

Contents

ANALYSIS.....	2
Problem Identification	3
Why is it suited to a Computational Solution	3
Stakeholders.....	5
Interview.....	5
Researching existing programs	9
Further meeting with stakeholders outcome:	11
REQUIREMENTS.....	12
Features of the proposed solution	13
Stakeholder Requirements	14
Success Criteria.....	15
ABSTRACTION DIAGRAM	17
DESIGN.....	18
DECOMPOSE THE PROBLEM:	19
INTERFACES: USABILITY FEATURES.....	19
INTERFACE ALGORITHMS AND TEST DATA	25
THE LOG IN INTERFACE DESIGN:	25
FIRST STAFF INTERFACE DESIGN:.....	30
INITIAL STUDENT INTERFACE DESIGN:.....	33
AMMENDED STUDENT INTERFACE DESIGN:.....	36
SECOND STAFF INTERFACE DESIGN:.....	40
THIRD STAFF INTERFACE DESIGN:	44
FINAL STAFF INTERFACE.....	57
STRUCTURE OF THE SOLUTION:	59
Stake holder Input	60
INPUTS, PROCESSES, OUTPUTS AND STORAGE.....	60
KEY VARIABLES AND STRUCTURES	61
Validation Table:.....	63
DEVELOPMENT STAGES:	64
DEVELOPMENT.....	66
STAGE 1.....	67
Login Interface.....	67

First Staff Interface	73
Student Interface:.....	80
Second Staff Interface:.....	87
Third Staff Interface:.....	92
STAGE 2.....	101
Arrange the girls!.....	107
STAGE 3.....	119
STAGE 4.....	122

ANALYSIS



ANALYSIS

Problem Identification

Currently at my school, taster courses for the Year 11s are not computerised and our Saturday administrator (Mrs Jackson) has to do the job by hand. This is very time consuming, which means our Saturday administrator has to waste her time on these tasks that could be simplified with a computer program. The Year 11s need to trial their potential A level subjects, so are given the option to do A level taster lessons on two available Saturdays. Sorting out the logistics for this are difficult as the number of teachers available at each session need to be considered as well as sizes of classes and how to sort the girls so they get all the taster lessons they opted for.

Features required in a computer system that could solve the problem:

- Software that can store a database
 - To store all the requests from the girls and also teacher availability
- Appropriate hardware and operating system to run the software being created on.

Why is it suited to a Computational Solution

This problem lends itself to a computational solution as it requires the manipulation of data that can be stored in tables in a database, and then the output of the processes performed on said data. To put it simply, the program will allow the input of choices from the students and also input of teacher availability and then process the data so that all students will be put with available teachers. The only other way to do this would be on paper, but this has proved inaccurate and also inconvenient. An additional computer program would be very useful as it speeds up the process to almost instantaneous. Finally, the webpages can be accessed from anywhere, making this method even more convenient.

Features suitable for a computer program:

- Input from students can be stored in a database, rather than handed in by hand.
- Database values are able to be written to a language's data structure and thus be operated on accordingly. For example, searched and sorted.
- Then the output can be sent via email to Mrs AE Jackson

Problem Recognition:

- The overall problem is sorting the girls into their taster lessons depending on their ranked choices
- The underlying problem is making sure all the constraints are met when sorting the girls into their lessons (eg max class sizes, number of sessions etc)
- Once this is sorted out, the rest of the solution is straightforward in comparison as they are mainly to do with the interface of the program and are not computationally as taxing.

Problem Decomposition:

This problem can be broken down into several steps to be done.

- Create interface for students to enter choices
- Create interface where teacher sees student choices and can edit
- Algorithm sorts girls into their taster lessons
 - Sorts girls by how they prioritised subjects
 - Checks to see if all constraints have been met, eg minimum and maximum class size
- Outputs result of operation to Saturday Administrator.

Divide and Conquer:

Breaking down the problem solution into these small steps, makes it seem much less daunting. I think I am capable of coding and designing each point I have described above. Bringing all these small steps together and combining them into a functioning solution makes use of the divide and conquer computational method of problem solving

Abstraction:

An aspect of abstraction is picking out the necessary details about the students to use in my algorithm from the information table from which I will be collecting the names of the students. There are many irrelevant details in the table initially, eg middle names, housemistress details etc.

All these steps require either a computer or computation showing how the problem lends itself to a computational solution.

Why valuable?

Will help Mrs AE Jackson place the Year 11s on their taster course in a fraction of the time than she would usually do, which gives her time to also accept last minute changes from the students, and do other tasks that her job requires. Program is efficient and more accurate than previous method. Program will also be able to be used anytime and anywhere as it is web-based.

Stakeholders

Stakeholders that will influence the development of the program:

- Mrs AE Jackson - Saturday Administrator (*primary stakeholder*)
 - All the software iterations will be passed through her as she will be the main user of the program
- Network manager (*secondary stakeholder*)
 - The software will be passed through the network manager at the penultimate stage of development, where the network manager tries to make sure the software can be made a part of the network
 - Need to discuss at the start of the Project also, to decide what best would be compatible with school system.
- Teachers (*secondary stakeholder*)
 - Teachers will be involved in beta testing of the program as they have to put in their availability and timings
- Students (*secondary stakeholder*)
 - Students will use the program to either input their options - beta testing

Interview

Questions for Mrs AE Jackson (primary stakeholder):

- 1) What is your current method of placing the girls on taster courses?
 - a) Are you happy with it?
 - b) What are the disadvantages/ advantages?
- 2) Have you ever tried a computational method of doing the task?
 - a) If so, what was it?
 - b) Was it useful?
 - c) What did you like about it?
- 3) What factors are taken into consideration when sorting the girls (eg size of class)
 - a) How are these factors considered? (e.g can a class have 30 people?)
 - b) How do these factors affect the way the girls are sorted? (eg. if a class has 30 people, what then?)
- 4) For the program, how would you prefer to interface with the software?
- 5) Would you like the students to interface with the software?
- 6) How would you like to feed in information about teachers' timetables, integral to how the girls are placed?
 - a) You will provide it?
 - b) Would you like Teacher's to have an interface to input their data.
- 7) What experience do you have with Computer Programs?
- 8) What input do you think the program will need?
- 9) How would you like the output of the program to be presented to you?
 - a) Via email?

Justification of questions:

What is your current method of placing the girls on taster courses?

This question helps me identify exactly what the stakeholder needs as well as how their current method of solving the problem helps them. By learning about the disadvantages and advantages of her current method, I can adapt my strategy of implementing the solution of this design to a way that best fits her needs. Eg by building on the advantages / using some of the good ways she does it now and avoiding / fixing the issues she is having with the current system.

Have you ever tried a computational method of doing the task?

I am asking this question to gauge her familiarity with computer programs in relation to this job, to figure out the level of user friendliness necessary. And also to see what good I can take from her last program used and how I can build and use useful aspects of the past program in my own as well as what to stay away from due to the preferences of the stakeholders.

What factors are taken into consideration when sorting the girls (eg size of class)

This question solely serves to tell me how the software will run and the prerequisites that need to be considered in relation to the parameters of the functions and procedures within the program. This way, I can get a feel of what the program needs to do and develop my application around this.

For the program, how would you prefer to interface with the software?

This question helps me design the front end of the software in a way that would be easy for my stakeholder to use, and user friendly.

Would you like the students to interface with the software?

Also helps me know how to design front end and gauge the number of different user interfaces necessary for each category of person using the software.

How would you like to feed in information about teachers' time tables, integral to how the girls are placed?

This will help me gauge the complexity of the algorithm required and also help me think about how this data will be stored (data structures) for design purposes and where validation will be required.

What experience do you have with Computer Programs?

Also helps me think about how user friendly the front end needs to be if they are not familiar with computer programs. But if they are, the front end does not have to particularly pretty and a level of complexity of design is lost, which is good.

What input do you think the program will need?

Also gives me insight into what data structures would be best to use and also shows me the factors that need to be considered in the computation in the program.

How would you like the output of the program to be presented to you?

This also tells me how she would like to receive the data and therefore what to code into the program that will make this happen according to the stakeholder's need.

Analysis of the answers I received to the Questions:

I carried out this interview to get a feel of what my stakeholder had been doing before hand and how I can help her through my Project and I think the meeting was very useful to outline what she had been doing before and why that wasn't working for her and how my program would help her.

I arranged a meeting face to face, because I thought this would be more effective in transferring each of our ideas and also because it was not inconvenient as I attend the school and can meet with her during my free periods - we have scheduled further weekly meetings to continue to discuss our ideas for the program.

Mrs AE Jackson is very familiar with computer programs and uses them on a day to day basis.

What she was doing before:

- Girls would input their choices through a Microsoft online form
 - She likes that these forms sent the choices they had chosen back to the user.
 - She does not like how the current form does not allow prioritised ranking
 - Also does not like how the table is imported into excel, and has given me the specific format she wants the form to be imported in, if possible.
 - Also is unhappy with how she has to do validation in regards to whether a Student has picked a subject twice
- What I took from this, is that I need to create an online form that allows ranked choices and takes this information into a table she is comfortable viewing. Also

include validation on the data entry page for the students so they can not input the same subject twice. (*below*)

This is how the data from Microsoft Forms is presented to her:

D9	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	1st choice	2nd choice	3rd Choice	4th choice	5th choice	6th Choice	choice 1	choice 2	choice 3	choice 4	choice 5	choice 6
2	19 Maths Alevel	Latin	Chemistry	Biology	Physics A-level	German	1	1	1	1	1	1	1
3	20 Maths Alevel	Maths IB	Physics A-level	Maths IB Studies	Theory of Knowledge	Computing	1	1	1	1	1	1	1
4	21 Biology	English IB	Chemistry	Theory of Knowledge	History	Spanish	1	1	1	1	1	1	1
5	22 Biology	Chemistry	History	Economics	Spanish	Physics A-level	1	1	1	1	1	1	1
6	23 Chemistry	Maths IB	Biology	Physics A-level	Maths Alevel	Spanish	1	1	1	1	1	1	1
7	24 Economics	French	Maths Alevel	Spanish	Biology	English A level	1	1	1	1	1	1	1
8	25 Biology	Chemistry	Economics	History	Physics A-level	Maths Alevel	1	1	1	1	1	1	1
9	26 Theory of Knowledge	Economics	History	English IB	Chinese	Chemistry	1	1	1	1	1	1	1
10	27 Economics	Government and Politics	History	Maths Alevel	Latin	RS A Level	1	1	1	1	1	1	1
11	28 Maths Alevel	Physics A-level	Computing	Chemistry	Economics	Geography	1	1	1	1	1	1	1
12	29 Chemistry	Physics A-level	French	Italian	Maths Alevel	Geography	1	1	1	1	1	1	1
13	30 Economics	Biology	Chemistry	English IB	Maths IB	History	1	1	1	1	1	1	1
14	31 Chemistry	Maths Alevel	Economics	Physics A-level	Biology	Government and Politics	1	1	1	1	1	1	1
15	32 Theory of Knowledge	English IB	Computing	History	Government and Politics	Economics	1	1	1	1	1	1	1
16	33 Biology	French	Chemistry	Maths Alevel	Spanish	English A level	1	1	1	1	1	1	1
17	34 Theory of Knowledge	English IB	Maths IB	Physics A-level	History	Government and Politics	1	1	1	1	1	1	1
18	35 Government and Politics	Economics	Philosophy IB	RS A Level	Theory of Knowledge	Geography	1	1	1	1	1	1	1
19	36 Chemistry	Biology	Spanish	Latin	Maths IB	History	1	1	1	1	1	1	1
20	37 Biology	History	Chemistry	Physics A-level	Geography	English A level	1	1	1	1	1	1	1
21	38 Chemistry	Maths Alevel	Biology	Physics A-level	Economics	Spanish	1	1	1	1	1	1	1
22	39 Government and Politics	History	Economics	Physics A-level	Spanish	Maths IB	1	1	1	1	1	1	1
23	40 Chemistry	History	Theory of Knowledge	Philosophy IB	Government and Politics	Economics	1	1	1	1	1	1	1
24	41 Chemistry	Biology	Economics	Maths Alevel	Physics A-level	Geography	1	1	1	1	1	1	1
25	42 Chemistry	Spanish	Maths IB	English IB	History	Biology	1	1	1	1	1	1	1
26	43 Government and Politics	Geography	English A level	RS A Level	Spanish	Economics	1	1	1	1	1	1	1
27	44 English IB	History	Theory of Knowledge	Philosophy IB	Government and Politics	Maths IB	1	1	1	1	1	1	1
28	45 Physics A-level	RS A Level	Maths Alevel	Spanish	History	Philosophy IB	1	1	1	1	1	1	1
29	46 Biology	Chinese	Chemistry	Maths IB	English IB	Chemistry	1	1	2	1	1	1	2
30	47 Art	History of Art	Chemistry	Maths Alevel	Economics	Biology	1	1	1	1	1	1	1
31	48 Chemistry	Physics A-level	Maths Alevel	Spanish	RS A Level	Art	1	1	1	1	1	1	1

This is how she wants the data to be presented to her:

AD12	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	un	% of choices entered																																				
2	19	6	1	2	3	4	5	6																														
3	20	6	1																																			
4	21	6																																				
5	22	6																																				
6	23	6	5	1	3	4	2																															
7	24	6	3																																			
8	25	6	6	2	1	5																																
9	26	6																																				
10	27	6	4	5																																		
11	28	6	1	4	2																																	
12	29	6	5	1	2																																	
13	30	6																																				
14	31	6	2	1	5	4																																
15	32	6																																				
16	33	6	4	3	1																																	
17	34	6																																				
18	35	6																																				
19	36	6																																				
20	37	6																																				
21	38	6	2	1	3	4																																
22	39	6	5	1	4																																	
23	40	6																																				
24	41	6	4	1	2	5																																
25	42	6																																				

This data is presented with the Id of the student down the side, the subjects across the top and the numbered choice in the box that links the two.

Exactly what she wants me to do (bare minimum):

- Take the choices of the students and organise them into (hopefully) their Top 4 choices in the 4 taster sessions planned
- When taking options do validation, and also allow their 6th choice to be null

- Will also send girls an email of the choices they have chosen
- Program would potentially access iSAMs database which gives access to the emails, ids and names of all the students for follow up questions
- The factors needing to be considered before doing this are:
 - Maximum class size is 9/10 because she needs to leave space for visitors coming which will bring the class size to the absolute maximum which is 12
 - Subjects that may need two teachers during any two sessions are chemistry, biology, economics and politics - so essentially these will have at least 6 classes available during the time.
 - Suggested that classes with small numbers become a fixed slot as program should minimise the number of teachers given the parameters
- Then wants to be able to access the sorted table.

What I feel like I could help her with:

- All of the Above except for maybe the iSAMs database, not sure how I would go around security and also data protection, but I will be looking further into this
- Extra:
 - Sending emails to girls that have not filled out the form
 - And also display a list of the girls who have currently not replied and have a button that allows her to send automatic emails to either girl, tutor or housemistress - we would know the house mistresses and tutors if I was able to access the iSAMs database.
 - Creating a data input page for her
 - No need for her to interface with actual table of choices data unless she would like to, she will also be able to edit these tables.
 - Or upload her own - image display of the way she should do it.
 - Button that allows her to sort the girls
 - Once girls are sorted, put all the information with actual names instead of ID into a table to be sent to tutors, with room numbers, teachers and timings and dates.
 - Program put each girls into their respective tutor groups
 - Will have the option of sorting out
 - Will also have an option for her to download the table onto her computer

I think I will be able to fully complete the program she wants me to do for her, and will therefore be completing many of the “What I feel like I could help her with” bullet points to help her further and stretch myself to the best of my ability!

Researching existing programs

Finding similar problems and current solutions:

When looking for the type of program I would like to implement, I have to roam the internet under the type of program my program is, which is: matching algorithm / priority allocation algorithm with a graphic interface.

It was very difficult finding an algorithm that fixed a similar problem to the one I was trying to solve, but I found a few general examples that I can take aspects of and implement in my solution. I did not find any actual software that was related to my project but I found that there were many well known algorithmic problems.

Two-Sided Matching Algorithm: Matching mentors to Mentees:<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6413920/>

This algorithm essentially finds a way to match the mentees to the mentors depending on both of their preferences and how they've each ranked each other respectively. It allows the mentees to rank the mentors in order of preference but also gives the option of allowing mentees and mentors to have people they have not ranked, as they may prefer to not get matched rather than being with the person they have not ranked.

My comments: Although this algorithm is related to what I am trying to achieve, mine is different in many ways. My program involves one sided ranking rather than ranking on both parts. In this case, only the rank of the individuals are considered but I would have to consider the rank of subjects from the students as well as class sizes and teacher availability, which in my opinion is slightly more complex. What I think I will be able to take away from this is the feature of not having to rank a person you would not like to partner with. I think this will be useful in my program as for the students' 5th option there will be an option to select no subject, which would simplify the code somewhat, as a spot on another taster lesson would not have to be sought after for these people.

Constraint Satisfaction

Problem:https://en.wikipedia.org/wiki/Constraint_satisfaction_problem

This algorithm as stated in its name is a Constraint Satisfaction Algorithm. Constraint satisfaction problems (CSPs) are mathematical questions defined as a set of objects whose state must satisfy a number of constraints or limitations. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which is solved by constraint satisfaction methods. Example sudoku- considering those that have to be true first to solve the problem - a strategy I plan to implement.

My comments: I think this algorithm is more useful than that stated above in the way it would help me solve the program although not without its own limitations. This solution does not take into account ranking but it does take factors into account when figuring out the constraints. This program considers the constraints and allocates the nodes accordingly and fixing them when necessary. I can do this when allocating girls to unpopular subjects, making constraints that that subject can only run on that day.

Stable Marriage Problem:

This algorithm is also a matching algorithm but with slightly different principles than the first.

Given n men and n women, where each person has ranked all members of the opposite sex in order of preference, marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners. When there are no such pairs of people, the set of marriages is deemed stable.

My comments: This is very similar to the first algorithm, but this one also considers whether at a specific ordering two people would prefer to be in different places. This concept is something I would like to implement in my program as it would give the students the best outcome possible as allocations are switched if necessary. Eg if a girl picked Computer science as option 3 and is not given this option but all the rest including Spanish(which was ranked lower), and somebody else picked Spanish second and got everything else excluding Spanish and including Computer Science (which was ranked lower), these two girls can be swapped for greater happiness.

Microsoft Form:

This is one of the few software programs that I found that could help me. Microsoft forms allow the client to give ranked options.

Hi RB, when you submit this form, the owner will be able to see your name and email address.

1. How do you feel about your knowledge of each of these topics.

1 = "I know nothing"

5 = "I could teach this class"

	Option 1	Option 2	Option 3	Option 4	Option 5
War of 1812	<input type="radio"/>				
Missouri Compromise	<input type="radio"/>				
Compromise of 1850	<input type="radio"/>				
Manifest Destiny	<input type="radio"/>				

My comments: This would be very useful as I need to rank the girls' choices, but the limitation of the Microsoft form is that one can only allow for 10 choices to rank. This is not good as there about 20 options for subjects. I will build on this limitation in my web based platform.

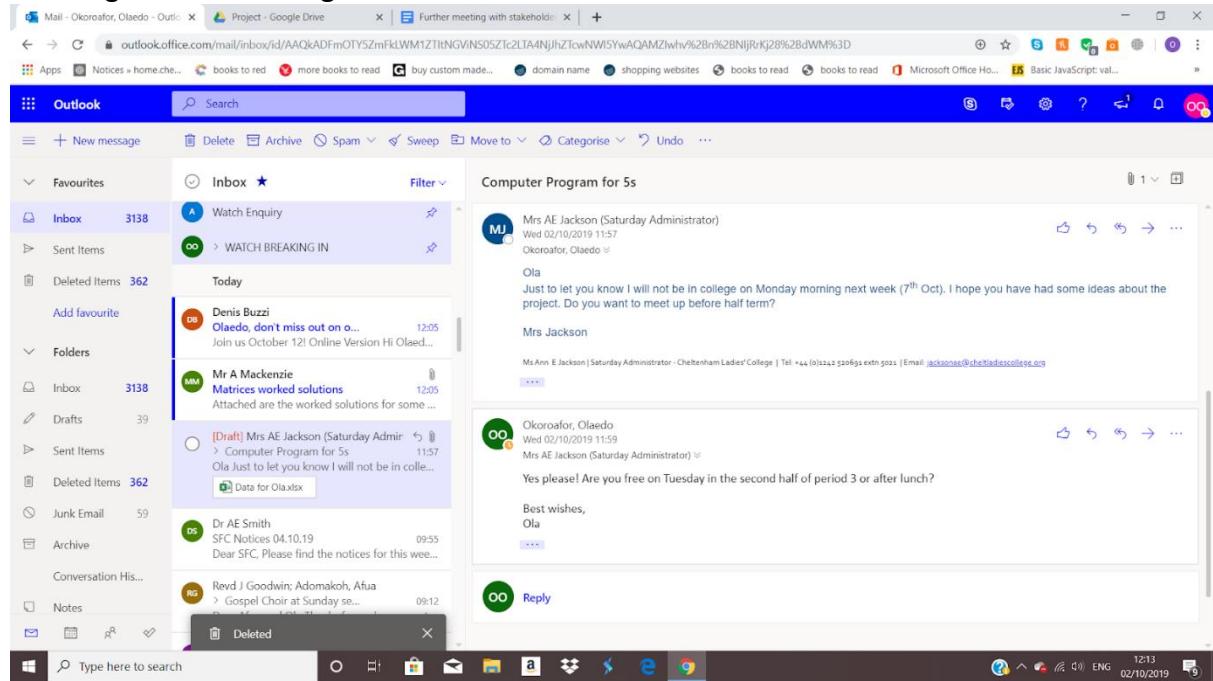
Further meeting with stakeholders outcome:

For my next meeting with my stakeholder, I plan to discuss the following points:

- The web interface I have chosen to implement (I chose myself)

- How the data will be outputted (I chose myself)
- I will also go over the notes from our last meeting to make sure we are on the same page.
- I will go over the research I conducted with her and discuss the implementations I plan to use and also discuss limitations previously discussed.
- Draw up list of user requirements with her.

Meeting has been arranged:



Notes from meeting:

- Mrs AE Jackson is fine with most of the requirements and also happy about the extra things I suggested for the program
- Mrs Jackson would rather input a list of subject choices than have me hard code them into the program for ease of use in the future - subject options change.
- Would prefer there was a livelink to iSAMs that connected the table of girls with interface.
- Also thinks it would be useful to split first name, last name and preferred name into different categories. This would be helpful for when she needs to contact a girl.

REQUIREMENTS

Hardware

For me:

- A laptop with an OS able to run applications (preferably not a chromebook), ie the development software for php ad mysql
- Computer does not have any other requirements other than the standard peripherals, eg a mouse

For client:

- A device that can access the internet with either a mouse or keyboard to navigate the system
- Device OS must allow downloads from the web or have enough storage for the table to be downloaded
- There are no specific hardware requirements in terms of CPU speed or amount of RAM.

Software

For me:

- MAMP - virtual server etc
 - A solution stack composed of free and open-source software running with proprietary commercial software, to run dynamic web sites on computers running macOS or Windows.
- Text editor - Notepad
 - To code my front end and also potential back end
- Microsoft software
 - To utilise Microsoft forms to potentially collect information from student choices
- MySQL - to make and manipulate databases
- Windows Linux or Mac operating system
 - The OSs allow for ease of coding in javascript and php

For client:

- Web browser
 - Javascript interpreter integrated into this for front end processing.

[Features of the proposed solution](#)

Initial concept of my solution considering this research and meeting with stakeholder:

The algorithms unfortunately did not give me any guidance on how my front end interface will look like but from the meeting with my stakeholder and advice from my computing teacher, a web based interface is what I have found to be best as that is the language I can code with strongest. The web interface will take in a table and do the processes the data accordingly, with the use of the concepts mentioned in the example algorithms. Then output the information in a spreadsheet on the website that can be downloaded by the stakeholder or sent by email. There will also be a web interface where the students can input their choices and this will be used in the table mentioned prior.

Limitations of my solution:

My solution is limited in the sense that it can not be used for a wide variety of purposes, it can only be applied in this specific situation because all the factors to be taken into consideration are hard coded into the program rather than taken in through user input to allow the program to be more diverse in its applications. This is not ideal, but will also make the coding process much longer and more complex with this additional factor to be considered.

Another limitation of my program is that it will not take in the teacher's and their availability prior to sorting out the groups of girls, by stakeholder's choice, but this removes an added layer of complexity which would have also extended the development period of this program.

Stakeholder Requirements

DESIGN

Requirement	Explanation
Clear staff interface	A web interface that is easy to use and where the buttons are clearly laid out.
Student Interface	A web interface that is also easy to use where students can enter their choices and pick their names.
Clear instructions for Student Interface	There will still be input validation if needed but instructions need to be clear to avoid erroneous input into the program

FUNCTIONALITY

Requirement	Explanation
Put the girls into appropriate taster lessons	This is the main function that the program will perform
Validation / Sanitisation of student input	This will make sure no student has put in 1 subject twice or just not entered options in required sections. Will ask them to reenter their data. This will probably be done using javascript as it is client side processing.
Login interface	This will be for Miss Jackson, so that she is the only one that can access this interface.
Upload iSAMS table and pull data from table to get list of students. (<i>iSAMS is my school's data management system</i>)	There must be an option on page to upload table, and then the sql behind the web interface will pull out just the names of the students for the student interface.
Display girls that have responded to survey in table on staff interface as well as a list of girls who haven't.	The data from the student interface will be sent to the staff interface to be displayed to Mrs Jackson so she can see the data so far.
Allow Mrs Jackson to edit table of choices.	Mrs Jackson will be able to change any options for the girls if necessary.

Mrs Jackson will be able to enter a list of subjects	These subjects will be displayed as options on the Student interface that they can pick from.
Mrs Jackson can then choose to arrange the girls.	Button for this function should be clearly placed, triggers the arrangement algorithm.
Display of list of arranged girls	Once the program to arrange the girls has been run, it will display list of girls in their Saturday options clearly.
Allow Mrs Jackson to download table from site.	There should be a button which allows Mrs Jackson to download table onto the PC she is using.

Limitations

- Will not be able to safeguard against girls' picking the wrong name from the list
- Does not take in any data about teacher availability which would have made the arrangements more specialised

Good to include

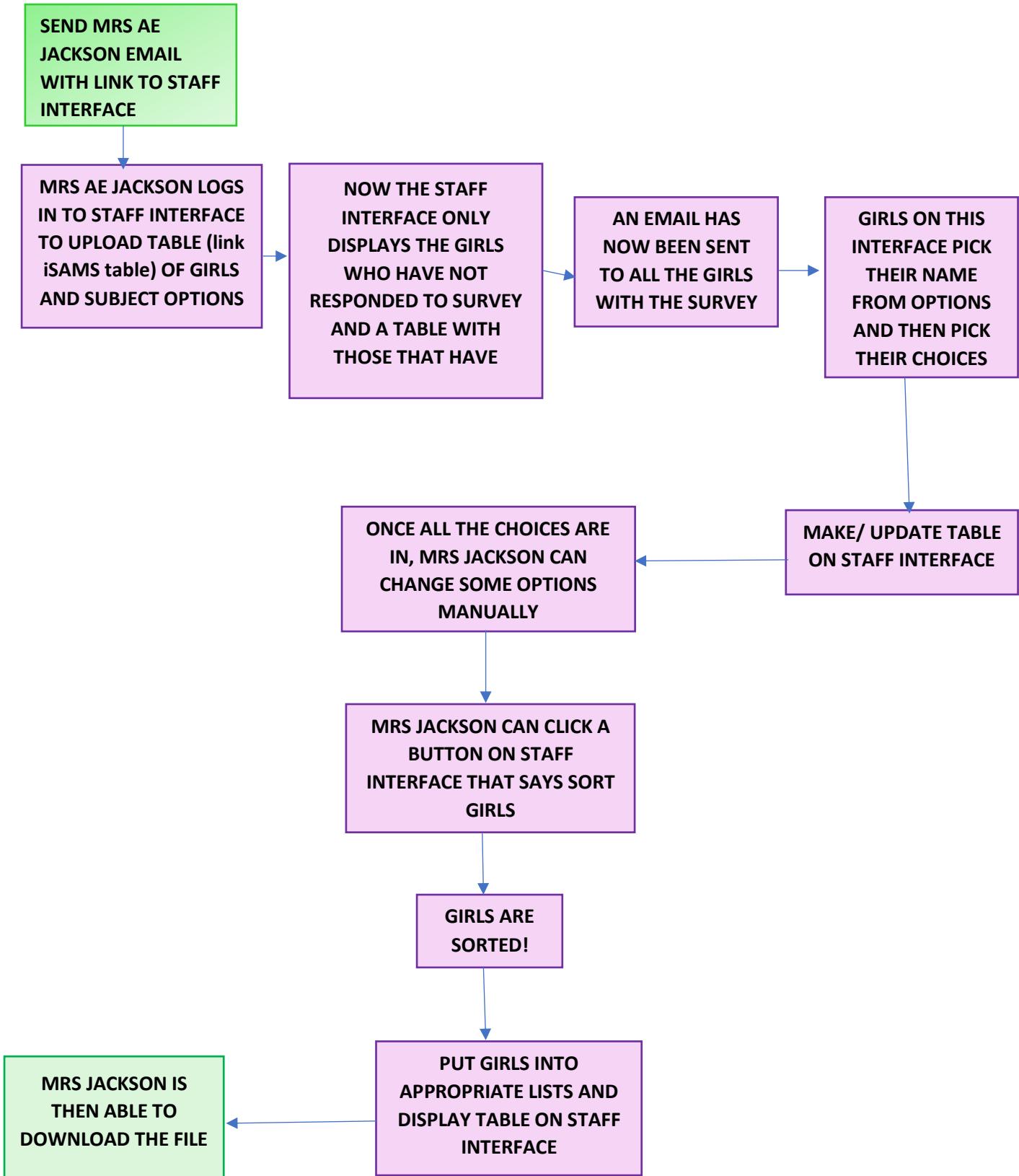
- Automatically send emails to girls who haven't filled in the form after a specific amount of time.
- Also send the finished lists to tutors with the times, dates and teachers for the Saturday sessions

Success Criteria

Criteria	How to evidence
Main window for staff interface	Screenshot showing all of the staff interface showing it loads up correctly
Clear and Simple design for both interfaces	Screenshot of the interfaces including big buttons and clear instructions on interface
Main window for student interface	Screenshot of student interface showing it loads up correctly.
Validation and Sanitisation of user input	Show the code that facilitates this as well as the testing evidence, with erroneous and borderline test data.
Login for staff interface	Screenshot for the log in input boxes as well as code that facilitated this and testing evidence to show it works and has validation.
Options of girls names show up to pick from.	Show that a drop down list of names is included, taken from the iSAMS table using a screenshot of screen as well as sql commands.
Options of subjects to show up to pick from	Show that a drop down list of subjects is included, taken from the list inputted by Mrs Jackson, using a screenshot of screen as well as sql commands.

