Caleb Griffy

Joshua Cox

Lucas Pillaga

Brian Baharloo

Elijah Dobbins

Final Project Report

Our final project aimed to create a planner with daily events for multiple users using PHP, HTML, CSS, SQL, and JavaScript. Users are able to log in to an account and see a calendar for a given month. From there, they can add or remove events to particular days. These events are saved for the next time the user logs in but are private to that specific user.

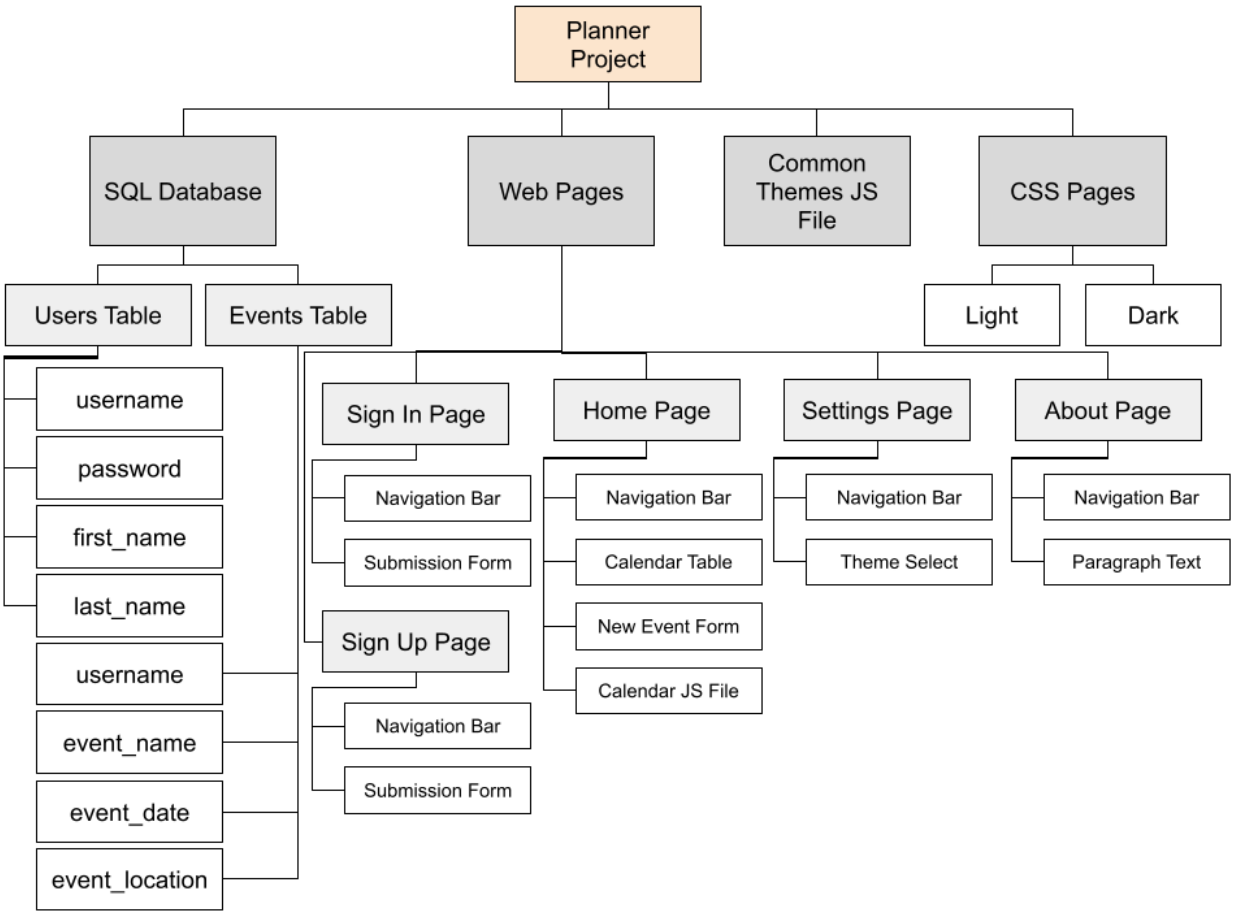
There are several HTML and PHP pages used for the planner. Across them all is a navigation bar at the top to travel to the site's other pages. The first page greets users with the login page so they can see their information as soon as possible. If a user does not yet have an account, they can follow a link to the 'Sign Up' page to create one. The primary page users will visit is the main calendar or 'Home' page. Here users can create and remove events. This page also has a display of the month for events. Users can click on the specific days for a popup to add an event to that day directly. The month display also has arrow buttons at the top that allow the user to cycle between different months and see events from the past or future. The home page is the only page not available on the navigation bar. This is so users cannot access the calendar until they have logged in. This page makes heavy use of their login information to pull events and populate the calendar. The 'Settings' page is where users can change their preferred theme and have the option to clear cookies. There is also an 'About' page to see information about the site's creators.

Two tables are used in an SQL database to store the data needed. A users table keeps track of the names, usernames, and passwords of the different users to allow them to log in. There is also an event table to keep track of the events added to the system. These events have columns for the different aspects of the event, such as its name and date. Each event also includes the username for whichever user created the event. This way, when a user loads in the calendar, the site can pull only the events needed for that user. When a user logs in, PHP code stores their username into a global variable so the site can remember the username it should search for across different web pages.

One feature added to the site is the ability to select different themes for the web page. To do this, we used multiple versions of the CSS file to stylize each web page. There is a default "light mode" with brighter colors, but users may prefer a "dark mode" with darker backgrounds and light text to contrast. From the settings page, a theme selector will use a JavaScript function to change the HTML code for what CSS file is loaded. With no further work, this would only change the appearance of the settings page. However, we store the theme the user selected as a cookie in the user's browser. This cookie will expire when the browser is closed, but until then, it saves the name of the theme the site should use. When loading each page, a JavaScript function reads for this cookie. It determines which theme was selected and changes the page to that theme on page load. The page also includes a button to "Clear Cookies". This button will set the expiration date for the theme cookie to a date from 1970 to remove them if the user wishes.

Our team divided the tasks among our group members in the following ways. Caleb worked on changing between different CSS styles. He created the alternate CSS designs besides the original and kept any changes for new features consistent across all CSS files. He also

researched how to use cookies to store the chosen styles between web pages and other visits by the browser along with clearing them from the settings page. Josh worked on the sign-up page, connecting it to a database table and adding functionality not to allow multiple users with the same username because the username is the method by which the event table divides up the events. Lucas added a sign-in page that allows users to create an account or sign in if they already have one. It features a header with links to other pages, a sign-in section with a form for users to enter their login credentials, and an error message section. The form uses the POST method to submit data to the server, which then authenticates user information by checking a MySQL database. If the authentication is successful, the user's information is stored in a session and the user is redirected to another page. The code also handles empty fields and displays error messages for users to correct their input. Brian worked on the calendar for the home page. This includes JavaScript code to create the days and cycle different months as well as making the pop-up form to add events to the page. He also worked on reading and displaying the events from the SQL database in the calendar. Elijah created a base design for the HTML pages and added the navigation bar.



Project Design Diagram