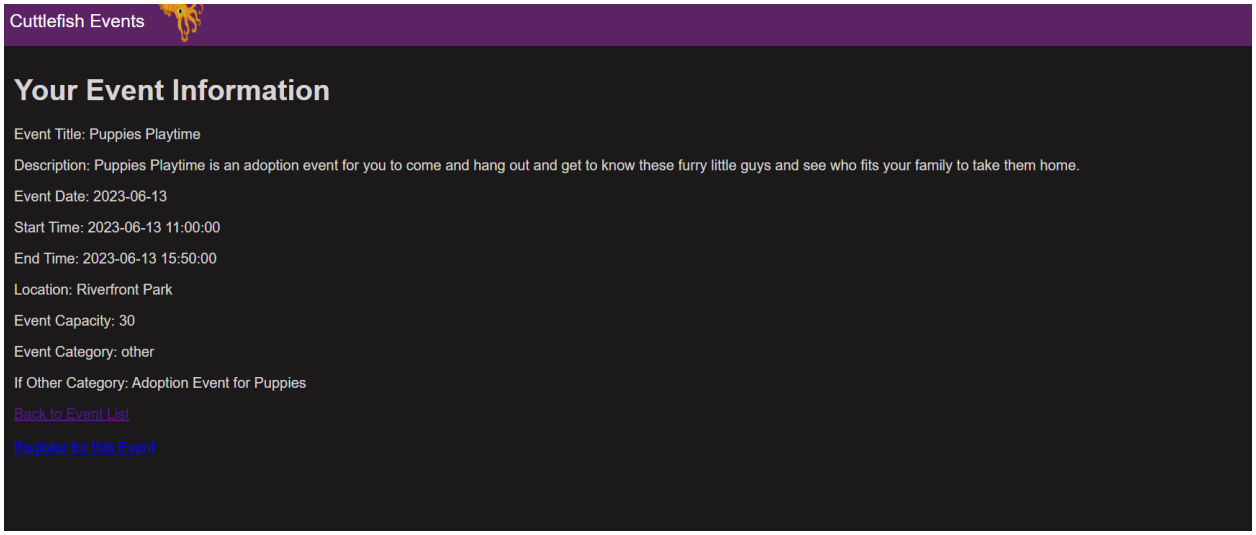
When the reset password is selected on the main page this is what you will see.

The new password will be input in the first input box. Then you will need to re-enter the password in the confirm password input box. The sign-in button is the submit button.

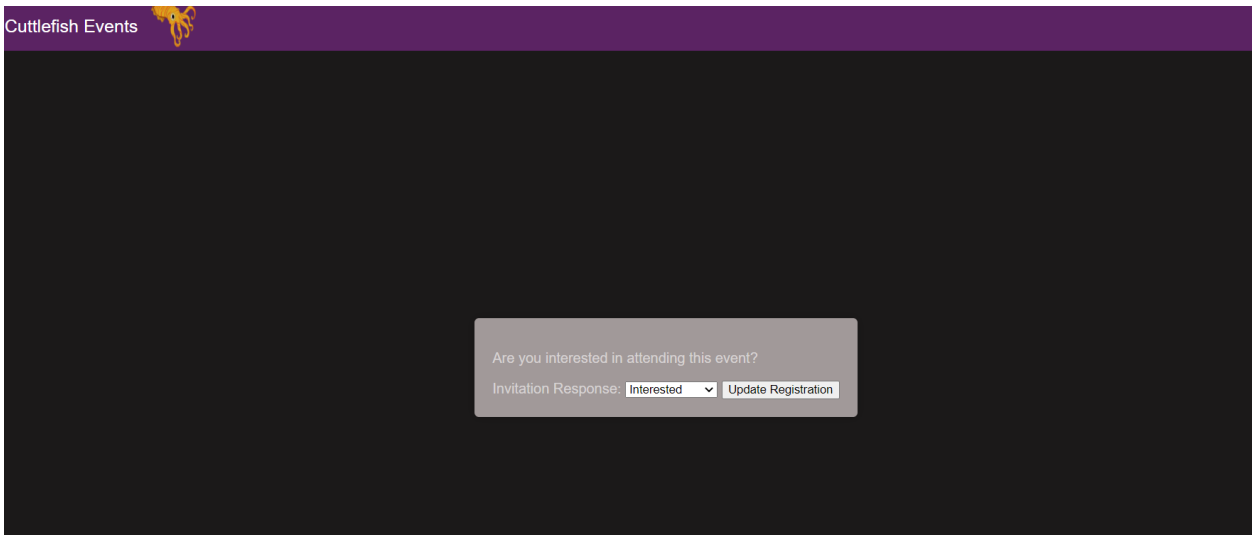


When the event is created, the details of the event will be displayed in the event list page. From left to right the the display shows the event title, description of the event, the event date, event start time, and then end time. The Location is displayed next with the capacity following that as well as the event category. If the category is other, then in the

other column will be filled with the type you labeled the event. The button select brings you to the event information page where you can register. The register button registers you for that event. The information page is displayed below:



When you click to register you will be redirected to this page where you will see the following prompt displayed below:



In this prompt, the user can select if they are interested and when they hit the update registration button it registers the user for that event.

