Analysis of Algorithms

Spring 2020

Members Details

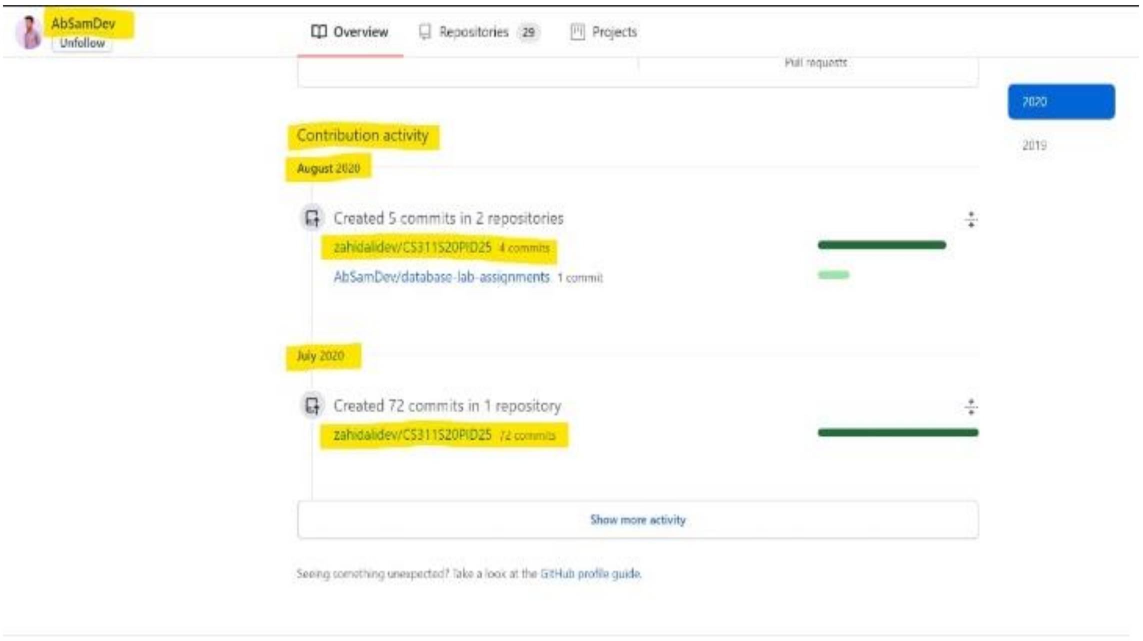
Group ID	CS311S20PID25
Registration Number of Group Members	2018-CS-136 [Zahid Ali] 2018-CS-141 [Abdul Samad]
Section	C

Project Details

Project	
Project Title	Activity Scheduling
Executive Summary	A clear description of the project. At least 300 words long] 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 for a department. This is automated scheduling tool for courses of department. This project reduce the resources used in department like this will use minimum class rooms for lectures and also minimum days for lectures, by building time table using this automated tool we can minimize resources and can maximize courses for classes in the output time table. Teachers and classes are 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 then classes are added as input in the 2nd step. Classroom names and classes should be unique otherwise; the tool will not let us proceed. After that, we proceed to next step means 3rd here we have to add information about teachers like teacher name and teacher time slots (i.e., at which time slot teacher is available during 8am to 4pm). We need to input time slots for teachers because teachers can be busy in other department. So 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 finally selecting the credit hours for each course like if we want them to have 3 consecutive lectures, we have to select 3 or if 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 takes the user to the time table page where user can generate time table for all classes of department by just clicking the generate time table button and at this page,

	we can also download generated timetables as an image.
Business Case	
Outline the business need for the project	Manually creating optimized time table is a time consuming task for all departments of universities. If they make time table manually, there will occur conflicts and clashes between different teachers and classes. And also there are free time slots available in different class rooms that can be used so we need more class rooms to manage everything. By using this automated time table generator tools the conflicts or clash between different teachers and classes are 0%. And also using minimum room we can manage more courses.
End user of the product	As this automated time table generating tool is related to departments of universities, so we are going to target the departments that build time table manually with this automated tool they can generate optimized time table.
Motivation for Project	We are motivated to build this project because University students face problems about time table 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. With this project that we have built, if we generate a time table then we 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 time table generating tool is used to build optimized time table for a university. This project can minimize the use of resources like minimzes 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 till 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 Requirement	The features of our projects are minimizing the resources like rooms etc. and managing the courses in the optimized way, and generating timetable by taking simple inputs

