

(1) Admin :-

(1.1) Admin `adminPage.h` :-

① Function name :- `addNewStudent()`

- this function creates a new object with type "Student" using the new keyword and saves the pointer in newStudent variable.

- When all details are asked this student data is saved in an array in backend class that stores all the students in the system. So we find the current index which is empty and save the data there.

(1.2) `adminPage.h`

① Function name :- `removeStudentFromSystem()`

- this function first shows all students to the admin using the `showAllStudents()` function.

- Then if there are students it asks for the index of the student that needs to be removed.

- If selection is correct. it ~~overites~~ overwrites the student at the selected index by the one in the front and so on.

- Index of ~~total~~ student length is decreased to keep track of the total students.

(1.3) adminPage.h

- ① Function Names :- updateStudentDetails (int, int)
 ② updateStudentValue (int, Student*)

update ① :- This function first displays all the students, asks for the student index that we want to edit and loads that student into memory in the variable "std".

It then asks for the value to edit in the Student object.

- ② Now the "updateStudentValue" function takes ~~the~~ the index of the value the admin wants to change and performs respective operations and input from the ~~user~~ admin to ~~over~~ overwrite the value.

(1.4) adminPage.h

- ① Function Name :- showStudentDetails (int)

- This function just takes the index of the student of which we want to show details of. This index is used to access the currstudents array in the backend class.

This function will also send the ~~first~~ current student (Student) pointer when called.

(1.5) adminPage.h

- ① Function Name :- unenrollAstudentFromCourse ()

- This function asks the user to enter course index from which a student needs to be removed. When course is selected

2022029

3

we just show all the students in the course and then ask for the index of the student that we want enrolled.

(1.6) adminpage.h

① Function Name :- addNewTeacher()

- The addNewTeacher function creates a new Teacher object named newTeacher with the new keyword and the pointer newTeacher.
- We then ask for all the values of teacher from the admin admin and save it in the newly created Teacher object.
- In the end we get the teachers array which is in the backend.h and "Backend" struct. we get the final available index and save the values of this teacher there.

(1.7) adminPage.h

① Function Name :- removeTeacherFromSystem()

- This function first shows all the teachers with the showAllTeachers(). function.
- Then checks if there are any teachers in the by backend.teacherlength 70.
- Then we ask for the index of a teacher that we want to delete.
- We get the index and remove the teacher by overwriting the data of the teacher by the next one and ~~so reduce the~~ reduce the teacher index to keep track of the teachers.

(1.8)

adminPage.h

① Function Name :-

~~updateTeacherDetails()~~

updateTeacherDetails(int, int)

- So this function takes two int one checks if selection is valid and the other checks if the screen should be cleared.

- First we show all the teachers to the admin with the "showAllTeachers(0)" function.

- Then we get the index of the teacher whose value we want to change.

- Then with this index we change all the values of the teacher at index-1 position at the teachers array.

(1.9)

adminPage.h

① Function Name :- showAllTeachers(int)

- This function has an int argument that we use to check if the screen needs to be cleared or not as we want to make this code reusable.

- We just go through all the teachers stored in the currentTeachers array in the Backend structure.

(1.10) adminPage.h

① Function Name:- add Course To System ()

- Similar to add teacher and student function.
- we just store the newCourse in the courses current Courses array in Backend structure.
- Similar to (1.1).

(1.11) adminPage.h

① Function Name:- remove Course From System ()

Similar to 1.2.

(1.12) adminPage.h

① Function Name:- update Course Details ()

Similar to 1.3.

(1.13) ~~adminPage.h~~ backEnd.h

① Function Name:- ~~show All~~ display Courses ()

(1.14) adminPage.h

① Function Name:- a loc Course ()

- The a loc Course first clears the screen.
- Then it displays all the courses so we can see their indexes.
- we then asks the admin to select the course to which we want to assign the course.

- After that course is selected then we then show all the teachers and tell the admin to select the index of teacher that he wants to assign to the course.
- After index is selected we make the teacher in the course (selected) object equal to the current Teachers [index].

(1.15) adminPage.h

① Function Name:-

deAllocCourse → Select Course

↓
Allocate empty teacher object to the course Teacher member,

(2) Student:-

(2.1) studenPage.h

Function Name:-

checkDetails()

- just checks the logged in student and access its members and displays them.

(2.2) | StudentPage.h

Function Name:- Enroll in

enroll In Course ()

- Displays all courses.
- Gets index from student of course.
- Assigns in that course.

(2.3) | StudentPage.h

Function Name:- ss | see Grades ()

- Go through all courses.
- Check where roll number matches.
- Shows the grades of that course where the roll number matches.

(2.4) | StudentPage.h

Function Name:-

displayCourseTeachers ()

- Go through all courses.
- Check where roll number matches.
- Display teacher of that course.

(2.5) | StudentPage.h

Function Name:- showCompleted Courses ()

- Go through all courses.
- Check where roll number matches in enrolled student array, the completed courses array, and
- Display that course name.

(2.6) Student Page.h

Function Name :- Show Dropped Courses()

Similar to 2.5 just checks the ~~dropped~~ dropped students array in course object.

(2.7) StudentPage.h

Function Name :- Show Completed Courses()

- Similar to (2.5) and (2.6) just checks the ~~complect~~ completed students to array.

(3) Teacher

(3.1) Teacher Page.h

Function Name :- Check Teacher Details()

Similar to 2.1 but displays the details of logged in teacher. like name, city etc.

(3.2) Teacher Page.h

Function Name :- Show Assigned Courses()

- Go through all courses
- Output where object matches. (Course name and code).

(3.3) Teacher Page.h

Function Name :- Show Students In Course()

- Get all courses allotted to teacher.
- In allotted courses go through each course and output the students in enrolled students array.

(3.4) teacherPage.h

② Function Name :- gradeStudents()

(3.5) teacherPage.h

② Function Name :-
seeTopStudentInCourse()

- Go through all students in all allotted course.
- Check if ~~Max~~ is ~~g~~ marks are greater than highestMarks and allot that student as highestMarks student. In end output topStudent roll number and variable.

3.6 teacherPage.h

② Function Name :- seeStudentsByGrades()

- Sort all graded students in array.
- Output their marks in decending order.