5. Remember me box allows the site to remember the user's login information and the sign-in button below allows users to submit their information to log into their account.

7. Challenges and Solutions

Describe any challenges you faced during the project and how you solved them. Reflect on what you learned from overcoming these challenges.

Some challenges we faced during this project were learning how to use the new languages that were very foreign to us and implementing the admin functionality. We understood how to implement admins but realized we wouldn't be able to implement admin and make sure everything else worked so we decided to focus on refining what we had and implementing the finishing touches rather than gambling on getting admin functions to work in the remaining time, knowing we wouldn't have a lot of time for troubleshooting after implementation. Learning the languages proved to be a challenge for us as well because we read the zybooks and followed them but when it came to implementing the concepts for our own project it was a level above what the zybooks provided and required us to do a lot of googling and searching to learn the concepts for implementation.

8. Future Improvements

Discuss any future improvements or features you would like to add to the system, considering the current scope of the project and feedback from users or peers.

Future improvements that can be applied to the project can be a search bar on the main page. Other things that could be implemented to improve the aesthetics of of the website would be to add a photo slider of past events or to add a page or section for people to upload images of the events being held. The Website also has a spot for current events to be listed on the main page that can be linked to to the events details. There is also a place to have a list of Event types. The Event type would provide you with a list of events with the specified type selected. Our website has scalability.

9. Conclusion

Summarize your experience working on this project and the skills you gained. Reflect on the project's success in meeting its initial objectives.

Our project was successful in meeting initial objectives. Our website has user creation and registration as well as user authentication. Users can create and register events. They also have administrative capabilities such as ban and delete. The project also implements WCAG guidelines. There are no flashing utilities on the website and

the pages have titles that are self-explanatory to their function. Another WCAG we have is contrasting colors. We have a black background and white text this allows readability for anyone who may be color-blind or sensitive to similarities. We believe overall our project meets the initial requirements.

10. Appendices

Include any additional supporting information such as database schemas, additional code snippets, or user manuals.

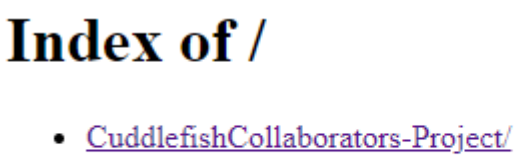
Appendix 1: IMPORTANT!!!: Please when opening follow these instructions and screenshots to set up your database to run our project:

Step 1) pull our project from GitHub and save it to the laragon www directory (you already know how to do this so I will not include instructions)

Step 2) click on the database option for laragon and sign into phpMyAdmin using username: “root” and leaving the password empty.

Step 3) Reset your password to “FinalProject”

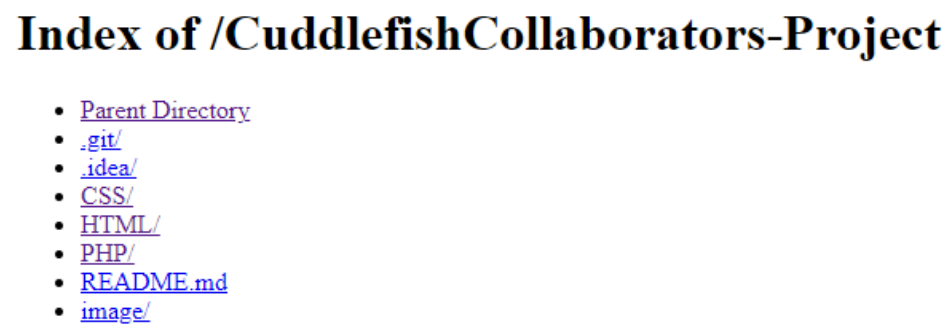
Step 4) Create a new database named “cudfishproject”



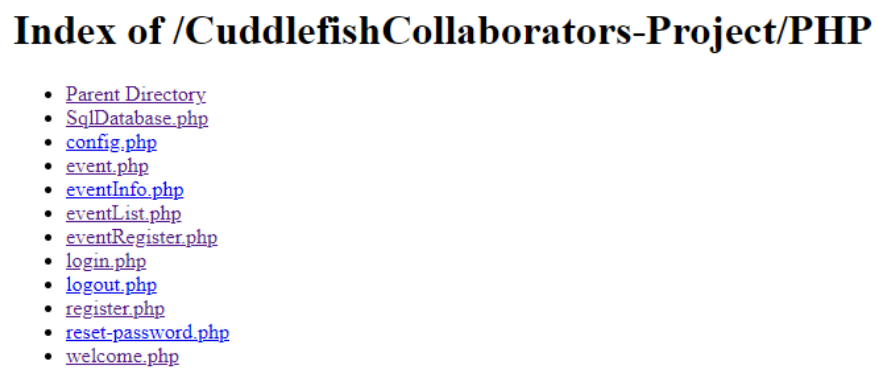
Step 5)

