Group 8 - HI, HH, MA, TM - Event Club Website

PROJECT GOALS

This paper aims to explain the majority of the steps taken in bringing the "Event Club" website to a both visual and functional level. To begin with, the website as the name "Event Club" suggests, serves the purpose of gaining students of Malardalen University the access to the events organized by the university. It is beneficial to emphasize that only the students of the Malardalen University can utilize this website. Through the website, users can:

- Browse through different events
- Add events to their favorites and/or bookings, and display them
- Filter the events according to their preferences (category, date)
- Add/Edit profile image
- Change username, password
- Search for friends (other users of the site) on the website, send them friend requests, display their profiles,, remove friends from their friends list.
- Delete their account

The administrator of the website should be able to:

- Browse through all the events
- Add a new event
- Edit/remove an existing event

KEYWORDS

Web development, HTML, CSS, JavaScript, PHP, MySql, BOOTSTRAP, JQuery, AJAX.

1 TOOLS USED FOR DEVELOPMENT

MySQL Database: If the application exists, the data exists. For example, the data of the users such as username, password, and email. No user can log in to a website without this data, so there is a need to store the data in some way that allows retrieving and manipulating it.

A database is a collection of data structured, organized, and stored in a computer system controlled by a Database Management System (DBMS) such as MySQL.

The most common way to store the data today is to store it in a relational way; it represents the data in tables (relations) which related to each other contain columns and rows (attributes and tuples), each tuple contains values, each value assigned to its corresponding attribute.

DBMS allows the user to retrieve, manipulate, and manage the organization and optimization of the data, so it represents an interface between the database and its users.

As mentioned before, MySQL is a DBMS, made for relational databases.

SQL in MySQL name stands for Structured Query Language. It is a programming language with a specific syntax used by relational databases for retrieving, manipulating, and defining the data. Therefore, we can Create tables, Read data, Update data, and Delete data (CRUD) operations.

The benefits of using MySQL: easy to use, therefore, any developer can install it and manage the database easily. Reliable, it has been tested in different scenarios. Scalable and secure.

PHP: An open source, general-purpose scripting language that is much suited for web development. PHP is a recursive acronym for *PHP: HYPERTEXT PREPROCESSOR* and mainly serves the purpose of being a server-side scripting language. PHP pages can contain HTML with embedded PHP code which is enclosed in the following style: <?php //code ?>. The execution of the PHP code happens on the server and the generated HTML is sent to the client afterwards. The sent HTML does not display the underlying PHP code used to generate it so the client does not have any information about it.

HTML: Being an acronym for Hypertext Markup Language, HTML is the standard markup language used for creating web pages. The Hypertext in the definition of HTML means that the text is not required to be linear, but instead it can contain links to other texts. By the nature of being a markup language, HTML uses tags to define the elements within the document and has a characteristic of being highly human-readable.

CSS: Being an acronym for Cascading Style Sheets , CSS is a markup language utilized to manipulate the presentation of an HTML or XML document. It describes how specific elements should be displayed on the user screen, or on other devices. Being placed among the core languages of the open web, according to W3C specifications, CSS follows a standardized format across different Web browsers. By means of separating content of a document from presentation-related data, it also comes with huge benefits such as being reusable, flexible, and accessible.

JavaScript: Being an interpreted and object-oriented language, Javascript is widely used as a client-side scripting language for web applications. It is a powerful scripting language in the sense that it can control the behavior of the web pages on the occurrence of different events. With JavaScript, it is possible to make web pages more interactive and since it is a lightweight language and does not require compilation to run, it is possible to write it inside an HTML document which will run automatically as the page loads. JavaScript is also the basic technology for AJAX.

AJAX: Being an acronym for Asynchronous JavaScript And XML, AJAX utilizes the combination of XMLHttpRequest object (browser built-in) to request information from the web server and JavaScript/HTML DOM (Document Object Model) to present the data. Due to AJAX, it is possible to refresh information on the web page without refreshing the page. Usually the procedure for achieving this: A specific event occurs in the web page, an XMLHttpRequest object is created in JavaScript which is then sent to the web server to process the request and then the response sent by the web server to the web page is read by JavaScript and a specific action is performed by JavaScript.