Staff interface allows for staff to enter a list of subjects	Screenshot of this with tested data as well as a screenshot of student interface with these implemented.
Link to student interface sent to Staff after above command.	Screenshot of the sent email as well as code that made this happen
Display of girls who haven't responded to survey.	Screenshot of this on staff interface as well as code that facilitated this on student interface.
Display of girls who have responded in table layout	Screenshot of table on staff interface
Staff can edit the above table	Screenshot of before and after it has been edited, as well as a screenshot in the my sql table to show it has saved.
Choice at the bottom of the page to sort girls.	Screenshot button that says sort girls
Arrange the girls into taster sessions	Code that facilitates as well as test evidence with appropriate test data
Displays table of sessions to the user	Screenshot of table
Download button on Staff interface once table of sessions is ready	Screenshot of button
Downloads file onto client's pc	Code that facilitates this as well as testing evidence with screenshot evidence that it opens correctly

ABSTRACTION DIAGRAM



DESIGN

DESIGN

DECOMPOSE THE PROBLEM:

I will be using my Abstraction diagram as the means of decomposition as it decomposed my problem as well!

INTERFACES: USABILITY FEATURES

LOG IN INTERFACE

Username:

Password:

LOG IN

FIRST STAFF INTERFACE

UPLOAD LINK TO STUDENT DATA

Input the Subjects one at a time!

ADD

Remove subject.

REMOVE

Here there will be a display of the subjects inputted into the textbox on the left. So this box will pop up after the button SUBMIT is clicked. Or this box will be continually be populated after the ADD button is clicked. Format will be something like:

- Chemistry
- Physics
- Biology
- Maths
- Latin IB
- Philosophy IB
- Economics
- ETC

SEND QUESTIONAIRE TO GIRLS

This interface is the progression from the last one, when not all the girls have filled out the survey.

Second Staff Interface

CURRENTLY AWAITING RESPONSES

Current Responses:

*where the numbers represent the rank of that subject

ID	Biology	Chemistry	Computing	Geography	Economics	Politics	TOK
004	6	5	1				4
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample

Awaiting Responses From:

Name	Surname	Email
Morgan	De Viliers Kuun	kuunm@cheltladiescollege.org
Iona	Clowes	clowesi@cheltladiescollege.org
Jeanna	Junady	junadyj@cheltladiescollege.org
Lorne	Rolinson	rolinsonl@cheltladiescollege.org
Evelyn	Nutton	nuttone@cheltladiescollege.org
Ivy	Manna	mannai@cheltladiescollege.org

[Send Email Reminder](#)

*Arrange girls without those waiting response

[NEXT PAGE](#)

Page when all the responses are in!

Third Staff Interface

ALL Responses:

*where the numbers represent the rank of that subject

Edit Table

ID/ Name of person you would like to edit

ANSWER

Subject you would like to assign number

ANSWER

Number you would like to assign to that Subject

ANSWER

All changes made in the Edit Table portion will be immediately updated to the screen.

UPDATE TABLE

ARRANGE GIRLS INTO SESSIONS

ARRANGED GIRLS

Session 1:

Subject	Name	Name	Name	Name	Name	Name	Name	Name	Name	Name
Biology	Ola	Morgan	Lorna	Jeanna	Iona	Ivy	Evie	Andrew	Alex	

Session 2:

Subject	Name	Name	Name	Name	Name	Name	Name	Name	Name	Name
Biology	Ola	Morgan	Lorna	Jeanna	Iona	Ivy	Evie	Andrew	Alex	

Session 3:

Subject	Name	Name	Name	Name	Name	Name	Name	Name	Name	Name
Biology	Ola	Morgan	Lorna	Jeanna	Iona	Ivy	Evie	Andrew	Alex	

Session 4:

Subject	Name	Name	Name	Name	Name	Name	Name	Name	Name	Name
Biology	Ola	Morgan	Lorna	Jeanna	Iona	Ivy	Evie	Andrew	Alex	

OR

Name	Session 1	Session 2	Session 3	Session 4
Ola Okoroafor	Computing	Maths	Chinese	Physics

Student Interface

Name:

SUBJECT CHOICE:

FIRST

SECOND

THIRD

Responses are ranked!

FOURTH

FIFTH

SIXTH

Pupil will only be able to access this page once, and once it has been accessed and the form has been submitted, this page below:

**THANK YOU FOR SUBMITTING YOUR CHOICES,
THEY ARE BEING PROCESSED.**

If there are any changes you wish to be made to your existing choices, please contact Mrs AE Jackson at JacksonAE@cheltladiescollege.org

INTERFACE ALGORITHMS AND TEST DATA

THE LOG IN INTERFACE DESIGN:

The diagram illustrates a simple login interface. It consists of a large outer frame with a black border. Inside, there is a smaller frame with a red border. On the left side of this inner red-bordered area, the word "Username:" is centered above a light red rectangular input field. On the right side, the word "Password:" is centered above another light red rectangular input field. In the bottom right corner of the inner red-bordered area, there is a small white rectangular button with a black border containing the text "LOG IN".

This Web Page is for Mrs AE Jackson specifically as it allows her to log in and from here, she will be directed to the next page where she will be able to upload the student table as well as list the subjects available. This web page is necessary as it authenticates the identity of Mrs Jackson so that if the link is spread further, she will be the only one able to log in and access this interface's features.

Links to the success criteria:

- Log in for Staff Interface

Usability Features:

- I have used a text box for the input box for both the username and password. The password box will show dots instead of what Mrs Jackson is typing for maximum security. The LOG IN button also makes it clear what Mrs Jackson should click next. The design is very standard and common to make it easier to use.

Design Template for the above interface: ALGORITHM

Since the system is only designed for one user, table will only have log in information for Mrs Jackson, therefore no need to allow registration.

I would need to create and populate the username and password table with this command though:

```
CREATE TABLE users (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

This creates the table users, then automatically sets the id to autoincrement and adds the fields username, password and time created.

```
INSERT INTO users (id, username, password, created_at)
VALUES (1, JacksonAE, Project123, null);
```

My SQL Table will then look like:

			id	username	password	created_at
<input type="checkbox"/>			1	JacksonAE	Project123	NULL
<input checked="" type="checkbox"/>						
<input checked="" type="checkbox"/>						
<input checked="" type="checkbox"/>						

Now for the actual log in web page algorithm:

Create a configuration file to link the table to the php page for the web page.

```
procedure ConfigFile()
```

```
Create config file with name "config.php"  
Connect to MySQL Database  
end procedure()
```

This then allows me to use my config file when referencing the password and username as it is my connection to the system.

Now for the actual log in stuff:

```
procedure loginPHP()  
    if userloggedin = True:  
        redirect to nextPage  
        connect to "config.php"  
  
    // Initiate variables and collect user input  
    username = ""  
    password = ""  
    username = input()  
    password = input()  
  
    // After user has submitted form we do validation to authenticate  
    // Check if username or password is empty  
    if username is empty or password is empty:  
        OUTPUT("Please enter username or password.") to screen;  
    else:  
        continue  
    // Validate credentials  
    // Prepare a select statement  
    sql = "SELECT id, username, password FROM users WHERE  
    username = ?";  
  
    // Bind variables to the prepared statement as parameters from MySQL  
    // table  
    bind_param(MySQL_users)  
    // Set username parameter  
    param_username = username  
    param_id = id  
    param_password = password  
    // Check if username exists, if yes then verify password
```

```

        if username in users:

// verify password for said username

            if verify(username, password) = True:

// Password is correct, so start a new session

                Loggedin = True

                redirect to nextPage

            else:

// Display an error message if password is not valid

                OUTPUT("The password you entered was not
valid.") to screen

            else:

// Display an error message if username doesn't exist

                OUTPUT("No account found with that username.") to
screen

            else:

                OUTPUT("Oops! Something went wrong. Please try again
later.")

        Close connection

end procedure

```

How the procedures work together:

ConfigFile is used within the login. These two files on the next page are constantly referenced on the subject page, to see if the user is logged in.

Test Plan:

Actual password and username will be JacksonAE and Project123

Test Data	Relation to success criteria	Expected Output
Username:JacksonAE Password:	Login for Staff Interface	Need to input Password
Username: Password:Project123	Login for Staff Interface	Need to input username
Username:jacksonae Password:Project123	Login for Staff Interface	Username not in database
Username:JacksonAE Password:project123	Login for Staff Interface	Incorrect password
Username: Password:	Login for Staff Interface	Need to input username

Username:me Password:me	Login for Staff Interface	Username not in database
Username:Project123 Password:Project123	Login for Staff Interface	Username not in database
Username:JacksonAE Password:JacksonAE	Login for Staff Interface	Incorrect Password
LOG IN BUTTON CLICK	Login for Staff Interface	Go to welcome page
LOG OUT BUTTON CLICK	Login for Staff Interface	Go back to login page

FIRST STAFF INTERFACE DESIGN:

Here there will be a display of the subjects inputted into the textbox on the left. So this box will pop up after the button SUBMIT is clicked. Or this box will be continually be populated after the ADD button is clicked. Format will be something like:

- Chemistry
- Physics
- Biology
- Maths
- Latin IB
- Philosophy IB
- Economics
- ETC

In this webpage Mrs Jackson will put in the link to the newest version of the 5s table. She will also be provided a text box to enter in the subject options for that year. There will also be an extra box that allows Mrs Jackson to remove elements that were a mistake. Alongside these text box will be the bullet points of Subjects currently entered. All the information collected here will be collected by the server for use in the student interface. Once subjects have

been sent in through button to submit questionnaire, Mrs Jackson will not be able to revisit this page.

Links to success criteria:

- Main window for staff interface
- Clear and Simple design
- Staff interface allows for staff to enter a list of subjects
- Link to student interface sent to Staff after above command.

Usability Features:

- Should probably be an option to remove subject choices, but I'm not sure how that would work
- Clear instructions to help Mrs Jackson know what comes next
- Buttons are clear and visible

Upload link to student data:

```
studentTable = input("")  
post link to studentInterface
```

First subject text box algorithms:

```
procedure addSubject()  
    var text = input() // from webpage  
    var subjectli = [] // would be defined outside the procedure  
    if text != "":  
        li.append(text.element)  
    OUTPUT(subjectli) // to webpage  
end procedure
```

Explanation:

This procedure would run every time the ADD button was clicked, and would append anything in the box to the subject list that will be saved as fields in the table for subject choices

```
procedure removeSubject()  
    toRemove = input() //from webpage  
    subjectli.remove(toRemove)  
    OUTPUT(subjectli) // to webpage  
end procedure
```

Explanation:

This procedure would run every time the REMOVE button was clicked, and would remove anything entered in the box from the list of subjects.

After the list has been finalised and Mrs AE Jackson has clicked the button to send the questionnaire to the pupils. This statement will run:

```
procedure EmailandFile()  
    studentemails = sql command(SELECT Emails FROM studentTable)  
    message = ("Fill out this link!", link)  
    mail(studentemails,message)  
    //create file  
    file.write(subjectli, "subject.csv")  
end procedure
```

Explanation:

This command will send the list of subjects to be displayed as options for the pupils as well as create the table for the student choices and will be ready to be filled with data once the girls have submitted their options.

How the procedures work together:

The add and remove subject functions are interlinked but work independently on their joint subject list. The subject list is saved to a file that can be accessed on any webpage on the server. The emailandfile procedure sends out the email without interacting with any other sub routine, but the emailandfile uses the subjectli from the other procedures to save the list.

Test Plan:

For the add and remove functions, Mrs Jackson should be able to enter anything but spaces.

Test Data	Relation to success criteria	Function Name	Expected Output
Any string	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should add string to the list on the screen and table.
Empty space	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should not add anything to the list or table.

A string in the list	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Should remove element in list and table.
A string or empty space not in list.	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Does not remove anything from list or table.
Click the button	Staff interface allows for staff to enter a list of subjects.	EmailandFile	List of subjects is saved to csv file, should be the same as that entered by Mrs Jackson
Click the button and list should be saved in table and list and emails sent to students.	Email students	EmailandFile	List should be saved in table and list and emails sent to students.
Upload link – (additional component)	n/a	n/a	File uploaded should be the same as the students csv file in htdocs.

INITIAL STUDENT INTERFACE DESIGN:

Name:

SUBJECT CHOICE:

FIRST

SECOND

THIRD

Responses are ranked!

FOURTH

FIFTH

SIXTH

SUBMIT CHOICES

On this interface, the pupils will be able to choose their name from a drop down list and also choose their subject choices. This page would involve user validation. They will not be able to access this site once they have submitted their choices and they will be sent to a finishing page that asks them to contact Mrs Jackson for queries.

Usability features:

- Selection from a drop down list of subjects is very clear and simple for the students to use.
- Them picking out their names from dropdown list taken from Mr Wain's database is also much more standardised as there will be no issues of standardisation as they might enter their own names differently from the way they appear on the system.
- Clear buttons and instructions that ease the students through the form on this page.

Links to success criteria:

- Main window for student interface
- Validation and Sanitisation of user input
- Options of girls names show up to pick from.
- Options of subjects to show up to pick from

Algorithms:

DROP DOWN NAME:

```
studentLi = []  
procedure dropname()  
    connect to studentTable  
    nameLi = list(sql command(SELECT Name FROM studentTable))  
    sName = from drop down nameLi(input("Please select your name."))  
end procedure
```

Explanation:

Provides a drop-down list of names to scroll through, and the names are taken from the student table using the appropriate sql command.

DROP DOWN SUBJECT:

```
procedure dropssubject()
```

```

receive subjectli from staffInterface // url would be specified
choice1 = from drop down subjectli(input())
choice2 = from drop down subjectli(input())
choice3 = from drop down subjectli(input())
choice4 = from drop down subjectli(input())
choice5 = from drop down subjectli(input())
choice6 = from drop down subjectli + "blank"(input())
end procedure

```

Explanation:

Provides a drop-down list of subjects to scroll through, and the subjects are taken from the list we created for Mrs Jackson. There is a blank included as an addition to the normal list of options for choice 6 as Mrs Jackson wanted them to be able to choose blank instead of a subject they really did not want to do.

SUBMIT CHOICES:

```

procedure studentvalid()
    choiceli = [choice1, choice2, choice3, choice4, choice5, choice6]
    current = choiceli
    current.insert(0, sName)
    studentLi.append(current)
    connect to studentChoices
    if (choice1 != empty) and (choice2 != empty) and (choice3 != empty) and (choice4 != empty) and (choice5 != empty) and (choice6 != empty):
        if choiceli has no repeating elements:
            for i=0 to 5:
                sql command(UPDATE studentChoices SET choiceli[i] = i+1 WHERE name = sName)
            //send to next page
            link(sample url)
        else:
            OUTPUT("Make sure all your choices are different")
    else:
        OUTPUT("Please fill out all options")

```

```
end procedure
```

Explanation:

Stores the choices in a list that will then be validated to make sure that all the values were actually entered and there were no blanks. The validation also makes sure that none of their choices are the same. Once their choices are validated they are stored in the MySQL database – studentChoices. In this interface, girls are stored like below:

ID	Biology	Chemistry	Computing	Geography	Economics	Politics	TOK
004	6	5	1				4
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample	Sample

AMMENDED STUDENT INTERFACE DESIGN:

The way in which I decided to display the studentChoices table would be inconvenient to code, and it would make the code much more efficient if the columns for this table where ID, Choice1, Choice2 etc.. In addition, I initially planned on using the student's actual names for the list they were able to pick their names from to submit their choices but then I realised that a student could pick another pupil's name, which is not ideal. Instead, I will allow the students to pick from a list of usernames, as the usernames are specific to them. I decided against using the students' Ids as these are how they will be stored in the table, because most student's will not know their Student Id. Therefore we allowed them to choose themselves from a list of secondary keys. Finally, I will be adding all the girls who have completed the list into a file named "completed", so that I can see who has answered the survey and thus deduce who has not.

Below are the ways I have adapted my pseudocode algorithms to fit the new method of implementation.

Algorithms:

DROP DOWN NAME:

```
studentLi = []
studentTable = []
procedure dropname()
    studentfile = file.ropen("student.csv") //puts into a 2d list
    for element in studentfile:
        studentTable.append([element])
```

```

//assuming the 2nd element in the list will be the username
for element in studentTable:
    studentLi.append(element[1])
//display studentLi as the list of elements in html
uName = input()
end procedure

```

Explanation:

Provides a drop-down list of names to scroll through, and the usernames are taken from the student table 2d list that has been converted from csv.

DROP DOWN SUBJECT:

```

procedure dropsubject()
    receive subjectli from staffInterface // url would be specified
    choice1 = from drop down subjectli(input())
    choice2 = from drop down subjectli(input())
    choice3 = from drop down subjectli(input())
    choice4 = from drop down subjectli(input())
    choice5 = from drop down subjectli(input())
    choice6 = from drop down subjectli + "blank"(input())
end procedure

```

Explanation:

Provides a drop-down list of subjects to scroll through, and the subjects are taken from the list we created for Mrs Jackson. There is a blank included as an addition to the normal list of options for choice 6 as Mrs Jackson wanted them to be able to choose blank instead of a subject they really did not want to do.

Student choices table has already been created by me through mysql Database with the Choices as the column names. The following function will populate external table.

SUBMIT CHOICES:

```

//completed file made prior
completedf = file.open(completed.csv)
completed = list(completedf)

```

```

file.close()

procedure studentvalid()
    connect to studentChoices

    if (choice1 != empty) and (choice2 != empty) and (choice3 != empty) and (choice4 != empty) and (choice5 != empty) and (choice6 != empty):
        if choiceli has no repeating elements and uName not in completed:
            for element in studentTable:
                if element[1] = uName:
                    uName = element[0]//assuming id is first
                    //populate table with given info
                    sql command(INSERT INTO studentchoices (StudentID, Choice1, Choice2, Choice3, Choice4, Choice5, Choice6)
VALUES (uName, 'choice1','choice2', 'choice3', 'choice4','choice5', 'choice6');")
                    //add their name to the completed csv file
                    completed.append(uName)
                    completedf = file.wopen()
                    completedf.write()
                    //send to next page
                    link(sample url)

            else:
                OUTPUT("Make sure all your choices are different")

        else:
            OUTPUT("Please fill out all options")

    end procedure

```

Explanation:

Stores the choices in a list that will then be validated to make sure that all the values were actually entered and there were no blanks. The validation also makes sure that none of their choices are the same and a girl that has already submitted her choices can not submit again. Once their choices are validated they are stored in the MySQL database – studentChoices. In this interface, girls are stored like below:

StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6
004	Computing	Maths	Chinese	Philosophy	Geography	
006	Biology	Maths	Computing	Spanish	Physics	Chemistry
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample

Once the studentvalid procedure has run, the students will be taken to the page below:

THANK YOU FOR SUBMITTING YOUR CHOICES, THEY ARE BEING PROCESSED.

If there are any changes you wish to be made to your existing choices, please contact Mrs AE Jackson at JacksonAE@cheltladiescollege.org

How the procedures work together:

The procedures for the drop down name and dropdown subject use files from the php server to display the subject choices and name options. Eg the students file and the subject file. The studentvalid procedure then interacts with the studentchoices table on the php myadmin database. The studentvalid procedure uses the input from the drop down subject and name procedures. None of the procedures on this webpage interact with any other page. The input taken from this page will change the appearance of the second and third staff interfaces.

Test Plan:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output
Be able to pick from list of names.	Options of girls names show up to pick from.	DROP DOWN NAME	List of names from student csv
Be able to pick from list of names.	Options of subjects show up to pick from	DROP DOWN SUBJECT	Should display list of subjects entered by Mrs Jackson
Click button submit with correct data		SUBMIT CHOICES	Submission into mysql database
Submit with empty subject choices	Validation of user input	SUBMIT CHOICES	Error message

Submit with empty name choices	Validation of user input	SUBMIT CHOICES	Error message
Submit with subjects that are the same	Validation of user input	SUBMIT CHOICES	Error message
Submit with name that has already been used.	Validation of user input	SUBMIT CHOICES	Error message

SECOND STAFF INTERFACE DESIGN:

This interface is the progression from the last one, when not all the girls have filled out the survey.

CURRENTLY AWAITING RESPONSES

Current Responses:

StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6
004	Computing	Maths	Chinese	Philosophy	Geography	
006	Biology	Maths	Computing	Spanish	Physics	Chemistry
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample	Sample	Sample

Awaiting Responses From:

Name	Surname	Email
Morgan	De Viliers Kuun	kuunm@cheltladiescollege.org
Iona	Clowes	clowesi@cheltladiescollege.org
Jeanna	Junady	junadyj@cheltladiescollege.org
Lorne	Rolinson	rolinsonl@cheltladiescollege.org
Evelyn	Nutton	nuttone@cheltladiescollege.org
Ivy	Manna	mannai@cheltladiescollege.org

[Send Email Reminder](#)

*Arrange girls without those waiting response

NEXT PAGE

On this page, the girls who have submitted their options are presented in a table showing their names and the options they have picked. There will also be another table displaying the names of the pupils who have not responded to the survey, with their email addresses provided alongside, so it is easy for Mrs Jackson to chase them up.

Links to success criteria:

- Display of girls who haven't responded to survey.
- Display of girls who have responded in table layout

Usability features:

- Buttons are clear and help Mrs Jackson through the process
- Offers to send email reminders to students, reduces work for Mrs Jackson
- Also allows Mrs Jackson to pass on to the next page without all the girls having responded to the survey, where she can edit the values the girls have put in.

Current Response Table:

```
procedure currentResponses()
    connect to studentChoices
    response = sql command(SELECT * FROM studentChoices)
    OUTPUT(response) in table
end procedure
```

Explanation:

This queries the MySQL database storing the information needed to be displayed in the table depending on whether they have responded to the

survey or not. Then displays the columns with the records pulled out of the table.

Awaiting Response Table:

```
procedure nonResponses()

    nresponse = []
    completedf = file.ropen("completed.csv")
    completedli = list(completedf)
    studentfile = file.ropen("student.csv") //puts into a 2d list
    for element in studentfile:
        studentTable.append([element])
    //assuming the 2nd element in the list will be the username
    for element in studentTable:
        studentLi.append(element[1])
    for i in range(0, studentLi):
        if studentLi[i] in completedli:
            //DO NOTHING
        else:
            nresponse.append(studentLi[i])

    OUPUT(nresponse)
end procedure
```

Explanation:

Uses the completed file that is on the server to deduce who out of the rest of the students has not submitted their options.

Send Email Reminder:

```
procedure sendEmail()

    message = "Please respond to the survey: link"
    mail(nresponse,message)
end procedure
```

Next Page:

```
//go to next webpage
link(sample url)
```

How the procedures work together:

The procedures for the current responses and the awaiting responses interact with a table on the server and a file on the server respectively. The awaiting response procedure does not interact with the current responses input nor output. The send email reminder procedure uses the awaitingresponses list generated in the awaiting responses procedure as recipients of the email. These procedures do not affect anything on any other webpages. The send email reminder procedure is also self-contained and executes without interacting with any other webpages.

Test Plan

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output
Leave Ola, Alexander in the completed csv file.	Display names of whom we are awaiting responses.	Awaiting Response Table	Morgan, Iona, Jeanna, Ivy, Lorna, Andrew
SEND Reminder button is clicked.	Email reminders – usability features*	Send Email Responses	Reminders are emailed to all those who have not filled out the survey.
Displays studentchoices table.	Display of girls who have responded in table layout.	Current Response table.	Should be the exact same table as in mysql table.
Next Page button clicked with awaiting responses from girls	Allows Mrs Jackson to pass onto the next page – usability feature*	Next Page	Take the user to staffinterface3

*not in success criteria as it is an additional feature.

THIRD STAFF INTERFACE DESIGN:

ALL Responses:

Edit Table

ID/ Name of person you would like to edit

ANSWER

What choice would you like to change?

--

What subject would you like to assign this choice!

ANSWER

UPDATE TABLE

Please make sure that none of the girls that have been changed, choose the same subject twice or have the same rank for more than one subject.

All changes made in the Edit Table portion will be immediately updated to the screen.

ARRANGE GIRLS INTO SESSIONS

On this page, all the girls will have submitted their responses or Mrs Jackson will have clicked to continue without the responses of the remaining girls. She will be able to enter and change values in the table incase a girl has put in her choices wrong. To reference the table, she must state the ID of the girl she wishes to change, the subject to change and the number to change it to. After this she will be able to arrange the girls into their specific sessions.

Links to Success Criteria:

- Staff can edit the above table
- Choice at the bottom of the page to sort girls.
- Arrange the girls into taster sessions

Usability Features:

- It would be useful to allow the table itself to be edited by Mrs Jackson, as it is a very tedious approach to have to state the ID, and then the subject, and then the number compared to selecting a cell and changing it.
 - If I figure out a way to do this I shall, but as it not necessary and not part of the briefing from my stake holder, it will be an adjustment made at the end if I have enough time
- The clear buttons and instructions on the screen guide Mrs Jackson through the process, making sure there is no confusion.

Algorithms:

TABLE DISPLAY:

```
procedure tableDisplay()
    connect to studentChoices
    response = sql command(SELECT * FROM studentChoices)
    OUTPUT(response)
end procedure
```

Explain:

This procedure displays all of the pupils and their respective responses to the survey.

EDIT TABLE:

```
procedure edtable()
```

```

connect to studentChoices

nid = input("Please enter the ID")

choice = input("Pick Choice") [dropdown html list for Choices
(Choice1, Choice2, Choice3, Choice4, Choice5, Choice6)]

subject = input("Please enter the new subject you want to
assign.")

//use dropsubject() procedure from Student Interface

//UPDATE BUTTON CLICKED

sql command(UPDATE studentChoices SET choice = subject WHERE id =
nid)

tableDisplay()

end procedure

```

Explain:

This procedure updates the table based upon the id, subject and rank Mrs Jackson has entered. It then displays the table to the screen, showing the update, so that Mrs Jackson can continue editing. The procedure also changes the mySQL database as a connection is established at the start of the procedure and the sql command was making changes to the database.

How the procedures work together:

The procedures for table display is self contained and links to the table on the server to display it to the screen. The edit table procedure on the other hand affects the display of the screen as once the inputs are submitted, they are updated dynamically on the webpage affecting the table the table display procedure actually displays.

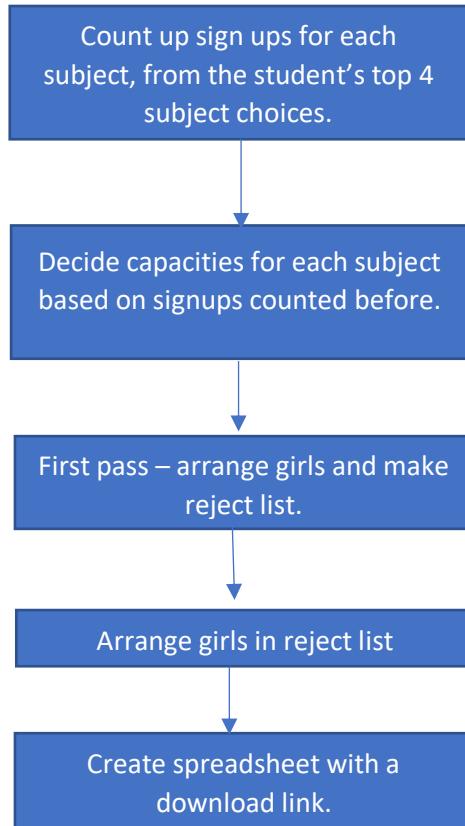
Arrange the girls:

This will be where the bulk of the programming of the project resides. To show the progression of ideas I intend to follow, I will be using a very high-level flow diagram and I will then expand on the algorithms behind each flow chart element using pseudocode. First a quick explanation:

First the program would need to calculate the number of votes for each subject, only dependent on the student's top 4 choices. Once we've found out the number of students who voted for a subject, we take 4 or fewer subjects in the least demand (under 10 sign ups) and fix them in a respective slot. These

subjects will only be offered for 1 session out of 4 unlike the other subjects. These subjects will be marked and all students who chose these subjects in their top 4 will immediately be allocated into these sessions. Then there will be 10 slots allocated in each session to all other subjects. If any of the slots need more classes, these will be decided depending upon how many more than 40 there were. If there was 60 – 70 signups to a certain class, 3 extra classes would be provided, but this will be represented as an increase in class capacity. Once the capacity for the 4 session's classes has been sorted out, we start to place the girls in these classes. Sort the girls that have chosen a fixed subject first, place them in their respective subjects, if in any instance they can not get their top 4 choices, place them in the reject list. Sort all the girls in this manner. From the reject list, I would look at the subject in their top 4, that they could not be put in and their available day. In the subject class she could not be put in, I would see if any of the girls in the class could be moved into that same subject class on a different day and switch to a different subject on the observed day (depending on class capacities), if this was not possible with the student compared, move on to next student. If the girl in the reject list is not able to be swapped, move onto the next girl in the reject list without removing girl from list. If the girl is able to be swapped, remove her from the reject list. Once we've run through the reject list once. Allocate the rest of the people in the reject list, either their 5th or 6th choice depending on class capacity. Make table with all the sorted data in the way Mrs Jackson would like.

Briefer explanation and relationship of procedures:



Would need an array for the four sessions!!!!

