**ASSIGNMENT COVER SHEET**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course: BSc Computing (SE)** | | | | **Year: 2** | | CSY2027 | | |
| **Group Project** | | **Title: The Design and Development of a Course Management Software System** | | | | | | |
| Date due out: | Date due in: | | Extension date: | | | | | Extension agreed by: |
| **Student Names (List each member of the group)** | | | | | | | **Tutor:** | |
| Student comment, specific request for feedback etc. | | | | | Marker’s General View of the work | | | |

**ASSESSMENT FEEDBACK:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RATING SCALE** | **Excellent** | | **Good** | | **Satisfactory** | **Needs some more work** | | **Needs much more work** |
| Elicitation Plan/ Interview(s)/Findings (10%) |  | |  | |  |  | |  |
| Requirement Specifications Documentation  (10%) |  | |  | |  |  | |  |
| System Design Documentation  (25%) |  | |  | |  |  | |  |
| Prototype Functionality and Quality of Application Code (35%) |  | |  | |  |  | |  |
| Test/Evaluation Strategy  (10%) |  | |  | |  |  | |  |
| Group Cohesion, Teamwork and Project Management/ System Presentation (10%) |  | |  | |  |  | |  |
| Specific aspects of your assignment that the marker likes: | | | | Specific aspects of your assignment that need more work: | | | | |
| Tutor’s Signature: | | Date: | | | | | Grade: | |

**Table of contents:**

[Group Project 1](#_Toc18812)

[1 Introduction 8](#_Toc19298)

[1.1 Project Background 8](#_Toc18783)

[1.2 Project Aims and Objectives 8](#_Toc7815)

[1.3 Project Development Methodology 9](#_Toc25382)

[2 Requirement Engineering 9](#_Toc29874)

[2.1 Elicitation Activities 9](#_Toc24255)

[2.1.1 Interview Plans 9](#_Toc19896)

[2.1.2 Interview findings 11](#_Toc7651)

[2.1.2.1 Interview - Simon White 11](#_Toc10450)

[2.1.2.2 Interview - Adam Blake 16](#_Toc23969)

[2.1.2.3 Interview - Raj Patel 18](#_Toc9872)

[2.1.2.4 Interview - Mark Williams 22](#_Toc7456)

[2.1.3 Other problem domain research: 24](#_Toc28420)

[2.1.3.1 Comparable Software System review: 24](#_Toc7052)

[2.1.3.1.1 Records Management Systems 24](#_Toc9864)

[2.1.3.1.2 Student Records/Information Portal 28](#_Toc11849)

[2.1.3.1.3 Student Records/Information Applications (Mobile Format) 30](#_Toc19865)

[2.1.3.2 Development Relevant Legislation: 31](#_Toc31199)

[2.1.3.2.1 Equality Act: 31](#_Toc3223)

[2.1.3.2 General Data Protection regulations (GDPR): 31](#_Toc11078)

[2.1.3.2.1 Education Relevant Legislation 32](#_Toc12901)

[2.1.3.3 Academic literature review 32](#_Toc31856)

[2.1.3.3.1 Example Project Relevant research 32](#_Toc9544)

[2.1.3.3.2 Development Relevant Implications of Research: 33](#_Toc26182)

[2.1.3.4 User group questionnaires: 33](#_Toc16240)

[2.1.3.4.1 Student Experience Questionnaire 33](#_Toc13172)

[2.1.3.4.2 Academic Staff Experience Questionnaire 34](#_Toc7439)

[2.2 Requirement Specification 35](#_Toc699)

[2.2.1 Problem Domain Description 35](#_Toc20462)

[2.2.1.1 Existing Business Operation 35](#_Toc31903)

[2.2.1.1.1 Student Life-Cycle 35](#_Toc19677)

[2.2.1.1.2 Personal Tutorial Life-Cycle 36](#_Toc24485)

[2.2.1.1.2 Optional process (Staff Life-Cycle) 39](#_Toc18802)

[2.2.1.2 Summary of existing business limitations requiring resolution and existing strengths 41](#_Toc24460)

[2.2.2 Functional Requirements 42](#_Toc13554)

[2.2.2.1.1 Record Management System 42](#_Toc30484)

[2.1.3.1.2 Student Records/Information Portal 53](#_Toc3422)

[2.1.3.1.3 Woodlands University College Corporate Website 54](#_Toc2398)

[2.2.3 Performance Requirements 54](#_Toc6643)

[2.2.3.1 Records Management System 54](#_Toc22248)

[2.2.3.1.1 Speed 54](#_Toc7838)

[2.2.3.1.2 Capacity 54](#_Toc11176)

[2.2.3.1.3 Reliability 55](#_Toc30780)

[2.2.3.1.5Accessibility 55](#_Toc5852)

[2.2.3.2 Student Records/Information Portal 55](#_Toc20374)

[2.2.3.2.1 Speed 55](#_Toc17034)

[2.2.3.2.2 Capacity 56](#_Toc16759)

[2.2.3.2.3 Reliability 56](#_Toc28611)

[2.2.3.2.4 Usability 56](#_Toc3160)

[2.2.3.2.5 Accessibility 56](#_Toc1565)

[2.2.3.3 Woodlands University College Corporate Website 56](#_Toc6253)

[2.2.3.3.1 Speed 56](#_Toc22545)

[2.2.3.3.2 Capacity 57](#_Toc5403)

[2.2.3.3.3 Reliability 57](#_Toc2833)

[2.2.3.3.4 Usability 57](#_Toc32541)

[2.2.3.3.5 Accessibility 57](#_Toc18844)

[2.2.4 Design Constraints: 57](#_Toc15593)

[2.2.5 Commercial Constraints: 59](#_Toc22165)

[2.2.6 Acceptance Tests (Total Project): 59](#_Toc11019)

[3. System Analysis & Design (Records Management System) 60](#_Toc5284)

[3.1 Preliminary Design Stages 60](#_Toc151)

[3.1.1 Textual Analysis 60](#_Toc26999)

[3.1.2 Significant Event Analysis 61](#_Toc25617)

[3.1.3 Commands Queries and Constraints. 71](#_Toc2554)

[Course management system 72](#_Toc1677)

[3.2 Detailed Static System Designs 73](#_Toc4126)

[3.2.1 First Draft BON system architecture diagram 73](#_Toc22269)

[3.2.2 BON System chart: 77](#_Toc29528)

[3.2.3 BON Cluster Charts 78](#_Toc14419)

[3.2.4 BON Class Charts 80](#_Toc5228)

[3.3 Detailed Dynamic System Designs 88](#_Toc21362)

[3.3.1 Event Charts 88](#_Toc24349)

[3.3.2 Object Creation Chart 92](#_Toc22997)

[3.3.3 System Scenario Charts 94](#_Toc26803)

[3.3.4 Dynamic Diagram 98](#_Toc30325)

[4 System Interface Designs: 108](#_Toc14287)

[4.1 Draft Interface Designs (Records Management) : 108](#_Toc9936)

[4.1.1 Wireframes: 108](#_Toc10190)

[4.1.3 System Screen Mock-ups: 125](#_Toc17731)

[4.1.4 System Activity Event Diagrams: 138](#_Toc11332)

[4.2 Design Revisions(Record Management System): 143](#_Toc7514)

[4.3 Heuristic Evaluation (Record Management): 143](#_Toc986)

[4.4 Draft Interface Designs (Student Records/Information Portal) 145](#_Toc15379)

[4.4.1 Wireframes 145](#_Toc2348)

[4.4.2 System Navigation Diagram 151](#_Toc21957)

[4.4.3 System Screen mock-ups 153](#_Toc21691)

[4.4.4 System Activity Event Diagram 158](#_Toc26696)

[4.5 Design Revisions (Student Records/Information Portal) 159](#_Toc65)

[4.6 Heuristic Evaluation (Student Records/Information Portal) 160](#_Toc29863)

[4.7 Draft Interface Designs (Woodlands University College Corporate Website) 161](#_Toc1519)

[4.7.1 Wireframes 161](#_Toc23644)

[4.7.2 System Navigation Diagram 163](#_Toc31036)

[4.7.4 System Activity Event Diagrams 167](#_Toc17750)

[5 System Build and Technical Notes: 169](#_Toc12104)

[5.1 Technically Difficult Code Section (RMS) 169](#_Toc9060)

[5.1.1 Showing all courses in database: 169](#_Toc6955)

[5.1.2 Authorize access: 170](#_Toc14773)

[5.1.3 Update and delete selected module : 171](#_Toc1513)

[5.1.4 Delete Student: 172](#_Toc15752)

[5.2 On-going testing methodology: 173](#_Toc18150)

[5.3 Final System Interface Display: 174](#_Toc19276)

[5.3.1 Record Management System: 174](#_Toc2312)

[6 Test Strategy: 188](#_Toc10679)

[6.1 Overview of Test Strategy: 188](#_Toc18930)

[6.2 Sample Test Results: 189](#_Toc29342)

[7 System Evaluation 194](#_Toc27635)

[7.1 Usability Evaluation Strategy 194](#_Toc14233)

[7.1.1 Target Audience Selection/Sample Size Selection 194](#_Toc14501)

[7.1.2 Trail Active Task List Creation/Design 195](#_Toc10422)

[7.2 Initial System Pilot Trials/Results 198](#_Toc24913)

[7.2.3.1 Thinking Aloud (Observations) and Results 198](#_Toc13765)

[7.2.3.2 Post Event Walk-through (Questions) and Results 198](#_Toc13195)

[7.2.3.3 List of Key System Amendments 198](#_Toc9988)

[7.3 Final System Trial/Results 199](#_Toc11179)

[7.3.1 Usability Questionnaire Design 199](#_Toc20533)

[7.3.2 Final Questionnaire (Used in Trials) 200](#_Toc7177)

[7.3.3 Questionnaire Results – Implications of Results 203](#_Toc22887)

[8. CONCLUSION 204](#_Toc5931)

[9 Project Management 205](#_Toc13046)

[9.1 Project Gantt chart: 205](#_Toc31808)

[9.2 Project Meeting Minutes: 205](#_Toc13963)

[9.3 Project Quality Plan/Strategy: 206](#_Toc16547)

[9.4 Project Work log: 206](#_Toc17700)

[10 References 210](#_Toc9780)

**List of Figures:**

[Figure 1 Student Life-Cycle data in the current system 36](#_Toc9241)

[Figure 2 Personal Tutorial Life-Cycle data in the current system 38](#_Toc24328)

[Figure 3 Staff life cycle in the current system 40](#_Toc15849)

[Figure 4 Admin Login (wireframe) 109](#_Toc27836)

[Figure 5 Admin Dashboard (wireframe) 109](#_Toc7880)

[Figure 6 Show course (wireframe) 110](#_Toc9100)

[Figure 7 Add course ( wireframe) 110](#_Toc30742)

[Figure 8 show course details (wire-frame) 111](#_Toc24580)

[Figure 9 Update course (wire-frame) 111](#_Toc11211)

[Figure 10 Show modules page(wire-frame) 112](#_Toc15860)

[Figure 11 Add module page (wire-frame) 112](#_Toc12136)

[Figure 12 Update module(wire-frame) 113](#_Toc13067)

[Figure 13 Show staffs page(wire-frame) 113](#_Toc6958)

[Figure 14 Add staff page (wire-frame) 114](#_Toc14247)

[Figure 15 Update Staff (wire-frame) 114](#_Toc14734)

[Figure 16 Show Assignments(wireframe) 115](#_Toc29164)

[Figure 17 Add assignment(wireframe) 115](#_Toc10978)

[Figure 18 Update Assignment(wireframe) 116](#_Toc5908)

[Figure 19 Show Attendance 116](#_Toc29018)

[Figure 20 Update Attendance 117](#_Toc30469)

[Figure 21 Do attendance 118](#_Toc31033)

[Figure 22 Show Attendance Report 118](#_Toc26153)

[Figure 23 Show Personal Tutor (wire-frame) 119](#_Toc28993)

[Figure 24 Add Tutor 119](#_Toc31098)

[Figure 25 Update Tutor 120](#_Toc1899)

[Figure 26 Add Timetable 120](#_Toc18652)

[Figure 27 Show Timetable 121](#_Toc24028)

[Figure 28 Update Timetable 121](#_Toc11184)

[Figure 29 Show Diaries 122](#_Toc19477)

[Figure 30 Add Event/Diary Page 122](#_Toc9126)

[Figure 31 Update Diary 123](#_Toc16490)

[Figure 32 Report Page (wireframe) 123](#_Toc27090)

[Figure 33 Report Page (wire-frame) 124](#_Toc21941)

[Figure 34 System Navigation Diagram of Record Management System 124](#_Toc20995)

[Figure 35 Admin login (mock-up) 125](#_Toc29003)

[Figure 36 Show Courses Mock-up 125](#_Toc30800)

[Figure 37 Add Course (mock-up) 126](#_Toc32047)

[Figure 38 Course-detail(mock-up) 126](#_Toc27891)

[Figure 39 Update course (mock-up) 127](#_Toc4742)

[Figure 40 Show Modules (Mock-up) 127](#_Toc21083)

[Figure 41 Add module (Mock-up) 128](#_Toc10315)

[Figure 42 Update Module (mock-up) 128](#_Toc8572)

[Figure 43 Show staffs (Mock-up) 128](#_Toc6724)

[Figure 44 Add staff (Mock-up) 129](#_Toc2610)

[Figure 45 Update Staff (Mock-up) 129](#_Toc12809)

[Figure 46 Screen mockup of Attendance page 130](#_Toc5491)

[Figure 47 Add attendance page (mockup 130](#_Toc11959)

[Figure 48 Update Attendance (mockup) 131](#_Toc24571)

[Figure 49 Do attendance (mockup) 131](#_Toc26237)

[Figure 50 show attendance (mockup) 132](#_Toc2468)

[Figure 51 Show Personal Tutor 132](#_Toc23491)

[Figure 52 Add Tutor 133](#_Toc12051)

[Figure 53 Update Tutor 133](#_Toc8083)

[Figure 54 Show Timetable (mockup) 134](#_Toc16590)

[Figure 55 Add Timetable (mockup) 134](#_Toc14030)

[Figure 56 Update Timetable (mockup) 135](#_Toc31698)

[Figure 57 Show Diaries (Mockup) 135](#_Toc24700)

[Figure 58 Add Diary 136](#_Toc648)

[Figure 59 Update Diary 136](#_Toc5728)

[Figure 60 Show Report Page 137](#_Toc502)

[Figure 61 Add/ Generate Report 137](#_Toc31695)

[Figure 62 Login 138](#_Toc14130)

[Figure 63 assignment 139](#_Toc11971)

[Figure 64 Courses 141](#_Toc26909)

[Figure 65 students 141](#_Toc23382)

[Figure 66 stafff 142](#_Toc3572)

[Figure 67 Login Page (final display) 174](#_Toc25262)

[Figure 68 show course 176](#_Toc3622)

[Figure 69 course detail (final display) 176](#_Toc23601)

[Figure 70 update course (final display) 177](#_Toc6806)

[Figure 71 show modules page 177](#_Toc25273)

[Figure 72 add module page 178](#_Toc407)

[Figure 73 Update module 178](#_Toc30920)

[Figure 74 show student (final display) 179](#_Toc10332)

[Figure 75 add student 179](#_Toc28876)

[Figure 76 update student 179](#_Toc21853)

[Figure 77 show staff page 180](#_Toc10913)

[Figure 78 update staff (final display) 181](#_Toc31228)

[Figure 79 show assignment 181](#_Toc19032)

[Figure 80 add assignment 181](#_Toc16233)

[Figure 81 update assignment 182](#_Toc19511)

[Figure 82 student portal dashboard (final display) 182](#_Toc23235)

[Figure 83 course that student is enrolled in. 183](#_Toc6656)

[Figure 84 student enrolled module 183](#_Toc27776)

[Figure 85 page showing module details 183](#_Toc1399)

[Figure 86 Showing assignment in that module (final display) 184](#_Toc28430)

[Figure 87 change password in student portal(final display) 184](#_Toc19486)

[Figure 88 staff dashboard(final display) 184](#_Toc30906)

[Figure 89 show modules of staff portal 185](#_Toc2115)

[Figure 90 showing module detail 185](#_Toc21714)

[Figure 91 change password in staff portal 185](#_Toc13116)

[Figure 92 University website homepage (final display) 186](#_Toc14342)

[Figure 93 about us page (final display) 187](#_Toc7410)

List of Tables:

[Table 1 Student Record Management System Attributes 43](#_Toc19439)

[Table 2 Staff Record Management System 44](#_Toc11993)

[Table 3 Department management 45](#_Toc30947)

[Table 4 Course record management 45](#_Toc20428)

[Table 5 module management 46](#_Toc24984)

[Table 6 assigment management 47](#_Toc6354)

[Table 7 attendance management 49](#_Toc998)

[Table 8 personal tutor 50](#_Toc4143)

[Table 9 timetable management 51](#_Toc32439)

[Table 10 Diary management system 51](#_Toc26557)

[Table 11 textual analysis 61](#_Toc2924)

[Table 12 admin 62](#_Toc24952)

[Table 13 63](#_Toc9852)

[Table 14 staff 64](#_Toc23872)

[Table 15 department 65](#_Toc11102)

[Table 16 course 65](#_Toc29659)

[Table 17 module 66](#_Toc26538)

[Table 18 assignment 67](#_Toc17210)

[Table 19 attendance 68](#_Toc27003)

[Table 20 personal tutor 69](#_Toc20477)

[Table 21 69](#_Toc3351)

[Table 22 70](#_Toc8249)

[Table 23 71](#_Toc3538)

[Table 24 feedback 71](#_Toc12048)

[Table 25 Bohn class chart 81](#_Toc12874)

[Table 26 82](#_Toc19827)

[Table 27 83](#_Toc16757)

[Table 28 83](#_Toc26140)

[Table 29 84](#_Toc32208)

[Table 30 85](#_Toc3073)

[Table 31 85](#_Toc27675)

[Table 32 86](#_Toc5376)

[Table 33 87](#_Toc5458)

[Table 34 87](#_Toc30488)

[Table 35 88](#_Toc28034)

[Table 36 Events 89](#_Toc1497)

[Table 37 object creation chart 94](#_Toc7388)

# 1 Introduction

This is a project report on a course management software system for Woodlands University College (WUC). The following report includes techniques to find out the problem in the clerically based system management of courses. This report includes the overall part of the pilot system software development process along with various steps. As a result, at the end of the report, everyone who has access to it would be familiar with the issues with the clerical-based course management system as well as viable solutions.

## 1.1 Project Background

The project is based on the problem faced by the current course management system of the university. They have been using a clerical system to manage their course degree course, due to which they are facing several problems. Their management system is facing problems to keep relevant data, keeping track of data, etc. Therefore, this project finds and works on the solution to those problems.

Here, in this report, using different techniques, every problem of the clerical system will be detailed and on that basis, an appropriate solution will be prepared, so that the new feature in the upcoming system will not give the same problem.

## 1.2 Project Aims and Objectives

* Use a variety of methodologies to determine the issue domain of the course management in the current system.
* Discover everything there is to know about the course management problems in the current system.
* Make a list of all challenges faced.
* Look out for the solutions to the challenges that have been documented.
* Development of the software to the client's requirements and expectations.

## 1.3 Project Development Methodology

Development of project follows specific methodology. Following right methodology is important to study, research and development of required software. Here, Agile methodology is considered as best testing methodology for the required prototype or system. The Agile methodology is a process to manage a project by breaking or dividing it into smaller parts. Different functionality is break down and completed step by step. It involves consistent collaboration or support with stakeholders. For desired prototype, team work in cycle of planning, executing and development. Continuous collaboration can be virtual or physical. Team and stakeholder have meeting either physically or virtually so that there is great possibility of accuracy and acceptance of required functionality in the system. Similarly requirement engineering is the way of agile methodology.

# 2 Requirement Engineering

## 2.1 Elicitation Activities

The elicitation process is carried out starting through the interview plans of the project where we can collect various information based on the problem domain and requirements.

### 2.1.1 Interview Plans

It is one of the important topics of requirement engineering where questions were prepared to be asked to the member of the Woodlands University College (WUC). The questions are listed below. During the interview time, some additional questions were asked for further clarification are all documented below in the Interview Findings section.

1. **Dr. Simon White (Computing Course Leader)**
2. What is your main goal in expanding this institution in the Internet world?
3. What about your roles and privilege in this university?
4. How course records are stored in the present system? Are there separate methods to record student and staff details?
5. Would you like to brief me about other management systems (like module management, assignment, attendance management, etc.)?
6. What disadvantages/disabilities have you experienced in the existing system?
7. How different sections (like course sand section, module section) should be implemented and organized?
8. How about integrating special features like SMS notifications, chatbox, virtual classes, etc. in the new system?
9. Would you like plain textual content or content description with animation/ videos?
10. Any content you would love to view on a new website?
11. Would you like a sticky navigation menu at top of the page or menu should only be visible after clicking the menu button?
12. How would you like to see the performance of the system? E.g. Data storage capacity at once, no user of software at a time, etc.
13. New software must be in a popular programming language like (Java /Python/JavaScript) or can be written in any programming language?
14. **Mr. Adam Blake (Course Administrator)**
15. What is your motivation to upgrade the system?
16. Is there any system that you recommend as the model for our system?
17. What are the difficulties in providing administrative support to course teams with the current system?
18. What are the difficulties in providing administrative support to course teams with the current system?
19. How does the clerical system operate?
20. What features you would like to include from the previously used clerical system?
21. What kind of problem are you facing being an administrator using the clerical system?
22. Do you prefer any other functionality that makes your responsibilities easier?
23. Are there any questions you feel we should have asked?
24. Do you want to share any other information regarding the system?
25. **Dr. Raj Singh (Senior Lecturer, module leader, and personal tutor)**
26. What benefits might you want to get from the new type of programming and what weaknesses could you encounter in the past framework?
27. What are the introduction necessities of this system and how can we apply them?
28. What methodologies do you utilize to preserve a course management system?
29. Should the model incorporate the highlights which will or may not be included within the course management system?
30. What ought to be the principal measures for the management portal?
31. What do you think are the challenges lecturers face when executing a course management system?
32. What sort of obviously the course management system would you say you are considering?
33. **Mr. Mark Williams (Existing Student)**
34. How would you like to give feedback on the current course management system?
35. What do you think is the most needed change in the current system of the university?
36. In your opinion, one most important benefit to the student because of the new system?
37. What are the improvements you think can help the university function better and that it should work on?
38. University might also conduct a meeting or any class on an online platform after this, are u comfortable with that?
39. Have you gone through any other university websites? If yes, have u got some insightful ideas that u want to share with us.

### 2.1.2 Interview findings

#### 2.1.2.1 Interview - Simon White

Interview Title: The computing course leader – Dr. Simon White

Interview Date: 5/20/2022

Duration: 30 minutes

Persons who take interview:

Aayush Shrestha

The inquiries and their separate solutions for this interview are point by point in the table beneath:

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Questions and Answers |
| Aayush | 01 | What is your main goal in expanding this institution in Internet world?  The main goal for the expanding this institution in Internet is to offer international information and verbal exchange access. The Internet and networking are critical for reinforcing engineering, and format research, further to worldwide safety and observation. |
|  | 02 | What about your roles and privilege in this university?  Get to the reservation framework and the capacity to book and organize occasions, get to college staff administrations, the capacity to ask stores from the TCU treasury, and get to the distribution of the occasion on the birth calendar tablets. |
|  | 03 | How course records are stored in the present system? Are there separate methods to record student and staff details?  The Records are place away in successive organize, upheld the regard of the plan key of each and every record. associated records of the fluctuated relations are placed away on a comparative piece all together that one I/O activity brings related records from every one of the relations. |
|  | 04 | What disadvantages/disabilities you experienced in the existing system?  The disadvantages and disabilities that I experienced in the existing system are the Absence of valuable gear or innovation to assist the person with being viable. |
|  | 05 | How different sections (like course section, module section) should be implement and organize?  Course and module portions are coordinated by parceling the subject into segregated material module lengths similar, utilizing a dependable design inside and among modules, and putting together the material modules similarly to the conventional graph. Also, will be completed. |
|  | 06 | Any colours, themes, images you would like to add in the new software?  E.g. different background for different users like students, staffs, etc.    Except if there's an especially legitimate justification for making use of brilliantly hued textual content don't. Stay with white or mild beige on a dull foundation or dark commonly extraordinarily dim variety on a mild basis |
|  | 07 | Would you like plain textual content or content description with animation/ videos?  Obviously, infographics specifically are noticeable portrayals of information and records created as diagrams, charts, and tables as well as videos. Introducing realities to the per user during this manner will basically have an effect on however content is gotten, comprehended, shared, and place away. |
|  | 08 | Would you like sticky navigation menu at top of page or menu should only be visible after clicking menu button?  Sticky menus are perfect for landing pages. You can urge clients to look over the whole page. As a matter of fact, there are times when sites have an extra noticeable piece of rich landing pages subsequent to changing the everyday fundamental route menu with a tacky menu. |
|  | 09 | Are there any documentation related with existing current system? If yes, can we have overview on it?  The framework survey really recognizes the ideal electric system innovation, equipment of electric, and hardware. This document recognizes each of the imperative added substances of the structure and realities them as a receipt of materials. |
|  | 10 | How data are secure and protected among university? Who would have access to new software?  We scramble records, make antique PC intense drives unintelligible, and watch homegrown and organizational wi-fi networks. Make encoded volumes for movable non-openly available reports archives for endlessly stable records so the handiest chiefs and pioneers can get to the pristine framework |

#### 2.1.2.2 Interview - Adam Blake

Interview Questions

Course Administrator: Mr. Adam Blake

1. Business Models

* Is there any system that you recommend as the model for our system?

1. Existing system

* How does the clerical system operate?
* What kind of problem are you facing being an administrator using the clerical system?
* How do you manage the appointment using the current system?
* What features you would like to include from the previously used clerical system?
* Do you prefer any other feature in the system?

1. Any other

* Are there any questions you feel we should have asked?
* Do you want to share any other information regarding the system?

Interview Title: Interview with the Administrator: Mr. Adam Blake

Interview Date: 00/00/2022

Duration: 30 minutes

|  |  |  |
| --- | --- | --- |
| Interviewer | Question  Number | Questions  Client-Response |
| Pragyan | 01 | Is there any system that you recommend as the model for our system?  Yes, there are different systems. Among many other systems, the NILE system can be one of the best models for the system. |
|  | 02 | How does the clerical system operate?  The clerical system is fully based on the paper. The manual process of recording various data has been a part of the university. The recording of daily activities of the university manually on the document and paper-based spreadsheets has made the more effort and time-consuming with more risks. Thus, every task is done in the paper-based system. |
|  | 03 | What kind of problem are you facing being an administrator using the clerical system?  There are many problems in using the clerical paper-based system. Scattering of data and risk of data loss are the major problem faced. Similarly, the manual recording of all the records is much more time-consuming. |
|  | 04 | How do you manage the appointment using the current system? |
|  | 05 | What features you would like to include from the previously used clerical system? |
|  | 06 | Do you prefer any other feature in the system?  All the features that have been included in the document are compulsory. The implementation of those features are enough for the system. |

#### 2.1.2.3 Interview - Raj Patel

Interview Title: A senior lecturer, module leader and personal tutor from the computing Department – Dr.raj singh

Interview Date: 5/20/2022

Duration: 30 minutes

Persons who take interview:

Sungava Subedi

The inquiries and their separate solutions for this interview are point by point in the table beneath:

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Questions and Answers |
| Sungava | 01 | What is the purpose of a course management system and how does it work?  The main purpose of Course Management System ordinarily comprises a couple of online hardware and settings, which incorporates a region for teachers to set up course substances that incorporate prospectus and syllabus. It is a variety of programming tools that provide a web-based environment for course communication. |
|  | 02 | What are the advantages and significance of a course management system?    Actually, this dispenses with the expenses of movement to preparing offices, internet preparing rentals, and written words.  The course management system makes it smooth for educators to excel sources with all in their understudies with the guide of involving in all actuality bringing in the sources. |
|  | 03 | What benefits might you want to get from the new type of programming and what weaknesses could you encounter in the past framework?  Course Management System is a product program answer that might help foundations control heading communications, extraordinarily for distance learning. |
|  | 04 | What are the introduction necessities of this system and how can we apply them?  The computerized security, limit, and strong verbal trade provided through method of method for PC frameworks might be a valuable resource. You maintain that a PC programming should accomplish the work. PCs draw in with various organizations through IT. |
|  | 05 | How do you want to transform your paper system into modern programming?  All records will be effectively open, and materials will be available through a solitary framework, on account of an efficient way to deal with overseeing paper-based frameworks. |
|  | 06 | What methodologies do you utilize to preserve a course management system?    The methodologies to preserve a course management system are the strategies of erasing unimportant substances and putting away old substances somewhere else, tidying up dead clients, changing or system corrections, arranging a free time, reworking modules in a predictable request, and eliminating the "build-up". |
|  | 07 | What ought to be the principal measures for the management portal?  The high satisfaction with the technology, the responsiveness of peoples to feedback, the operating environment of the system, and the productivity of the team all point to the presence of a good control portal. |
|  | 08 | What do you think are the challenges lecturers face when executing a course management system?  In my point of view, these are the difficulties that speakers face issues with Course Administration and Substance, Personalization issues and problems, issues with customer Interaction, Ineffective instructor Preparation, and consequently the need for power point Integration. |
|  | 09 | What sort of obviously the course management system would you say you are considering?  I would consider the course management system by assessing a course the executive's framework, judging specialized issues and restrictions, pondering the capacities of the preparation improvement group, surveying the skill and foundation of the CMS seller, and considering the data following abilities of the CMS. |

#### 2.1.2.4 Interview - Mark Williams

Interview: Date: 09/05/2022

Duration: 9 minutes

Persons in attendance:

Ritesh Koirala

|  |  |  |
| --- | --- | --- |
| Interviewer | Question Number | Questions and Answers |
| Ritesh | 01 | How would you like to give feedback on the current course management system?  Its even for the student, to manage everything with hand and paper it started becoming hard so nowadays it has even been the essential part of the university to keep it. |
| Ritesh | 02 | What do you think is the most needed change in the current system of the university?  They should implement different section of university in very organized way so that everyone who has the access to it have the easiness in working with it. It should be attractive, easy to find and easy to understand. |
| Ritesh | 03 | In your opinion, one most important benefit to the student because of the new system?  The content, which describe the courses, module, the message given by the university high authority and the description of the course will be clear in understanding and influencing. |
| Ritesh | 04 | What are the improvements you think can help the university function better and that it should work on?  Integrated other management system like module, assignment, attendance management can lead in functional and effective way of management of different departments of this university which help in better function of university. |
| Ritesh | 05 | University might also conduct a meeting or any class on an online platform after this, are u comfortable with that?  Yes, as world is moving with technology we have been using different functions like video chat, live streaming. This functions are similar with meeting or online class. We will easily cooperate with this decision of university. Therefore, It will be some more interesting features of our university. |
| Ritesh | 06 | Have you gone through any other university websites? If yes, have u got some insightful ideas that u want to share with us?  Yes, I have visited site of university of northampton. It’s in UK. They have proper navigation system for new user. This site provide all learning materials like slides, videos, books for student. They conduct examination online which are beneficial for student living in out of the country. They provide grades. Student can learn anything they are enrolled in. Similarly, other students presentation videos, or tutorial videos are available in its site (Kaltura). |

### 2.1.3 Other problem domain research:

The given topic of the assignment looks for the different existed software systems and relevant legislation that will define to complete the project by understanding other problems.

#### 2.1.3.1 Comparable Software System review:

In this part of the project, various software systems are researched. Similarly, their advantages and disadvantages are also listed.

##### 2.1.3.1.1 Records Management Systems

The procedure of keeping records is one of the most important task in any of the company. It plays a vital role as a university also needs to store many records from different sectors such as, course records, staff records, student records, etc. Record management was all based on clerical system (paper-based system). Create, amend, assign, display was not easy in paper based system. Due to which computerized system has been introduced through which we can easily record every single data. Hence, the major reason to introduce computerized course management system is to provide easy way to perform task such as create, amend, archive, display and assign which is the best way to secure data for the future.

Some comparable record management system are:

|  |  |
| --- | --- |
| System | Description |
| **Vanderbilt University**  https://cft.vanderbilt.edu/university-courses/#course | The Vanderbilt University Center for Teaching is a university in Tennessee. The university sites have made the university members' activities easier. It promotes new and creative learning. The site seems user-friendly with eye-catchy color. |
|  | Advantages and Disadvantages:   |  |  | | --- | --- | | Advantages | Disadvantages | | It provides different kinds of services. |  | | User-friendly and eye-catchy site. |  | | A search bar is present on the site through which different courses can be searched. |  | | It addresses enduring historical question, cross cutting societal themes. |  | | It takes a unique approach to a topic by looking at research from a diverse view. |  | |
| **OpenKm**  https://www.openkm.com/ | This software is free and made for record management. The site is eye-catchy. This system is aimed for creating and maintaining authentic, reliable and useable records. |
|  | Advantages and Disadvantages are:   |  |  | | --- | --- | | Advantages | Disadvantages | | The main page is presented well. The key features are all clearly displayed. | Search bar feature is not available. | | It includes a lot of literature with terms related to document management |  | | It provides training services for the new users. |  | | It indicates where record are stored. |  | |

##### 2.1.3.1.2 Student Records/Information Portal

|  |  |
| --- | --- |
| **UoN (Nile)**  **https://nile.northampton.ac.uk/** | This software is student information portal of university of northampton. Different functionalities of this site is presented in side-bar.  Screenshot (379) |
|  | Advantages and Disadvantages are:   |  |  | | --- | --- | | Advantages | Disadvantages | | Plain and attractive UI |  | | Student can get course that they are enrolled in |  | | The module items consists of contents like (Lectures, tutorials, assignments, timetables, grades, etc) |  | | With student’s username and password. They can access its other site like (kaltura, blackboard, Turnitin) |  | |

##### 2.1.3.1.3 Student Records/Information Applications (Mobile Format)

|  |  |
| --- | --- |
| **UoN (Nile)** | This is the mobile application introduced by the University of Northampton. It is for the student's records. The course and module are provided to the enrolled students of Northampton. The mobile application has created useful impacts on students. So, some of the advantages and disadvantages are given below: |
|  | Advantages and Disadvantages are:   |  |  | | --- | --- | | Advantages | Disadvantages | | User-friendly site. |  | | Students can perform different tasks |  | | Viewing courses, modules and results has made this app convenient and easy to use. |  | |

#### 2.1.3.2 Development Relevant Legislation:

##### 2.1.3.2.1 Equality Act:

All the students in the university must be treated equally. Any education provider, including a private or independent provider, is prohibited from discriminating against students on the basis of disability, race, sex, gender reassignment, pregnancy and motherhood, religion or belief, or sex under the Equality Act 2010.

Discrimination on caste, race, sex, religion, gender is unlawful activities. So to avoid any kind of discrimination government has issued some equality act.

1. Prohibit discrimination
2. Harassment
3. Victimization
4. Special provision for disability

##### 2.1.3.2 General Data Protection regulations (GDPR):

This course management software carries different data and information from the administrative department, course leader, lecturer, students, and others who are related to this field. The information should be confidential so it should be handled very carefully under certain rules and regulations.

Different requirements should be enforced under the Data Protection Act 2018 of the United Kingdom (UK), as stated by the General Data Protection Regulation (GDPR) requirement in law. According to GOV.UK, 2018, the following are the rules regarding privacy and protection of data and information maintained within the system.

* The data should only be used within the systems of specific organizations.
* Unwanted data storage should be avoided.
* Only certified members should have access to the organization's vital data and information.
* Personal data should be used legally and transparently.

##### 2.1.3.2.1 Education Relevant Legislation

The Act Relating to Compulsory and free Education, 2075 (2018), Act No. 16,

* Right to get education
* Liability of State to provide education
* Duty of citizen to send to school
* To provide compulsory education
* To acquire education from convenient school
* Not to refuse to get admission
* Not to expel from school
* To provide transfer certificate

#### 2.1.3.3 Academic literature review

##### 2.1.3.3.1 Example Project Relevant research

In addition, some research has been carried out to support the course management system in academic journals which is given below:

|  |  |  |
| --- | --- | --- |
| Title | Author/Date | Description |
| Applying the Technology Acceptance Model to explore the effects of a Course Management System | Tsai, Yea-Ru/2015-01-01 | This research has been carried out to transform the current clerical-based course management system into a computerized system. It deals with the corporate way of course management system making smart learning and teaching process. |
| Implementation of a Course Management System: Experiences and Students' Thoughts | Williams, Gary/2003-03-01 | The given article describes the implementation of a course management system in a university or college. This paper also describes the way to help teachers and students using this system. |

##### 2.1.3.3.2 Development Relevant Implications of Research:

This research has been carried out to transform old course management technology to modern technology. It deals with cooperate way for course management system with smooths in learning and teaching process. Implementation of course management result in student/staff/university performance. So proper knowledge from research will help a lot in gathering the knowledge of author. From this “Applying the Technology Acceptance Model to explore the effects of a Course Management System” we learn to explore the record management compatibility with modern technologies. The different items are gathered on single screen so that it will be many times better than current system. “Implementation of a Course Management System: Experiences and Students' Thoughts” provides the systematic way of implementation of course management system in Internet world.

#### User group questionnaires:

Here we conducted surveys/ questionnaire with users of system of this university. Their opinions are very useful for sustainable system development.

##### Student Experience Questionnaire

###### Questionnaire Development:

1) Your opinion on current paper based system.

2) Whether the system has good response on it?

3) Whether this system is preferable or not?

4) Is this system good for your university?

###### Questionnaire Results:

|  |  |
| --- | --- |
| Feedback | value |
| Positive | 33.3% |
| Negative | 65.6% |
| Did Not Use | 1.1% |

###### Questionnaire Analysis:

- Complicated system of paper-based system.

- Inserting, searching, deleting and other operation on this current system was too time consuming.

- No backup for lost data. Requires an efficient way of record management system.

##### 2.1.3.4.2 Academic Staff Experience Questionnaire

###### 2.1.3.4.2.1 Questionnaire Development

1. How do you like to rate the current course management system of this university?
2. What is the important changes needed for the university course management system?
3. What kind of fonts/color you like to see?
4. What feature would really help the staff?
5. What kind of issues you found on current system?

###### 2.1.3.4.2.2 Questionnaire Results

|  |  |
| --- | --- |
| Feedback | value |
| Positive | 23.3% |
| Negative | 74.5% |
| Did Not Use | 2.2% |

###### 2.1.3.4.2.3 Questionnaire Analysis

**-** Bad ratings by users.

- Hard to manage courses and its module.

- Student records management is very difficult to handle in this system.

## Requirement Specification

### 2.2.1 Problem Domain Description

This portion of the report outlines the current business practices that include documentation deployed for strategically important. Diagram techniques are also used.

#### 2.2.1.1 Existing Business Operation

The proper description and analysis should be done on the current system while developing a new system. The merits and demerits of the existing system should be known. Proper knowledge of the current system is helpful in requirement specification which includes student life-cycle, and personal tutor life-cycle.

##### Student Life-Cycle

A clerical system is used in this course management system. Records management is also a clerical-based system. Clerical documentation is used in every sector. The student life cycle is based on student record management. Hence, Students' life-cycle process has a different phase.

Students apply to the university. Then, the life cycle starts from their arrival at the university. The applied application is accessed by the university admission team. If the application doesn't meet the admission criteria, the application gets rejected. But if the criteria are met paper is created for the student that contains all the relevant information. Then the case paper is then stored in a provisional filing cabinet. And then upon the arrival of a student at the university to enroll, the admin team completes the student case paper using the information received from enrolment. Then the case paper is stored in a file and moved to the dormant cabinet. Here, students are divided between these processes based on new enrollment and transfer. Similarly, during the enrolment, students choose courses of their choice recorded in the same paper case.

Moreover, students are also being checked regularly in all manner. Home-works and assignments that are assigned to the students are checked regularly and the collected results as data of different students are stored in the paper-based system and filed those case papers separately for each student. Passed and failed students are separated. Different activities are recorded separately using a clerical system.

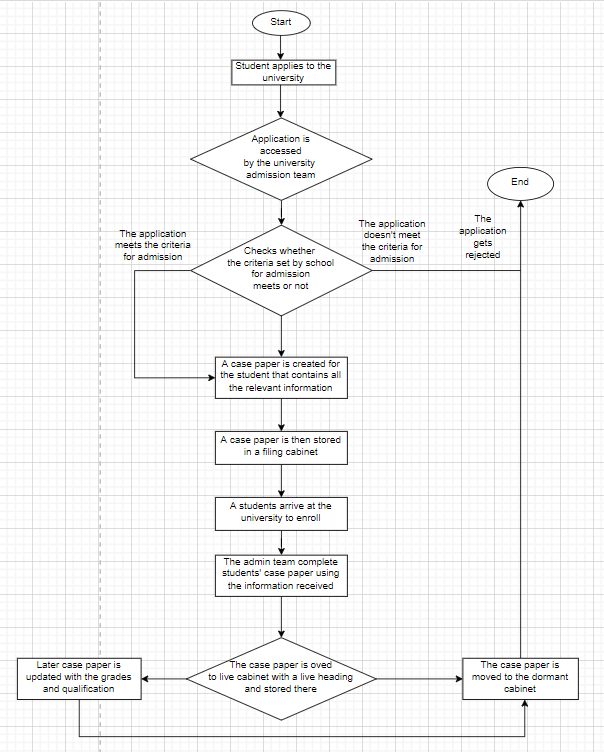


Figure 1 Student Life-Cycle data in the current system

##### 2.2.1.1.2 Personal Tutorial Life-Cycle

The personal tutorial life cycle has various steps. Starting from the application process of tutor moves into the guidance and monitoring of the students too in their every difficulty. Every aspect of this cycle is based on a clerical system.

The recruitment process of tutors starts with the application. The applied application is accessed by the team. The team checks for the criteria required according to the university rule. If the criteria meet then interviewing process, advice, and guidance, referring to the careers team starts. Planning the program, and the enrolment process starts. Then the case paper for the tutor is created which contains all the information i.e. stored in a filing cabinet. The tutor is assigned to a group or assigned personally. Then the actual tutorial starts with tracking and monitoring and setting some targets for the students. During this timeframe attendance and punctuality are also monitored. Continuing with this process students are assigned assignments and projects. The marking process is followed by grades or points that are stored in the student's case paper. Later the file is updated with markings. Personal tutors help students who could not complete their assignments or projects and properly guide them. Changing the students' study plan also lies within the cycle. The tutorial moves to the end of the program to collect all the grades and update the case paper.

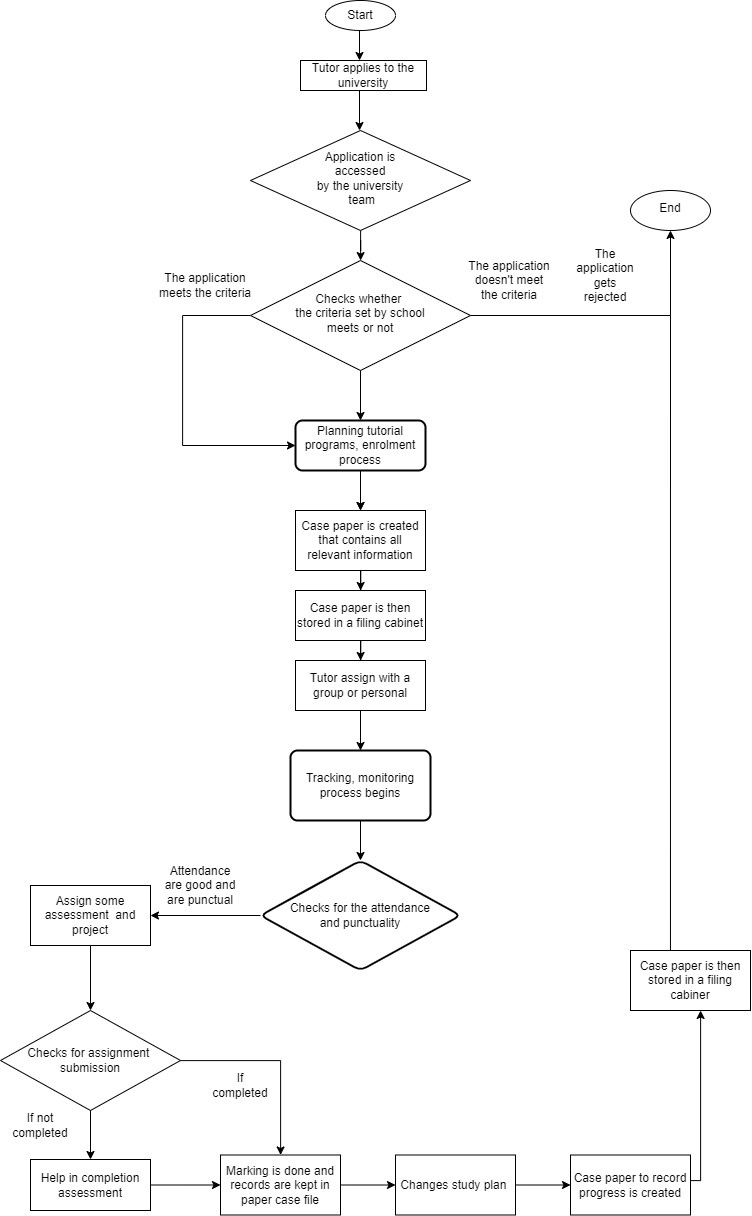


Figure 2 Personal Tutorial Life-Cycle data in the current system

##### 2.2.1.1.2 Optional process (Staff Life-Cycle)

The current system is based on clerical type. Paper-based documentation is used to store every data. The below flowchart is of a staff life cycle. The various procedure is used in the staff cycle in the existing system.

As for student and personal tutors, the staff life cycle also begins from the application process. In the same way, the administrator checks the applied application on the basis of their qualification and as per the rule of the university. In case the criteria don't meet the application gets rejected. But when the criteria meet the process move to another phase. The interview process begins with those applicants whose applications were selected. In case applicants could not complete the interview phase again get rejected. Those who had selected in the interview evaluation and offer would be made by the university admin team. Finally, on completion of all the stages applicants will be permanently employed. Staff case paper is created and is stored and moved to the dormant cabinet. All the responsibilities and duties will be given to the new staff as per their qualification. Student records and grades are updated by the staff in their respective case paper. The life cycle is shown below in the flowchart.