Bootstrap: One Of the most popular CSS frameworks with the main usage of designing the interface of a webpage and making it responsive without much effort. It gives us access to a huge library and classes that we can use to create very creative and beautiful designs in many ways. The bootstrap was developed by twitter and until now it is still getting updates continuously. The bootstrap, being a front-end development framework, usually consists of HTML, CSS and JavaScript, and it depends on other libraries such as Jquery and PopperJs. Bootstrap relies on Jquery for the effects it adds to the designs and anything related to display properties. Also, it relies on PopperJs to improve the user experience for touch screens, such as the user dealing with the page through their phone or any tablet device.

jQuery: jQuery is a JavaScript library, it’s easy to use and contains too many features, it makes things like HTML document traversal and handling, event handling, animations, and Ajax much simpler with an easy-to-use API that works across multiple browsers. It’s a combination of versatility and scalability. The jQuery library contains these features: 1. HTML/DOM manipulation 2.CSS manipulation 3.HTML event methods 4. Effects and animations 5. AJAX. Google, Microsoft, IBM, and Netflix also utilize jQuery. In summary, jQuery is lightweight, write less, do more.

API: It stands for Applications Programming Interface. It is a standard way to communicate between two softwares with a certain set of rules, definitions, and protocols. It allows the client to send a request to some server and get a response with the information needed for the software. As an example, Google ReCaptcha is a free service provided by Google to protect the software from any attack by ensuring that the program is not trying to be accessed by a violator (i.e., a robot).

2 DESIGN PROCESS

Main Page: Not only the first page a new user interacts with after the registration, but also the main root to other pages in the website. Considering this, we wanted to come up with a design that would be completely user friendly as to attract newcomers along with entertaining the old users every time they spend time on the page. In order to achieve this, we firstly considered minimizing the number of clicks to the minimum level without causing any ambiguity and losing the core functionalities of this page. In addition, we chose mid-green and mid-orange colors as the main theme of our website and made good use of them on this page. The page itself contains the simplicity and attractiveness of the orange-green combination. We designed the layout of this page as a header on top and the main content under it. In the header, we decided to put our logo on the top left position as a signature of the site. Right near to it, there is the search bar located and it looks centered on the top of the page emphasizing its core functionality. To the right of the search bar, on the top left position of the header, we put the user avatar in a circular form as to get rid of the harshness of the simple rectangular shape and a circular ring around it in a dark-orange tone to match with the mid-orange of the logo. To the right of the user avatar, we decided to locate the logout button in a simple form of a rectangle with an orange background as other buttons. Also the header of the website is sticky so that users can anytime access its functionalities without being lost after too much scrolling and having to scroll up again to see it. As the header part is explained, in the main-content part of the website we reserved some space on the left for the side-bar in order to display filter functionality. They are styled as checkboxes to be self-explanatory and the date selection and submit buttons hold our signature of orange backgrounded rectangular shapes, increasing the consistency across the button elements. Also the side-bar is sticky meaning that it will not scroll down with the rest of the content as to make users able to apply filters on any amount of scrolling through the events. The left space of the main-container is filled with event-related information as slideshow being on the top and events coming below it in a form of 4 cards per each row. For the slideshow, we reserved the whole row width so that we could emphasize the importance of it. In our website, slideshow serves the role of welcoming the user and getting them familiar with the main idea of the page. On the slideshow, we display the titles of the events displayed in a light-green background rectangular shape to make them readable regardless of the image presented while not ruining the overall consistency. Below the slideshow, we display all the events as card elements with the properties of the top part holding the image, middle part holding the information of event date and event title, and the bottom of the card is divided into three columns in order to increase the user-friendliness. The first column of the bottom of the card holds a "reserve a seat" button, the middle part displays a text regarding the number of seats left which we thought was important information, and the right column holds an "add to favorites" button. As a group, we landed on showing the events as cards because of the possibility of organizing them in a nice grid system and showing the necessary information on them without being harsh on the eyes.