s	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 time table can be generated for departments.
Implementation Details	
Link to Github Repository	https://github.com/zahidalidev/CS311S20PID25
Total Number of commits in repository	Expected: 172 Actual: 72 Sir as you can see in Screenshots of our github profile contribution activity, Abdul Samad (2018-CS-141) has done 76 commits (1st Screenshot) and Zahid Ali (2018-CS-136) has 96 commits (2nd Screenshot) but in 3rd Screenshot, you can see the total commits of the whole repository is 72. So sir we want to say that here we have some kind of conflict 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 have not enough knowledge about github so we don't know what happened with us about our all commits. …1



AbSamDev
Unfollow

Contribution activity

August 2020

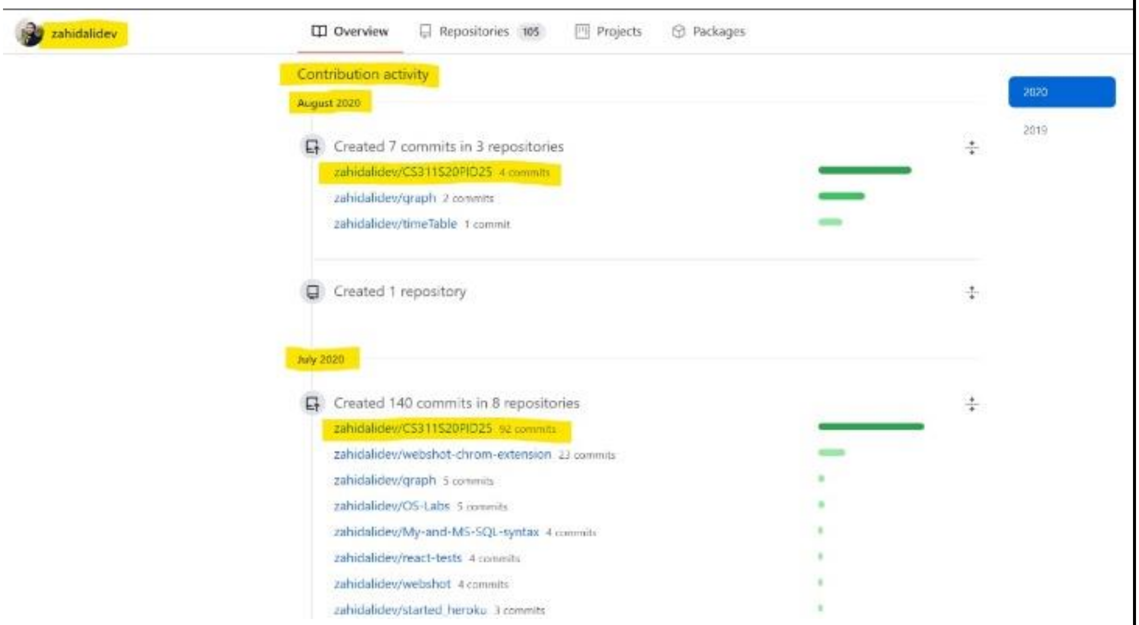
Created 5 commits in 2 repositories
zahidalidev/CS311S20PID25 4 commits
AbSamDev/database-lab-assignments 1 commit

July 2020

Created 72 commits in 1 repository
zahidalidev/CS311S20PID25 72 commits

Seeing something unexpected? Take a look at the GitHub profile guide.