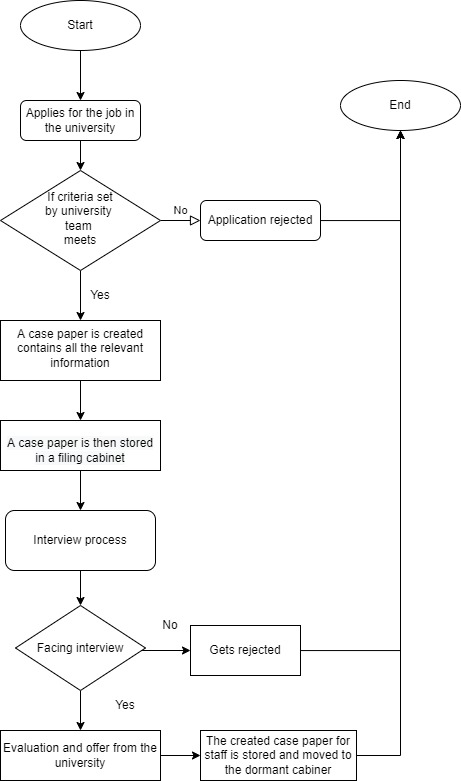


Figure 3 Staff life cycle in the current system

#### 2.2.1.2 Summary of existing business limitations requiring resolution and existing strengths

While working on this project, we found that there are some limitations to this system. So, some of the major limitations are listed below:

* Easily damaged

Paper is a very sensitive material that can easily get damaged by any means. These documents may easily get harmed by different natural factors such as water, fire, etc. If they are destroyed once, they can't be recovered unless and until we find a copy of that. Therefore, there is a high risk of data loss which can be solved by a computerized system. Even natural disasters cannot harm the data. In case of any data loss or harm, data saved online can be restored by various processes.

* Security Problem

There is a high risk of data getting lost. It could be stolen and related people may mishandle the data. So there is less chance of data being stolen while using a computerized system.

* Need more space

Paper-based documents require more space to store more data. Recording daily activities of lecturers and students need more paper which needs more area to store. So using a computer-based system doesn't need lots of space. Data can be stored within a storage device that occupies very less space.

* High cost

Keeping, thousands of data needs thousands of paper which is very expensive. So, handling all the data and documents can be managed in software and will be easily accessible.

* Low Efficiency of data transfer

The transfer of paper-based data and documents from one place to another is difficult and risky which may result in data damage and loss. But data transfer can be easily done using a computerized system (online system).

### 2.2.2 Functional Requirements

Functional requirements define the capabilities, features of the software as well as descriptions of a software application's characteristics and functionality. It helps with characterizing the framework's planned assistance and conduct. This facilitates perceiving unmet desires and absolutely outline predicted device offerings.

##### 2.2.2.1.1 Record Management System

The record management system is to deal with an association's records all through the records lifecycle is the obligation of the records the board framework. It is the control of an association's data and comprises the creation, stockpiling, support, and annihilation of data. It also co-ordinate records safety to permit inner and outside gets entry to records safety and business-associated records. It records the executives present your association's obligation and all around coordinated get admission to data.

1. **Student Record Management System:**

An automatic device for managing, protecting, and retaining instructional databases as well as for noticing and adjusting to student data is known as student record management system. It’s co-ordinate timetables and realities concerning understudy related matters.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Student\_id | NUMBER(8) | Id of student. |
| Student \_name | VARCHAR2(20) | Name of student. |
| Address | VARCHAR2(14) | Address of student. |
| Gender | CHAR | Gender of student. |
| Age | NUMBER(4) | Student’s age. |
| Date\_ of\_Birth | NUMBER(10) | Student’s date of birth. |
| Email | VARCHAR(20) | Email id of students. |
| Phone\_number | NUMBER(10) | Student’s phone number. |

Table 1 Student Record Management System Attributes

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | No Access |
| Archive | No Access | Access | No Access |
| Display | Access | Access | Access |
| Assign | No Access | Access | No Access |

1. Staff Record Management System:

Representative information is made in an assortment of arrangements. It is generally a paper record or mechanized information, however it can likewise incorporate email, advanced reports, accounts, distributions, other text, varying media, and robotized virtual data.

It empowers human resources gathering of laborers to tune specialist realities like income subtleties, clinical realities, participation records, and general execution.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Staff\_id | NUMBER(9) | Id of staff. |
| Staff \_name | VARCHAR2(21) | Name of staff. |
| Address | VARCHAR2(12) | Address of staff. |
| Gender | CHAR | Gender of staff. |
| Age | NUMBER(7) | Staff’s age. |
| Date\_ of\_Birth | NUMBER(13) | Staff’s date of birth. |
| Email | VARCHAR(25) | Email id of staff. |
| Phone\_number | NUMBER(11) | Staff’s phone number. |
| Quality | VARCHAR(20) | Staff’s quality. |
| Department | VARCHAR(17) | In which department they related to. |
| Responsibility | VARCHAR(14) | Responsibility of staff. |

Table 2 Staff Record Management System

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | No Access |
| Archive | No Access | Access | No Access |
| Display | Access | Access | Access |
| Assign | No Access | Access | No Access |

1. Department Record Management System:

Department management system is the precise control of the association's records, ways of life cycle, functional endeavor necessities, and get together lawbreaker and monetary prerequisites.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Department\_id | NUMBER(13) | Id of department. |
| Department \_name | VARCHAR2(170) | Name of department. |
| Title | VARCHAR2(140) | Title of department. |
| Description | VARCHAR(1200) | Description of department. |

Table 3 Department management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | No Access |
| Archive | Access | Access | No Access |
| Display | Access | Access | Access |
| Delete | No Access | Access | No Access |
| Assign | Access | Access | Access |

1. Course Record Management System:

Course management system is the precise control of the association's records, ways of life cycle, functional endeavor necessities, and get together lawbreaker and monetary prerequisites.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Course\_id | NUMBER(13) | Id of course. |
| Course \_name | VARCHAR2(17) | Name of course. |
| Title | VARCHAR2(14) | Title of course. |
| Course\_number | NUMBER(13) | Number of course. |
| Credits | NUMBER(9) | Credit hour of course. |
| Code | NUMBER(11) | Course code. |
| Description | VARCHAR(12) | Description of course. |
| Type | VARCHAR(8) | Course type. |

Table 4 Course record management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | No Access |
| Archive | Access | Access | No Access |
| Display | Access | Access | Access |
| Delete | No Access | Access | No Access |
| Assign | Access | Access | Access |

1. Module Management System:

The module inside the suspension can be a module intended to be mounted in a compartment that deals with the sharp edges, go through modules, and different parts housed inside outlines is module management. A module the board comfort is a web issue that helps withinside the control and the executives of modules and ensures consistence with different institutional and specialist guidelines.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Module\_id | NUMBER(18) | Id of module. |
| Module \_name | VARCHAR2(20) | Name of module. |
| Module\_number | NUMBER(15) | Number of module. |
| Credits | NUMBER(10) | Credit hour of module. |
| Code | NUMBER(5) | Module code. |
| Description | VARCHAR(18) | Module Description. |
| Type | VARCHAR(6) | Module types. |

Table 5 module management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | No Access |
| Delete | Access | Access | No Access |
| Archive | Access | Access | No Access |
| Display | Access | Access | Access |
| Assign | Access | Access | No Access |
| Mark Grade | Access | Access | Access |

1. Assignment Management System:

Assignment Management System is a CAC-empowered stage that keeps up with our arrangement of utilizations to verify that the appropriate avionics representatives play the legitimate obligations on the legitimate time.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Assignment\_id | NUMBER(10) | Id of assignment. |
| Title | VARCHAR2(20) | Title of Assignment. |
| Start\_date | NUMBER(8) | Start date of assignment. |
| End\_date | NUMBER(10) | End date of assignment. |
| Category | VARCHAR(15) | Assignment category. |
| Description | VARCHAR(18) | Assignment  description. |
| Type | VARCHAR(6) | Assignment types. |

Table 6 assigment management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | Access |
| Amend | Access | Access | Access |
| Delete | Access | Access | No Access |
| Archive | Access | Access | Access |
| Display | Access | Access | Access |
| Assign | Access | Access | Access |
| Mark Grade | Access | Access | Access |

1. Attendance Record Management:

An authoritative philosophy for observing worker time and gathering conduct data is known as attendance record management. It records everyday cooperation, dynamic hours, downtime, log in and out time as well as accurate records are maintained.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Attendance\_id | INT(12) | Attendance Id. |
| Student\_id | VARCHAR(13) | Student id of attendance. |
| Staff\_id | VARCHAR(14) | Staff id of attendance. |
| Session | INT (20) | Session id of attendance. |
| Date | NUMBER(10) | Attendance date |
| Roll no | NUMBER(10) | Roll number of attendance. |
| Description | VARCHAR(15) | Attendance  description. |
| Type | VARCHAR(9) | Attendance  types. |

Table 7 attendance management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | Access |
| Amend | Access | Access | Access |
| Archive | Access | Access | Access |
| Moniter | Access | Access | Access |
| Display | Access | Access | Access |
| Action Poor | Access | Access | Access |
| Attendence | Access | Access | Access |

1. Personal Tutor Management:

Personal tutor Management is an academic member of staff who can participate in an active pastime that benefits students' educational development and faculty experience, and which pleases them.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Id | NUMBER(17) | Id of tutor. |
| First\_Name | VARCHAR(20) | First name of tutor. |
| Last\_Name | VARCHAR(20) | Last name of tutor. |
| Phone\_Number | NUMBER(10) | Phone number of phone number. |
| Email\_Address | VARCHAR(23) | Tutor email address. |
| Salary | NUMBER(30) | Monthly salary of tutor. |
| Age | NUMBER(2) | Age of tutor. |
| Student\_id | VARCHAR(13) | Student id |
| Address | VARCHAR(18) | Tutor address. |
| Course | VARCHAR(15) | Tutor course. |

Table 8 personal tutor

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | Access |
| Assign | Access | Access | No Access |
| Display | Access | Access | Access |

1. Timetable Management:

Timetable management is a kind of plan that units the examples for extraordinary events to happen and a table that establishes the point in time stretches all through that particular occasions are envisioned to occur . It permits format the timing and participation of teachers and understudies.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Timetable\_Id | NUMBER(19) | Timetable id. |
| Course | VARCHAR(12) | Course. |
| module | VARCHAR(12) | Course’s module time table. |
| Date\_time | VARCHAR(13) | Date and time of timetable. |
| Class | NUMBER(13) | Class timetable. |
| Description | VARCHAR(19) | Description of timetable. |
| Instructor | VARCHAR(10) | Timetable instructor. |

Table 9 timetable management

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | Access |
| Delete | Access | Access | No Access |
| Assign | Access | Access | No Access |
| Display | Access | Access | Access |

1. Diary Management:

Diary management refers day to day journal and ordinary control of business undertaking pioneers. It controls all internal and outside gatherings, set up updates, report liabilities and cutoff times.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Id | NUMBER(16) | Diary id. |
| Type | VARCHAR(18) | Types of Dairy. |
| Class | NUMBER(12) | . |
| Description | VARCHAR(12) | Description of diary. |

Table 10 Diary management system

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Amend | Access | Access | Access |
| Display | Access | Access | Access |
| Prompt | Access | Access | Access |
| Initiate Automated Action | No Access | Access | No Access |

1. Report Generation / Management:

Report management system refers to a complex program of software answers that permit you to manipulate the complete felony process of reporting.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Id | INT(30) | Report id. |
| Name | VARCHAR(12) | Report name. |
| Type | VARCHAR(16) | Types of Report. |
| Date | NUMBER(15) | Report date. |
| Description | VARCHAR(20) | Description of Report. |

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Display | Access | Access | Access |
| Print | Access | Access | Access |

1. Feedback Generation / Management:

Feedback management system refers to a complex program of software answers that permit you to manipulate the complete felony process of reporting.

|  |  |  |
| --- | --- | --- |
| Attributes | Data Type | Description |
| Id | Int (16) | Feedback id. |
| Type | VARCHAR(18) | Types of Feedback |
| Date | NUMBER(12) | Feedback Date |
| Description | VARCHAR(12) | Description of Feedback. |
| Name | VARCHAR(20) | Feedback Name |

Access Rights:

The following desk indicates the records approximately the get right of entry to rights of users.

|  |  |  |  |
| --- | --- | --- | --- |
| Essential Functionality | Leader | Administrator | Lecturer |
| Create | Access | Access | No Access |
| Display | Access | Access | Access |
| Print | Access | Access | Access |

##### 2.1.3.1.2 Student Records/Information Portal

Student records/information portal is a web-fundamentally based generally entirely stage that permits schools and employees to get to understudy records online for better administration and perceivability. It helps the admissions department explains the number of students in observation and registration performance faculty. These frameworks are intended to facilitate configuration timetables and discussions among resources for undergrads.

The functions of Student records/information portal are as follow:

1. The system embeds another understudy record into the data set.
2. The record will be erased if student are graduates or exits.
3. It makes acclimations to the researcher’s data.
4. The system appearance for a specific understudy or bunch of understudies.
5. It is a system that declares to follow and monitor information from undergraduate students.

##### 2.1.3.1.3 Woodlands University College Corporate Website

This is the website of the University of Texas within the United States, registered with many Zambian institutions. the faculty offers online certificates and bachelor' credentials programs. The objective of the University website is to supply information in a really clear and open vogue all together so that clients have the best feasible experience. It works to upgrade the social transaction and verbal trade capacities of educators and undergrads through the method of methods for sending and getting messages. A website offers information on other colleges and sponsors, giving you the opportunity to find out more about the university.

The site incorporates a landing page, application, login page, self-administrations, and so on. We can achieve all pages from the route of the home site page which incorporates all realities roughly the college.

### 2.2.3 Performance Requirements

#### 2.2.3.1 Records Management System

##### Speed

It shows the speed of the system that determines the group action and transaction:

Throughput - To methodology, 100 transactions regularly require a few minutes. For 1 to 50 transactions, it usually takes roughly 30 to 40 seconds to system. It can handle 100 exchanges at one time.

##### Capacity

It suggests the number of facts that may be recorded withinside the machine and the wide variety of operations that may be executed simultaneously.

1. Over 500 clients can access web-based frameworks at the same time.

2. Maximum 250 range of data records the framework will store in just a single time.

##### Reliability

The unit of time between disappointment or an indication of system availability.

1. This internet based system are reachable for 22 hours, out of a possible 24 hours in an extremely day

2. In our ongoing system, the potential for system accessibility and margin time is a to be had a component of the reinforcement system.

**2.2.3.1.4 Usability**

Usability assessment standards are every now and then hard to define, however they need to supply a few indications of how ‘easy’ the machine is to operate.

1. Client ought to have the option to test this machine for a preparation expense of 10 hours every week without preparing for half a month.

##### 2.2.3.1.5Accessibility

The capacity of being ventured or placed into system.

1. The system access keeps track of your whole program by managing its complete information lifecycle.

#### Student Records/Information Portal

##### Speed

Actually, the speed of the service is based on the number of researchers that are dealing with it toward the start of the semester, yet it typically arrives at an ordinary level on ordinary days of the semester. For example, it takes 10 seconds to get machine data on a normal day, but it can soak up to 30-40 seconds on the first few days.

##### Capacity

Our information portal has a capacity of more than 15,000 individuals, but we only allow 11000 people to use it because the information system can only handle that many. For example, if we enable 11000 individuals to log on at the same time, the server will respond fast and in a matter of seconds. However, allowing 15000 people on the server at once may cause it to lag or slow down.

##### Reliability

The information system is obtainable for 22 hours out of 24 hours on a busy day, and we have backup plans for 2 hours just in case of downtime.

##### **Usability**

This information system is productive and compelling, and clients are satisfied with utilizing it to achieve their goals. An information system is convenience decides anyway rapidly and easily it will follow through with its jobs.

##### **2.2.3.2.5 Accessibility**

The information system is capable of being accessed and understood via way of means of the character or institution for which it's miles designed.

#### Woodlands University College Corporate Website

##### Speed

The speed of the university website is determined according to different conditions because there are one-of-a-kind areas comprising of getting section to the page, application, acknowledgment, and others which are client based and tedious. For example, if there are 100 to 200 college students seeking to observe it takes 10-20 seconds, however at the beginning days, it takes 2 to 3 min to use due to the fact the scholars may have greater than 500 college students.

##### Capacity

The website's ability is based on how many site visitors it can manage at any given moment. The range of clients is 500 who can get admission to this device simultaneously.

##### Reliability

Quality, which capacities dependably or in every case appropriately at the site, is the dependability of the site. On our site framework, it will be functional for hundred and sixty hours out of a likely 168 hours consistently.

##### Usability

The usability of a university website is based on accessibility, lucidity, mindfulness, dependability, and significance. Our site is appropriate for each registering gadget and cell gadget. Because of the developing utilization of cell web, clients could likewise moreover have a couple of site capacities on their phones and work areas.

##### Accessibility

The devices and innovations at the university website are planned and progressed to allow individuals with incapacities to apply PC structures for all, regardless of the kind of recognition of inability.

### Design Constraints:

There isn't any prerequisite for a specific design or specific configuration.

Since the client has now not covered or forced any configuration requirement, a PC with the least equipment and programming program application capacities can help access this web webpage on the web. Thus, we have not found any exact limitations, considering the utilization of a straightforward programming program that sudden spikes in demand for every running design and approaches all gadgets. We noticed no exact limits because of the reality the purchaser wasn't forcing or such any design necessities. We lease an essential programming program application utility this is similar to all running designs and offers us get admission to the entirety of our gadgets.

Ordinary plan imperatives for the most frameworks include:

Local or Distributed architecture:

The parts of the system can be used in disconnected phases of spread or local plan, and the parts can connect with one another over a correspondence game plan to accomplish their goals. The inclinations of this designing are sharing of PC program resources, versatility while using PC programs from various venders, improvements, and equal dealing with to gain ground in execution. Counting resources can make progress throughput to be sure after a slip-up occurs. This incorporate will continue to work.

Target Operation Systems:

For the objective system, the gear and program designs to keep notwithstanding the goal machine are recorded. Contains a troublesome and speedy of machine program libraries and unmistakable reports which you essentially set up to your goal machine. Microsoft Windows, Mac OS, Linux,and Android are the best notable and focused running designs.

Front ends Styles of Graphics:

Front-end designs are a secluded blend of all your item's purchaser interface parts, whole with case code that the developer can imitate and glue as required. Among various buyer interface parts, it comprises of images, buttons, way menus, and shape enter parts.

Required hardware requirements:

CPU: Intel core I3 and above

Moniter:14” LCD

Internal Storage:100GB

Operating system: Windows/Mac

Memory: 16GB RAM

Network Adapter:812.21ac.4/5GHz adapter

### Commercial Constraints:

Information of Assignment:

Total length of Project: 5 week

Hourly Cost: $50000 per person

Number of members in group:5 members

|  |  |  |  |
| --- | --- | --- | --- |
| SECTIONS Of PROJECT | WEEKS TAKEN | HOURS WEEK PER PERSON IN PROJECT | TOTAL COST OF WHOLE SECTIONS |
| Requirements Specification | 2 | 5 | $10,00,00,000 |
| Design and Analysis | 1 | 5 | $22,00,00,000 |
| System Interface Designs | 1 | 4 | $50,00,00,000 |
| Requirement Engineering | 1 | 4 | $67,00,000 |
| Problem Domain | 1 | 3 | $65,00,000 |

|  |  |
| --- | --- |
| REASON | COST |
| Software cost | $98,00,00,000 |
| Base cost | $44,14,00,000 |
| Total cost of build | $21,14,00,000 |
| Cost of Office | $65,65,00,000 |
|  |  |
| Other expenses (40%) | $4,23,40,00,000 |
| Profit margin (10%) | $20,45,00,000 |
| Cost Of Total Project | $90,00,00,00,00,500,000 |

### Acceptance Tests (Total Project):

# System Analysis & Design (Records Management System)

It is a method of collection and deciphering facts, characteristic the issues, and decomposition of a system into its elements.

System analysis is conducted for the aim of finding out a system or its components so as to spot its objectives. it's a haul determination technique that improves the system and ensures that each one the elements of the system work with efficiency to accomplish their purpose.

## 3.1 Preliminary Design Stages

Preliminary analysis is the first step in a project's development that determines whether the concept is practical. The goals of preliminary data analysis are to update the data in order to prepare it for further analysis, characterize the data's essential qualities, and report the results. This stage also aids in determining whether or not any present issues exist. This component implements the system analysis and design documents.

Evaluating the reliability of measurements, evaluating the efficiency of any changes, analyzing the distribution of independent factors, and identifying outliers are all preliminary analysis stages.

### 3.1.1 Textual Analysis

Textual analysis, often known as content analysis, is a way of extracting information from a text. With our Textual Analysis tool, you may utilize a rich-text document editor to document a project, a user problem, or other textual information, and then extract model elements from the text.

Using *Textual Analysis* compile a list of candidate classes and routines for the course

management system.

|  |  |
| --- | --- |
| **Category** | **Category Identification** |
| Students | Create, Amend, Archive, Display, Assign. |
| Department | Create, Amend, Archive, Display, Assign. |
| Staffs | Create, Amend, Archive, Display, Assign. |
| Course | Create Structure, Amend, Display, Delete, Archive. |
| Modules | Create, Amend, Delete, Archive, Display, Assign. |
| Personal Teacher | Create, Amend, Assign, Display, |
| Timetable | Create, Amend, Delete, Archive, Display |
| Attendance | Create, Amend, Archive, Monitor, Display, Action Poor Attendance. |
| Assignment | Create, Amend, Delete, Archive, Display, Assign, Mark/Grade |
| Diary management | Create, Amend, Display, Prompt, Initiate Automated Action. |
| Feedback | Create, Display, Print |
| Report | Create, Display, Print |

Table 11 textual analysis

### **3.1.2 Significant Event Analysis**

This is the process which include the classes with the lead performer who is handling the classes in real scenario and even the attributes of the class which will be the taken value to be stored in the database.

Admin

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Admin login | Administration | Username, password |
| Admin logout | Administration |  |

Table 12 admin

Student

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Student login | Student, administration | Username, password |
| Student logout | Student |  |
| Enroll Student | Administration | Student\_ID, firstname, familyname,  middlename, E-mail, Address,  DOB, age, Telephone No, gender,  year, message |
| Display student | Administration, student | Student\_ID, Name, E-mail, Address,  DOB, age, Telephone No, gender,  year |
| Modify student | Administration | Name, E-mail, age, Address, DOB,  Telephone No, gender |
| Assign student | Administration | Course, module |
| Archive student | Administration | Student\_ID |

Table 13

Staff

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Staff login | Staff, administration | Username, password |
| Staff logout | Staff, administration |  |
| Enroll Staff | Administration | Staff\_ID, firstname, familyname,  middlename, E-mail, phone\_  number, qualification, address,  Gender, responsibility, age,  Department |
| Display staff | Administration, staff | Staff\_ID, firstname, familyname,  middlename, E-mail, phone\_  number, qualification, address,  Gender, responsibility, age,  Department |
| Modify staff | Administration | Name, E-mail, phone\_  number, qualification, Gender,  responsibility, age, Department |
| Assign staff | Administration | Course\_Module |
| Archive student | Administration | Staff\_ID |

Table 14 staff

Department

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add department  Display Department | Administration | Department\_id, Name, email, Telephone No. |
| Modify Department | Administration | Department id, Name, email, Telephone No. |
| Assign Department | Administration | Course, module, staff |
| Delete Department  Archive Department | Administration | Department\_ID |

Table 15 department

Course

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Course | Administration | Course \_id, Name, description, time\_table\_id, credits |
| Display Course | Administration, staff | Course \_id, Name, description, time\_table\_id, credits |
| Modify Course | Administration, department | Name, description, time\_table\_id, credits |
| Delete Course  Archive Course | Administration | Course \_ID |

Table 16 course

Module

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Module | Administration | Module \_id, Name, description, credits, code, duration, course\_ID |
| Display Module | Administration, staff | Module \_id, Name, description, credits, code, duration, course\_ID |
| Modify Module | Administration, department | Name, description, credits, duration |
| Assign Module | Administration, department | Staff, student, time\_table. Attendance, assignment |
| Delete Module  Archive Module | Administration | Module \_ID |

Table 17 module

Assignment

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Assignment | Administration | Assignment \_id, Name, description, duration, module\_ID, result |
| Display Assignment | Administration, staff | Assignment \_id, Name, description, duration, module\_ID |
| Modify Assignment | Administration, department | Name, description, duration |
| Assign Assignment | Administration, department | Staff, student, time |
| Mark/Grade | Module\_staff | Assignment\_id |
| Delete Assignment  Archive Assignment | Administration, department | Assignment \_ID |

Table 18 assignment

Attendance

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Attendance | Administration | Attendance \_id, Name, description,  Date, student\_id, module\_ID, |
| Display Attendance | Administration, staff | Attendance \_id, Name, description,  Date, student\_id, module\_ID, |
| Modify Attendance | Administration, department | Name, description, Date |
| Assign Attendance | Administration, department | Staff, student, module |
| Monitor Attendance  Action poor attendance  Archive Attendance | Administration, department | Attendance \_ID |

Table 19 attendance

Personal\_tutor

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Create personal\_tutor | Administration | Staff\_id, student\_id, module\_id |
| Display personal\_tutor | Administration, course, staff | Staff\_id, student\_id, module\_id |
| Modify personal\_tutor | Administration | Staff\_id, student\_id, module\_id |
| Assign personal\_tutor | Administration, department | Student, module |

Table 20 personal tutor

Time\_Table

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Timetable | Administration, department | Timetable \_id, Name, description, module\_id, course\_id |
| Display Timetable | Administration, department | Timetable \_id, Name, description, module\_id, course\_id |
| Modify Timetable | Administration, department | Name, description |
| Delete Timetable  Archive Timetable | Administration | Timetable \_ID |

Table 21

Diary Management

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Diary  Display Diary | Department, student | Diary \_id, Name, description, module\_id, course\_id, student\_id, staff\_id |
| Modify Diary | Department, student | Name, description |
| Prompt diary  Initiated automated action | Student | Diary \_ID |

Table 22

Report Generation

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Report  Display Report | Administration, Department | Report \_ID, Name, description, module\_id, course\_id, student\_id, staff\_id |
| Delete Report | Administration, Department | Report\_ID |

Table 23

Feedback

|  |  |  |
| --- | --- | --- |
| Event | Performers | Attributes |
| Add Feedback  Display Feedback | Administration, Department, staff, personal\_tutor | Feedback \_ID, Name, description, module\_id, course\_id, student\_id, staff\_id |
| Delete Feedback | Administration, Department, staff, personal\_tutor | Feedback \_ID |

Table 24 feedback

### 3.1.3 Commands Queries and Constraints.

This analysis method helps to bring all the requirements that is to be used in the course management system where the queries, commands and constraints stand for:

• Queries – lists applicable queries.

• Commands – lists applicable commands.

• Constraints – lists consistency requirement of the class and its operations as well as general business rules and other information that may affect the design and implementation of the class.

The CQC chart is given below:

This is for the overall system since individual class will be explained in the class chart below.

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | Course management system | | Part: 1/1 |
| TYPE OF OBJECT  **Stores all records within the course index.** | | INDEXING  Created:  05/16/2022 | |
| Queries | Count | | |
| Commands | make, add\_student, add\_staff, add\_course, add\_modules, add\_timetable, add\_attendence, add\_assignment, add\_personal\_Tutor, add diary, add\_feedback,add\_report, delete, amend, display\_by\_name, display\_by\_id, display\_by date, open\_database, close\_datebase, get\_new\_student\_id | | |
| Constraints | * *At creation the list must be empty.* * *Before any index can be deleted the list must exist.* * *An index cannot be deleted from the list if it does not exist within the list.* * *After an index is deleted from the list the number of elements in the list is reduced by one.* * *Before an index can be added to the list the list must exist.* * *A valid, unique, id code must be supplied for each new index being added to the list.* * *After the index has been added to the list the number of elements in the list is increased by one.* * *Before index details can be amended the index list must exist.* * *A valid unique id must be supplied for any given index before index data can be amended.* * *The index must exist in the list before it can be amended.* * *The number of elements in the index list does not change after the amend operation has been performed.* * *Before index can be displayed the index list must exist.* * *A valid unique id, name must be supplied before an index can be displayed.* * *The index must exist in the list before it can be displayed.* * *The number of elements in the index does not change after the display operation has been performed.* * *Only valid detail can be added to the index list.* * *The list can never contain a negative detail.* | | |

## 3.2 Detailed Static System Designs

The static model explains the system's classes, their interfaces, how they are connected to one another, and how they are clustered. It depicts the system architecture as well as the agreements that each class component has with its clients.

### 3.2.1 First Draft BON system architecture diagram

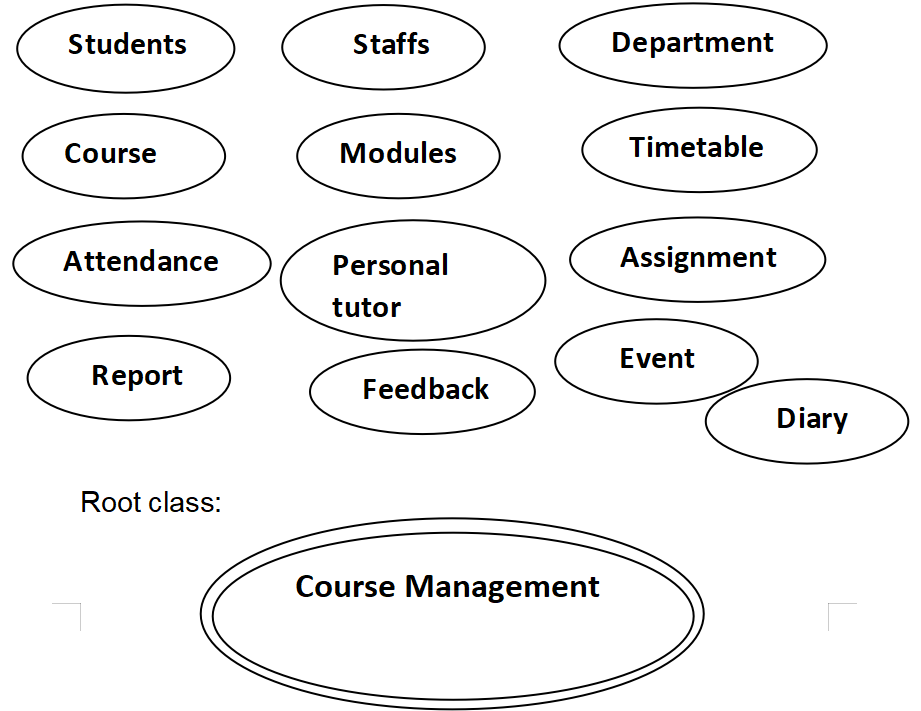
The structure, behavior, and many other properties of a system are defined by its architecture. It contains a set of concepts for modeling object-oriented software, as well as a supporting notation (one graphical and one textual) and a set of rules and guidelines for developing the models. This architecture is made up of three parts: Class, Root Class, and Cluster.

Class Diagram:

It is made up of structural characteristics that represent the state of a class object and behavioral features that define how objects interact. On an elliptical circle, the classes are displayed.

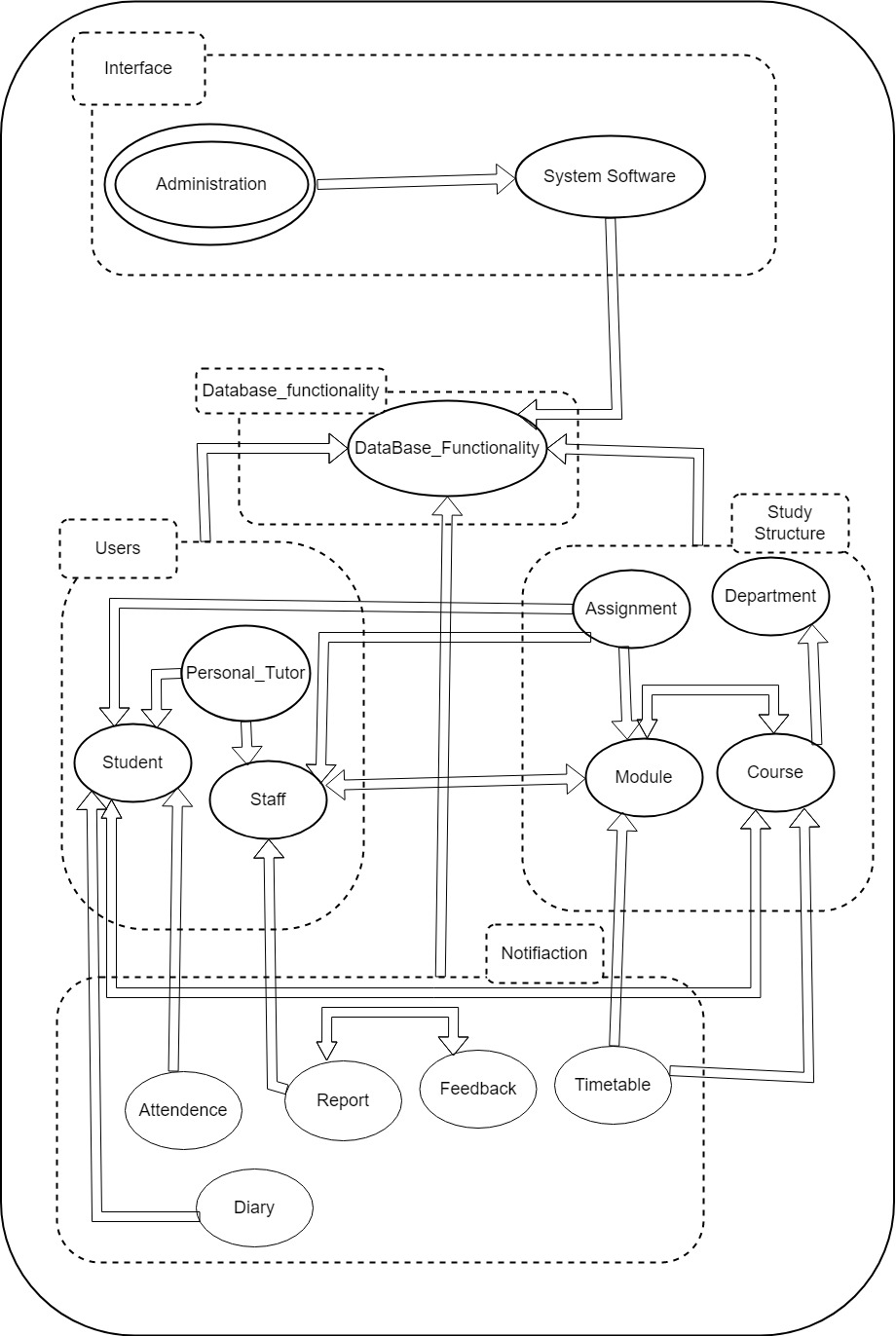
Our course management system consists of students, staffs, department, course, modules, timetable, attendance, exam, assignment, report feedback, event class.

These are the classes used in this project:



When an object-oriented process is started, one instance of this class is created, and the root class's initialization procedure controls the execution.

Overall BON architecture diagram



Set of diagrams representing possibly nested clusters, class headers, and their relationships. Bird’s eye view of the system (zoomable).

|  |  |
| --- | --- |
| Purpose:  For having the complete detail of the people who are connected to the university as teacher, student. | Indexing:  Author: Ritesh Koirala  Keyword: Records |

|  |  |  |
| --- | --- | --- |
| SYSTEM | Course management System | Part:1/1 |

|  |  |
| --- | --- |
| **CLUSTER** | **DESCRIPTION** |
| Users | People who are responsible to use and make the important use of the web application. |
| Study Structure | This is the part of the university who will be responsible for managing the study for student. |
| Notification | Generally, the parts of user and department to maintain its integrity. |
| Database\_Functionality | Keeps all the record to be extracted by the users. |

### 3.2.2 BON System chart:

The system chart includes a summary of each top-level cluster. A BON system is made up of one or more clusters, each with its own set of classes

### 3.2.3 BON Cluster Charts

A cluster chart describes each class and sub-cluster in the cluster briefly. Sub-cluster names are enclosed in parenthesis to distinguish them from class names. The recommended approach is to list all classes first, then any sub-clusters. This is because local services required by a cluster's highest classes are typically aggregated by sub-clusters. A sub-cluster, for example, could contain all of a class's specialized descendent classes.

Users

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | Users | | Part 1/1 |
| PURPOSE:  Keeps the detail of the people who are related with the university | | INDEXING  Author – Ritesh Koirala  Created – 23 May, 2022 | |
|  | | | |
| Class/(Cluster) | | Description | |
| Student | | Taken as the main source of university to be using a website. | |
| Staff | | People responsible for handling the various activity done by student. | |
| Personal\_Tutor | | Responsible for taking the classes and manage the student. | |
| (Admin) | | Who monitors all the activity done by the anyone on the web application and maintain it. | |

Study Structure

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | Study Structure | | Part 1/1 |
| PURPOSE:  Handles all the course and teaching material of the student and teacher. | | INDEXING  Author – Ritesh Koirala  Created – 23 May, 2022 | |
|  | | | |
| Class/(Cluster) | | Description | |
| Department | | General department abstraction. | |
| Course | | To manage each course that is taught in the university. | |
| Module | | Generally, the topics of the main things to be taught to the student | |
| Assignment | | .Work given by the staff according to the modules to check the level of syudent. | |

Notification

|  |  |  |  |
| --- | --- | --- | --- |
| CLUSTER | Notification | | Part 1/1 |
| PURPOSE:  The major key point of the university to know the detail or progress of study. | | INDEXING  Author – Ritesh Koirala  Created – 23 May, 2022 | |
|  | | | |
| Class/(Cluster) | | Description | |
| Attendance | | The track the daily present of the student. | |
| Report | | The portion which shows the progress of the student. | |
| Feedback | | Response of the report that they have received. | |
| Diary | | A document that shows the daily work document of the study. | |

### 3.2.4 BON Class Charts

The class charts are used to represent individual classes. The data in the class charts is obtained from the responses to the following questions, which are viewed as black boxes.

• What knowledge can this class impart to other classes? As a result, queries relevant to the class are generated.

• What services could this class be asked to provide to other classes? This results in commands that are unique to each class.

• What are the rules that the class, as well as its customers, must adhere to? As a result, the class has limitations.

The three thing that is included in the chart:

• Queries – lists applicable queries.

• Commands – lists applicable commands.

• Constraints – lists consistency requirement of the class and its operations as well as general business rules and other information that may affect the design and implementation of the class.

Student

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STUDENT | | Part 1/1 |
| TYPE OF OBJECT:  The key component of university. | | INDEXING  cluster: STUDENT  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, age, gender, address, e-mail, course, telephone number, DOB . | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must write the same name which is in the citizenship card. 2. All the detail should be filled with zero null. 3. Must be enrol in any one course. 4. Should be equal or more than 18 years old. | |

Table 25 Bohn class chart

Department

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | DEPARTMENT | | Part 1/1 |
| TYPE OF OBJECT:  The part which manages to distribute courses and manage it. | | INDEXING  cluster: DEPARTMENT  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, email, course, telephone number, staff\_id , student\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must have at least one course. 2. Must have teacher enrolled in every course. 3. Must have help-service email and number. | |

Staff

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | STAFF | | Part 1/1 |
| TYPE OF OBJECT:  The qualified people who educate other. | | INDEXING  cluster: STAFF  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, age, gender, address, e-mail, course, telephone number, DOB , level\_of\_study, experience. | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must write the same name which is in the citizenship card. 2. All the detail should be filled with zero null. 3. Must have 2year experience. 4. Should have completed the bachelor degree. | |

Table 26

Course

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | COURSE | | Part 1/1 |
| TYPE OF OBJECT:  The programs that is taught in university. | | INDEXING  cluster: COURSE  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, email, staff\_id, Student\_id, Time\_period | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Student must be enrolled in one course. 2. Must have more than one module. 3. Must have course leader in each course. 4. Must have exams. | |

Table 27

Module

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | MODULES | | Part 1/1 |
| TYPE OF OBJECT:  The subject that is taught in course. | | INDEXING  cluster: MODULES  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, email, staff\_id, Student\_id, Time\_period, lessons | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Modules should have more than 5 students. 2. Must have a practical lab. 3. Must have at least 2 tutors. 4. Must have exams. | |

Table 28

Timetable

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | TIMETABLE | | Part 1/1 |
| TYPE OF OBJECT:  The plan for conducting classes in time. | | INDEXING  cluster: TIMETABLE  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id, class | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must not be scheduled in the holidays. 2. Must have all the modules included. 3. Should be from 10am to 4pm time within a day. 4. At least one module should repeat 2 time in a week. | |

Table 29

Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ATTENDENCE | | Part 1/1 |
| TYPE OF OBJECT:  Checking the daily presence of student. | | INDEXING  cluster: ATTENDENCE  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id, class | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Should have the name of every student listed. 2. Must have date in the top. 3. Must be daily checked by the department. 4. Must complain if the student is most often absent. | |

Table 30

Personal Tutor

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | PERSONAL TUTOR | | Part 1/1 |
| TYPE OF OBJECT:  The work to check the level of student. | | INDEXING  cluster: Personal tutor  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id, student\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must have assign in any course. 2. Can be for multiple students. 3. Should give time for one student at a time. | |

Table 31

Diary Management

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | DIARY MANAGEMENT | | Part 1/1 |
| TYPE OF OBJECT:  The work to check the level of student. | | INDEXING  cluster: Diary Management  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id, student\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must have record of every subject. 2. Must be written every day. 3. Must be signed by the parent and class teacher. 4. Must be in good position. 5. Diary should be neat and clean. | |

Table 32

Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | ASSIGNMENT | | Part 1/1 |
| TYPE OF OBJECT:  The work to check the focus of student on class. | | INDEXING  cluster: ASSIGNMENT  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must have at least of one month of the time to complete the assignment. 2. Must be strictly implement the code of plagiarism while checking paper. 3. Tutor must give the brief of the assignment. 4. Late submission is not allowed. | |

Table 33

Report

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | REPORT | | Part 1/1 |
| TYPE OF OBJECT:  The result of the exam and assignment. | | INDEXING  cluster: REPORT  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Parent must be present to take report. 2. Must be given in average grade of all module. 3. If fail have to repeat module. 4. Student must recheck paper and report if something isn’t right in report. | |

Table 34

Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| CLASS | FEEDBACK | | Part 1/1 |
| TYPE OF OBJECT:  The response of the exam/assignment. | | INDEXING  cluster: FEEDBACK  created: 05/31/2022 | |
|  | | | |
| Queries | | Name, ID, course\_id, module\_id | |
| Commands | | Create, edit, delete, amend, assign, archive | |
| Constraints | | 1. Must have the signature of teacher. 2. Once received by the student should be signed by the parents. 3. Should mainly focus on the weak point of student. 4. Student must improve according to report. | |

Table 35

## 3.3 Detailed Dynamic System Designs

Although analysis and design do not entail implementation, they do include ensuring that execution is feasible at a fair cost and with appropriate product performance. As a result, even in the early phases of development, we need at least a hazy understanding of how the operations of our high-level classes can satisfy their specifications by invoking other operations (in the same or in other classes). The objective of the BON dynamic model is to aid in the capturing and communication of this concept.

### 3.3.1 Event Charts

External events eventually cause the object interactions that make up a system's execution. As a result, it's typically a good idea to construct a list of external events that could cause object communication and so reflect different forms of system behavior. The event list is then used as a reference for selecting situations for inclusion in a system dynamic model. However, only a small percentage of all conceivable external occurrences are noteworthy and representative enough to be included in the list.

For example, before getting admission in the collage you go for the counselling to know about the college for that you call the college first then they ask your detail, review it, then set you a schedule to have a counselling, this is an internal even but after your call they fix the time and person you will be getting counselling and call you back which is external event.

Internal/External event

|  |  |  |  |
| --- | --- | --- | --- |
| EVENTS | STUDENT\_SUPPORT | | Part 1/2 |
| COMMENT  Representative sorts of conduct are triggered by internal and external circumstances. | | INDEXING  Created: 2022-06-02  revised: 2022-06-02 | |
|  | | | |
| External(incoming) | | Involved object types | |
| Request to login to the system. | | LOGIN | |
| Request to add a new student record | | STUDENT, COURSE | |
| Request to edit a student record | | STUDENT, COURSE | |
| Request to archive the student record | | STUDENT, COURSE | |
| Request to view the details of a student | | STUDENT, COURSE | |

Table 36 Events

|  |  |  |  |
| --- | --- | --- | --- |
| EVENTS | COURSE\_SUPPORT | | Part 1/2 |
| COMMENT  Representative sorts of conduct are triggered by internal and external circumstances. | | INDEXING  Created: 2022-06-02  revised: 2022-06-02 | |
|  | | | |
| External(incoming) | | Involved object types | |
| Request to login to the system. | | LOGIN | |
| Request to add a new course record | | COURSE, MODULE | |
| Request to edit a course record | | COURSE, MODULE | |
| Request to archive the course record | | COURSE, MODULE | |
| Request to view the details of a course | | COURSE, MODULE | |

|  |  |  |  |
| --- | --- | --- | --- |
| EVENTS | MODULE\_SUPPORT | | Part 1/2 |
| COMMENT  Representative sorts of conduct are triggered by internal and external circumstances. | | INDEXING  Created: 2022-06-02  revised: 2022-06-02 | |
|  | | | |
| External(incoming) | | Involved object types | |
| Request to login to the system. | | LOGIN | |
| Request to add a new module record | | COURSE, STAFF, MODULE | |
| Request to edit a module record | | COURSE, STAFF, MODULE | |
| Request to archive the module record | | COURSE, STAFF, MODULE | |
| Request to view the details of a module | | COURSE, STAFF, MODULE | |

|  |  |  |  |
| --- | --- | --- | --- |
| EVENTS | STAFF\_SUPPORT | | Part 1/2 |
| COMMENT  Representative sorts of conduct are triggered by internal and external circumstances. | | INDEXING  Created: 2022-06-02  revised: 2022-06-02 | |
|  | | | |
| External(incoming) | | Involved object types | |
| Request to login to the system. | | LOGIN | |
| Request to add a new staff record | | STAFF, MODULE | |
| Request to edit a staff record | | STAFF, MODULE | |
| Request to archive the staff record | | STAFF, MODULE | |
| Request to view the details of a staff | | STAFF, MODULE | |

|  |  |  |  |
| --- | --- | --- | --- |
| EVENTS | ASSIGNMENT\_SUPPORT | | Part 1/2 |
| COMMENT  Representative sorts of conduct are triggered by internal and external circumstances. | | INDEXING  Created: 2022-06-02  revised: 2022-06-02 | |
|  | | | |
| External(incoming) | | Involved object types | |
| Request to login to the system. | | LOGIN | |
| Request to add a new assignment record | | ASSIGNMENT, MODULE, STAFF | |
| Request to edit a assignment record | | ASSIGNMENT, MODULE, STAFF | |
| Request to archive the assignment record | | ASSIGNMENT, MODULE, STAFF | |
| Request to view the details of a assignment | | ASSIGNMENT, MODULE, STAFF | |

### 3.3.2 Object Creation Chart

The study of the formation of new items could provide a link between the static and dynamic models. Some classes in a static diagram are delayed, which implies they contain operations that will never be implemented in the class itself, but will only be implemented in descendant classes. Because such classes will never be created, objects of the same type will never appear in the dynamic model.

|  |  |  |  |
| --- | --- | --- | --- |
| **CREATION** | **COURSE\_MANAGEMENT\_SYSTEM** | | **PART 1/1** |
| **COMMENT**  These are the listing of creating instance in the system. | | Created – 10 May, 2022 | |
| **Class** | | **Creates instances of** | |
| DEPARTMENT | | COURSE - | |
| STUDENT | | COURSE | |
| STAFF | | MODULE | |
| COURSE | | MODULE | |
| MODULE | | STAFF, COURSE | |
| ASSIGNMENT | | MODULE, STUDENT | |
| ATTENDANCE | | STUDENT | |
| PERSONAL\_TUTOR | | STAFF, STUDENT | |
| TIMETABLE | | COURSE, MODULE | |
| DIARY | | STUDENT, STAFF | |
| REPORT\_GENERATION | | STUDENT, STAFF | |
| Feedback | | STUDENT, STAFF | |

Table 37 object creation chart

### 3.3.3 System Scenario Charts

We can select a set of relevant system situations to highlight essential features of the overall system behavior using the system event chart as a starting point. The system's scenario chart then collects a brief description of each scenario (from which a dynamic diagram can be generated later). Each entry has two short descriptions: the first is a few words that can also be used as a scenario name for easy reference, and the second is a few phrases that go into greater detail about the scenario.