Admin page: In the design process of the Admin page, to protect the consistency with other pages and the general theme of our website, we used the similar tone of orange and green colors throughout the page. On this page, we have divided the whole page into two sections, the header and the main-content right below. In the header, it is possible to see the logo of the university on the top left position, the same position it was located in the homepage and it keeps our design consistent. And, on the rightmost part of the header, we have put the logout button for the administrator. The buttons all share the same characteristics of orange background on rectangular shape across all pages to protect the overall consistency. Below the header, we have centered a rectangular box on the main content part to make a simple but fully functional operation point. On top of this box, we have put the text Admin to capture the box and make its purpose clear. Inside this box, we have put a text on top of the search bar to clarify what to search for with a vertical orange line to make it look sleek. Below the search box, we have located a total of three buttons, two in the first row and the last one in the second row. The first two buttons are similar in functionality as admin clicks on them after searching and finding the needed event. But the last button has the core functionality of adding an event to the website and to emphasize its importance we made its width the total width of the two above buttons. The text inside the buttons are white as to be simple and easy to read through the orange background.

Edit / Add Event pages: The page layout follows that of their parent, the admin page. The header, as in the admin page, contains the logo of the webpage on the top left as a signature of our consistent design. The logout button is on the top right position following the same pattern. Below the header, we have put a larger rectangular box to make space for the input fields to look better. The box has a text Edit Event as its header, centered as to clarify the purpose of the box. Below the text, all the needed input fields have been put as their width takes up the space as needed and together they look to fulfill the box. At the end of the box, we located the Update button on the left which follows the same guidelines as our other buttons do in this website, again to protect the consistency. Secondly, it is important to mention that even though the Add Event page exactly follows the same layout as the Edit Event page (described above), there are minor differences. In the Add Event page, the title of the box is different as Add Event which explains the purpose of the box. Another minor difference is at the end of the box, as the button has a text as Publish, following the same style as that of other buttons in the website.

User page: Designing the user page in a consistent way and presenting all the functionalities of the user with the minimum number of clicks was not easy, so we made certain compromises. We divide the page layout into two parts as in the admin page, the first part being the header and main content right below it. The header is following the same design pattern as that of the admin page. Below the header, we have the main content part and we came up with the design of making a double box system at this point. Also the color scheme for the main content box changed to mainly gray and orange combination as we wanted to make the user page unique in a certain way. The left side of the box, or the side-bar, consists of the user profile avatar in circular form at the top with an orange ring around it to keep it consistent with the homepage. Below the avatar, the username is displayed with a certain margin to distinguish it from other elements. Below the username, we have displayed the navigation bar with different navigation elements sitting on top of each other. Also, each navigation element has an orange background once selected by the user as to remind the user of their selection. To reduce the number of clicks, once clicked on a navigation element, instead of redirecting to a new page, it displays its respective content on the right side of the box. For the navigation items "settings" and "passwords", we are displaying the input fields in a way as to make use of the available space as much as possible. The design for the update button at the bottom left of the page is consistent with the overall design of the buttons in the website and looks nicely positioned. Once clicked on the navigation element "All Events", we display the text "My Events" on the top of the main-content box and below it, we display the corresponding events as cards as two per each row. The design of the cars in this section is different from those at the homepage as here the cards are not as oval and displayed in the grayish color rather than green. We made such a design choice because we wanted it to look well suited to the user page and even though we divide the card into three rows we do not display as much information as in the cards in the homepage. Here, the cards show the image first, the title in the second row, and a browse button and "add to favorite", "reserve a seat" on the same third row. As before all the buttons used in the user page follow the same style as the other buttons we have in the website. The "Friends" navigation item, once selected, displays the text "Friends Page" to clarify the purpose of the box. We have put a search bar with a “send request” button to the right of it since it looks well suited in the available space. Below, we are displaying the friends of the user in the specific format of a rectangular box containing profile image of the friend, username, and two buttons next to each other at the right end of the box to keep them away from informational context. The “Requests” navigation element uses the same layout as the “Friends Page” with two minor differences being the “request” text on top and no search bar and send request button. The last navigation item “more settings” has only one functionality so we occupied the first row after the “More settings” title and placed the “Delete account” label and the “delete” button distant from each other to make good use of the space. Also, once the “delete” button is pressed, a pop-up alert is displayed to the user to confirm as to not accidentally delete their account.