…2



zahidalidev

Contribution activity

August 2020

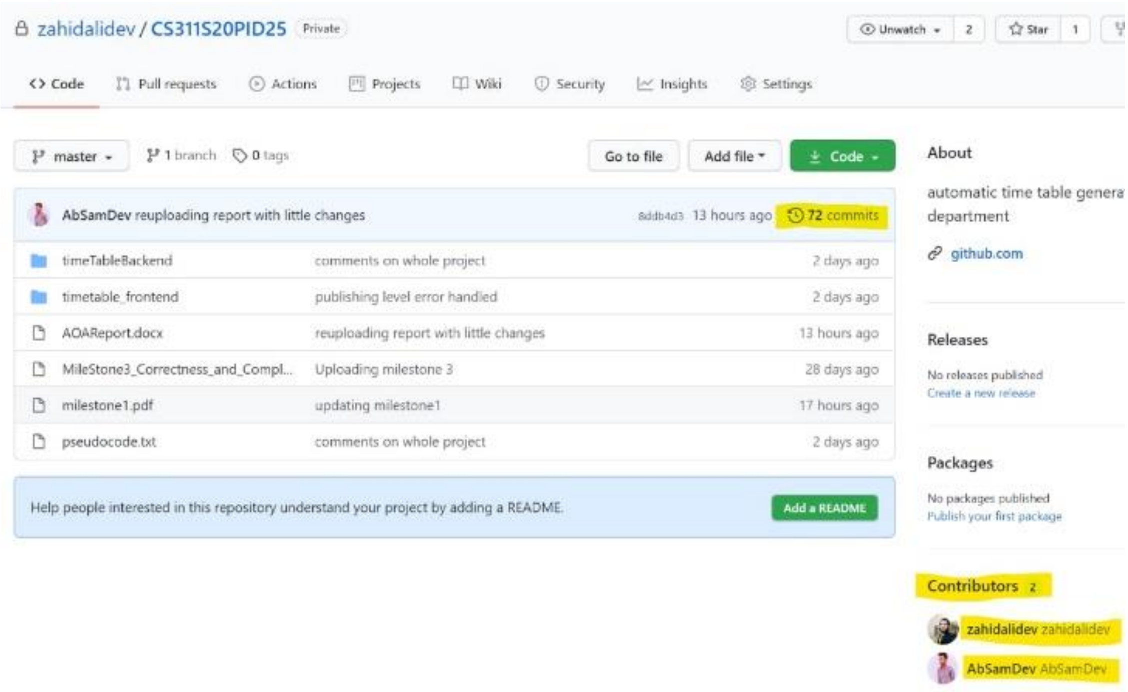
Created 7 commits in 3 repositories
zahidalidev/CS311S20PID25 4 commits
zahidalidev/graph 2 commits
zahidalidev/timeTable 1 commit

Created 1 repository

July 2020

Created 140 commits in 8 repositories
zahidalidev/CS311S20PID25 92 commits
zahidalidev/webshot-chrom-extension 23 commits
zahidalidev/qraph 5 commits
zahidalidev/OS-Labs 5 commits
zahidalidev/My-and-MS-SQL-syntax 4 commits
zahidalidev/react-tests 4 commits
zahidalidev/webshot 4 commits
zahidalidev/started_heroku 3 commits

…3



AbSamDev reuploading report with little changes 8ddb4d3 13 hours ago 72 commits

timeTableBackend	comments on whole project	2 days ago
timetable_frontend	publishing level error handled	2 days ago
AOAReport.docx	reuploading report with little changes	13 hours ago
MileStone3_Correctness_and_Compl...	Uploading milestone 3	28 days ago
milestone1.pdf	updating milestone1	17 hours ago
pseudocode.txt	comments on whole project	2 days ago

Help people interested in this repository understand your project by adding a README. Add a README

About

automatic time table genera department

github.com

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

zahidalidev zahidalidev
AbSamDev AbSamDev

Commits in Github Repository By each member	2018-CS-141: 76 commits 2018-CS-136: 96 commits

Commits in github repository by each member

Member Registration No.	**Total Commits**
2018-CS-141	76
2018-CS-136	96

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	Jul 3, 2020	2018-CS-136

	teacher function. The function is checking the of availbility teacher for current course.		
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 commit, I have handled all input fields on different pages to handle inputs for timetable on front-end.	Jul 19, 2020	2018-CS-136
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	Jul 25,2020 Check date	2018-CS-141

	changed the footer design to make it look better and also made some modification in other respective files.		
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?	We have implemented this algorithm on our own.
Formats of input	We have four pages for input data. In first page we are taking input as a text format in input field that is the name of classrooms and we can add more rooms name by clicking add more fields button. In 2nd page, we are taking classes name as text input same as that of the input format in first page. In 3rd 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. Finally, in 4rth 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 input data about them.
Validations	On input fields we have applied following validations: **Same value:** We have applied this validation so user cannot add same room name. Class name and teacher name without having a unique name user cannot goto next page because if the same name exists then 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. **Minimum value:** We have applied this validation so the user cannot delete the input field if there is only one input field left on one page so the user should add one name of class room , class and teacher and course to go ahead. **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 until he enters values in all input fields. Routes protected: This validation is applied so the user cannot go next until he fills the requirements of the previous page there is no way to go forward until filling the previous page requirements. **Again Generate Time Table:** The generate table button will disappear if the time table is generated once so to again generate time table he should go to home page to step 1 and fulfill all requirements

