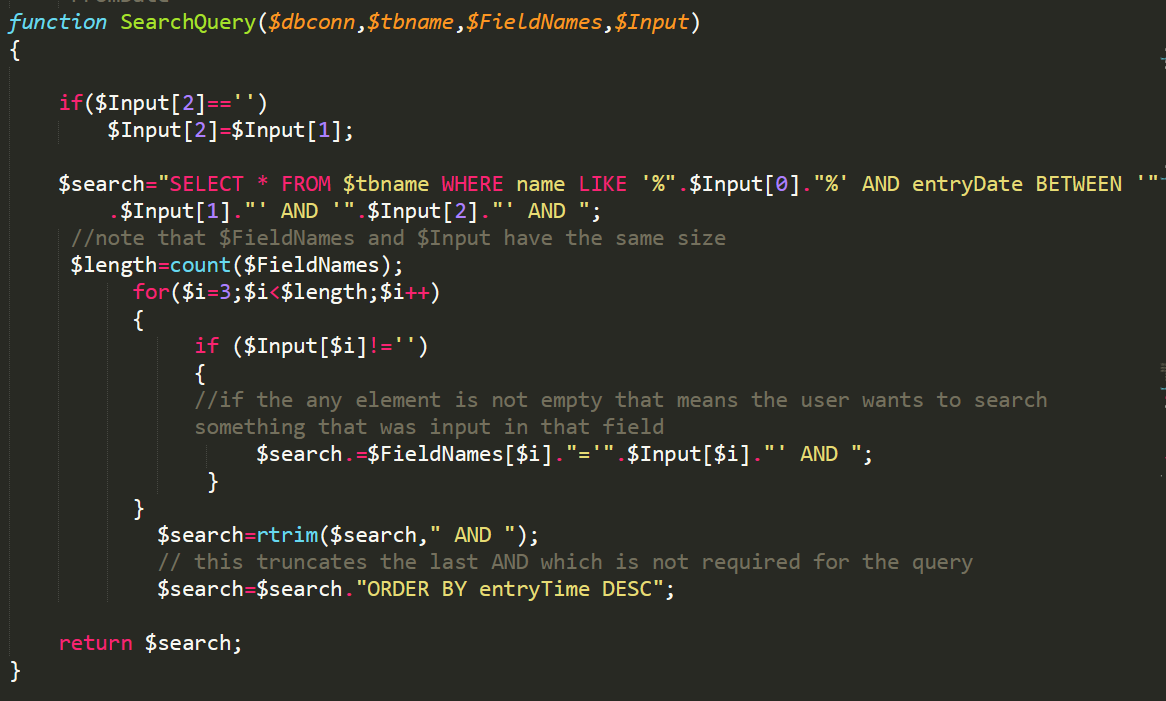
**CRITERIA C**

1. Search (Wild cards, Function, array, queries, database operations)
2. JavaScript
3. CSS
4. Functions
5. Encryption
6. Use of Include
7. Cookies and Session
8. Libraries
9. Database operations and Queries
10. Displaying Tables

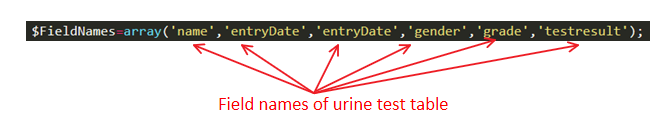
**1.Search**

To create a search queries, I had to combine different complexities like if else, functions, wild cards, arrays, queries, database operations, for loop, and predefined functions (like rtrim, count and string concatenation).

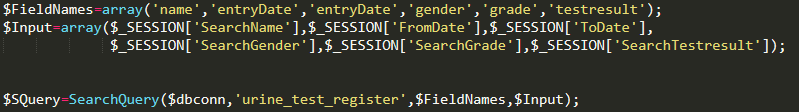


The use of Wild cards allows the search to be more flexible. For example, if the user wants to search for say- “Richard Feynman” in the database then even entering “feynman” or “FEynMan” will give the give the result “Richard Feynman”.

The function makes the code general and can be called for different tables and inputs. For example, the function is called for creating search query for urine test table, based on inputs from user and fields relevant to urine test table.