```
session1 = []
session2 = []
session3 = []
session4 = []
```

COUNT UP SIGNUPS FOR EACH SUBJECT:

```
Procedure signups()
    Connect to studentchoices
    subjectchoices = []

    //get studentid list from the way we got the list before, which was
    //from students.csv

    studentids = [list of student ids]
    for i = 0 to len(studentids):
        choices = array(sqlcommad("SELECT Choice1, Choice2, Choice3,
        Choice4 FROM studentchoices WHERE studentid ="studentids[i];"))
        subjectchoices.append(choices)
    signups = {}
```

```

for j = 0 to len(subjectchoices):
    for k = 0 to 4:
        if subjectchoices[j][k] is in signups:
            signups[subjectchoices[j][k]][0]+=1
        else:
            signups.append{subjectchoices[j][k]:[1]}

```

Explain:

This procedure creates a new dictionary, with the name of the subjects as the keys and a list with the number of signups as the value. Algorithm runs through list and if there is an unfamiliar subject, it is added to the dictionary.

CAPACITIES:

```

Function findindex(item, list)
    index ==-1
    for i =0 to len(list):
        if list[i][0] == item:
            index = i
            break
    return index

procedure capacities()
    fixedSub = []
    for key in singups():
        if signups[key[0]] < = 8:
            fixedSub.append(key)
    if len(fixedsub) > 4:
        while len(fixedsub) >4:
            fixedSub.remove(max(fixedSub))
    if len(fixedsub) =  1:
        session1.append([fixedsub[0],10,0])
        signups[fixedsub[0]].append(1)
    elif len(fixedsub) =  2:
        session1.append([fixedsub[0],10,0])

```

```

        session2.append([fixedsub[1],10,0])
        signups[fixedsub[0]].append(1)
        signups[fixedsub[1]].append(2)

    elif len(fixedsub) == 3:
        session1.append([fixedsub[0],10,0])
        session2.append([fixedsub[1],10,0])
        session3.append([fixedsub[2],10,0])
        signups[fixedsub[0]].append(1)
        signups[fixedsub[1]].append(2)
        signups[fixedsub[2]].append(3)

    elif len(fixedsub) == 4:
        session1.append([fixedsub[0],10,0])
        session2.append([fixedsub[1],10,0])
        session3.append([fixedsub[2],10,0])
        session4.append([fixedsub[3],10,0])
        signups[fixedsub[0]].append(1)
        signups[fixedsub[1]].append(2)
        signups[fixedsub[2]].append(3)
        signups[fixedsub[3]].append(4)

for each key,value in signups():

    if signups[key].isin(fixedsub) == False:
        session1.append([signups[key],10,0])
        session2.append([signups[key],10,0])
        session3.append([signups[key],10,0])
        session4.append([signups[key],10,0])

for each key,value in signups():

    if signups[value[0]]>40:
        c = signups[value[0]] - 40
        //rounds c up to integer above always
        c = roundup(c/10)

        //upon speculation 68 is the largest number of people who
        have picked a choice, so we will scale up to 80

        if c == 1:

            index = findindex(signups[key], session1)

```

```

        session1[index][1]+=10

    if c==2:

        index = findindex(signups[key], session1)
        session1[index][1]+=10

        index = findindex(signups[key], session2)
        session2[index][1]+=10

    if c==3:

        index = findindex(signups[key], session1)
        session1[index][1]+=10

        index = findindex(signups[key], session2)
        session2[index][1]+=10

        index = findindex(signups[key], session3)
        session3[index][1]+=10

    if c==4:

        index = findindex(signups[key], session1)
        session1[index][1]+=10

        index = findindex(signups[key], session2)
        session2[index][1]+=10

        index = findindex(signups[key], session3)
        session3[index][1]+=10

        index = findindex(signups[key], session4)
        session4[index][1]+=10

end procedure

```

Explain:

This procedure allocates the number of spots that should be available for each subject depending on the number of people who have signed up for the subjects. These are stores in the session lists.

First Pass:

```

for w = 0 to len(session1):
    session1[w].append([])

for x = 0 to len(session2):

```

```

    session2[x].append([])

for y = 0 to len(session3):
    session3[y].append([])

for z = 0 to len(session4):
    session4[z].append([])

rejectlist = []

Procedure firstpass:

    studentplaces = {}

    For i=0 to len(studentids):
        Studentplaces.append({studentids[i]: [[], [], [], []]})

    //girls who have picked a subject in the reject list

    for i=0 to len(studentids):

        choices = array(sqlcommad("SELECT Choice1, Choice2, Choice3,
Choice4 FROM studentchoices WHERE studentid =" + studentids[i] + ";"))

        for j = 0 to len(choices):
            for k = 0 to len(fixedsub):
                if choices[j] == fixedsub[k]:
                    //I was basically trying to add the person
                    to the correct session list and also show the correct session and
                    subject together in the studentplaces dictionary.

                    //do this 4 times for each section changing
                    number depending on the session

                    if findindex(choices[j], session1) != -1:
                        index = findindex(choices[j], session1)
                        session1[index][3].append(studentids[i])
                        session1[index][2] += 1
                        for x = 0 to 4://formatting off
                            if studentplaces[studentids[i]][x].
                                isempty():
                                studentplaces[studentids[i]][x].append(choices[j],
                                1)
                            break

```

Explain:

This procedure places the girls who have chosen a subject in the fixed subject list into their respective sessions and also adds the session and subject they've been allocated to the student list.

Actual Pass:

```
procedure actualpass()
    for i=0 to len(studentids):
        choices = array(sqlcommad("SELECT Choice1, Choice2, Choice3, Choice4
FROM studentchoices WHERE studentid ="studentids[i];")
        for j = 0 to len(choices):
            //do this 4 times for each section changing number depending on
            the session
            if findindex(choices[j],session1) != -1:
                index =findindex(choices[j],session1)
                if session1[index][2] < session1[index][1]:
                    session1[index][3].append(studentids[i])
                    session1[index][2] +=1
                for x = 0 to 4://formatting off
                    if studentplaces[studentids[i]][x].isempty()
                        studentplaces[studentids[i]][x].append(choices[j],
                        1)
                    break
                //this checks the girls that still have not filled up all their spots
                and puts them in the reject list
                if (studentplaces[studentids[i]][0] or studentplaces[studentids[i]][1]
                or studentplaces[studentids[i]][2] or
                studentplaces[studentids[i]][3]).isempty():
                    rejectlist.append(studentids[i])
```

Explain:

This procedure tries to place all the girls into a session from one of the subjects they have chosen, if the program cannot fill the girls list then they are placed in the reject list, which essentially will assign the girls their 5th and 6th if possible. If this is also not possible, lable these girls as problems

Reject Pass:

```

procedure rejectpass()
    for i=0 to len(studentids):
        choices = array(sqlcommad("SELECT Choice5, Choice6 FROM studentchoices
WHERE studentid ="studentids[i]""))

        for j = 0 to len(choices):
            //do this 4 times for each section changing number depending on
the session

            if findindex(choices[j],session1) != -1:
                index =findindex(choices[j],session1)

                if session1[index][1] >= session1[index][1]:
                    session1[index][3].append(studentids[i])
                    session1[index][2] +=1

                //this basically puts the girls allocation inside
her list and makes sure that they are not overwriting a current option by checking
to make sure they are closed.

                for x = 0 to 4://formatting off
                    if studentplaces[studentids[i]][x].
isempty():

                        studentplaces[studentids[i]][x].append(choices[j],
1)

                    break

```

Explain:

This procedure is almost identical to the actualpass() procedure, only this procedure only considers the 5th and 6th choices as options. This function is really unlikely to get used as there are enough slots to cater for everybody's choices. This is a fail-safe algorithm, in the unlikely case that spots on a particular subject are not able to cater to students' needs.

Creating Files:

Creating a csv file for session1, session2, session3 and session4 to display on next interface and also create a csv file for studentplaces: Ideal format of csv file is below:

Name	Session 1	Session 2	Session 3	Session 4
Ola Okoroafor	Computing	Maths	Chinese	Physics

Session csv files:

```

procedure writefiles():
//writes session1 to csv file
file = wopen("session1.csv")

```

```

file.write(session1)
file.close()

//writes session2 to csv file
file = wopen("session2.csv")
file.write(session2)
file.close()

//writes session3 to csv file
file = wopen("session3.csv")
file.write(session3)
file.close()

//writes session1 to csv file
file = wopen("session4.csv")
file.write(session4)
file.close()

c = 0
final = []
for key, value in studentplaces:
    final.append([studentplaces[key]])
    final[c].append(' ', ' ', ' ', '')//4 empty places
    for i = 0 to 4:
        final[c][i+1] = studentplaces[value][i][0]
    c = c+1

//this sorts the 2d lists stored in the student choice
file = wopen("final.csv")
file.write(final)
file.close()

```

Explain:

This procedure, essentially writes all the information that we have collected into csv files, all the sessions are collated into csv files. The final files is also collated and will be displayed in a table on the 4th Staff Interface to edit.

How the procedures work together:

This is demonstrated with the flowchart above.

Test Plan: *for interface*

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output
Valid Student ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	Page refreshes, and values have been changed on screen and in php database
Invalid (non existent) Student ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated.
Valid Student ID, Invalid (non existent) Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated. Also drop down list makes this impossible.
Valid Student ID, Valid Choice, Invalid (non existent) Subject	Staff can edit the above table	EDIT TABLE	Drop down list of subjects input only allows correct subject submission.
Visible button	Choice at the bottom of the page to sort girls	ARRANGE GIRLS	Screenshot visible and functioning button
After validation in prior areas of code, there will be no erroneous data in the program. Therefore no need to test under those circumstances. Will do white box testing here along the way to make sure the programs subroutines work. Then test with the data from last year to simulate a real issue.	Arrange the girls into correct sessions	ARRANGE GIRLS	Sorts girls to Mrs Jackson's liking. – all girls should have 4 options, no girl should have any option twice, the success of this aspect is up to Mrs Jackson

*All other aspects of this page have been tested in the second staff interface, as they have the exact same functionality

ARRANGED GIRLS

Name	Session 1	Session 2	Session 3	Session 4
Ola Okoroafor	Computing	Maths	Chinese	Physics
Sample	Sample	Sample	Sample	Sample
Sample	Sample	Sample	Sample	Sample

DOWNLOAD

On this page, all the girls will have been arranged and sorted into a csv file that corresponds to the table displayed on this page. This is the final webpage Mrs Jackson will interact with. She will be able to download the table as a csv file to her laptop, to manipulate the data as she pleases.

Links to Success Criteria:

- Displays table of sessions to the user
- Download button on Staff interface once table of sessions is ready
- Downloads all final files onto client's pc

Usability Features:

- The clear buttons: download button is clear, easy for Mrs Jackson to see and use.

Algorithms:

DISPLAY TABLE:

```
Procedure displaycsv()
    file = open_r("final.csv")
    for each line in file()
        table_row = line
    end for
```

Explain:

This displays the final csv file in the table as presented in the design.

DOWLOAD TABLE:

```
Procedure download()
    to_download =
open("final.csv", "session1.csv", "session2.csv", "session3.csv", "session4.csv")
    download(to_download)
```

Explain:

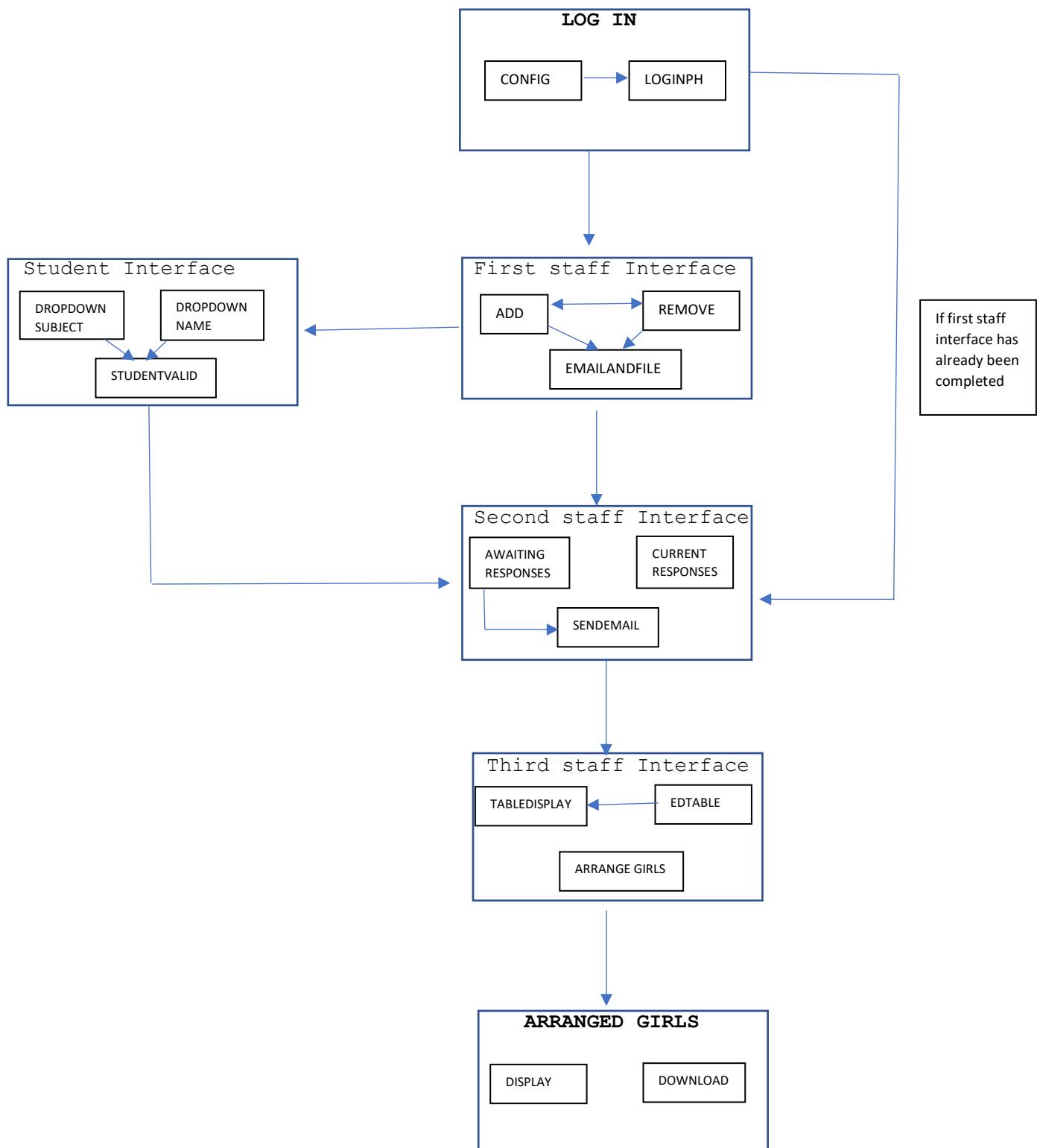
This program downloads the final csv file to Mrs Jackson's computer.

Test Plan:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output
Data in the final.csv file.	Display table of sessions to the user	DISPLAY TABLE	Values in csv file, must be the same as those presented in the table
Invalid (non existent) Staff ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated.
Valid Staff ID, Invalid (non existent) Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated. Also drop down list makes this impossible.
Is Download button there?	Download button on Staff interface once table of sessions is ready	DOWNLOAD TABLE	Download link should be clear and visible
Data in the final csv file	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Final csv file should be the same as the file downloaded by user
Data in the session1 csv file	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Session 1 csv file should be the same as the file downloaded by user
Data in the session2 csv	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Session 2 csv file should be the same as the file downloaded by user
Data in the session3 csv file	Download button on Staff interface once	DOWNLOAD TABLE	Session 3 csv file should be the same as

	table of sessions is ready.		the file downloaded by user
Data in the session4 csv file	Download button on Staff interface once table of sessions is ready	DOWNLOAD TABLE	Session 4 csv file should be the same as the file downloaded by user

STRUCTURE OF THE SOLUTION:



Stake holder Input

I sent my stakeholder a summary sheet for my design, and she was completely fine with the design of the webpages and the functionality behind each.

The screenshot shows a Microsoft Teams message thread. The top message is from Okoroafor, Olaedo, dated Mon 11/11/2019 12:22, containing a file named 'Interfaces.docx' (68 KB). The response is from Mrs AE Jackson (Saturday Administrator), dated Mon 11/11/2019 12:30. The message reads:

Dear Mrs Jackson,
I have completed the design for the web interfaces, and I am currently in the midst of designing the algorithms that go along with them.
If you have any suggestions or complaints, please do let me know! The arrange girls page is still under review.
Best wishes,
Ola

Below this, there is a redacted section followed by another message from Mrs AE Jackson:

Mrs AE Jackson (Saturday Administrator) Mon 11/11/2019 12:30
Okoroafor, Olaedo
These look very clear, I'll look forward to seeing a test version as you develop it

INPUTS, PROCESSES, OUTPUTS AND STORAGE

Inputs	Processes	Outputs	Storage
Username and Password text boxes, with login button	Loginphp algorithm	Will display a message depending on whether or not the authentication was successful.	Username and password that input should be compared to are stored in table called users on php myadmin database.
Upload button	Uploads the student table csv file to php server	No output	Adds the csv file to the php server
Subjects text box, and add button	Add subject algorithm	Will display subject just added.	Added subject is stored in javascript list on the webpage.
Subjects textbox and remove button	Remove subject algorithm	Will remove the subject displayed on screen.	Removed subject will be removed from javascript list on the webpage

Send questionnaire to girls button	Email and file algorithm	Direct user to second staff interface.	Stores javascript subject list in file on the xampp server.
Send Email Reminder button	Send email algorithm	No output on page	Uses javascript awaiting responses list created by another algorithm to send the emails
Next page Button	Connects the user to the third staff interface	The third staff interface display	No storage
ID and Choice and Subject text boxes, with the update table button	Editable algorithm	If no validation issue, then updates table on screen.	Updates the studentchoices table on the phpMyAdmin xampp server.
Arrange girls into sessions	Arrange girls subroutines	Takes the user to the arrange girls interface.	In these subroutines, files and the studentchoices table will be accessed from the xampp server.
Download button	Download algorithm	No output	Replicates a copy of the final csv file from xampp server, and offers it for download.
Drop down name options, drop down subject options and submit button.	Student valid algorithm	Direct user to thank you page!	Input from this page is stored in the studentchoices table on the xampp server for mysql

KEY VARIABLES AND STRUCTURES

Name	Data Type	What is stored	Why is it needed
username	Variable	Input text	To be used in the login algorithm to authenticate Mrs Jackson
password	Variable	Input text	To be used in the login algorithm to

			authenticate Mrs Jackson
loggedin	Variable	Boolean	Only give Mrs Jackson access to her pages if sheloggedin = True
studentTable	2d array	All the student information – information corresponding to each student(ID and username)	Used to find emails and Ids for uses throughout the project.
subjectli	1d array	List of subjects entered by Mrs Jackson	Used to display a list of subjects to the students to pick from.
completedli	1d array	List of students who have completed the survey.	Used to figure out who has not completed survey, for display
nresponse	1d array	List of people who have not completed the survey	Used for email reminders
subjectchoices	1d array	This lists all the choices that were made as the students' 1 st to 4 th choices. Concatenates all of these into one list.	Used to help count up all the sign ups for each subject
signups	Associative array (with a 1d array as value)	This array stores the subject as the key, then a 1d list as the value. The 1d array includes the number of signups, and thus how many sessions the subject has been allocated.	Used to deduce how many sessions each subject needs to be added to, depending on their signups.
fixedsub	1d array	Stores 4 or less of the subjects with the least signups	Used to deduce which subjects are in fixed places, and so are included in

			the first pass algorithm
sessionx: - session1 - session2 - session3 - session4	3d array	3d list that stores thing in this format: Sessionx = [[subject, capacity, occupied, [student1, student2,...],...]	These are used to deduce which sessions are full or empty when the program is allocating students subjects
studentplaces	Associative array (with a 1d array as value)	Stores the id of the student as the key, and as the value it stores a list of their allocated subjects.	Used as a template for the final csv file which will then be downloaded by Mrs Jackson
rejectli	1d array	List of student ids that haven't been stored.	Used to allow the girls that have not been allocated all their subjects to try again with the reject pass algorithm
to_download	2d array	A 2d array that stores the girls and their respective subjects.	Used to allow Mrs Jackson to download a copy of what is on the php server.

Validation Table:

To make the program as robust as possible, all the inputs to the program will need to be validated if possible. So that the program acts in the defined manner in all cases, for this we would need to *consider* all cases. Most of the validation in this project will occur on the student interface, as there is no real way to validate Mrs Jackson's files. (Discussed further below) All html items such as drop down lists, buttons, and textboxes are already robust and I thus do not need to consider these in the validation process.

Inputs	How they may be validated?
Username and Password	Both will be checked. The username will be checked to make sure it is in the system. The password will be checked to make sure that it matches prescribed password. Further all the string will be escaped when

	querying the database to avoid SQL injection
The dropdown choice lists for name and subjects.	This essentially pre-validated as the lists are predefined for them, meaning all the input they may enter is expected.
The edit the table procedure	This asks for a student id, a choice and a subject. As the choice and the subject are from a drop down list, they are pre validated and all input will be expected. However if Mrs Jackson enters an invalid student id, their query will not alter the database, making it sufficiently validated.
The students csv file	This csv file will not have to be validated, as it is created by the data manager for the website, and as we don't know what form the studentids may be or all the student usernames, trying to validate would be impossible.

IMPORTANT TO ADD THAT MY ALPHA TESTING TABLES WILL BE THE SAME AS MY BETA TESTING TABLES

DEVELOPMENT STAGES:

Stage 1:

- Involves No CSS
- Involves No extra validation, than what is stated in design
- Involves none of the helping hand textboxes
- Functional Log In Interface
- Functional First Staff Interface, without the newly uploaded link.
- Functional Student Interface
- Functional Second Staff Interface
- Functional Third Staff Interface, excluding the aspect of arranging the girls into sessions.
- Meeting with stakeholders: Mr Wain, Mrs AE Jackson. Will need to request actual data from the UC5 girls from Mr Wain. Ask for feedback from Mrs Jackson
- Note changes and suggestions.

Stage 2:

- Implement changes and suggestions from previous stage.

- Functional First Staff Interface, with the newly uploaded link.
- Functional Third Staff Interface, including the aspect of arranging girls into sessions.
- Meeting with stakeholder, Mrs Jackson. Ask for feedback and changes/suggestions on completed sections.

Stage 3:

- Implement changes and suggestions from previous stage.
- Add all the validation absent from the stages before.
- Meeting with stakeholder, Mrs Jackson. Ask for feedback and changes/suggestions on completed sections.

Stage 4:

- Implement changes and suggestions from previous stage.
- Add all the css to make the webpage look presentable, and also include helping hand textboxes.
- Include the last interface where Mrs Jackson can download files with CSS.
- Ask for final stakeholder feedback.
- Implement changes and suggestions from above meeting.

DEVELOPMENT

DEVELOPMENT

STAGE 1

- Involves No CSS
- Involves No extra validation, than what is stated in design
- Involves none of the helping hand textboxes
- Functional Log In Interface
- Functional First Staff Interface, without the newly uploaded link.
- Functional Student Interface
- Functional Second Staff Interface
- Functional Third Staff Interface, excluding the aspect of arranging the girls into sessions.
- Meeting with stakeholders: Mr Wain, Mrs AE Jackson. Will need to request actual data from the UC5 girls from Mr Wain. Ask for feedback from Mrs Jackson.
- Note changes and suggestions.

Login Interface

The program for the log in interface will require 4 files, all with PHP script and html. For the start I will not be using a css style sheet until the future.

For the configuration file, this is the code which allows me to connect to mySQL database and allows me to use the php commands to query the database and store its contents in variables on the html page. The name of this file is config.php

```
1 <?php
2 /* Database credentials. Assuming you are running MySQL
3 -server with default setting (user 'root' with no password) */
4 define('DB_SERVER', 'localhost');
5 define('DB_USERNAME', 'root');
6 define('DB_PASSWORD', '');
7 define('DB_NAME', 'mysql');
8
9 /* Attempt to connect to MySQL database */
10 $link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
11
12 // Check connection
13 if($link === false){
14     die("ERROR: Could not connect. " . mysqli_connect_error());
15 }
16 ?>
```

The following file is the actual log in file which I have named login.php. This file does exactly what my login algorithm specified except, I did not know that the function “password_verify()” requires me to use a hashed password. Which essentially meant that the passwords in the database were supposed to be hashed. To overcome this, I hashed the password on the html page rather than in the database. Because I am not using hashed passwords within my database as it is unnecessary and also inconvenient, I have decided not to alter the database to include this. This is because, there will only be one user and no sign up page hash the password initially. The database only has one user and I want to be able to test different logins without hashing them, which makes it easier for me to test. The lines that I am referring to are highlighted in blue below.

```
18 // Processing form data when form is submitted
19 if($_SERVER["REQUEST_METHOD"] == "POST"){
20
21     // Check if username is empty
22     if(empty(trim($_POST["username"]))){
23         $username_err = "Please enter username.";
24     } else{
25         $username = trim($_POST["username"]);
26     }
27
28     // Check if password is empty
29     if(empty(trim($_POST["password"]))){
30         $password_err = "Please enter your password.";
31     } else{
32         $password = trim($_POST["password"]);
33     }
34
35     // Validate credentials
36     if(empty($username_err) && empty($password_err)){
37         // Prepare a select statement
38         $sql = "SELECT id, username, password FROM users WHERE username = ?";
39
40         if($stmt = mysqli_prepare($link, $sql)){
41             // Bind variables to the prepared statement as parameters
42             mysqli_stmt_bind_param($stmt, "s", $param_username);
43
44             // Set parameters
45             $param_username = $username;
46
47             // Attempt to execute the prepared statement
48             if(mysqli_stmt_execute($stmt)){
49                 // Store result
50                 mysqli_stmt_store_result($stmt);
51
52                 // Check if username exists, if yes then verify password
53                 if(mysqli_stmt_num_rows($stmt) == 1){
54                     // Bind result variables
55                     mysqli_stmt_bind_result($stmt, $id, $username, $password);
56                     $hashed_password = password_hash($password, PASSWORD_DEFAULT);
57                     if(mysqli_stmt_fetch($stmt)){
58                         if(password_verify($password, $hashed_password)){
59                             // Password is correct, so start a new session
60                             session_start();
61
62                             // Store data in session variables
63                             $_SESSION["loggedin"] = true;
64                             $_SESSION["id"] = $id;
65                             $_SESSION["username"] = $username;
66
67                             // Redirect user to welcome page
68                             header("location: welcome.php");
69                         } else{
70                             // Display an error message if password is not valid
71                             $password_err = "The password you entered was not valid.";
72                         }
73                     }
74                 } else{
75                     // Display an error message if username doesn't exist
76                     $username_err = "No account found with that username.";
77                 }
78             } else{
79                 echo "Oops! Something went wrong. Please try again later.";
80             }
81         }
82     }
}
```

```

83     // Close statement
84     mysqli_stmt_close($stmt);
85   }
86
87   // Close connection
88   mysqli_close($link);
89 }
?>
91
92 <!DOCTYPE html>
93 <html lang="en">
94 <head>
95   <meta charset="UTF-8">
96   <title>Login</title>
97   <style type="text/css">
98     body{ font: 14px sans-serif; }
99     .wrapper{ width: 350px; padding: 20px; }
100    </style>
101   </head>
102 <body>
103   <div class="wrapper">
104     <h2>Login</h2>
105     <p>Please fill in your credentials to login.</p>
106     <form action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
107       <div class="input-group "<?php echo (!empty($username_err)) ? 'has-error' : ''; ?>">
108         <label>Username</label>
109         <input type="text" name="username" class="form-control" value=<?php echo $username; ?>>
110         <span class="error success"><?php echo $username_err; ?></span>
111       </div>
112       <div class="input-group "<?php echo (!empty($password_err)) ? 'has-error' : ''; ?>">
113         <label>Password</label>
114         <input type="password" name="password" class="form-control">
115         <span class="error success"><?php echo $password_err; ?></span>
116       </div>
117       <div class="input-group">
118         <input type="submit" class="btn btn-primary" value="Login">
119       </div>
120     </form>
121   </div>
122 </body>
123 </html>

```

For the login page, I used code from this location (<https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>) This page was very helpful and provided me with an almost perfect series of files for me to use to develop my login system. The program for the login system initially was attached to a table developed by their sign up page, but because I am not making a sign up page, I connected these ends so that the login was connected to my initial users database. The initial program also uses hashed passwords in the database table which I have decided not to use. I also did not specify a logout page nor a logout button because I did not think it was necessary but it is as otherwise I would stay logged in and not be able to log out and thus not be able to test different inputs for the login system.

The next file is just a filler file for the next interface Mrs AE Jackson should be connected to as I have not started it yet, for now it only has the log out button.

```

1 <?php
2 // Initialize the session
3 session_start();
4
5 // Check if the user is logged in, if not then redirect him to login page
6 if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
7   header("location: login.php");
8   exit;
9 }
?>
10
11 <!DOCTYPE html>
12 <html lang="en">
13 <head>
14   <meta charset="UTF-8">
15   <title>Welcome</title>
16   <style type="text/css">
17     body{ font: 14px sans-serif; text-align: center; }
18   </style>
19 </head>
20 <body>
21   <div class="page-header">
22     <h1>Hi, <b><?php echo htmlspecialchars($_SESSION["username"]); ?></b>. Welcome to our site.</h1>
23   </div>
24   <p>
25     <a href="logout.php" class="btn btn-danger">Sign Out of Your Account</a>
26   </p>
27 </body>
28 </html>

```

The next file is the logout file which essentially quits the session and setsloggedin = False, allowing me to login again for testing.

```

1  <?php
2  // Initialize the session
3  session_start();
4
5  // Unset all of the session variables
6  $_SESSION = array();
7
8  // Destroy the session.
9  session_destroy();
10
11 // Redirect to login page
12 header("location: login.php");
13 exit;
14 ?>
```

Testing:

Test Data	Relation to success criteria	Expected Output	PASS OR FAIL	Testing Evidence
Username:JacksonAE Password:	Login for Staff Interface	Need to input Password	P	1
Username: Password:Project123	Login for Staff Interface	Need to input username	P	2
Username:jacksonae Password:Project123	Login for Staff Interface	Log in	P	3
Username:JacksonAE Password:project123	Login for Staff Interface	Incorrect password	P	4
Username: Password:	Login for Staff Interface	Need to input username Need to input password	P	5
Username:me Password:me	Login for Staff Interface	Username not in database	P	6
Username:Project123 Password:Project123	Login for Staff Interface	Username not in database	P	7
Username:JacksonAE Password:JacksonAE	Login for Staff Interface	Incorrect Password	P	8
LOG IN BUTTON CLICK with correct credentials	Login for Staff Interface	Go to welcome page	P	9
LOG OUT BUTTON CLICK	Login for Staff Interface	Go back to login page	P	10

Testing evidence:

*when password is incorrect, it disappears and is not in screenshot

1	Login Please fill in your credentials to login. Username <input type="text" value="JacksonAE"/> Password <input type="password"/> Please enter your password. <input type="button" value="Login"/>
2	Login Please fill in your credentials to login. Username <input type="text"/> Please enter username. Password <input type="password"/> <input type="button" value="Login"/>
3	Hi, JacksonAE. Welcome to our site. Sign Out of Your Account
4	Login Please fill in your credentials to login. Username <input type="text" value="JacksonAE"/> Password <input type="password"/> The password you entered was not valid. <input type="button" value="Login"/>
5	Login Please fill in your credentials to login. Username <input type="text"/> Please enter username. Password <input type="password"/> Please enter your password. <input type="button" value="Login"/>
6	Login Please fill in your credentials to login. Username <input type="text" value="me"/> No account found with that username. Password <input type="password"/> <input type="button" value="Login"/>
7	Login Please fill in your credentials to login. Username <input type="text" value="Project123"/> No account found with that username. Password <input type="password"/> <input type="button" value="Login"/>
8	Login Please fill in your credentials to login. Username <input type="text" value="JacksonAE"/> Password <input type="password"/> The password you entered was not valid. <input type="button" value="Login"/>
9	Hi, JacksonAE. Welcome to our site. Sign Out of Your Account
10	Login Please fill in your credentials to login. Username <input type="text"/> Password <input type="password"/> <input type="button" value="Login"/>

Stake holder response:

- Positive
 - o Just need some css, but this is detailed later.

First Staff Interface

The program for the first interface will require 3 files, all with PHP script and html, but one of these pages will be linked to the next interface. For the start I will not be using a css style sheet until most of the functionality is in place.

As stated in my Ammended Student Interface, I have decided to implement the studentChoices table differently. Therefore, I had to also adapt the initial algorithms designed for this interface to conform with the new method of implementation:

- Instead of making each of the subjects headers of a table in the mySQL database, I have decided to put the subjects into a csv file to be referenced later as I have changed the way I wish my table to be formatted.
 - o I have decided to format my tables with the ranking of choice as the column names (eg. Choice 1, Choice 2....)
 - o I did this as it would be significantly easier to code and for me to place the data into the tables as, the column names would be definite and not dependent on the subjects Mrs Jackson enters. Also helps solve some algorithmic issues down the line.
 - o This means for this interface, there will be no contact with the studentChoices table. I have also decided to order the table by the Ids specified by college for the future.
- All of the above, will require me to redesign some of the functions I previously planned.

CODE:

The code below is the html form, the values of which will be posted to the next interface.