Scenario chart for counselling and event

|  |  |  |  |
| --- | --- | --- | --- |
| SCENARIO | STUDENT\_SUPPORT | | Part: 1/1 |
| COMMENT  Set of representative scenarios to show important types of system behavior. | | INDEXING  created: 2022-06-02 | |
|  | | | |
| **Check if the Id and password is correct:**  These fields are validated against the database when the user Id and password have been entered. And if it is, you are logged in; if not, you are unable to do so. | | | |
| **Create a student record:**  To store the data, all required fields must be filled out completely. The database contains all of the entered data. The course class is used for one field. | | | |
| **Edit the record just created:**  The selected record in the database is updated as necessary, and all information that has been recorded about that Student is shown. | | | |
| **Archive the created record:**  The database record is then archived alongside all other previously kept records. | | | |
| **View the record:**  You can look up information about the selected student in the database. Each and every detail from that database is visible. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| SCENARIO | COURSE\_SUPPORT | | Part: 1/1 |
| COMMENT  Set of representative scenarios to show important types of system behavior. | | INDEXING  created: 2022-06-02 | |
|  | | | |
| **Check if the Id and password is correct:**  These fields are validated against the database when the user Id and password have been entered. And if it is, you are logged in; if not, you are unable to do so. | | | |
| **Create a Course record:**  To store the data, all required fields must be filled out completely. The database contains all of the entered data. The course class is used for one field. | | | |
| **Edit the record just created:**  The selected record in the database is updated as necessary, and all information that has been recorded about that course is shown. | | | |
| **Archive the created record:**  The database record is then archived alongside all other previously kept records. | | | |
| **View the record:**  You can look up information about the selected course in the database. Each and every detail from that database is visible. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| SCENARIO | MODULE\_SUPPORT | | Part: 1/1 |
| COMMENT  Set of representative scenarios to show important types of system behavior. | | INDEXING  created: 2022-06-02 | |
|  | | | |
| **Check if the Id and password is correct:**  These fields are validated against the database when the user Id and password have been entered. And if it is, you are logged in; if not, you are unable to do so. | | | |
| **Create a module record:**  To store the data, all required fields must be filled out completely. The database contains all of the entered data. The course class is used for one field. | | | |
| **Edit the record just created:**  The selected record in the database is updated as necessary, and all information that has been recorded about that module is shown. | | | |
| **Archive the created record:**  The database record is then archived alongside all other previously kept records. | | | |
| **View the record:**  You can look up information about the selected module in the database. Each and every detail from that database is visible. | | | |

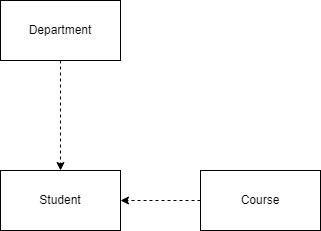
|  |  |  |  |
| --- | --- | --- | --- |
| SCENARIO | STAFF\_SUPPORT | | Part: 1/1 |
| COMMENT  Set of representative scenarios to show important types of system behavior. | | INDEXING  created: 2022-06-02 | |
|  | | | |
| **Check if the Id and password is correct:**  These fields are validated against the database when the user Id and password have been entered. And if it is, you are logged in; if not, you are unable to do so. | | | |
| **Create a staff record:**  To store the data, all required fields must be filled out completely. The database contains all of the entered data. The course class is used for one field. | | | |
| **Edit the record just created:**  The selected record in the database is updated as necessary, and all information that has been recorded about that staff is shown. | | | |
| **Archive the created record:**  The database record is then archived alongside all other previously kept records. | | | |
| **View the record:**  You can look up information about the selected staff in the database. Each and every detail from that database is visible. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| SCENARIO | ASSIGNMENT\_SUPPORT | | Part: 1/1 |
| COMMENT  Set of representative scenarios to show important types of system behavior. | | INDEXING  created: 2022-06-02 | |
|  | | | |
| **Check if the Id and password is correct:**  These fields are validated against the database when the user Id and password have been entered. And if it is, you are logged in; if not, you are unable to do so. | | | |
| **Create an assignment record:**  To store the data, all required fields must be filled out completely. The database contains all of the entered data. The course class is used for one field. | | | |
| **Edit the record just created:**  The selected record in the database is updated as necessary, and all information that has been recorded about that assignment is shown. | | | |
| **Archive the created record:**  The database record is then archived alongside all other previously kept records. | | | |
| **View the record:**  You can look up information about the selected assignment in the database. Each and every detail from that database is visible. | | | |

### 3.3.4 Dynamic Diagram

We're now ready to talk about the BON dynamic notation, which is employed in system scenarios. A dynamic diagram is made up of a group of communicating objects that exchange messages. An object's type, which is its class name, and an optional object qualifier are used to represent it.

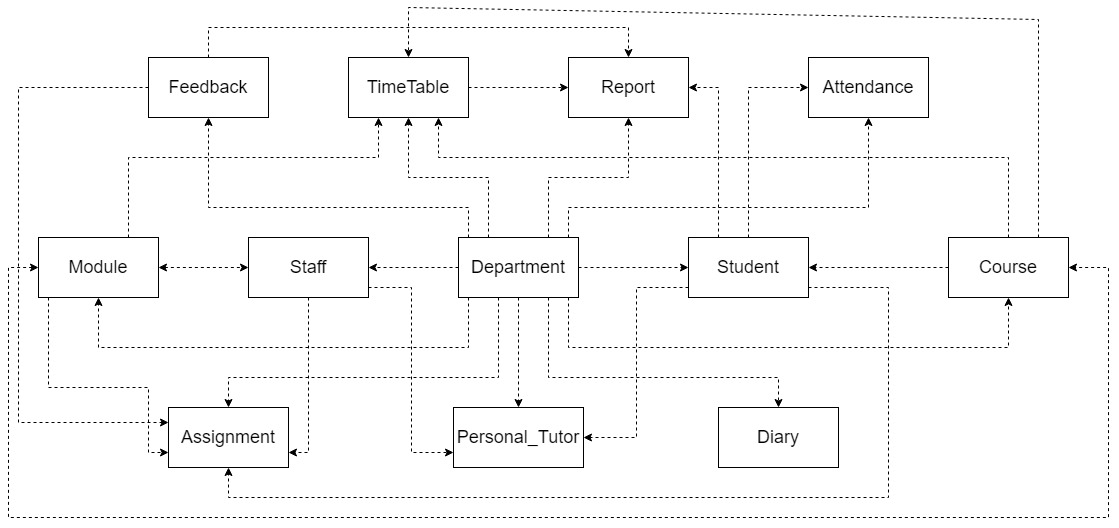
|  |
| --- |
| Scenario:  1. Accessible student file.  2. Request the course ID.  3. Include the course ID in the student record.  4. Present the student record. |



BON Dynamic Diagram of Student record

Here is an overall scenario of the course management system as well as its dynamic diagram.

|  |
| --- |
| Scenario:  1. Open student after logging into the system.  2. Use the course ID that is provided.  3. Showcase the student.  4. Examine the staff.  5. Add the module Id.  6. Present staff.  7. get course.  8. Open the module list.  9. Show the course.  10. Open the module.  11. Import course ID from course and staff ID from staff.  12. Screen module  13. Get task  14. Access the staff, module, and student IDs.  15. Show Assignment.  16. Pick your attendance.  17. Acquire a student ID.  18. Indicate attendance.  19. Open-access private tutor.  20. Import student ID and inherit from staff.  21. Display tutor.  22. Verify the schedule.  23. Access the module and course IDs.  24. Display a schedule.  25. Open a diary.  26. Student access to the section.  27. Show your diary.  28. choose report generation.  29. Use staff and student IDs.  30. Present the report. |



BON Dynamic Diagram

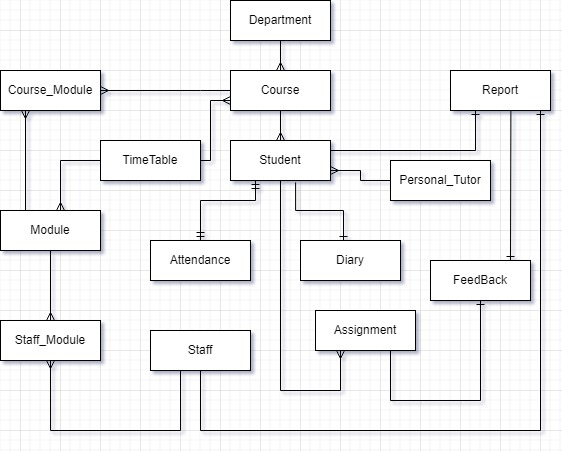
System Database Design

Database design refers to a set of procedures that make it easier to plan, create, implement, and maintain enterprise data management systems. A well-designed database is simple to manage, increases data consistency, and saves money on disk storage space.

ER – DIAGRAM

Entity Relationship Diagram (ERD) or Entity Relationship Diagram (ERD) is a diagram that shows the relationship between entity sets contained in a database. In other words, ER diagrams aid in the explanation of database logical structure. Entities, attributes, and relationships are the three main notions that ER diagrams are built on.

Rectangles are used to represent entities, ovals are used to describe characteristics, and diamond shapes are used to show relationships in ER Diagrams.



Attribute Listing

This list all the attributes that has been listed in the above classes.

admin

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| admin\_id | NUMBER(8) | PK |  |
| password | VARCHAR(20) | NOT NULL |  |

student

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id | NUMBER(8) | PK |  |
| firstname | VARCHAR(50) | NOT NULL |  |
| middlename | VARCHAR(50) |  |  |
| familyname | VARCHAR(50) | NOT NULL |  |
| address | VARCHAR(50) | NOT NULL |  |
| Telephone number | VARCHAR(15) | NOT NULL |  |
| Year\_joined | VARCHAR(4) | NOT NULL |  |
| course\_id\* | NUMBER(8) | FK, NOT NULL |  |
| email | VARCHAR(50) | NOT NULL |  |
| Message | VARCHAR(200) | NOT NULL |  |

staff

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id | NUMBER(8) | PK |  |
| firstname | VARCHAR(100) | NOT NULL |  |
| middlename | VARCHAR(100) | NOT NULL |  |
| lastname | VARCHAR(100) | NOT NULL |  |
| address | VARCHAR(100) | NOT NULL |  |
| Gender | CHECK | NOT NULL |  |
| Age | NUMBER(3) | NOT NULL |  |
| Phone\_number | VARCHAR(13) | NOT NULL |  |
| Email | VARCHAR(100) | NOT NULL |  |
| responsibility | VARCHAR(100) | NOT NULL |  |
| module\_id\* | NUMBER(10) | FK, NOT NULL |  |
| Department\_id | NUMBER(10) | FK, NOT NULL |  |

Department

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| department\_id | NUMBER(5) | PK |  |
| department\_name | VARCHAR(100) | NOT NULL |  |

course

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id | NUMBER(5) | PK |  |
| course\_name | VARCHAR(100) | NOT NULL |  |
| Course\_credit | NUMBER(5) | NOT NULL |  |
| description | VARCHAR(100) | NOT NULL |  |

module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| module\_id | NUMBER(5) | PK, AUTO INCRE |  |
| module\_name | VARCHAR(100) | NOT NULL |  |
| description | VARCHAR(100) | NOT NULL |  |
| credit | NUMBER(5) | NOT NULL |  |
| duration | DATE | NOT NULL |  |
| Course\_id | NUMBER(5) | FK, NOT NULL |  |

staff\_module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id\* | NUMBER(5) | PK, FK, NOT NULL |  |
| module\_id\* | NUMBER(5) | PK, FK, NOT NULL |  |

course\_module

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id\* | NUMBER(5) | PK, FK, NOT NULL |  |
| module\_id\* | NUMBER(5) | PK, FK, NOT NULL |  |

assignment

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| assignment\_id | NUMBER(5) | PK,AUTO INCRE |  |
| assignment\_name | VARCHAR(100) | NOT NULL |  |
| description | VARCHAR(1000) | NOT NULL |  |
| module\_id\* | NUMBER(5) | FK, NOT NULL |  |
| duration | DATE | NOT NULL |  |

attendance

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(5) | FK |  |
| student\_name | VARCHAR(100) | NOT NULL |  |
| Year | VARCHAR(4) | NOT NULL |  |
| Today’s\_date | DATE | NOT NULL |  |
| Module\_id | NUMBER(5) | NOT NULL |  |
| attendance\_percent | NUMBER(5) | NOT NULL |  |

personal\_tutor

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| staff\_id\* | NUMBER(5) | FK, NOT NULL |  |
| student\_id\* | NUMBER(5) | FK, NOT NULL |  |

timetable

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| course\_id\* | NUMBER(5) | FK, NOT NULL |  |
| day | VARCHAR(20) | NOT NULL |  |
| time | TIME | NOT NULL |  |
| module\_id\* | NUMBER(5) | FK |  |

report

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(5) | FK |  |
| staff\_id\* | NUMBER(5) | FK |  |
| report | BINARY(2000) | NOT NULL |  |

diary

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| id | NUMBER(5) | PK |  |
| Daily\_activity | VARCHAR(300) | NOT NULL |  |
| schedule | VARCHAR(100) | NOT NULL |  |
| student\_id\* | NUMBER(5) | FK |  |

Feedback

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Datatype | Constraint | Default |
| student\_id\* | NUMBER(5) | FK |  |
| staff\_id\* | NUMBER(5) | FK |  |
| Feedback | VARCHAR(200) | NOT NULL |  |

# **4 System Interface Designs:**

## **4.1 Draft Interface Designs (Records Management) :**

### **4.1.1 Wireframes:**

Login Page for admin system:

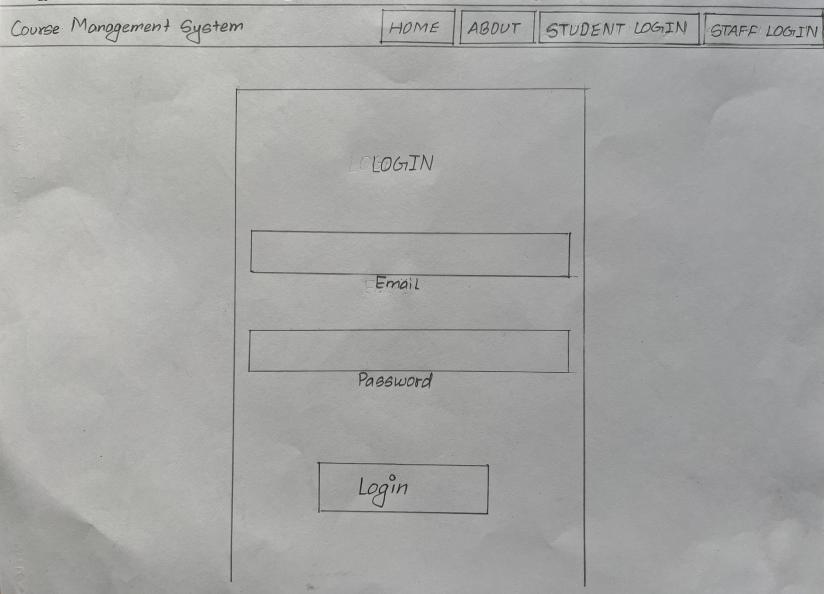


Figure 4 Admin Login (wireframe)

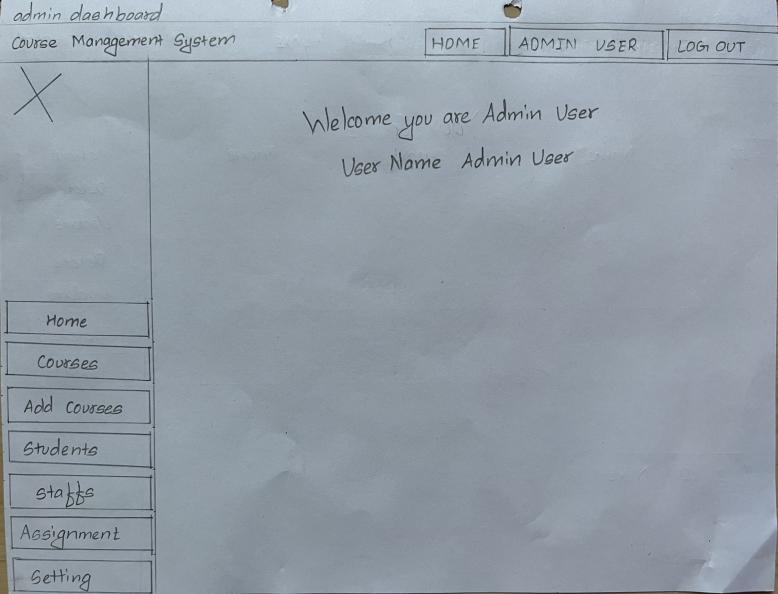


Figure 5 Admin Dashboard (wireframe)

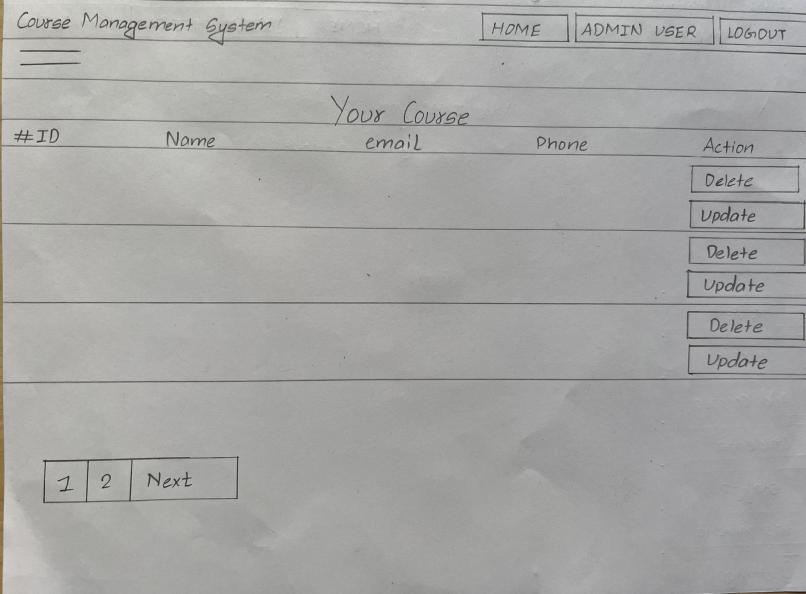


Figure 6 Show course (wireframe)

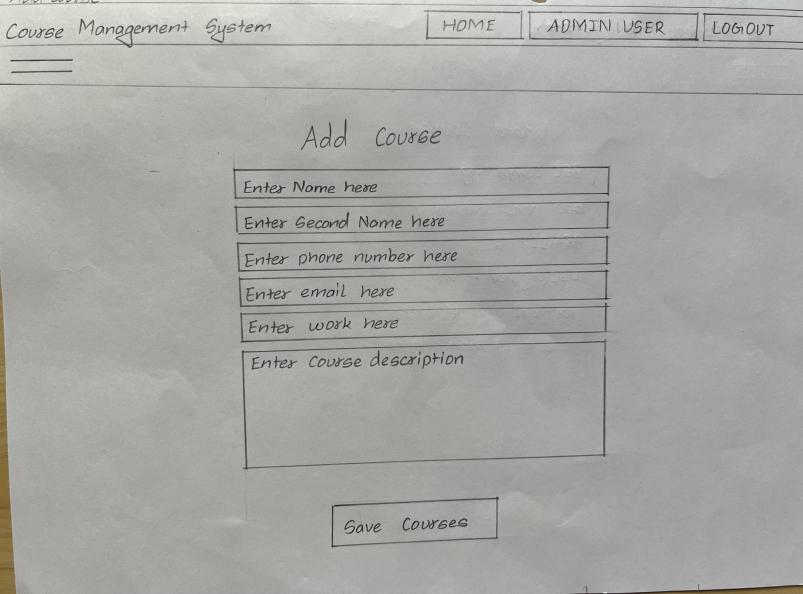


Figure 7 Add course ( wireframe)

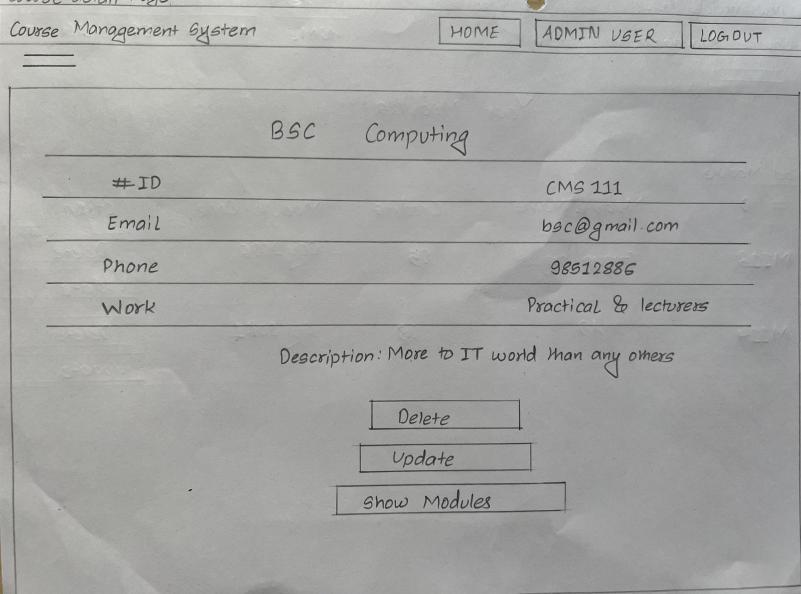


Figure 8 show course details (wire-frame)

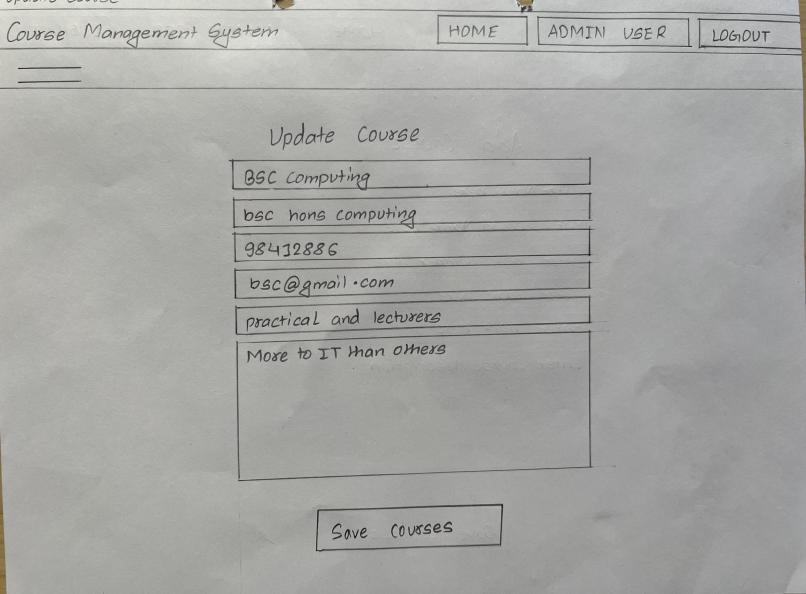


Figure 9 Update course (wire-frame)

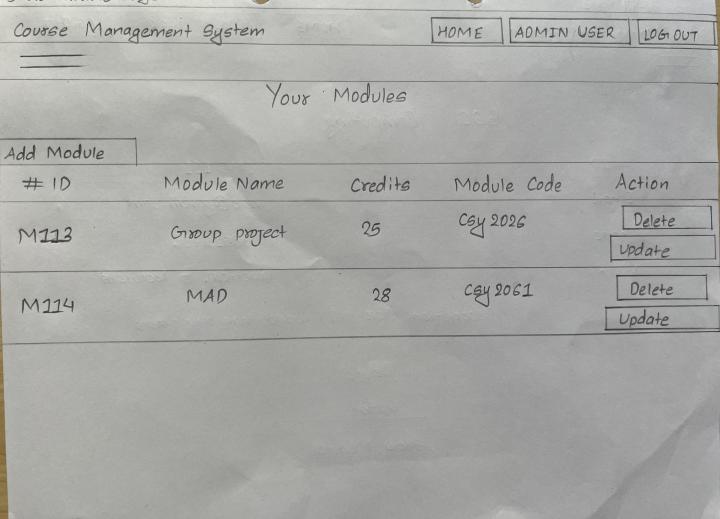


Figure 10 Show modules page(wire-frame)

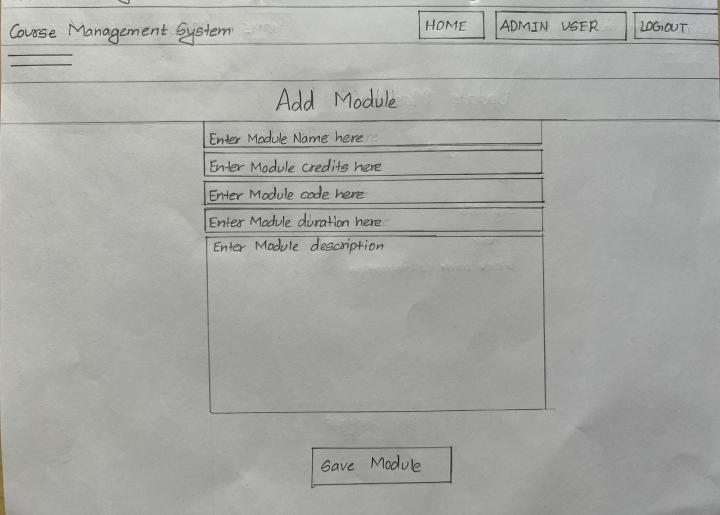


Figure 11 Add module page (wire-frame)

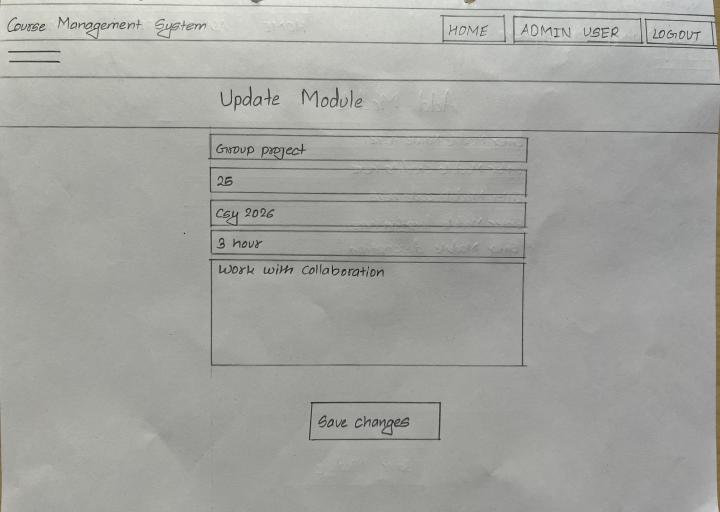


Figure 12 Update module(wire-frame)

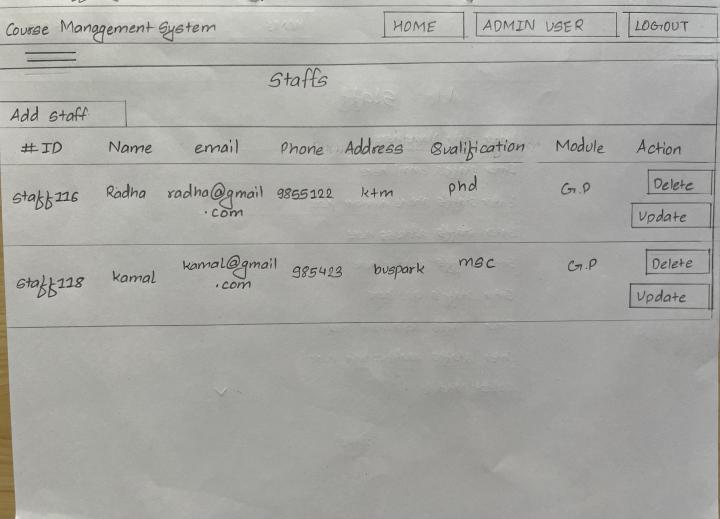


Figure 13 Show staffs page(wire-frame)

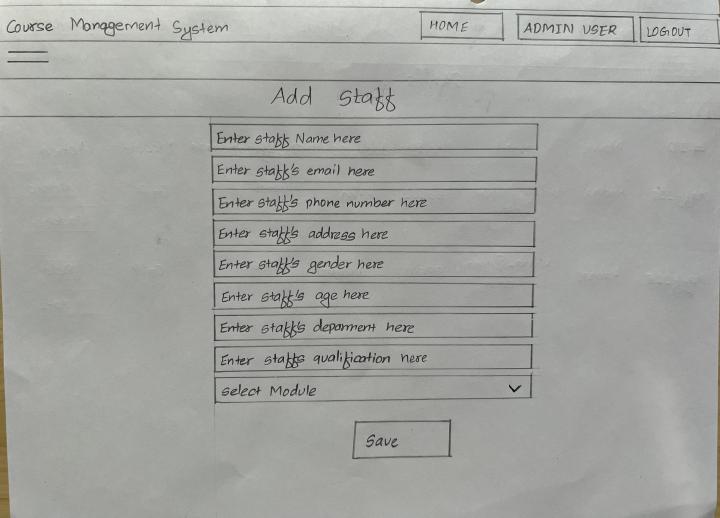


Figure 14 Add staff page (wire-frame)

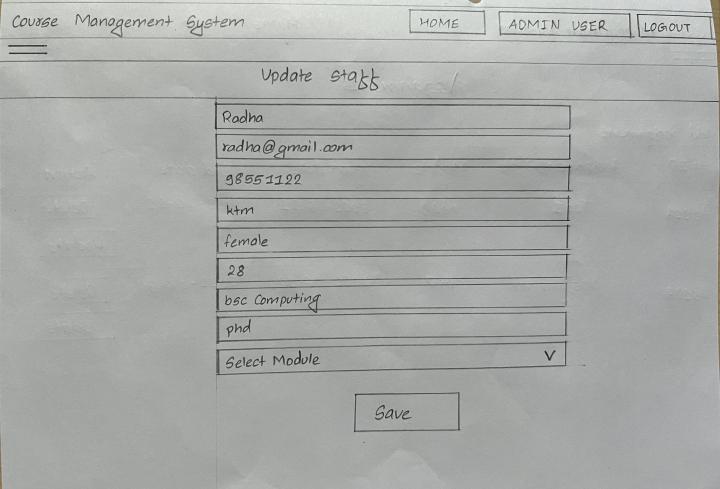


Figure 15 Update Staff (wire-frame)



Figure 16 Show Assignments(wireframe)

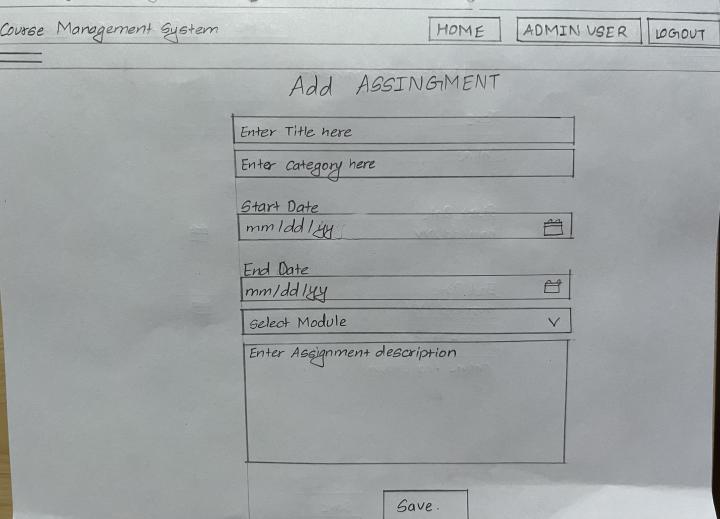


Figure 17 Add assignment(wireframe)

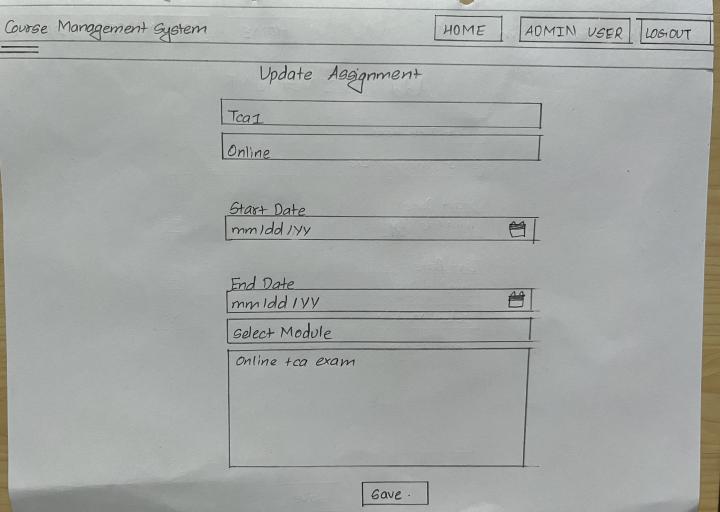


Figure 18 Update Assignment(wireframe)

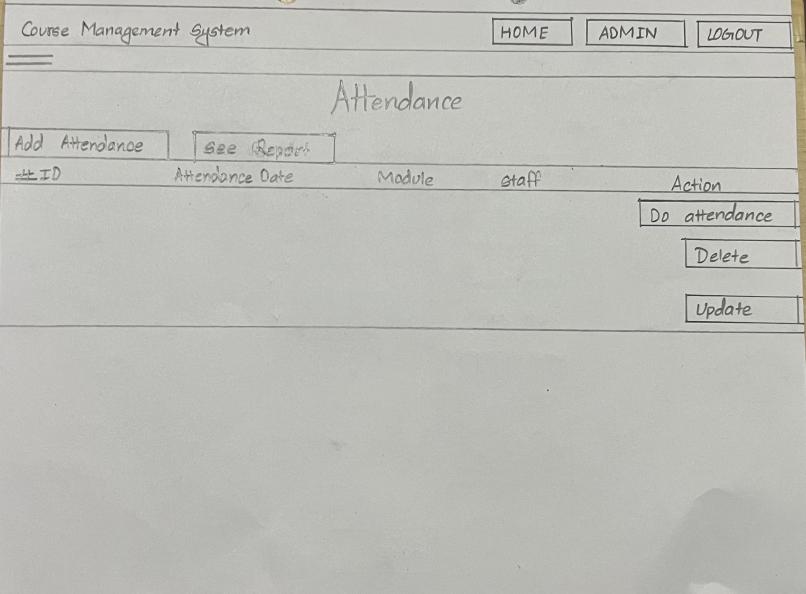
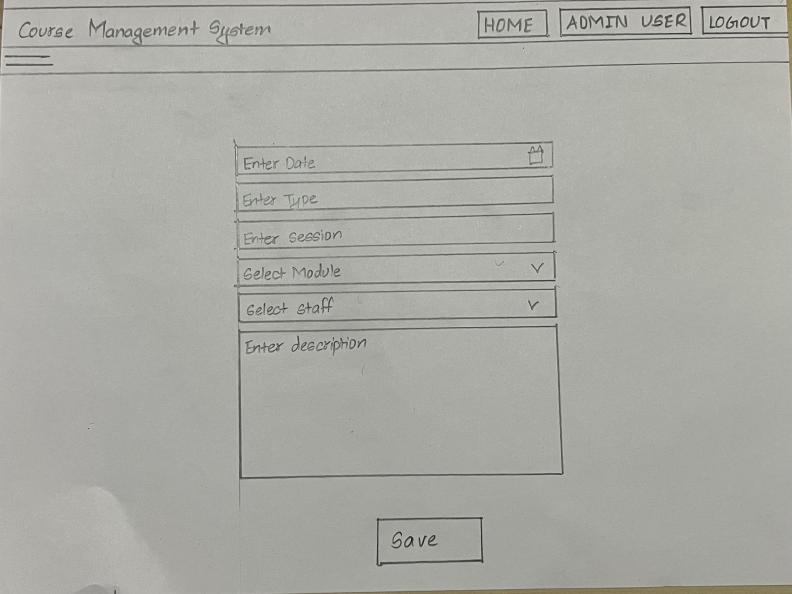


Figure 19 Show Attendance



**Fig: Add attendance**

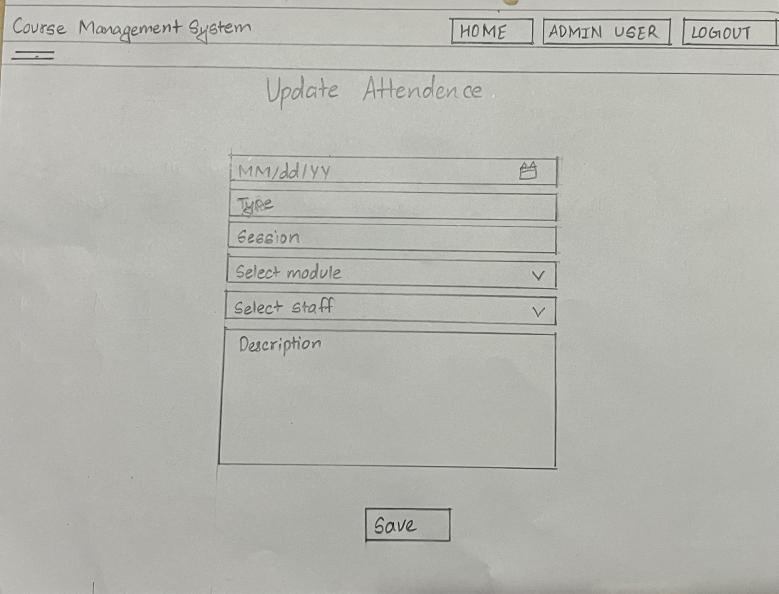


Figure 20 Update Attendance

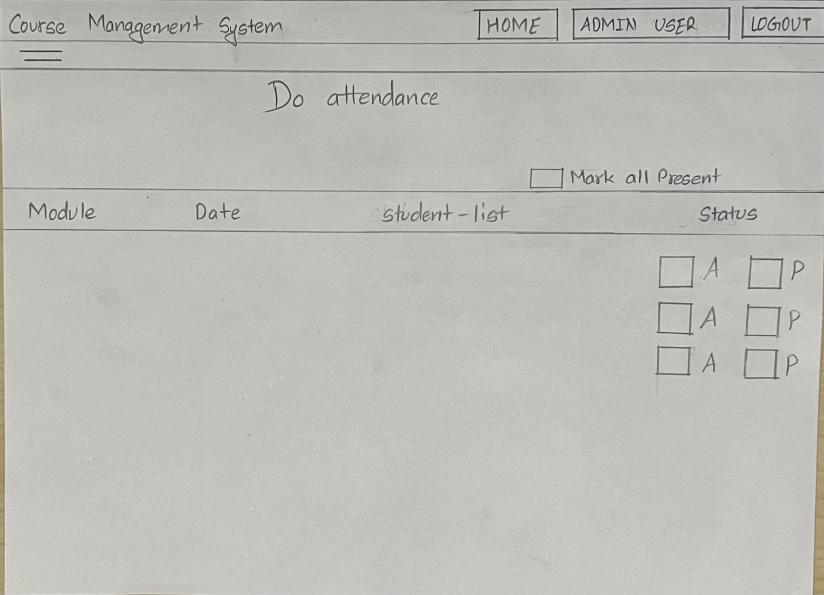


Figure 21 Do attendance

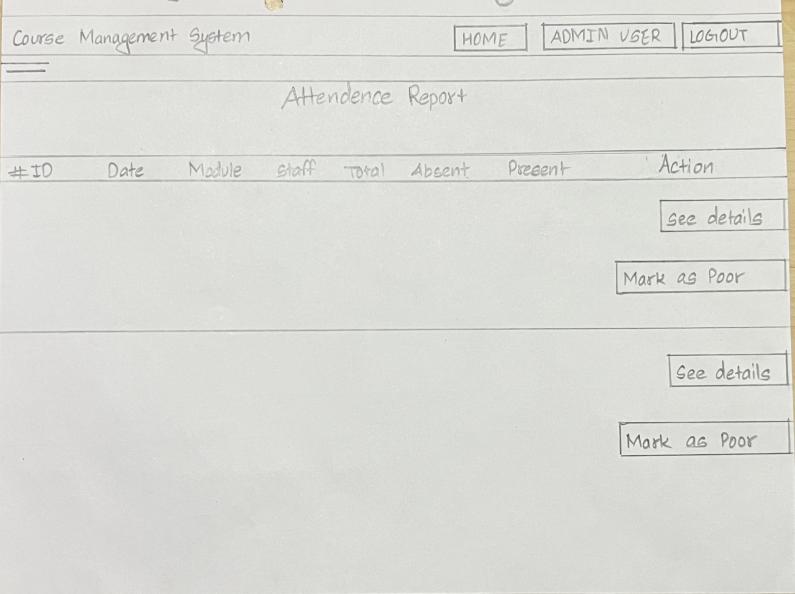


Figure 22 Show Attendance Report

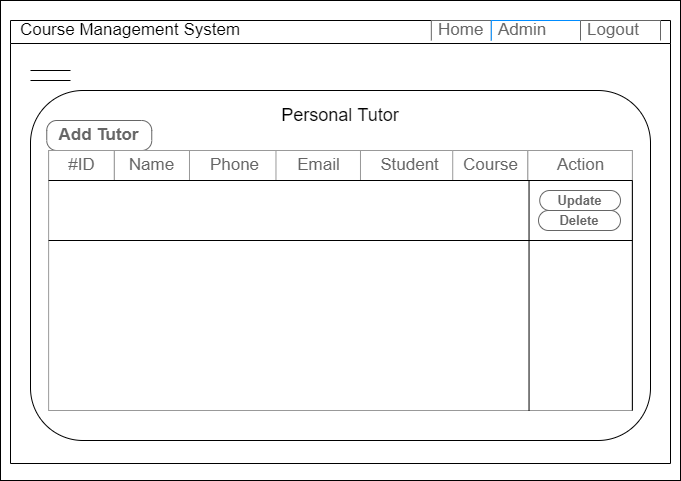
****

Figure 23 Show Personal Tutor (wire-frame)