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 1 class.
Deployment	Yes we have deployed or project here are the details: **Backend (Nodejs):-** I 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. **FrontEnd (ReactJs):-** I have deployed a frontend on cPanel that I got from namecheap shared hosting plan. I have used a subdomain for this frontend. I have deployed this application after building it. Here is the link of application : http://az.mrfixer.pk/

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 avaible mean course have teacehr for all its current credit hourse

              if chCourse[0] is true
                for m = 0 to courses[y].crHouurs[0]     //making bussy to all slots until credit hourse 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].crHouurs.splice(0, 1);        //removing credit that are used

                if courses[y].crHouurs.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 ant 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 buusy 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 consecutive credit hours and a result return false if not available and true with the index of teacher if teacher is available for all consecutive credit hours.
checkClassRoom(): In this function we are checking that rooms is available for current course consecutive credit hours and returning true or false.

checkCourse(): In this function we are counting the result of above two functions and also the horse of days with credit hours because there maybe circumstances that 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 main function here we start with the rooms loop inside that is days loop, inside that is horse loop then here in this loop we are checking each course for current horse to be include in the time table, the course will be punched in the time table array if everything is available like if teacher is available, the available horse of the current day are more than or equal to the consecutive credit hours of the course and also room and the class whose course is this is also available then we push this course to time table array and if all credit horse of this course are managed then we will delete this course from course array.if current day have not available something like teacher, horse of day or class then we check for the next day and if room is not available after checking everything then 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:-

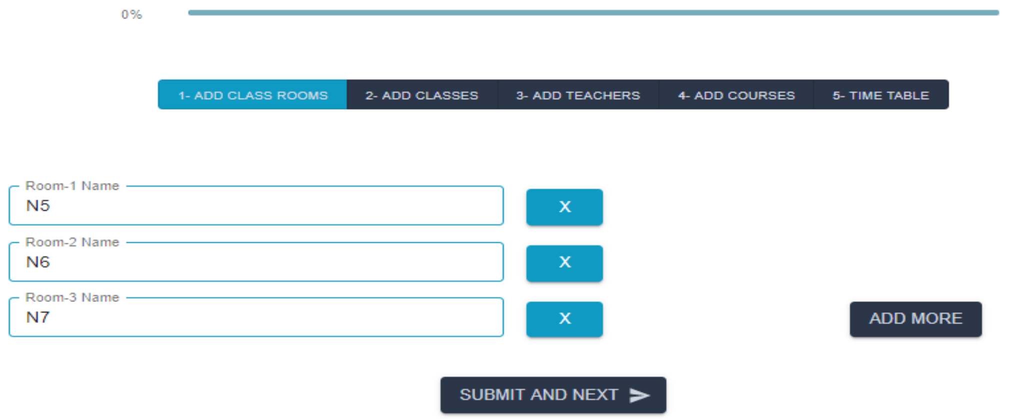
The program has the main function as Generate timeTable which is further using sub modules as checkTeacher and checkCourse. An instance of the generateTable problem is a four-tuple (cl, t.len,