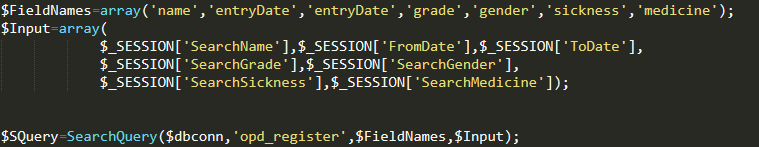


The function is called, and the return value is stored in variable $SQuery.



The same function is called for creating search query in OPD table.

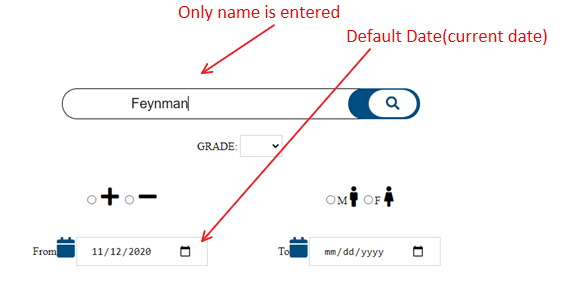




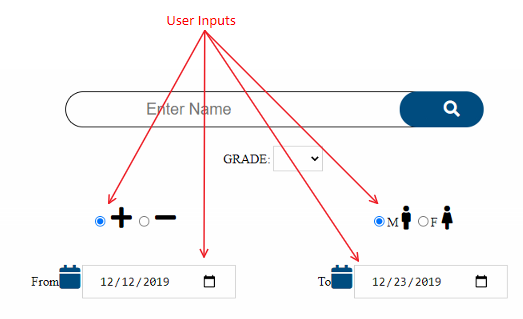
This reduces redundancies and makes the code more efficient.

The function also makes it possible to make dynamic query based on user input. For example, if the user only enters name and leaves all other parameters blank, then Query returned by the function is: SELECT \* FROM urine\_test\_register WHERE name LIKE '%Feynman%' AND entryDate BETWEEN '2020-11-12' AND '2020-11-12'

Where the date was the default date(current date)



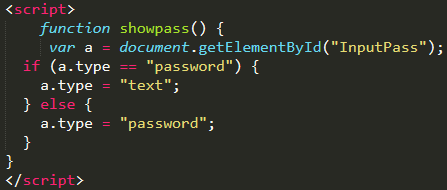
Say for example if the user wants to see how many males testes positive from 2019-12-12 to 2019-12-23 then the query returned by the function is: SELECT \* FROM urine\_test\_register WHERE name LIKE '%%' AND entryDate BETWEEN '2019-12-12' AND '2019-12-23' AND gender='M' AND testresult='+'



**2. Java Script**

|  |  |
| --- | --- |
|  |  |
| Before clicking on the checkbox | After clicking on the checkbox |

This is achieved by the following JavaScript code



This JavaScript converts the password box which has been given the id= “InputPass” and converts its type to text if the type was “password” whenever the user calls the function by clicking on the checkbox labeled “Show”.

**3. CSS**

For styling I have used CSS. For example, to style the navigation bar the following CSS code is used:

|  |
| --- |
|  |
|  |
| |  |  | | --- | --- | |  |  | | **Normal Styling of Buttons** | | |  |  | | **Styling when cursor is hoveing over button** | | |  |  | | **Styling when button is active (clicked)** | | |

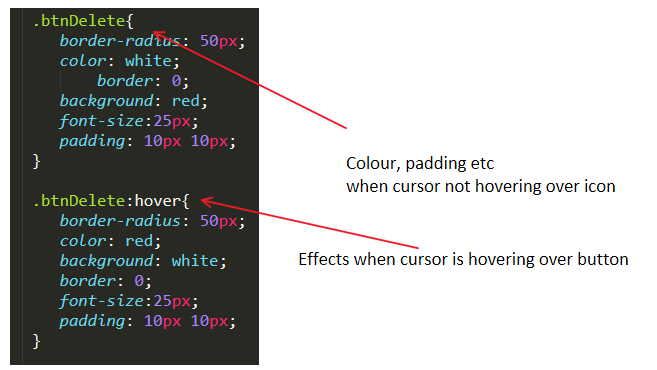
The same is also implemented for opd form and opd test table buttons. For home, search and logout buttons slightly different styles are implemented but the basic principle remains the same.

