

Analysis of Algorithms

Spring 2020

Members Details

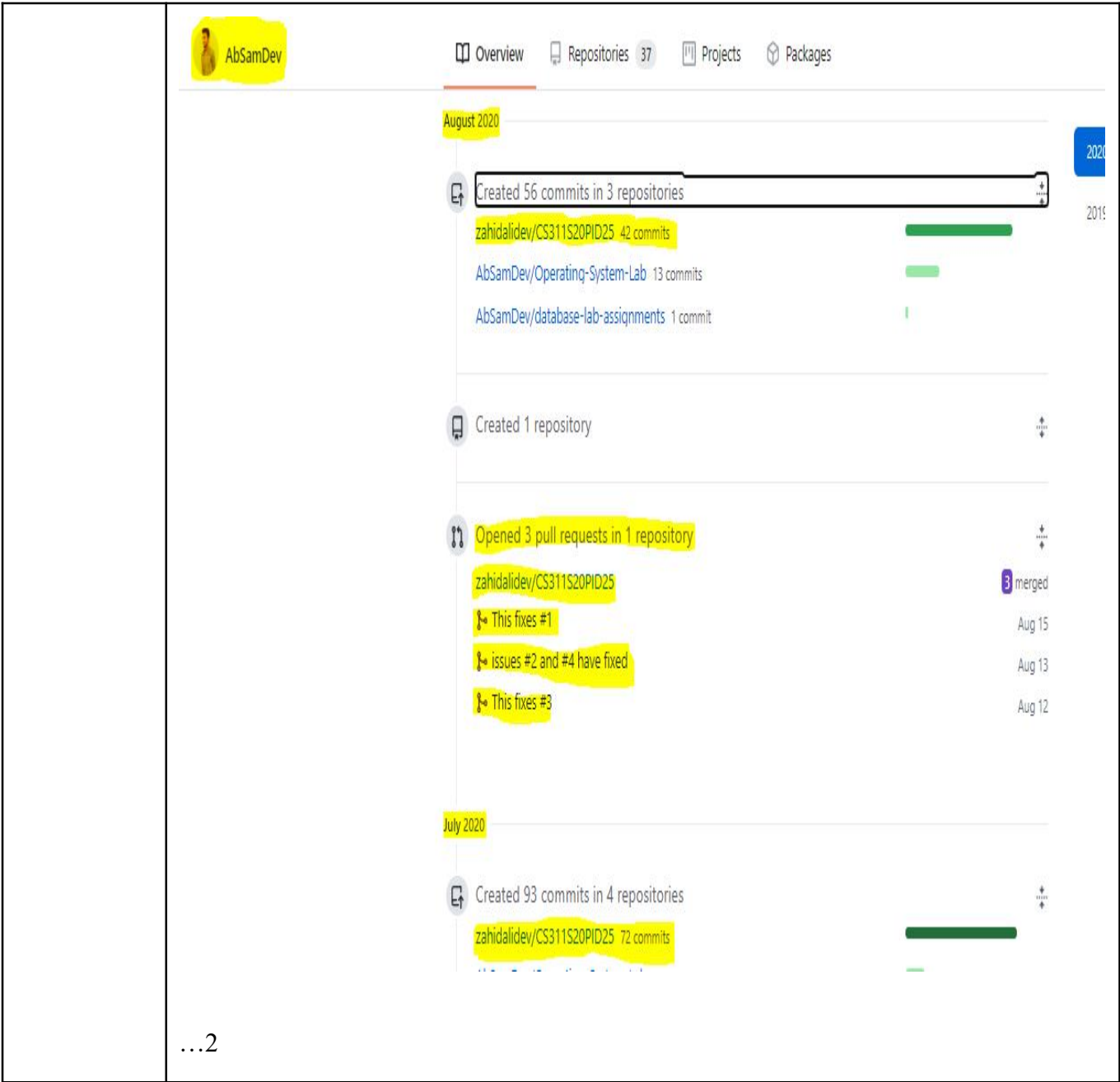
Group ID	CS311S20PID25
Registration Number of Group Members	2018-CS-136 [Zahid Ali] 2018-CS-141 [Abdul Samad]
Section	C