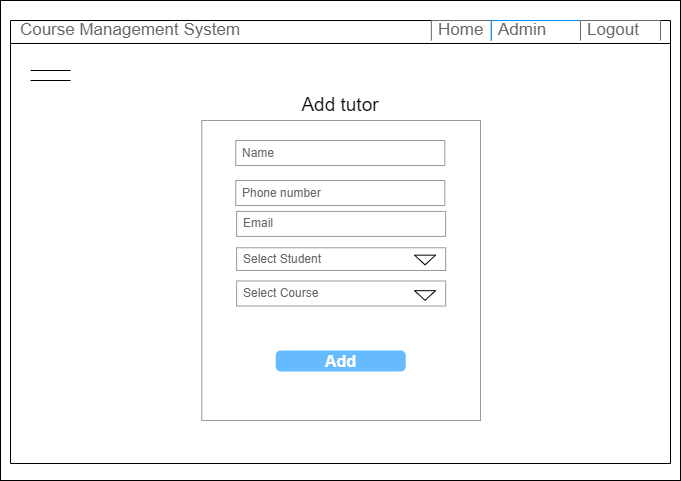
****

Figure 24 Add Tutor

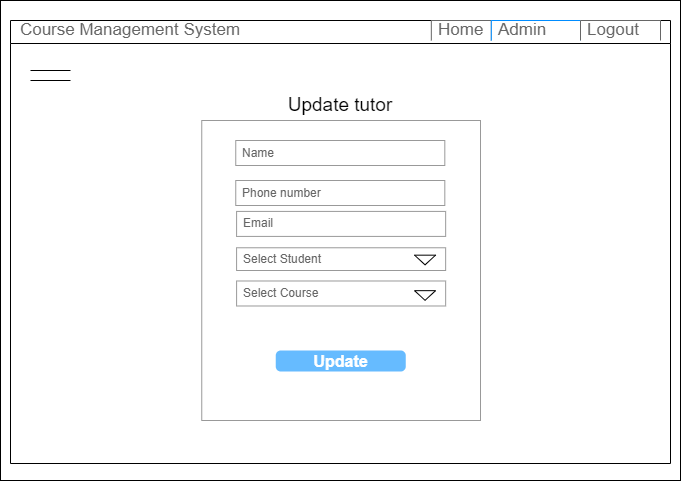
****

Figure 25 Update Tutor

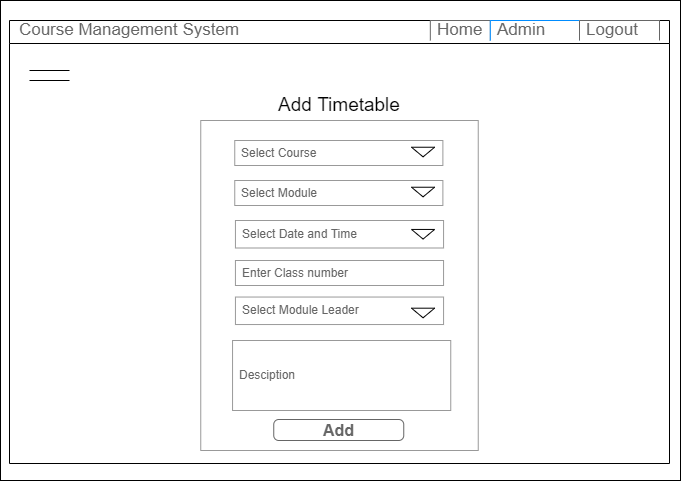
****

Figure 26 Add Timetable

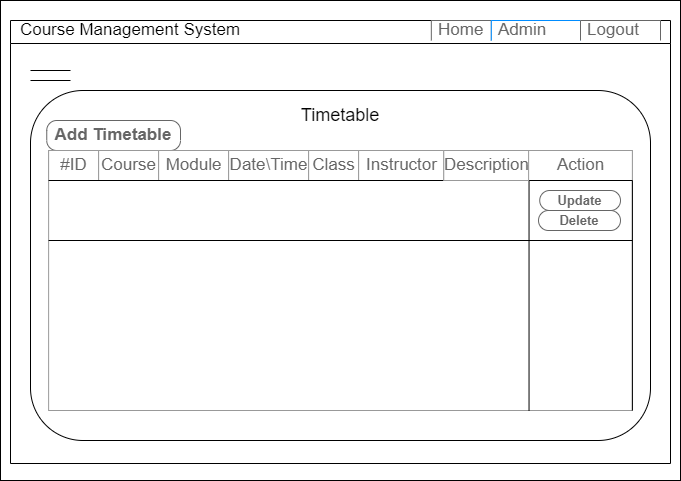
****

Figure 27 Show Timetable

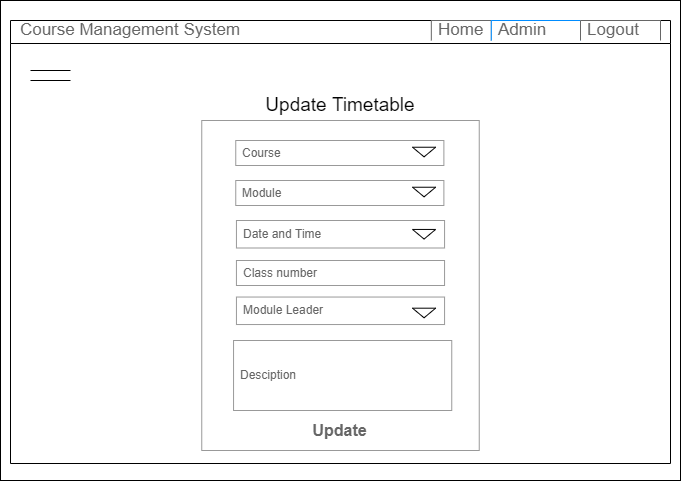
****

Figure 28 Update Timetable

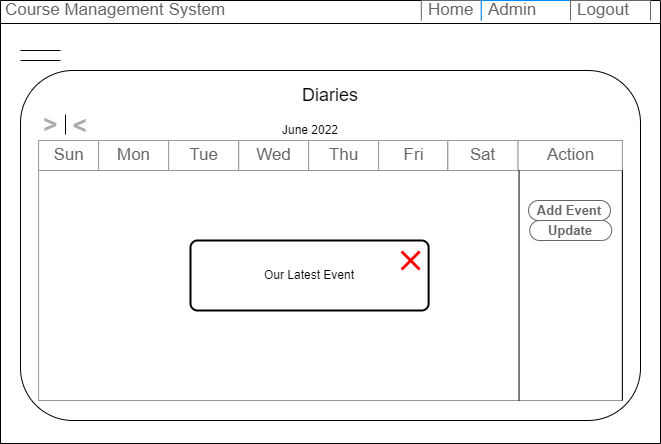
****

Figure 29 Show Diaries

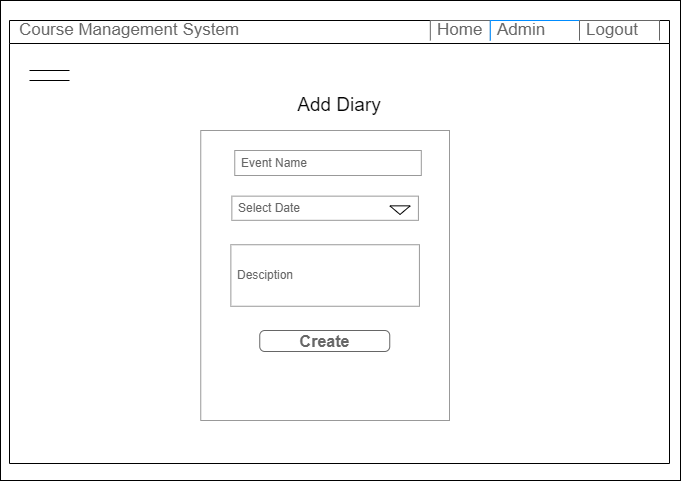
****

Figure 30 Add Event/Diary Page

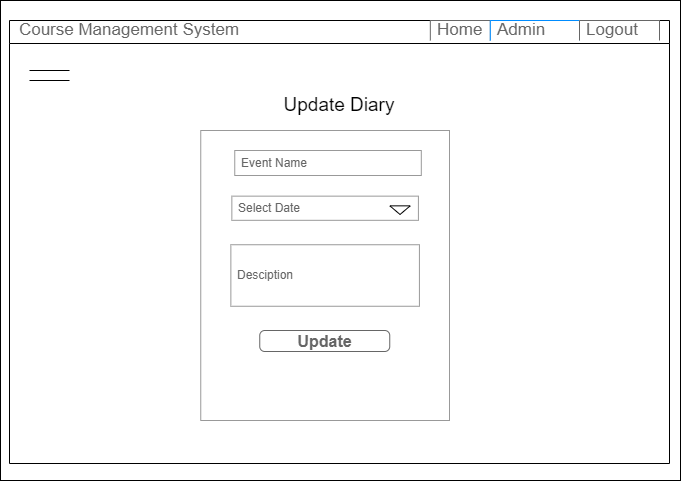
****

Figure 31 Update Diary

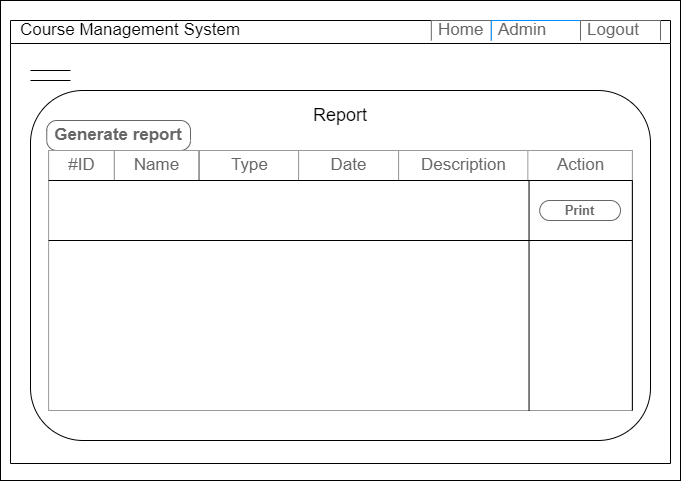
****

Figure 32 Report Page (wireframe)

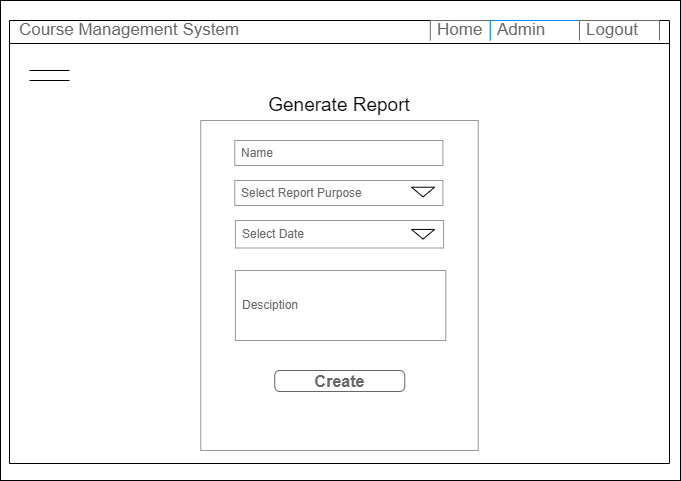
****

Figure 33 Report Page (wire-frame)

**4.1.2 System Navigation Diagram**

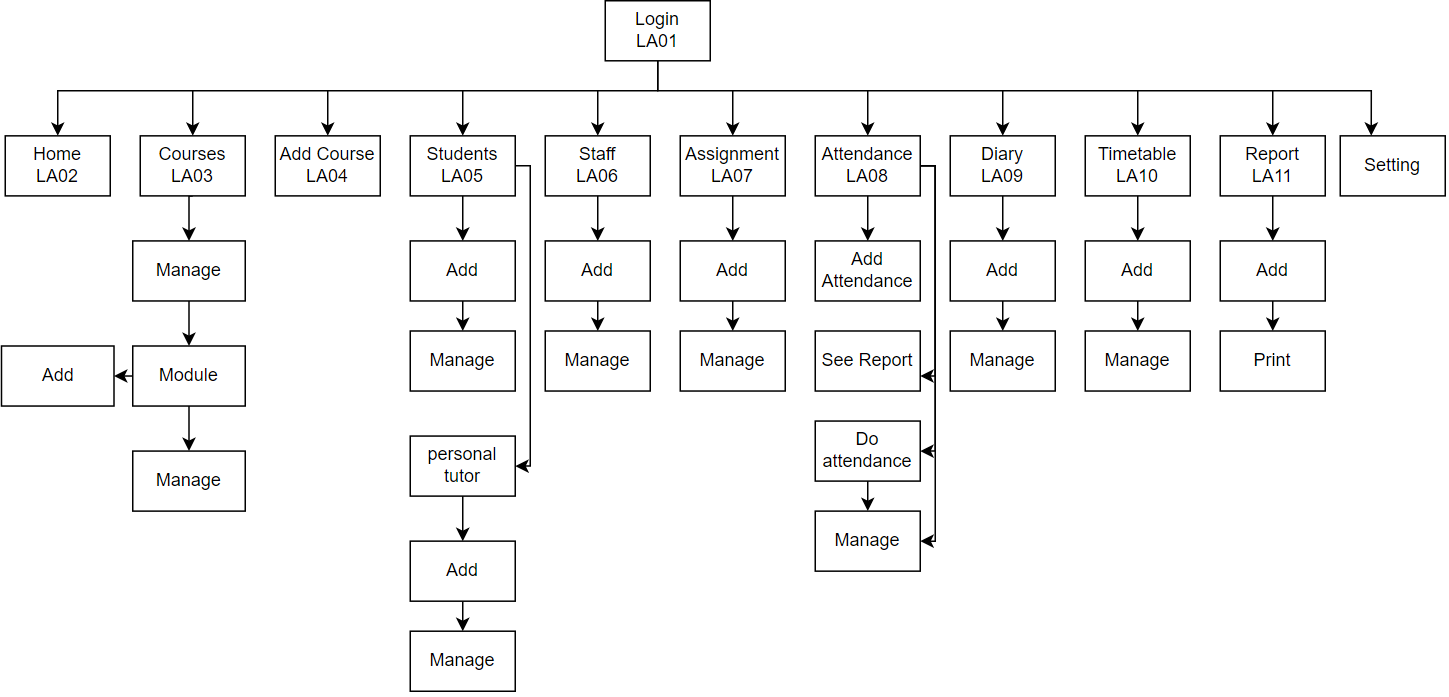
****

Figure 34 System Navigation Diagram of Record Management System

### 4.1.3 System Screen Mock-ups:

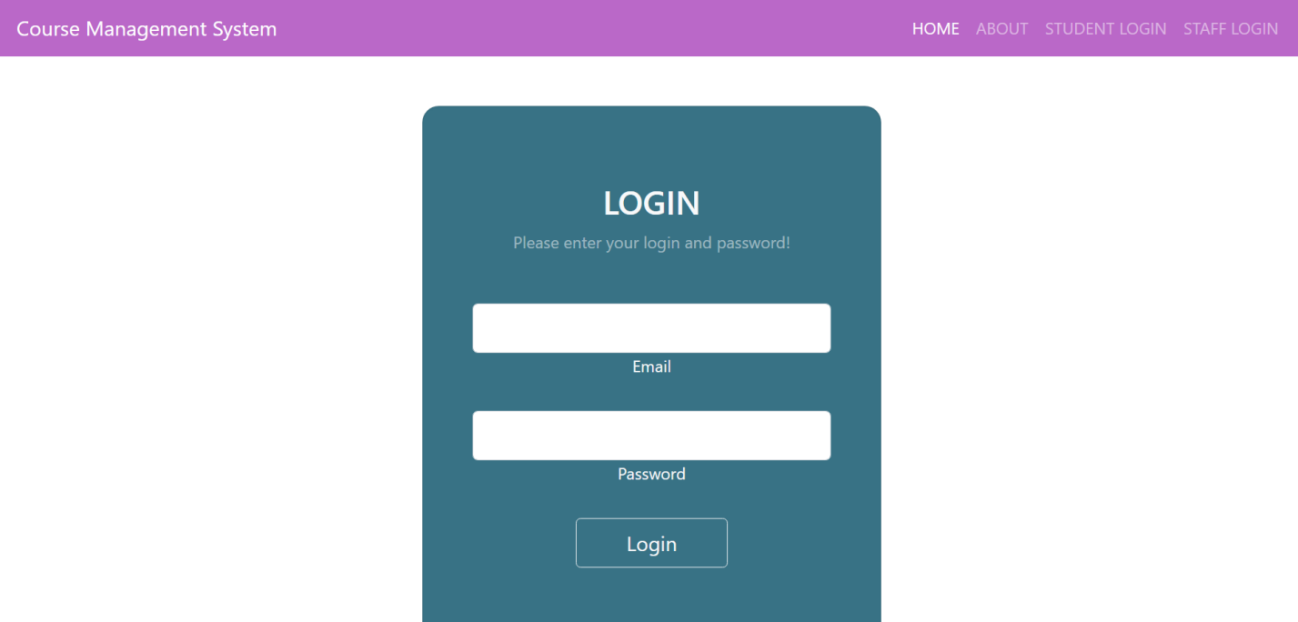


Figure 35 Admin login (mock-up)

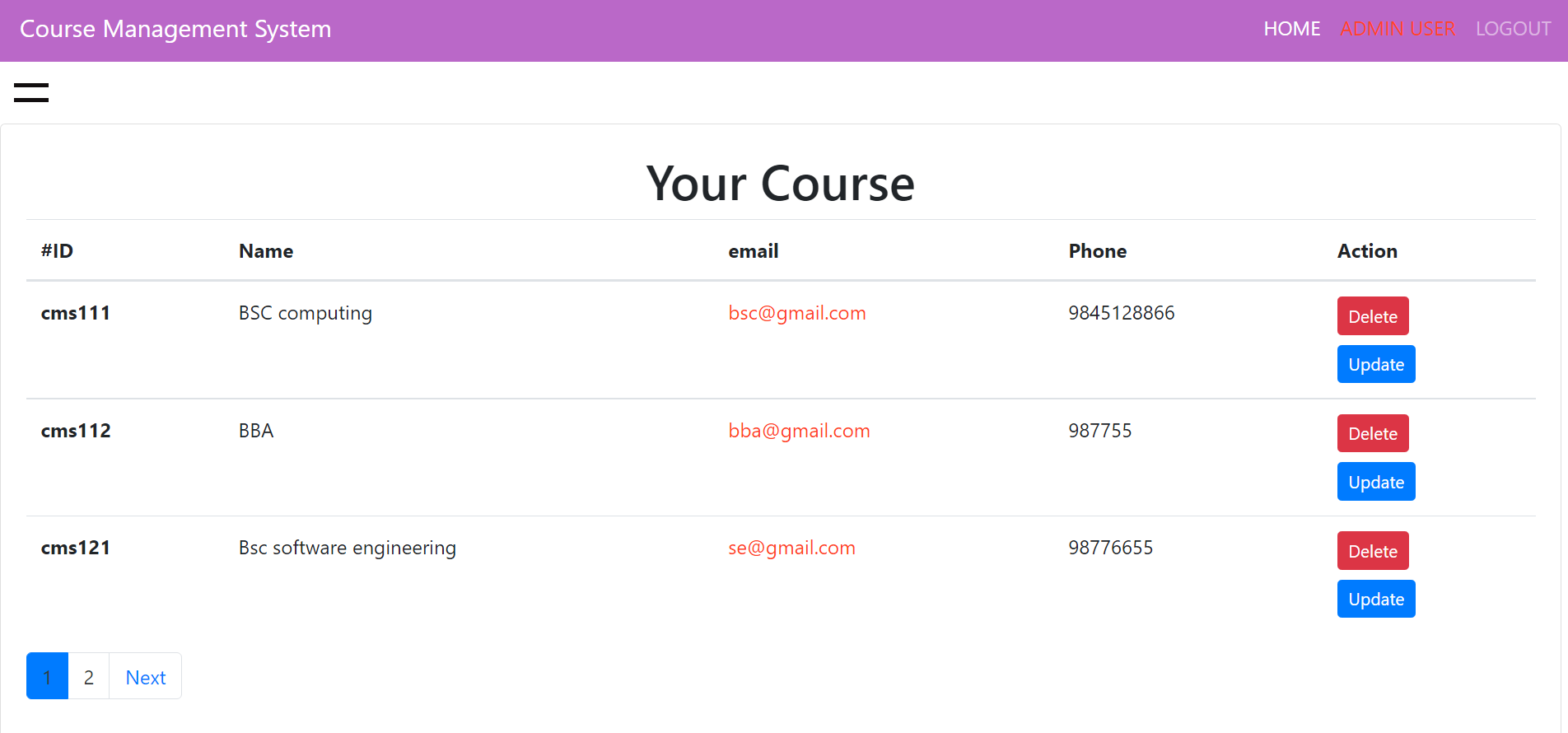
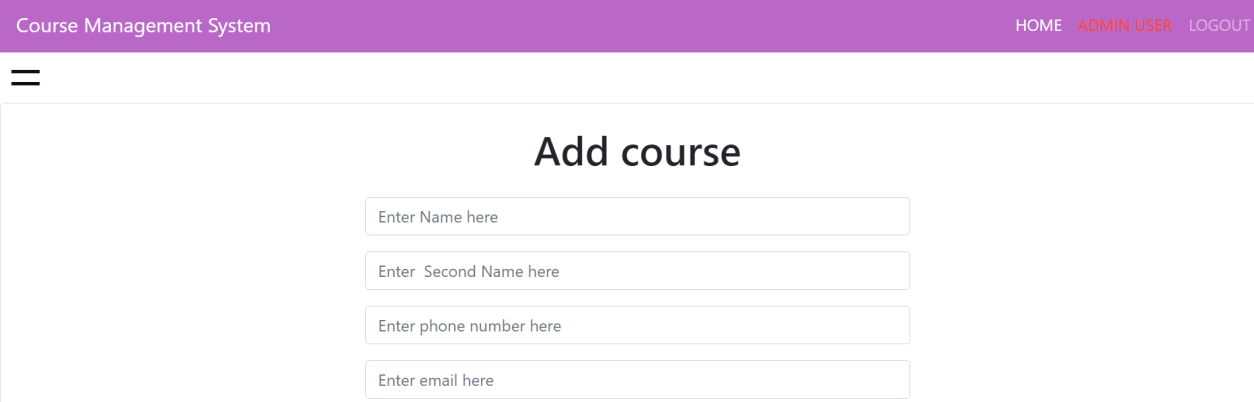


Figure 36 Show Courses Mock-up



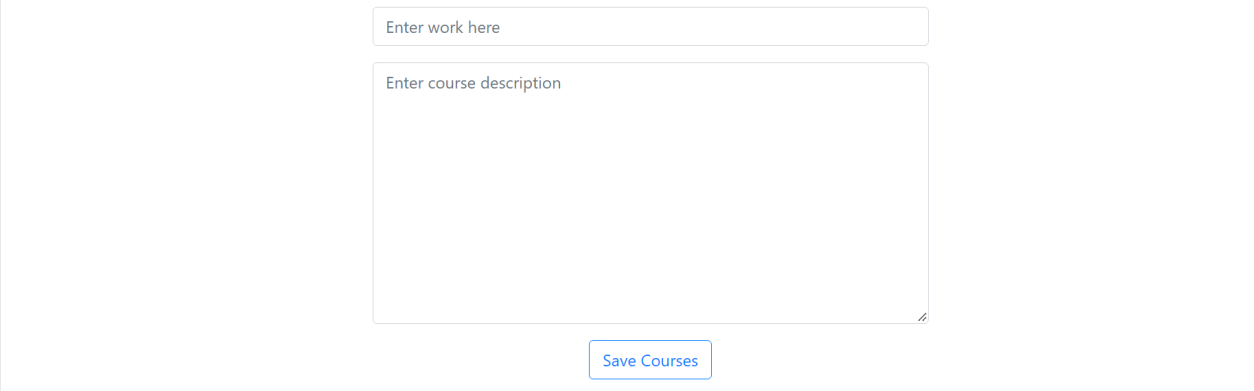


Figure 37 Add Course (mock-up)

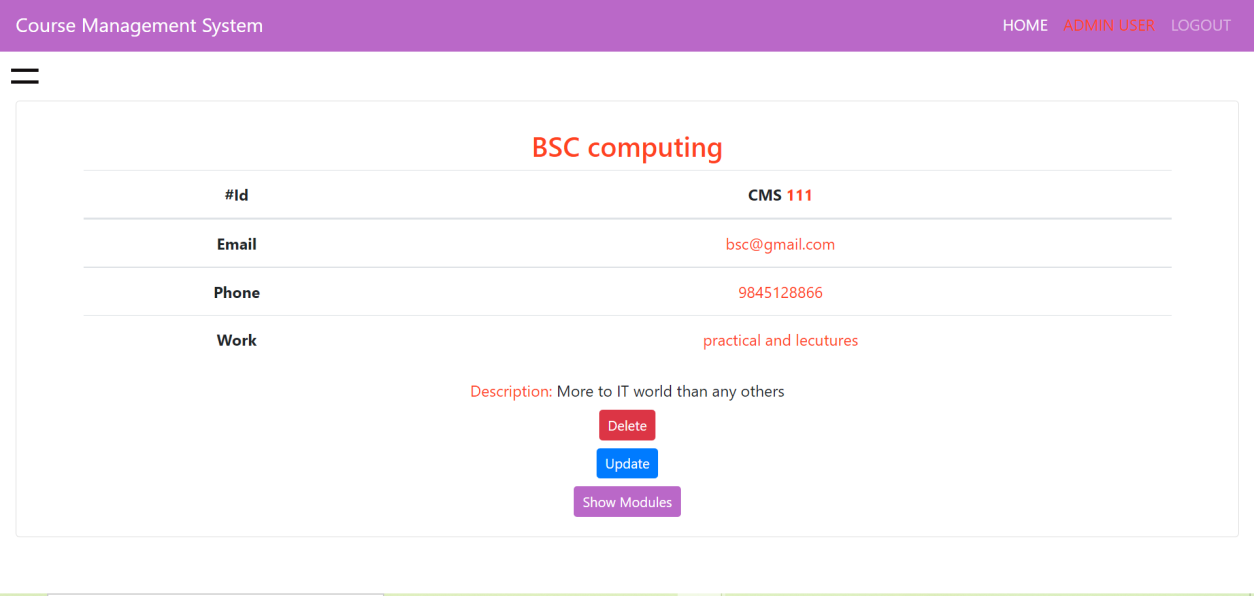
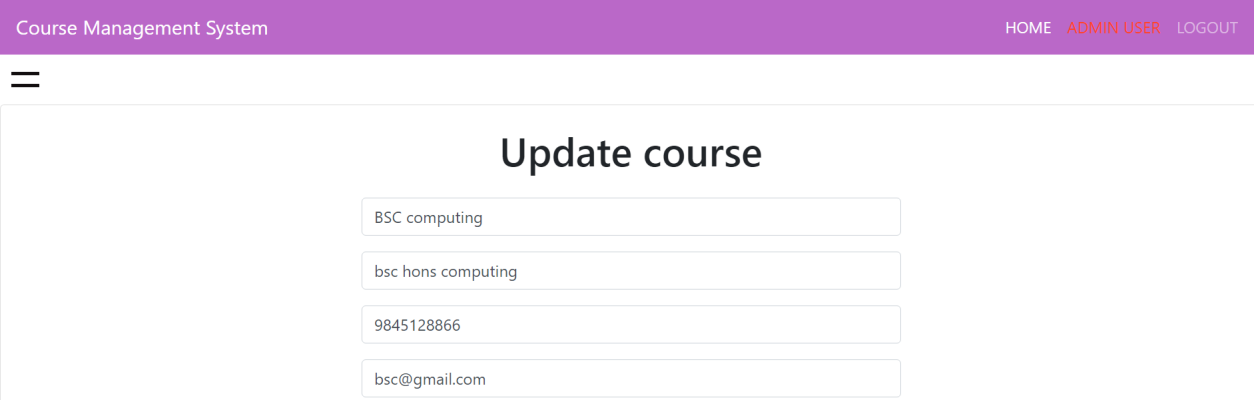


Figure 38 Course-detail(mock-up)



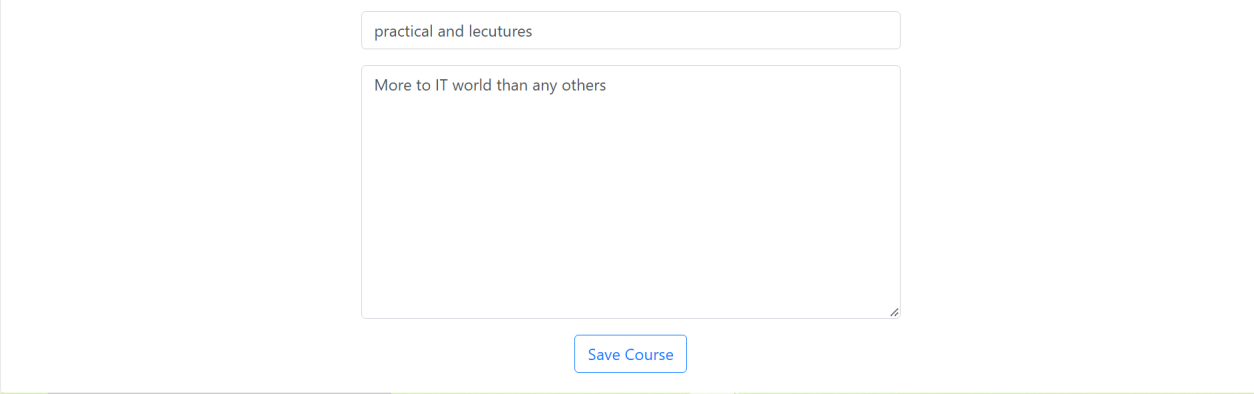


Figure 39 Update course (mock-up)

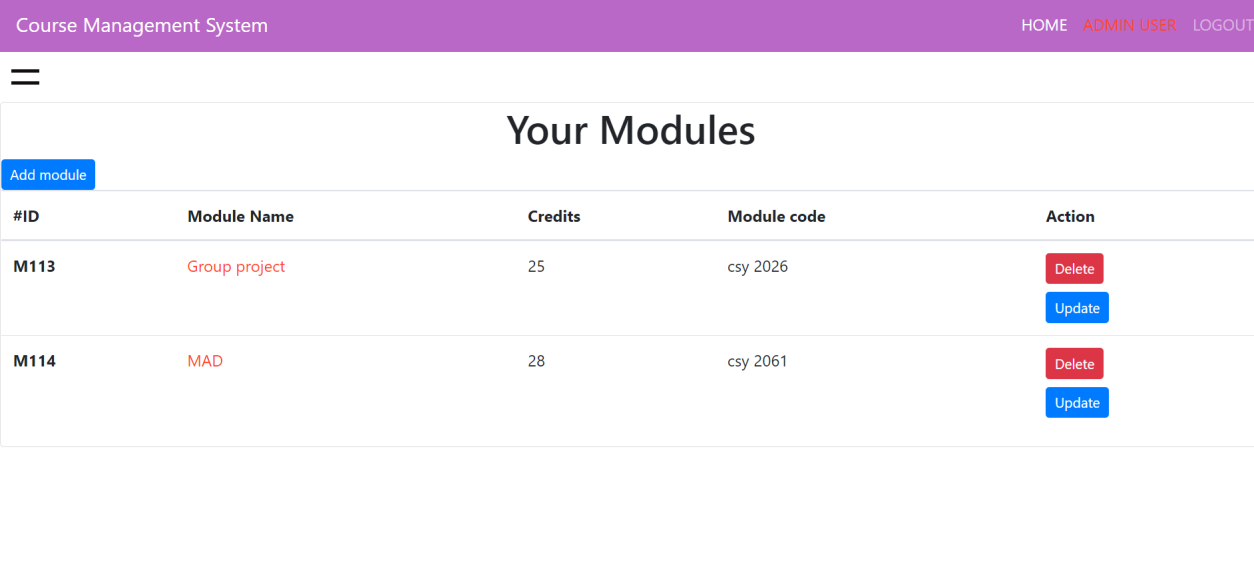


Figure 40 Show Modules (Mock-up)

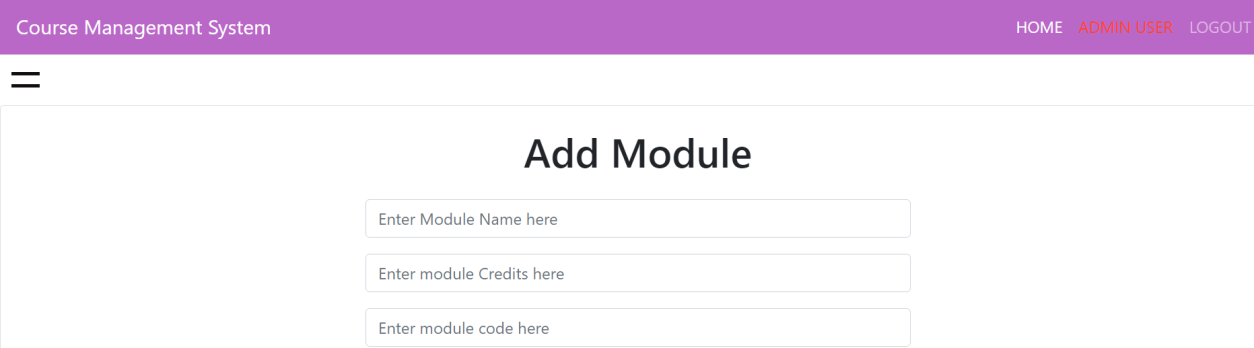




Figure 41 Add module (Mock-up)



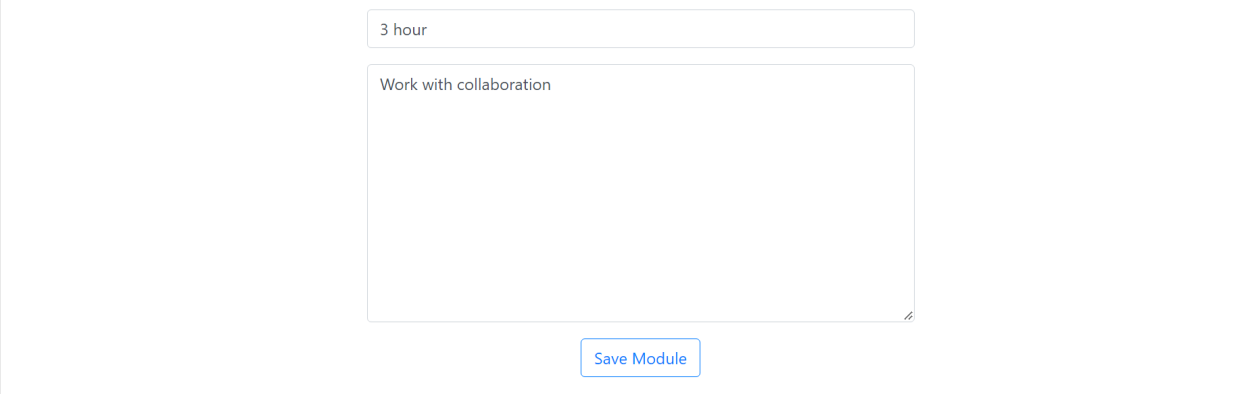


Figure 42 Update Module (mock-up)

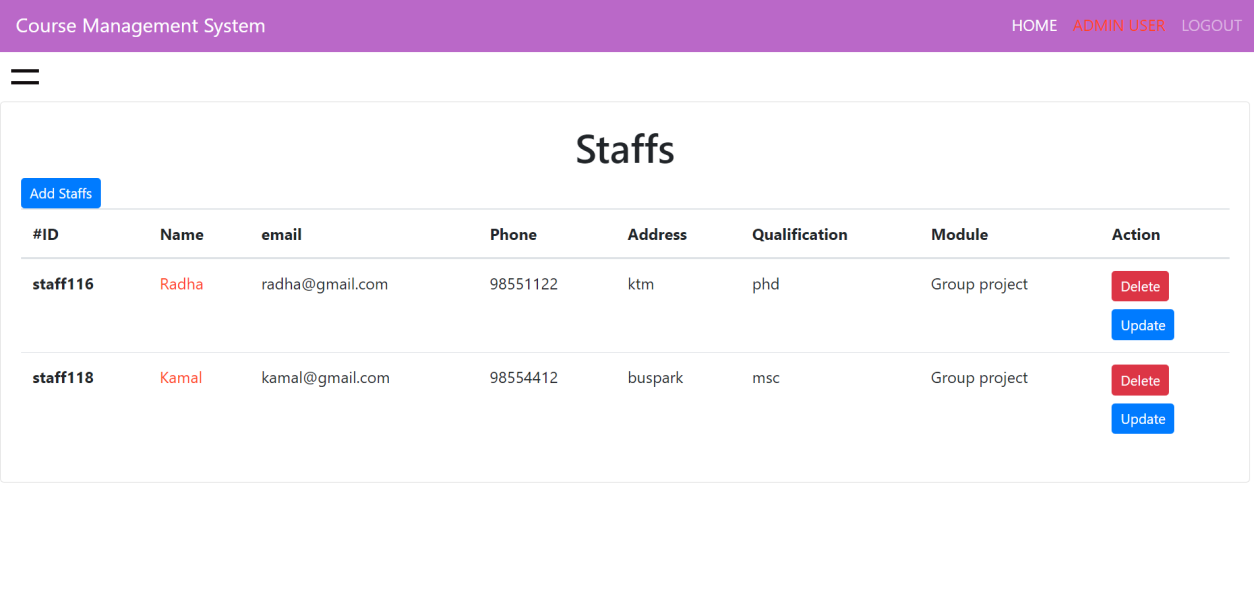
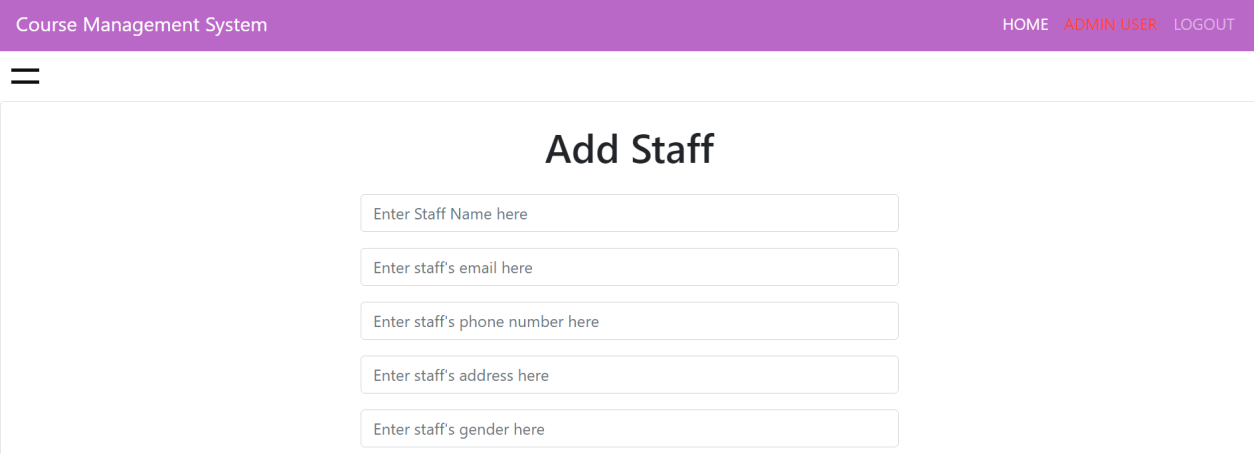


Figure 43 Show staffs (Mock-up)



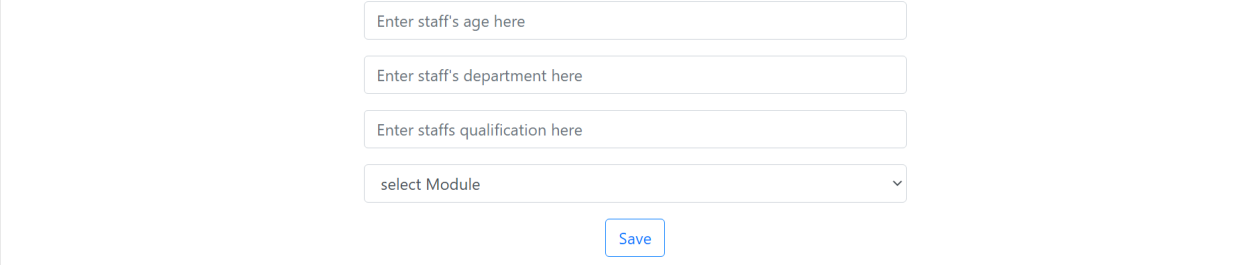
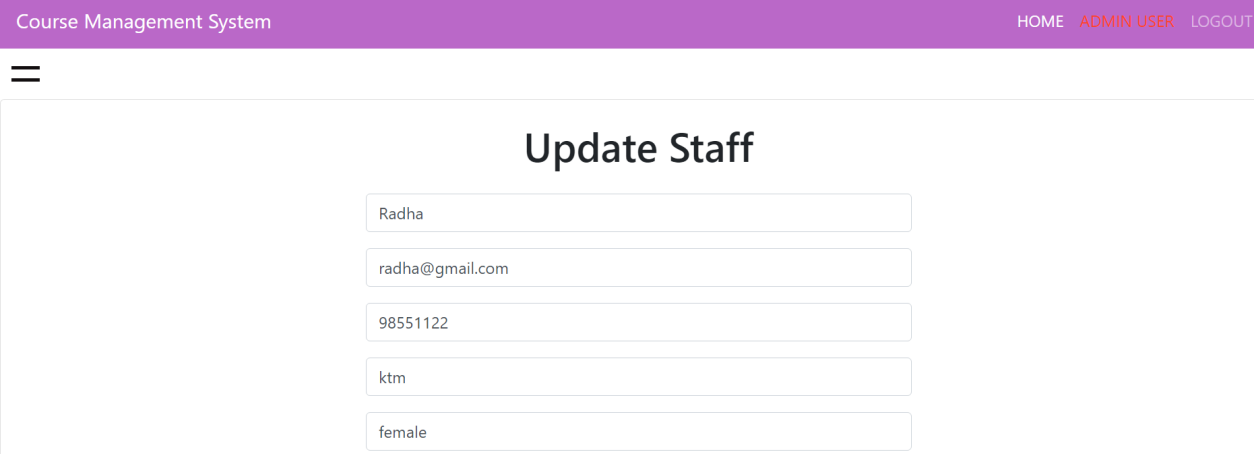


Figure 44 Add staff (Mock-up)



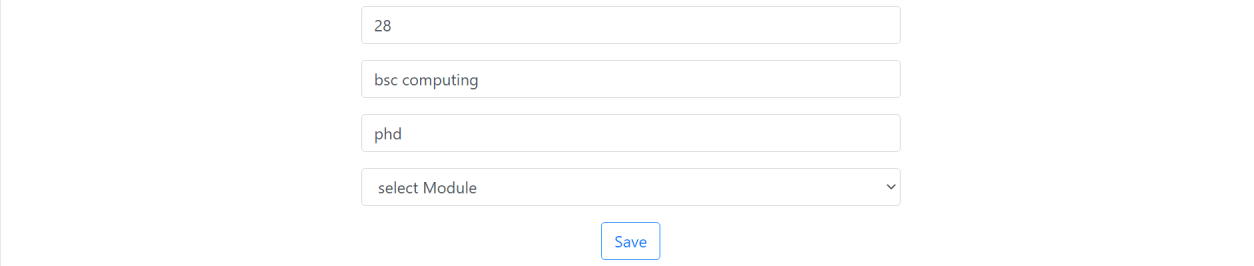


Figure 45 Update Staff (Mock-up)

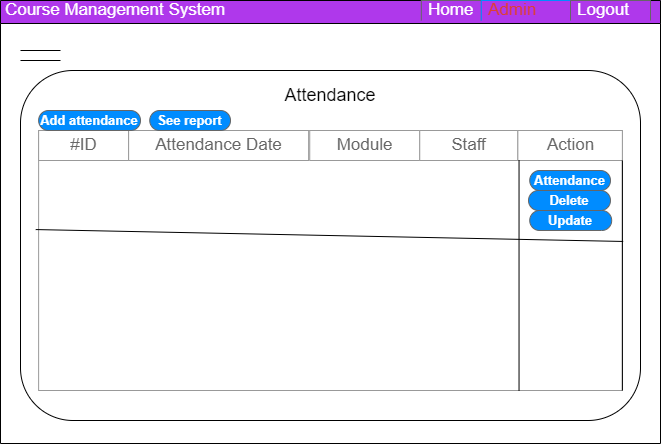
****

Figure 46 Screen mockup of Attendance page

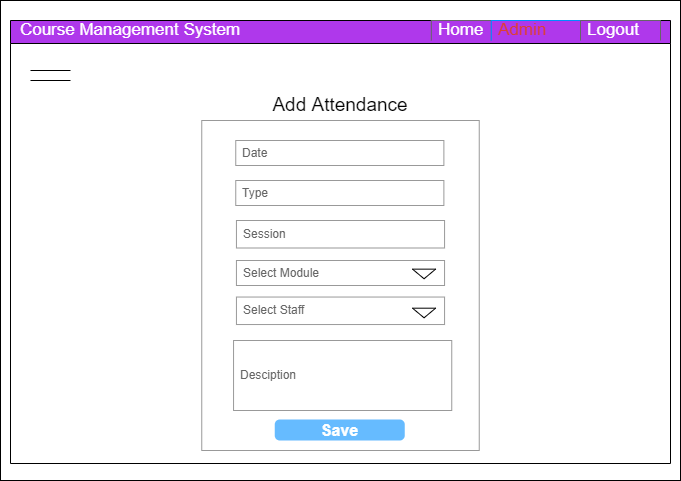
****

Figure 47 Add attendance page (mockup

****

Figure 48 Update Attendance (mockup)

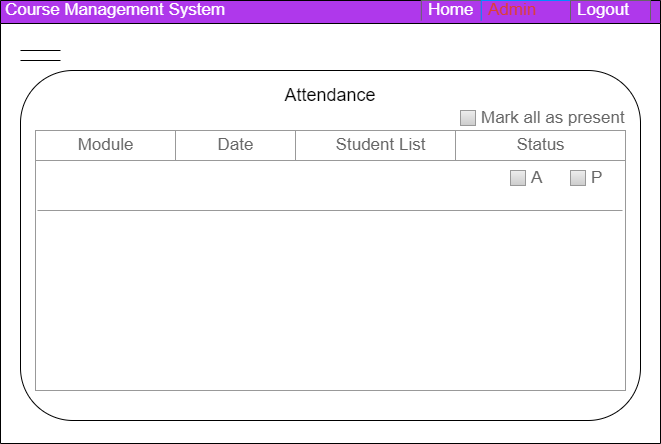
****

Figure 49 Do attendance (mockup)

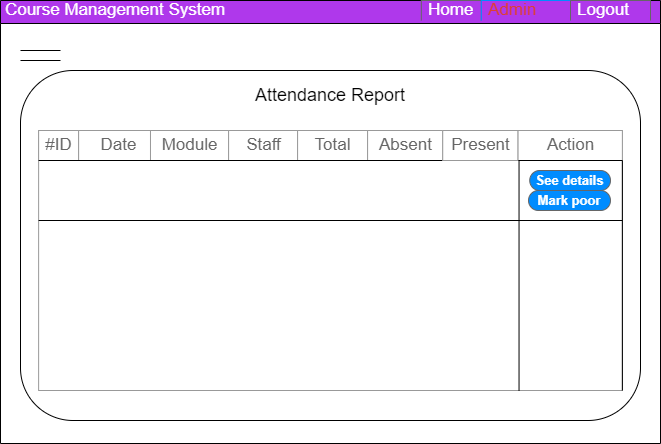
****

Figure 50 show attendance (mockup)