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:-

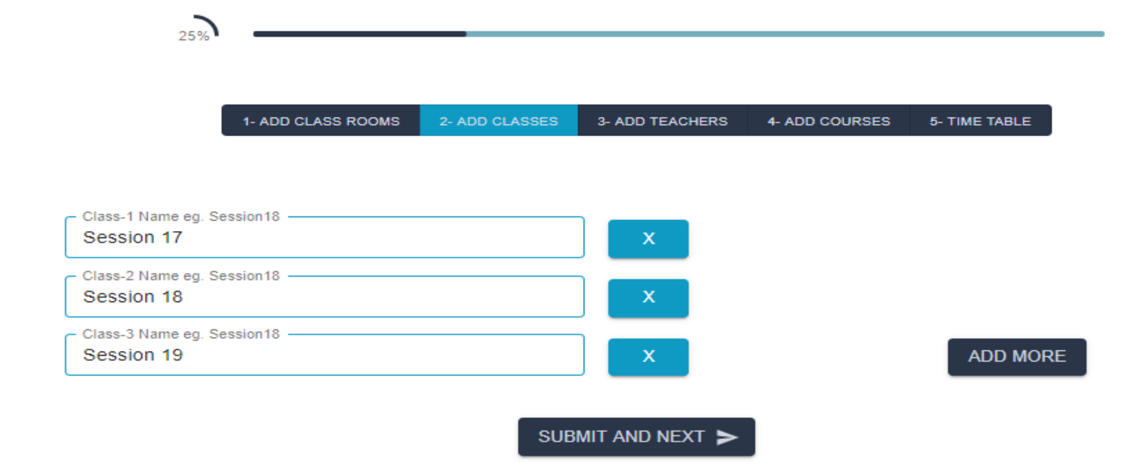
In this page, we are taking input as a text format in an input field which is the name of the classroom and we can add more room names by just clicking add more button. We cannot move forward unless we enter at least one classroom or unless each text field does not have unique name (otherwise submit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



Room-1 Name: N5
Room-2 Name: N6
Room-3 Name: N7

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 delete the text field by clicking the cross button.



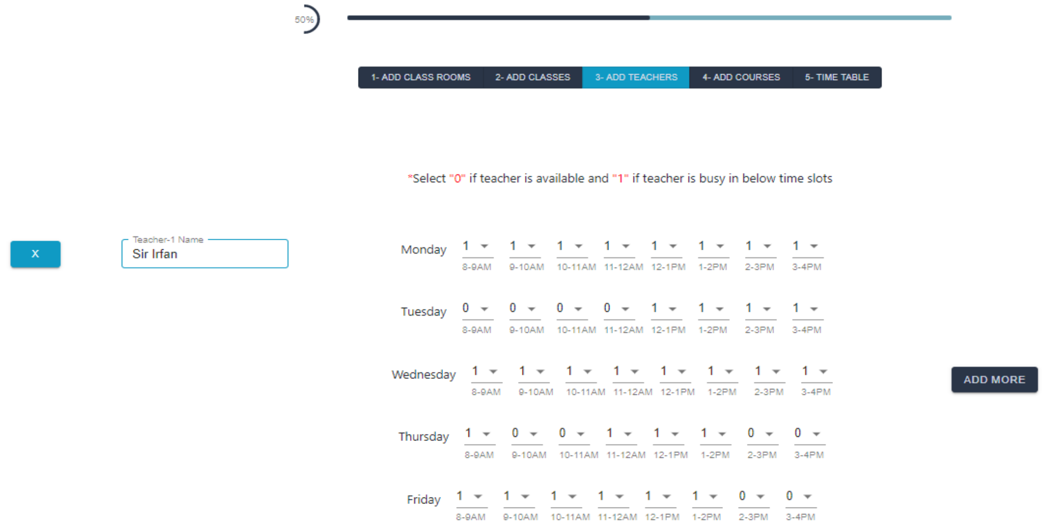
Class-1 Name eg. Session18: Session 17

Class-2 Name eg. Session18: Session 18

Class-3 Name eg. Session18: Session 19

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.