```
1 <html>
2   <h1> Fill in Subjects </h1>
3   <div id='list'>
4   </div>
5   <p> When Submit button is clicked, you can not come back to this page!</p>
6   <p id = "test" ></p>
7   <form method="POST" action='interface2.php'>
8     <input type='text' id='input' />
9     <input type='button' value='add subject' id = "add" />
10    <input type='text' id='input2' />
11    <input type='button' value='remove subject' id='remove' />
12    <input type = 'hidden' name = 'myList' value = subjectcsv id = 'myList' />
13    <input type='submit' value='Submit' id = "subjectsub" name = "complete" />
14  </form>
```

The main thing being posted from this form is the subjectcsv. This is a variable that stores all the subjects Mrs Jackson will submit in a longstring in csv format. This then makes it easier to post the data to php, explode it into a php array and then write the array to a csv file.

The code below is the javascript I used to remove list elements from the list on the html document and from the actual subjectli and subjectcsv. I found it difficult to change what was already on the string, so I decided to just clear all on the screen and repopulate the list elements on the html document. It was much simpler changing the actual lists as I simply had to pop it.

```
15 <script type="text/javascript">
16 //initialising variables
17 var subjectli = [];
18 var subjectcsv = "";
19 //function called remove which removes subject
20 document.getElementById("remove").onclick = function(){
21     //finds index of the item to remove and removes it from list
22     document.getElementById("list").innerHTML = "";
23     var toRemove = document.getElementById("input2").value;
24     var index = subjectli.indexOf(toRemove);
25     if (index > -1) {
26         subjectli.splice(index, 1);
27     }
28     //resets the list displayed the altered list with removed element
29     var sList = "";
30
31     for (var I = 0; I < subjectli.length; I++){
32         sList = "<li>" + subjectli[I] + "</li>";
33         document.getElementById("list").innerHTML += sList;
34     }
35     document.getElementById("input2").value = ""; // clear the value
36     var subjectcsv = "";
37     for (var I = 0; I < subjectli.length; I++){
38
39         subjectcsv = subjectcsv + subjectli[I] + ",";
40     }
41     //after all the new elements in the str have been appended separated by commas
42     //removes the last comma and
43     subjectcsv = subjectcsv.slice(0, -1);
44     //document.getElementById("myList").value = subjectcsv;
45 }
```

The code below is the javascript I used to add elements to the html page and also populate the subjectli which was then turned into the subjectcsv string through concatenation of elements of subjectli and commas in a for loop.

```

46 //adds all the elements in the way it was done above
47 document.getElementById("add").onclick = function() {
48
49     var text = document.getElementById("input").value;
50     //following lines make sure empty elements are not added to the list!
51     if (text === "") {
52     }
53     else{
54         subjectli.push(text);
55         var li = document.createElement("li");
56         li.textContent = text;
57         document.getElementById("list").appendChild(li);
58     }
59
60     document.getElementById("input").value = ""; // clear the value
61     var subjectcsv = "";
62     for (var I = 0; I < subjectli.length; I++)
63     {
64         ....
65         subjectcsv = subjectcsv + subjectli[I] + ",";
66     }
67     subjectcsv = subjectcsv.slice(0, -1);
68     //document.getElementById("myList").value = subjectcsv;
69
70 }

```

All the code above was contained within the subjects.php file. All the values of this form were posted over to interface2.php. I created a temporary csv file for this stage of development to test for the delivery of emails, as one has not been created yet.

This is the temporary file I created which is named students.csv

```

1 What, What, What, What, What, What, What,okoroaforo@cheltladiescollege.org
2 What, What, What, What, What, What, What,kuunm@cheltladiescollege.org
3 What, What, What, What, What, What, What,clowesi@cheltladiescollege.org
4 What, What, What, What, What, What, What,junadyj@cheltladiescollege.org
5 What, What, What, What, What, What, What,mannai@cheltladiescollege.org
6 What, What, What, What, What, What, What,rolinsonl@cheltladiescollege.org
7 What, What, What, What, What, What, What,webbad@cheltladiescollege.org
8 What, What, What, What, What, What, What,faytera@cheltladiescollege.org

```

The file below is interface2.php which will eventually show Mrs Jackson the second interface design but for now manipulates the values gotten from the last page. As shown by the way in which I commented my code, the php in this file will save subjectcsv as a csv file named subject.csv, then it will email all the students a link to the survey as the subjects available are now known.

```

2  <?php
3  // Initialize the session
4  session_start();
5
6  // Check if the user is logged in, if not then redirect him to login page
7  if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] != true) {
8      header("location: login.php");
9      exit;
10 }
11 //this is where what is posted from the last page is handled
12 //The only thing posted was the string of comma delimited subjects created
13 if (isset($_POST['complete'])) {
14     //this variable stores an array, as exlodes the csv file past over
15     $subjectA = explode(",",$_POST['myList']);
16     //save the list for future use.
17     $fp = fopen('subject.csv', 'w');
18     fputcsv($fp, $subjectA);
19
20     fclose($fp);
21     //puts the csv file provided into a 2d array
22     $studentTable = array();
23     if (($handle = fopen("students.csv", "r")) !== FALSE) {
24         while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
25             $studentTable[] = $data;
26         }
27     }
28 }
29 //creates list that will store list of girls emails
30 $toEmail = array();
31 //runs through studentTable and appends emails to toEmail.
32 for ($I = 0; $I < sizeof($studentTable); $I++) {
33     array_push($toEmail, $studentTable[$I][9]);
34 }
35
36 $email_to = implode(',', $toEmail); // your email address
37 $email_subject = "Fill out this survey"; // email subject line
38 $from = 'JacksonAE@cheltladiescollege.org';
39
40 // To send HTML mail, the Content-type header must be set
41 $headers = 'MIME-Version: 1.0' . "\r\n";
42 $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
43
44 // Create email headers
45 $headers .= 'From: ' . $from . "\r\n".
46     'Reply-To: ' . $from . "\r\n".
47     'X-Mailer: PHP/' . phpversion();
48
49 // Compose a simple HTML email message
50 $message = '<html><body>';
51 $message .= '<h3> SURVEY! </h3>';
52 $message .= '<p style="color:#080;font-size:18px;">Please fill out';
53 $message .= '</body></html>';
54 mail($email_to, $email_subject, $message, $headers);
55

```

Below is the html for this webpage, which doesn't do anything but welcome the user and provide a log out button just yet, I only need this page to test my final log-in system.

```

59 <html lang="en">
60 <head>
61     <meta charset="UTF-8">
62     <title>Welcome</title>
63     <style type="text/css">
64         body{ font: 14px sans-serif; text-align: center; }
65     </style>
66 </head>
67 <body>
68     <div class="page-header">
69         <h1>Hi, <b>Mrs Jackson</b>. Welcome to our site.</h1>
70     </div>
71     <p>
72         <a href="logout.php" class="btn btn-danger">Sign Out </a>
73     </p>
74 </body>
75 </html>

```

Testing:

For the add and remove functions, Mrs Jackson should be able to enter anything but spaces.

Test Data	Relation to success criteria	Function Name	Expected Output
Any string	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should add string to the list on the screen and table.
Empty space	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should not add anything to the list or table.
A string in the list	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Should remove element in list and table.
A string or empty space not in list.	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Does not remove anything from list or table.
Click the button	Staff interface allows for staff to enter a list of subjects.	EmailandTable	Create studentChoices table in mySQL database
Click the button	Email students	EmailandTable	List should be saved in table and list and emails sent to students.

The is the original test table, but due to some changes, the test table will now be.

Test Data	Relation to success criteria	Function Name	Expected Output	PASS/ FAIL	Testing evidence
Any string	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should add string to the list on the screen and csv file .	P	1&2
Empty space	Staff interface allows for staff to enter a list of subjects.	Add Subject	Should not add anything to the list or csv file	P	3
A string in the list	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Should remove element in list and csvfile .	P	4&5
A string or empty space not in list.	Staff interface allows for staff to enter a list of subjects.	Remove Subject	Does not remove anything from list or csv file .	P	6&7
Click the button	Email students	EmailandTable	List should be saved in csv file and list and emails sent to students.	P	8,9&10
Upload link – (additional component)	n/a	n/a	File uploaded should be the same as the students csv file in htdocs.	F – next stage of development	

Testing evidence:

1	Fill in Subjects				
When Submit button is clicked, you can not come back to this page!					
	<input type="text" value="Hi"/>	<input type="button" value="add subject"/>	<input type="text"/>	<input type="button" value="remove subject"/>	<input type="button" value="Submit"/>

2	<h2>Fill in Subjects</h2> <ul style="list-style-type: none"> Hi <p>When Submit button is clicked, you can not come back to this page!</p> <input type="text"/> <input type="button" value="add subject"/> <input type="text"/> <input type="button" value="remove subject"/> <input type="button" value="Submit"/>
3	I clicked the add subject button with no string and nothing happened! Can't really show this though :/
4	<h2>Fill in Subjects</h2> <ul style="list-style-type: none"> Hi <p>When Submit button is clicked, you can not come back to this page!</p> <input type="text"/> <input type="button" value="add subject"/> <input type="text" value="Hi"/> <input type="button" value="remove subject"/> <input type="button" value="Submit"/>
5	<h2>Fill in Subjects</h2> <p>When Submit button is clicked, you can not come back to this page!</p> <input type="text"/> <input type="button" value="add subject"/> <input type="text"/> <input type="button" value="remove subject"/> <input type="button" value="Submit"/>
6	<p>Could not show test for empty space but this is the removal of a random element in sample list.</p> <h2>Fill in Subjects</h2> <ul style="list-style-type: none"> Hello World ! <p>When Submit button is clicked, you can not come back to this page!</p> <input type="text"/> <input type="button" value="add subject"/> <input type="text" value="Hi"/> <input type="button" value="remove subject"/> <input type="button" value="Submit"/>
7	<h2>Fill in Subjects</h2> <ul style="list-style-type: none"> Hello World ! <p>When Submit button is clicked, you can not come back to this page!</p> <input type="text"/> <input type="button" value="add subject"/> <input type="text"/> <input type="button" value="remove subject"/> <input type="button" value="Submit"/>
8	<p>Submit button is clicked and the following evidences show:</p> <p>Hi, Mrs Jackson. Welcome to our site.</p> <p>Sign Out</p>
9	<p>This is the subject file:</p> <p>subject.csv - Notepad File Edit Format View Help Hello,World,!</p>
10	Email to my test subjects:

Contact Form Message

JacksonAE@cheltladiescollege.org
Fri 13/12/2019 11:42
Okoroafor, Olaedo; De Villiers Kuun, Morgan; Clowes, Iona; junadyj@cheltladiescollege.org; Manna, Ivy; Rolinson, Lorna +2 others ↗

SURVEY!

Please fill out this survey to pick your Sixth form taster sessions! "link" - this is a tester email there is no actual link If you receive this its from Ola, please send me a screenshot!

Student Interface:

For this interface, I tweaked the design for this interface prior to coding it, as stated above, I found better ways to implement the algorithms.

The student interface includes approximately 2 php files (studentI.php and handlestudents.php) and uses a pre-existing completed.csv file, which stores the usernames of the students who have completed the survey.

The file below displays 7 drop-down lists for the students to pick choices from. One of these drop-down lists allows the student to pick their username from a list taken from the students.csv file given to us by Mr. Wain.

```

1 <html>
2 <body>
3 <?php
4 //student interface
5 $file = fopen("subject.csv", "r");
6 $subjects = fgetcsv($file);
7 fclose($file);
8 //puts the csv file provided into a 2d array
9 $studentTable = array();
10 if (($handle = fopen("students.csv", "r")) !== FALSE) {
11     while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
12         $studentTable[] = $data;
13     }
14 }
15 //creates list that will store list of girls emails
16 $usernames = array();
17 //runs through studentTable and appends emails to toEmail.
18 for ($I = 0; $I < sizeof($studentTable); $I++) {
19     array_push($usernames, $studentTable[$I][1]);
20 }
21
22 ?>
```

In the following html and php, I have used the two languages concurrently in displaying my html elements from php lists. This section is also lacking validation, but I will complete this in my next iteration of coding.

```
23 | <h2> Select your subjects! </h2>
24 | <form method="POST" action='handlestudents.php'>
25 |   <p> Select your name </p>
26 |   <select required="required" id='userN' name='userN'>
27 |     <option value="">Choose one</option>
28 |     <?php
29 |       // Iterating through the usernames array to display them
30 |       foreach($usernames as $item) {
31 |         ?>
32 |         <option value="<?php echo $item; ?>><?php echo $item; ?></option>
33 |         <?php
34 |         }
35 |         ?>
36 |       </select>
37 |     <p> Select your 1st choice subject </p>
38 |     <select required="required" id='choice1' name='choice1'>
39 |       <option value="">Choose a subject</option>
40 |       <?php
41 |         // Iterating through the subjects array
42 |         foreach($subjects as $item) {
43 |           ?>
44 |           <option value="<?php echo $item; ?>><?php echo $item; ?></option>
45 |           <?php
46 |           }
47 |           ?>
48 |         </select>
49 |     <p> Select your 2nd choice subject </p>
50 |     <select required="required" id='choice2' name='choice2'>
51 |       <option value="">Choose a subject</option>
52 |       <?php
53 |         // Iterating through the subjects array
54 |         foreach($subjects as $item) {
55 |           ?>
56 |           <option value="<?php echo $item; ?>><?php echo $item; ?></option>
57 |           <?php
58 |           }
59 |           ?>
60 |         </select>
```

```

61 <p> Select your 3rd choice subject </p>
62 <select required="required" id='choice3' name='choice3'>
63   <option value="" >Choose a subject</option>
64   <?php
65     // Iterating through the subjects array
66     foreach($subjects as $item) {
67       ?>
68       <option value="php echo $item; ?&gt;"&gt;&lt;?php echo $item; ?&gt;&lt;/option&gt;
69     ?
70     ?&gt;
71   ?
72 &lt;/select&gt;
73 &lt;p&gt; Select your 4th choice subject &lt;/p&gt;
74 &lt;select required="required" id='choice4' name='choice4'&gt;
75   &lt;option value="" &gt;Choose a subject&lt;/option&gt;
76   &lt;?php
77     // Iterating through the subjects array
78     foreach($subjects as $item) {
79       ?&gt;
80       &lt;option value="<?php echo $item; ?&gt;"&gt;&lt;?php echo $item; ?&gt;&lt;/option&gt;
81     ?
82     ?&gt;
83   ?
84 &lt;/select&gt;
85 &lt;p&gt; Select your 5th choice subject &lt;/p&gt;
86 &lt;select required="required" id='choice5' name='choice5' &gt;
87   &lt;option value="" &gt;Choose a subject&lt;/option&gt;
88   &lt;?php
89     // Iterating through the subjects array
90     foreach($subjects as $item) {
91       ?&gt;
92       &lt;option value="<?php echo $item; ?&gt;"&gt;&lt;?php echo $item; ?&gt;&lt;/option&gt;
93     ?
94     ?&gt;
95   ?
96 &lt;/select&gt;
97 &lt;p&gt; Select your 6th choice subject (none is an option) &lt;/p&gt;
98 &lt;select required="required" id='choice6' name='choice6'&gt;
99   &lt;option value="" &gt;Choose a subject&lt;/option&gt;
100  &lt;?php
101    // Iterating through the subjects array
102    foreach($subjects as $item) {
103      ?&gt;
104      &lt;option value="<?php echo $item; ?&gt;"&gt;&lt;?php echo $item; ?&gt;&lt;/option&gt;
105    ?
106    ?&gt;
107      &lt;option value = ""&gt; None &lt;/option&gt;
108    ?
109  &lt;/select&gt;
110 &lt;input type="submit" value="Submit"&gt;
111 &lt;/form&gt;
112 &lt;/body&gt;
113 &lt;/html&gt;
</pre

```

The next php file, named handlestudents.php, essentially inserts the students choices into the mysql database table, studentChoices, by their college assigned ID. This page also appends the username of the girl that has just completed the survey into the file completed.csv for future reference.

```
1  <html>
2  <body>
3  <h1> THANK YOU FOR SUBMITTING YOUR SURVEY! </h1>
4  <h4>If you have a problem, please email: JacksonAE@cheltladiescollege.org </h4>
5  <?php
6  //puts students.csv into a 2d array called studentTable
7  if (($handle = fopen("students.csv", "r")) !== FALSE) {
8      while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
9          $studentTable[] = $data;
10     }           fclose($handle);
11 }
12 // creates connection with mysql table, confusing stuff
13 if ($_POST) {
14     $con = mysqli_connect("localhost", "root", "", "mysql");
15
16     if (!$con)
17     {
18         die('Could not connect: ' . mysqli_connect_error());
19     }
20     //require_once "config.php";
21 //opens completed.csv file and appends username of girl who's done survey
22 $completedf = fopen("completed.csv", "r");
23 $completedli = fgetcsv($completedf);
24 fclose($completedf);
25 array_push($completedli, $_POST['userN']);
26 $completedf = fopen('completed.csv', 'w');
27 fputcsv($completedf, $completedli);
28 fclose($completedf);
29 //gets the id of the student with chosen username
30 $studentID = "";
31 for ($I = 0; $I < sizeof($studentTable); $I++) {
32     if ($studentTable[$I][1] == $_POST['userN']){
33         $studentID = $studentTable[$I][0];
34     }
35 }
```

```

36 //places all posted data in variables
37 $choice1 = $_POST['choice1'];
38 $choice2 = $_POST['choice2'];
39 $choice3 = $_POST['choice3'];
40 $choice4 = $_POST['choice4'];
41 $choice5 = $_POST['choice5'];
42 $choice6 = $_POST['choice6'];
43 //
44 // $studentID = mysql_real_escape_string($studentID);
45 // removes all special characters so that it won't affect the sql statements
46 $choice1 = mysqli_real_escape_string($con,$choice1);
47 $choice2 = mysqli_real_escape_string($con,$choice2);
48 $choice3 = mysqli_real_escape_string($con,$choice3);
49 $choice4 = mysqli_real_escape_string($con,$choice4);
50 $choice5 = mysqli_real_escape_string($con,$choice5);
51 $choice6 = mysqli_real_escape_string($con,$choice6);
52 //query places girls choices into studentchoices table usin sql statement
53 $query = "
54     INSERT INTO studentchoices (StudentID, Choice1, Choice2, Choice3,
55         Choice4, Choice5, Choice6) VALUES ($studentID, '$choice1',
56         '$choice2', '$choice3', '$choice4',
57         '$choice5', '$choice6');";
58 //disconnects from database
59 if (mysqli_query($con, $query)) {
60     echo "Bye!";
61 }
62
63 mysqli_close($con);
64 }
65 ?>
66 </body>
67 </html>

```

The completed.csv file is empty until populated.

Testing:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Pass/Fail	Testing Evidence
Be able to pick from list of names.	Options of girls names show up to pick from.	DROP DOWN NAME	List of names from student csv	P	1
Be able to pick from list of subjects.	Options of subjects show up to pick from	DROP DOWN SUBJECT	Should display list of subjects entered by Mrs Jackson	P	2
Click button submit with correct data		SUBMIT CHOICES	Submission into mysql database	P	3,4,5
Submit with empty subject choices	Validation of user input	SUBMIT CHOICES	Error message	P	6
Submit with empty name choices	Validation of user input	SUBMIT CHOICES	Error message	P	7

Submit with subjects that are the same	Validation of user input	SUBMIT CHOICES	Error message	F	8,9
--	--------------------------	----------------	---------------	---	-----

Testing Evidence:

The Subject list is: Maths, English, Chemistry, Biology, French, Chinese

List of Names is: Ola,Morgan,Iona,Jeanna,Ivy,Lorna,Alexander,Andrew

1	<p>Select your subjects!</p> <p>Select your name</p> <p>Choose one ▾</p> <p>Choose one choice subject</p> <p>Ola</p> <p>Morgan</p> <p>Iona</p> <p>Jeanna</p> <p>Ivy</p> <p>Lorna</p> <p>Alexander</p> <p>Andrew</p> <p>Select your 5th choice subject</p>	<p>Select your subjects!</p> <p>Select your name</p> <p>Ola ▾</p> <p>Select your 1st choice subject</p> <p>Choose a subject ▾</p>
2	<p>Select your 1st choice subject</p> <p>Choose a subject ▾</p> <p>Choose a subject choice subject</p> <p>Maths</p> <p>English</p> <p>Chemistry</p> <p>Biology</p> <p>French</p> <p>Chinese</p> <p>Select your 5th choice subject</p>	<p>Select your subjects!</p> <p>Select your name</p> <p>Ola ▾</p> <p>Select your 1st choice subject</p> <p>Maths ▾</p>

3	<p>Select your subjects!</p> <p>Select your name</p> <p>Ola ▾</p> <p>Select your 1st choice subject</p> <p>Maths ▾</p> <p>Select your 2nd choice subject</p> <p>English ▾</p> <p>Select your 3rd choice subject</p> <p>Chemistry ▾</p> <p>Select your 4th choice subject</p> <p>Biology ▾</p> <p>Select your 5th choice subject</p> <p>French ▾</p> <p>Select your 6th choice subject (none is an option)</p> <p>Chinese ▾</p> <p><input type="button" value="Submit"/></p>														
4	<p>THANK YOU FOR SUBMITTING YOUR SURVEY!</p> <p>If you have a problem, please email: JacksonAE@cheltladiescollege.org</p> <p>Bye!</p>														
5	<p>+ Options</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">StudentID</th> <th style="text-align: left; padding: 2px;">Choice1</th> <th style="text-align: left; padding: 2px;">Choice2</th> <th style="text-align: left; padding: 2px;">Choice3</th> <th style="text-align: left; padding: 2px;">Choice4</th> <th style="text-align: left; padding: 2px;">Choice5</th> <th style="text-align: left; padding: 2px;">Choice6</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 2px;">1</td> <td style="text-align: center; padding: 2px;">Maths</td> <td style="text-align: center; padding: 2px;">English</td> <td style="text-align: center; padding: 2px;">Chemistry</td> <td style="text-align: center; padding: 2px;">Biology</td> <td style="text-align: center; padding: 2px;">French</td> <td style="text-align: center; padding: 2px;">Chinese</td> </tr> </tbody> </table> <p style="margin-top: 20px;"> <input type="checkbox"/> Show all Number of rows: <select style="width: 40px; height: 1.2em; border: 1px solid #ccc; border-radius: 2px; padding: 2px 5px; font-size: 0.9em;">25 ▾</select> Filter rows: <input style="width: 150px; height: 1.2em; border: 1px solid #ccc; border-radius: 2px; padding: 2px 5px; font-size: 0.9em;" type="text"/> </p>	StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6	1	Maths	English	Chemistry	Biology	French	Chinese
StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6									
1	Maths	English	Chemistry	Biology	French	Chinese									
6	<p>Select your subjects!</p> <p>Select your name</p> <p>Morgan ▾</p> <p>Select your 1st choice subject</p> <p>Choose a subject ▾</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-left: 10px;"> <p>! Please select an item in the list.</p> </div>														
7	<p>Select your subjects!</p> <p>Select your name</p> <p>Choose one ▾</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin-left: 10px;"> <p>! Please select an item in the list.</p> </div> <p>Choose a subject ▾</p> <p>Select your 2nd choice subject</p> <p>Choose a subject ▾</p>														

8	<p>Select your subjects!</p> <p>Select your name <input type="text" value="Morgan"/></p> <p>Select your 1st choice subject <input type="text" value="Maths"/></p> <p>Select your 2nd choice subject <input type="text" value="Maths"/></p> <p>Select your 3rd choice subject <input type="text" value="Maths"/></p> <p>Select your 4th choice subject <input type="text" value="Maths"/></p> <p>Select your 5th choice subject <input type="text" value="Maths"/></p> <p>Select your 6th choice subject (none is an option) <input type="text" value="Maths"/></p> <p style="text-align: center;">THANK YOU FOR SUBMITTING YOUR SURVEY!</p> <p>If you have a problem, please email: JacksonAE@cheltladiescollege.org</p> <p><input type="button" value="Submit"/> <input type="button" value="Bye!"/></p>																					
9	<p>+ Options</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">StudentID</th> <th style="background-color: #cccccc;">Choice1</th> <th style="background-color: #cccccc;">Choice2</th> <th style="background-color: #cccccc;">Choice3</th> <th style="background-color: #cccccc;">Choice4</th> <th style="background-color: #cccccc;">Choice5</th> <th style="background-color: #cccccc;">Choice6</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Maths</td> <td>English</td> <td>Chemistry</td> <td>Biology</td> <td>French</td> <td>Chinese</td> </tr> <tr> <td>2</td> <td>Maths</td> <td>Maths</td> <td>Maths</td> <td>Maths</td> <td>Maths</td> <td>Maths</td> </tr> </tbody> </table> <p style="margin-top: 10px;"> <input type="checkbox"/> Show all Number of rows: <input type="text" value="25"/> Filter rows: <input type="text" value="Search this table"/> </p>	StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6	1	Maths	English	Chemistry	Biology	French	Chinese	2	Maths	Maths	Maths	Maths	Maths	Maths
StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6																
1	Maths	English	Chemistry	Biology	French	Chinese																
2	Maths	Maths	Maths	Maths	Maths	Maths																

In the appropriate stage of development, I will add the validation to the criteria I failed.

Second Staff Interface:

All the processing for this interface is on one php page names staffinterface2.php. This page allows Mrs Jackson, to see students who have and haven't completed the survey yet. It also allows her to send them an email reminder of their obligation to fill it out. All the code for this page is on one php document named staffinterface2.php

```

1  <?php
2  $completedf = fopen("completed.csv", "r");
3  $completedli = fgetcsv($completedf);
4  fclose($completedf);
5
6  //puts the csv file provided into a 2d array
7  $studentTable = array();
8  if (($handle = fopen("students.csv", "r")) !== FALSE) {
9      while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
10         $studentTable[] = $data;
11     }
12 }
13 //creates list that will store list of girls emails
14 $usernames = array();
15 $responses = array();
16 //runs through studentTable and appends emails to toEmail.
17 for ($I = 0; $I < sizeof($studentTable); $I++) {
18     array_push($usernames, $studentTable[$I][1]);
19 }
20 for ($I = 0; $I < sizeof($usernames); $I++) {
21     if (in_array($usernames[$I], $completedli) == False) {
22         array_push($responses, $usernames[$I]);
23     }
24 } else{
25
26 }
27
28 ?>

```

The code above puts all the girls who have completed the survey in a list from the existing file. It then also puts the students csv file into a 2d array. And then from this 2d array, a list of studentusernames is created. I then use this usernames list to determine who hasn't completed the survey with the completed response list.

```

29  <?php
30  if(isset($_POST['SubmitButton'])) {
31      $to_email = array();
32      $email = "";
33      for ($I = 0; $I < sizeof($nresponses); $I++) {
34          $email = $nresponses[I] + "cheltladiescollege.org";
35          array_push($to_email, $email);
36      }
37      $email_subject = "Reminder to fill out this survey"; // email subject line
38      $from = 'JacksonAE@cheltladiescollege.org';
39
40      // To send HTML mail, the Content-type header must be set
41      $headers = 'MIME-Version: 1.0' . "\r\n";
42      $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
43
44      // Create email headers
45      $headers .= 'From: ' . $from . "\r\n".
46                  'Reply-To: ' . $from . "\r\n".
47                  'X-Mailer: PHP/' . phpversion();
48
49      // Compose a simple HTML email message
50      $message = '<html><body>';
51      $message .= '<h3> SURVEY! </h3>';
52      $message .= '<p style="color:#080;font-size:18px;">Please fill out this sur';
53      $message .= '</body></html>';
54      //mail($email_to, $email_subject, $message, $headers);
55  }
56
57 ?>

```

The above code sends an email reminder to all those in the nresponses list, if the button send reminder is clicked.

```

59  <html>
60  <head>
61      <h1 align = "centre"> RESPONSES </h1>
62      <h3> RESPONSES RECEIVED </h3>
63      <style>
64          table, th, td {
65              border: 1px solid black;
66          }
67      </style>
68  </head>
69  <body>
70  <?php
71      //connect to mysql server with host,username,password
72      $connection=mysqli_connect('localhost','root','','mysql') or die(mysqli_connect_error());
73
74      //execute a mysql query to retrieve all the users from users table
75      //if query fails stop further execution and show mysql error
76      $query=mysqli_query($connection,"SELECT * FROM studentchoices") or die(mysqli_connect_error());
77
78      //if we get any results we show them in table data
79      if(mysqli_num_rows($query)>0):
80
81  ?>

```

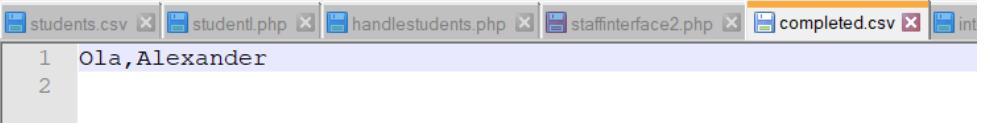
```
82 <table>
83   <tr>
84     <td align="center">Id</td>
85     <td align="center">First Choice</td>
86     <td align="center">Second Choice</td>
87     <td align="center">Third Choice</td>
88     <td align="center">Fourth Choice</td>
89     <td align="center">Fifth Choice</td>
90     <td align="center">Sixth Choice</td>
91   </tr>
92   <?php
93   // looping
94   while($row=mysqli_fetch_object($query)) : ?>
95   <tr>
96     <td align="center"><?php echo $row->StudentID; ?></td>
97     <td align="center"><?php echo $row->Choice1; ?></td>
98     <td align="center"><?php echo $row->Choice2; ?></td>
99     <td align="center"><?php echo $row->Choice3; ?></td>
100    <td align="center"><?php echo $row->Choice4; ?></td>
101    <td align="center"><?php echo $row->Choice5; ?></td>
102    <td align="center"><?php echo $row->Choice6; ?></td>
103  </tr>
104  <?php endwhile; ?>
105 </table>
106 <?php
107 // no result show
108 else: ?>
109 <h3>No Results found.</h3>
110 <?php endif; ?>
111 </table>
112 <h3> AWAITING RESPONSES </h3>
113 <ul>
114   <?php
115   // Iterating through the product array
116   foreach($nresponses as $item) {
117     ?
118     <li> <?php echo $item; ?> </li>
119   <?php
120   ?
121   ?
122 </ul>
123 <form action="#" method="post">
124   <p> SEND REMINDER </p>
125   <input type="submit" name="SubmitButton">
126 </form>
127
128 <p> EDIT THE GIRLS </p>
129   <a href="staffinterface3.php" >NEXT PAGE </a>
130 </body>
131 </html>
```

The code pictures above display the table to the user and also display the people who haven't replied to the survey. Also provides a button to send reminders to pupils as well as a button to travel to the next page.

Testing:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Pass/Fail (P/F)	Testing Evidence
Leave Ola, Alexander in the completed csv file.	Display names of whom we are awaiting responses.	Awaiting Response Table	Morgan, Iona, Jeanna, Ivy, Lorna, Andrew	P	1,2
SEND Reminder button is clicked.	Email reminders – usability features*	Send Email Responses	Reminders are emailed to all those who have not filled out the survey.	F	3
Displays studentchoices table.	Display of girls who have responded in table layout.	Current Response table.	Should be the exact same table as in mysql table.	P	4,5
Next Page button clicked with awaiting responses from girls	Allows Mrs Jackson to pass onto the next page – usability feature*	Next Page	Take the user to staffinterface3	P	

Testing Evidence:

1	
2	<p>AWAITING RESPONSES</p> <ul style="list-style-type: none"> • Morgan • Iona • Jeanna • Ivy • Lorna • Andrew <p>Send Reminder</p>
3	To test this, the computer tech team would have to whitelist my computer like before as emails are currently blocked. I will come back to test this during my blackbox testing phase.

4	<p>+ Options</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">StudentID</th><th>Choice1</th><th>Choice2</th><th>Choice3</th><th>Choice4</th><th>Choice5</th><th>Choice6</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">2</td><td>Math</td><td>Math</td><td>English</td><td>French</td><td>Math</td><td></td></tr> </tbody> </table> <p style="margin-top: 20px;"> <input type="checkbox"/> Show all Number of rows: <select style="display: inline-block; width: 40px;">25</select> Filter rows: <input type="text" value="Search this table"/> </p> <p style="margin-top: 10px; border: 1px solid #ccc; padding: 2px; display: inline-block;">Query results operations</p>	StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6	2	Math	Math	English	French	Math	
StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6									
2	Math	Math	English	French	Math										
5	<h1 style="font-size: 1.5em; font-weight: bold;">RESPONSES</h1> <h2 style="font-size: 1em; font-weight: bold; margin-top: 10px;">RESPONSES RECEIVED</h2> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="text-align: left;">Id</th><th>First Choice</th><th>Second Choice</th><th>Third Choice</th><th>Fourth Choice</th><th>Fifth Choice</th><th>Sixth Choice</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">2</td><td>Math</td><td>Math</td><td>English</td><td>French</td><td>Math</td><td></td></tr> </tbody> </table> <p style="margin-top: 10px;">.</p>	Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice	2	Math	Math	English	French	Math	
Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice									
2	Math	Math	English	French	Math										

Third Staff Interface:

All the processing for this interface is on one php page names staffinterface3.php. These files will allow Mrs Jackson to see the database and also edit the database. She is also given the option to arrange the girls on this page but this will be done at a later stage of development.

