# Brief Description of Technologies and Structure

1. Server Side Technologies

* PHP

In this website I have used php for making the website dynamic. I have used php by creating webpages in .php extension. It allows us to add php code inside it. <?php ?> I have added all code of php language inside of this tags. Using this tags I have rendered all of website content and images.

* Mysql

I Used Mysql database tables as following formats

* + - 1. Events
         1. Event\_id
         2. Event\_title
         3. Event\_description
         4. Posted by
         5. Type
         6. Category
         7. Event\_url
         8. Rank
         9. Time and date of event
         10. Time of event created
      2. Event list (for storing information of attendees of events)
         1. Event\_id
         2. User\_name (who shows his interest in Events)
      3. Rank ( Store rank Interest numbers)
         1. Username (A name of user attending an event)
         2. Event\_id (id of event which is user going to attend)
      4. Users (storing user details)
         1. User\_id (unique id is given to all users)
         2. Username (everyone has their own unique user\_name)
         3. Usermai (email of user)
         4. Password (user password in md5 format)

1. Client Side Technologies
   * + HTML

Using html’s tags I have made structure of Webpages. They have Using <section></section> and <div></div> tags I created structure of each section and Inserted another inline tags and anchor tags.

* + - CSS (cascading style sheet)

Css helped me to design and add color or backgrpunds to website

Using <link href=”name.css”> css stylesheet has been added in html structure

* + - Javascript

I used javascript by adding <script src=”app.js”> tags in main html structure.

* + - Bootstrap

Bootstrap has been added to help me to give style to html structure by adding classes to it . It is initialised liked this <div class=”text-center text-whie justify-content-center”>

**Implementation of the required functions:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Functions** | **Main Source File(s)** | **Notes** | | | | | | | | | | | | |
| *S1. View the basic information of all events* | ***Find.php line:444*** | Students are able to sight the relevant information for the uploaded event. For instance, the name, description, time, number of interests and the event organiser.    I used php for making database connection. mysqli\_conn($db ,$user) is the key to connect wesite with database. Using Mysqli\_query($connection,$query) I have fetched result from database.  echo $row[‘result’] to render it on site. | | | | | | | | | | | | |
| S2. List the basic event information by one of the three catalogues. | ***Find.php line:375*** | The requirement of categorising the relevant event information into three catalogues (sport, culture, and others) was exceeded.  Students are able to employ a more extensive groups when listing the event information.  **Client side technology**  I used dropdown like this for navigation.  1.sport  2. Arts  3.Buisness  By clicking on each of them allows user to display the result which only belongs to that categories. This creates a POST request on server  **Server side** **technology**  First of all I get value of category value from “POST” request. Then I  Made query like ‘select \* from events where category= “user entered category”, Then by using echo and mysqli\_fetch\_assoc I rendered all events from that category on website. | | | | | | | | | | | | |
| S3. List the basic event information ordered by the  interest ranking | The feature that sorts the relevant information by the  interest ranking is positioned in the ***“find.php” line:357*** | Students are able to sorts the relevant event information by the interest ranking.  **Client Side Technology**   1. I have added on select dropdown at webpage in “find.php “ where user selects sort by rank option and I have managed it to 2. I used to wrap this dropdown in form field so whenever user selects this field I send its value by POST method to server.   **Server Side Technology**   1. By using request\_method==”POST,   I check if user want’s to check events by their interest rank poition. I run query to “select \* from events ASC” which sends the list of events in ascending order. | | | | | | | | | | | | |
| S4. Click an event to view more details of the event (basic info + picture, organiser contact, venue etc.) and showing your interest to this event  (e.g. by clicking a | Clicking on event allows student’s to take them on another site where the y can see additional information about event is positioned at ***“find.php” line:457*** | Students are able to check additional details of event by clicking on it  **Client Side Technology**   1. I have used <a></a> tag for displaying the result box which works as a link redirecter. It takes user to “event.php” page.   **Server Side Technology**   1. In each anchor tag I have added event Id of that event. When user clicks on it page generates one GET request and on event.php first I checks if it’s valid event id then I database fetch all details of that event from “events” table and render it by using echo function. | | | | | | | | | | | | |
| button) which should increase the interest ranking by one | The feature that allows students to portray their interests for an event by clicking the ‘Interests’ button is positioned in the***“event.php” line:311*** | Students to portray their interests for an event by clicking the ‘Interests’ button.  **Client Side Technology**  At ”event.php” below the organiser name there is button which names interest rank. When students clicks it it generates post request.  **Server Side Technology**  When user clicks on “interest rank” button Server get’s the value of that event through post request. I used that event id and username from session cookies to insert it in “rank” table . I create another row and after inserting this details. Number of interested peoples gets increased. | | | | | | | | | | | | |
| S5. Register to become an organiser. | The feature that enables students to register to become an event  organiser is positioned in the “***signup.php” from line:1*** | Students are able to become an event organiser by registering on the website. The website askes the user for relevant information needed to register. For instance, an email and password. Furthermore, the student has the possibility to upload a profile picture while registering.  **Client Side Technology**  On “signup.php” user fills all details and after clicking on signup button wewpage generates “POST” request.  **Server Side Technology**  First of all I get’s all information which user gives from input field through post request. Then I checks if user is already registered with that username and login by query of “select \* from users where username=”input value”. If it didn’t match then registered as a new user | | | | | | | | | | | | |
|  | | |  | | | | | |  | | | |
|  | |  | | | |  | | |
|  | | | | | | |  |  | | |
| E1. Login/out the system. | The feature that logs the student in and out is positioned in the  “***Signup.php”*** | On the website, students can login | | | | | | | | | | | | |
| and sign out. | And Logout.  **Client Side Technology**  I have added one popup for login. By clicking it user can enter his username and password in order to login.  **Server Side Technology**  Popoup gets all the information from which user has entered and by using “select \* from users where username=””and password=”user input”. If table contains result then session starts and user redirects to “find.php” page | | | | | |  | | | | | |
|  | |  | | | | | |  | | | | |
|  | | | |  | |
|  | |  | |
| E2. Add a new event to the system. | The feature that allows the event organisers to develop new events  is positioned in the ***“create\_event.php”*** | Event organisers can add new events to the website system**.** | | | | | | | | | | | | |
|  | | **Client Side Technology**  By clicking on create event user redirects to create\_event.php where | | |  | | | | | | | |
| **Client Side Technology**  When user gets on create\_event.php I gets his username from session cookies | | | | | | | | | |  |
| Then after he enters all the details about events then server generates “POST” request and I get all data in array by using $POST array I exploits that data and add it to the events table where posted by is set to username | | | | | | | | |  |
| E3. List all events organised by you (the current organiser). | The feature that allows event organisers to view all the events  they have created is positioned in  the ”***your\_events.php”*** | Event organisers can add new events to the website system.  **Client Side Technology**  By clicking on “Your Events” from user dropdown he can see the list of past events he organised.  **Client Side Technology**  When user gets on “your\_event”.php I gets his username from session cookies and runs a query “select \* from events where posted by= username” and this returns us data in which it contains all of the event details of that user and their events. | | | | | | | | | | | | |
|  | |  | | |  | | | | | | | |
|  | | | | | | | | | |  |
|  | | | | | | | | |  |
| E4. Update an event organised by you (the current organiser). | The feature that allows event organisers to update the events they have previously made is  positioned in the “***edit\_events.php”*** | Event organisers can add update the events that they have made on the website system**.** | | | | | | | | | | | | |
|  | | **Client Side Technology** | | |  | | | | | | | |
| At “your\_event.php” user have an option of eidt event . That button redirects user to a new “edit\_event” page, where user can reupload all of its content.  **Server side Technology**  When user comes to “your\_events.php” I used session[‘username’] to get username. And running sql query for | | | | | | | | | |  |
| “select \* from events where posted by =username” which returns only that events which user creates. After That I used anchor tag to redirect user on new page for edit details which is based on “get” method that anchor tag hods value of event\_id. And on “edit\_event.php” user submits post request for updating new details. | | | | | | | | |  |