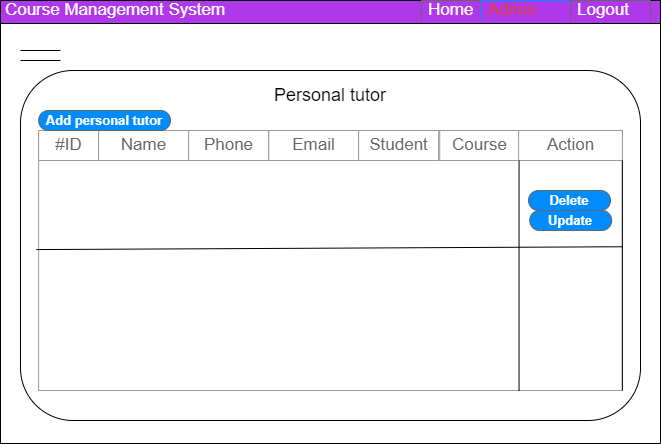
****

Figure 51 Show Personal Tutor

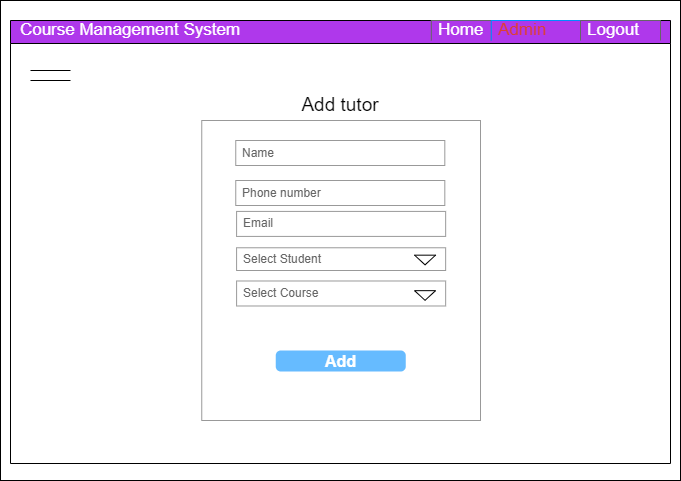
****

Figure 52 Add Tutor

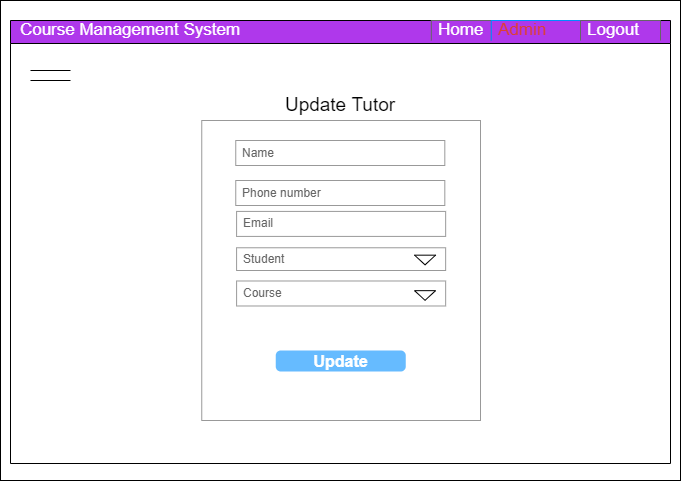
****

Figure 53 Update Tutor

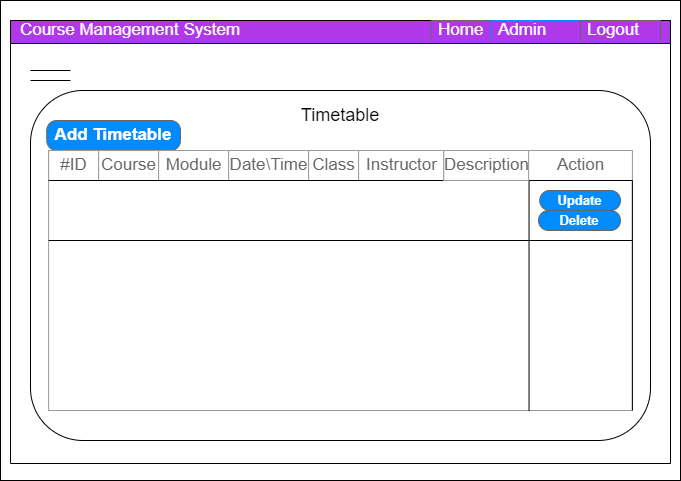
****

Figure 54 Show Timetable (mockup)

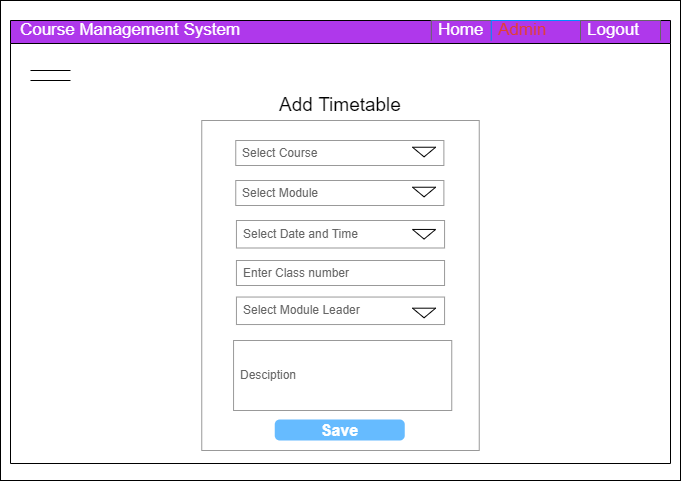
****

Figure 55 Add Timetable (mockup)

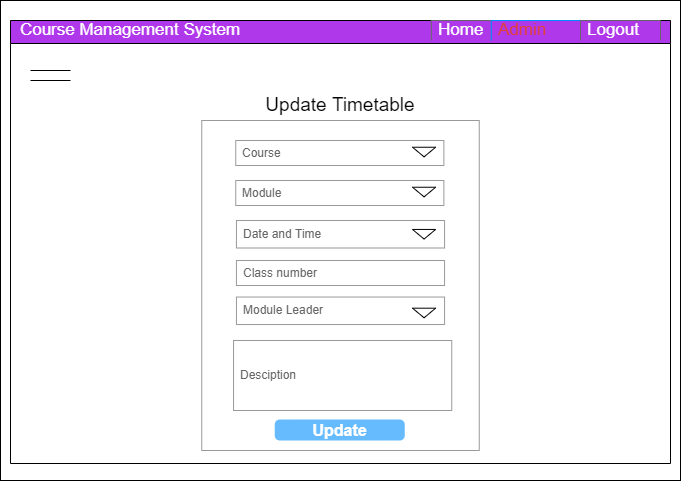
****

Figure 56 Update Timetable (mockup)

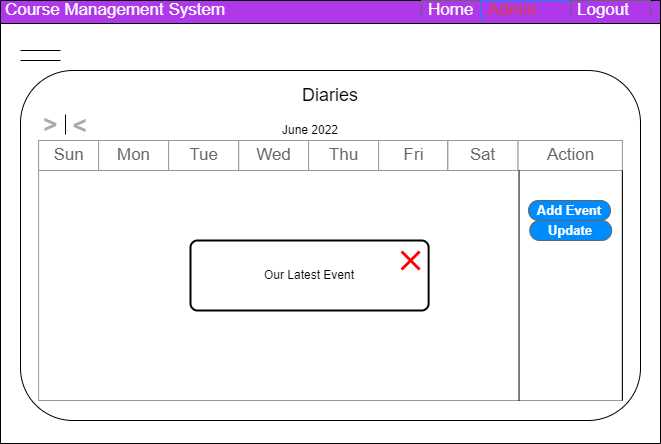
****

Figure 57 Show Diaries (Mockup)

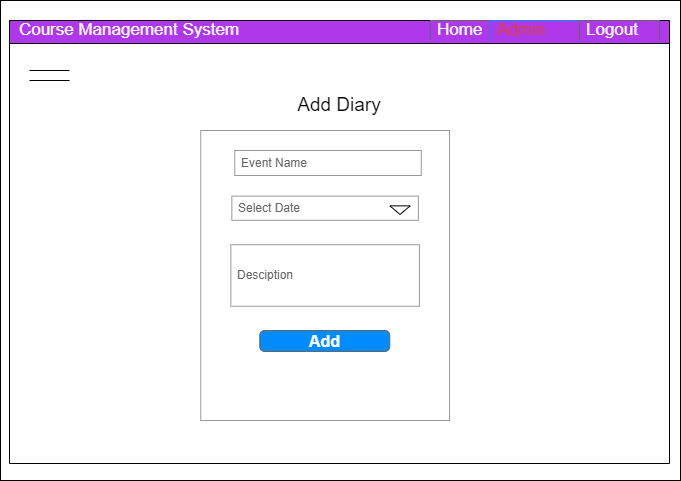
****

Figure 58 Add Diary

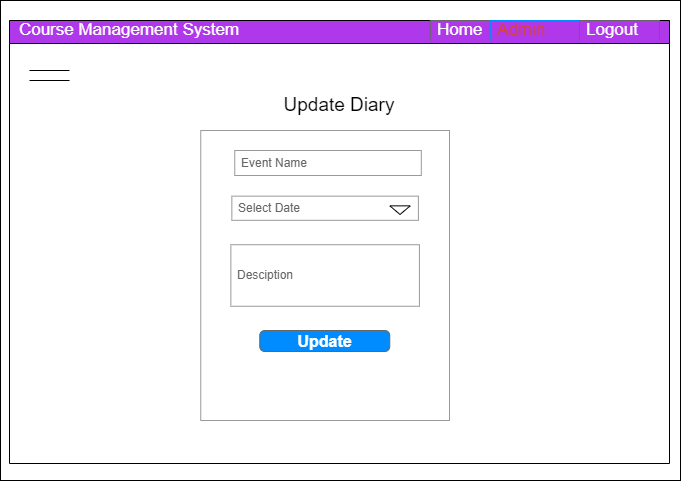
****

Figure 59 Update Diary

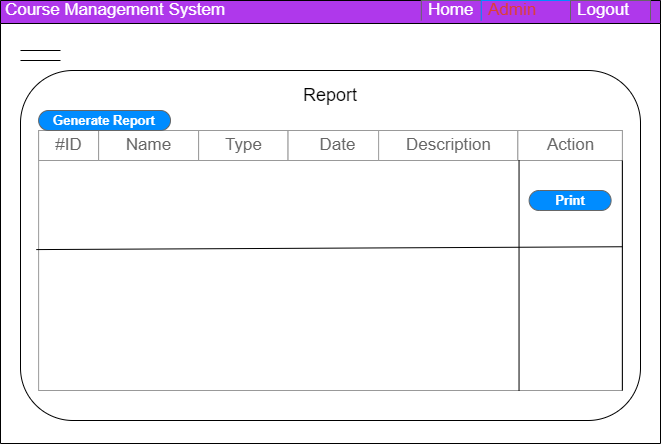
****

Figure 60 Show Report Page

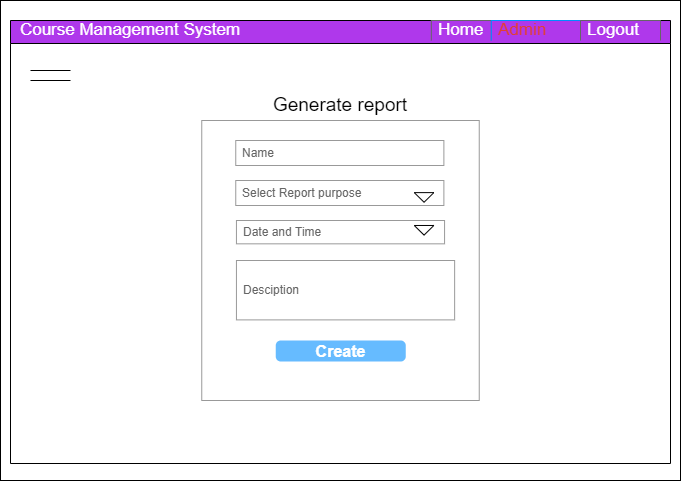
****

Figure 61 Add/ Generate Report

### 4.1.4 System Activity Event Diagrams:

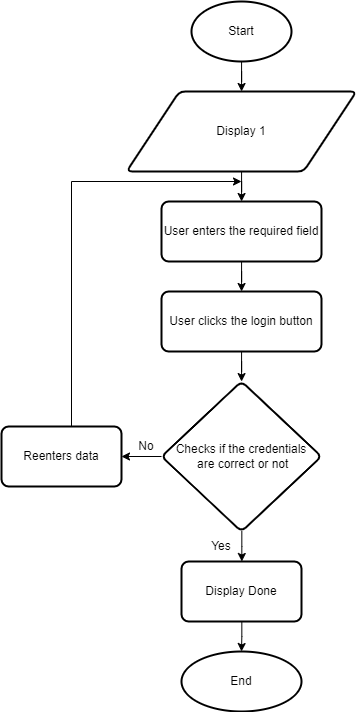


Figure 62 Login

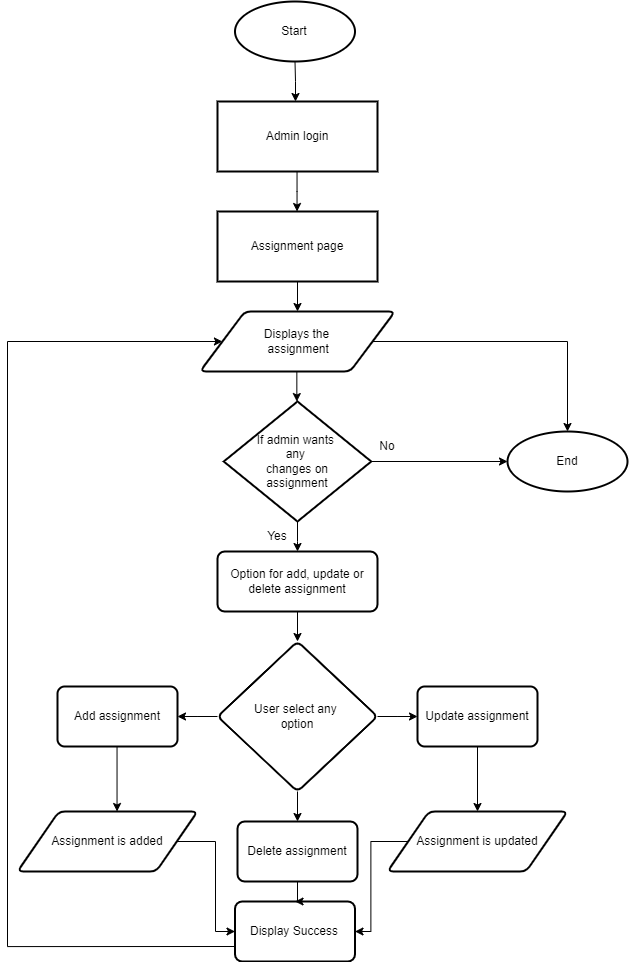


Figure 63 assignment

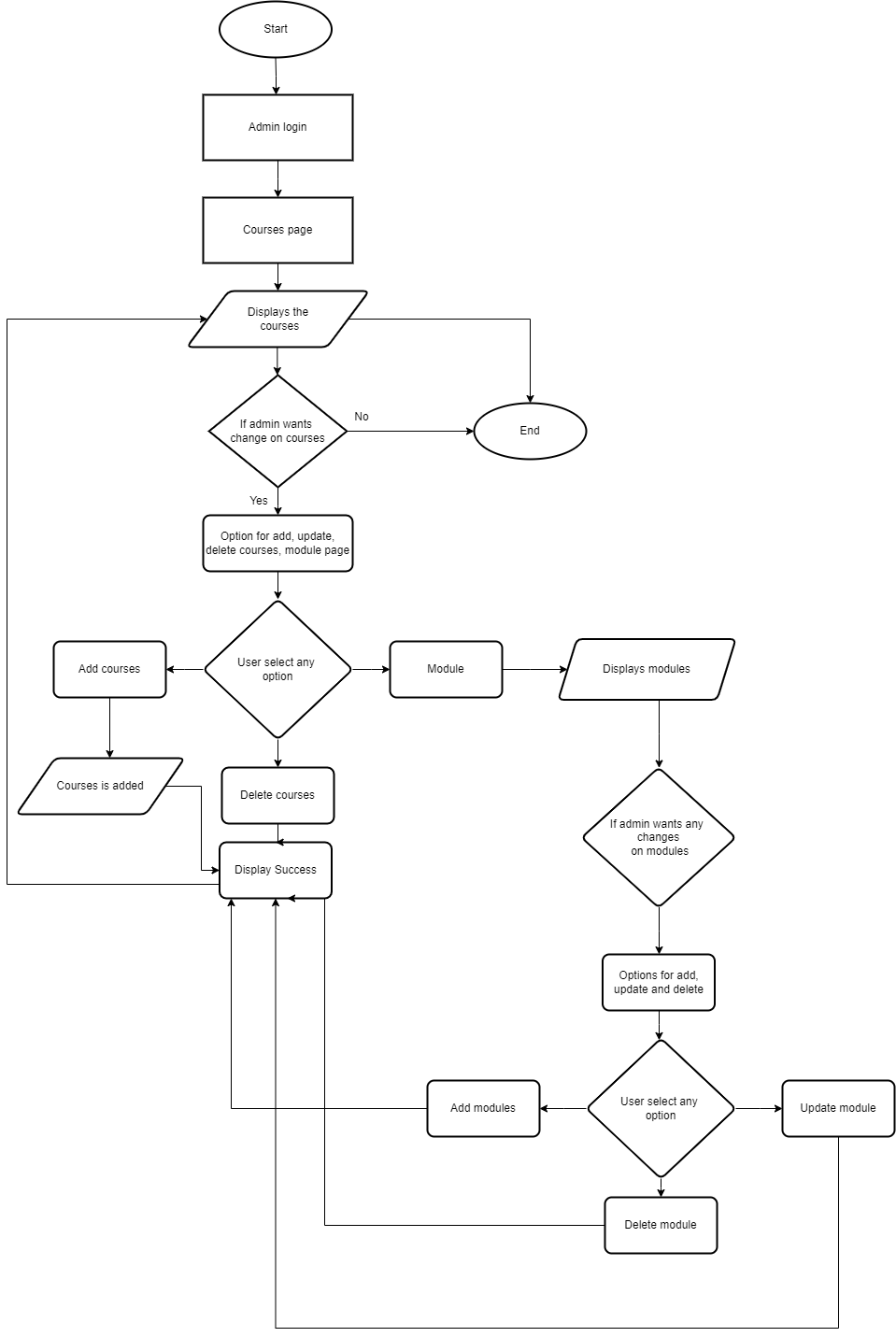


Figure 64 Courses

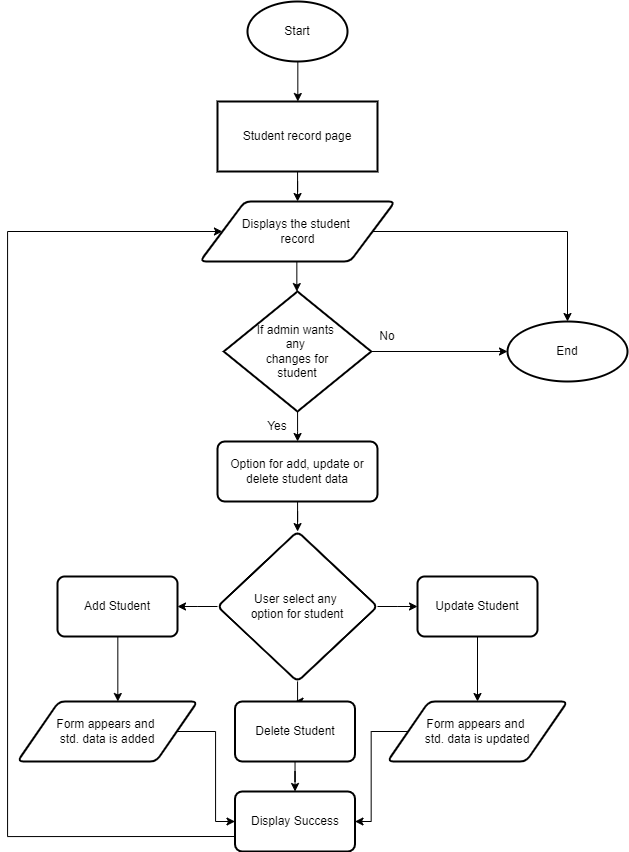


Figure 65 students

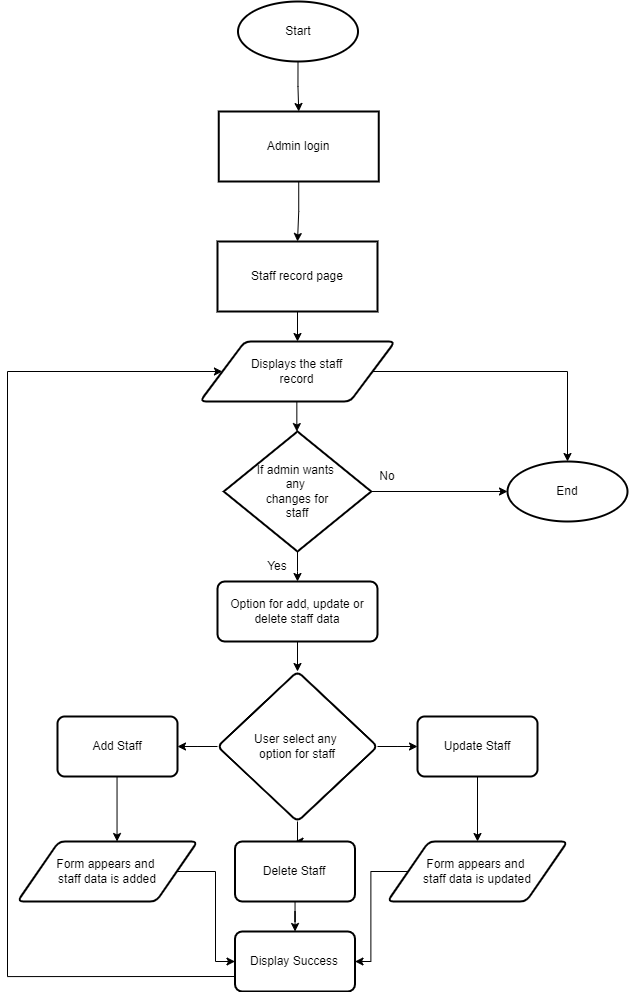


Figure 66 stafff

## 4.2 Design Revisions(Record Management System):

The final display or system will consist/ appear as shown in wireframe and mockups.The final system will be revised version of mockup.

The layout design and workflow is prepared according to system analysis and system design.

## 4.3 Heuristic Evaluation (Record Management):

Conformity Key

1 = Conforms in all parts.

2 = Conforms in most parts.

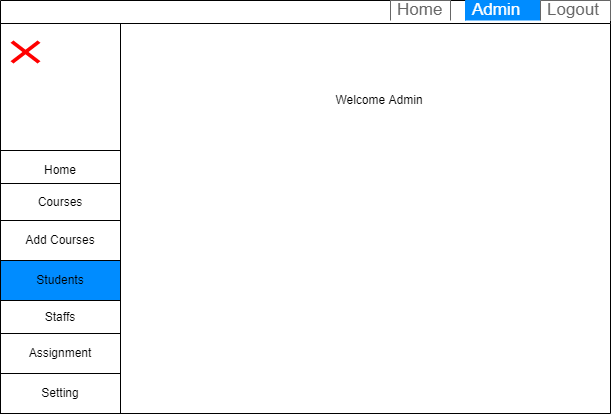
3 = Conforms in some parts.

4 = Does not conform.

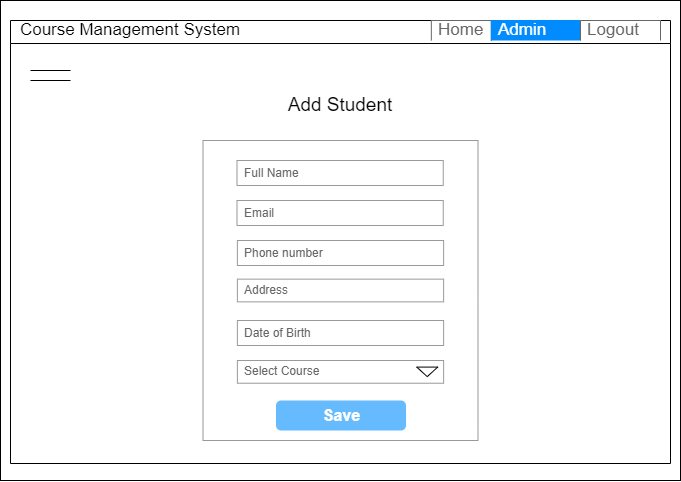
|  |  |  |  |
| --- | --- | --- | --- |
| **Usability Criteria (Heuristics)**  **Appropriate use of:** | **Level of Conformity**  **(1-4)** | **Evidence/Examples of application** | **Additional Comments/Issued Raised** |
| **Consistency/**  **Perceived Stability** | **1** | **Screenshot (273)** | **Consistency is maintained by use of same template in all management system. Eg. Login layout for admin, staff, student appears same** |
| **WYSIWYG** | **2** | **Course Detail**  **Screenshot (273)** | **The course detail page and module detail page show same style of display** |
| **User Centred**  **Control** | **2** | **Show Course** | **The controls are user friendly, no need to have technical knowledge to use this system.** |
| **Feedback and Dialog** | **1** | **Screenshot (374)** | **Green signal for successful message or feedback.** |
| **Buttons** | **1** | **Show Course** | **Same size of button. Red button for delete and blue for update** |
| **Colour** | **2** | **Screenshot (372)** | **Plain and simple color used. Pinkish color for navbar and orange color to highlight.** |
| **Modelessness** | **1** | **-** | **No extra modes are present.** |

## **4.4 Draft Interface Designs (Student Records/Information Portal)**

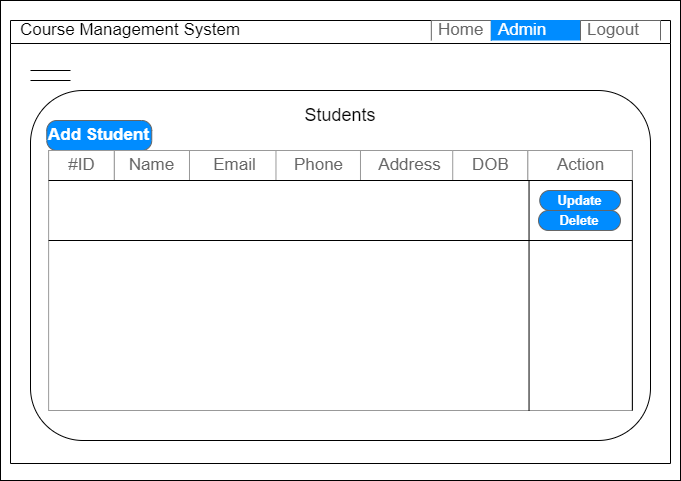
### 4.4.1 Wireframes

The sketch of how the initial system looks is a wireframe. The wireframes of the Student Records/Information Portal are shown below.

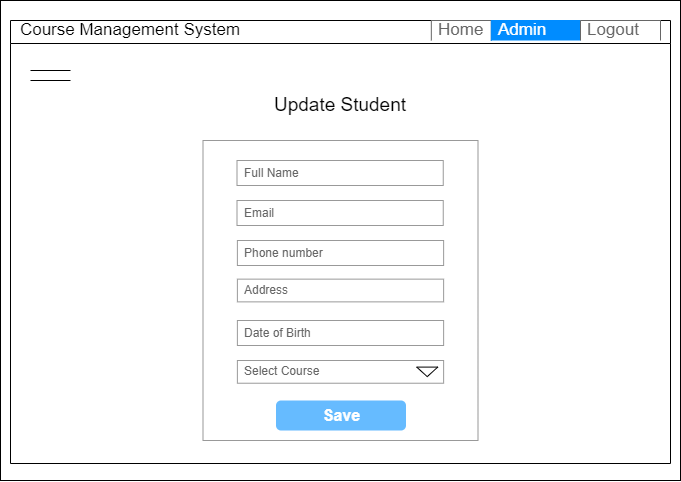
**Fig: Admin Dashboard**



**Fig: Add Student page(admin View)**

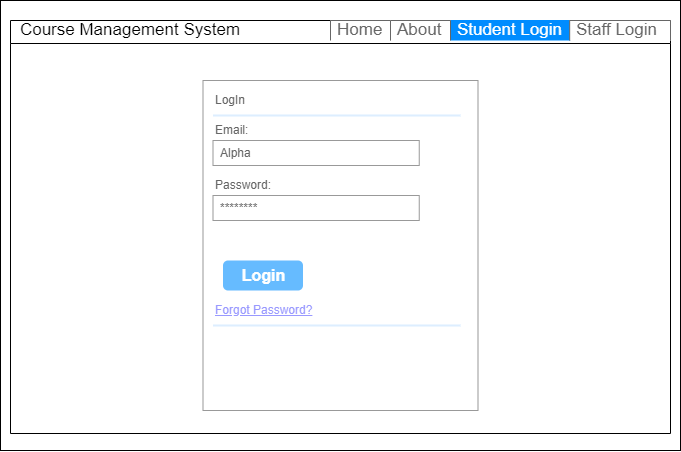


**Fig: Show Student Page (Admin View)**

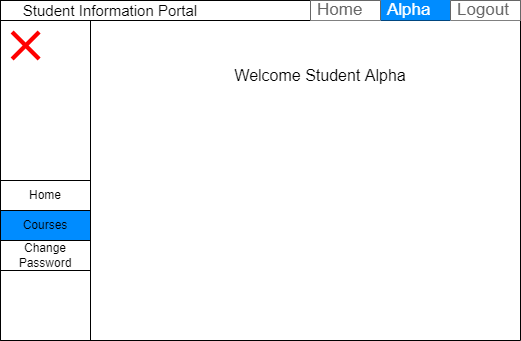


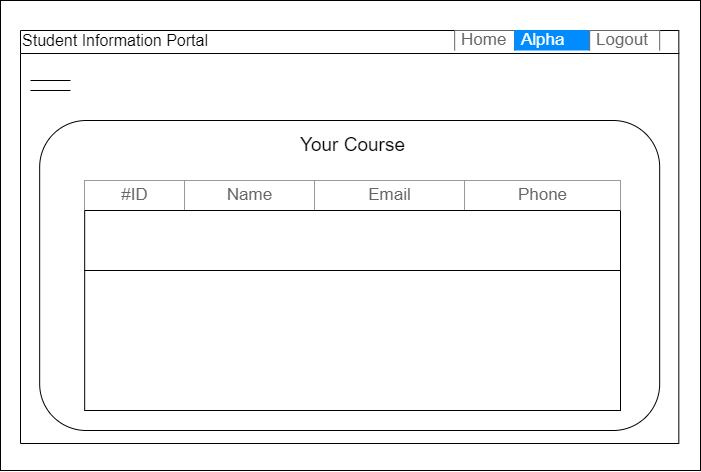
**Fig: Update Student Page (Admin View)**

Similarly, the wireframe of a page from the student side of the view is given below:

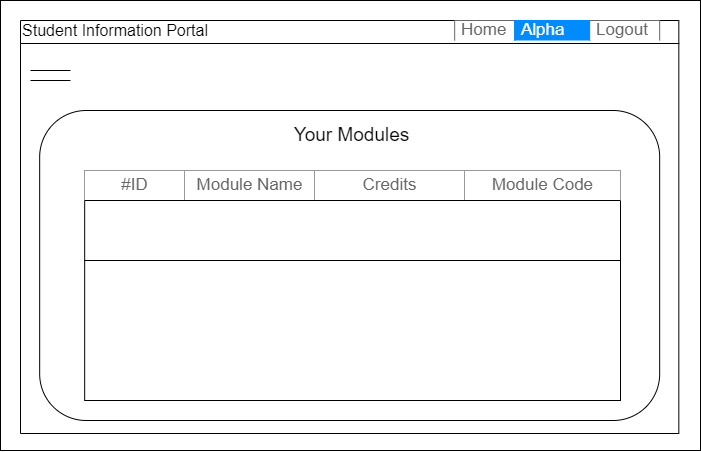


**Fig: Student Login page**

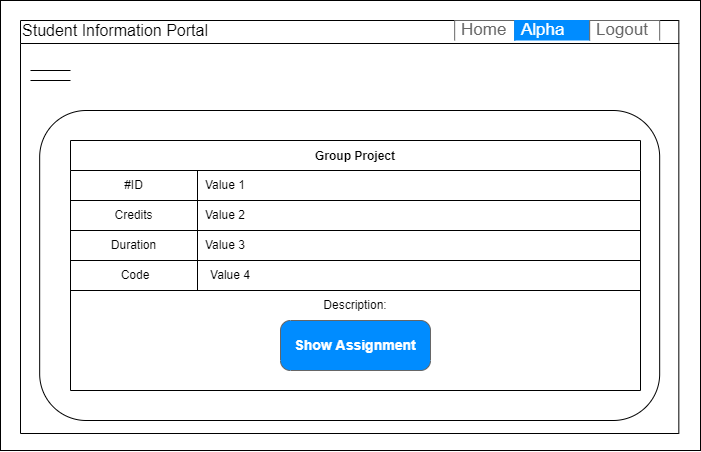


**Fig: Student home page**

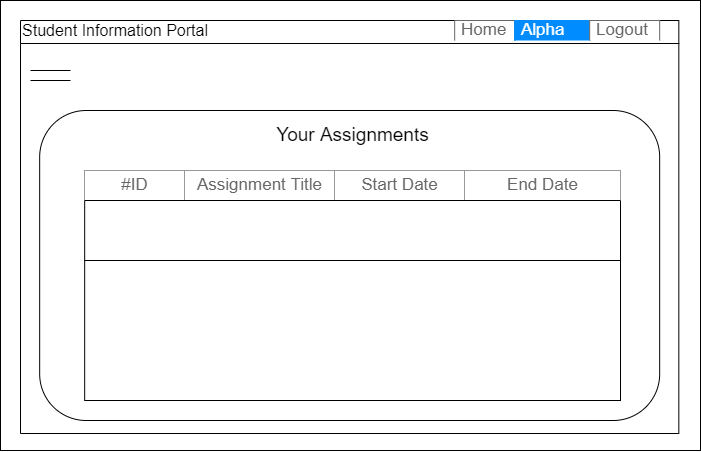
**Fig: Enrolled courses page (Student view)**



**Fig: Module page (Student view)**

**Fig: Particular module information**

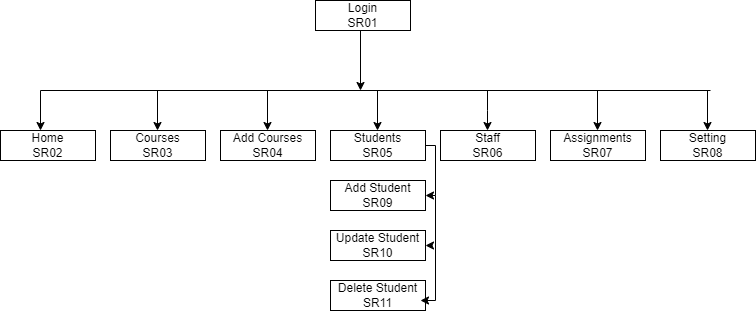
**Fig: Assignment information (Student View)**



### 4.4.2 System Navigation Diagram

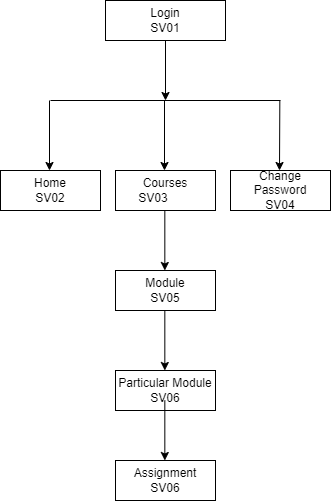
This topic includes an overview of the system, i.e. it navigates the pathways of the student record. The system navigation diagram includes screen names, serial numbers, and arrows to illustrate the pathways.

Here is the navigation diagram of the student record (Admin view):

****

**Fig: System navigation diagram of student record (Admin view)**

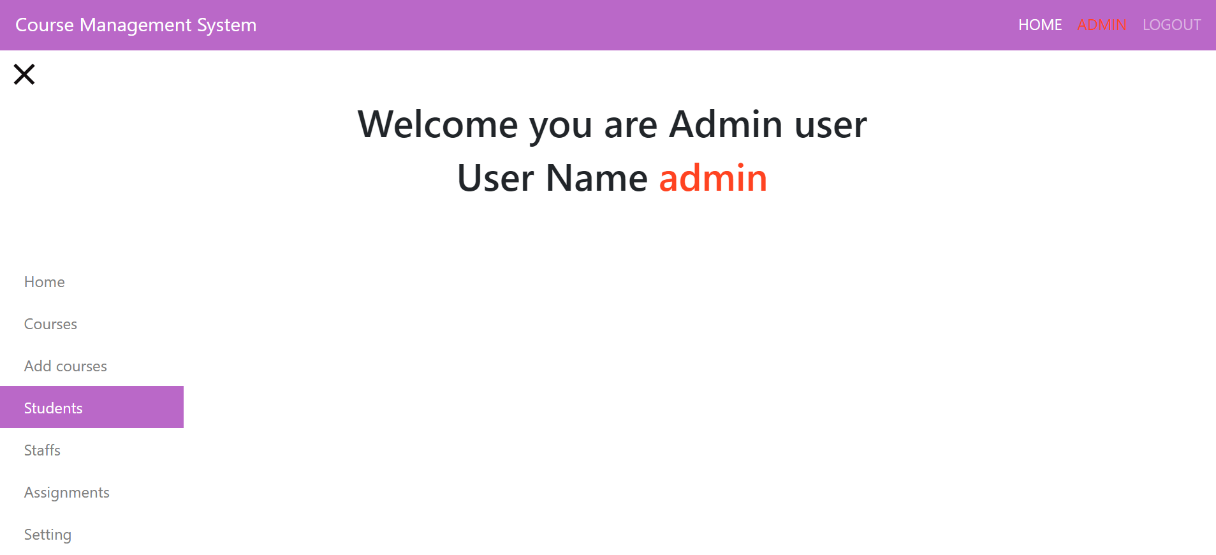
Similarly, navigation diagrams from the enrolled student side are also given below:



**Fig: System navigation diagram (Enrolled student view)**

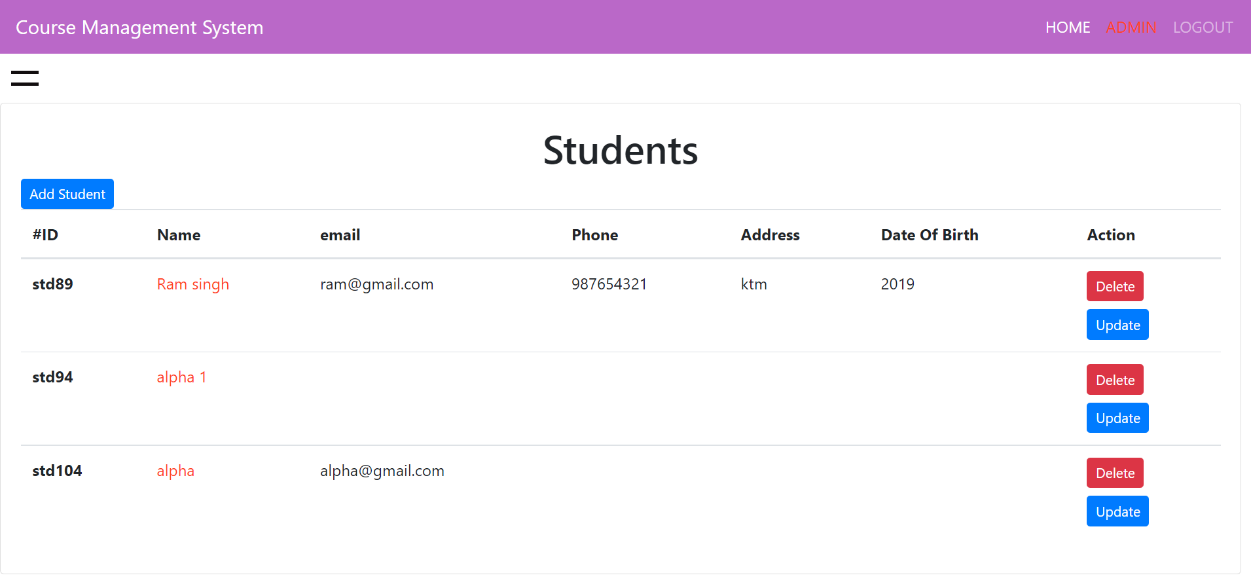
### 4.4.3 System Screen mock-ups

It is the final layout of the system. This layout is what the system looks like. System Screen mock-ups for the student record/information portal are given below:

 **Fig: Admin page**

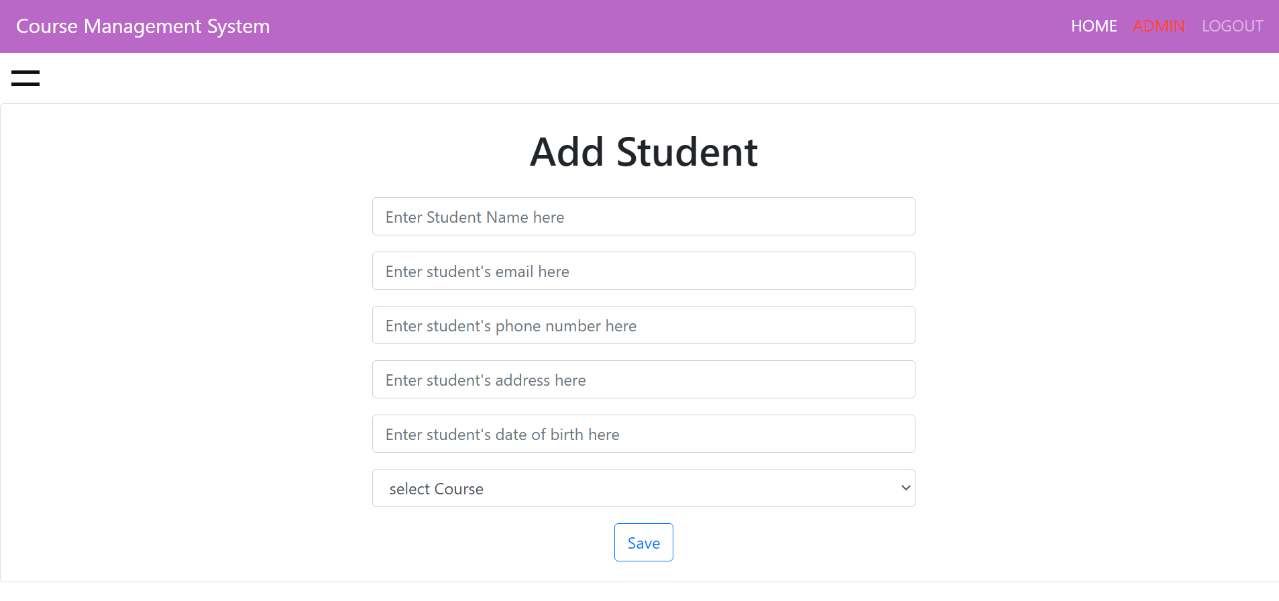
It displays the admin page where the admin login then the dashboard with different titles appears such as home, courses, add courses, students, etc.

If the student's record should be managed then admin move to students page from the dashboard.

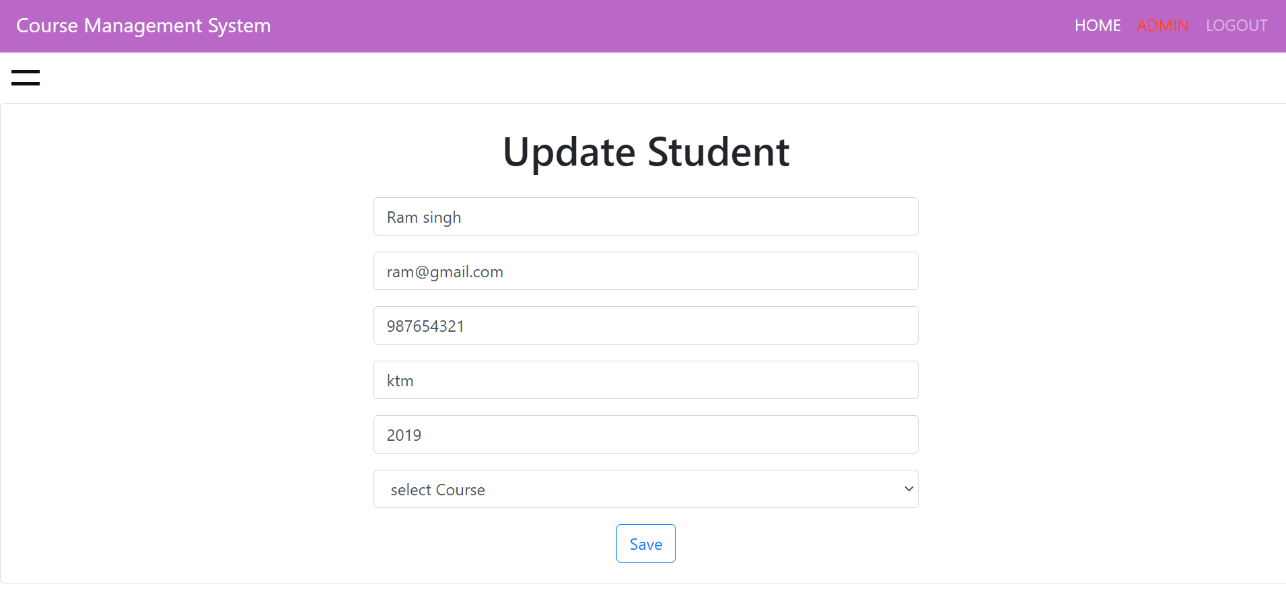


**Fig: Student Record page mock-up (Admin View)**

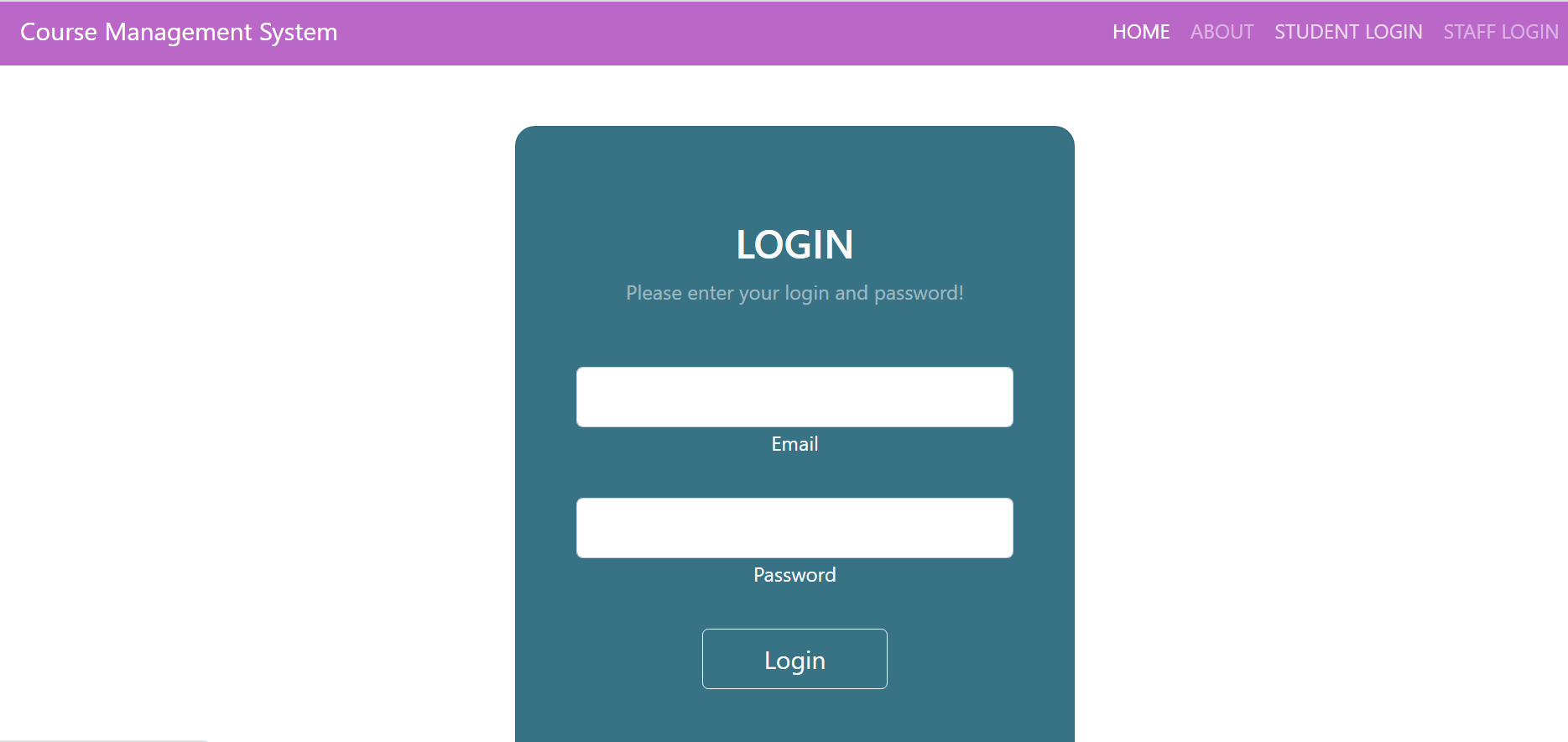
The given mock-up displays the student record when the admin opens the student panel from the dashboard. It has the feature to add a student from add student button. Similarly, it contains the

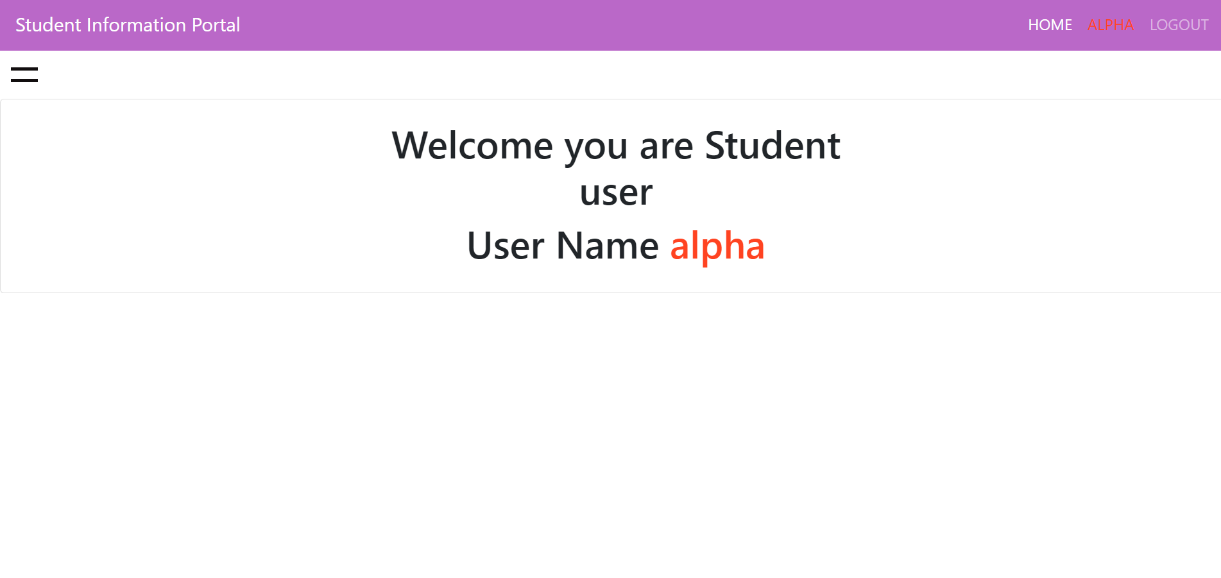
 students' information. Also, the admin can delete and update the student record from the table row action.