Project Details

<i>Project</i>	
Project Title	Activity Scheduling
Executive Summary	<p>Our project is about to create a timetable for a department of a university by taking some input data as rooms, classes, teachers and courses. In this project, we have built an optimized timetable generator for a department. This is automated scheduling tool for courses of the department. This tool reduces the resources used in department i.e., it will use minimum classrooms and days for lectures. By generating timetable using this automated tool, we can minimize resources and maximize courses for classes in the output timetable. In this Project, teachers and classes are been managed in such a way that they have no more than one lecture at a time. At first level, we add classroom names as input. After that, classes are added as input in the 2nd step. Classroom names and classes should be unique otherwise; the tool will not let us proceed (i.e., navigate to next button will be disabled). Then we proceed to next step after getting input in previous steps. Here we have to add information about teachers like teacher name and teacher's available time slots (i.e., at which time slots the teacher is available during 8am to 4pm). We need to input available time slots for teachers because teachers can be busy in other department if we randomly assign time using this tool. Therefore, to avoid this problem, we can ask teachers at which time slot they are available so we can manage to schedule time for lectures. Then at the 4th and final step, we have to add courses for whole department by entering course name selecting its teacher, class from available teachers and classes that we have entered in previous steps. Then, selecting the credit hours for each course like if we want them to have 3 consecutive lectures, we have to select 3 in the select box. On the other hand, if we want all credit hours of its course to be managed by tool then we select 1, 1, 1 as credit hours and so on for other courses accordingly. Then we enter submit button which take us to the timetable generator page, where user can generate</p>

	timetable for all classes of department by just pressing the generate timetable button and at this page, we can download the generated timetables as an image as well.
Business Case	
Outline the business need for the project	Manually creating timetable is a time consuming task for all the departments of the universities. If they make timetable manually, there will occur conflicts and clashes between different teachers and classes. In addition, there could be time slots available in different classrooms that are free which can be used if timetable is made efficiently. By using this automated timetable generator tool, the conflicts or clash between different teachers and classes will be 0%. Also by using minimum rooms, we can manage more courses.
End user of the product	As this automated timetable-generating tool is related to departments of universities, so we are going to target the departments that build timetable manually. With this automated tool, they can generate optimized timetables.
Motivation for Project	We are motivated to build this project because University students face problems about timetable issues. We (as students) have also faced clashes of rooms as when we have lectures in one room, there is a class that is already having lecture in that room ☹️. And often there are free time slots available in many rooms that are supposed to be used. By using this project that we have built, if we generate a time table then we will be have 0% chance of clash and we can also minimize the rooms so that we can manage more courses, more classes in appropriate number of rooms without any clash (unpleasant scheduling).
Description of the project objective(s)	This automated timetable-generating tool is used to build optimized time table for a university. This project can minimize the use of resources as it minimizes the rooms and can maximize the courses like we can manage more courses in minimum rooms.
State the level of impact expected should the project proceed and implications of not proceeding	Our project is working fine until now as we expected. We have not found any bug because our project generating the optimized timetables for a department which prevent clashes and minimize the resources like classrooms etc.
Functional	The features of our projects are minimizing the resources like rooms etc. and managing