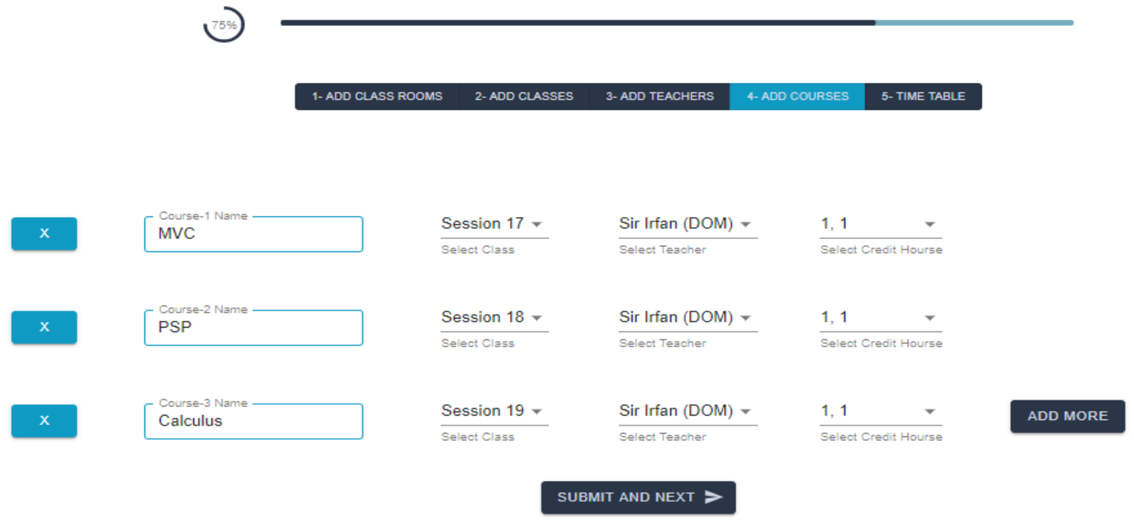
*Select "0" if teacher is available and "1" if teacher is busy in below time slots

Teacher-1 Name: Sir Irfan

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

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 input data about them.



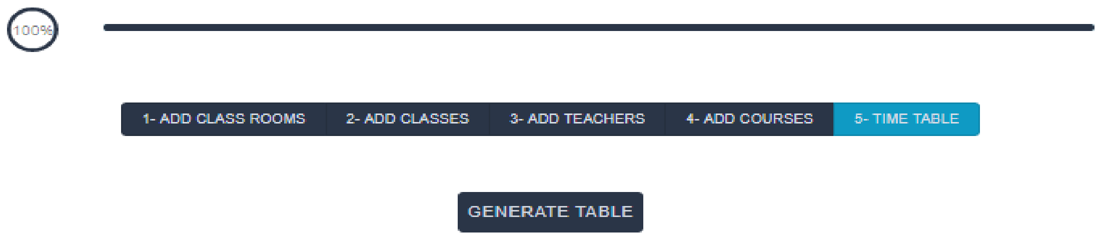
Course-1 Name: MVC — Session 17 (Select Class) — Sir Irfan (DOM) (Select Teacher) — 1, 1 (Select Credit Hourse)

Course-2 Name: PSP — Session 18 (Select Class) — Sir Irfan (DOM) (Select Teacher) — 1, 1 (Select Credit Hourse)

Course-3 Name: Calculus — Session 19 (Select Class) — Sir Irfan (DOM) (Select Teacher) — 1, 1 (Select Credit Hourse)