**Fig: Add student page**

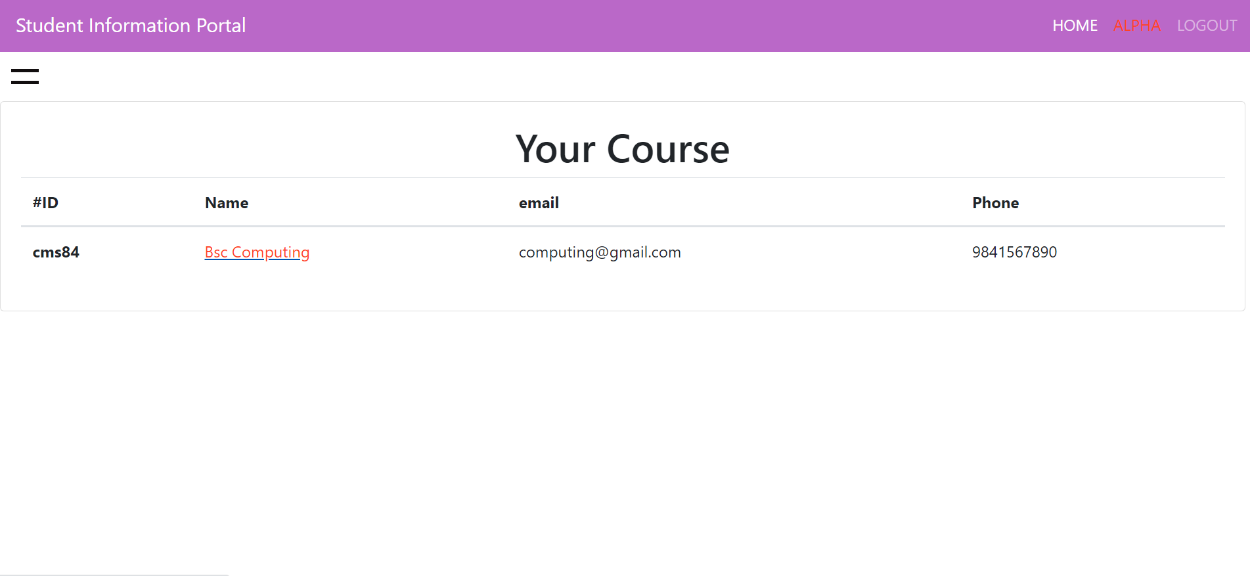


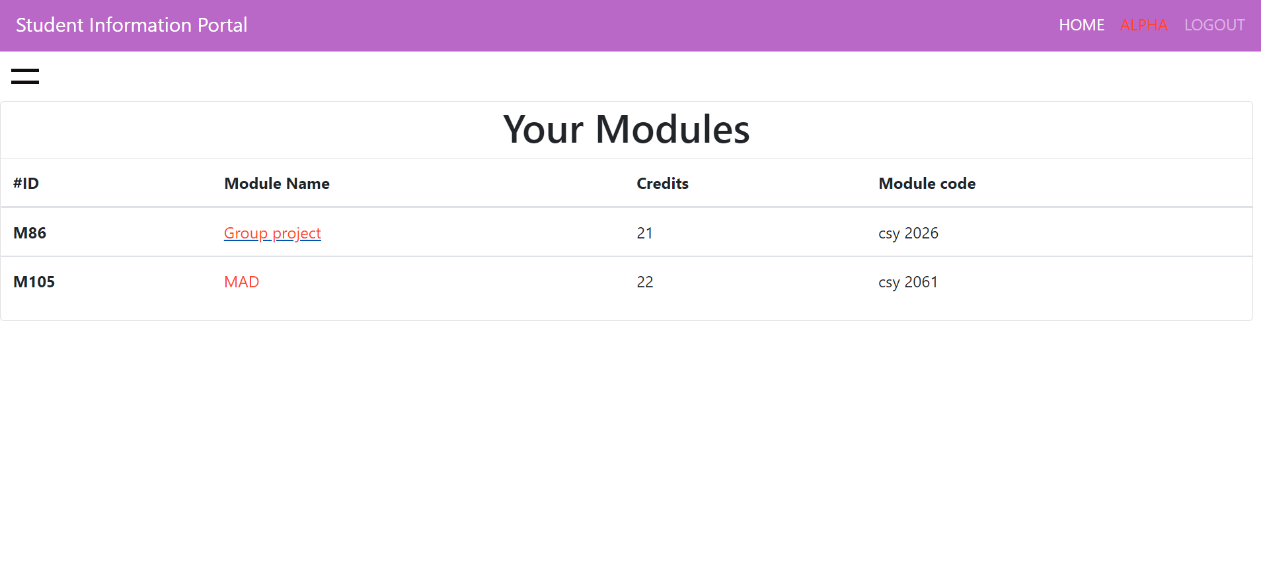
**Fig: Update student page**

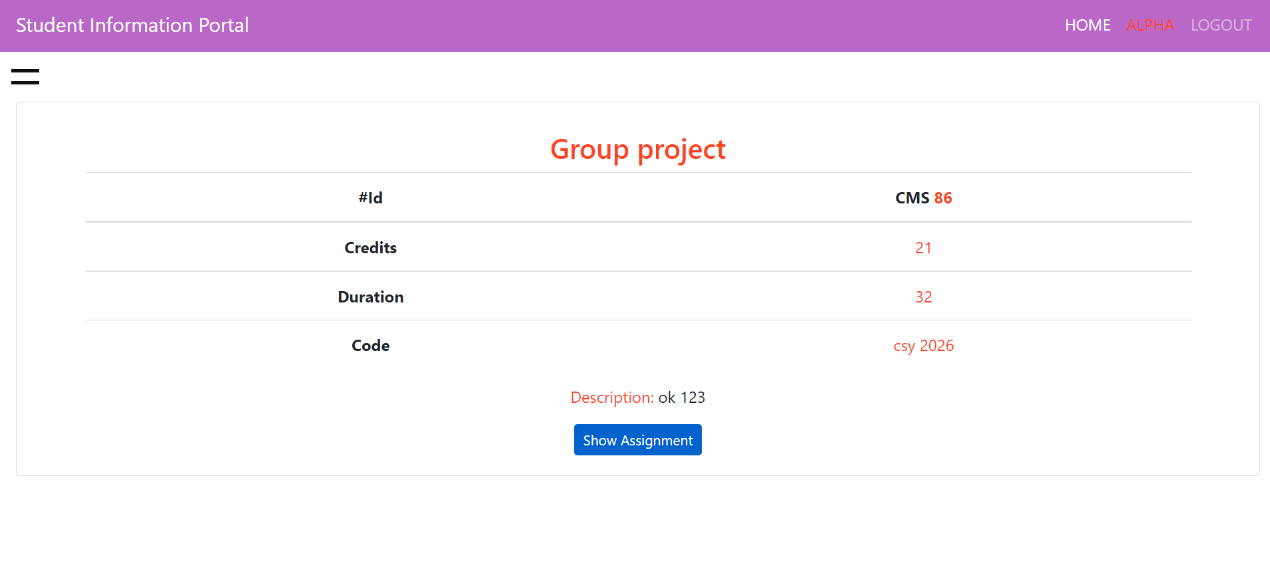
**Fig: Student Login page**



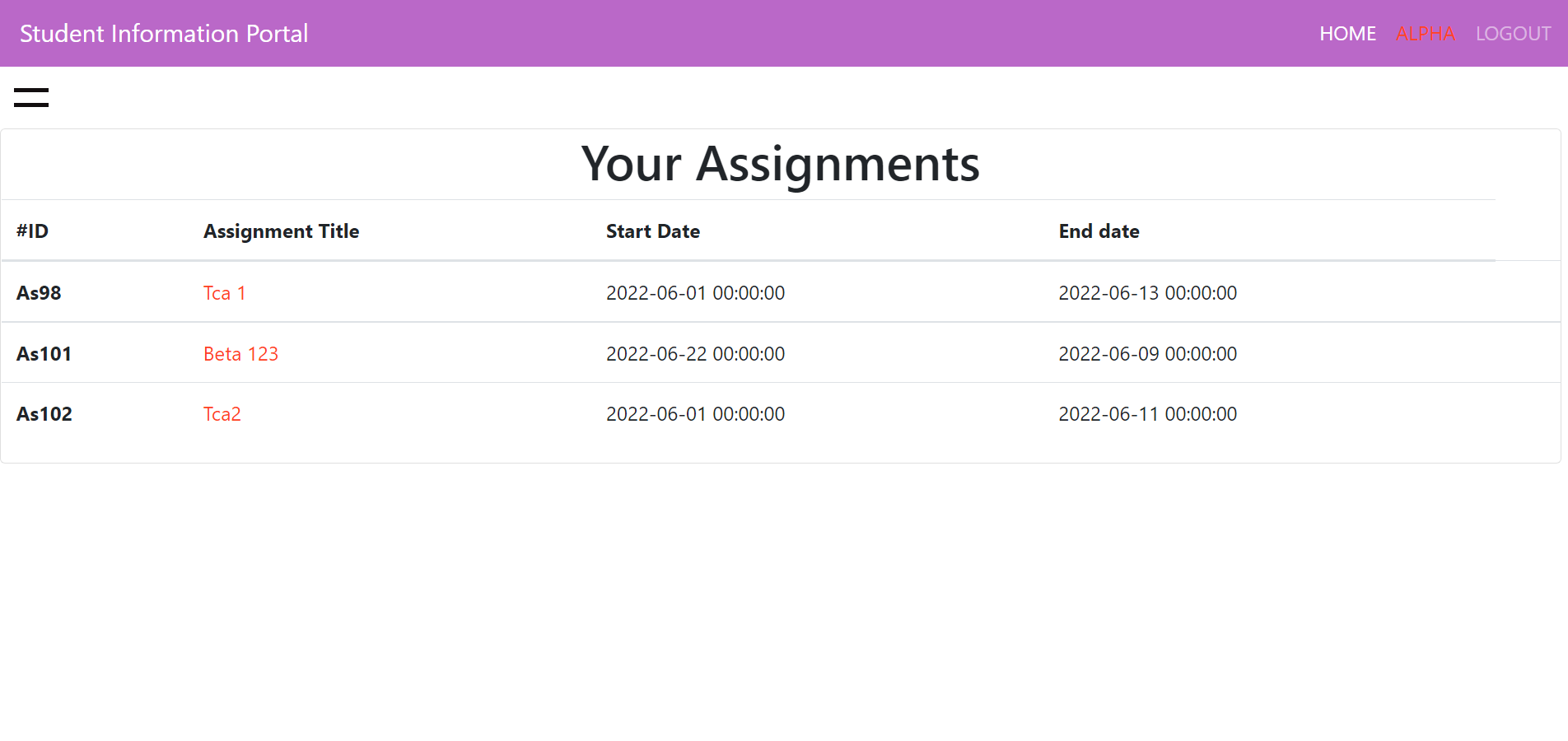
**Fig: Student Home Page**



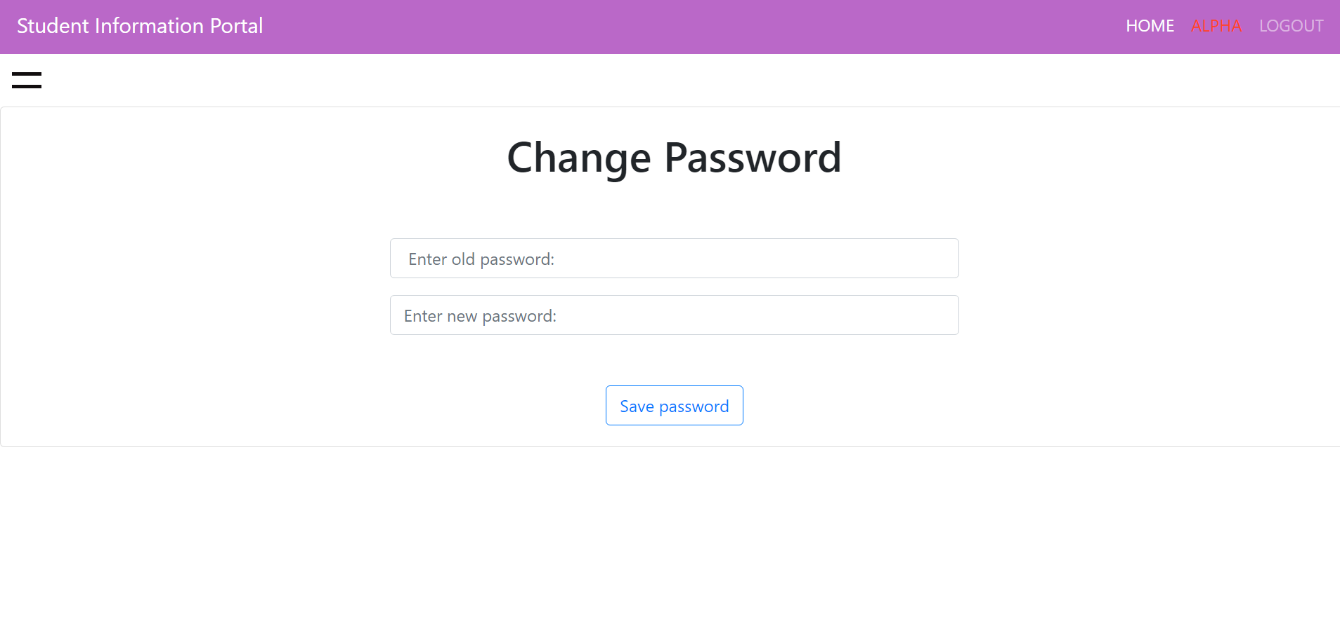
**Fig: Course page Fig: Enrolled module page (Student view)**

****

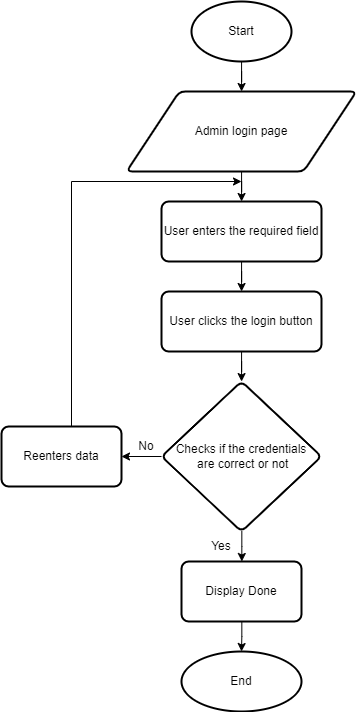
**Fig: Particular module**

****

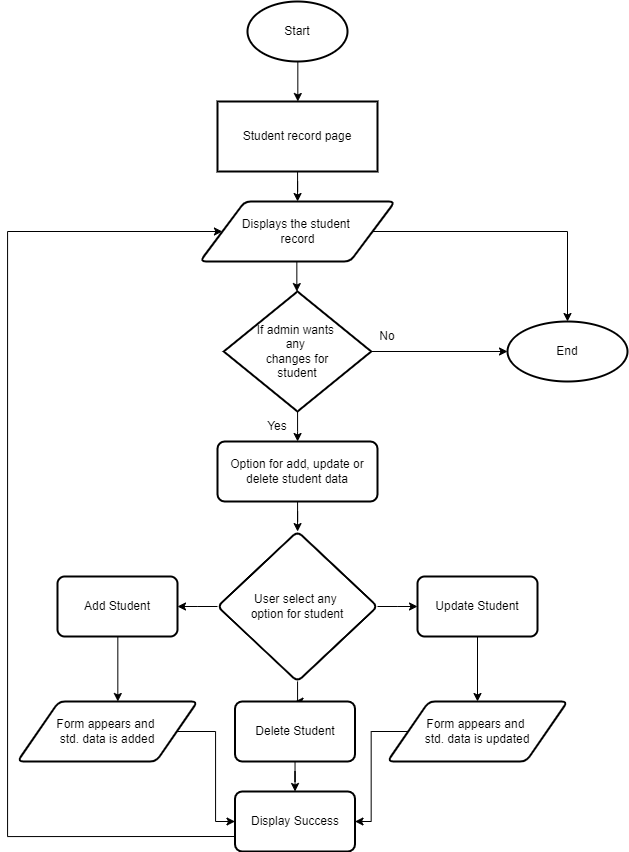
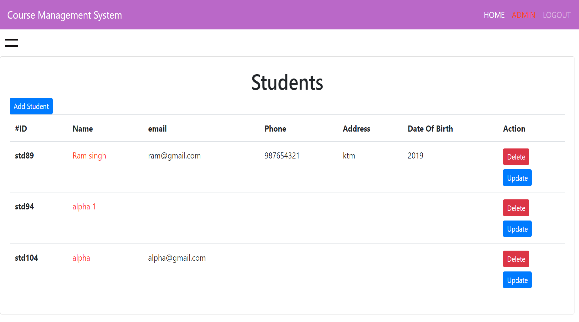
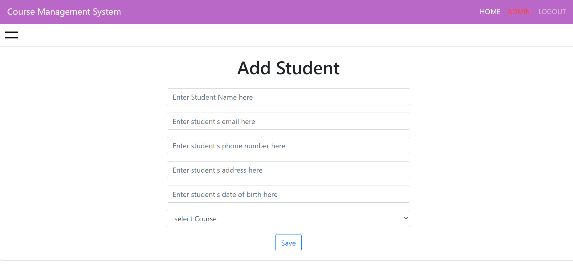
**Fig: Assignment page**

 **Fig: Change Password**

### 4.4.4 System Activity Event Diagram

The system activity diagram for student records is given below:

**Fig: Activity Event Diagram Admin Login**

**Fig: Activity Event Diagram Student Record (Admin View)**

## **4.5 Design Revisions (Student Records/Information Portal)**

The above-displayed wireframe, system navigation, mockups, and activity event diagram are all the draft interface designs for the course management system.

## **4.6 Heuristic Evaluation (Student Records/Information Portal)**

1 = fully conforms

2 = conforms in most areas

3 = conforms in few area

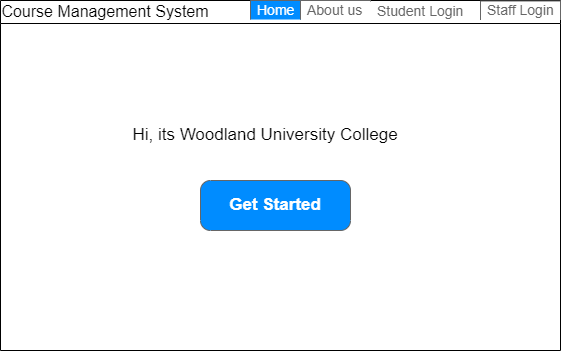
4 = does not conform at all

|  |  |  |  |
| --- | --- | --- | --- |
| **Usability Criteria (Heuristics)**  **Appropriate use of:** | **Level of Conformity**  **(1-4)** | **Evidence/Examples of application** | **Additional Comments/Issued Raised** |
| **Metaphors** | **1** | **-** | **No such metaphor** |
| **Consistency/**  **Perceived Stability** | **1** | **Screenshot (262)** | **Consitence layout for student portal** |
| **WYSIWYG** | **1** | **Screenshot (273)** | **Same pattern of displaying modules** |
| **User Centred**  **Control** | **2** | **Screenshot (367)** | **Easy to control** |
| **Feedback and Dialog** | **2** | **Change password in student portal** | **Successful**  **message appears and password gets changed** |
| **Buttons** | **2** | **Screenshot (367)** | **Same color button for delete and button** |
| **Colour** | **1** | **Screenshot (372)** | **Pinkish color in navbar** |
| **Modelessness** | **1** | **-** | **No any special mode.** |

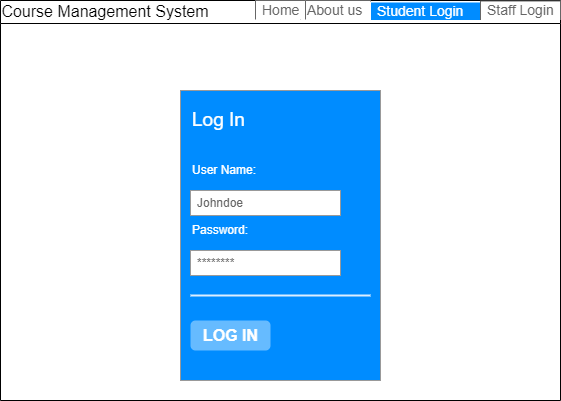
## **4.7 Draft Interface Designs (Woodlands University College Corporate Website)**

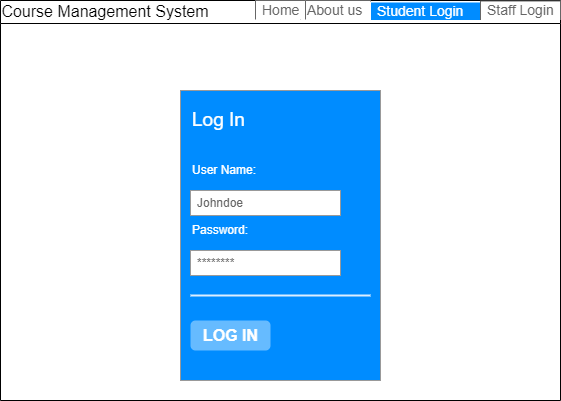
### 4.7.1 Wireframes

The sketch of how the initial system looks is a wireframe. The wireframes of the Woodlands University College Corporate Website are:



**Fig: Home page of corporate website**

**Fig: Student login to visit Student’s information portal**

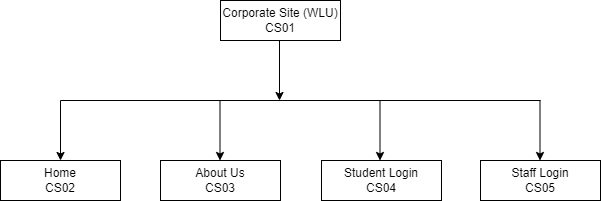


**Fig: Staff Login for staff portal**

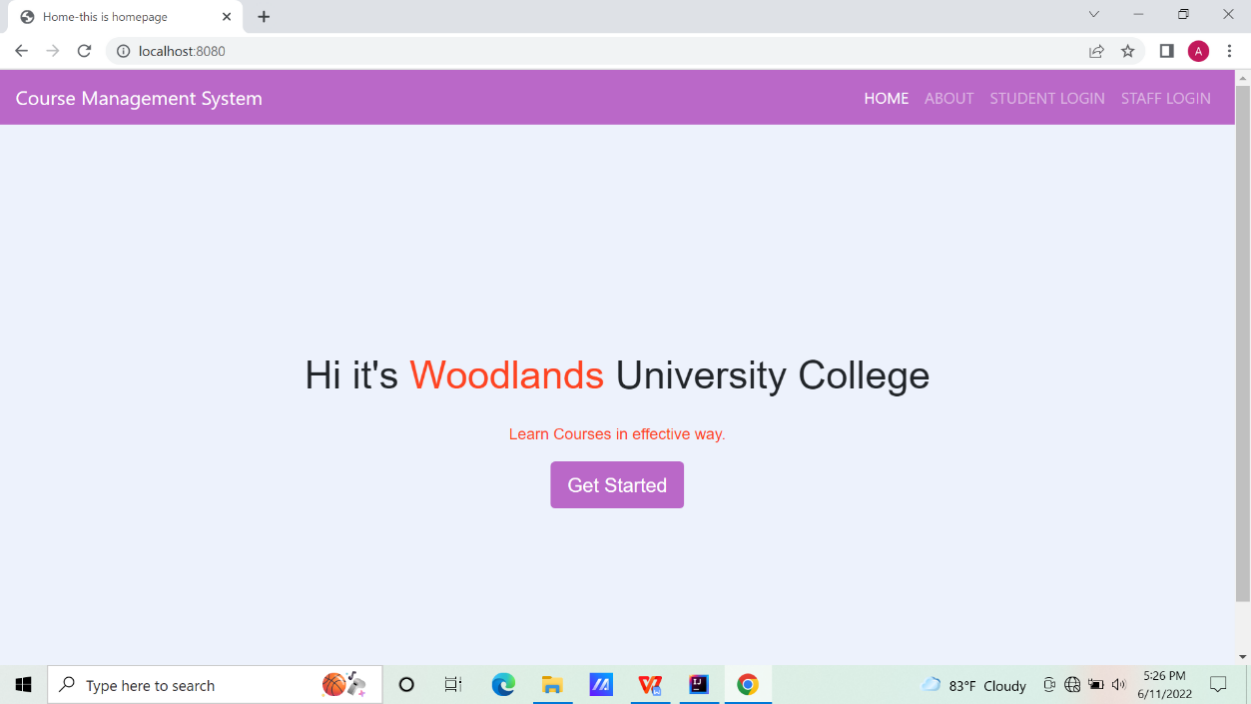
### 4.7.2 System Navigation Diagram

This topic includes an overview of the system, i.e. it navigates the pathways of the corporate website of woodland University. The system navigation diagram includes screen names, serial numbers, and arrows to illustrate the pathways.

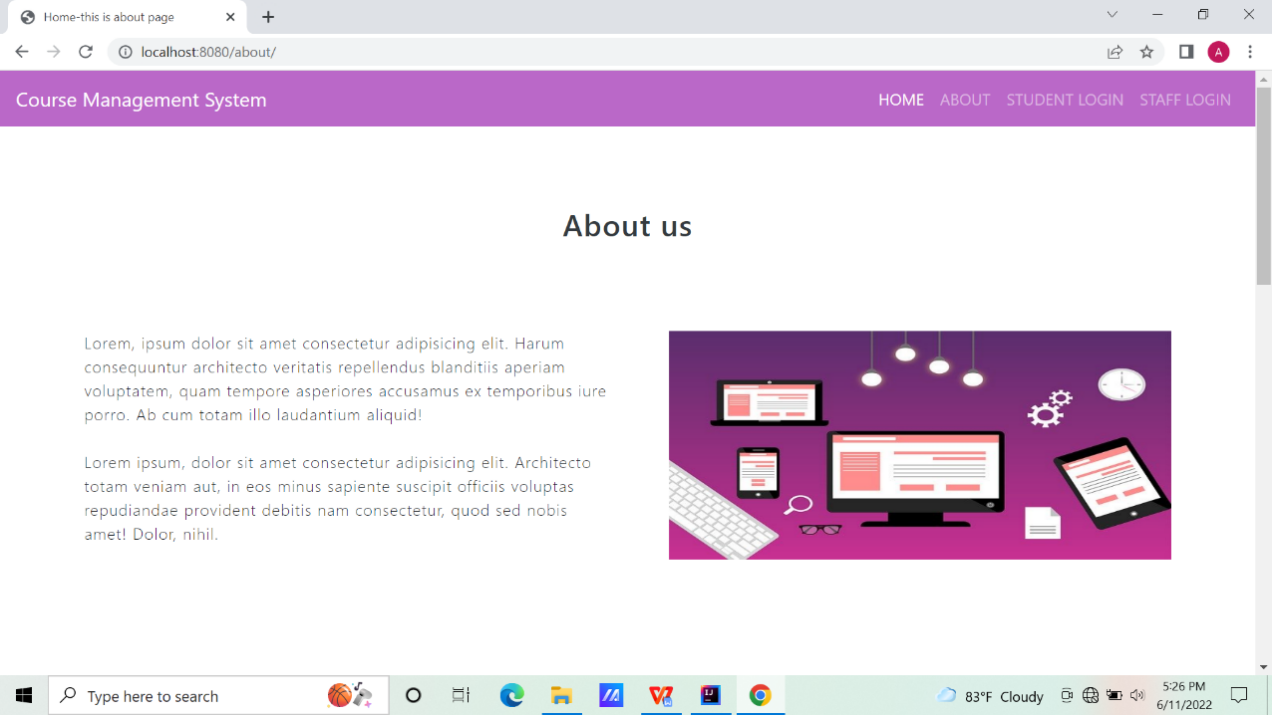
The system navigation diagram is given below:

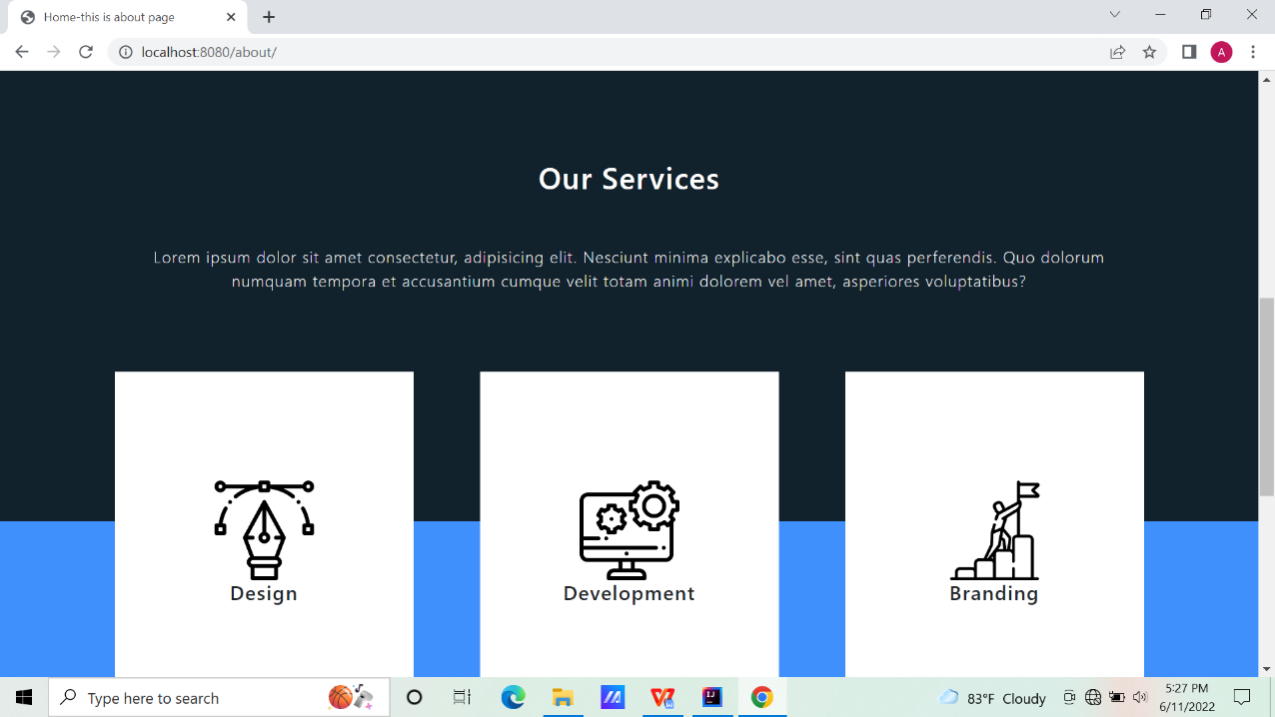


**4.7.3 System Screen mock-ups**

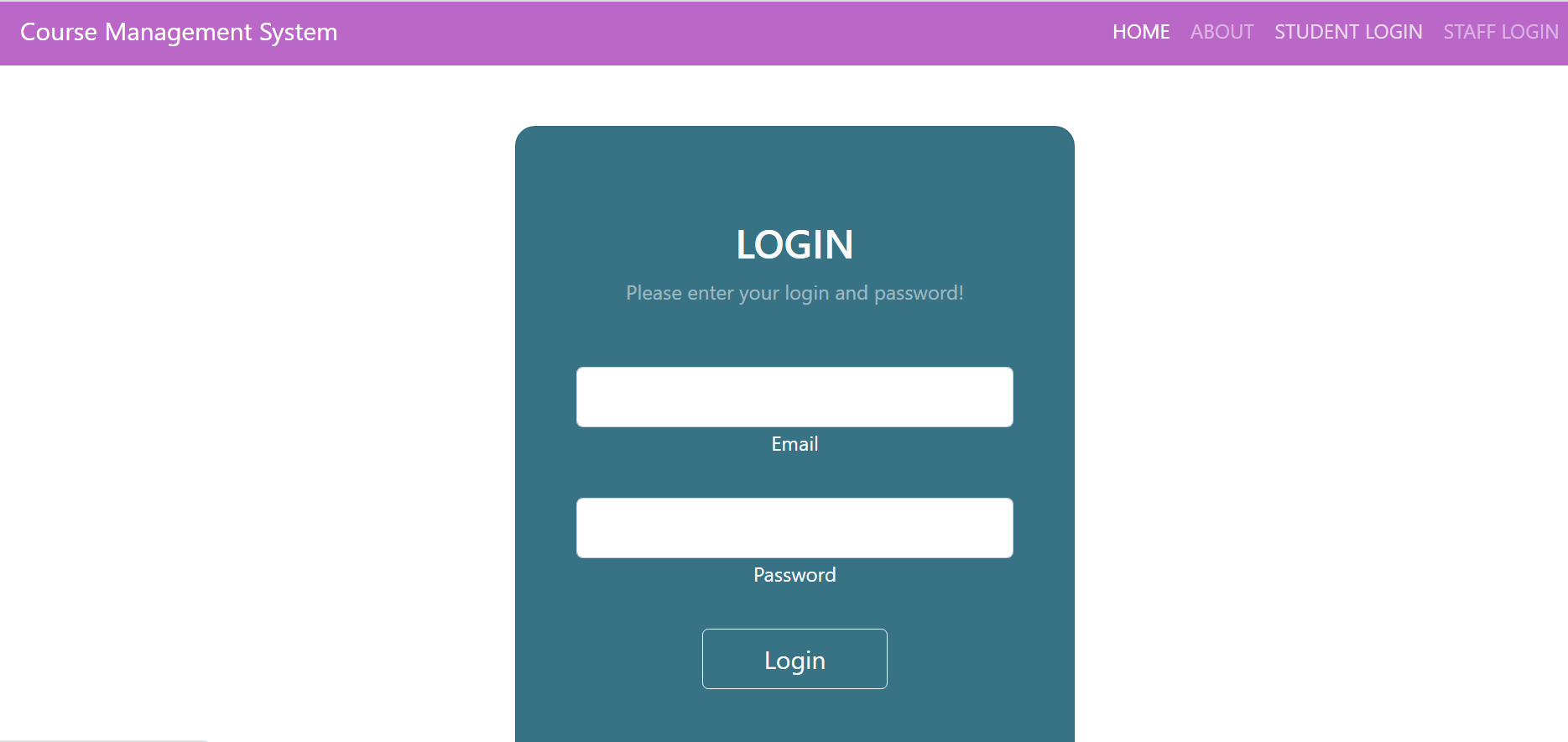
It is the final layout of the system. This layout is what the system looks like. System Screen mock-ups for the student record/information portal are given below:

**Fig: Mock-up of the home page of the corporate website of the university**





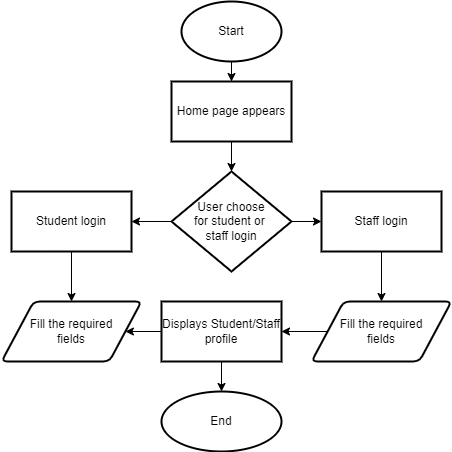
**Fig: Mock-up of about us page of the corporate website of the university**



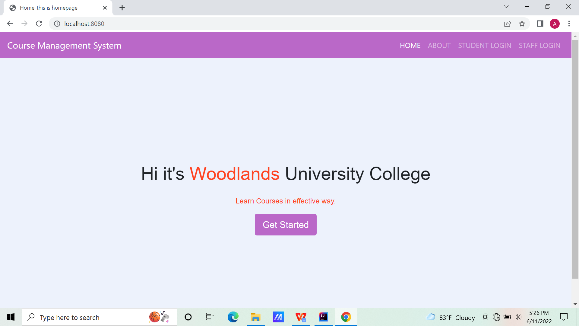
F**ig: Mock-up of Student Login page of the corporate website of the university**

### **4.7.4 System Activity Event Diagrams**

The system activity diagram for the corporate website of the university is given below:



**Fig: Activity Event Diagram Website**

**4.8 Design Revisions (Woodlands University College Corporate Website)**

The above-displayed wireframe, system navigation, mockups, and activity event diagram are all the draft interface designs for the course management system.

**4.9 Heuristic Evaluation (Woodlands University College Corporate Website)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Usability Criteria (Heuristics)**  **Appropriate use of:** | **Level of Conformity**  **(1-4)** | **Evidence/Examples of application** | **Additional Comments/Issued Raised** |
| **Metaphors** | **1** | **-** | **No metaphors** |
| **Consistency/**  **Perceived Stability** | **1** | **Screenshot (299)** | **Consistence landing page** |
| **WYSIWYG** | **1** | **Screenshot (302)** | **Courses represented with their logo** |
| **Buttons** | **2** | **Screenshot (299)** | **Button for get started** |
| **Icons** | **1** | **-** | **No icons used** |
| **Colour** | **1** | **Screenshot (300)Screenshot (301)** | **Simple Color combination for attractive UI.** |
| **Modelessness** | **1** | **-** | **No modelessness** |

# 5 System Build and Technical Notes:

The Server side language for the system is Spring-boot. This language is framework of java that helps to build web application in efficient way. Front-end of this system includes HTML, CSS and JavaScript. Java HTML5 template engine named as Thyme-leaf is used to cooperate with web environment. It is best for web application that is based on MVC. MySQL is use here to handle the database. Different dependencies like spring-web, spring-security, MySQL Driver, Spring Data JPA, Thyme-leaf, etc are used in pom.xml file of the system. These dependencies are essential for perfect web apps.

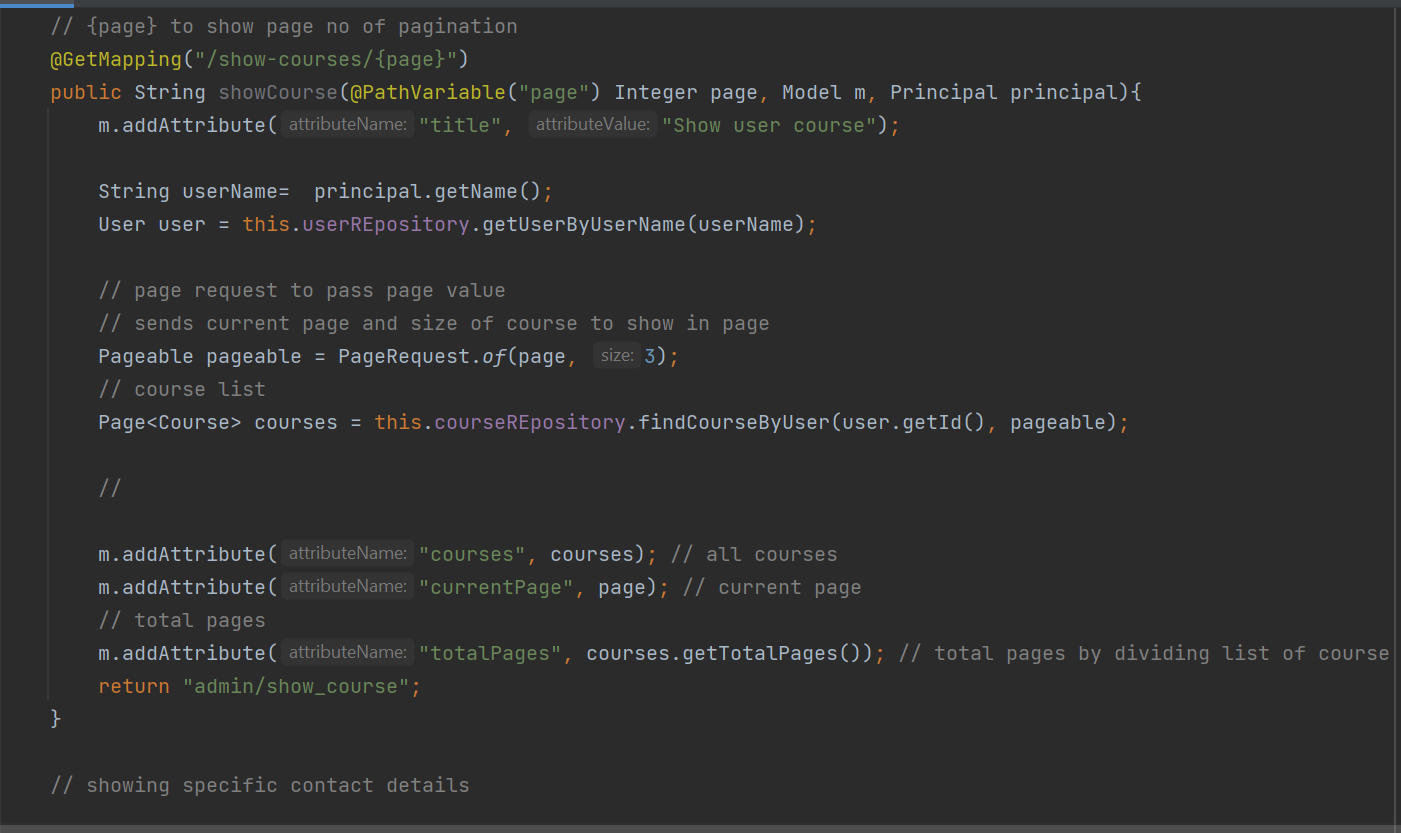
## 5.1 Technically Difficult Code Section (RMS)

Model-View-Controller helps to separate and provide functionality and business logic that helps in proper maintenance of the system. System compromises of different coding section and different logic’s. Collaboration of multiple business logic and coding section helped in implementation in this system. Different packages for each section can also help in efficient management of system. There are different difficult coding part in the system. An example of Technically Difficult Code Section are:

### 5.1.1 Showing all courses in database:

Showing all courses that admin has added was difficult task. The course that admin has added is referenced by using admin id and selection of course is done by courseREpository. Principal refers to the current logged in user details(I.e. of admin)

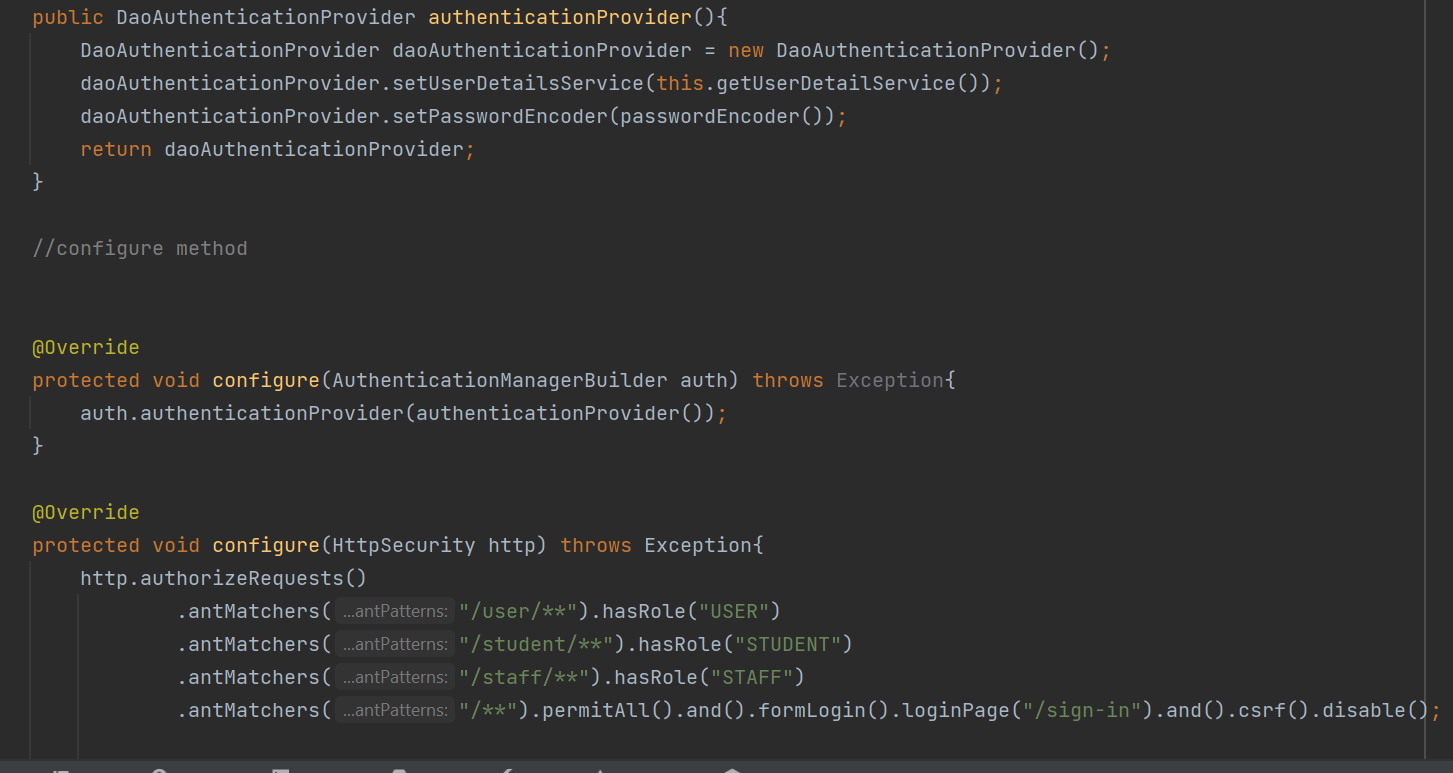
After getting admin id, course is obtained by using findCourseByUser, which query is written in courseREpository. If there are huge no of courses, showing and viewing list of all courses would be difficult task. So the concept of pagination is taken here. Each page having only 3 course list is requested by using PageRequest.of(page, 3). After getting all courses, courses are divided and each page showing list of 3 courses are returned in show\_course.html file.