These styles are imported by using the following code in the script:

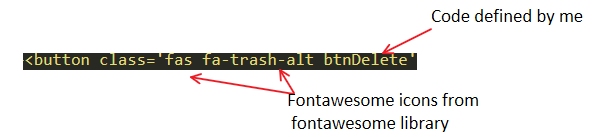


I have used CSS libraries I have created (StyleFiles.css) with font awesome libraries to get a richer UI.

For example,



When implementing style for a delete button the following code is used:



This creates the following effect

|  |  |
| --- | --- |
|  |  |
| When cursor not pointing over icon | when cursor is hovering over icon |

This is what makes some of the dynamic UI possible.

**4.Functions**

To create and display medicine report of any date, the following function is used:

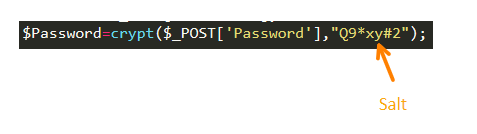
It makes use of query statements, associative array and conditional statements.



Functions have been used multiple times in my code for different situations, giving rise to modularity and usability of the same block of code for performing operations based on inputs that are relevant to the situation.

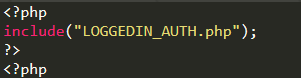
**5.Encription**

The crypt function is used with salt. This makes the passwords secure.



**6.Use of Include**

The LOGGEDIN\_AUTH.php script authenticates the user. The following code is included on top of every page that is displayed to the user so that the LOGGEDIN\_AUTH.php script is imported.



This prevents any unauthorized access. So, when the user has logged out. Anyone who changes the URL to get access to the website will be sent automatically to the log in page.

For example:



If a person now changes the URL to the following:



Then the person is taken automatically to:



This would not happen if the user had not logged out. Hence this provides security against such attacks by unauthorized person.

Another use of include is to import the script FUNCTIONS.php



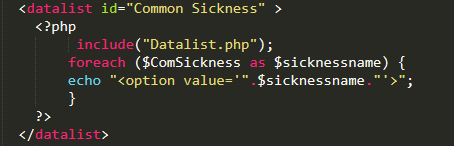
Helps shorten the code and gives the code more general character and helps abstract unnecessary details.

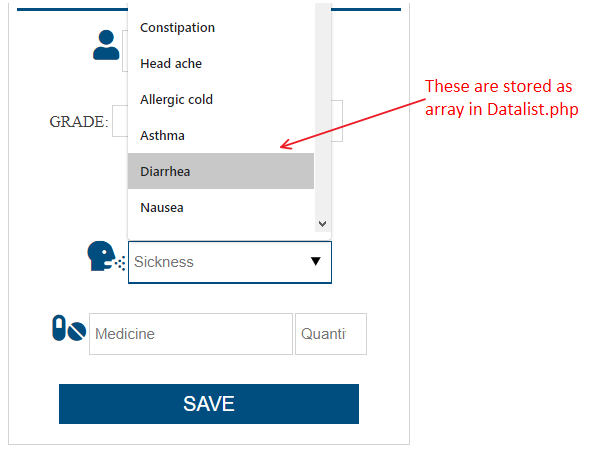
Datalist.php contains large arrays (but not too large that it needs to be stored in database) that stores the list of common sickness and Medicine.



*Fig: Datalist.php code*

I have used the include function here to call the array in datalist.php without making the code too long unnecessarily.