Requirements	the courses in the optimized way, and generating timetable by taking simple inputs from user on front-end, which saves a lot of time too.
Benefits	
What benefits are expected/anticipated?	By using our automated tool departments of universities can generate timetables in less time and also can minimize the resources like rooms etc. and prevent the clashes between different classes. Therefore, by using our automated tool optimized timetable can be generated for departments.
Implementation Details	
Link to Github Repository	https://github.com/zahidalidev/CS311S20PID25
Total Number of commits in repository	<p>Expected: 214 Actual: 111</p> <p>Sir as you can see in Screenshots of our github profile contribution activity, Abdul Samad (2018-CS-141) has done 72+42 commits (1st Screenshot) and Zahid Ali (2018-CS-136) has 5+92+3 commits (2nd Screenshot) but in 3rd Screenshot, you can see the total commits of the whole repository is 111. So sir we want to say that here we are encountered by some sort of conflict/problem in GitHub repository about of our commits that many commits of both of us have been lost and we have tried a lot but we are not able to get them back but sir we mostly remember our major commits that we have made about our project so we are going to explain them in details of commit section. Sir we did not have enough knowledge about github that how to work as collaborator. So we don't know what happened with us about our all commits.</p> <p>Afterwards from submitting the report without testing milestone, all of our commits are okay and we have come to know how to work as collaborator in github.</p> <p>...1</p>



...3

zahidalidev / CS311S20PID25
Private

<> Code
! Issues
↕ Pull requests
▶ Actions
📁 Projects
📖 Wiki
🛡 Security
📈 Insights

🔑 master ▾
🔗 1 branch
🏷 0 tags
Go to file
Add file ▾
↓ Code ▾

AbSamDev Merge pull request #7 from AbSamDev/master
d46d71a 12 hours ago
🕒 111 commits

📁 timeTableBackend	comments on whole project	12 days ago
📁 timetable_frontend	This fixes #1	12 hours ago
📄 AOAProjectReport.pdf	Added project report via upload	10 days ago
📄 MileStone3_Correctness_and_Comple...	Uploading milestone 3	last month
📄 README.md	removing an extra breal line tag	8 days ago
📄 milestone1.pdf	updating milestone1	11 days ago
📄 pseudocode.txt	comments on whole project	12 days ago

Commits in Github Repository By each member	2018-CS-141: 114 commits 2018-CS-136: 100 commits
--	--

Commits in github repository by each member

Member Registration No.	Total Commits
2018-CS-141	114
2018-CS-136	100

Details of commits

Sr. No.	Details of commit	Date	Member Reg No.
1	CheckCouesesModule: In this commit I have added the pseudo code of the check course function. I am checking that everything that course needs is available or not.	Jul 2, 2020	2018-CS-141
2	checkTeacher added: In this commit I have added the check teacher function. The function is checking the of availbility teacher for current course.	Jul 3, 2020	2018-CS-136
3	Uploading milestone 3: In this commit, I have uploaded milestone for complexity analysis and correctness of pseudo code.	Jul 8, 2020	2018-CS-141
4	Modified generate table function: In this commit I have updated the generate time table function that is handling the generation of time table with calling other functions like checkTeacher, checkCourses within iterative processes.	Jul 9, 2020	2018-CS-141
5	final table and updated generate table: In this commit, I have handled some problems in generating timetable and removed some bugs. It gives optimized time table on console.	Jul 10, 2020	2018-CS-136
6	final post req: In this commit, I have added the features to handle post requests from any frontend towards the backend.	Jul 13, 2020	2018-CS-136
7	pagination and routes handled: In this commit, I have added pagination bars and routes in the frontend application to visit and track different pages.	Jul 17, 2020	2018-CS-141
8	navigation bar and appBar done: In this commit have added app bar for the front-end interface.	Jul 18, 2020	2018-CS-141
9	integrated with backend: In this commit, I have integrated frontend with backend to send and receive data from backend.	Jul 18, 2020	2018-CS-136
10	unique input from all fields: In this	Jul 19, 2020	2018-CS-136