Click on this link once running the project via the web function of laragon

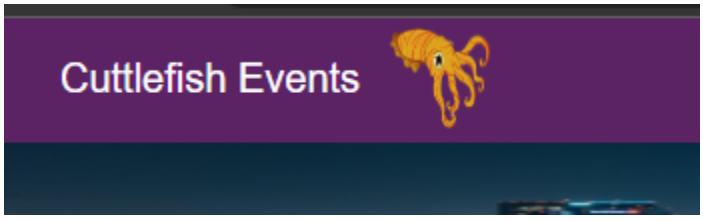
Step 6)



Click the PHP option to get this list



once you're at this list select the sqlDatabase.php item to properly create the tables in your database then you can go back to using the back arrow on your webpage and then select the HTML page to be led to the main page of our website where you can use the login function to begin testing (IMPORTANT NOTE: if the website doesn't let you use the login function this may be a bug, please click the cuttlefish events button to refresh the page and try again(picture below)

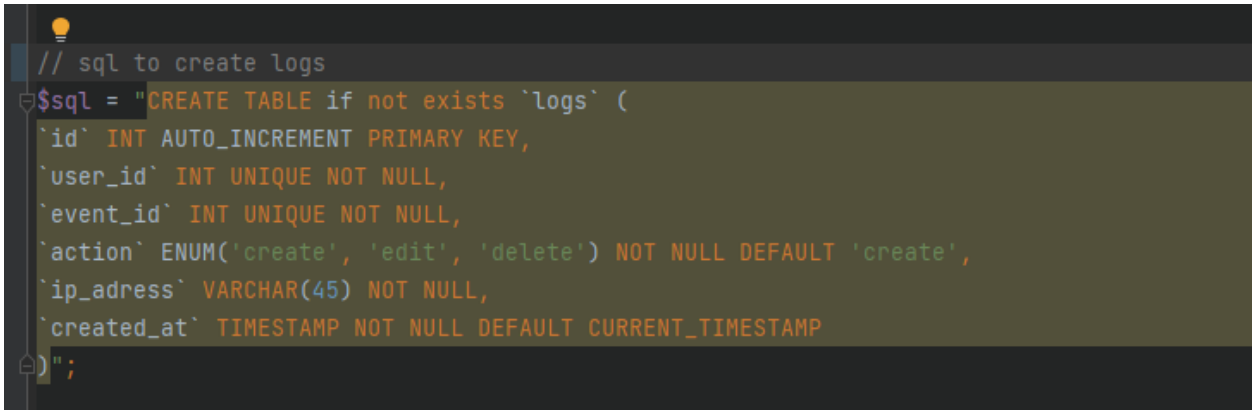


Appendix 2: Database Schema(for current use, future use would involve more function inclusion for saving admins' input):

```php
// sql to create event
$sql = "CREATE TABLE if not exists `event` (
`id` INT AUTO_INCREMENT PRIMARY KEY,
`organizerid` INT NOT NULL,
`title` VARCHAR(255) NOT NULL,
`description` TEXT NOT NULL,
`event_date` DATE NOT NULL,
`start_time` DATETIME NOT NULL,
`end_time` DATETIME NOT NULL,
`location` VARCHAR(255) NOT NULL,
`capacity` INT NOT NULL,
`category` ENUM ('wedding', 'family-reunion', 'birthday', 'other') NOT NULL DEFAULT 'wedding',
`other` VARCHAR(255) NOT NULL,
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
`updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";
```

```php
// sql to create table
$sql = "CREATE TABLE if not exists `users` (
`id` INT AUTO_INCREMENT PRIMARY KEY,
`email` VARCHAR(255) UNIQUE NOT NULL,
`password` VARCHAR(255) NOT NULL,
`first_name` VARCHAR(50) NOT NULL,
`last_name` VARCHAR(50) NOT NULL,
`user_role` ENUM('admin', 'event_organizer', 'participant') NOT NULL DEFAULT 'participant',
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
`updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";
```

```php
// sql to create registration
$sql = "CREATE TABLE if not exists `registration` (
`id` INT AUTO_INCREMENT PRIMARY KEY,
`userid` INT NOT NULL,
`eventid` INT NOT NULL,
`status` ENUM('interested', 'attending', 'not-attending') NOT NULL DEFAULT 'not-attending',
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
`updated_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";
```



```php
// sql to create logs
$sql = "CREATE TABLE if not exists `logs` (
`id` INT AUTO_INCREMENT PRIMARY KEY,
`user_id` INT UNIQUE NOT NULL,
`event_id` INT UNIQUE NOT NULL,
`action` ENUM('create', 'edit', 'delete') NOT NULL DEFAULT 'create',
`ip_adress` VARCHAR(45) NOT NULL,
`created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
)";
```

11. References

List any external resources or references used during the project.

[1] "Bootstrap," Available: https://getbootstrap.com/, Accessed on: June 8, 2023.

[2] T. V., "CuddlefishCollaborators-Project," Available: https://github.com/tienv9/CuddlefishCollaborators-Project, Accessed on: June 8, 2023.

[3] "Discord," Available: https://discord.com/, Accessed on: June 8, 2023.

[4] "W3Schools," Available: https://www.w3schools.com/, Accessed on: June 9, 2023.

[5] "Stack Overflow," Available: https://stackoverflow.com/, Accessed on: June 9, 2023.