```

1  <?php
2  $completedf = fopen("completed.csv", "r");
3  $completedli = fgetcsv($completedf);
4  fclose($completedf);
5
6  //puts the csv file provided into a 2d array
7  $studentTable = array();
8  if (($handle = fopen("students.csv", "r")) !== FALSE) {
9      while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
10         $studentTable[] = $data;
11     }
12 }
13 //creates list that will store list of girls emails
14 $usernames = array();
15 $responses = array();
16 //runs through studentTable and appends emails to toEmail.
17 for ($I = 0; $I < sizeof($studentTable); $I++) {
18     array_push($usernames, $studentTable[$I][1]);
19 }
20 for ($I = 0; $I < sizeof($usernames); $I++) {
21     if (in_array($usernames[$I], $completedli) == False) {
22         array_push($responses, $usernames[$I]);
23     }
24 }
25
26 }
27
28 //student interface subject list
29 $file = fopen("subject.csv", "r");
30 $subjects = fgetcsv($file);
31 fclose($file);
32 ?>

```

This section of the file above, essentially puts the files (completedf, students) into php data structures such as 1d arrays and 2d arrays. Once this is done, it deduces the non responses list from the completed list and all the students list.

```

33  <?php
34  //runs through what happens when submit button is clicked.
35  if(isset($_POST['SubmitButton'])) {
36      $con = mysqli_connect("localhost","root","","mysql");
37
38      if (! $con)
39      {
40          die('Could not connect: ' . mysqli_connect_error());
41      }
42 //places all posted data in variables
43 $studentID = $_POST['StudentID'];
44 $choice = $_POST['choice'];
45 $subject = $_POST['subject'];
46
47 // $studentID = mysql_real_escape_string($studentID);
48 //removes all special characters so that it won't affect the sql statements
49 $studentID = mysqli_real_escape_string($con,$studentID);
50 $choice = mysqli_real_escape_string($con,$choice);
51 $subject = mysqli_real_escape_string($con,$subject);
52
53 //query places girls choices into studentchoices table usin sql statement
54 $query = "
55     UPDATE studentchoices
56     SET $choice = '$subject'
57     WHERE StudentID = $studentID;";
58 //disconnects from database
59     if (mysqli_query($con, $query)) {
60         echo "Query changed!";
61     }
62
63
64 mysqli_close($con);
65
66 }
67 ?>

```

This section of the code above, denotes the actions to take place once the submit button is clicked. These code blocks carry out the steps denoted in the editable() procedure that I created in the design section. This algorithm edits the studentchoices table using user input, and displays the results on the page.

```

68 <html>
69 <head>
70 <h1 align = "centre"> RESPONSES </h1>
71 <h3> RESPONSES RECEIVED </h3>
72 <style>
73 table, th, td {
74     border: 1px solid black;
75 }
76 </style>
77 </head>
78 <body>
79 <?php
80 //connect to mysql server with host,username,password
81 //if connection fails stop further execution and show mysql error
82 $connection=mysqli_connect('localhost','root','','mysql') or die(mysqli_connect_error());
83 //select a database for given connection
84 //if database selection fails stop further execution and show mysql error
85 //mysqli_select_db('mysql',$connection) or die(mysqli_error());
86
87 //execute a mysql query to retrieve all the users from users table
88 //if query fails stop further execution and show mysql error
89 $query=mysqli_query($connection,"SELECT * FROM studentchoices") or die(mysqli_connect_error());
90
91 //if we get any results we show them in table data
92 if(mysqli_num_rows($query)>0):
93
94 ?>

```

The code above in the php section connects to the database, to allow my code to access its contents and display them on the screen.

```
95 <table>
96   <tr>
97     <td align="center">Id</td>
98     <td align="center">First Choice</td>
99     <td align="center">Second Choice</td>
100    <td align="center">Third Choice</td>
101    <td align="center">Fourth Choice</td>
102    <td align="center">Fifth Choice</td>
103    <td align="center">Sixth Choice</td>
104  </tr>
105  <?php
106  // looping
107  while($row=mysqli_fetch_object($query)) :?>
108  <tr>
109    <td align="center"><?php echo $row->StudentID; ?></td>
110    <td align="center"><?php echo $row->Choice1; ?></td>
111    <td align="center"><?php echo $row->Choice2; ?></td>
112    <td align="center"><?php echo $row->Choice3; ?></td>
113    <td align="center"><?php echo $row->Choice4; ?></td>
114    <td align="center"><?php echo $row->Choice5; ?></td>
115    <td align="center"><?php echo $row->Choice6; ?></td>
116  </tr>
117  <?php endwhile;?>
118 </table>
119 <?php
120 // no result show
121 else: ?>
122 <h3>No Results found.</h3>
123 <?php endif; ?>
124 </table>
```

The html code above defines the table and fills said table with the data in the studentchoices table from the mysql database. If there are no results in the table, a message is displayed to highlight this.

```
125 <h3> AWAITING RESPONSES </h3>
126 <ul>
127   <?php
128   // Iterating through the product array
129   foreach($nresponses as $item) {
130     ?>
131     <li> <?php echo $item; ?> </li>
132   <?php
133   }
134   ?>
135 </ul>
```

The html and php above, combined does the job of the awaiting responses procedure, and displays a list of those who have not filled out the survey. This list was created prior on the page.

```

136 <h4> Edit The Table</h4>
137 <form action="#" method="post">
138 <p> Id of the person you want to edit </p>
139 <input type="number" name="StudentID"> <br>
140 <p> Which choice would you like to edit (Choice1, Choice2 etc) </p>
141 <select id = "choice" name = "choice" >
142 <option value="Choice1"> Choice 1 </option>
143 <option value="Choice2"> Choice 2 </option>
144 <option value="Choice3"> Choice 3 </option>
145 <option value="Choice4"> Choice 4 </option>
146 <option value="Choice5"> Choice 5 </option>
147 <option value="Choice6"> Choice 6 </option>
148 </select>
149 <p> What subject would you like to change this to (Must be exact)</p>
150 <select id='subject' name='subject'>
151 <option value="" >Choose a subject</option>
152 <?php
153 // Iterating through the subjects array
154 <foreach($subjects as $item){ <?
155 <?
156 <option value="<?php echo $item; ?>"><?php echo $item; ?></option>
157 <?php
158 <?
159 <?
160 </select>
161 <input type="submit" name="SubmitButton">
162 </form>

```

The mix of php and html above performs the editable procedure. Here, the html is responsible for making a drop down list of the subject list and the choices that Mrs Jackson can pick from to edit the database.

```

164 <p> Previous Page </p>
165 <a href="staffinterface2.php" >Back </a>
166 <p> Sort Girls </p>
167 <a href="sortgirls.php" >SORT THE GIRLS! </a>
168
169
170 </body>
171 </html>

```

The code above provides a link to the previous page and a link to the page that will sort the girls, which will be completed in the second cycle of development!

Testing:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Pass/Fail (P/F)	Testing evidence
Valid Student ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	Page refreshes, and values have been changed on screen and in php database	P	1&2
Invalid (non existent) Student	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so	P	3&4

ID, Valid Choice, Valid Subject			table is not updated.		
Valid Student ID, Invalid (non existent) Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	Drop down list makes this impossible.	P	5
Valid Student ID, Valid Choice, Invalid (non existent) Subject	Staff can edit the above table	EDIT TABLE	Drop down list of subjects input only allows correct subject submission, so makes this impossible.	P	6
Visible link	Choice at the bottom of the page to sort girls	ARRANGE GIRLS	Screenshot visible and functioning button	P	7
After validation in prior areas of code, there will be no erroneous data in the program. Therefore no need to test under those circumstances. Will do white box testing here along the way to make sure the programs subroutines work. Then test with the data from last year to simulate a real issue.	Arrange the girls into correct sessions	ARRANGE GIRLS	Sorts girls to Mrs Jackson's liking.	F	N/A – not completed yet

Testing Evidence:

1

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
2	Math	Math	English	French	Math	

AWAITING RESPONSES

- Morgan
- Iona
- Jeanna
- Ivy
- Lorna
- Andrew

Edit The Table

Id of the person you want to edit

Which choice would you like to edit (Choice1, Choice2 etc)

What subject would you like to change this to (Must be exact)

2

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
2	Math	Math	Chemistry	French	Math	

[+ Options](#)

StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6
2	Math	Math	Chemistry	French	Math	

Show all | Number of rows: Filter rows:

3

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
2	Math	Math	Chemistry	French	Math	

AWAITING RESPONSES

- Morgan
- Iona
- Jeanna
- Ivy
- Lorna
- Andrew

Edit The Table

Id of the person you want to edit

Which choice would you like to edit (Choice1, Choice2 etc)

What subject would you like to change this to (Must be exact)

4

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
2	Math	Math	Chemistry	French	Math	

	<table border="1"> <thead> <tr> <th colspan="7">+ Options</th></tr> <tr> <th>StudentID</th><th>Choice1</th><th>Choice2</th><th>Choice3</th><th>Choice4</th><th>Choice5</th><th>Choice6</th></tr> </thead> <tbody> <tr> <td>2</td><td>Math</td><td>Math</td><td>Chemistry</td><td>French</td><td>Math</td><td></td></tr> </tbody> </table> <p>Show all Number of rows: <input type="button" value="25"/> Filter rows: <input type="text" value="Search this table"/></p>	+ Options							StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6	2	Math	Math	Chemistry	French	Math	
+ Options																						
StudentID	Choice1	Choice2	Choice3	Choice4	Choice5	Choice6																
2	Math	Math	Chemistry	French	Math																	
5	<p>Edit The Table</p> <p>Id of the person you want to edit</p> <input type="text" value="2"/> <p>Which choice would you like to edit (Choice1, Choice2 etc)</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <input type="button" value="Choice 1 ▾"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 10px; margin-bottom: 5px;" type="button" value="Choice 1"/> <input type="button" value="Choice 2"/> <input type="button" value="Choice 3"/> <input type="button" value="Choice 4"/> <input type="button" value="Choice 5"/> <input type="button" value="Choice 6"/> </div> <p>What subject would you like to change this to (Must be exact)</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <input type="button" value="Submit"/> </div>																					
6	<p>Edit The Table</p> <p>Id of the person you want to edit</p> <input type="text" value="2"/> <p>Which choice would you like to edit (Choice1, Choice2 etc)</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <input type="button" value="Choice 1 ▾"/> </div> <p>What subject would you like to change this to (Must be exact)</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> <input type="button" value="Choose a subject ▾"/> <input style="background-color: #007bff; color: white; border: none; padding: 2px 10px; margin-bottom: 5px;" type="button" value="Choose a subject"/> <input type="button" value="Submit"/> </div> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> Chemistry Maths English </div> <p>SORT THE GIRLS! DOWNLOAD FILE</p>																					
7	<p>Sort Girls</p> <p><u>SORT THE GIRLS!</u></p>																					

Stakeholder Meetings:

Feedback from Mrs AE Jackson:

- Everything is good so far
- However, she had an additional suggestion for the add subject area. For that section, she said she would like an additional option to add subjects, which allows her to upload a csv file which contains preadded subjects.

Notes from meeting with Mr Wain:

- Currently, my code is based off a students file which includes student id, username and email.
- Mr Wain suggested that I only stored studentid and username as it is a waste of memory as username and email are variations of each other.
- He then provided me with the UC5 students file for future testing.

This is what the students file looks like:

```

1 StudentID,Username
2 084626705547,ClarkP
3 085447868892,CheungJ
4 085807705547,Ipkendanzo
5 090357705547,KoTY
6 091504705547,HewettA
7 091539643505,stokm
8 091801705547,YipV
9 092647705547,howeK
10 092927533424,JonesMa
11 092954705547,FeeO
12 093245705547,ZhongE
13 094003533424,LeeC
14 094606579518,KeatingeH
15 094854533424,BamberG
16 095358533424,LongM
17 095424410122,GeP
18 101130705547,MillerE
19 101258705547,McMahonE
20 101321965465,cavanc
21 101730705547,PanA
22 101900533424,DuToitI
23 102023353588,HanS
24 102220705547,PampamK
25 102659533424,LloydP
26 102855705547,LustmanR
27 102906533424,TaylorF
28 103819533424,IbrahimM
29 103922705547,MickoI
30 104109705547,AgarwalK
31 104144705547,ZhangV
32 104244705547,BinghamL
33 104903705547,KanazawaR
34 104917533424,BrutonJ
35 105040579518,IttipakornX
36 105401705547,NurjanovaA
37 105959727947,QianSu
38 110117705547,StevensH
39 110812705547,VathanakulJa
40 110920533424,Picton-JonesA
41 111131533424,AkinyemiA
42 111401705547,FungJ

```

Notes and Changes:

- Add an upload csv link section to the enter subjects page for Mrs Jackson
- Adapt code to accommodate for the new format of the students csv file given by Mr Wain.

STAGE 1 COMPLETED

STAGE 2

- Implement changes and suggestions from previous stage.
- Functional First Staff Interface, with the newly uploaded link.
- Functional Third Staff Interface, including the aspect of arranging girls into sessions.
- Meeting with stakeholder, Mrs Jackson. Ask for feedback and changes/ suggestions on completed sections.

Implementation of Changes and Suggestions:

Add an upload csv file section to staff interface 1, for the subjects:

To complete this section, I have added an extra file named handlesubjects.php and extra code on the subjects.php page. Previously, the subjects.php page displayed text boxes for the addition and removal of subject choices. Now in addition to this, I have added html code that provides the upload form for the user. The php code on the handlesubjects.php file saves the file in a location the rest of my files will be able to access, under the name "subject.csv", as this is how the rest of my code has been addressing the file, meaning I did not have to change any code outside of this webpage. In addition, I copy and pasted the code from interface2.php, which is a page that allows Mrs Jackson to log out or go to the second staff interface as well as send students a link to the student interface via email . I did this as the two pages are meant to have the same function.

Extra testing:

I am going to need to check the data in subject.csv to see if it matches the data in the file, I have just submitted. I will also need to show, that the file does its function when displaying the dropdown list of subjects on the student interface and third staff interface.

Code:

```
</form>
Or you may upload a csv file with your predetermined subjects
<form method="post" action='handlesubjects.php' enctype = "multipart/form-data">
    Select csv file to upload:
    <input type="file" name="file" id="file">
    <input type="submit" value="Upload csv file" name="submit">
</form>
```

This is the additional html code above that I added to the subjects.php file, so that it would allow the user upload files onto my server. This indicated by the file upload form.

```

1  <?php
2  //outlines the name I would like to give the file
3  $storagename = "subject.csv";
4  //if a file has been submitted
5  if(isset($_POST["submit"])) {
6      //this puts the file given by the user into a file named subject csv
7      //this file is automatically put into the same folder as the rest of documents
8      if (move_uploaded_file($_FILES["file"]["tmp_name"], $storagename)) {
9          echo "The file ". basename( $_FILES["file"]["name"]). " has been uploaded.";
10     } else {
11         //tells the user when things do not go as planned
12         echo "Sorry, there was an error uploading your file.";
13     }
14 }
15 ?>

```

The code above saves the file given by user into the same folder as all my files with the name subject.csv. The code below is identical to that on interface2.php and sends emails to girls with the “link” to their questionnaire.

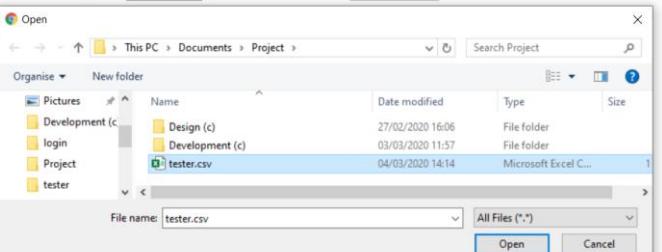
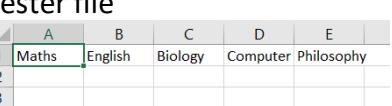
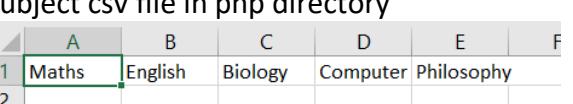
```

18 <?php
19 //puts the students.csv file into a 2d array
20 $studentTable = array();
21 //runs through file appending each line
22 if (($handle = fopen("students.csv", "r")) !== FALSE) {
23     while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
24         $studentTable[] = $data;
25     }
26 }
27 //creates list that will store list of girls emails
28 $toEmail = array();
29 //runs through studentTable and appends emails to toEmail.
30 for ($I = 0; $I < sizeof($studentTable); $I++) {
31     array_push($toEmail, $studentTable[$I][2]);
32 }
33
34 $email_to = implode(',', $toEmail); // your email address
35 $email_subject = "Fill out this survey"; // email subject line
36 $from = 'JacksonAE@cheltladiescollege.org';
37
38 // To send HTML mail, the Content-type header must be set
39 $headers = 'MIME-Version: 1.0' . "\r\n";
40 $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
41
42 // Create email headers
43 $headers .= 'From: ' . $from . "\r\n".
44     'Reply-To: ' . $from . "\r\n".
45     'X-Mailer: PHP/' . phpversion();
46
47 // Compose a simple HTML email message
48 $message = '<html><body>';
49 $message .= '<h3> SURVEY! </h3>';
50 $message .= '<p style="color:#080;font-size:18px;">
51 Please fill out this survey to pick your Sixth form taster sessions!
52 "link" - this is a tester email there is no actual link .....</p>';
53 $message .= '</body></html>';
54 //mail($email_to, $email_subject, $message, $headers);
55 ?>

```

```
57 <head>
58     <meta charset="UTF-8">
59     <title>Welcome</title>
60     <style type="text/css">
61         body{ font: 14px sans-serif; text-align: center; }
62     </style>
63 </head>
64 <body>
65
66     <div class="page-header">
67         <h1>Hi, <b>Mrs Jackson</b>. Welcome to our site.</h1>
68     </div>
69     <p> </p>
70         <a href="logout.php" class="btn btn-danger">Sign Out </a>
71     <p> </p>
72         <a href="staffinterface2.php" >NEXT PAGE </a>
73 </body>
74 </html>
```

Evidence that the code works:

<p>Step 1 – upload file</p>	<p>Or you may upload a csv file with your predetermined subjects Select csv file to upload: <input type="button" value="Choose File"/> No file chosen <input type="button" value="Upload csv file"/></p> <p>Then</p> <p>Or you may upload a csv file with your predetermined subjects Select csv file to upload: <input type="button" value="Choose File"/> No file chosen <input type="button" value="Upload csv file"/></p> 
	<p>Then</p> <p>Or you may upload a csv file with your predetermined subjects Select csv file to upload: <input type="button" value="Choose File"/> tester.csv <input type="button" value="Upload csv file"/></p> <p>Then</p> <p>The file tester.csv has been uploaded.</p> <p>Hi, Mrs Jackson. Welcome to our site.</p> <p>Sign Out</p> <p>NEXT PAGE</p>
<p>Step 2 – check if file has been uploaded</p>	<p>Tester file</p>  <p>Subject csv file in php directory</p> 

Step 3 – check student interface	<p>Select your subjects!</p> <p>Select your name</p> <p><input type="button" value="Choose one ▾"/></p> <p>Select your 1st choice subject</p> <p><input type="button" value="Choose a subject ▾"/> Choose a subject</p> <ul style="list-style-type: none"> Maths English Biology Computer science Philosophy
Step 4 – check third staff interface	<p>What subject would you like to change this to (Must be exact)</p> <p><input type="button" value="Choose a subject ▾"/> Choose a subject</p> <p><input type="button" value="Submit"/></p> <p><input type="button" value="Choose a subject ▾"/> Choose a subject</p> <ul style="list-style-type: none"> Maths English Biology Computer science Philosophy <p>SORT THE GIRLS!</p>

*All these areas functioned as expected after testing.

Adapt code to accommodate for the new format of the students csv file given by Mr Wain.

For this stage of development. I will have to change all the areas that involve me emailing the students, so this includes files such as interface2.php, handlesubjects.php. I will have to change the value they use to reference the third element in a students list element, as there is not third element, as emails have been removed from the file. Instead I will need to concatenate the second element with “cheltladiescollege.org” to create the email. I will also make sure all the loops that run through the studentTable start at 1 instead of 0, as this will exclude, the “StudentID, Username” present as the column headings.

Extra testing:

I will test this section of the interface at the end of my development, as the tech department will whitelist my email to send emails, but if no error message is displayed in this process, it means the code will work regardless.

Code:

interface2.php & handlesubjects.php – before:

```

29 //creates list that will store list of girls emails
30 $toEmail = array();
31 //runs through studentTable and appends emails to toEmail.
32 for ($I = 0; $I < sizeof($studentTable); $I++) {
33     array_push($toEmail, $studentTable[$I][2]);
34 }
35

```

I altered the 2 highlighted in red, to a 1, as this references a non existent element. I will also concatenate “@cheltladiescollege.org” as this will make the emails of the students.

After:

```

25 //creates list that will store list of girls emails
26 $toEmail = array();
27 //runs through studentTable and appends emails to toEmail.
28 for ($I = 1; $I < sizeof($studentTable); $I++) {
29     array_push($toEmail, ($studentTable[$I][1]. "@cheltladiescollege.org")));
30 }

```

*None of the code above proved erroneous during testing

Functional First Staff Interface, with the newly uploaded link:

I understand that in my design, the upload link for student data is on the same page as the subject submission, however I believe the page would be less cluttered and more straight forward if the upload link for student data was on the page between subject submission and the Second Staff interface. As there are two ways Mrs Jackson may choose to submit her data, the same upload students html code will need to be on the two separate pages which are interface2.php and handlesubjects.php, but the actual php code to handle the actual files uploaded will be on the same page regardless of the method used to add subjects.

Code:

```

70 <form method="post" action='studentfile.php' enctype = "multipart/form-data">
71     Select csv file to upload:
72     <input type="file" name="file" id="file">
73     <input type="submit" value="Upload csv file" name="submit">
74 </form>

```

This is the html form upload code that links to the studentfile.php. This is where the upload will be taken care of.

```

1 <html>
2 <?php
3 //outlines the name I would like to give the file
4 $storagename = "students.csv";
5 //if a file has been submitted
6 if(isset($_POST["submit"])) {
7     //this puts the file given by the user into a file named subject csv
8     //this file is automatically put into the same folder as the rest of documents
9     if (move_uploaded_file($_FILES["file"]["tmp_name"], $storagename)) {
10         echo "The file ". basename( $_FILES["file"]["name"]). " has been uploaded.";
11     } else {
12         //tells the user when things do not go as planned
13         echo "Sorry, there was an error uploading your file.";
14     }
15 }
16 ?>
17 <a href="handlesubjects.php" >Go back </a>
18 </html>

```

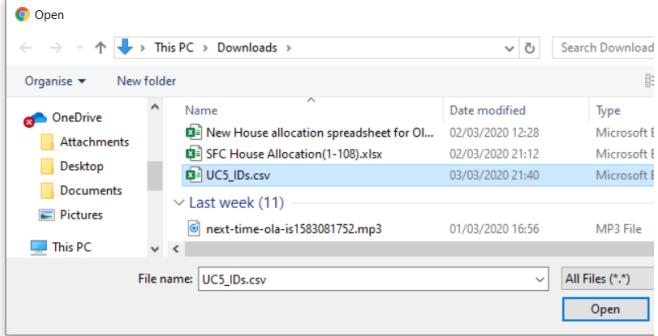
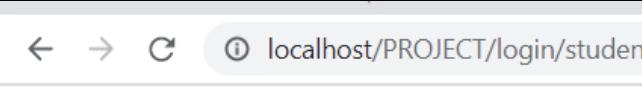
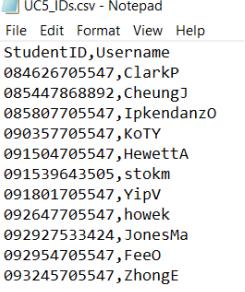
The code above saves the file given by user into the same folder as all my files with the name students.csv.

Testing:

Test Data	Relation to success criteria	Function Name	Expected Output	PASS/ FAIL	Testing evidence
-----------	------------------------------	---------------	-----------------	------------	------------------

Upload link — (additional component)	n/a	n/a	File uploaded should be the same as the students csv file in htdocs.	P	1,2,3,4&5
--	-----	-----	--	---	-----------

Testing Evidence:

1	<p>Hi, Mrs Jackson. Welcome to our site.</p> <p>Select csv file to upload: <input type="button" value="Choose File"/> No file chosen <input type="button" value="Upload csv file"/></p> <p>Sign Out NEXT PAGE</p>
2	<p>Select csv file to upload: <input type="button" value="Choose File"/> UC5_IDs.csv <input type="button" value="Upload csv file"/></p> <p>Sign Out NEXT PAGE</p> 
3	<p>Hi, Mrs Jackson. Welcome to our site.</p> <p>Select csv file to upload: <input type="button" value="Choose File"/> UC5_IDs.csv <input type="button" value="Upload csv file"/></p> <p>Sign Out NEXT PAGE</p>
4	 <p>The file UC5_IDs.csv has been uploaded.Go back</p>
5	<p>UC5_IDs.csv:</p>  <p>StudentID,Username 084626705547,ClarkP 085447868892,CheungJ 085807705547,Ipkendanzo 090357705547,KoTY 091504705547,HewettA 091539643505,stokm 091801705547,YipV 092647705547,howek 092927533424,JonesMa 092954705547,FeeO 093245705547,ZhongE</p> <p>Students.csv:</p>

	students.csv	student1.php	handlesStudents
1	StudentID,Username		
2	084626705547,ClarkP		
3	085447868892,CheungJ		
4	085807705547,IpkendanzO		
5	090357705547,KoTY		
6	091504705547,HewettA		
7	091539643505,stokm		
8	091801705547,YipV		
9	092647705547,howek		
10	092927533424,JonesMa		
11	092954705547,FeeO		
12	093245705547,ZhongE		
13	094003533424,LeeC		
14	094606579518,KeatingeH		
15	094854533424,BamberG		

Arrange the girls!

Functional Third Staff Interface, including the aspect of arranging girls into sessions:

The aspect of the third staff interface that I had not completed is the arranging girls feature. This feature was a huge part of my project, as it required huge amounts of computation. This computation was done on the sortgirls.php page, which was linked to the staffinterface3.php page. I pretty much followed my design exactly when programming this section of my project, other than some extra code I used to fix some aspects of the coding that I did not plan for, and some technicalities to do with the actual language that I was using.

```

1  <?php
2  //make connection with database
3  $con = mysqli_connect("localhost", "root", "", "mysql");
4  //fail safe for if the connection dies
5  if (!$con)
6  {
7      die('Could not connect: ' . mysqli_connect_error());
8  }
9  else{
10    //initialising session arrays
11    $session1 = array();
12    $session2 = array();
13    $session3 = array();
14    $session4 = array();
15    $subjectchoices = array();
16
17    //puts the student csv file provided into a 2d array
18    $studentTable = array();
19    if (($handle = fopen("students.csv", "r")) !== FALSE) {
20        while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
21            $studentTable[] = $data;
22        }
23        fclose($handle);
24    }
25    //creates list that will store list of girls ids
26    $Studentids = array();
27    //runs through studentTable and appends the student ids to Studentids.
28    for ($I = 1; $I < sizeof($studentTable); $I++){
29        array_push($Studentids, $studentTable[$I][0]);
}

```

In the above code, my code makes a connection to the mysql database on the xampp server. Then begins to initialise the session arrays as described in my design. It also uses the same method used previously to create the studentids list from the studentcsv file that is put into a 2d array.

```

30 //counting up all the signups for each subject
31 for ($X = 0; $X < sizeof($Studentids); $X++){
32     $girl = $Studentids[$X];
33     //gets the choices from the student choices table
34     $query = "SELECT Choice1, Choice2, Choice3, Choice4 FROM studentchoices WHERE StudentID = $girl;";
35     $result = mysqli_query($con, $query);
36     //puts result in an array
37     $choices = mysqli_fetch_array($result);
38     array_push($subjectchoices, $choices);
39 }
40 //initialises signups array
41 $signups = array();
42 //these two for lists run through the subjectchoices array and adds the subjects in the list to signups
43 for ($J = 0; $J < sizeof($subjectchoices); $J++){
44     for ($K = 0; $K < 4; $K++){
45         //checks if specific subject has already been added to signups
46         if ((array_key_exists($subjectchoices[$J][$K], $signups)) and ($subjectchoices[$J][$K] != [])){
47             $add = $signups[$subjectchoices[$J][$K]][0]+1;
48             $signups[$subjectchoices[$J][$K]][0] = $add;
49         }
50     }
51 }
52 else{
53     if ($subjectchoices[$J][$K] != []){
54         $signups[$subjectchoices[$J][$K]] = array(1);
55     }
56 }
57 }
58 }
59 }

```

The code in this section is the signups subroutine, which adds all the subjects and the number of signups to the signups list. The only modification that has been made to this subroutine is that, the code in specified selective statements does not run if the element being looked at is empty. Without this extra clause in the if statement, empty spaces were being added to the subjects in signups, and the number of empty spaces was also being counted and considered as a real subject.