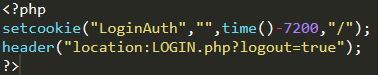


**7.Cookies and Session variables**

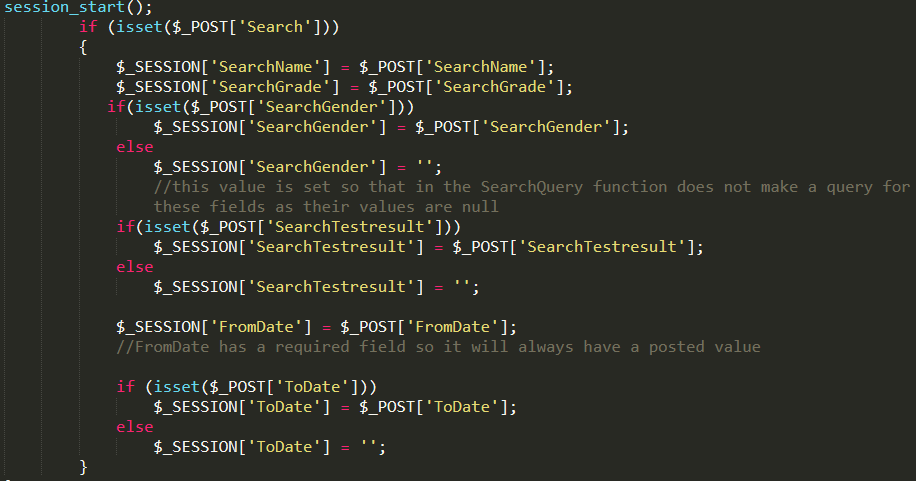
Cookies are used to keep track of the user, especially to keep track of if the user is logged in or not. Setting cookies when an authentic user enters the correct username and password



When the user clicks log out the cookie value is set so that cookie expires



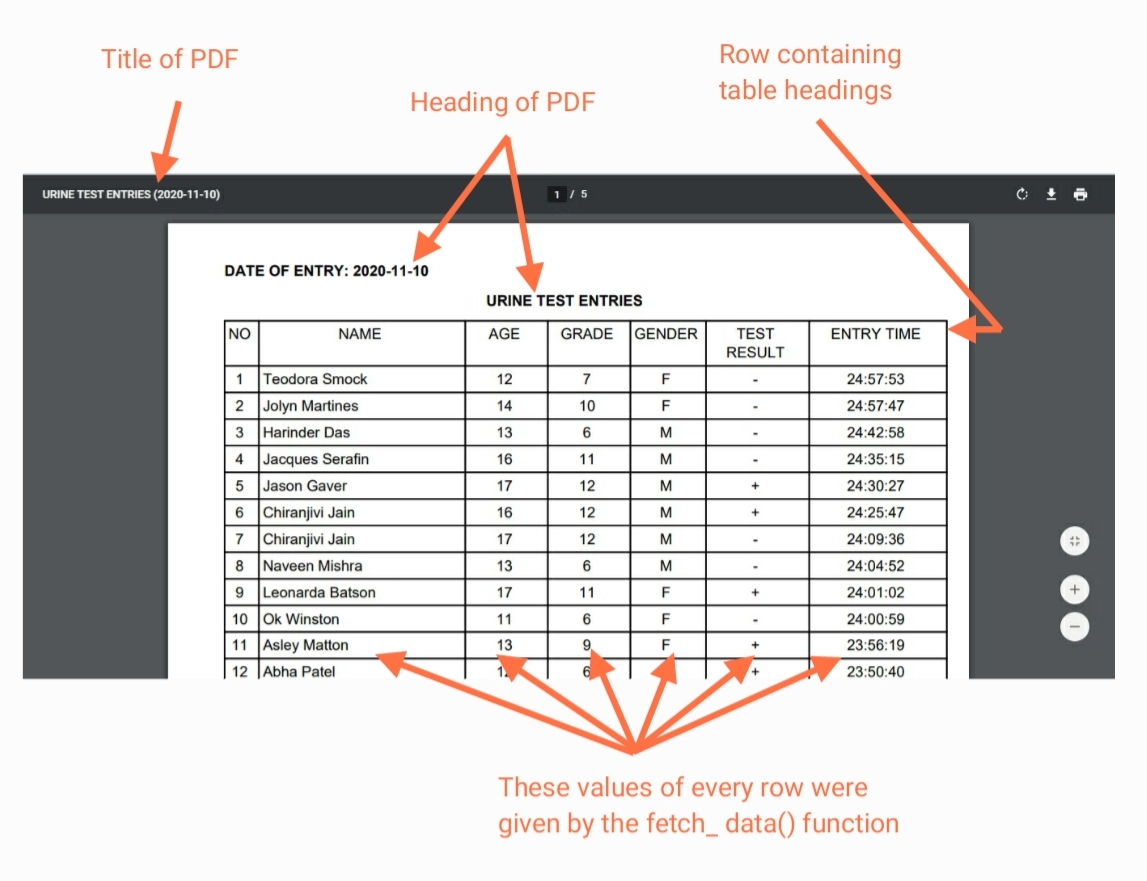
Session variables do not change even if the page is refreshed or even if the user goes to another page. This gives me a way to display the same table after user has edited the table or deleted entries from it (editing and deleting takes the user to another page). So, when the user is on a page like the search results and choses an option like edit, generate pdf, or delete then returns to the search result page the same search result previously queried shows up. If, however no session variables were used then after operations like generate pdf, edit or delete the user has to re-query the search.