Login & Sign-Up pages: In the Login and Sign-Up pages, we followed a similar pattern as to minimize the ambiguity and maximize the functionality while not losing the consistency of the main theme of the website. The pages are in a double-box format, left-box is the same color as the main theme of the website and displays the logo of the university as a signature of the university. For the box on the right, where all the functionalities that these pages serve are contained, we chose a grayish color to make it easy on the eyes and easily distinguish the input fields from background. The buttons on the right box are displayed in orange color which is the same as the university logo on the left-box to improve the consistency and to merge them easily.

Event-content page: The page is using a similar layout as that of the home page with the header being identical in terms of the elements it contains, and also their positions are the same. Since the user makes a transition to this page by clicking on any of the event cards in the homepage, we would need a consistent design so as not to confuse the user. In this page, below the header, we reserve all the space to display the detailed information of a specific event. We first display the image of the event with high width and height to make use of the available space. Below the image, on the left hand side we display the title in bold and the description of the event right below the title. To the opposite of the title, on the right hand side of the image, we display the date the event has been added in bold. To the right of the description, we display a box with a green background and inside we put the necessary details of the event as key value pairs, one per each row. The reason for the green background is to keep consistency with the homepage and make the transition smooth between pages. At the bottom of the box, we put the “add to favorites” and “reserve a seat” buttons next to each other in such an alignment that would make good use of the space available.

3 DEVELOPMENT PROCESS

Before starting the implementation of the design choice and ideas we had, we first wrote a detailed review of all the functionalities and specifications we wanted to achieve in the project in a simple notepad editor. By carefully grouping the tasks as cards, we constructed daily objectives to complete so as not to lag behind the deadline. To start with the development, we decided to keep everything as separate as possible so that later there would be an ease to add new functionality. Below, there is an easy to follow schematic of everything we did in the backend to achieve the functionalities desired.

Designing the database: for designing the database, we have to know what kind of data we need to store and what is the relationship between them

The description of Event Club website in terms of data:

- There are events, each event has a unique event id, unique title, body, image, category, number of tickets or places, event date, and the date of creating the event. event id is used as a primary key.
- Users can sign up and log in to the website; therefore, each user has a unique user id, unique username, unique student email, password, profile image. user id is used as a primary key.
- Users can send a friend request to each other and they are able to accept or reject the request, if the user accepts then they become friends.
- Users can book/unbook the events; we need to save the number of reserved tickets/places for each event to calculate the available tickets.
- Users can like/unlike the events.
- The ones who can add/remove/edit the events are the admins, they are part of users so we need to check if the user is admin or not.

For visual description, we built an ER Diagram as shown in figure 1.

To understand the shapes:

- Rectangle: the entity like user and event.
- Circular shape: the attributes like username and email. (underlined attributes represent the uniqueness)
- Diamond: the relationship between the tables.
- Solid line: mandatory partial participation.
- Double solid line: mandatory total participation.
- Dotted line: optional participation.

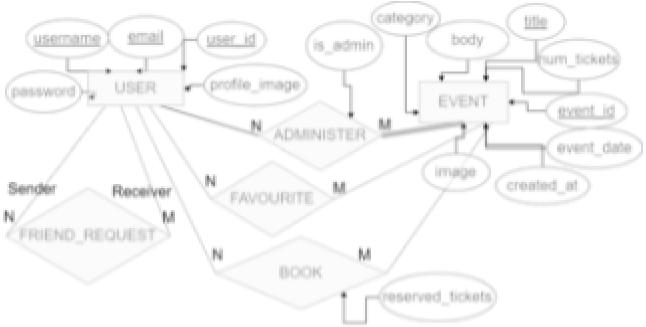


Figure 1. ER Diagram for event club's database

After that, we built the schema to show the tables that meet the required data and the relationships between them, as shown in figure 2.