	commit, I have handled all input fields on different pages to handle inputs for timetable on front-end.		
11	responsive done: In this commit, I have made the application responsive to work well on mobile view.	Jul 19, 2020	2018-CS-136
12	new footer design and some modification: In this commit, I have changed the footer design to make it look better and also made some modification in other respective files.	Jul 25,2020 Check date	2018-CS-141
13	last 25: In this commit, I have finalized the whole project.	Jul 25,2020	2018-CS-136
Have you used built in algorithms or you have implemented yourself?	No, we have implemented this algorithm on our own.		
Formats of input	We have four pages for input data. In the first page, we are taking rooms name input as a text format in an input field and we can add more rooms name by clicking add more fields button and can remove fields in the similar way. In 2nd page, we are taking classes name as text input same as that of the input format in first page. In the 3rd page, we are taking teacher name as text and teacher available time slots on every hours of days as a select from 0 or 1 where 0 means availability of teacher and 1 means teacher is not available in that time slot. Finally, in 4rth page we are taking details of courses as course name and other details like credit hours, teachers name and classes name as select input from previously entered input data about them.		
Validations	<p>On input fields we have applied the following validations:</p> <p>Same value: We have applied this validation so user cannot add same room name. Without having a unique name user cannot navigate to the next pages because if the same name exists, the user will see the error notification and the next button will be disabled until the user enters the unique name in every input field in the same page.</p> <p>Minimum value: We have applied this validation so that user cannot delete the input field if there is only one input field left on that page. The user should add at least one name of the classroom, class, teacher and course to make progress.</p> <p>Empty field: Because of this validation, user cannot leave input fields empty. He will not be able to go forward because the next button will be disabled unless he enters values in all input fields.</p>		

	<p>Routes protected: This validation is applied so the user cannot navigate to next page until he fills the requirements of the previous page. There is no way to go forward until filling the requirements of the previous page.</p> <p>Again Generate Time Table: The generate table button will disappear if the time table is generated once so to again generate time table, user should go to home page to step 1 and fulfill all requirements</p>
Format of output	By clicking the generate table button, output will be shown in tables (row header contains times from 8AM to 4PM and column header contains days). In the same way, respective boxes contain the teacher's name, classroom name and course. The number of tables to be generated depends on the number of classes as each timetable shows showtime table of class.
Deployment	<p>Yes we have deployed our project here are the details:</p> <p>Backend (Nodejs):- We have deployed the backend of the application on Heroku (https://www.heroku.com/) so I can make requests from any frontend application to send the input data and can get the output data from there.</p> <p>FrontEnd (ReactJs):- We have deployed a frontend on cPanel that I got from namecheap shared hosting plan. We have used a sub-domain for this frontend. We have deployed this application after building it. Here is the link of application : http://az.mrfixer.pk/</p>

Details of algorithms

Pseudo Code:

//function to check the teacher for particular course in particular hourse is available

checkTeacher(currentCourse, i, j, cr)

if currentCourse.crHouurs[cr] == 0 or currentCourse.crHouurs.length == 0
return [false]

for t = 0 to teachers.length

if currentCourse.teacher == teachers[t][0]

//checking teacher for all credit hourse of course is available

tHour = j

for let l = 0 to currentCourse.crHouurs[cr]

if teachers[t][1][i][tHour] == 0

tHour++

else

return [false]