**8. Use of Libraries**

To generate PDF the TCPDF library is used. Below is an example of how it is used in the case of Urine test table.



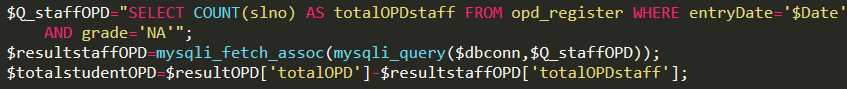


The same library and code are used to generate PDF of search results, opd test tables, medicine reports of any date but the fetch\_data() function is slightly modified in these cases. Also, the headings, titles and tables are modified to meet the needs for the particular cases.

**9. Database Operation**

Throughout my program multiple database operations are used. Some of them are given below.



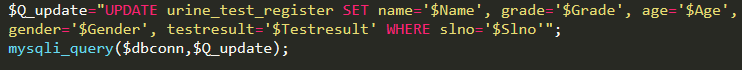


The above code counts the number of “slno” fields that satisfies the conditions that its date of entry is $Date and its grade is “NA”. I have used “slno” field to count because “slno” is given primary key.



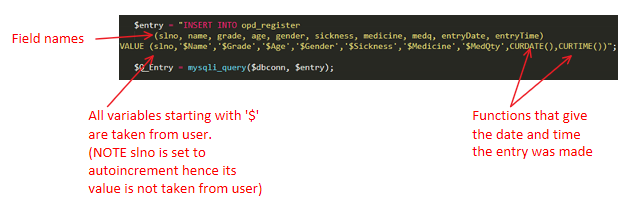
The above code deletes from the table name stored in the variable $tbname. It deletes the entry that satisfy the condition its “slno” field is equal to the number stored in variable $Slno

While editing in Urine test register:



The above code updates the values in the fields (name, grade, age, gender and testresult) of urine\_test\_register table where the slno field has the value specified by the variable $Slno. It updates the values to the values stored in the variables $Name, $Grade, $Age, $Gender, $Testresult, in the respective fields.

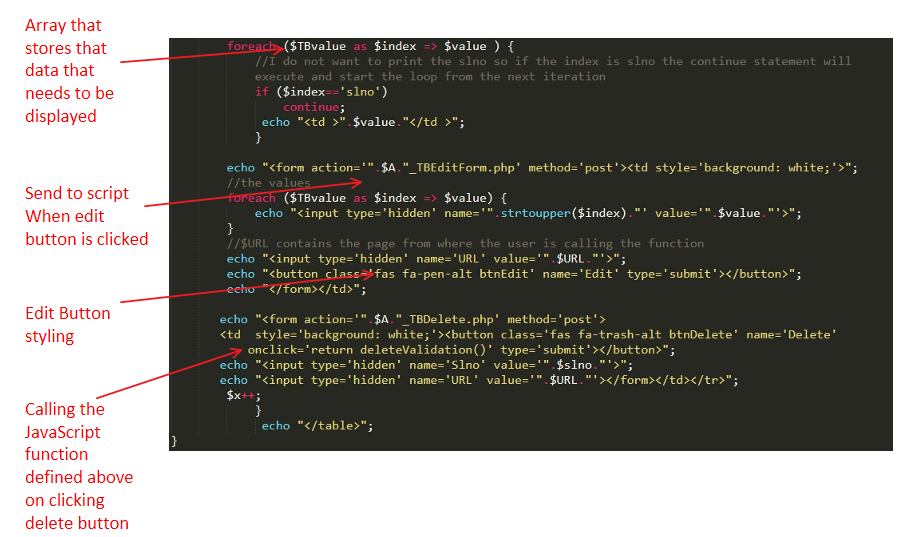
When an input is gotten from user:

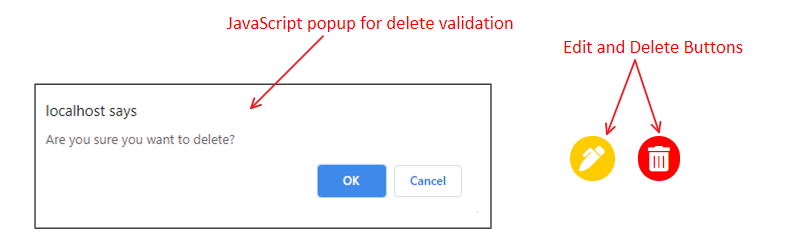


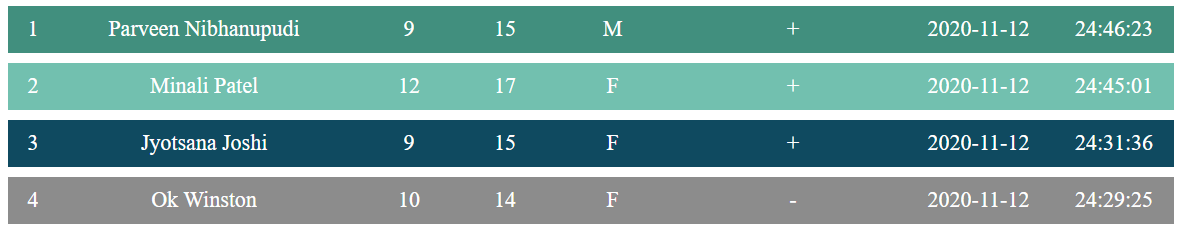
**10. Displaying Tables**

There are a lot of times I need to display something from the database. For example, displaying urine test table, displaying opd search results. The context is different but the basic task to perform is displaying. So, to accomplish this I created a function for displaying, the code is given below:

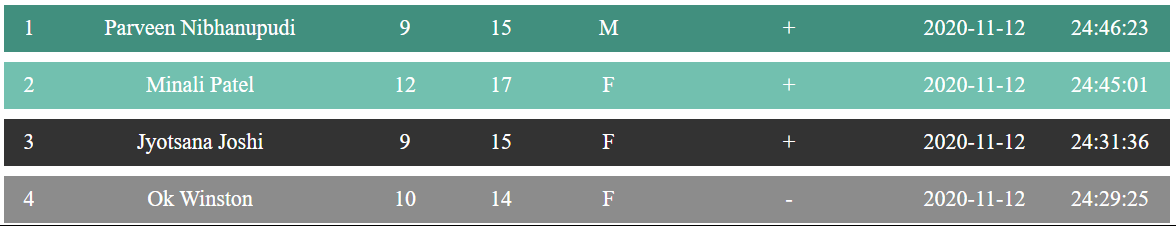








*Colours when not hovering over any row.*



*Colour when hovering over row 3*

The above function is called to display database content in various contexts for example,



The above code displays OPD table.





The above code displays the search result of urine test table.

And all other places that require displaying content from database this function is called with relevant parameters.

**IMPLEMENTATION:**

The plan for implementation is to host the website on school’s server. If that does not work, I will install xampp on client’s computer and run it locally. (see appendix 2 section B6)

**BIBLIOGRAPHY:**

Mraj. “Generate HTML Table Data To PDF From MySQL Database Using TCPDF In PHP ~ SoftAOX.” *Linux, Angular, Angular JS, JQuery, PHP, MySQL and Web Development Tutorials*, Default, 16 Oct. 2020, softaox.info/generate-html-table-data-to-pdf-using-tcpdf-in-php/.

“HTML.” *W3Schools Online Web Tutorials*, www.w3schools.com/.

Skills, Infinite. “Free MySQL Tutorial - Beginner PHP and MySQL Tutorial.” *Udemy*, Udemy, www.udemy.com/share/101sfeAEMbdF1aTXoB/.

“Where Developers Learn, Share, & Build Careers.” *Stack Overflow*, stackoverflow.com/.

Nixon, Robin. *Learning PHP, MySQL & JavaScript: with Jquery, CSS & HTML5*. O'Reilly Media, Inc., 2018.

TrollbrotTrollbrot 99122 gold badges99 silver badges1515 bronze badges, and FitziFitzi 1. “Fancy Css Table Design with Overlaying Colors.” *Stack Overflow*, 1 Oct. 1967, stackoverflow.com/questions/53427469/fancy-css-table-design-with-overlaying-colors.