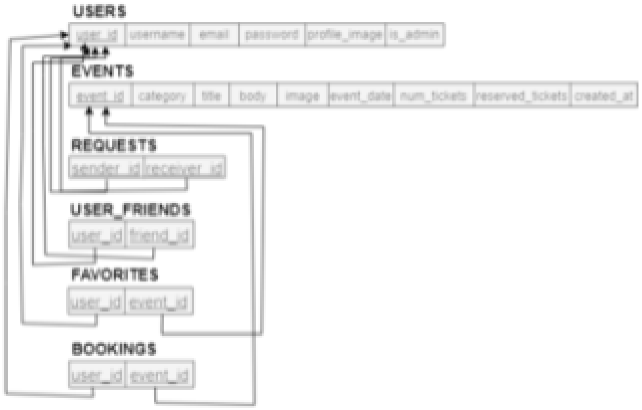
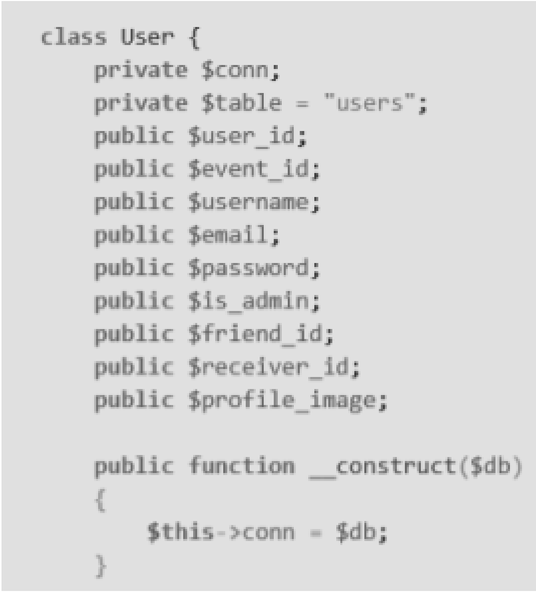


Figure 2. the schema of event club's database

Designing the classes to implement core functionalities in PHP: In order to achieve the core functionalities that we need in Event Club, we designed template classes both for users and events. Through these classes, we have figured out what type of data we need to read, update, delete, and create. So, depending on that, we built appropriate methods for each class and in this way, we managed to reach a clean design for handling the requests rather than increasing the redundancy by coding many queries. Below, the design of the User class is shown with its corresponding attributes.

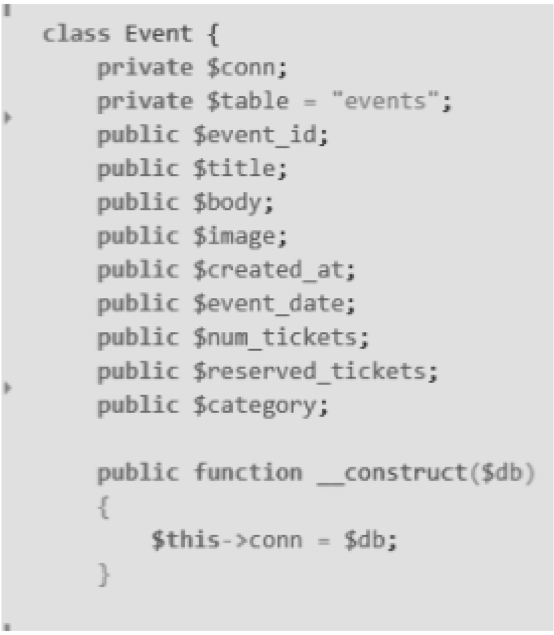


```
class User {
    private $conn;
    private $table = "users";
    public $user_id;
    public $event_id;
    public $username;
    public $email;
    public $password;
    public $is_admin;
    public $friend_id;
    public $receiver_id;
    public $profile_image;

    public function __construct($db)
    {
        $this->conn = $db;
    }
```

Figure 3. The User class definition

As a follow-up on the consistency, all the user related functionalities such as creating a new user, validating the existing user, displaying the friends list, user specific events in favorites and/or bookings, and many other small bits related to a specific user are constructed with the template user class. Similarly, we apply the same technique for all the event related functionalities. There is a definition of the Event class attached below.



```
class Event {
    private $conn;
    private $table = "events";
    public $event_id;
    public $title;
    public $body;
    public $image;
    public $created_at;
    public $event_date;
    public $num_tickets;
    public $reserved_tickets;
    public $category;

    public function __construct($db)
    {
        $this->conn = $db;
    }
```

Figure 4. The Event class definition

In our website, the event class is the root of every single event related functionality such as fetching the latest events from the database, displaying specific categories/dates of events, manipulating the existing events in terms of adding, updating, and deleting them. It is noteworthy to mention that some of the functionalities depend on both the user and the event classes. So, there is a strong connection between these two classes which improve both the readability and flexibility of the code.