if l == currentCourse.crHouurs[cr] - 1

return [true, t] //returning index of teacher and true if teacher is available

```

checkClassRoom(currentCourse, i, j, cr, cl)
//checking class room for all credit hourse of course is available
clHour = j
for l = 0 to currentCourse.crHouurs[cr]
    if classRooms[cl][1][i][clHour] == 0
        clHour++
    else
        return false
if l == currentCourse.crHouurs[cr] - 1
    return true

checkCourse(currentCourse, i, j, cr, cls, cl)
count = 0
if currentCourse.crHouurs[cr] == 3 and j >= 6 //if redit of course is greater then available hourse
then course should not be allocate
    count--
if currentCourse.crHouurs[cr] == 2 and j >= 7 //if redit of course is greater then available hourse
then course should not be allocate
    count--
if classes[cls][0] == currentCourse.session //if class is available for current course
    count++
chTeacher = checkTeacher(currentCourse, i, j, cr)
if chTeacher[0] is true //if teacher is available making count increase
    count++
if checkClassRoom(currentCourse, i, j, cr, cl) //if class room available count increase
    count++
if count == 3 //if all resources available then return chTeacher with teacher index and true
    return chTeacher
return [false]

generateTimeTable()
cl = 0
while cl < classRooms.length
    for cls = 0 to < classes.length
        for i = 0 to daysPerWeek
            j = 1
            while j < hoursPerDay

                //if time table, class room and class mean all have time slots available
                if timeTable[i][j] == 0 and classRooms[cl][1][i][j] == 0 and classes[cls][1][i][j] == 0
                    y = 0
                    while y < courses.length //checking for courses on every hour of the day so that can be
alloted

```

```

        if j === hoursPerDay
            j = 1
        chCourse = checkCourse(courses[y], i, j, 0, cls, cl) //if course is available mean
course have teacher for all its current credit hours

        if chCourse[0] is true
            for m = 0 to courses[y].crHours[0] //making busy to all slots until credit
hour and allocating course
                timeTable[i][j] = courses[y].name + ', ' + courses[y].teacher + ', ' +
classRooms[cl][0]
                classRooms[cl][1][i][j] = 1
                classes[cls][1][i][j] = 1
                teachers[chCourse[1]][1][i][j] = 1
                j++

            courses[y].crHours.splice(0, 1); //removing credit that are used

            if courses[y].crHours.length === 0
                courses.splice(y, 1) //removing course if its all its credit hours is used so
length of course array will be less by 1
            else
                y++
            j++
        else
            j++;

//pushing current session table to allTable array and making him empty
let table[k] = timeTable[k]
for ta = 0 to allTables.length or allTables.length === 0
    if allTables.length !== 0
        if classes[cls][0] === allTables[ta][0]
            for dpw = 0 to dpw < daysPerWeek
                for hpd = 1 to hoursPerDay
                    if allTables[ta][2][dpw][hpd] === 0 and table[dpw][hpd] !== 0 //if allTable slots is
free and time table slot is busy then copy that slot to allTable
                        allTables[ta][2][dpw][hpd] = table[dpw][hpd]

            else if allTables[ta][1] === classRooms[cl][0]
                count = 0
                for s = 0 to allTables.length
                    if allTables[s][0] === classes[cls][0]
                        count++

            if count === 0
                allTables.push([classes[cls][0], classRooms[cl][0], table]) //pushing current room

```

name, class name and table to all table array

```

        else
            allTables.push([classes[cls][0], classRooms[cl][0], table])    //pushing current room name,
class name and table to all table array

//freeing timrTable for furthor use
for o = 0 to < daysPerWeek
    for p = 1 to hoursPerDay
        timeTable[o][p] = 0
    cl++

```

Description:

checkTeacher(): In this function, we are checking that the teacher of the current course is available for all the consecutive credit hours and result returns false if not available and it returns true with the index of teacher if teacher is available for all consecutive credit hours.

checkClassRoom(): In this function we are checking that the room is available for current course consecutive credit hours. It simply returns true or false.

checkCourse(): In this function we are counting the results of above two functions and also the hours of days with credit hours. It is because of the reason that there may occur some circumstances where we have current course of consecutive 3 credit hours and the hours in a day we have left are three so we don't want include that course here in the time table so the counter in this function will be decrease and we will return false.

generateTimeTable(): This is the major function where all things of the code are managed. Here we start with the rooms loop (main boundary) and inside it we have days loop, and hours loop in a nested manner. Here in this loop (hours loop) we are checking for all available slots with classroom and class name, that each course for current hours to be included in the timetable. The course will be punched in the timetable array if everything is available. If the available hours of the current day are more than or equal to the consecutive credit hours of the course checking with rooms and classes whose course is also available then we push this course to timetable array. And if all credit hours of this course are managed and filled, we will delete this course from course array, if current day is not available with something like teacher, hours of day or class, then we go for the next day if room is not available after checking everything, we will use another room for that course.