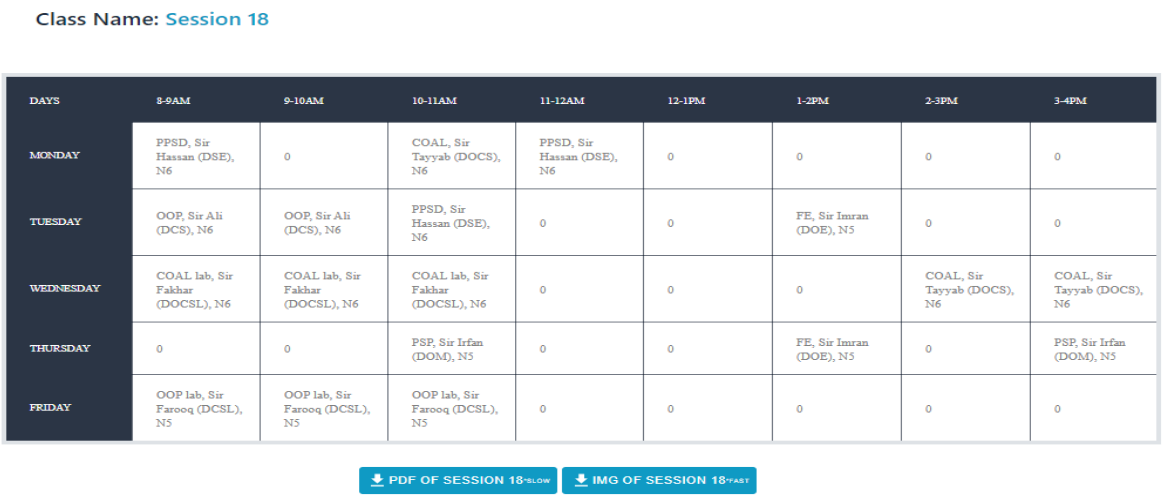
5- Generate Timetable:-

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



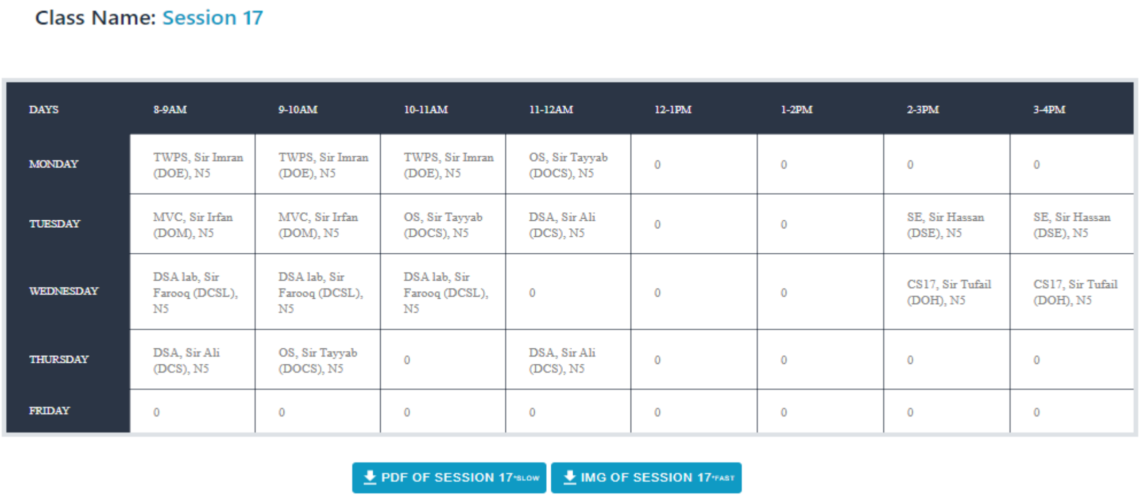
6- Generated Tables:-

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.



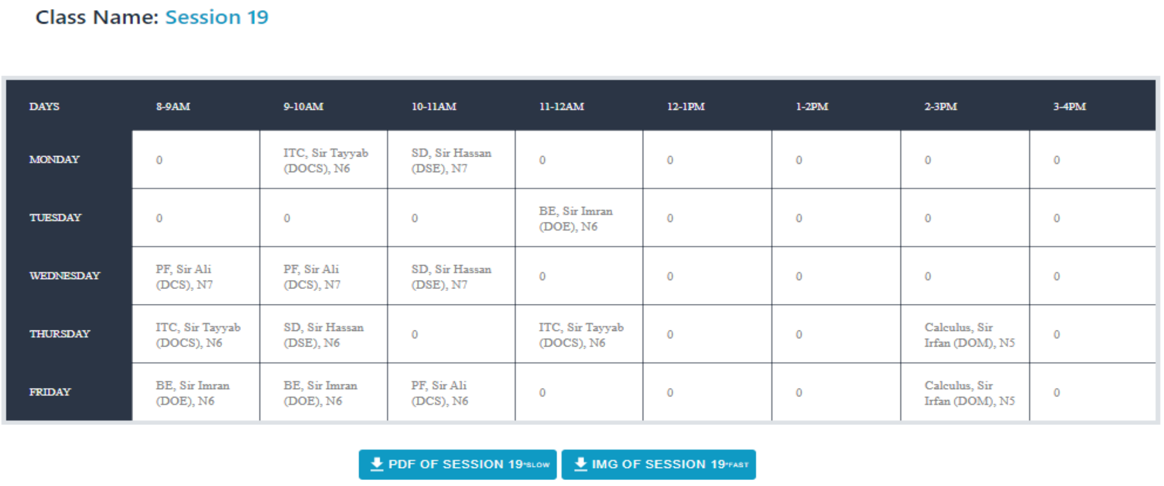
Class Name: Session 18

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



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



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

Integration

We have used express in node js 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 time table.

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 got this solution while searching about it from stackoverflow and this worked fine for us.

Change Requests

Yes, we 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.

Technology	
Programming Language	JavaScript [backend: nodeJS; frontend: reactJS]
Platform	[Web Application]