```

60 //defines findIndex function, which essentially finds the index of a value in a 2d list
61 function findIndex($item, $li) {
62     //if item not in list function returns negative -1
63     $index = -1;
64     for ($I = 0; $I < sizeof($li); $I++) {
65         if ($li[$I][0] == $item) {
66             $index = $I;
67         }
68     }
69     return $index;
70 }
71
72 //defines in2D function, which finds out if a specific element is in a multidimensional array
73 function in2D($item, $li) {
74     //returns false if element not there
75     $here = False;
76     for ($I = 0; $I < sizeof($li); $I++) {
77         for ($J = 0; $J < sizeof($li[$I]); $J++) {
78             if ($li[$I][$J] == $item) {
79                 $here = True;
80             }
81         }
82     }
83     return $here;
84 }

```

In this code the function findIndex is defined in the same way it was designed, and an extra function which was not planned for was also designed. The additional function (in2D), finds out if an item is

present anywhere in a 2d array and returns a Boolean, whereas the `findIndex` function actually returns the index of the number being searched for.

```
86 //defines fixed subject array
87 $fixedSub = array();
88 //if specific subjects have less than 9 signups, they get added to the fixed subjects
89 foreach ($signups as $key => $value) {
90     if ($signups[$key][0] < 9) {
91         $fixedSub[$key] = $value[0];
92     }
93 }
94 //takes the smallest 4 subjects, and leaves them in the list, everything else is removed.
95 if (sizeof($fixedSub) > 4){
96     //runs until list is less than or equal to 4 subjects.
97     while(sizeof($fixedSub) > 4){
98         $toR = max($fixedSub);
99         $key = array_search($toR, $fixedSub);
100        unset($fixedSub[$key]);
101    }
102 }
103 $i = 0;
104 //reformats the list for ease of programming
105 foreach ($fixedSub as $key => $value) {
106     $fixedSub[$i] = $key;
107     unset($fixedSub[$key]);
108     $i = $i+1;
109 }
110
111 //this whole block of ifs below, depending on how long the list is,
112 //appends the fixed subjects to the sessions.
113 if (sizeof($fixedSub) == 1){
114     array_push($session1, array($fixedSub[0],10,0));
115     array_push($signups[$fixedSub[0]], "1");
116 }
117 if (sizeof($fixedSub) == 2){
118     array_push($session1, array($fixedSub[0],10,0));
119     array_push($signups[$fixedSub[0]], "1");
120     array_push($session2, array($fixedSub[1],10,0));
121     array_push($signups[$fixedSub[1]], "2");
122 }
123 if (sizeof($fixedSub) == 3){
124     array_push($session1, array($fixedSub[0],10,0));
125     array_push($signups[$fixedSub[0]], "1");
126     array_push($session2, array($fixedSub[1],10,0));
127     array_push($signups[$fixedSub[1]], "2");
128     array_push($session3, array($fixedSub[2],10,0));
129     array_push($signups[$fixedSub[2]], "3");
130 }
131 if (sizeof($fixedSub) == 4){
132     array_push($session1, array($fixedSub[0],10,0));
133     array_push($signups[$fixedSub[0]], "1");
134     array_push($session2, array($fixedSub[1],10,0));
135     array_push($signups[$fixedSub[1]], "2");
136     array_push($session3, array($fixedSub[2],10,0));
137     array_push($signups[$fixedSub[2]], "3");
138     array_push($session4, array($fixedSub[3],10,0));
139     array_push($signups[$fixedSub[3]], "4");
140 }
141 //this adds all the rest of the subjects to every session, with the capacities as well
142 foreach ($signups as $key => $value) {
143     if (in_array($key, $fixedSub) == False) {
144         array_push($session1, array($key,10,0));
145         array_push($session2, array($key,10,0));
146         array_push($session3, array($key,10,0));
147         array_push($session4, array($key,10,0));
148     }
149 }
```

```

150      ' //This procedure allocates the number of spots that should be available for each subject
151      //depending on the number of people who have signed up for the subjects. These are stores in the session lists.
152      foreach ($signups as $key => $value) {
153          if ($value[0]>40) {
154              $c = $value[0] - 40;
155              $c = ceil($c/10);
156          if ($c==1){
157              $index = findIndex($key,$session1);
158              $session1[$index][1] = $session1[$index][1] + 10;
159          }
160          if ($c==2){
161              $index = findIndex($key,$session1);
162              $session1[$index][1] = $session1[$index][1] + 10;
163              $index = findIndex($key,$session2);
164              $session2[$index][1] = $session2[$index][1] + 10;
165          }
166          if ($c==3){
167              $index = findIndex($key,$session1);
168              $session1[$index][1] = $session1[$index][1] + 10;
169              $index = findIndex($key,$session2);
170              $session2[$index][1] = $session2[$index][1] + 10;
171              $index = findIndex($key,$session3);
172              $session3[$index][1] = $session3[$index][1] + 10;
173          }
174          if ($c==4){
175              $index = findIndex($key,$session1);
176              $session1[$index][1] = $session1[$index][1] + 10;
177              $index = findIndex($key,$session2);
178              $session2[$index][1] = $session2[$index][1] + 10;
179              $index = findIndex($key,$session3);
180              $session3[$index][1] = $session3[$index][1] + 10;
181              $index = findIndex($key,$session4);
182              $session4[$index][1] = $session4[$index][1] + 10;
183          }
184      }
185  }

```

The code above is essentially the capacities subroutine with minor adjustments to suit the php language, for example when I formatted the fixed subject dictionary into an array, as the number of signups for the subjects needed to be accessed when finding out which subject to remove from the fixedsubject list if its length was bigger than 4. Otherwise, I would have had to use a multidimensional array, which is more complex than necessary. With the key of the array as the subject and the number of signups as the value, I was able to add and remove elements based on their value much easier than I would have been able to in a two dimensional array. I formatted the associative array back into a disassociative array as this is the form required for programming down the line, and only the subject is necessary.

```

186 //sets up the rejected list
187 $rejectlist = array();
188 //First pass procedure below, as well as the definition on studentplaces
189 $studentplaces = array();
190 //initialises the lists that store the respective subjects and sessions for each student
191 //this is stored in the studentplaces dictionary
192 for ($i = 0; $i < sizeof($Studentids); $i++){
193     $studentplaces[$Studentids[$i]] = array(array(),array(),array());
194 }
195 $choices = array();
196 //for testing reasons, i am using another loop.
197 //runs through list of all students
198 for ($i = 0; $i < sizeof($Studentids); $i++){
199     //puts the choices of the girls into a list to be operated on later down in the code
200     $girl = $Studentids[$i];
201     $query = "SELECT Choice1, Choice2, Choice3, Choice4 FROM studentchoices WHERE StudentID = $girl;";
202     $result = mysqli_query($con, $query);
203     $choices1 = mysqli_fetch_array($result);
204     $choices[0] = $choices1[0];
205     $choices[1] = $choices1[1];
206     $choices[2] = $choices1[2];
207     $choices[3] = $choices1[3];
208     //runs through the list of their choices
209     for ($j = 0; $j < sizeof($choices); $j++){
210         //runs through the list of fixed subjects
211         for ($k = 0; $k < sizeof($fixedSub); $k++){
212             //asks if the subject being compared is one in the list of fixed subjects
213             if ($choices[$j] == $fixedSub[$k]){
214                 //finds out if subject is in session1
215                 if (findIndex($choices[$j], $session1) != -1){
216                     //finds index of subject in session1
217                     $index = findIndex($choices[$j], $session1);
218                     //add the student being handled to the subject array in session 1
219                     array_push($session1[$index], $Studentids[$i]);
220                     //add 1 to the number in the session array that counts the student in that subject
221                     $session1[$index][2] = $session1[$index][2] +1;
222                     //adds the subject to the girls key in student places if the spot is not empty
223                     for ($x = 0; $x < 4; $x++){
224                         if (empty($studentplaces[$Studentids[$i]][$x])){
225                             array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
226
227                             //the addition of the 1, is to signify that the subject is in session 1
228                             array_push($studentplaces[$Studentids[$i]][$x], 1);
229                             break;
230                         }
231                     }
232                     //session2 now
233                     //this code is a replica of the code above, only in relation to session2 instead of 1
234                     elseif (findIndex($choices[$j], $session2) != -1){
235                         $index = findIndex($choices[$j], $session2);
236                         array_push($session2[$index], $Studentids[$i]);
237                         $session2[$index][2] = $session2[$index][2] +1;
238                         for ($x = 0; $x < 4; $x++){
239                             if (empty($studentplaces[$Studentids[$i]][$x])){
240                                 array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
241                                 array_push($studentplaces[$Studentids[$i]][$x], 2);
242                                 break;
243                             }
244                         }
245                     }
246                     //session3 now
247                     //this code is a replica of the code above, only in relation to session3 instead of 1
248                     elseif (findIndex($choices[$j], $session3) != -1){
249                         $index = findIndex($choices[$j], $session3);
250                         array_push($session3[$index], $Studentids[$i]);
251                         $session3[$index][2] = $session3[$index][2] +1;
252                         for ($x = 0; $x < 4; $x++){
253                             if (empty($studentplaces[$Studentids[$i]][$x])){
254                                 array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
255                                 array_push($studentplaces[$Studentids[$i]][$x], 3);
256                                 break;
257                             }
258                         }
259                     }
260                     //session4 now
261                     //this code is a replica of the code above, only in relation to session4 instead of 1
262                     elseif (findIndex($choices[$j], $session4) != -1){
263                         $index = findIndex($choices[$j], $session4);
264                         array_push($session4[$index], $Studentids[$i]);
265                         $session4[$index][2] = $session4[$index][2] +1;

```

```

266     for ($x = 0; $x < 4; $x++) {
267         if (empty($studentplaces[$Studentids[$i]][$x])){
268             array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
269             array_push($studentplaces[$Studentids[$i]][$x], 4);
270             break;
271         }
272     }
273 }
274 }
275 }
276 }
277 }

```

The code above is the first pass specified in my design. No specific modifications were made in the design of this section. The repeating code defining what happens in each session was not outlined in my design, but it has been presented 4 times in the code.

```

278 //ACTUAL PASS
279 //the code in this section is pretty much exactly the same as the code above, with some extra functionality which
280 //I will comment to highlight
281 for ($i = 0; $i < sizeof($Studentids); $i++){
282     $girl = $Studentids[$i];
283     $query = "SELECT Choice1, Choice2, Choice3, Choice4 FROM studentchoices WHERE StudentID = $girl;";
284     $result = mysqli_query($con, $query);
285     $choices1 = mysqli_fetch_array($result);
286     $choices[0] = $choices1[0];
287     $choices[1] = $choices1[1];
288     $choices[2] = $choices1[2];
289     $choices[3] = $choices1[3];
290     //print_r($choices);
291     for ($j = 0; $j < sizeof($choices); $j++){
292         //no need to check if the specific subject is in the fixed subject list anymore
293         if (findIndex($choices[$j], $session1) != -1){
294             $index = findIndex($choices[$j], $session1);
295             //checks to see if the maximum class limit has been reached, and makes sure the girl has not already
296             //been added to the specific section.
297             if (($session1[$index][2] < $session1[$index][1]) and (in2D($Studentids[$i], $session1) == False)
298                 and (in2D($choices[$j], $studentplaces[$Studentids[$i]]) == False)){
299                 array_push($session1[$index], $Studentids[$i]);
300                 $session1[$index][2] = $session1[$index][2] +1;
301                 for ($x = 0; $x < 4; $x++){
302                     if (empty($studentplaces[$Studentids[$i]][$x])){
303                         array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
304                         array_push($studentplaces[$Studentids[$i]][$x], 1);
305                         break;
306                     }
307                 }
308             }
309         }
310         //session2 now
311         //same as code above
312         if (findIndex($choices[$j], $session2) != -1){
313             $index = findIndex($choices[$j], $session2);
314             //checks to see if the maximum class limit has been reached, and makes sure the girl has not already
315             //been added to the specific section., it also makes sure this specific subject isn't already in their subject list
316             if (($session2[$index][2] < $session2[$index][1]) and (in2D($Studentids[$i], $session2) == False)
317                 and (in2D($choices[$j], $studentplaces[$Studentids[$i]]) == False)){
318                 array_push($session2[$index], $Studentids[$i]);
319                 $session2[$index][2] = $session2[$index][2] +1;
320                 for ($x = 0; $x < 4; $x++){
321                     if (empty($studentplaces[$Studentids[$i]][$x])){
322                         array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
323                         array_push($studentplaces[$Studentids[$i]][$x], 2);
324                         break;
325                     }
326                 }
327             }
328             //session3 now
329             //same as code above
330             if (findIndex($choices[$j], $session3) != -1){
331                 $index = findIndex($choices[$j], $session3);
332                 //checks to see if the maximum class limit has been reached, and makes sure the girl has not already
333                 //been added to the specific section. it also makes sure this specific subject isn't already in their subject list
334                 if (($session3[$index][2] < $session3[$index][1]) and (in2D($Studentids[$i], $session3) == False)
335                     and (in2D($choices[$j], $studentplaces[$Studentids[$i]]) == False)){
336                     array_push($session3[$index], $Studentids[$i]);
337                     $session3[$index][2] = $session3[$index][2] +1;
338                     for ($x = 0; $x < 4; $x++){
339                         if (empty($studentplaces[$Studentids[$i]][$x])){
340                             array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
341                             array_push($studentplaces[$Studentids[$i]][$x], 3);
342                             break;
343                         }
344                     }
345                 }
346             }

```

```

347         //session4 now
348         //same as code above
349         if (findIndex($choices[$j], $session4) != -1){
350             $index = findIndex($choices[$j], $session4);
351             //checks to see if the maximum class limit has been reached, and makes sure the girl has not already
352             //been added to the specific section. it also makes sure this specific subject isn't already in their subject list
353             if (($session4[$index][2] < $session4[$index][1]) and (in2D($Studentids[$i], $session4) == False)
354                 and (in2D($choices[$j], $studentplaces[$Studentids[$i]]) == False)){
355                 array_push($session4[$index], $Studentids[$i]);
356                 $session4[$index][2] = $session4[$index][2] +1;
357                 for ($x = 0; $x < 4; $x++){
358                     if (empty($studentplaces[$Studentids[$i]][$x])){
359                         array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
360                         array_push($studentplaces[$Studentids[$i]][$x], 4);
361                         break;
362                     }
363                 }
364             }
365         }
366     }
367     //if any of the girls have empty sessions, they will be added to the reject list, as not all their spaces are filled
368     if (empty($studentplaces[$Studentids[$i]][0]) or empty($studentplaces[$Studentids[$i]][1])
369         or empty($studentplaces[$Studentids[$i]][2]) or empty($studentplaces[$Studentids[$i]][3])){
370         array_push($rejectlist, $Studentids[$i]);
371     }
372 }

```

The code above is the actual pass specified in the design. Not many modifications were made here either, but two additional clauses were added to all the if statements that basically make sure of two things. The 1st thing is that if the girl has already been added to that session in a prior subject, to not overwrite this session with the new subject, but instead check in the other sessions. The 2nd clause makes sure that no subject is present more than once in a student's session. These new clauses make use of the additional function specified at the beginning. I did not design these two clauses because I did not foresee these issues, but I soon realised in my white box testing of this section.

```

373     //same exact code as the code above, only it takes into account, different Choices, which I will specify.
374     for ($i = 0; $i < sizeof($Studentids); $i++){
375         $girl = $Studentids[$i];
376         //select choice5 and choice6, instead of choice1,2,3,4, as these are there back up questions
377         $query = "SELECT Choice5, Choice6 FROM studentchoices WHERE StudentID = '$girl';";
378         $result = mysqli_query($con, $query);
379         $choices1 = mysqli_fetch_array($result);
380         $choices[0] = $choices1[0];
381         $choices[1] = $choices1[1];
382         for ($j = 0; $j < sizeof($choices); $j++){
383             if (findIndex($choices[$j], $session1) != -1){
384                 $index = findIndex($choices[$j], $session1);
385                 if (($session1[$index][2] < $session1[$index][1]) and (in2D($Studentids[$i], $session1) == False)){
386                     array_push($session1[$index], $Studentids[$i]);
387                     $session1[$index][2] = $session1[$index][2] +1;
388                     for ($x = 0; $x < 4; $x++){
389                         if (empty($studentplaces[$Studentids[$i]][$x])){
390                             array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
391                             array_push($studentplaces[$Studentids[$i]][$x], 1);
392                             break;
393                         }
394                     }
395                 }
396             }
397             //session2 now
398             elseif (findIndex($choices[$j], $session2) != -1){
399                 $index = findIndex($choices[$j], $session2);
400                 if (($session2[$index][2] < $session2[$index][1]) and (in2D($Studentids[$i], $session2) == False)){
401                     array_push($session2[$index], $Studentids[$i]);
402                     $session2[$index][2] = $session2[$index][2] +1;
403                     for ($x = 0; $x < 4; $x++){
404                         if (empty($studentplaces[$Studentids[$i]][$x])){
405                             array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
406                             array_push($studentplaces[$Studentids[$i]][$x], 2);
407                             break;
408                         }
409                     }
410                 }
411             }

```

```

412         //session3 now
413     elseif (findIndex($choices[$j], $session3) != -1){
414         $index = findIndex($choices[$j], $session3);
415         if (($session3[$index][2] < $session3[$index][1]) and (in2D($Studentids[$i], $session3) == False)){
416             array_push($session3[$index], $Studentids[$i]);
417             $session3[$index][2] = $session3[$index][2] + 1;
418             for ($x = 0; $x < 4; $x++){
419                 if (empty($studentplaces[$Studentids[$i]][$x])){
420                     array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
421                     array_push($studentplaces[$Studentids[$i]][$x], 3);
422                     break;
423                 }
424             }
425         }
426     }
427     //session4 now
428     elseif (findIndex($choices[$j], $session4) != -1){
429         $index = findIndex($choices[$j], $session4);
430         if (($session4[$index][2] < $session4[$index][1]) and (in2D($Studentids[$i], $session4) == False)){
431             array_push($session4[$index], $Studentids[$i]);
432             $session4[$index][2] = $session4[$index][2] + 1;
433             for ($x = 0; $x < 4; $x++){
434                 if (empty($studentplaces[$Studentids[$i]][$x])){
435                     array_push($studentplaces[$Studentids[$i]][$x], $choices[$j]);
436                     array_push($studentplaces[$Studentids[$i]][$x], 4);
437                     break;
438                 }
439             }
440         }
441     }
442 }
443

```

This code is the reject pass subroutine. The only changes that have been made to this that don't align with the design have been specified above for the actual pass.

```

445 //writes the 2d list into a csv file
446 $session_1 = "session1.csv";
447 $outFile = fopen($session_1, "w") or die("Unable to open file!");
448 //as it is a 2d array, the list is written by one inner list at a time.
449 foreach($session1 as $value){
450     fputcsv($outFile, $value);
451 }
452 fclose($outFile);
453
454 //writes the 2d list into a csv file
455 $session_2 = "session2.csv";
456 $outFile = fopen($session_2, "w") or die("Unable to open file!");
457 //as it is a 2d array, the list is written by one inner list at a time.
458 foreach($session2 as $value){
459     fputcsv($outFile, $value);
460 }
461 fclose($outFile);
462
463 //writes the 2d list into a csv file
464 $session_3 = "session3.csv";
465 $outFile = fopen($session_3, "w") or die("Unable to open file!");
466 //as it is a 2d array, the list is written by one inner list at a time.
467 foreach($session3 as $value){
468     fputcsv($outFile, $value);
469 }
470 fclose($outFile);
471
472 //writes the 2d list into a csv file
473 $session_4 = "session4.csv";
474 $outFile = fopen($session_4, "w") or die("Unable to open file!");
475 //as it is a 2d array, the list is written by one inner list at a time.
476 foreach($session4 as $value){
477     fputcsv($outFile, $value);
478 }
479 fclose($outFile);

```

```

481 //initialises final array, that will be able to download
482 $c = 0;
483 $final = array();
484 //runs through the dictionary of students and their choices
485 foreach ($studentplaces as $key => $value) {
486     //adds their name and 4 empty spots for their subjects
487     array_push($final, array($key));
488     array_push($final[$c], " ", " ", " ", " ");
489     for ($i = 0; $i < 4; $i++) {
490         //if their subject space isn't empty, then it adds the subject to their specified session space
491         if (!empty($value[$i][0]) == False) {
492             $final[$c][$value[$i][1]] = $value[$i][0];
493         }
494     }
495     $c = $c +1;
496 }
497
498 //writes the 2d list into a csv file
499 $finalfile = "final.csv";
500 $outFile = fopen($finalfile, "w") or die("Unable to open file!");
501 //as it is a 2d array, the list is written by one inner list at a time.
502 foreach($final as $value){
503     fputcsv($outFile, $value);
504 }
505 fclose($outFile);
506
507 //close sql database connection
508 mysqli_close($con);
509 }
?>

```

The code above is specified under session csv files. In my coding I did not stray from the design at all, except from an additional clause in one of my if statements that ignores all empty values and does not add them to the final csv file. So extra validation was added.

Test Plan:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Testign evidence	Pass or Fail
Valid Student ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	Page refreshes, and values have been changed on screen and in php database	N/A	PASSED
Invalid (non existent) Student ID, Valid Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated.	N/A	PASSED
Valid Student ID, Invalid (non existent) Choice, Valid Subject	Staff can edit the above table	EDIT TABLE	SQL command will be incorrect, so table is not updated. Also drop down list makes this impossible.	N/A	PASSED
Valid Student ID, Valid Choice, Invalid (non existent) Subject	Staff can edit the above table	EDIT TABLE	Drop down list of subjects input only allows correct subject submission.	N/A	PASSED

Visible button	Choice at the bottom of the page to sort girls	ARRANGE GIRLS	Screenshot visible and functioning button	N/A	PASSED
After validation in prior areas of code, there will be no erroneous data in the program. Therefore no need to test under those circumstances. Will do white box testing here along the way to make sure the programs subroutines work. Then test with the data from last year to simulate a real issue.	Arrange the girls into correct sessions	ARRANGE GIRLS	Sorts girls to Mrs Jackson's liking. – all girls should have 4 options, no girl should have any option twice, the success of this aspect is up to Mrs Jackson	1,2,3,4&5	UP TO STAKE HOLDER – her comment

*All other aspects of this page have been tested in the second staff interface, as they have the exact same functionality

Testing Evidence using last year's data:

1	A	B	C	D	E
1	84626705547	Maths Alevel	Latin	Chemistry	Biology
2	85447868892	Maths Alevel	Maths IB	Physics A-level	Maths IB Studies
3	85807705547	Biology	English IB	Chemistry	Theory of Knowledge
4	90357705547	Biology	Chemistry	History	Economics
5	91504705547	Chemistry	Maths Alevel	Maths IB	Biology
6	91539643505	Economics	French	Maths Alevel	Spanish
7	91801705547	Biology	Chemistry	Economics	History
8	92647705547	Theory of Knowledge	Economics	History	English IB
9	9292753424	Economics	Politics	History	Maths Alevel
10	92954705547	Maths Alevel	Physics A-level	Computing	Chemistry
11	93245705547	Chemistry	Physics A-level	French	Italian
12	94003533424	Economics	Biology	Chemistry	English IB
13	94606579518	Chemistry	Maths Alevel	Economics	Physics A-level
14	94854533424	Theory of Knowledge	English IB	Computing	History
15	95358533424	Biology	French	Chemistry	Maths Alevel
16	95424410122	Theory of Knowledge	English IB	Maths IB	Physics A-level
17	1.01131E+11	Politics	Economics	Philosophy IB	RS A Level
18	1.01259E+11	Chemistry	Biology	Spanish	Latin
19	1.01322E+11	Biology	History	Chemistry	Physics A-level
20	1.01731E+11	Chemistry	Maths Alevel	Biology	Physics A-level
21	1.01901E+11	Politics	History	Economics	Physics A-level
22	1.02023E+11	Chemistry	History	Theory of Knowledge	Politics
23	1.02221E+11	Chemistry	Biology	Economics	Maths Alevel
24	1.0266E+11	Chemistry	Spanish	Maths IB	English IB
25	1.02856E+11	Geography	English A level	Politics	RS A Level
26	1.02907E+11	English IB	History	Theory of Knowledge	Philosophy IB
27	1.0382F+11	Physics A-level	RS A level	Maths Alevel	Spanish

None of the students have been assigned the same subject twice! – this is the final csv file.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1 Latin	10	4	8.46E+10	1.01E+11	1.1E+11	1.14E+11								
2 Maths Alev	10	7	8.54E+10	9.3E+10	1.05E+11	1.05E+11	1.21E+11	1.44E+11	1.44E+11					
3 Chemistry	20	20	9.15E+10	9.32E+10	9.46E+10	1.02E+11	1.02E+11	1.02E+11	1.03E+11	1.04E+11	1.05E+11	1.13E+11	1.15E+11	1.21E+11
4 Biology	20	20	8.58E+10	9.04E+10	9.18E+10	9.54E+10	1.01E+11	1.04E+11	1.12E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11	1.21E+11
5 Maths IB	10	1	1.73E+11											
6 Physics A-I	10	3	1.04E+11	1.2E+11	1.44E+11									
7 English IB	10	2	1.03E+11	1.35E+11										
8 Theory of	10	5	9.26E+10	9.49E+10	9.54E+10	1.14E+11	1.21E+11							
9 History	10	2	1.11E+11	1.63E+11										
10 Economics	20	20	9.15E+10	9.29E+10	9.4E+10	1.04E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11	1.22E+11	1.23E+11	1.32E+11	1.44E+11
11 French	10	1	1.45E+11											
12 Spanish	10	0												
13 Politics	20	17	1.01E+11	1.02E+11	1.03E+11	1.05E+11	1.06E+11	1.11E+11	1.14E+11	1.15E+11	1.15E+11	1.16E+11	1.31E+11	1.32E+11
14 Computing	10	2	1.14E+11	1.15E+11										
15 Philosophy	10	4	1.13E+11	1.13E+11	1.61E+11	1.61E+11								
16 RSA Level	10	3	1.15E+11	1.52E+11	1.62E+11									
17 Geography	10	0												
18 English A le	10	5	1.11E+11	1.12E+11	1.15E+11	1.16E+11	1.33E+11							
19 Chinese	10	1	1.45E+11											
20 Art	10	5	1.04E+11	1.11E+11	1.63E+11	1.65E+11	2.23E+09							
21 History of	10	1	1.61E+11											

Session1 – There are no students that have been placed

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	C
1 Maths IB S	10	3	8.54E+10	1.21E+11	1.62E+11											
2 Maths Alev	10	7	8.46E+10	9.15E+10	9.46E+10	1.02E+11	1.1E+11	1.32E+11	1.55E+11							
3 Chemistry	20	12	9.04E+10	9.18E+10	1.01E+11	1.05E+11	1.11E+11	1.12E+11	1.13E+11	1.2E+11	1.21E+11	1.41E+11	1.41E+11	1.52E+11		
4 Biology	10	10	9.4E+10	1.02E+11	1.14E+11	1.15E+11	1.15E+11	1.21E+11	1.44E+11	1.6E+11	1.63E+11	1.7E+11				
5 Maths IB	10	3	1.04E+11	1.21E+11	1.35E+11											
6 Physics A-I	10	9	9.3E+10	9.32E+10	1.04E+11	1.05E+11	1.13E+11	1.13E+11	1.7E+11	1.91E+09	2.24E+09					
7 English IB	10	5	8.58E+10	9.49E+10	9.54E+10	1.05E+11	1.45E+11									
8 Theory of	10	2	1.05E+11	1.15E+11												
9 History	10	10	1.01E+11	1.02E+11	1.02E+11	1.03E+11	1.15E+11	1.32E+11	1.44E+11	1.53E+11	1.55E+11	1.62E+11				
10 Economics	20	20	9.26E+10	1.01E+11	1.11E+11	1.13E+11	1.13E+11	1.14E+11	1.15E+11	1.16E+11	1.16E+11	1.21E+11	1.31E+11	1.44E+11	1.44E+11	
11 French	10	4	9.15E+10	9.54E+10	1.14E+11	1.62E+11										
12 Spanish	10	3	1.03E+11	1.2E+11	1.35E+11											
13 Politics	10	10	9.29E+10	1.11E+11	1.12E+11	1.14E+11	1.2E+11	1.23E+11	1.44E+11	1.52E+11	1.52E+11	1.61E+11				
14 Computing	10	0														
15 Philosophy	10	4	1.06E+11	1.15E+11	1.15E+11	1.33E+11										
16 RSA Level	10	5	1.04E+11	1.33E+11	1.45E+11	1.46E+11	1.61E+11									
17 Geography	10	5	1.03E+11	1.21E+11	1.44E+11	1.55E+11	1.65E+11									
18 English A le	10	2	1.15E+11	1.62E+11												
19 Chinese	10	3	1.04E+11	1.11E+11	1.61E+11											
20 Art	10	2	1.3E+11	1.73E+11												
21 History of	10	4	1.04E+11	1.22E+11	1.51E+11	2.23E+09										

Session2 – There are no students that have been placed.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1 Italian	10	5	9.32E+10	1.05E+11	1.21E+11	1.45E+11	1.46E+11								
2 Maths Alev	10	8	9.15E+10	1.04E+11	1.04E+11	1.11E+11	1.11E+11	1.13E+11	1.15E+11	1.62E+11					
3 Chemistry	20	13	8.46E+10	8.58E+10	9.4E+10	9.54E+10	1.01E+11	1.04E+11	1.04E+11	1.1E+11	1.14E+11	1.22E+11	1.23E+11	1.35E+11	1.61E+11
4 Biology	10	9	1.01E+11	1.02E+11	1.04E+11	1.05E+11	1.15E+11	1.21E+11	1.45E+11	1.62E+11	1.63E+11				
5 Maths IB	10	4	8.54E+10	9.15E+10	9.54E+10	1.03E+11									
6 Physics A-I	10	8	1.14E+11	1.3E+11	1.32E+11	1.35E+11	1.52E+11	1.52E+11	1.55E+11	1.63E+11					
7 English IB	10	1	1.2E+11												
8 Theory of	10	5	1.02E+11	1.03E+11	1.21E+11	1.33E+11	1.44E+11								
9 History	10	7	9.04E+10	9.26E+10	9.29E+10	1.11E+11	1.33E+11	1.52E+11	2.23E+09						
10 Economics	20	19	9.18E+10	9.46E+10	1.02E+11	1.02E+11	1.05E+11	1.05E+11	1.06E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11		
11 French	10	3	1.13E+11	1.15E+11	1.44E+11										
12 Spanish	10	1	1.16E+11												
13 Politics	10	10	1.13E+11	1.13E+11	1.21E+11	1.44E+11	1.53E+11	1.55E+11	1.61E+11	1.62E+11	1.62E+11	1.7E+11			
14 Computing	10	2	9.3E+10	9.49E+10											
15 Philosophy	10	7	1.01E+11	1.14E+11	1.15E+11	1.32E+11	1.52E+11	1.6E+11	1.61E+11						
16 RSA Level	10	4	1.15E+11	1.44E+11	1.44E+11	1.61E+11									
17 Geography	10	6	1.14E+11	1.15E+11	1.41E+11	1.61E+11	1.7E+11	1.73E+11							
18 English A le	10	7	1.03E+11	1.12E+11	1.31E+11	1.44E+11	1.45E+11	1.45E+11	1.65E+11						
19 Chinese	10	0													
20 Art	10	1	1.41E+11												
21 History of	10	3	1.11E+11	1.12E+11	1.16E+11										

Session3 – There are no students that have been placed.

5	A	B	C	D	E	F	G	H	I	J	K	L	M
1	German	10	3	1.45E+11	1.45E+11	1.61E+11							
2	Maths Alev	10	10	9.29E+10	9.54E+10	1.02E+11	1.04E+11	1.05E+11	1.06E+11	1.14E+11	1.2E+11	1.44E+11	1.45E+11
3	Chemistry	10	10	9.3E+10	1.04E+11	1.11E+11	1.13E+11	1.13E+11	1.16E+11	1.21E+11	1.21E+11	1.3E+11	1.35E+11
4	Biology	10	9	8.46E+10	9.15E+10	1.55E+11	1.55E+11	1.61E+11	1.61E+11	1.62E+11	1.7E+11	2.24E+09	
5	Maths IB	10	5	1.04E+11	1.63E+11	1.22E+11	1.23E+11	1.44E+11					
6	Physics A-I	10	10	8.54E+10	9.46E+10	9.54E+10	1.01E+11	1.02E+11	1.02E+11	1.1E+11	1.41E+11	1.55E+11	1.62E+11
7	English IB	10	6	9.26E+10	9.4E+10	1.03E+11	1.21E+11	1.21E+11	1.52E+11				
8	Theory of	10	4	8.58E+10	1.15E+11	1.21E+11	1.61E+11						
9	History	10	10	9.18E+10	9.49E+10	1.12E+11	1.13E+11	1.14E+11	1.14E+11	1.15E+11	1.15E+11	1.41E+11	1.52E+11
10	Economics	10	10	9.04E+10	1.11E+11	1.14E+11	1.15E+11	1.15E+11	1.15E+11	1.65E+11	2.23E+09	1.44E+11	1.62E+11
11	French	10	2	9.32E+10	1.32E+11								
12	Spanish	10	6	9.15E+10	1.01E+11	1.04E+11	1.04E+11	1.15E+11	1.6E+11				
13	Politics	10	10	1.02E+11	1.05E+11	1.11E+11	1.13E+11	1.15E+11	1.2E+11	1.7E+11	1.73E+11	1.91E+09	1.44E+11
14	Computing	10	2	1.32E+11	1.44E+11								
15	Philosophy	10	6	1.03E+11	1.05E+11	1.16E+11	1.2E+11	1.35E+11	1.52E+11				
16	RS A Level	10	5	1.01E+11	1.03E+11	1.11E+11	1.31E+11	1.44E+11					
17	Geography	10	7	1.12E+11	1.13E+11	1.21E+11	1.33E+11	1.52E+11	1.53E+11	1.44E+11			
18	English A le	10	3	1.45E+11	1.46E+11	1.61E+11							
19	Chinese	10	3	1.05E+11	1.33E+11	1.51E+11							
20	Art	10	1	1.62E+11									
21	History of	10	0										
22													

Session4 – There are no students that have been placed.

Stakeholder Meeting:

Feedback from Mrs AE Jackson: *email thread*

Okoroafor, Olaedo
Mon 23/03/2020 10:25
Mrs AE Jackson (Saturday Administrator) ▾

session2.csv 2 KB session3.csv 2 KB session4.csv 2 KB

session1.csv 2 KB final.csv 7 KB

5 attachments (16 KB) Download all Save all to OneDrive - Cheltenham Ladies' College

Dear Mrs Jackson,
I have tested my software with the data that you gave me from last year and the results are shown below. Mr Wain gave me the usernames and student ids for the current 5s year group. I used this data alongside the data you gave me, as they had no specific names. This is the result and the files you will be able to download once the overall program has run. Please comment on the results and whether you are satisfied with these results. In the session csv files, there are class limits and capacities. If the class limit is 20, it means, this class will be split into 2. In these files, all the subjects in these classes are given the ids of the students taking them.
Best wishes,
Ola



Mrs AE Jackson (Saturday Administrator)
Mon 23/03/2020 12:39
Okoroafor, Olaedo ✎



Thank you Ola

I've looked at the data you sent and it looks as if it fits the requirements I have for sorting this information except for one aspect. For a subject with overall low numbers eg History of Art, Art, Computing I can see that they appear across several of the columns in the final document whereas ideally where the overall number is low they really need to appear in only one or at most 2 columns to avoid groups of only 1 or 2 students. Is that a possible amendment

Mrs Jackson

Ms Ann E Jackson | Saturday Administrator - Cheltenham Ladies' College | Tel: +44 (0)1242 520691 extn 5021 | Email: jacksonae@cheltladiescollege.org

...



Okoroafor, Olaedo
Mon 23/03/2020 13:17
Mrs AE Jackson (Saturday Administrator) ✎



Dear Mrs Jackson,

I completely understand what you mean. The algorithm in the program was designed to give the subjects with the least signups one fixed spot in the 4 subjects (these subjects are the first row in the session files), where if a student had picked one of these subjects they would immediately be sorted into these sessions. I then gave every other subject a slot for each of the four sessions. This is to avoid me putting two fixed subjects in one session. The issues with this would have been that, a student could have picked both these subjects and thus would only be able to participate in one of them if they are both fixed in the same session. I'll try my best to get around this issue. Thank you for the comments!

Best wishes,

Ola

...

Notes and Changes:

- The new requirement she has suggested will take a significant amount of programming, that has not been planned or designed for as I was previously told the classes did not have a minimum size. I will probably not be able to complete this section during the project time constraints, but I will be working on it after to complete this change for Mrs Jackson
- She has suggested no further changes to be made to this section, as it meets the requirements.

STAGE 2 COMPLETED

STAGE 3

- Implement changes and suggestions from previous stage.
- Add all the validation absent from the stages before.
- Meeting with stakeholder, Mrs Jackson. Ask for feedback and changes/ suggestions on completed sections.

Implement changes and suggestions from previous stage:

Nothing to Implement.

Add all the validation absent from the stages before:

All the validation I still need to add is on the Student Interface, as this is the only Interface that needs validation. On the webpages that the students have access to, I will need to make sure that they can not enter more than one of the same subject, and that a girl can not have more than one set of subjects logged under her name.

I have made the following changes to my handlestudents.php file to include the validation specified by my stakeholder.