Correctness:-

Initialization:-

The function is initialized by checking that we have available slots of classRooms, classes (like sessions 17,18, 19), timeTable and the course we are dealing with and the teachers available in those time slots.

Maintenance:-

If we have all slots available that we have mentioned above then generateTable function will start

inserting the courses iteratively into the timeTable according to their creditHours and if all credit hours are been used for that course then the function will delete that course specific from the array of courses in the respective iteration. One iteration will work keeping in view the upper limits and insert the course in respective positions.

Termination:-

The function will be terminated when all slots of all classRooms are used or no other teacher is available or all courses are used which means the courses list is empty.

Complexity:-

The program has the main function as Generate timeTable which is further using sub modules as checkTeacher and checkCourse. An instance of the generateTable problem is a four-tuple (cl, t.len, y.len, i*j)

where

cl is the number of classRooms,

t.len is the number of teachers

y.len is the number of courses


And timeTable i*j denoting the number of courses(y) taught by teacher t to class cls.(i is the number days and j is number of hours in ith day)

with 'y' denoting the number of lessons taught by teacher 't' to class 'cl'. The time slot assignment (subject+teacher positioning) in the timetable is modeled by a set of variables (cls, t, y), (for cls = 1.....cl; t = 1.....t.length; y= 1.....y.length) with (cls, t, y) = 1 if a lesson for class 'cls' taught by teacher 't' is taught at specific time slot, and (cls, t, y) = 0 otherwise.

Interfaces for your project

1- Add Classrooms:-

In this page, we are taking input as a text format in an input field, which is the name of the classroom, and we can add more room names by just clicking add more button and remove the field similarly. We cannot move forward unless we enter at least one classroom or unless each text field does not have unique names (otherwise submit and next button will be disabled).

0% 

1- ADD CLASS ROOMS 2- ADD CLASSES 3- ADD TEACHERS 4- ADD COURSES 5- TIME TABLE

Room-1 Name

Room-2 Name

Room-3 Name

2-

Add Classes:-

In this page, we are taking input as a text format in an input field, which is the name of the class, and we can add more room names by just clicking add more button. We cannot move forward unless we enter at least one class or unless each text field does not have unique name (otherwise submit and next button will be disabled). We can also remove the text field by clicking the cross button.

25%

1- ADD CLASS ROOMS 2- ADD CLASSES 3- ADD TEACHERS 4- ADD COURSES 5- TIME TABLE

Class-1 Name eg. Session18
Session 17

Class-2 Name eg. Session18
Session 18

Class-3 Name eg. Session18
Session 19

ADD MORE

SUBMIT AND NEXT >

3- Add Teachers:-

In this page we are taking teacher name as text and also teacher available time slots on every hours of days as a select from 0 or 1 where 0 means availability of teacher and 1 means teacher is not available in that time slot.

50%

1- ADD CLASS ROOMS 2- ADD CLASSES 3- ADD TEACHERS 4- ADD COURSES 5- TIME TABLE

*Select "0" if teacher is available and "1" if teacher is busy in below time slots

Teacher-1 Name
Sir Irfan

Monday	1	1	1	1	1	1	1	1
	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
Tuesday	0	0	0	0	1	1	1	1
	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
Wednesday	1	1	1	1	1	1	1	1
	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
Thursday	1	0	0	1	1	1	0	0
	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
Friday	1	1	1	1	1	1	0	0
	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM

ADD MORE

4- Add Courses:-

In this page, we are taking details of courses as course name, its credit hours as select input and other details like teacher name and class name as a select input from previously entered input data about them.