# Implementation of the security features

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Features** | **Main Source File(s)** | **Notes** | | | | | |
| *Authentication*  */Authorisation* | The authentication security feature is positioned in the “***signup.php”*** | The authentication was done by asking user for his password on login popup.  **Server Side Technology**  When user enters his password I get it by “POST” method and after that I used mdf5(user password) which returns us a new md5 string which cannot be identifies and we check if that password is belongs to that user by running simple query like “select \* from users where email=usermail and password=”entered password”” | | | | | |
| Form Validation | The form validation security feature is positioned in the each of input tag | The form validation was done by employing the I have used input type=”email,text,number”which are inline properties which allows us to validate form automatically. | | | | | |
|  |  |  | | | |
|  | | | |  |
|  | | |  |
|  | |  |
| Handle Injections (SQL/HTML) | The handle injections security feature is positioned in the “***signup.php” line:10*** | The handle injects was done by employing the mysqli\_escape\_string  ($dbConn,$\_POST[‘userinput’])  This method removes unwanted html tags and after that I insert it in users table. | | | | | |
| Hash Password | The hash password security feature is positioned in the ***“signup.php” line:10*** | The hash password was done by employing the md5(“entered password”) this returns a md5 string which is one side function.  We can’t even imagine what is password by looking at it | | | | | |
|  |  |  | | | |
|  | | | |  |
|  | | |  |
|  | |  |
| Restrict File Upload To Only Images | The limited picture file upload security feature is positioned in the every input field which accepts file | The limited picture file upload was done by employing the input type=”file” accept=”image/.jpg” which restricts user to upload only image type of files. | | | | | |  |
|  |  |  | | | |
|  | | | |  |
|  | | |  |
|  | |  |
| Cross-Site Request Forgery | The cross-site request forgery security feature is positioned in the ***“php.ini” line:1*** | The cross-site request forgery was done by employing the  setcookie('samesite-test', '1', 0, '/; samesite=strict')*;*  this restricts to use cookies only for same site. So any outside can’t attact and access cookies. | | | | | |
|  |  |  | | | |
|  | | | |  |
|  | | |  |
|  | |  |

