**APPENDIX 3**

**E1: CLIENT FEEDBACK**

I had tried running a prototype of the website on client’s computer, but it did not go as planned mostly due to the fact that we had to do it virtually. So, to get client feedback I had sent her a video of full functionality.

|  |  |
| --- | --- |
|  |  |
| Sending video of software to client |  |

|  |
| --- |
|  |

Here are the screenshots of the pdf of client feedback:

**FEEDBACK FORM pg 1:**

|  |
| --- |
|  |

**FEEDBACK FORM pg2:**

|  |
| --- |
|  |

**FEEDBACK FORM pg3:**

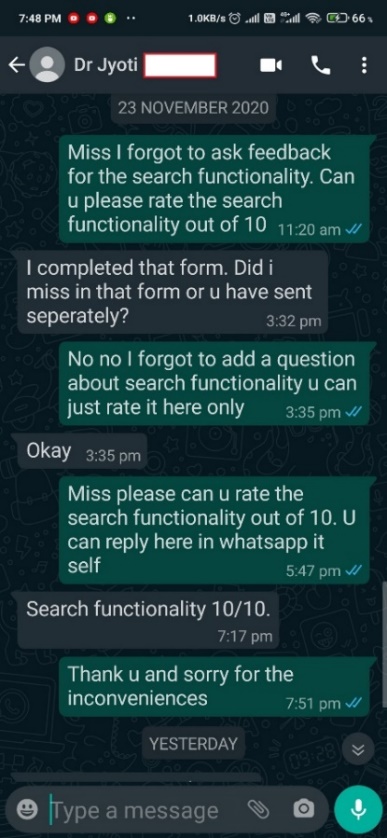
|  |
| --- |
|  |

**FEEDBACK FORM pg4:**

|  |
| --- |
|  |

**FEEDBACK FORM pg5:**

In the form I had forgotten to add a question pertaining to search functionality. So, I simply asked feedback for it in WhatsApp.



**E2: TECHINICALITY OF IMPROVEMENTS**

1. **Multiple Logins:**

To create multiple logins, some login tables need to be created with teacher logins and student login details. Some of the teacher logins will have an administrative account and other will have the normal teachers, student login or parent login. To keep track of this a column in the login tables could be named “type” which will be an integer data type. Now a convention can be adopted by assigning 0 to administrative account, 1 to teacher account and 2 to parent account and 3 to student account. A 4 will be assigned to the doctor’s account. The administrative accounts will be able to make any account an administrative account. Based on the type of account different information relevant to that account will show up.

1. **Home Page Statistics:**

First a function needs to be created that gathers the necessary information into an array that needs to be displayed. Then using JSON format and libraries, graphs can be plotted.

1. **Digital Note:**

A table called “digital note” can be created, it will have columns like name of student, and slno of student (which will be able to identify the student account from the login table), name of teacher, set\_time, arrived\_time, and a column has\_arrived. The teacher will have a page for filling the forms whenever a student is sent to the infirmary, upon submitting the form the set\_time column in the table will be filled with the time when the submit button was clicked. Now the doctor will also have a page that has all the student lists that are going to arrive to the infirmary. The doctor will have an option to set “arrived” to a student, this will set the arrived\_time in the table to the time when “arrived” was clicked. This will also set has\_arrived to yes. There will also be a parallelly running condition which will check the difference in the set\_time for all the list of students for whom the has\_arrived is not set to “yes”. If the difference is passed a certain accepted time an email will be sent (using the email functionality) to the doctor, teacher and student. The has\_arrived will be set to “no”. The doctor can override this and set “yes” if she wants. A pdf of the final list of students who has not arrived will be created and emailed to the administrator at the end of the day.

1. **Delete Functionality:**

To accommodate multiple deletes, first the displayTB function needs to be modified and the delete button needs to be turned to checkboxes. There will be one delete button only and upon clicking it the slno of the checked rows needs to be passed to the delete script. The delete script will now execute the delete query where all the slno that has been passed. There will also be a delete all option which when clicked will run the delete query to delete all the entries for that date.

1. **Dropbox Functionality:**

To improve on the dropbox, two tables can be created: one to old the sickness name and one to hold the medicine names. These could be assigned to an array in the datalist. Now to add or delete a simple editing and adding form can be created much like the Urine test and OPD forms but they can be put in the OPD forms page just beside the form.

1. **Email Function:**

To accomplish this the built-in mail function can be used.



The above mail function can be used inside another function that calls the mail function with appropriate inputs and message created from database.

1. **Special Cases:**

A table for special cases can be created in the and it could function the same as OPD and Urine tests. So, the same code can be modified a bit to accommodate this.

1. **Image Input:**

The image input can be accomplished by having the input type as file in HTML input tag.



This tag can be inside a form tag that posts the input file so that in can be stored in the database.

There is one more aspect of the code that can be improved but it won’t really add to the functionality. To generate PDF of tables I have essentially made the different codes for the different cases for example, to generate PDF of OPD table the script OPD\_PDF.php is used, to generate PDF of search results of Urine test table the script SEARCHUT\_RES\_PDF.php and so on. It would be nice to have one single script to do all this PDF generation for different cases. This will make the code more general and make it more elegant. This is something I haven’t been able to figure out how to do efficiently.