SYSC 4504

Project documentation

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student number | FirstName LastName | Email @cmail.carleton.ca |
| 1 | 100870746 | Mostafa Shabka | mostafashabka |
| 2 | 100833573 | Abdullah Adeeb | abdullahadeeb |
| 3 | 100819643 | Mohamad Ghadieh | mohamadghadieh |

2/The program used to test the application was: CSE

3/ The TA assigned for the marking of our project is Mr.Kazi

4/Abdullah Adeeb: Timetable.php, Server.php, functions2.js,View1.php, View2.php, javaclient,

Mostafa Shabka: Server.php, View1.php, View2.php, functions.js, functions2.js, install.php, javaclient, database.php

Mohamad Ghadieh: Server.php, View1.php, View2.php, functions.js, functions2.js, install.php, javaclient, index.php

5/ index.php: takes in the username, password and program of the person logging in and redirects to view1.php

* View1.php: displays the timetable of the program and lets the user check the courses that he or she have finished. Upon checking the “Done” checkbox, other courses get opened if the courses that are prerequisites got checked. There is also a register button that for the courses that the user wants to register in, based on these courses the conflict free timetable will be produced. To view that schedule, press next and then view1 redirects the user to view2.php.
* View2.php: contains the conflict free timetable that was based on the courses that had “register” checkbox, checked in view1. It displays the conflict free tables contained in the array that was forwarded from the server in a table
* Functions.js: includes all the javascript functions called in order to display a correct view1.php. These functions are for view1.php only and get called upon loading the window (some functions call other functions in the same file .js file). It can be considered as the connection between the server and the client view1.php
* Functions2.js: includes all the javascript functions called in order to display correct conflict free schedules in view2.php. These functions are for view2.php only and get called upon loading the window (some functions call other functions in the same file .js file). It can be considered as the connection between the server and the client view2.php
* Timetable.php: It runs the algorithm that comes up with courses that do not conflict with each other and places them in an array that gets sent to the server, which forward s it to view2.php
* Database.php: Connects all the different files to the database and also contains functions that interact with the tables in the database in order for them to be used by the functions in the .js files
* Install.php: Installs the appropriate database and tables needed, the user must provide the localhost, username and password in order to install the aforementioned elements so that the other files could coherently interact.
* Server.php: Interacts with the tables in the database using db.php which gets imported at the beginning and forwards the appropriate info to the .js functions as well as the clients based on the typeofrequest submitted by the respective client. The interactions are done using XMLHTTPRequest. The response from the server is always a string since it cannot return an array of objects

6/The code of this project needs minimal change to implement **ALL engineering majors and also ANY major** in Carleton, file that would need to be changed in order to suit other engineering programs would be the localhost.sql, since table containing all the courses for the additional major would have to be added in the database. In the view1.php, the major would have to be added to the dropdown. Once selected, the code runs the same way since it is robust and the only change required would be the addition of the courses of the new major in the database as mentioned before.

7/In server.php, function isCourseSatisfied is used to check the prerequisite year of the course and compare it the year standing of the user. It is also used to check for the courses that can be taken concurrently. If the course can be taken concurrently, then it should be in the prerequisite courses and are in the registered courses list. Our application does check for program transfer because each course that is added to the registered courses list is checked for **ALL** necessary prerequisites.

8/ The TimeTable.php contains a function buildTimeTable() which is a recursive backtracking algorithm that gets all the conflict free table possibilities. Because this function could take a long time, it will return finish once it finds 5 unique conflict-free timetables OR if it didn't find 5 tables within 5 seconds it will timeout and return no tables resolved.

For the example given, the following will happen

1- Lectures and labs of the same section will be grouped

2- Sort the list of courses starting with the course having the least amount of session(lectures and labs)

3- try the first lecture with the first lab if it worked you move on if it didn't you take them out and try another session option for the same lecture.

4- do step number three until all courses has been added with a lecture and lab.

5- backtrack and try a different lab then diffent lecture.

In summary, the function will exhaust every possible combination of courses. Keeping in consideration that the section must not be full, and lab must be of the same section also available in time and space

9/The entity that identifies the courses that can be taken based on the completed courses and the prerequisites is Server.php and the function is isCourseSatisfied(course). Regarding the conflict free timetable, Timetable.php and the function that calls it is located in functions2.js and is called buildTables(timetable, courses).

These choices were based on the fact of dedicating every file for a specific goal, meaning that the view files were meant for display purposes (containing divs to be filled), on window load, these view files called functions located in functions.js and functions2.js that interact with the server and get the info required. Plus there are multiple clients, so the server does all the operations and sends it to the equivalent clients including java depending on the variable requestype.

**Instructions on how the application works(HTML):**

Please note: The username and password functionality holds no real purpose, although their values are being retrieved by the server but it’s not being validated as the requirements were misleading and ambiguous on this aspect and didn’t give any marking worth to this part. We believe that we have demonstrated a good understanding of user ajax interaction with the server/Database on other parts that this part should be overlooked.

* Go to the db.php and change the username and password to the user’s own. Same procedure for install.php
* Run the install.php file first
* Run the index.php file using a browser
* Input a username, password and choose the major (currently 1 major CSE if more to be added, see question 6 on how to do so)
* In view1.php:

-Retrieves all the courses in the major from the Database as mentioned above and displays them appropriately.

-Put the Year standing on top, and the term that the user would like to register in, since the timetable is per semester. Some courses are available during the fall and the winter, if the semester selected is fall and the user wants to register in a course in the winter but doesn’t exist in the fall, an error message will be displayed saying that the course isn’t available for the desired term.

- Check the Done checkbox if the course has been completed, it opens the courses that are in grey and require some courses to be done since these courses are prerequisites

-Can’t check done if register is done, and can’t register if course is done.

-The user can click on a course to get its details, like lab sections and times.

-Once all the courses to be registered are done, press next which will lead to the display of multiple conflict free timetables. If there are none then a message stating so will be displayed on the browser.

-Select which timetable is more convenient and press register, which officially registers the student in the course and decreases the room cap of all the courses in the timetable.

**Java Client:**

Please note: the Java client description was very detailed, we asked the prof and he said that it should be simple and needs an on and off pattern.

* Run the jar file
* Input the username and password
* If user is off pattern, display the tree to let them select which courses they have taken
* If user on pattern, get the year standing and display the courses accordingly if 4th year for example, only courses of fourth will be displayed on screen.