# Implementation of stretchers

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Stretchers** | **Main Source File(s)** | **Notes** | | | | | | | | | |
| Allow for the displaying tables to be sorted based on headings. (e.g. by date, name ). | This feature is positioned in the “***find.php” line:355*** |  | By using this method user’s can check events by entering their custom date. I have used bootstrap input field as a “date” type which gives user one calender so user can tick a date and make “POST” request. Then from Server side I gets that input field and check it from event table is there any event which is occurring between that date if yes it gives a list of that events and if no it returns false. | | | | | | | | |  |
|  | | | | | | | | |
| Allow an event to have multiple pictures. | The multiple pictures feature is positioned in the “***create\_event.php”*** | Multiple pictures can be used when an event is being uploaded on the website. When the Student is in the process of creating the event. There is an element that prompts them to upload more than one photo to be displayed.  **Client Side Technology**  On “create\_event.php” user can upload more than one image which renderes when user loads page,  **Server Side Technology**  When user upload their image I get by “POST”[‘file’] method after that I uploads that image in folder by using move\_uploaded\_file()  Function. So webpage can load it. | | | | | | | | | |
|  |  | | |  | | | | | |
|  | | | | | | |  | |
|  | | | | |  | |
|  | | | |  |
| Allow a student to send an email to an event organiser. | The email organiser feature is positioned in the ***“event.php” line:302*** | An ‘Email Organiser’ button was developed to enable the students to send an email to the event host. Once the button is pressed, the Student is redirected to an email format, allowing them to proceed with the email.  **Client Side Technology**  User have an option below of event organiser name where they can send a mail to organiser .  **Server Side Technology**  I have achived this by joining events table where posted by =username and users table where username = posted by and “select \* from email where username=posted\_by”. | | | | | | | | | |
|  | |  | | | | |  | | |
|  |  | | | | | | |  | |
|  | | | | |  | |
|  | | | |  |
| Allow an organiser to link one event to another relevant event and/or another web source. | The web source organiser feature is positioned in the “***create\_event.php”*** | Due to the increasing rates of virtual events, the event organiser can insert a virtual event link (web source) while creating the event. Furthermore, related events appear when the students are browsing events. For instance, if a photography event is uploaded, another creative event will be there.  **Client Side Technology**  User can select option of his event category from create\_event.php from a category dropdown.  **Server Side Technology**  I have achived this by creating one column named “event\_category” which stores the category of that event when user uploads it to the database. After that when anyone goes to event.php page by taking event\_id from their url ‘  I finds which category is of that event and at the bottom page I used to display 4 recently added events which belongs to that category and the query of it is “select \* from events where category = page event category” | | | | | | | | | |
|  | | |  | | | | | |  |
|  |  | | | | | | |  |
|  | | | | |  | |
|  | | | |  |

# Database Structure

I Used Mysql database tables as following formats

* + - 1. Events
         1. Event\_id
         2. Event\_title
         3. Event\_description
         4. Posted by
         5. Type
         6. Category
         7. Event\_url
         8. Rank
         9. Time and date of event
         10. Time of event created
      2. Event list (for storing information of attendees of events)
         1. Event\_id
         2. User\_name (who shows his interest in Events)
      3. Rank ( Store rank Interest numbers)
         1. Username (A name of user attending an event)
         2. Event\_id (id of event which is user going to attend)
      4. Users (storing user details)
         1. User\_id (unique id is given to all users)
         2. Username (everyone has their own unique user\_name)
         3. Usermai (email of user)
         4. Password (user password in md5 format)

# Further Notes

# User

This website is created by using core php 7.0 . Anyone can access the website by using it’s url. This website allows everyone to signup on website and host their own events, check events organised by their friends or teachers There are no limitations on users they can use website freely. At a point this website is for free of cost to use everyone , there are no any charges for signup or create an event.

# Database

All php and html code is structured in each of their files. Files starts with php if there are any post requests, They try to fetch results as that request. Some bootstrap classes has been used to make it responsive and add custom styles to html structure. In partials folder there is one folder named “database.php” which connects website to the database. When we publish this website on hosting services or any other vps . We have to make sure that variables in “database.php” are named correctly, otherwise website couldn’t run.

# Website structure

For creating an event description I have used ckeditor which gives one user interface to add anything into it . By using it user can add links images or wise description. There are no words limit. User can as much words as he wants. In “partials” folder there are another two files named “login\_hander.php” and “logout\_handler.php” This files helps us to log in to user’s account and make them able t create, edit and explore the events. After deploying website it is necessary to change header value in “loginahnder.php” else it will not work.