```

31 //creates array for choices, to see if there are any duplicates
32 $unique = array();
33 //appends the choices above to the list
34 array_push($unique, $choice1, $choice2, $choice3, $choice4, $choice5, $choice6);
35 //this if statement asks if the student is already in the completed list
36 //meaning she has already completed the survey
37 if (in_array($_POST['userN'], $completedli) ){
38     //this notifies the student that the username theyve picked is already in use
39     //makes sure form isnt submitted twice
40     echo "You have not successfully submitted your survey! This username has already been used.";
41 }
42 //this if statement removes all the duplicates from the unique array and compares
43 //the length to the length of the original array
44 //if it is smaller it means that the original array had duplicates
45 elseif (count(array_unique($unique))<count($unique) ){
46     //notifies the student of their issue
47     echo "You can only choose a subject once, please try again with valid subject choices!";
48 }
49 Else{
50     array_push($completedli, $_POST['userN']);

```

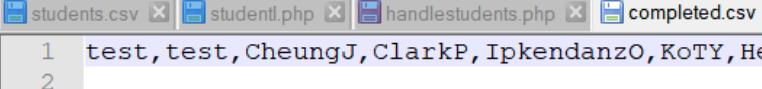
I have added 2 if clauses. These clauses make sure that none of the code that adds the student and their choices to the studentchoices table runs. The clauses in the image above perform their purpose and are also explained more thoroughly in the code's comments. The else at the bottom contains the code that would run if all the checks were passed.

Test Plan:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Testing Evidence	Pass/Fail(P/F)
Be able to pick from list of names.	Options of girls names show up to pick from.	DROP DOWN NAME	List of names from student csv	N/A	Passed
Be able to pick from list of names.	Options of subjects show up to pick from	DROP DOWN SUBJECT	Should display list of subjects entered by Mrs Jackson	N/A	Passed
Click button submit with correct data		SUBMIT CHOICES	Submission into mysql database	N/A	Passed
Submit with empty subject choices	Validation of user input	SUBMIT CHOICES	Error message	N/A	Passed
Submit with empty name choices	Validation of user input	SUBMIT CHOICES	Error message	N/A	Passed

Submit with subjects that are the same	Validation of user input	SUBMIT CHOICES	Error message	1,2	P
Submit with name that has already been used.	Validation of user input	SUBMIT CHOICES	Error message	3,4,5	P

Testing Evidence:

1	<p>Select your subjects!</p> <p>Select your name</p> <p><input type="text" value="Stewart-RichardsonEm ▾"/></p> <p>Select your 1st choice subject</p> <p><input type="text" value="Maths Alevel ▾"/></p> <p>Select your 2nd choice subject</p> <p><input type="text" value="Latin ▾"/></p> <p>Select your 3rd choice subject</p> <p><input type="text" value="Chemistry ▾"/></p> <p>Select your 4th choice subject</p> <p><input type="text" value="Computing ▾"/></p> <p>Select your 5th choice subject</p> <p><input type="text" value="Economics ▾"/></p> <p>Select your 6th choice subject (none is an option)</p> <p><input type="text" value="Maths Alevel ▾"/> <input type="button" value="Submit"/> Maths A Level is repeated!</p>
2	<p>If you have a problem, please email: JacksonAE@cheltladiescollege.org</p> <p>You can only choose a subject once, please try again with valid subject choices! TRY AGAIN IF UNSUCCESSFUL</p>
3	<p>ClarkP is already in the completed file below:</p> <p> 1 test,test,CheungJ,ClarkP,IpkendanzO,KoTY,He 2</p>

4	<p>Select your subjects!</p> <p>Select your name <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="ClarkP"/></p> <p>Select your 1st choice subject <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="Maths Alevel"/></p> <p>Select your 2nd choice subject <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="Latin"/></p> <p>Select your 3rd choice subject <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="Chemistry"/></p> <p>Select your 4th choice subject <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="History"/></p> <p>Select your 5th choice subject <input style="width: 150px; height: 20px; border: 1px solid black; margin-bottom: 5px;" type="text" value="Spanish"/></p> <p>Select your 6th choice subject (none is an option) <input style="width: 150px; height: 20px; border: 1px solid blue; margin-bottom: 5px;" type="text" value="Theory of Knowledge"/> <input style="width: 100px; height: 20px; border: 1px solid black;" type="button" value="Submit"/></p>
5	<p>If you have a problem, please email: JacksonAE@cheltladiescollege.org</p> <p>You have not successfully submitted your survey! This username has already been used. TRY AGAIN IF UNSUCCESSFUL</p>

Stakeholder Meetings:

Feedback from Mrs AE Jackson:

- Validation is as it should be. Mrs Jackson is content.

Notes and Changes:

- None

STAGE 3 COMPLETED

STAGE 4

- Implement changes and suggestions from previous stage.
- Add all the css to make the webpage look presentable, and also include helping hand textboxes.
- Include the last interface where Mrs Jackson can download files (this page should also have CSS)
- Outline and include maintenance features.
- Ask for final stakeholder feedback.
- Implement changes and suggestions from above meeting.

Implement changes and suggestions from previous stage:

Nothing to Implement.

Add all the css to make the webpage look presentable, and also include helping hand textboxes:

For this section of development, the presentation of the student interface is the most important. Mrs Jackson's webpages will mostly be for admin so this is less important. I will work on making each of the webpages clear and spacious. I will also make use of the helping hand text boxes specified in my design Interfaces to guide the users through the process.

Starting with the page Mrs Jackson uses to enter the subjects (First Staff Interface):

Old Screen:

Fill in Subjects

When Submit button is clicked, you can not come back to this page!

<input type="text"/>	<input type="button" value="add subject"/>	<input type="text"/>	<input type="button" value="remove subject"/>	<input type="button" value="Submit"/>
----------------------	--	----------------------	---	---------------------------------------

Or you may upload a csv file with your predetermined subjects

Select csv file to upload: No file chosen

If you have already completed this section, please click the link below: [Next](#)

The screen above is quite cluttered and confusing. For this page, I will be spacing out the buttons and also clarifying the process to Mrs Jackson.

New screen:

The screenshot shows a web browser window with the URL 'localhost/PROJECT/login/subjects.php'. The page title is 'Fill in Subjects'. A blue header bar contains the title. Below it is a large white form area. At the top of the form, there is a message: 'You may choose to upload a csv file with your subjects or enter them manually below.' Below this message is a horizontal row of buttons: an input field, an 'add subject' button, another input field, a 'remove subject' button, and a 'Submit' button. Underneath this row, there is another message: 'Or you may upload a csv file with the subjects.' Below this message is a second horizontal row of buttons: 'Choose a file from your computer to upload. Then Click upload csv file.', an 'input' field, a 'Choose File' button, an 'No file chosen' label, and an 'Upload csv file' button. At the bottom of the form, there is a final message: 'Select csv file to upload: Choose File No file chosen Upload csv file'. The entire form is enclosed in a blue border.

This is what this page currently looks like after the addition of some style elements, which I will clarify below.

```

3  <style>
4   body {
5     background: lightblue;
6   }
7   .content {
8     max-width: 800px;
9     margin: auto;
10    background: white;
11    padding: 10px;
12  }
13 </style>
14 </head>
15 <body>
16 <div class = "content">
17 <h1> Fill in Subjects </h1>
18 <p style="padding: 0 7em 2em 0; border-width: 2px; border-color: blue;
19 border-style:solid; width:fit-content; width:-webkit-fit-content; width:-moz-fit-content;">
20 You may choose to upload a csv file with your subjects or enter them manually below.</p>
21 <p id = "test" ></p>
22 <div id='list'>
23 </div>

```

The html in the style tag elements, applies these conditions over the whole page. The body tag makes the body of the page blue, and the content class centres all the elements on the page and makes this background white. Through out the code I added many
 elements to break up the text and make it more spacious. The helping hand boxes as depicted are guiding the user through the process. As you can see they are bordered by blue lines, to draw attention to them and for this I used the css styling elements in the

tag depicted. The

then wraps around all the contents of the website, signifying all the styling elements under the content class will apply to them.

Student Interface Form Screen:

Old Screen:

Select your name

Select your 1st choice subject

Select your 2nd choice subject

Select your 3rd choice subject

Select your 4th choice subject

Select your 5th choice subject

Select your 6th choice subject (none is an option)

This page is also quite cluttered with limited direction.

New Screen:

The screenshot shows a web page titled "student.php". The main content area has a white background and a blue border. At the top, it says "Select your subjects!". Below that is a red-bordered box containing the text "You may not choose the same subject twice. You may not submit this form twice.". Underneath this box are several input fields: "Select your name" (dropdown menu "Choose one"), "Select your 1st choice subject" (dropdown menu "Choose a subject"), "Select your 2nd choice subject" (dropdown menu "Choose a subject"), "Select your 3rd choice subject" (dropdown menu "Choose a subject"), "Select your 4th choice subject" (dropdown menu "Choose a subject"), "Select your 5th choice subject" (dropdown menu "Choose a subject"), and "Select your 6th choice subject" (dropdown menu "Choose a subject"). At the bottom right of the input area are two buttons: "Choose a subject" and "Submit".

I did this using the css elements I spoke about before. However, I did change the font of the text on this page to verdana. The text in the helping hand box is also red, to draw attention to it, as it is very important. – This is one of the usability features.

```
38 | <p style= "font-family:verdana;color:red;padding: 0 7em 2em 0;border-width: 2px; border-color: blue;
39 |   border-style:solid; width:fit-content; width:-webkit-fit-content; width:-moz-fit-content;">
40 |   You may not choose the same subject twice. You may not submit this form twice.</p>
41 | <form method="POST" action='handlestudents.php'>
42 | <p style= "font-family:verdana;"> Select your name </p>
```

As you can see above, I added the color red and also the font family verdana.

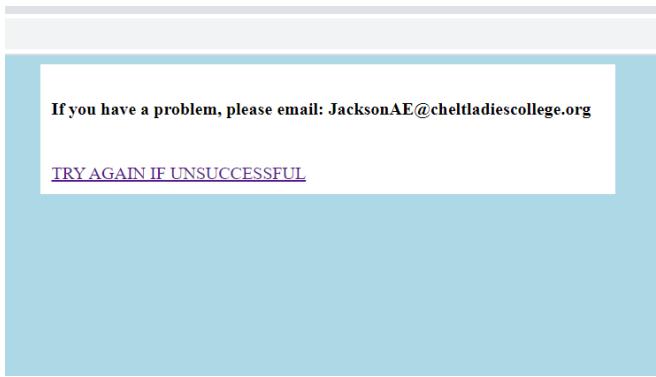
Student Interface Confirmation Screen:

Old Screen:



This page is fine, I added style elements for the sake of consistency.

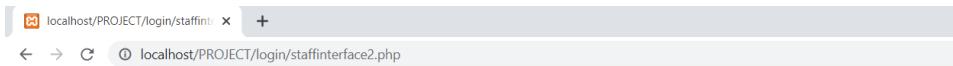
New Screen:



CSS used is the same as in previous stages.

Second Staff Interface:

Old Screen:



RESPONSES

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
85447868892	Maths Alevel	Maths IB	Physics A-level	Maths IB Studies	Theory of Knowledge	Computing
84626705547	Maths Alevel	Latin	Chemistry	Biology	Physics A-level	German
85807705547	Biology	English IB	Chemistry	Theory of Knowledge	History	Spanish
90357705547	Biology	Chemistry	History	Economics	Spanish	Physics A-level
91504705547	Chemistry	Maths Alevel	Maths IB	Biology	Physics A-level	Spanish
91539643505	Economics	French	Maths Alevel	Spanish	Biology	English A level
91801705547	Biology	Chemistry	Economics	History	Physics A-level	Maths Alevel
92647705547	Theory of Knowledge	Economics	History	English IB	Chinese	Chemistry
92927533424	Economics	Politics	History	Maths Alevel	Latin	RS A Level
92954705547	Maths Alevel	Physics A-level	Computing	Chemistry	Economics	Geography
93245705547	Chemistry	Physics A-level	French	Italian	Maths Alevel	Geography
94003533424	Economics	Biology	Chemistry	English IB	Maths IB	History
94606579518	Chemistry	Maths Alevel	Economics	Physics A-level	Biology	Politics
94854533424	Theory of Knowledge	English IB	Computing	History	Polities	Economics
95358533424	Biology	French	Chemistry	Maths Alevel	Spanish	English A level
95424410122	Theory of Knowledge	English IB	Maths IB	Physics A-level	History	Politics
101130705547	Politics	Economics	Philosophy IB	RS A Level	Theory of Knowledge	Geography
101258705547	Chemistry	Biology	Spanish	Latin	Maths IB	History
101321965465	Biology	History	Chemistry	Physics A-level	Geography	English A level
101730705547	Chemistry	Maths Alevel	Biology	Physics A-level	Economics	Spanish
101900533424	Politics	History	Economics	Physics A-level	Latin	Spanish
102023353588	Chemistry	History	Theory of Knowledge	Politics	Philosophy IB	Economics
102220705547	Chemistry	Biology	Economics	Maths Alevel	Physics A-level	Geography
102659533424	Chemistry	Spanish	Maths IB	English IB	History	Biology
102855705547	Politics	Geography	English A level	RS A Level	Spanish	Economics

155259705547	Economics	History	Politics	Physics A-level	Spanish	Biology
160210705547	Biology	Chemistry	Philosophy IB	Spanish	English IB	Economics
160803533424	Philosophy IB	Chinese	Chemistry	Biology	English IB	Maths Alevel
160900705547	Politics	Economics	Philosophy IB	Biology	Maths IB	Theory of Knowledge
160922705547	Economics	Politics	Geography	English A level	History	French
161018705547	Philosophy IB	Economics	RS A Level	Chemistry	Maths Alevel	Theory of Knowledge
161127705547	History of Art	Politics	RS A Level	History	German	Economics
161728705547	Chemistry	English A level	Maths Alevel	History	Physics A-level	English A level
161731705547	Economics	History	Politics	Biology	French	Chinese
161914705547	Biology	Chemistry	Maths Alevel	Maths IB Studies	Art	Economics
162324705547	RS A Level	French	Politics	Chemistry	Economics	Maths Alevel
162557705547	History	Economics	Biology	Maths IB	Philosophy IB	Physics A-level
162617533424	Biology	Art	Physics A-level	Chemistry	Maths Alevel	Economics
164522705547	Art	Geography	English A level	Economics	History	Biology
170405705547	Economics	Biology	Geography	Politics	History	Chemistry
170447705547	Economics	Politics	Biology	Physics A-level	Chinese	Maths Alevel
173410705547	Maths IB	Politics	Art	Geography	Physics A-level	Maths IB Studies
1906487607	Chemistry	Politics	Physics A-level	Economics	History	Biology
2230808502	Art	History	Economics	History of Art	French	RS A Level
2235712028	Economics	Biology	Chemistry	Physics A-level	Chemistry	Philosophy IB
3455871538	Maths Alevel	Chemistry	Physics A-level	Biology	Economics	

AWAITING RESPONSES

- TSeZ
 - Brooks-HughesH
 - PalmerN
 - AshbyL
 - Stewart-RichardsonEm
- [Send Reminder](#)

EDIT THE GIRLS

[NEXT PAGE](#)

Elements, are not spread out at the bottom of the page.

New Screen:

RESPONSES						
Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
85447868892	Maths Alevel	Maths IB	Physics A-level	Maths IB Studies	Theory of Knowledge	Computing
84626705547	Maths Alevel	Latin	Chemistry	Biology	Physics A-level	German
85807705547	Biology	English IB	Chemistry	Theory of Knowledge	History	Spanish
90357705547	Biology	Chemistry	History	Economics	Spanish	Physics A-level
91504705547	Chemistry	Maths Alevel	Maths IB	Biology	Physics A-level	Spanish
91539643505	Economics	French	Maths Alevel	Spanish	Biology	English A level
91801705547	Biology	Chemistry	Economics	History	Physics A-level	Maths Alevel
92647705547	Theory of Knowledge	Economics	History	English IB	Chinese	Chemistry
92927533424	Economics	Politics	History	Maths Alevel	Latin	RS A Level
92954705547	Maths Alevel	Physics A-level	Computing	Chemistry	Economics	Geography
93245705547	Chemistry	Physics A-level	French	Italian	Maths Alevel	Geography
94003533424	Economics	Biology	Chemistry	English IB	Maths IB	History
94606579518	Chemistry	Maths Alevel	Economics	Physics A-level	Biology	Politics
94854533424	Theory of Knowledge	English IB	Computing	History	Politics	Economics
95358533424	Biology	French	Chemistry	Maths Alevel	Spanish	English A level
95424410122	Theory of Knowledge	English IB	Maths IB	Physics A-level	History	Politics
101130705547	Politics	Economics	Philosophy IB	RS A Level	Theory of Knowledge	Geography
101258705547	Chemistry	Biology	Spanish	Latin	Maths IB	History
101321965465	Biology	History	Chemistry	Physics A-level	Geography	English A level
101730705547	Chemistry	Maths Alevel	Biology	Physics A-level	Economics	Spanish
101900533424	Politics	History	Economics	Physics A-level	Latin	Spanish
102023353588	Chemistry	History	Theory of Knowledge	Politics	Philosophy IB	Economics
102220705547	Chemistry	Biology	Economics	Maths Alevel	Physics A-level	Geography

2230808502	Art	History	Economics	History of Art	French	RS A Level
2235712028	Economics	Biology	Chemistry	Physics A-level	Chemistry	Philosophy IB
3455871538	Maths Alevel	Chemistry	Physics A-level	Biology	Economics	

AWAITING RESPONSES

- TseZ
- Brooks-HughesH
- PalmerN
- AshbyL
- Stewart-RichardsonEm

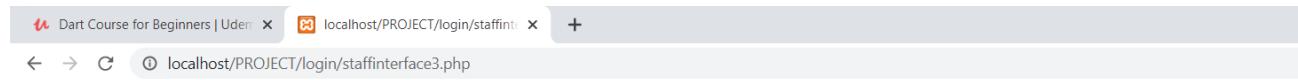
[Send Reminder](#)

[NEXT PAGE](#)

CSS used is the same as in previous stages.

Third Staff Interface:

Old Screen:



RESPONSES

RESPONSES RECEIVED

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
85447868892	Maths Alevel	Maths IB	Physics A-level	Maths IB Studies	Theory of Knowledge	Computing
84626705547	Maths Alevel	Latin	Chemistry	Biology	Physics A-level	German
85807705547	Biology	English IB	Chemistry	Theory of Knowledge	History	Spanish
90357705547	Biology	Chemistry	History	Economics	Spanish	Physics A-level
91504705547	Chemistry	Maths Alevel	Maths IB	Biology	Physics A-level	Spanish
91539643505	Economics	French	Maths Alevel	Spanish	Biology	English A level
91801705547	Biology	Chemistry	Economics	History	Physics A-level	Maths Alevel
92647705547	Theory of Knowledge	Economics	History	English IB	Chinese	Chemistry
92927533424	Economics	Politics	History	Maths Alevel	Latin	RS A Level
92954705547	Maths Alevel	Physics A-level	Computing	Chemistry	Economics	Geography
93245705547	Chemistry	Physics A-level	French	Italian	Maths Alevel	Geography
94003533424	Economics	Biology	Chemistry	English IB	Maths IB	History
94606579518	Chemistry	Maths Alevel	Economics	Physics A-level	Biology	Politics
94854533424	Theory of Knowledge	English IB	Computing	History	Politics	Economics
95358533424	Biology	French	Chemistry	Maths Alevel	Spanish	English A level
95424410122	Theory of Knowledge	English IB	Maths IB	Physics A-level	History	Politics
101130705547	Politics	Economics	Philosophy IB	RS A Level	Theory of Knowledge	Geography
101258705547	Chemistry	Biology	Spanish	Latin	Maths IB	History
101321965465	Biology	History	Chemistry	Physics A-level	Geography	English A level
101730705547	Chemistry	Maths Alevel	Biology	Physics A-level	Economics	Spanish
101900533424	Politics	History	Economics	Physics A-level	Latin	Spanish
102023353588	Chemistry	History	Theory of Knowledge	Politics	Philosophy IB	Economics
102220705547	Chemistry	Biology	Economics	Maths Alevel	Physics A-level	Geography
102659533424	Chemistry	Spanish	Maths IB	English IB	History	Biology
102855705547	Politics	Geography	English A level	RS A Level	Spanish	Economics

164522705547	Art	Geography	English A level	Economics	History	Biology
170405705547	Economics	Biology	Geography	Politics	History	Chemistry
170447705547	Economics	Politics	Biology	Physics A-level	Chinese	Maths Alevel
173410705547	Maths IB	Politics	Art	Geography	Physics A-level	Maths IB Studies
1906487607	Chemistry	Politics	Physics A-level	Economics	History	Biology
2230808502	Art	History	Economics	History of Art	French	RS A Level
2235712028	Economics	Biology	Chemistry	Physics A-level	Chemistry	Philosophy IB
3455871538	Maths Alevel	Chemistry	Physics A-level	Biology	Economics	

AWAITING RESPONSES

- TseZ
- Brooks-HughesH
- PalmerN
- AshbyL
- Stewart-RichardsonEm

Edit The Table

Id of the person you want to edit

Which choice would you like to edit (Choice1, Choice2 etc)

What subject would you like to change this to (Must be exact)

[Previous Page](#)

[Back](#)

Sort Girls

[SORT THE GIRLS!](#)

New Screen:

The old page had a lot of unnecessary information, therefore to improve this page, I will be removing the unnecessary words above the links to other pages as the links themselves are self-explanatory. I will also be removing the responses received header as it is also unnecessary. Other than these changes, I have applied the same style elements as I have applied in the previous pages to make these pages uniform.

ners | Uden x localhost/PROJECT/login/staffint... x +

localhost/PROJECT/login/staffinterface3.php

RESPONSES

Id	First Choice	Second Choice	Third Choice	Fourth Choice	Fifth Choice	Sixth Choice
85447868892	Maths Alevel	Maths IB	Physics A-level	Maths IB Studies	Theory of Knowledge	Computing
84626705547	Maths Alevel	Latin	Chemistry	Biology	Physics A-level	German
85807705547	Biology	English IB	Chemistry	Theory of Knowledge	History	Spanish
90357705547	Biology	Chemistry	History	Economics	Spanish	Physics A-level
91504705547	Chemistry	Maths Alevel	Maths IB	Biology	Physics A-level	Spanish
91539643505	Economics	French	Maths Alevel	Spanish	Biology	English A level
91801705547	Biology	Chemistry	Economics	History	Physics A-level	Maths Alevel
92647705547	Theory of Knowledge	Economics	History	English IB	Chinese	Chemistry
92927533424	Economics	Politics	History	Maths Alevel	Latin	RS A Level
92954705547	Maths Alevel	Physics A-level	Computing	Chemistry	Economics	Geography
93245705547	Chemistry	Physics A-level	French	Italian	Maths Alevel	Geography
94003533424	Economics	Biology	Chemistry	English IB	Maths IB	History
94606579518	Chemistry	Maths Alevel	Economics	Physics A-level	Biology	Politics
94854533424	Theory of Knowledge	English IB	Computing	History	Politics	Economics
95358533424	Biology	French	Chemistry	Maths Alevel	Spanish	English A level
95424410122	Theory of Knowledge	English IB	Maths IB	Physics A-level	History	Politics
101130705547	Politics	Economics	Philosophy IB	RS A Level	Theory of Knowledge	Geography
101258705547	Chemistry	Biology	Spanish	Latin	Maths IB	History
101321965465	Biology	History	Chemistry	Physics A-level	Geography	English A level
101730705547	Chemistry	Maths Alevel	Biology	Physics A-level	Economics	Spanish
101900533424	Politics	History	Economics	Physics A-level	Latin	Spanish
102023353588	Chemistry	History	Theory of Knowledge	Politics	Philosophy IB	Economics
102220705547	Chemistry	Biology	Economics	Maths Alevel	Physics A-level	Geography
102659533424	Chemistry	Spanish	Maths IB	English IB	History	Biology
102855705547	Politics	Geography	English A level	RS A Level	Spanish	Economics
102906533424	English IB	History	Theory of Knowledge	Philosophy IB	Politics	Maths IB

162557705547	History	Economics	Biology	Maths IB	Philosophy IB	Physics A-level
162617533424	Biology	Art	Physics A-level	Chemistry	Maths Alevel	Economics
164522705547	Art	Geography	English A level	Economics	History	Biology
170405705547	Economics	Biology	Geography	Politics	History	Chemistry
170447705547	Economics	Politics	Biology	Physics A-level	Chinese	Maths Alevel
173410705547	Maths IB	Politics	Art	Geography	Physics A-level	Maths IB Studies
1906487607	Chemistry	Politics	Physics A-level	Economics	History	Biology
2230808502	Art	History	Economics	History of Art	French	RS A Level
2235712028	Economics	Biology	Chemistry	Physics A-level	Chemistry	Philosophy IB
3455871538	Maths Alevel	Chemistry	Physics A-level	Biology	Economics	

AWAITING RESPONSES

- TseZ
- Brooks-HughesH
- PalmerN
- AshbyL
- Stewart-RichardsonEm

Edit The Table

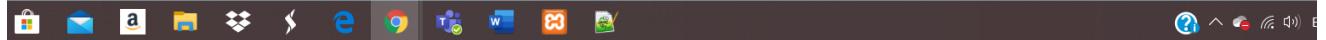
Id of the person you want to edit

Which choice would you like to edit (Choice1, Choice2 etc)

What subject would you like to change this to (Must be exact)

[Previous Page](#)

[SORT THE GIRLS!](#)



Include the last interface where Mrs Jackson can download files with CSS: (Arranged girls)

This functionality and screen view of this interface was located on the sortgirls.php and also on the download_done.php files, where the download_done.php file force downloads a zip of all the session files and the final file onto the user's computer when a download link is clicked. The screen view on the sortgirls.php page displays the final.csv file on the screen as well as prepares the zip package to be downloaded.