****

**Fig : Show all Courses code**

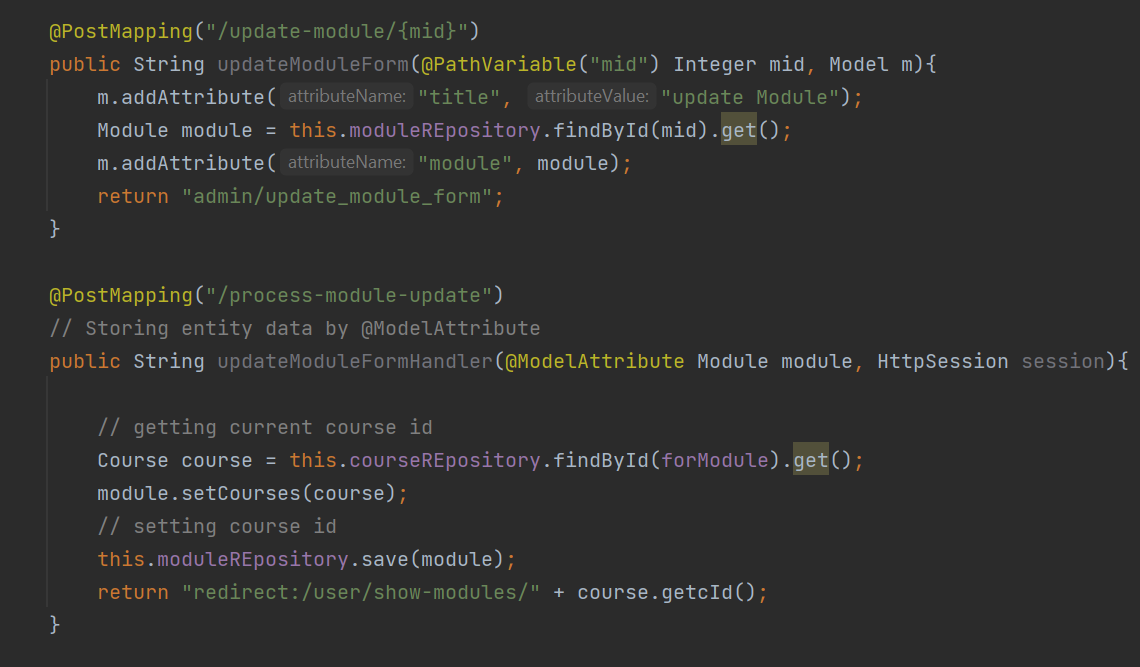
### 5.1.2 Authorize access:

Authorized access in the system is one of the important part of the system. Spring security helps us to provide security in the required system. Giving all authorization to single user(I.e. admin) and giving selected responsibility to selected user is managed by this code section. DaoAuthenticationProvider access the user information from relational database especially username and password. If user name and password are matched then different HTTP request are handled for different user. If user has role “USER” in database which is authorize as admin user in this system, then the system accept URL’s request containing “/user” for only admin user. Like wise user having role as “STUDENT” can only access URL containing “/student” and user having role as “STAFF” can only access URL containing “/staff”.

****

**Fig: Code for authorization access**

### 5.1.3 Update and delete selected module :

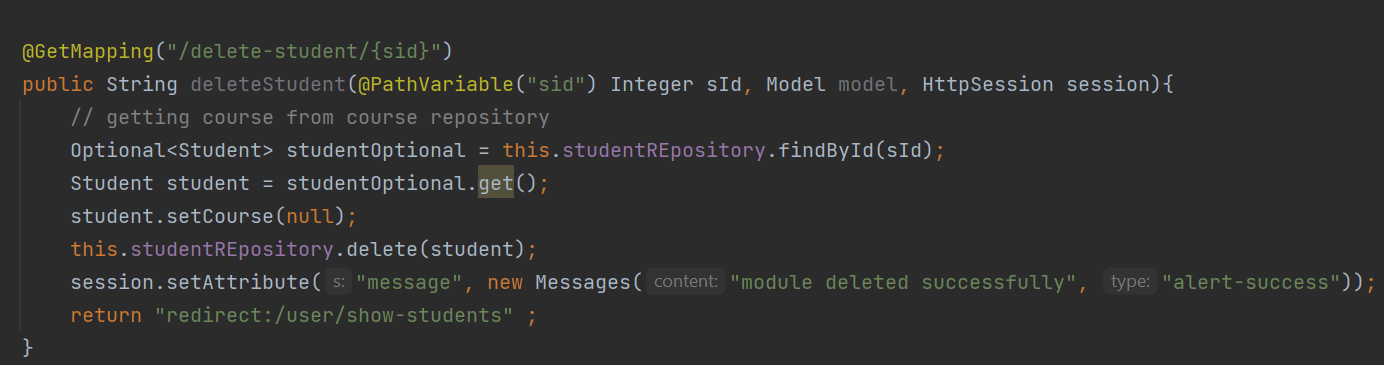


**Fig: Code section of update and delete module**

Module is related or bind by course id. One course can have many modules. So, there are list of modules in the specific course. In CRUD operation we can update the items from database. Deleting specific content from database can be headache in the programming. In this system, by clicking update button on the specific module, the module Id is passed with URL , which is stored in “@PathVariable” in upadteModuleForm method. After that update form for module is returned. By updating value if we click in save button the processing of module update begins. In this process, “@ModelAttribute” gets all the value entered in form. Course is obtained and stored in new updated module table. Stored value is saved in module table and the user is redirected to show modules page.

### 5.1.4 Delete Student:

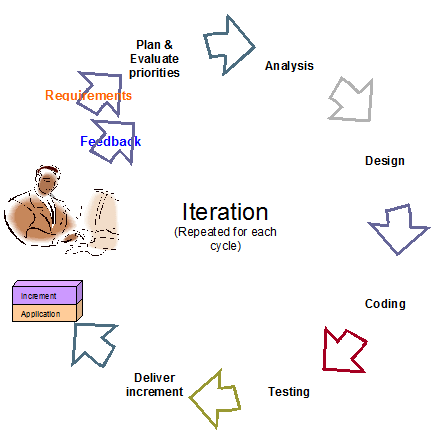
In CRUD operation deleting specific content is very important task. Deleting specific selected object without affecting other objects detail makes this task technically sophisticated. Deletion of student is done by deleteStudent method. The student id is obtained from the url. The desired student is get by student id. To delete student course field of that object is set null to handle error. Deletion of that student record is done by using studentREpository. After deletion “Successfully deleted” message is sent to HTML and the user is redirected to show students page.

****

**Fig: Delete Student Section in controller**

### 5.2 On-going testing methodology:

Software development relates development and testing of system side-by-side. In pace of software development proper testing is considered as important point for successful software or prototype. So, there are many testing methodologies that helps to obtain required system. Some of them are: agile methodology, waterfall model. Here, Agile methodology is considered as best testing methodology for the required prototype or system. The Agile methodology is a process to manage a project by breaking or dividing it into smaller parts. Different functionality is break down and completed step by step. It involves consistent collaboration or support with stakeholders. For desired prototype, team work in cycle of planning, executing and development. Continuous collaboration can be virtual or physical. Team and stakeholder have meeting either physically or virtually so that there is great possibility of accuracy and acceptance of required functionality in the system.



**Fig: showing the cycle of agile methodology testing.**

Here, whole system is divided into different segments (like course management system, module management system, student management system). Taking an example of course management system, It was extracted from system and the requirement of this system gathered. After that, different plans were gathered as part of elicitation by virtual and physical meeting with stakeholders. Analysis of system was in process through different ER, UML diagrams. Different entities/class were designed. Wire-frames and mock-up of required system was prepared with the collaboration with team members. After successful design of system, required environment for system was setup and coding began along with different testing strategies like Black box testing. Where, required output was obtained from list of inputs. Functionality of system was tested along with non functional (like speed, capacity and usability) of system was done. As designed functionality was sufficient then the system was integrated for complete system. Acceptance of system was confirmed with different set of stakeholder and users and after successful confirmation, development process for module management system began. Similarly, other parts of whole management system followed the path of agile methodology.

### 5.3 Final System Interface Display:

### **5.3.1 Record Management System:**

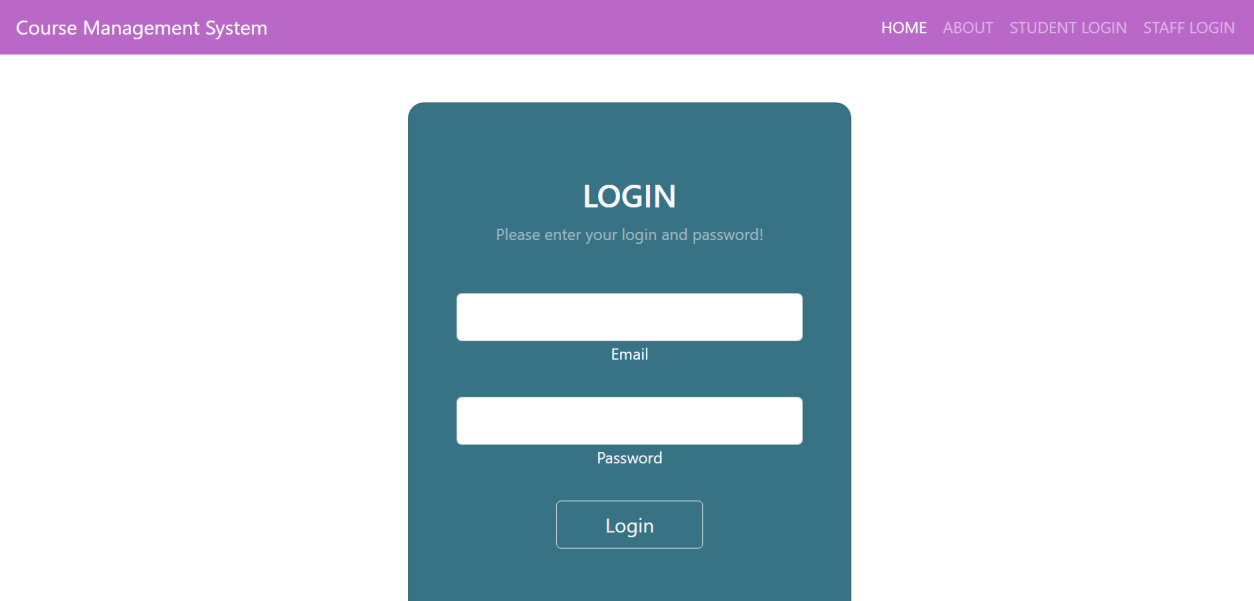
****

Figure 67 Login Page (final display)

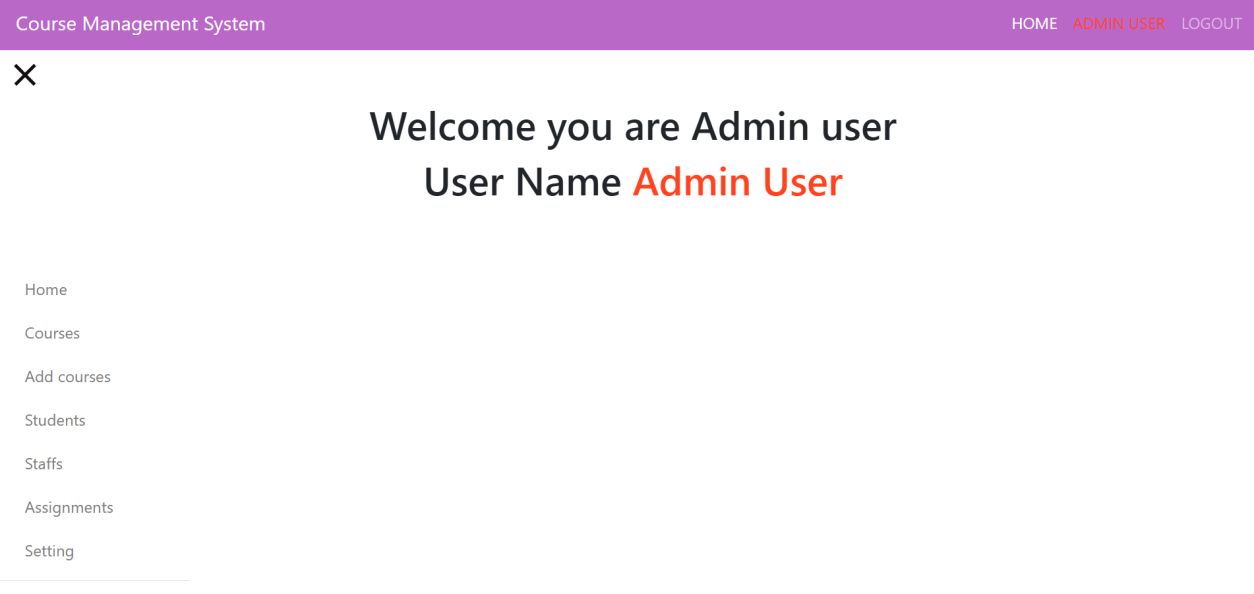
****

Figure 5 admin dashboard (final display)

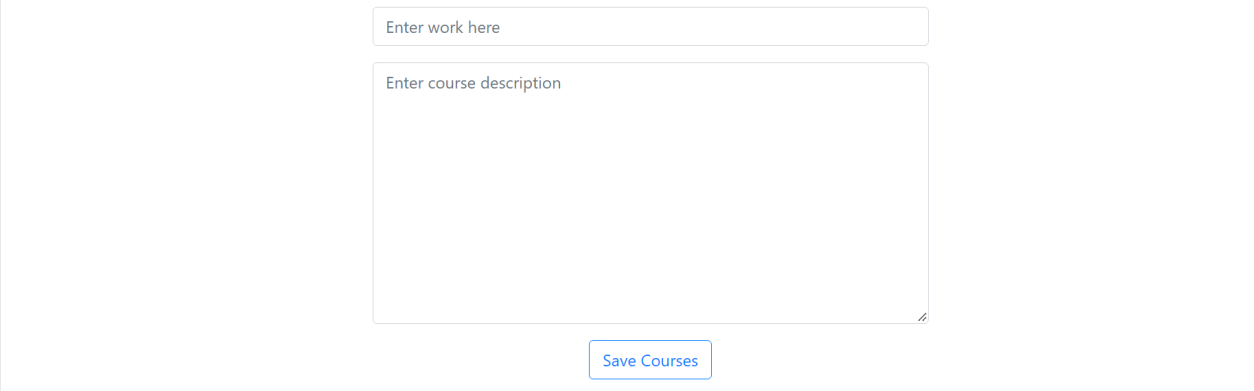
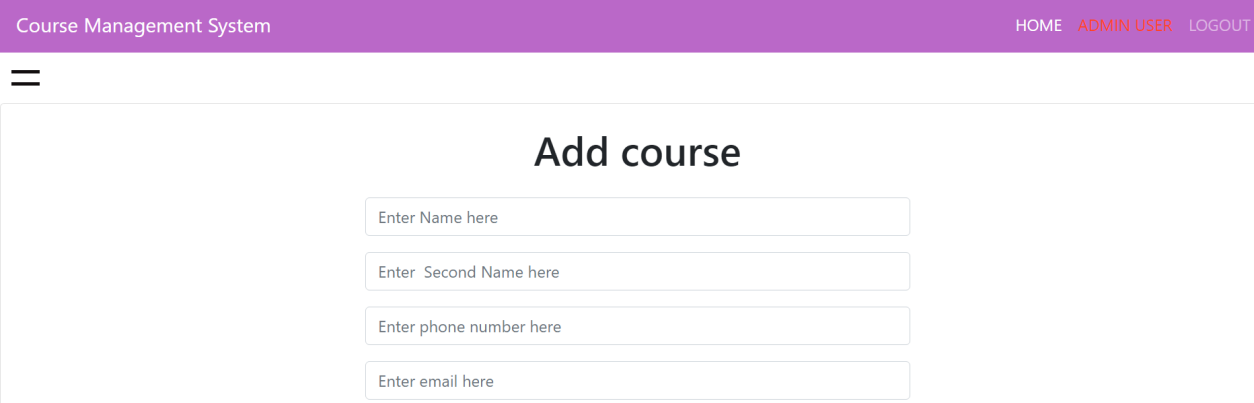


Figure 6 add course (final Interface)

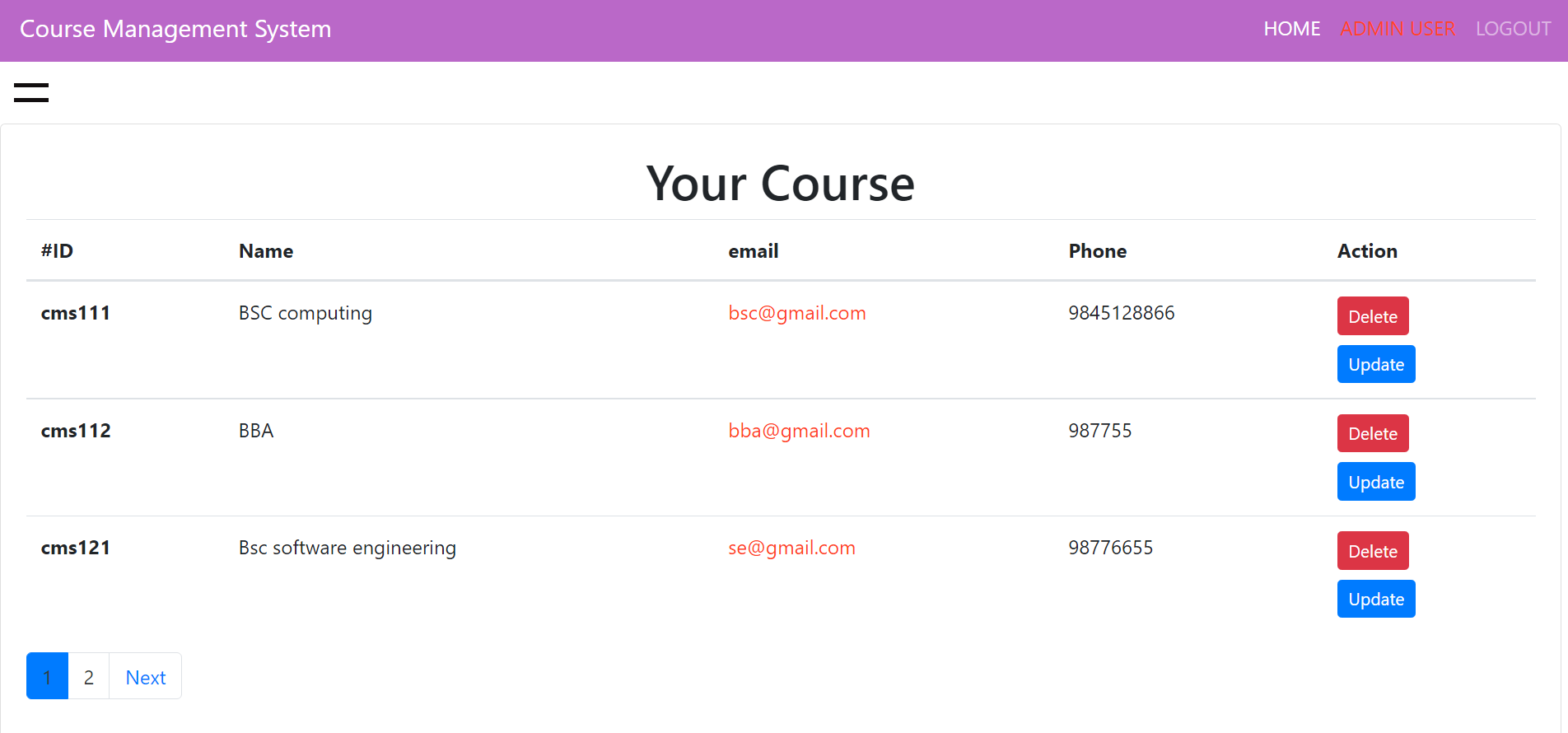
****

Figure 68 show course

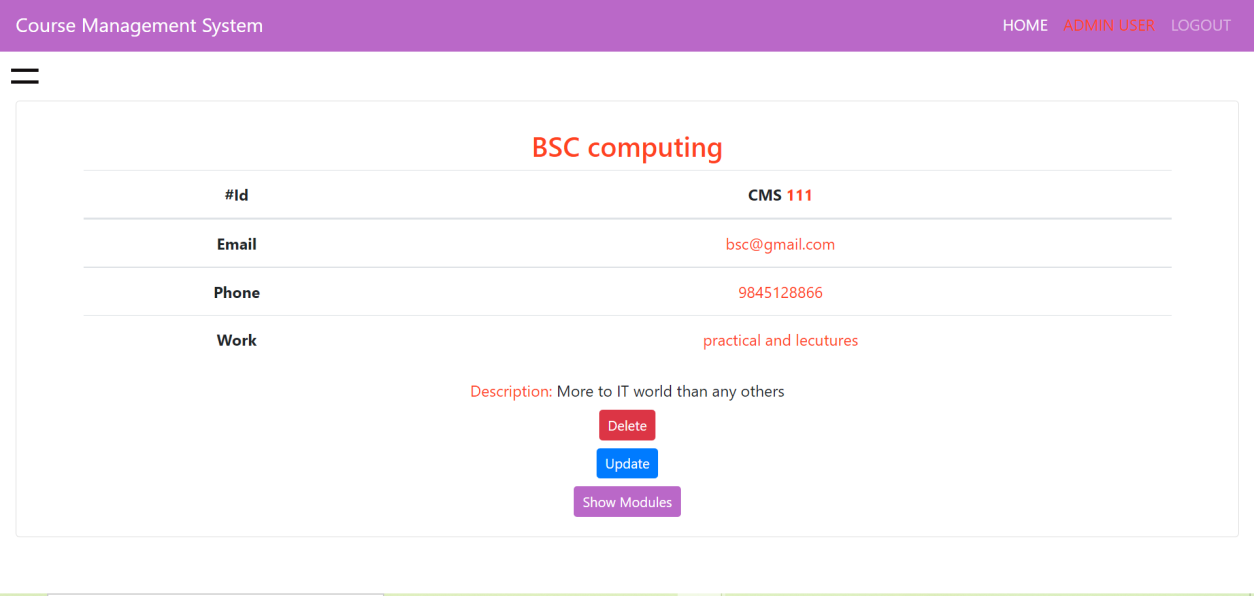
****

Figure 69 course detail (final display)

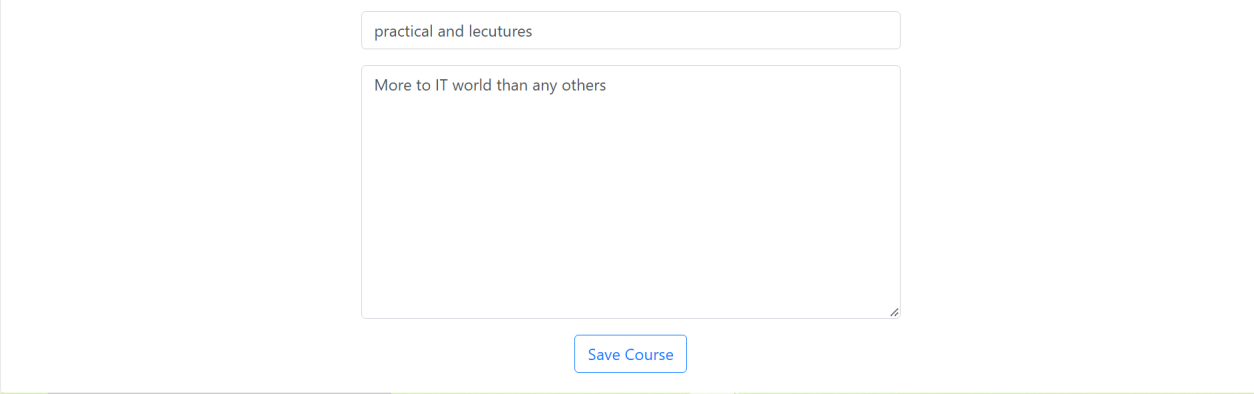
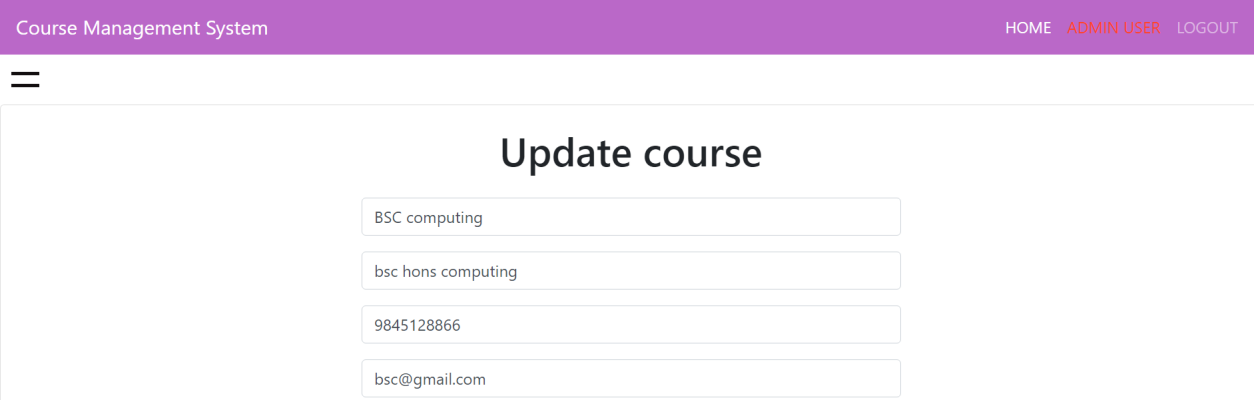


Figure 70 update course (final display)

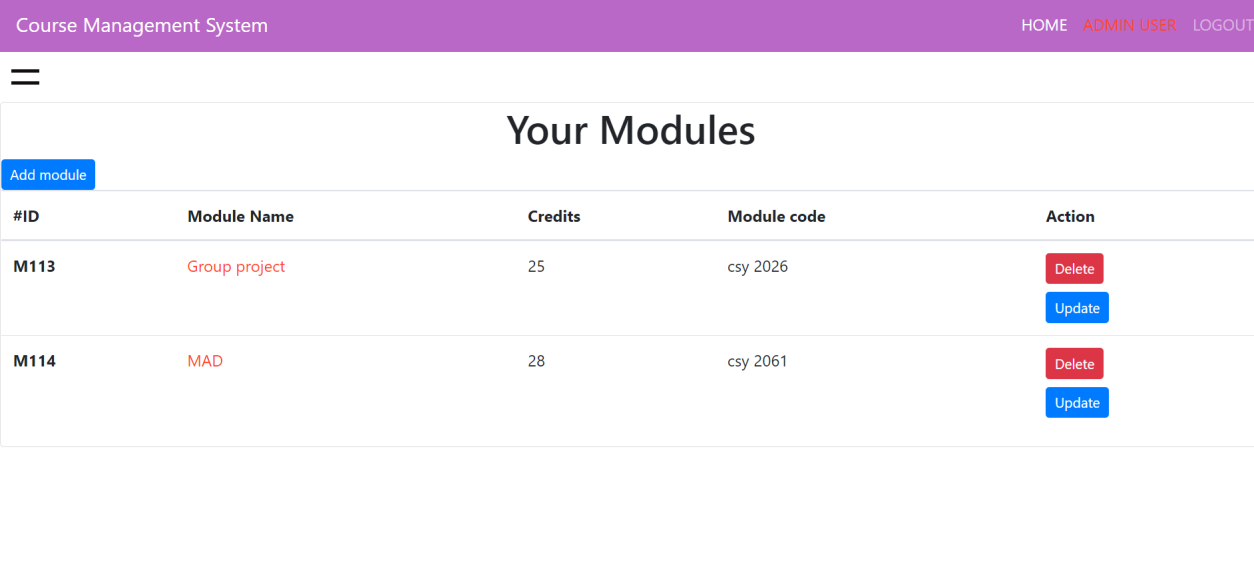
****

Figure 71 show modules page

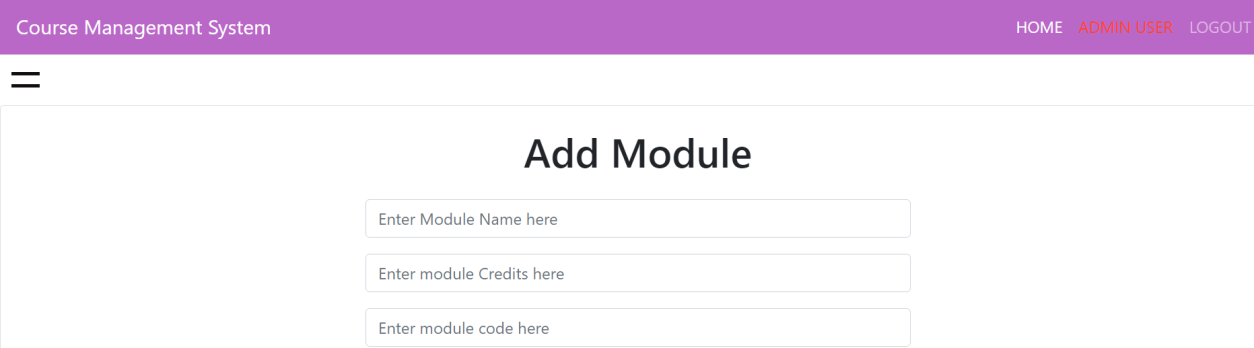
****

Figure 72 add module page



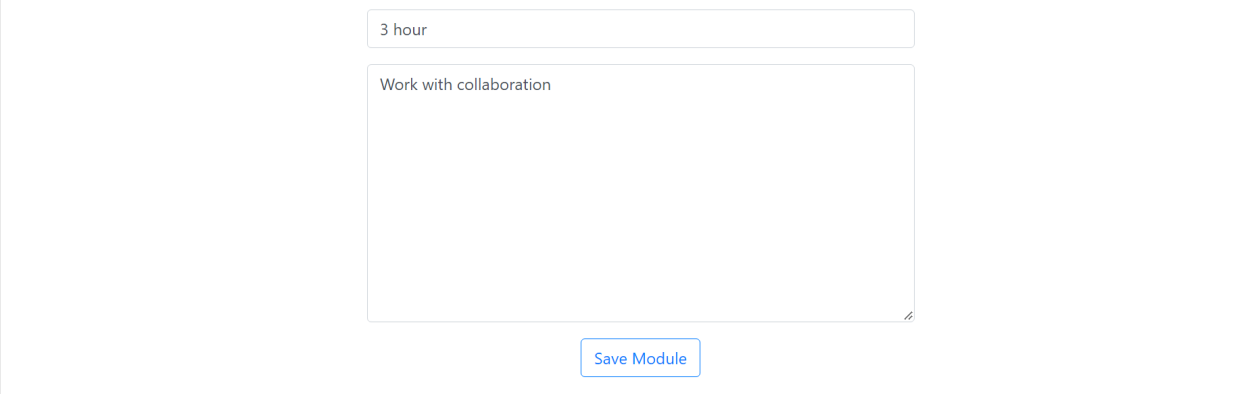


Figure 73 Update module



Figure 74 show student (final display)

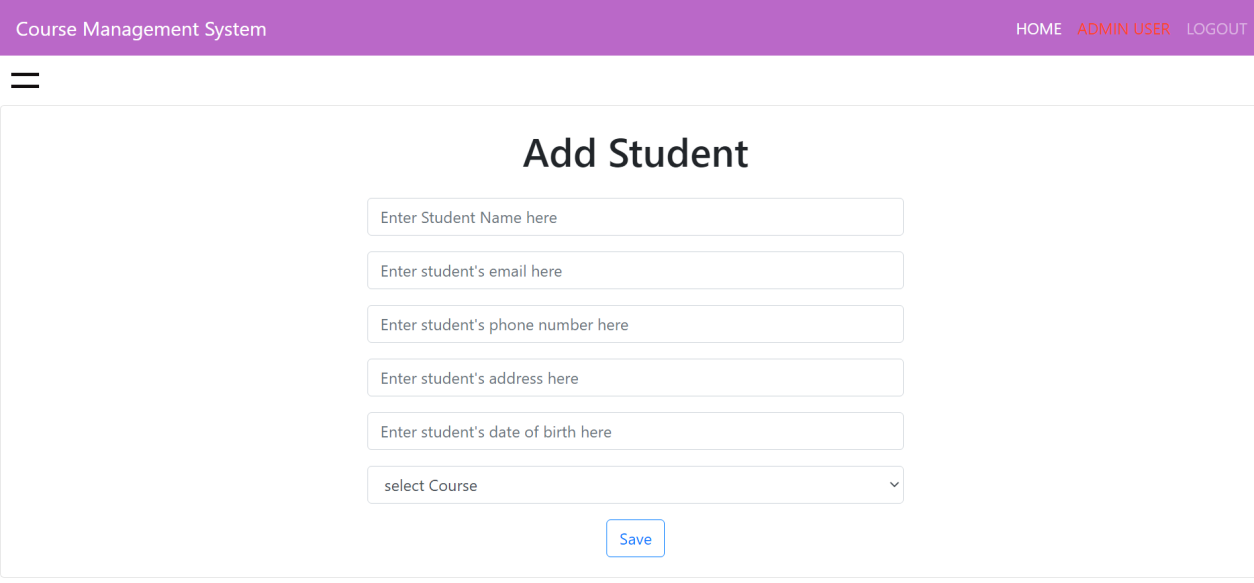


Figure 75 add student

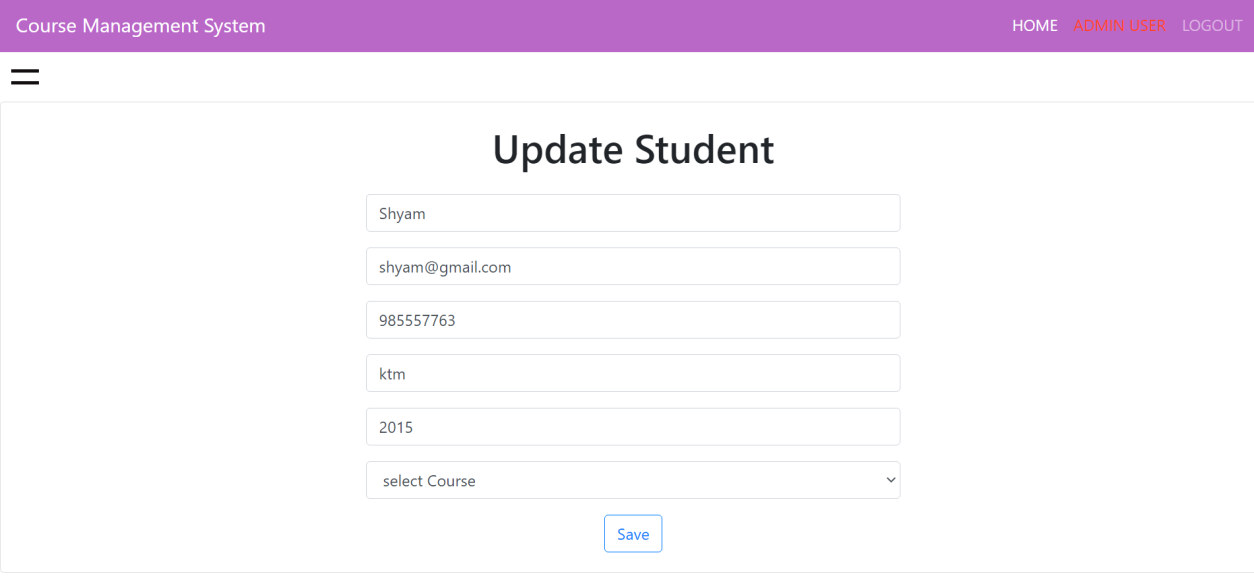


Figure 76 update student

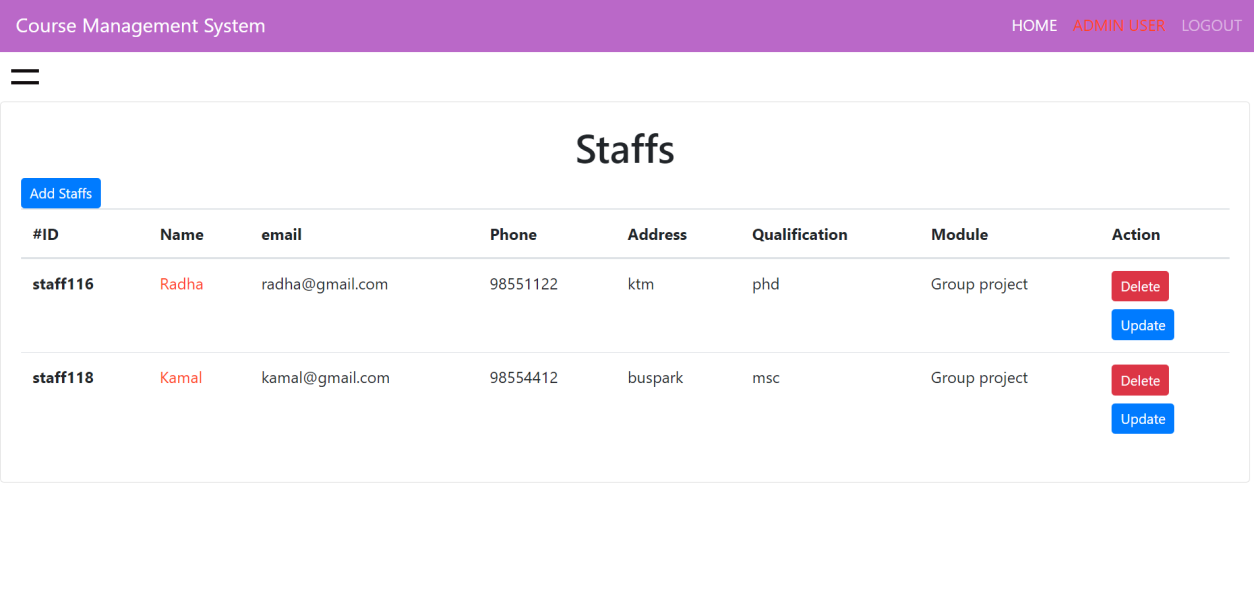
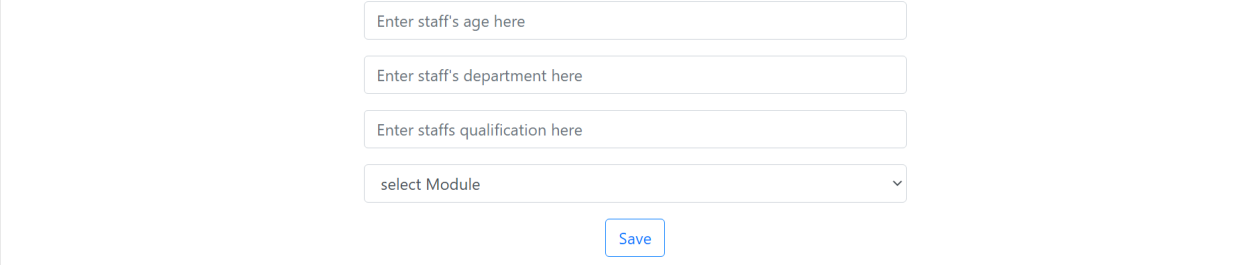
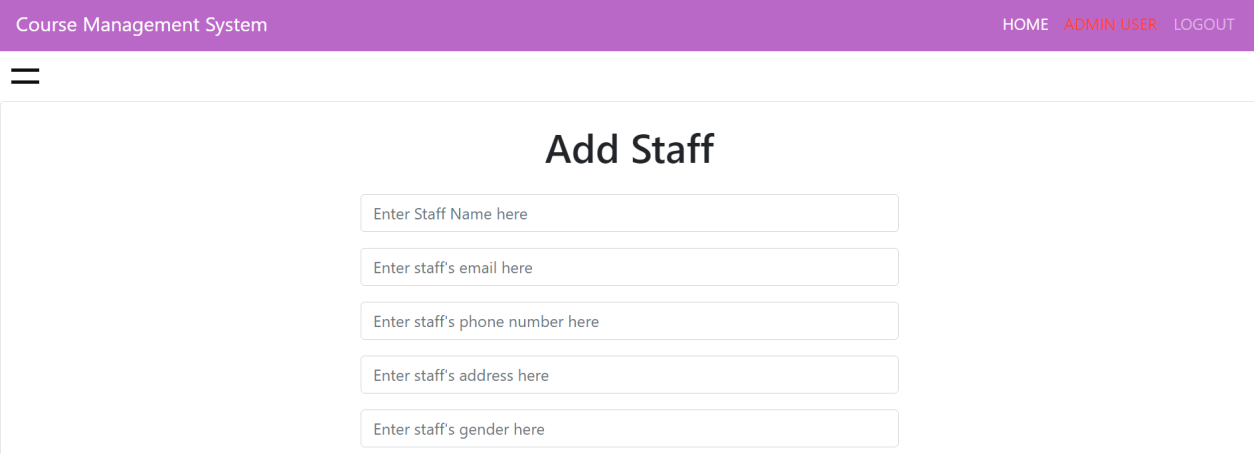
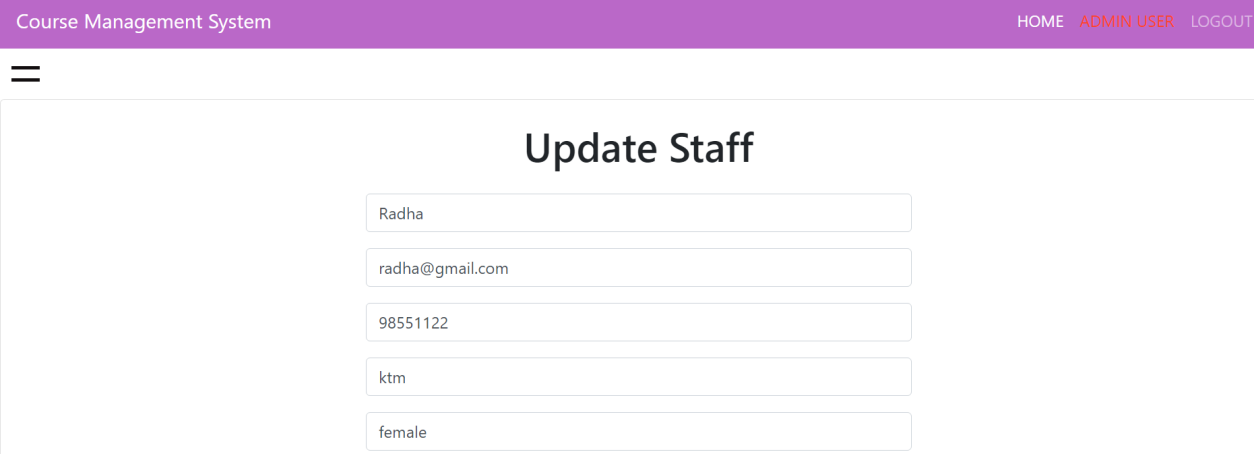
****

Figure 77 show staff page

****

****

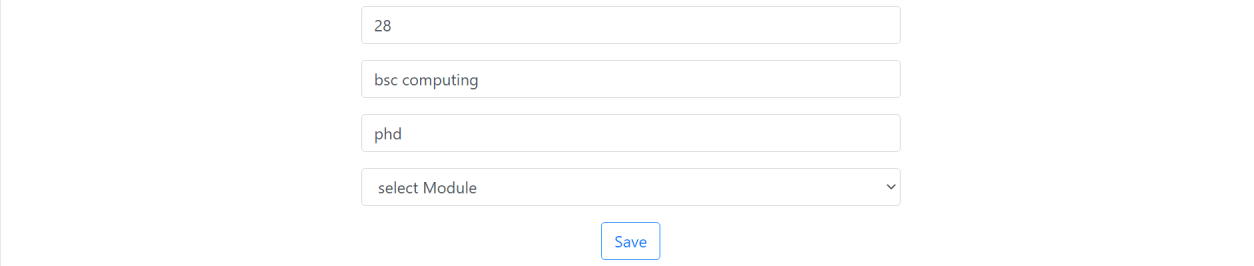
****

Figure 78 update staff (final display)