75%

1- ADD CLASS ROOMS 2- ADD CLASSES 3- ADD TEACHERS 4- ADD COURSES 5- TIME TABLE

X	Course-1 Name MVC	Session 17 Select Class	Sir Irfan (DOM) Select Teacher	1, 1 Select Credit Hour
X	Course-2 Name PSP	Session 18 Select Class	Sir Irfan (DOM) Select Teacher	1, 1 Select Credit Hour
X	Course-3 Name Calculus	Session 19 Select Class	Sir Irfan (DOM) Select Teacher	1, 1 Select Credit Hour

ADD MORE

SUBMIT AND NEXT ➤

5- Generate Timetable:-

In this page we just need to click on generate timetable button to be directed towards class timetables.

100%

1- ADD CLASS ROOMS 2- ADD CLASSES 3- ADD TEACHERS 4- ADD COURSES 5- TIME TABLE

GENERATE TABLE

6- Generated Tables:-

The generated timetables will be equal to the number of classes entered in the 2nd page (Add Class page). Each table will show timetable of one class. There is also a download button below the timetable where we can download the timetable of the classes we want.

Class Name: **Session 18**

DAYS	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
MONDAY	PPSD, Sir Hassan (DSE), N6	0	COAL, Sir Tayyab (DOCS), N6	PPSD, Sir Hassan (DSE), N6	0	0	0	0
TUESDAY	OOP, Sir Ali (DCS), N6	OOP, Sir Ali (DCS), N6	PPSD, Sir Hassan (DSE), N6	0	0	FE, Sir Imran (DOE), N5	0	0
WEDNESDAY	COAL lab, Sir Fakhar (DOCSL), N6	COAL lab, Sir Fakhar (DOCSL), N6	COAL lab, Sir Fakhar (DOCSL), N6	0	0	0	COAL, Sir Tayyab (DOCS), N6	COAL, Sir Tayyab (DOCS), N6
THURSDAY	0	0	PSP, Sir Irfan (DOM), N5	0	0	FE, Sir Imran (DOE), N5	0	PSP, Sir Irfan (DOM), N5
FRIDAY	OOP lab, Sir Farooq (DCSL), N5	OOP lab, Sir Farooq (DCSL), N5	OOP lab, Sir Farooq (DCSL), N5	0	0	0	0	0

[PDF OF SESSION 18-SLOW](#)
[IMG OF SESSION 18-FAST](#)
Class Name: **Session 17**

DAYS	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
MONDAY	TWPS, Sir Imran (DOE), N5	TWPS, Sir Imran (DOE), N5	TWPS, Sir Imran (DOE), N5	OS, Sir Tayyab (DOCS), N5	0	0	0	0
TUESDAY	MVC, Sir Irfan (DOM), N5	MVC, Sir Irfan (DOM), N5	OS, Sir Tayyab (DOCS), N5	DSA, Sir Ali (DCS), N5	0	0	SE, Sir Hassan (DSE), N5	SE, Sir Hassan (DSE), N5
WEDNESDAY	DSA lab, Sir Farooq (DCSL), N5	DSA lab, Sir Farooq (DCSL), N5	DSA lab, Sir Farooq (DCSL), N5	0	0	0	CS17, Sir Tufail (DOH), N5	CS17, Sir Tufail (DOH), N5
THURSDAY	DSA, Sir Ali (DCS), N5	OS, Sir Tayyab (DOCS), N5	0	DSA, Sir Ali (DCS), N5	0	0	0	0
FRIDAY	0	0	0	0	0	0	0	0

[PDF OF SESSION 17-SLOW](#)
[IMG OF SESSION 17-FAST](#)
Class Name: **Session 19**