```
531 //array below includes all the files I intend to include in my zip file
532 $files = array('session1.csv', 'session2.csv', 'session3.csv', 'session4.csv', 'final.csv');
533 //the name of the zip file
534 $zipname = 'Tasterpack.zip';
535 //uses this php functionality to create the zip file
536 $zip = new ZipArchive;
537 $zip->open($zipname, ZipArchive::CREATE);
538 //runs through array adding each file
539 foreach ($files as $file) {
540     $zip->addFile($file);
541 }
542 //closes the file
543 $zip->close();
```

The code above creates the zip package, and is further explained in the comments of the code.

```
545 <?php
546 //creates a function that outlines the format
547 //of the rows in the table
548 function print_row(&$item) {
549     //html elements necessary
550     echo('<tr>');
551     //calls the function below here
552     array_walk($item, 'print_cell');
553     echo('</tr>');
554 }
555 //creates a function that outlines the format
556 //of the cells in the table
557 function print_cell(&$item) {
558     //html elements necessary
559     echo('<td>');
560     echo($item);
561     echo('</td>');
562 }
563 ?>
```

The code above is the php used to format the appearance of the table, and is also explained further in the comments.

```

564 <head>
565   <style>
566     body {
567       background: lightblue;
568     }
569   .content {
570     max-width: 1000px;
571     margin: auto;
572     background: white;
573     padding: 10px;
574   }
575 </style>
576 </head>
577 <body>
578   <div class = "content">
579     <table>
580       <?php array_walk($final, 'print_row');//calls functions?>
581     </table>
582     <br>
583     <a href="download_done.php" >DOWNLOAD FILES </a>
584     <br>
585   </div>
586 </body>
587 </html>

```

The code above is where the table itself is displayed. The style of the page is the theme outlined throughout the website. The a href at the bottom of the page links to the file that I previously stated downloads the zip package prepared on this page onto the user's PC. Below is the page that the a href links to.

```

1 <?php
2   // output headers so that the file is downloaded rather than displayed
3   //name of the file in the server docs
4   $zipname = 'Tasterpack.zip';
5   //identify what should be done with file and forces download
6   header('Content-Type: application/zip');
7   header('Content-disposition: attachment; filename='.$zipname);
8   header('Content-Length: ' . filesize($zipname));
9   readfile($zipname);
10 ?>

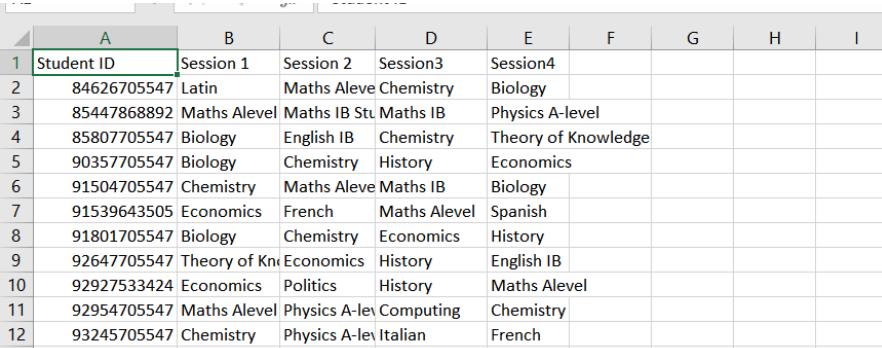
```

Test Plan:

Test Data	Relation to Success Criteria	Algorithm Name	Expected Output	Testing Evidence	Pass or Fail (P/F)
Data in the final.csv file.	Display table of sessions to the user	DISPLAY TABLE	Values in csv file, must be the same as those presented in the table	1&2	P
Is Download button there?	Download button on Staff interface once table of sessions is ready	DOWNLOAD TABLE	Download link should be clear and visible	3	P/F
Data in the final csv file	Download button on Staff interface once table of	DOWNLOAD TABLE	Final csv file should be the same as the file downloaded by user	4,5	P

	sessions is ready.				
Data in the session1 csv file	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Session 1 csv file should be the same as the file downloaded by user	4,6	P
Data in the session2 csv	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Session 2 csv file should be the same as the file downloaded by user	4,7	P
Data in the session3 csv file	Download button on Staff interface once table of sessions is ready.	DOWNLOAD TABLE	Session 3 csv file should be the same as the file downloaded by user	4,8	P
Data in the session4 csv file	Download button on Staff interface once table of sessions is ready	DOWNLOAD TABLE	Session 4 csv file should be the same as the file downloaded by user	4,9	P

Testing Evidence

1	Data in CSV file:																																																																																																																														
	 <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> <th>H</th> <th>I</th> </tr> </thead> <tbody> <tr><td>1 Student ID</td><td>Session 1</td><td>Session 2</td><td>Session3</td><td>Session4</td><td></td><td></td><td></td><td></td></tr> <tr><td>2 84626705547</td><td>Latin</td><td>Maths</td><td>A leve</td><td>Chemistry</td><td>Biology</td><td></td><td></td><td></td></tr> <tr><td>3 85447868892</td><td>Maths Alevel</td><td>Maths IB</td><td>Stu</td><td>Maths IB</td><td>Physics A-level</td><td></td><td></td><td></td></tr> <tr><td>4 85807705547</td><td>Biology</td><td>English IB</td><td></td><td>Chemistry</td><td>Theory of Knowledge</td><td></td><td></td><td></td></tr> <tr><td>5 90357705547</td><td>Biology</td><td>Chemistry</td><td></td><td>History</td><td>Economics</td><td></td><td></td><td></td></tr> <tr><td>6 91504705547</td><td>Chemistry</td><td>Maths A leve</td><td></td><td>Maths IB</td><td>Biology</td><td></td><td></td><td></td></tr> <tr><td>7 91539643505</td><td>Economics</td><td>French</td><td></td><td>Maths Alevel</td><td>Spanish</td><td></td><td></td><td></td></tr> <tr><td>8 91801705547</td><td>Biology</td><td>Chemistry</td><td></td><td>Economics</td><td>History</td><td></td><td></td><td></td></tr> <tr><td>9 92647705547</td><td>Theory of Kn</td><td>Economics</td><td></td><td>History</td><td>English IB</td><td></td><td></td><td></td></tr> <tr><td>10 92927533424</td><td>Economics</td><td>Politics</td><td></td><td>History</td><td>Maths Alevel</td><td></td><td></td><td></td></tr> <tr><td>11 92954705547</td><td>Maths Alevel</td><td>Physics A-le</td><td></td><td>Computing</td><td>Chemistry</td><td></td><td></td><td></td></tr> <tr><td>12 93245705547</td><td>Chemistry</td><td>Physics A-le</td><td></td><td>Italian</td><td>French</td><td></td><td></td><td></td></tr> <tr><td>13 04002522474</td><td>Economics</td><td>Biolog</td><td></td><td>Chemictor</td><td>English IP</td><td></td><td></td><td></td></tr> </tbody> </table>	A	B	C	D	E	F	G	H	I	1 Student ID	Session 1	Session 2	Session3	Session4					2 84626705547	Latin	Maths	A leve	Chemistry	Biology				3 85447868892	Maths Alevel	Maths IB	Stu	Maths IB	Physics A-level				4 85807705547	Biology	English IB		Chemistry	Theory of Knowledge				5 90357705547	Biology	Chemistry		History	Economics				6 91504705547	Chemistry	Maths A leve		Maths IB	Biology				7 91539643505	Economics	French		Maths Alevel	Spanish				8 91801705547	Biology	Chemistry		Economics	History				9 92647705547	Theory of Kn	Economics		History	English IB				10 92927533424	Economics	Politics		History	Maths Alevel				11 92954705547	Maths Alevel	Physics A-le		Computing	Chemistry				12 93245705547	Chemistry	Physics A-le		Italian	French				13 04002522474	Economics	Biolog		Chemictor	English IP			
A	B	C	D	E	F	G	H	I																																																																																																																							
1 Student ID	Session 1	Session 2	Session3	Session4																																																																																																																											
2 84626705547	Latin	Maths	A leve	Chemistry	Biology																																																																																																																										
3 85447868892	Maths Alevel	Maths IB	Stu	Maths IB	Physics A-level																																																																																																																										
4 85807705547	Biology	English IB		Chemistry	Theory of Knowledge																																																																																																																										
5 90357705547	Biology	Chemistry		History	Economics																																																																																																																										
6 91504705547	Chemistry	Maths A leve		Maths IB	Biology																																																																																																																										
7 91539643505	Economics	French		Maths Alevel	Spanish																																																																																																																										
8 91801705547	Biology	Chemistry		Economics	History																																																																																																																										
9 92647705547	Theory of Kn	Economics		History	English IB																																																																																																																										
10 92927533424	Economics	Politics		History	Maths Alevel																																																																																																																										
11 92954705547	Maths Alevel	Physics A-le		Computing	Chemistry																																																																																																																										
12 93245705547	Chemistry	Physics A-le		Italian	French																																																																																																																										
13 04002522474	Economics	Biolog		Chemictor	English IP																																																																																																																										
2	Data displayed:																																																																																																																														

3	<p>I said this was a P/F – because it is not a button but a link, I thought a button would be unnecessarily complex:</p> <table border="1"> <tr><td>2230808502</td><td>Art</td><td>History of Art</td><td>History</td><td>Economics</td></tr> <tr><td>2235712028</td><td>Chemistry</td><td>Physics A-level</td><td>Economics</td><td>Biology</td></tr> <tr><td>3455871538</td><td>Maths Alevel</td><td>Chemistry</td><td>Physics A-level</td><td>Biology</td></tr> </table> <p>DOWNLOAD FILES</p>	2230808502	Art	History of Art	History	Economics	2235712028	Chemistry	Physics A-level	Economics	Biology	3455871538	Maths Alevel	Chemistry	Physics A-level	Biology																											
2230808502	Art	History of Art	History	Economics																																							
2235712028	Chemistry	Physics A-level	Economics	Biology																																							
3455871538	Maths Alevel	Chemistry	Physics A-level	Biology																																							
4	<p>Zip file downloaded:</p> <p>In Zip file:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Compressed size</th> <th>Password p...</th> <th>Size</th> <th>Ratio</th> <th>Date modified</th> </tr> </thead> <tbody> <tr><td>final.csv</td><td>Microsoft Excel Comma S...</td><td>2 KB</td><td>No</td><td>8 KB</td><td>80%</td><td>03/04/2020 13:32</td></tr> <tr><td>session1.csv</td><td>Microsoft Excel Comma S...</td><td>1 KB</td><td>No</td><td>2 KB</td><td>59%</td><td>03/04/2020 13:32</td></tr> <tr><td>session2.csv</td><td>Microsoft Excel Comma S...</td><td>1 KB</td><td>No</td><td>2 KB</td><td>59%</td><td>03/04/2020 13:32</td></tr> <tr><td>session3.csv</td><td>Microsoft Excel Comma S...</td><td>1 KB</td><td>No</td><td>2 KB</td><td>59%</td><td>03/04/2020 13:32</td></tr> <tr><td>session4.csv</td><td>Microsoft Excel Comma S...</td><td>1 KB</td><td>No</td><td>2 KB</td><td>59%</td><td>03/04/2020 13:32</td></tr> </tbody> </table>	Name	Type	Compressed size	Password p...	Size	Ratio	Date modified	final.csv	Microsoft Excel Comma S...	2 KB	No	8 KB	80%	03/04/2020 13:32	session1.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32	session2.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32	session3.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32	session4.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32
Name	Type	Compressed size	Password p...	Size	Ratio	Date modified																																					
final.csv	Microsoft Excel Comma S...	2 KB	No	8 KB	80%	03/04/2020 13:32																																					
session1.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32																																					
session2.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32																																					
session3.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32																																					
session4.csv	Microsoft Excel Comma S...	1 KB	No	2 KB	59%	03/04/2020 13:32																																					
5	<p>Data in final file:</p> <p>Downloaded data:</p>																																										

	1	Student ID	Session 1	Session 2	Session3	Session4		
2	84626705547	Latin		Maths Aleve	Chemistry	Biology		
3	85447868892	Maths Alevel		Maths IB	Stu	Maths IB	Physics A-level	
4	85807705547	Biology		English IB	Chemistry	Theory of Knowledge		
5	90357705547	Biology		Chemistry	History	Economics		
6	91504705547	Chemistry		Maths Aleve	Maths IB	Biology		
7	91539643505	Economics		French	Maths Alevel	Spanish		
8	91801705547	Biology		Chemistry	Economics	History		
9	92647705547	Theory of Kn		Economics	History	English IB		
10	92927533424	Economics		Politics	History	Maths Alevel		
11	92954705547	Maths Alevel		Physics A-le	Computing	Chemistry		
12	93245705547	Chemistry		Physics A-le	Italian	French		

6 Data in session 1 file:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Latin	10	4	8.46E+10	1.01E+11	1.1E+11	1.14E+11							
2	Maths Alev	10	8	8.54E+10	9.3E+10	1.05E+11	1.05E+11	1.21E+11	1.44E+11	1.44E+11	3.46E+09			
3	Chemistry	20	20	9.15E+10	9.32E+10	9.46E+10	1.02E+11	1.02E+11	1.03E+11	1.04E+11	1.05E+11	1.13E+11	1.15E+11	
4	Biology	20	20	8.58E+10	9.04E+10	9.18E+10	9.54E+10	1.01E+11	1.04E+11	1.12E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11
5	Maths IB	10		1	1.73E+11									
6	Physics A-l	10	3	1.04E+11	1.2E+11	1.44E+11								
7	English IB	10	2	1.03E+11	1.35E+11									
8	Theory of	10	5	9.26E+10	9.49E+10	9.54E+10	1.14E+11	1.21E+11						
9	History	10	2	1.11E+11	1.63E+11									
10	Economics	20	20	9.15E+10	9.29E+10	9.4E+10	1.04E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11	1.23E+11	1.32E+11	
11	French	10	1	1.45E+11										
12	Spanish	10	0											
13	Politics	20	17	1.01E+11	1.02E+11	1.03E+11	1.05E+11	1.06E+11	1.11E+11	1.14E+11	1.15E+11	1.16E+11	1.31E+11	
14	Computing	10	2	1.14E+11	1.15E+11									

Downloaded Data:

A	B	C	D	E	F	G	H	I	J	K	L	
1	Latin	10	4	8.46E+10	1.01E+11	1.1E+11	1.14E+11					
2	Maths Alev	10	8	8.54E+10	9.3E+10	1.05E+11	1.05E+11	1.21E+11	1.44E+11	1.44E+11	3.46E+09	
3	Chemistry	20	20	9.15E+10	9.32E+10	9.46E+10	1.02E+11	1.02E+11	1.03E+11	1.04E+11	1.05E+11	
4	Biology	20	20	8.58E+10	9.04E+10	9.18E+10	9.54E+10	1.01E+11	1.04E+11	1.12E+11	1.13E+11	1.15E+11
5	Maths IB	10	1	1.73E+11								
6	Physics A-l	10	3	1.04E+11	1.2E+11	1.44E+11						
7	English IB	10	2	1.03E+11	1.35E+11							
8	Theory of	10	5	9.26E+10	9.49E+10	9.54E+10	1.14E+11	1.21E+11				
9	History	10	2	1.11E+11	1.63E+11							
10	Economics	20	20	9.15E+10	9.29E+10	9.4E+10	1.04E+11	1.13E+11	1.15E+11	1.2E+11	1.21E+11	1.22E+11

7 Data in session 2 file:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Maths IB S	10	3	8.54E+10	1.21E+11	1.62E+11									
2	Maths Alev	10	7	8.46E+10	9.15E+10	9.46E+10	1.02E+11	1.1E+11	1.32E+11	1.55E+11					
3	Chemistry	20	13	9.04E+10	9.18E+10	1.01E+11	1.05E+11	1.11E+11	1.12E+11	1.13E+11	1.2E+11	1.21E+11	1.41E+11	1.41E+11	3.46E+09
4	Biology	10	10	9.4E+10	1.02E+11	1.14E+11	1.15E+11	1.15E+11	1.21E+11	1.44E+11	1.6E+11	1.63E+11	1.7E+11		
5	Maths IB	10	3	1.04E+11	1.21E+11	1.35E+11									
6	Physics A-l	10	9	9.3E+10	9.32E+10	1.04E+11	1.05E+11	1.13E+11	1.13E+11	1.17E+11	1.7E+11	1.91E+09	2.24E+09		
7	English IB	10	5	8.58E+10	9.49E+10	9.54E+10	1.05E+11	1.45E+11							
8	Theory of	10	2	1.05E+11	1.15E+11										
9	History	10	10	1.01E+11	1.02E+11	1.02E+11	1.03E+11	1.15E+11	1.32E+11	1.44E+11	1.53E+11	1.55E+11	1.62E+11		
10	Economics	20	20	9.26E+10	1.01E+11	1.11E+11	1.13E+11	1.13E+11	1.14E+11	1.15E+11	1.16E+11	1.16E+11	1.21E+11	1.31E+11	1.44E+11
11	French	10	4	9.15E+10	9.54E+10	1.04E+11	1.14E+11	1.62E+11							
12	Spanish	10	3	1.03E+11	1.2E+11	1.35E+11									
13	Politics	10	10	9.29E+10	1.11E+11	1.12E+11	1.14E+11	1.2E+11	1.23E+11	1.44E+11	1.52E+11	1.52E+11	1.61E+11		
14	Computing	10	0												
15	Philosophy	10	4	1.06E+11	1.15E+11	1.15E+11	1.33E+11								
16	RS A Level	10	5	1.04E+11	1.33E+11	1.45E+11	1.46E+11	1.61E+11							
17	Geography	10	5	1.03E+11	1.21E+11	1.44E+11	1.55E+11	1.65E+11							
18	English A lk	10	2	1.15E+11	1.62E+11										
19	Chinese	10	3	1.04E+11	1.11E+11	1.61E+11									
20	Art	10	2	1.3E+11	1.73E+11										
21	History of	10	4	1.04E+11	1.22E+11	1.51E+11	2.23E+09								

Downloaded Data:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Maths IB S	10	3	8.54E+10	1.21E+11	1.62E+11										
2	Maths Alev	10	7	8.46E+10	9.15E+10	9.46E+10	1.02E+11	1.1E+11	1.32E+11	1.55E+11						
3	Chemistry	20	13	9.04E+10	9.18E+10	1.01E+11	1.05E+11	1.11E+11	1.12E+11	1.13E+11	1.2E+11	1.21E+11	1.41E+11	1.41E+11	3.46E+09	
4	Biology	10	10	9.4E+10	1.02E+11	1.14E+11	1.15E+11	1.15E+11	1.21E+11	1.44E+11	1.6E+11	1.63E+11	1.7E+11			
5	Maths IB	10	3	1.04E+11	1.21E+11	1.35E+11										
6	Physics A-l	10	9	9.3E+10	9.32E+10	1.04E+11	1.05E+11	1.13E+11	1.13E+11	1.17E+11	1.7E+11	1.91E+09	2.24E+09			
7	English IB	10	5	8.58E+10	9.49E+10	9.54E+10	1.05E+11	1.45E+11								
8	Theory of	10	2	1.05E+11	1.15E+11											
9	History	10	10	1.01E+11	1.02E+11	1.02E+11	1.03E+11	1.15E+11	1.32E+11	1.44E+11	1.53E+11	1.55E+11	1.62E+11			
10	Economics	20	20	9.26E+10	1.01E+11	1.11E+11	1.13E+11	1.13E+11	1.14E+11	1.15E+11	1.16E+11	1.16E+11	1.21E+11	1.31E+11	1.44E+11	1.44E+11
11	French	10	4	9.15E+10	9.54E+10	1.04E+11	1.14E+11	1.62E+11								
12	Spanish	10	3	1.03E+11	1.2E+11	1.35E+11										
13	Politics	10	10	9.29E+10	1.11E+11	1.12E+11	1.14E+11	1.2E+11	1.23E+11	1.44E+11	1.52E+11	1.52E+11	1.61E+11			
14	Computing	10	0													
15	Philosophy	10	4	1.06E+11	1.15E+11	1.15E+11	1.33E+11									
16	RS A Level	10	5	1.04E+11	1.33E+11	1.45E+11	1.46E+11	1.61E+11								
17	Geography	10	5	1.03E+11	1.21E+11	1.44E+11	1.55E+11	1.65E+11								
18	English A lk	10	2	1.15E+11	1.62E+11											
19	Chinese	10	3	1.04E+11	1.11E+11	1.61E+11										
20	Art	10	2	1.3E+11	1.73E+11											
21	History of	10	4	1.04E+11	1.22E+11	1.51E+11	2.23E+09									

8 Data in session 3 file:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1 Italian	10	5	9.32E+10	1.05E+11	1.21E+11	1.45E+11	1.46E+11									
2 Maths Alev	10	8	9.15E+10	1.04E+11	1.04E+11	1.11E+11	1.11E+11	1.13E+11	1.15E+11	1.62E+11						
3 Chemistry	20	13	8.46E+10	8.58E+10	9.4E+10	9.54E+10	1.01E+11	1.04E+11	1.04E+11	1.1E+11	1.14E+11	1.22E+11	1.23E+11	1.35E+11	1.61E+11	
4 Biology	10	9	1.01E+11	1.02E+11	1.04E+11	1.05E+11	1.15E+11	1.21E+11	1.45E+11	1.62E+11	1.63E+11					
5 Maths IB	10	4	8.54E+10	9.15E+10	9.54E+10	1.03E+11										
6 Physics A-I	10	9	1.14E+11	1.3E+11	1.32E+11	1.35E+11	1.52E+11	1.52E+11	1.55E+11	1.63E+11	3.46E+09					
7 English IB	10	1	1.2E+11													
8 Theory of	10	5	1.02E+11	1.03E+11	1.21E+11	1.33E+11	1.44E+11									
9 History	10	7	9.04E+10	9.26E+10	9.29E+10	1.11E+11	1.33E+11	1.52E+11	2.23E+09							
10 Economics	20	19	9.18E+10	9.46E+10	1.02E+11	1.05E+11	1.05E+11	1.06E+11	1.13E+11	1.15E+11	1.15E+11	1.2E+11	1.21E+11	1.21E+11	1.21E+11	1.21E+11
11 French	10	3	1.13E+11	1.15E+11	1.44E+11											
12 Spanish	10	1	1.16E+11													
13 Politics	10	10	1.13E+11	1.13E+11	1.21E+11	1.44E+11	1.53E+11	1.55E+11	1.61E+11	1.62E+11	1.62E+11	1.7E+11				
14 Computing	10	2	9.3E+10	9.49E+10												
15 Philosophy	10	7	1.01E+11	1.14E+11	1.15E+11	1.32E+11	1.52E+11	1.6E+11	1.61E+11							
16 RSA Level	10	4	1.15E+11	1.44E+11	1.44E+11	1.61E+11										
17 Geography	10	6	1.14E+11	1.15E+11	1.41E+11	1.61E+11	1.7E+11	1.73E+11								
18 English A-Ik	10	7	1.03E+11	1.12E+11	1.31E+11	1.44E+11	1.45E+11	1.45E+11	1.65E+11							
19 Chinese	10	0														
20 Art	10	1	1.41E+11													
21 History of	10	3	1.11E+11	1.12E+11	1.16E+11											
22																

Downloaded Data:

1 Italian	10	5	9.32E+10	1.05E+11	1.21E+11	1.45E+11	1.46E+11									
2 Maths Alev	10	8	9.15E+10	1.04E+11	1.04E+11	1.11E+11	1.11E+11	1.13E+11	1.15E+11	1.62E+11						
3 Chemistry	20	13	8.46E+10	8.58E+10	9.4E+10	9.54E+10	1.01E+11	1.04E+11	1.04E+11	1.1E+11	1.14E+11	1.22E+11	1.23E+11	1.35E+11	1.61E+11	
4 Biology	10	9	1.01E+11	1.02E+11	1.04E+11	1.05E+11	1.15E+11	1.21E+11	1.45E+11	1.62E+11	1.63E+11					
5 Maths IB	10	4	8.54E+10	9.15E+10	9.54E+10	1.03E+11										
6 Physics A-I	10	9	1.14E+11	1.3E+11	1.32E+11	1.35E+11	1.52E+11	1.52E+11	1.55E+11	1.63E+11	3.46E+09					
7 English IB	10	1	1.2E+11													
8 Theory of	10	5	1.02E+11	1.03E+11	1.21E+11	1.33E+11	1.44E+11									
9 History	10	7	9.04E+10	9.26E+10	9.29E+10	1.11E+11	1.33E+11	1.52E+11	2.23E+09							
10 Economics	20	19	9.18E+10	9.46E+10	1.02E+11	1.05E+11	1.05E+11	1.06E+11	1.13E+11	1.15E+11	1.15E+11	1.2E+11	1.21E+11	1.21E+11	1.21E+11	1.21E+11
11 French	10	3	1.13E+11	1.15E+11	1.44E+11											
12 Spanish	10	1	1.16E+11													
13 Politics	10	10	1.13E+11	1.13E+11	1.21E+11	1.44E+11	1.53E+11	1.55E+11	1.61E+11	1.62E+11	1.62E+11	1.7E+11				
14 Computing	10	2	9.3E+10	9.49E+10												
15 Philosophy	10	7	1.01E+11	1.14E+11	1.15E+11	1.32E+11	1.52E+11	1.6E+11	1.61E+11							
16 RSA Level	10	4	1.15E+11	1.44E+11	1.44E+11	1.61E+11										
17 Geography	10	6	1.14E+11	1.15E+11	1.41E+11	1.61E+11	1.7E+11	1.73E+11								
18 English A-Ik	10	7	1.03E+11	1.12E+11	1.31E+11	1.44E+11	1.45E+11	1.45E+11	1.65E+11							
19 Chinese	10	0														
20 Art	10	1	1.41E+11													
21 History of	10	3	1.11E+11	1.12E+11	1.16E+11											
22																

9

Data in session 4 file:

A	B	C	D	E	F	G	H	I	J	K	L	M
1 German	10	3	1.45E+11	1.45E+11	1.61E+11							
2 Maths Alev	10	10	9.29E+10	9.54E+10	1.02E+11	1.04E+11	1.05E+11	1.06E+11	1.14E+11	1.2E+11	1.44E+11	1.45E+11
3 Chemistry	10	10	9.3E+10	1.04E+11	1.11E+11	1.13E+11	1.13E+11	1.16E+11	1.21E+11	1.21E+11	1.3E+11	1.35E+11
4 Biology	10	10	8.46E+10	9.15E+10	9.55E+10	1.05E+11	1.61E+11	1.61E+11	1.62E+11	1.7E+11	2.24E+09	3.46E+09
5 Maths IB	10	5	1.04E+11	1.63E+11	1.22E+11	1.23E+11	1.44E+11					
6 Physics A-I	10	10	8.54E+10	9.46E+10	9.54E+10	1.01E+11	1.02E+11	1.1E+11	1.41E+11	1.55E+11	1.62E+11	
7 English IB	10	6	9.26E+10	9.4E+10	1.03E+11	1.21E+11	1.21E+11	1.52E+11				
8 Theory of	10	4	8.58E+10	1.15E+11	1.21E+11	1.61E+11						
9 History	10	10	9.18E+10	9.49E+10	1.12E+11	1.13E+11	1.14E+11	1.14E+11	1.15E+11	1.15E+11	1.41E+11	1.52E+11
10 Economics	10	10	9.04E+10	1.11E+11	1.14E+11	1.15E+11	1.15E+11	1.15E+11	1.65E+11	2.23E+09	1.44E+11	1.62E+11
11 French	10	2	9.32E+10	1.32E+11								
12 Spanish	10	6	9.15E+10	1.01E+11	1.04E+11	1.04E+11	1.15E+11	1.6E+11				
13 Politics	10	10	1.02E+11	1.05E+11	1.11E+11	1.13E+11	1.15E+11	1.2E+11	1.7E+11	1.73E+11	1.91E+09	1.44E+11
14 Computing	10	2	1.32E+11	1.44E+11								
15 Philosophy	10	6	1.03E+11	1.05E+11	1.16E+11	1.2E+11	1.35E+11	1.52E+11				
16 RSA Level	10	5	1.01E+11	1.03E+11	1.11E+11	1.31E+11	1.44E+11					
17 Geography	10	7	1.12E+11	1.13E+11	1.21E+11	1.33E+11	1.52E+11	1.53E+11	1.44E+11			
18 English A-Ik	10	3	1.45E+11	1.46E+11	1.61E+11							
19 Chinese	10	3	1.05E+11	1.33E+11	1.51E+11							
20 Art	10	1	1.62E+11									
21 History of	10	0										
22												

Downloaded Data:

1 German	10	3	1.45E+11	1.45E+11	1.61E+11							
2 Maths Alev	10	10	9.29E+10	9.54E+10	1.02E+11	1.04E+11	1.05E+11	1.06E+11	1.14E+11	1.2E+11	1.44E+11	1.45E+11
3 Chemistry	10	10	9.3E+10	1.04E+11	1.11E+11	1.13E+11	1.13E+11	1.16E+11	1.21E+11	1.21E+11	1.3E+11	1.35E+11
4 Biology	10	10	8.46E+10	9.15E+10	9.55E+10	1.05E+11	1.61E+11	1.61E+11	1.62E+11	1.7E+11	2.24E+09	3.46E+09
5 Maths IB	10	5	1.04E+11	1.63E+11	1.22E+11	1.23E+11	1.44E+11					
6 Physics A-I	10	10	8.54E+10	9.46E+10	9.54E+10	1.01E+11	1.02E+11	1.02E+11	1.1E+11	1.41E+11	1.55E+11	1.62E+11
7 English IB	10	6	9.26E+10	9.4E+10	1.03E+11	1.21E+11	1.21E+11	1.52E+11				
8 Theory of	10	4	8.58E+10	1.15E+11	1.21E+11	1.61E+11						
9 History	10	10	9.18E+10	9.49E+10	1.12E+11	1.13E+11	1.14E+11	1.14E+11	1.15E+11	1.15E+11	1.41E+11	1.52E+11
10 Economics	10	10	9.04E+10	1.11E+11	1.14E+11	1.15E+11	1.15E+11	1.15E+11	1.65E+11	2.23E+09	1.44E+11	1.62E