Sending/Receiving data process: After implementing the classes with the required methods, we made separate PHP files for each method, in each file we made a class object, assigned the values of the object's attributes by GET/POST requests, and then the appropriate method is called.

To display or manipulate the data, we included the PHP method files as needed for each page we have. For example: to display the events in the main page, we included a "readEvent.php" file, got the data from it, and assigned it to the right HTML elements to display it.

4 ACHIEVEMENTS

All of the functionalities mentioned in the goals of the project section are achieved through the design and development procedures as we explained above. Additionally, we managed to apply responsiveness to every single page existing in our website and made their appearance adjust accordingly depending on the screen properties of the user device. Even though there are minor details that may mismatch with the initial proposal we had, the core interactions find their place in our website. To mention some discrepancies with the initial proposal, we omitted the promise that "Admin can remove members from an event" since this would not be appropriate in the scope of the website, meaning no reasonable circumstance would lead to this act. Also, we improved the privacy of the website by narrowing its user scope only to the students of the Malardalen University rather than anyone with the link. This change, we believe, is to protect the safety and privileges of the university students.

5 POSSIBLE IMPROVEMENTS

Due to the time constraints, not all of the functionalities that we desired were possible to be managed. Despite this, the solid groundwork makes it possible to add improvements in the future. Firstly, there can be added a notifications tab on the homepage for the user in order to display the received friend requests or remind the user of an upcoming booking. Also, even though not very critical, we would like to add a maps feature to display the location of events with markers and possibly combine it with some filters such as sorting by distance, green area, and etc. Last but not least, verifying the email is one of the things that must be added as a further security improvement. We believe these functionalities to be nice additions and can easily be implemented given enough time.

REFERENCES

[1] Amazon. What is an API?. Retrieved October 29, 2022 from https://aws.amazon.com/what-is/api/

[2] Google. What is ReCaptcha?. Retrieved October 29, 2022 from https://developers.google.com/recaptcha

[3] JavaScript Info. An Introduction to JavaScript. Retrieved 28 from October, 2022 https://javascript.info/intro

[4] jQuery. What is jQuery?. Retrieved October 27, 2022 from https://jquery.com/

[5] MDN Web Docs. CSS: Cascading Style Sheets. Retrieved October 28, 2022 from https://developer.mozilla.org/en-US/docs/Web/CSS

[6] MDN Web Docs. JavaScript. Retrieved October 28, 2022 from https://developer.mozilla.org/en-US/docs/Web/JavaScript

[7] Nick Schäferhoff. 2020. More information about the bootstrap?. Retrieved October 28, 2022 from https://websitesetup.org/bootstrap-tutorial-for-beginners/

[8] Oracle. What Is a Database?. Retrieved October 26, 2022 from https://www.oracle.com/database/what-is-database/

[9] Oracle. What Is MySQL?. Retrieved October 26, 2022 from https://www.oracle.com/mysql/what-is-mysql/#mysql-attributes

[10] Oracle. What is a Relational Database (RDBMS)? Retrieved October 26, 2022 from https://www.oracle.com/database/what-is-a-relational-database/

[11] One Month. 2017. jQuery vs Javascript. YouTube. Retrieved October 29, 2022 from https://youtu.be/S8ZuLY-Flm0

[12] Saad Aldabbi. 2021. What is the use of the bootstrap and how do we use it? Retrieved October 28, 2022 from https://harmash.com/posts/what-is-bootstrap-and-how-to-use-it

[13] W3Schools. What Is HTML?. Retrieved October 28, 2022 from https://www.w3schools.com/html/html_intro.asp

[14] W3Schools. What is the bootstrap?. Retrieved October 28, 2022 from https://www.w3schools.com/whatis/whatis_bootstrap.asp

[15] W3Schools. What is PHP?. Retrieved October 28, 2022 from https://www.php.net/manual/en/intro-whatis.php

[16] W3Schools. How to use Jquery?. Retrieved October 29, 2022 from https://www.w3schools.com/jquery/jquery_intro.asp

[17] W3Schools. Why we use Jquery?. Retrieved October 29, 2022 from https://www.geeksforgeeks.org/why-we-use-jquery-in-our-web-application/

[18] W3Schools. AJAX Introduction. Retrieved October 28, 2022 from https://www.w3schools.com/xml/ajax_intro.asp