DAYS	8-9AM	9-10AM	10-11AM	11-12AM	12-1PM	1-2PM	2-3PM	3-4PM
MONDAY	0	ITC, Sir Tayyab (DOCS), N6	SD, Sir Hassan (DSE), N7	0	0	0	0	0
TUESDAY	0	0	0	BE, Sir Imran (DOE), N6	0	0	0	0
WEDNESDAY	PF, Sir Ali (DCS), N7	PF, Sir Ali (DCS), N7	SD, Sir Hassan (DSE), N7	0	0	0	0	0
THURSDAY	ITC, Sir Tayyab (DOCS), N6	SD, Sir Hassan (DSE), N6	0	ITC, Sir Tayyab (DOCS), N6	0	0	Calculus, Sir Irfan (DOM), N5	0
FRIDAY	BE, Sir Imran (DOE), N6	BE, Sir Imran (DOE), N6	PF, Sir Ali (DCS), N6	0	0	0	Calculus, Sir Irfan (DOM), N5	0

[PDF OF SESSION 19-SLOW](#)
[IMG OF SESSION 19-FAST](#)
Integration

We have used express in nodeJS to accept the post request from the frontend application and receive the input data in JSON format and as a response to that request, we are sending the result as a generated timetable in JSON format.

In frontend, we have used the axios library to send http post requests with all data collected from input fields and receiving response as a generated timetable.

While integration, we have faced one problem and that is our backend was not accepting the request from the frontend application but while sending requests using Postman is working exactly fine as expected to handle this problem. We have used cors in express app.use() function to accept requests from any url. We had come to the solution while searching about it on stackoverflow and this worked fine for us.

Change Requests

Yes, we have requested changes for 2 milestones after the deadline and these changes are; we have updated pseudo code and we have updated demo user interface of the application.

Testing

In this section, you are required to mention the issues report and solution proposed.

Issue#2: Issues found in the responsiveness of GUI:

In this issue, we had the problem of CSS overflow in output (generated table) when we input too many characters in any input field so to resolve this issue we have added css styling in the div of table tag that is parent of the table tag and also added the max width property to limit the width of the table. The css style that we added in the div element: `style={{overflow:"auto", maxWidth: "1130px",` and to make this table responsive, we have added this property in the media query that is in the tableDesign.css: `max-width: 240px !important;` after adding above properties we have got a scrollable output generated table even if we add too many characters the table will remain responsive.

Issue#3: constraint for blank space:

In this issue, if we add only white space in the input table then the submit and next button will be enabled and we can go to next page of input fields so at the end we can generate table and the table will be contain empty fields apparently, so to resolve this issue we have added constraints to prevent the only white space character in the input fields like we are comparing the length of character with number of spaces in and if they are equal then we are showing the toast error and also disabling the button. The method we have made and used in that component is:

```
// handling the input fields with white space
checkEmptyField = (Array) => {
}
```

We have called this method in handleChange, addMoreFields and handleRemove methods. This method has major use in handleChange method as it checks that specific textfield as a whole whereas we have used this method in addMoreFields and handleRemove methods to check it whenever add more field button or remove field button is pressed.

Issue#4: Error in response from the Server

In this issue if we have large input fields (means we have too many characters in the input fields), then there will be an error because of our cheap servers. So to resolve this issue, we have added the constraints to limit the length of input fields because of this if we try to add more characters than the required string, we will receive an error and also the submit and next button will be disabled. So to navigate to the next page, we have to enter characters in the input fields under the limit that we have defined.

```
// handling the maximum length of input field
```

```
  maxLengthConstraint = (Array) => {  
    }
```

Similar to that of previous issue, We have called this method in handleChange, addMoreFields and handleRemove methods. If we do not meet the constraint requirement, we cannot submit current form and navigate to next unless we minimize the characters down to the defined limit.

Issue#1: No Issue

In this issue, we were required to make the front-end more attractive. To solve this issue we have modify the front-end a lot and make it more attractive for the users.

<i>Technology</i>	
Programming Language	JavaScript [backend: nodeJS; frontend: reactJS]
Platform	[Web Application]