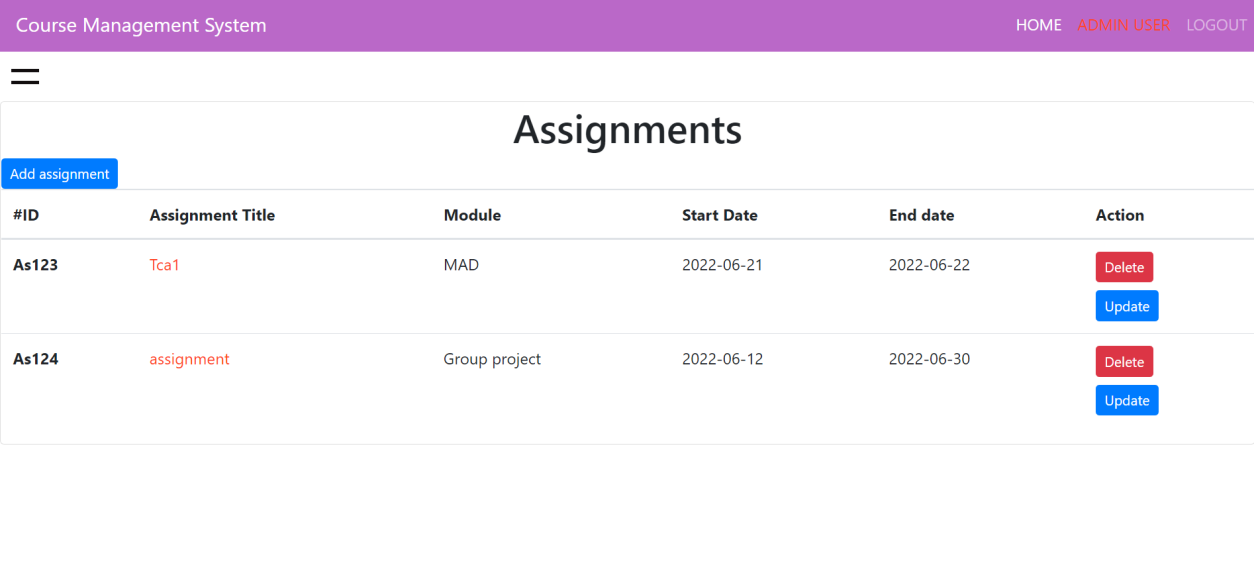
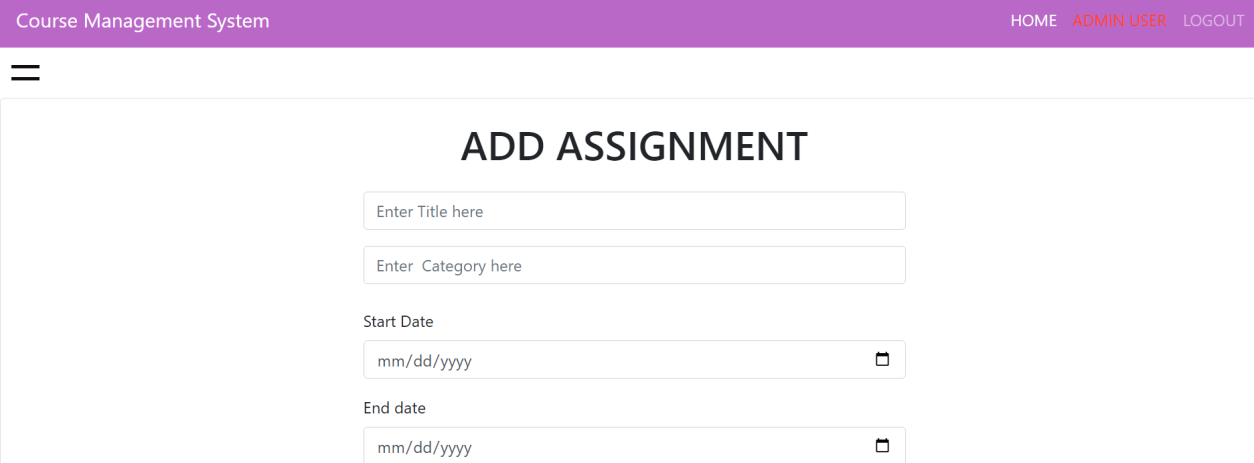


Figure 79 show assignment

****

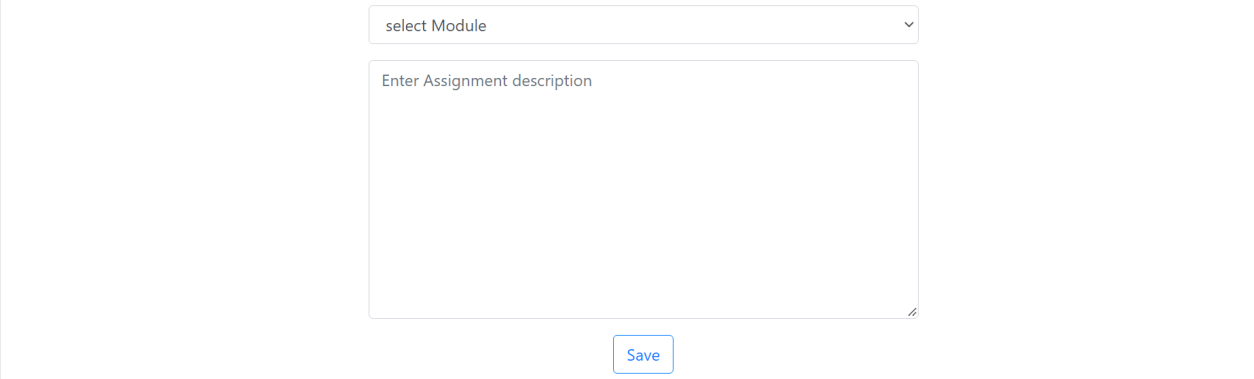
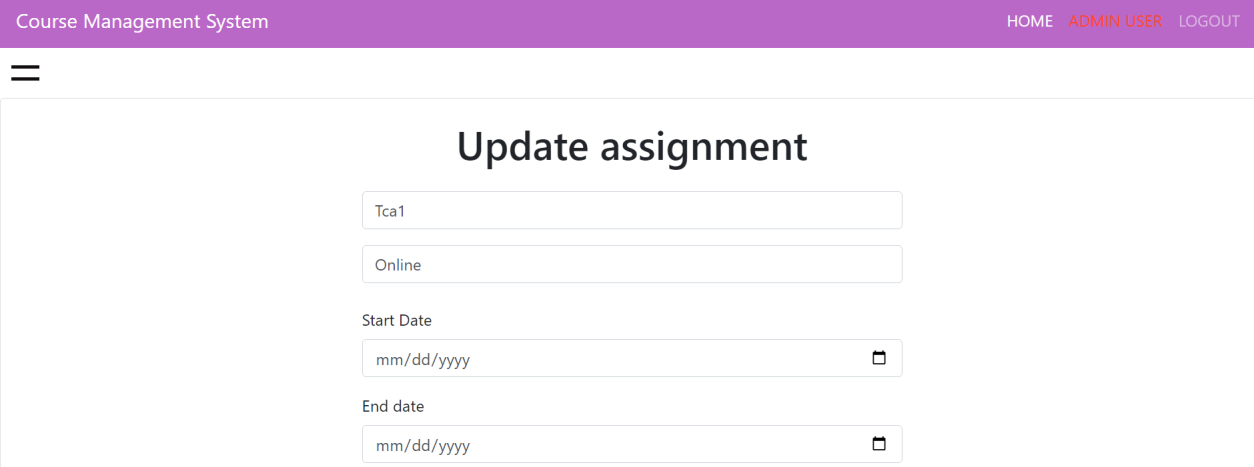
****

Figure 80 add assignment

****

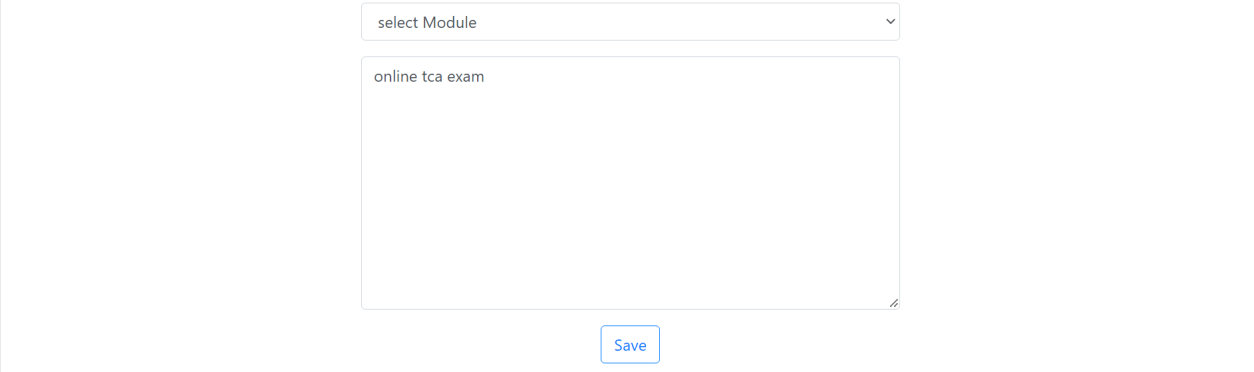
****

Figure 81 update assignment

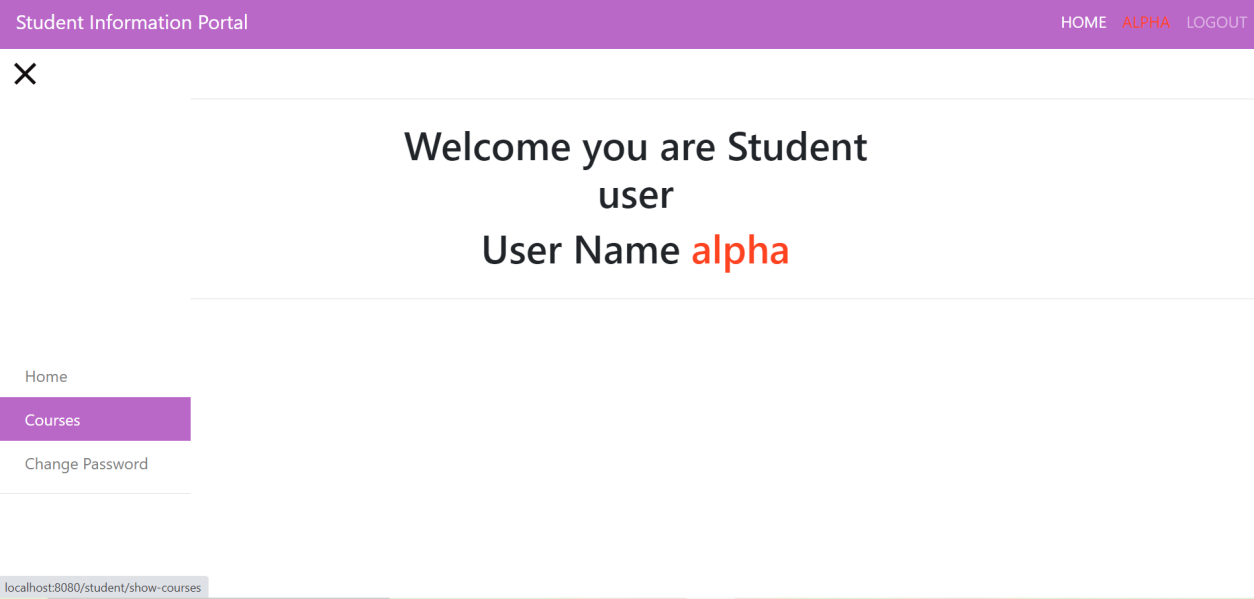
****

Figure 82 student portal dashboard (final display)

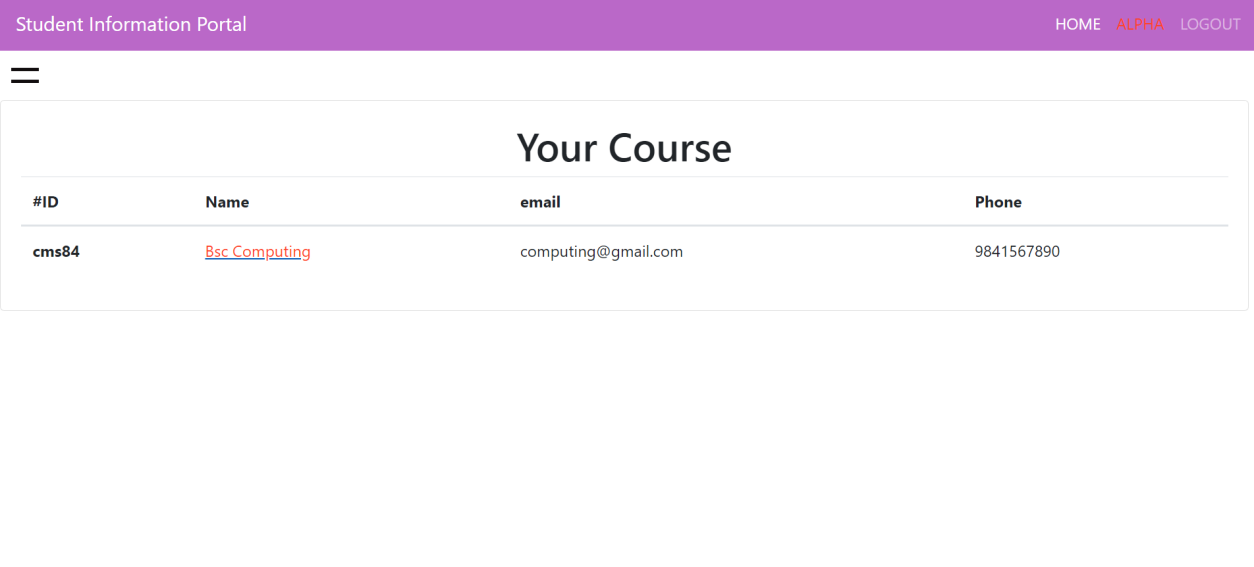


Figure 83 course that student is enrolled in.

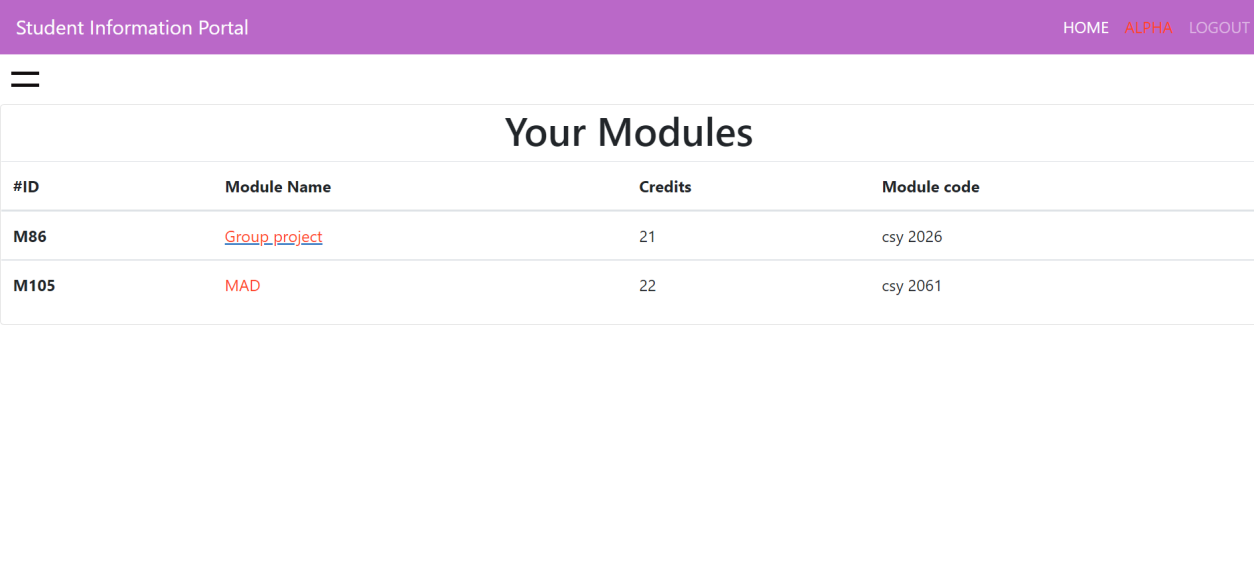


Figure 84 student enrolled module

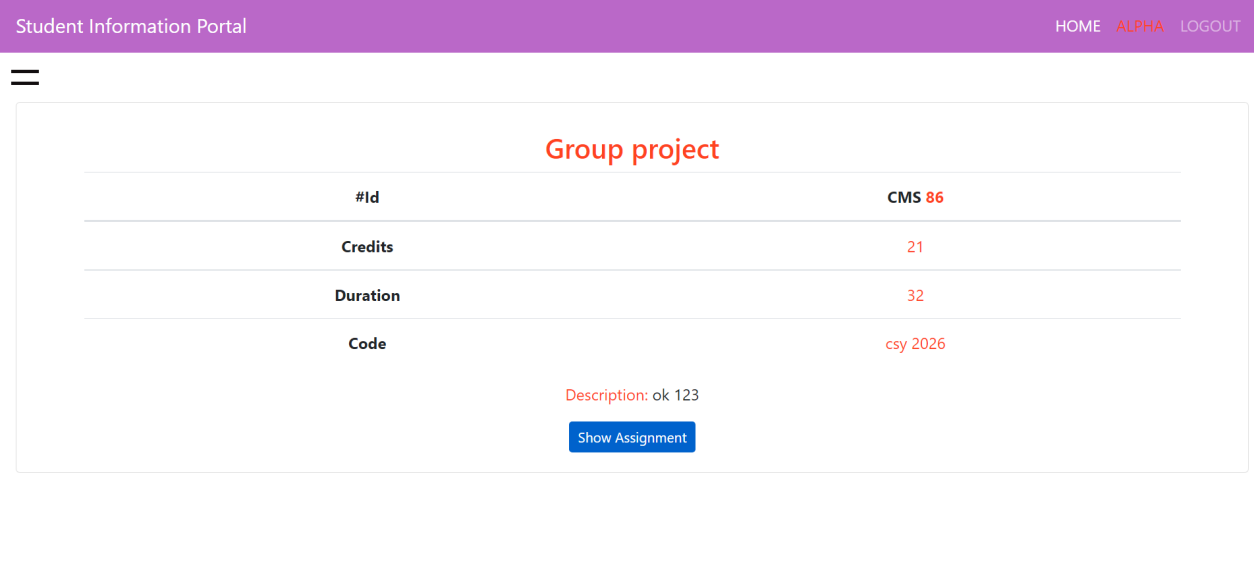


Figure 85 page showing module details

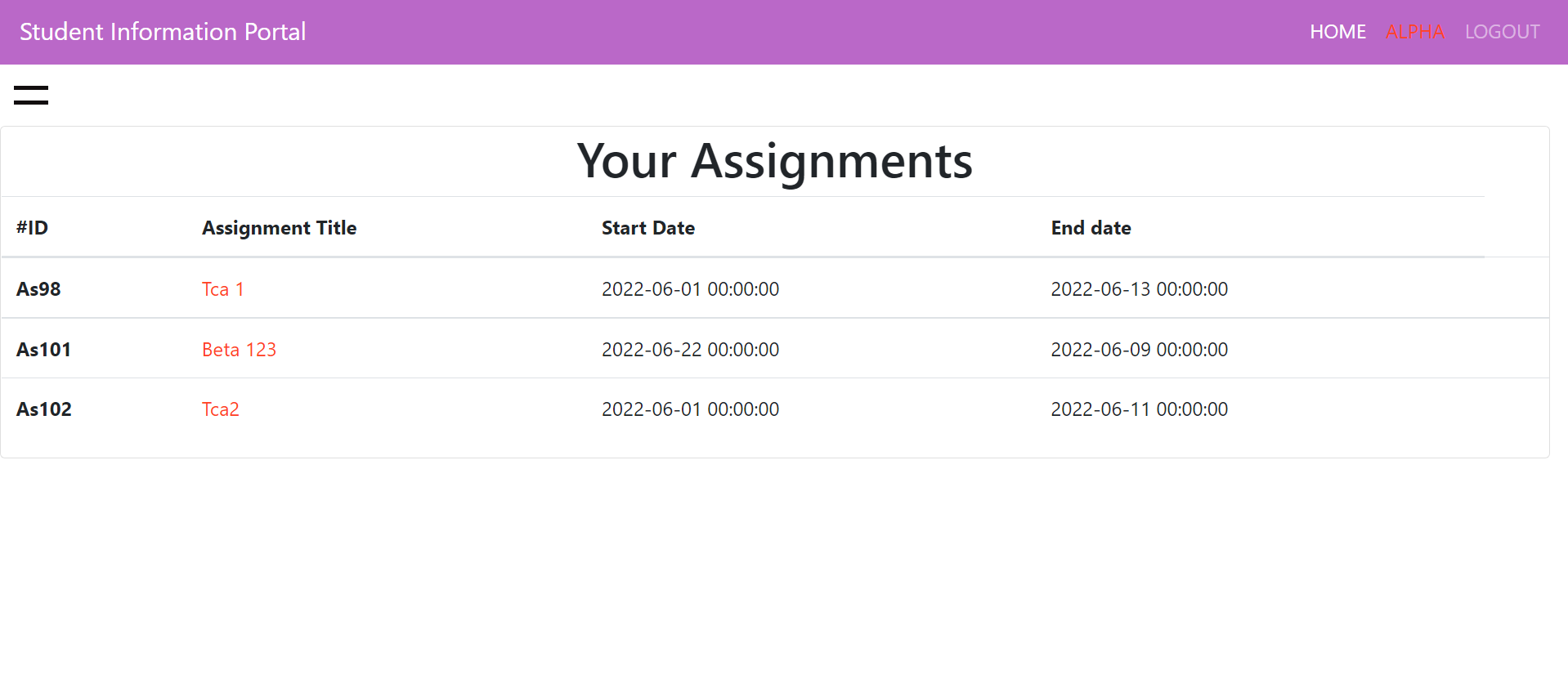
****

Figure 86 Showing assignment in that module (final display)

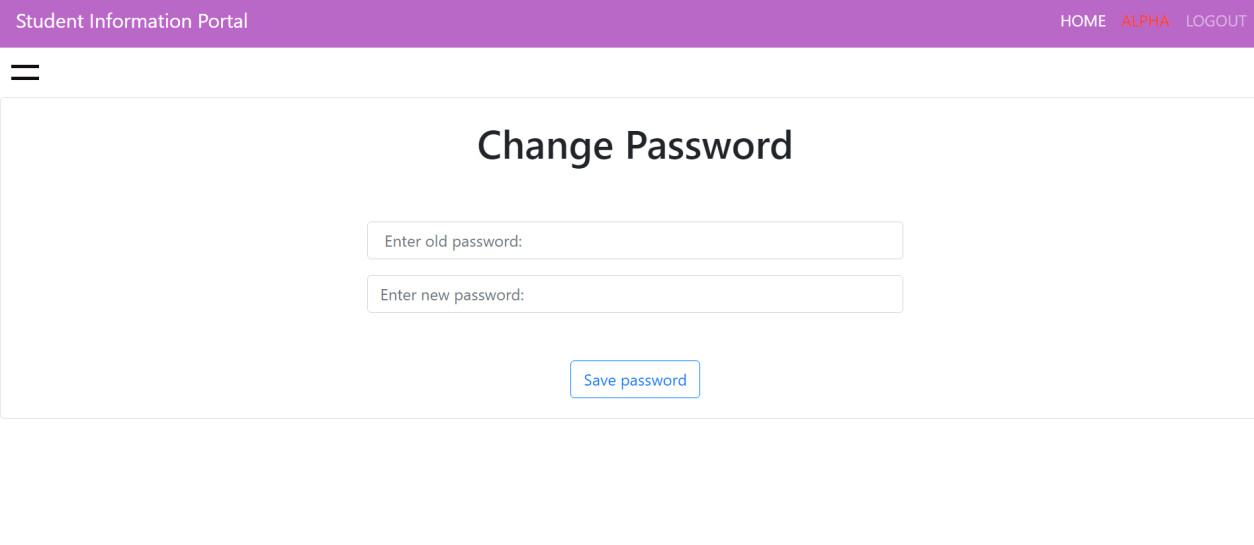


Figure 87 change password in student portal(final display)

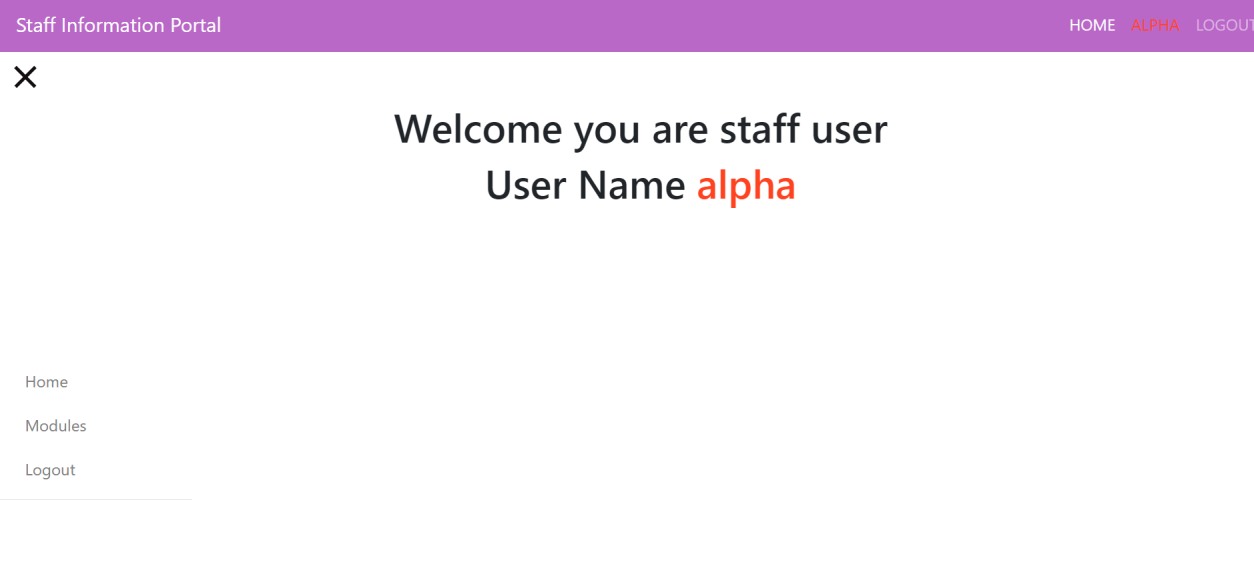


Figure 88 staff dashboard(final display)

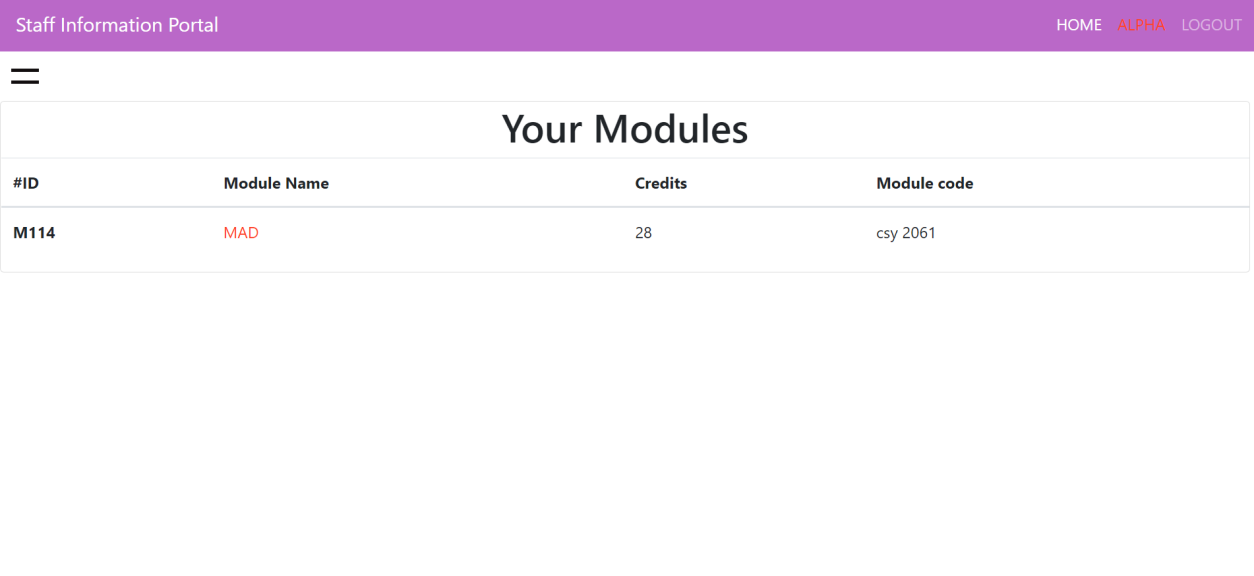


Figure 89 show modules of staff portal

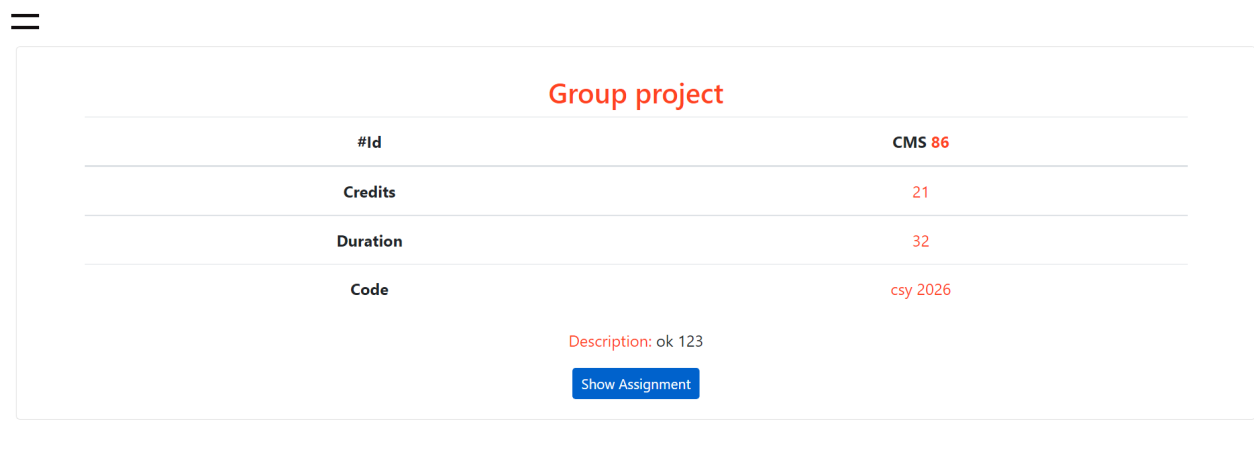


Figure 90 showing module detail

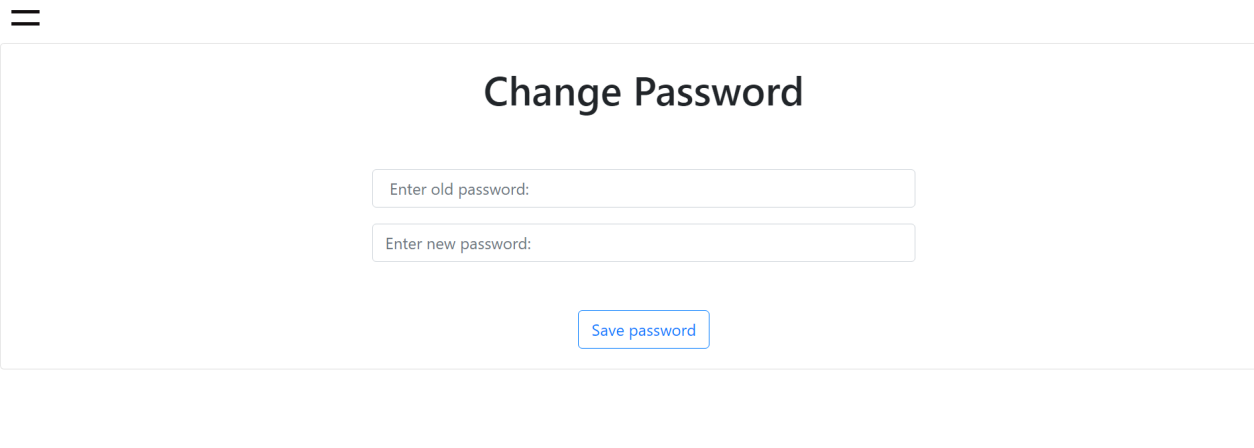


Figure 91 change password in staff portal

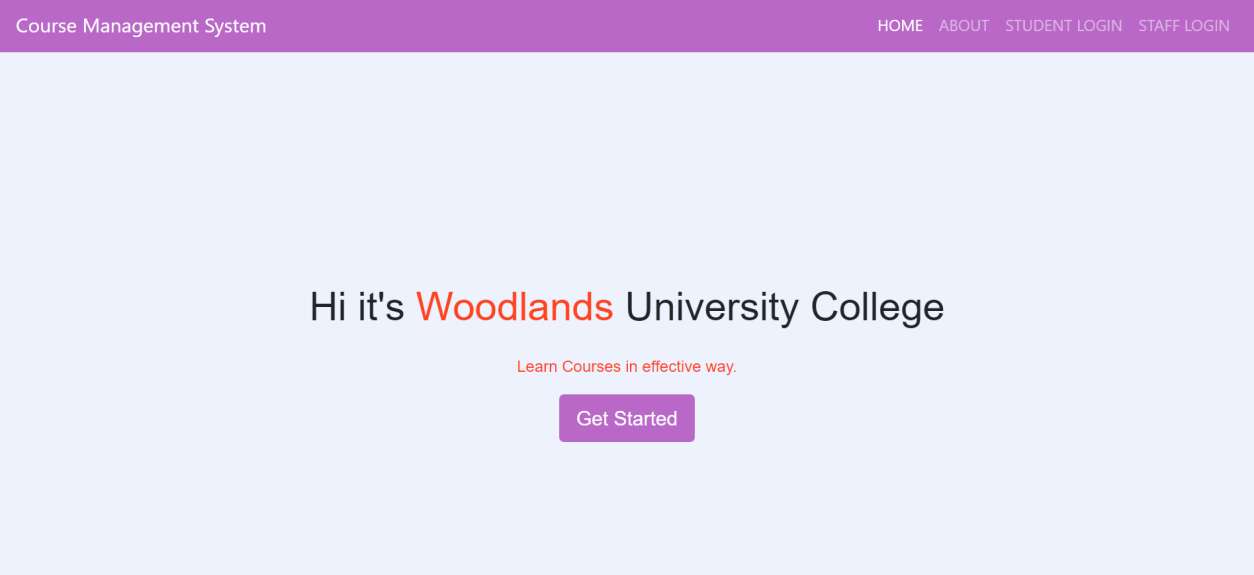


Figure 92 University website homepage (final display)

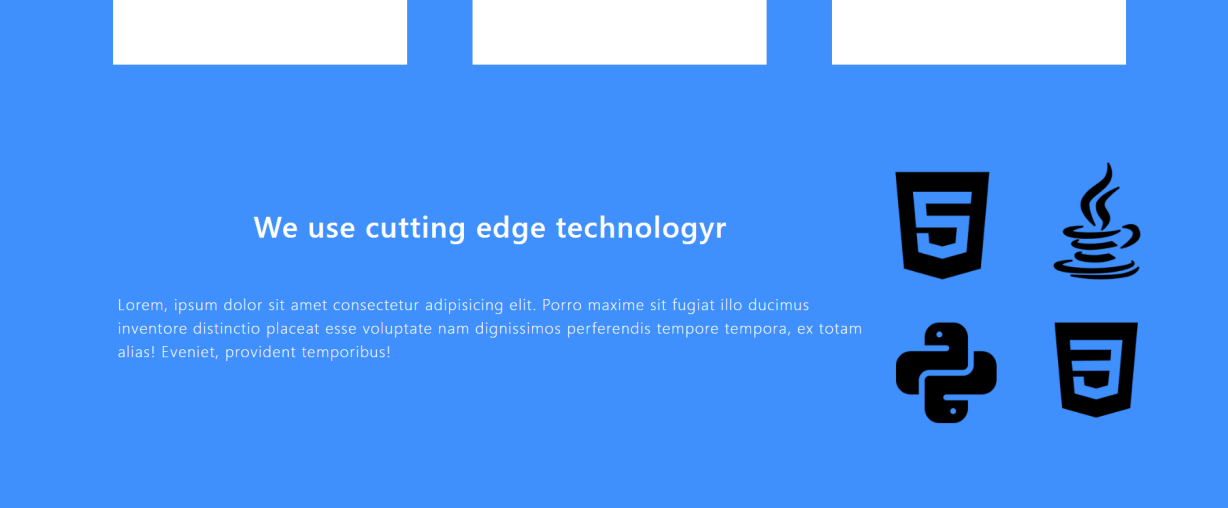
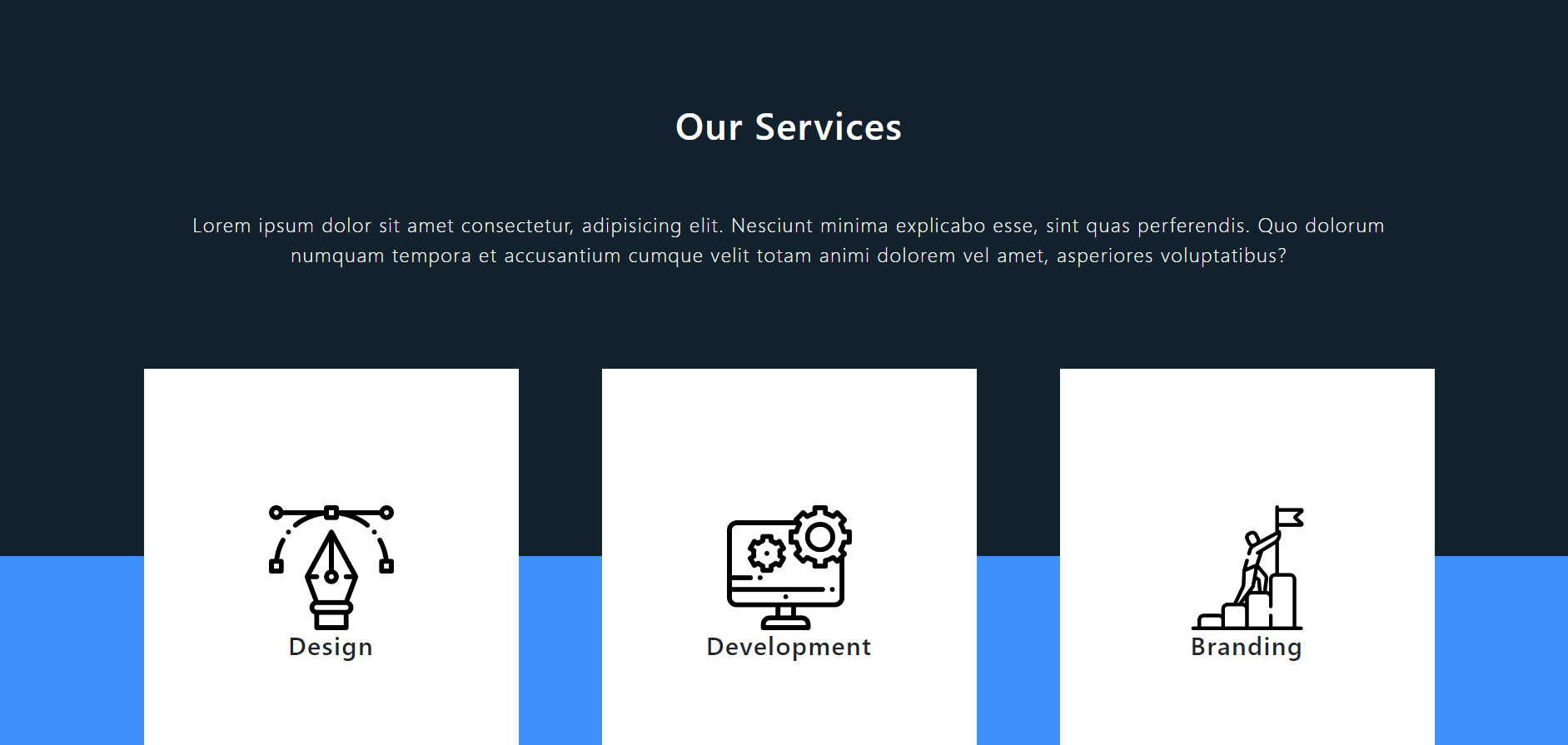
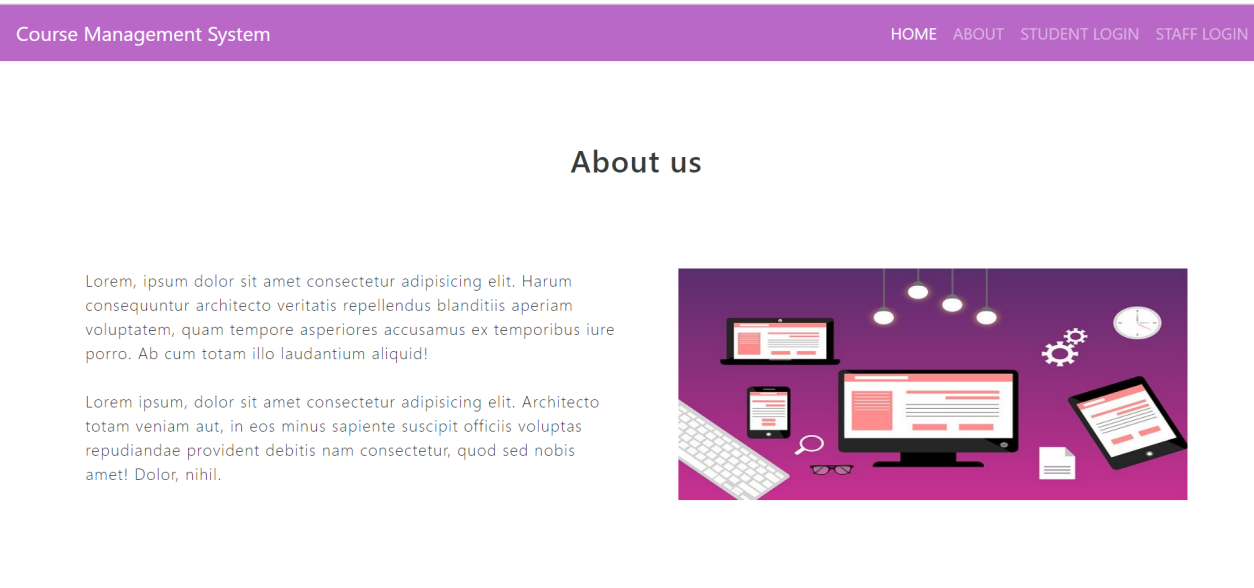


Figure 93 about us page (final display)

# 6 Test Strategy:

## 6.1 Overview of Test Strategy:

In act of software development, proper methodology of software development should be followed. Proper implementation of functionality by using best methodology can result in sustainable system. To enhance concept sustainable required system, we followed Agile development methodology in our software development process. Agile Methodology helped us to filter the system requirement for each management system. Working with smaller segments of system, designing, developing, testing and implementing each segment to whole project was only possible due to this methodology. As part of agile methodology there are different testing strategies so, we performed manual and black box testing. The testing is concerned input and required output. Here we conducting the testing strategy for both desktop and mobile browsers. All test was performed to obtain desired output.

We manually entered input and the desired output was obtained as shown in the list. Here we performed test for admin , student, etc. Different login system was managed on the basis of role. At first, Course was successfully added, added course were successfully displayed in the list, We can update and delete the course from the list. After successful crud operation in course, Module was successfully added in specific course. We can view the list of modules in that selected course. We can update and delete the desired modules from the list. Module update was performed successfully along with delete operation. Similarly, Students were enrolled in the course. Student were added in the selected course. After successfully added, student enrolled in the courses were visible. Student can be updated successfully and deleted successfully. Added student can get into login into student information portal. Students can view courses that they are enrolled in. The modules inside that courses is displayed along with the assignment of that module. Student can change their password through change password option in student portal.

Similarly, admin can add staffs in the modules. The details of staffs can be update and delete. Staffs are enrolled in modules. Added staff can be logged in via staff portal. Admin can add assignment to given module. Addition, deletion, updation and assign of assignment were successfully done as given in table below.

Different assignment can be added, viewed, delete and update in the given module.

Password encryption for each user password were successfully done or tested. Admin cannot get logged in student or staff portal. Likewise, student or staff cannot access admin dashboard. The system works on both mobile and desktop. The system ensures of proper security of system by ( password encryption, authorize access). Further details on testing are presented in below table.

## 6.2 Sample Test Results:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Test** | **Test Steps** | **Expected Result** | | **Result**  **(Desktop)** | **Result**  **(Mobile)** |
| 1 | Admin login to dashboard | - Navigate to login page  - Enter correct username and password | The valid user enters to the dashboard | | Same as expected | Same as expected |
| 2 | Incorrect login | - Move to login page  - Enter incorrect credentials | Message showing invalid username and password | | Same as expected | Same as expected |
| **For admin user only(After successfully login)** | | | | | | |
| 3 | Add new course | Navigate to -> add courses  - Fill the details correctly. | | Successfully added message is shown | Same as expected | Same as expected |
| 4 | Show courses | Navigate to -> show courses. | | List of courses appears with action buttons of update and delete | Same as expected | Same as expected |
| 5 | Update course | - Click to update button.  - Fill the updated data correctly | | Successful update of data showing message | Same as expected | Same as expected |
| 6 | Delete Course | - Click to delete button | | Successful deletion of selected course details | Same as expected | Same as expected |
| 7 | Show Course details | - Click to the name of courses  Highlighted  by orange color | | Details of selected course appears with show modules button | Same as expected | Same as expected |
| 8 | Show modules | - Navigate to show modules  through  show module button in course details page | | Table showing list of modules with add module button at top. | Same as expected | Same as expected |
| 9 | Add module | - Click on the add module button.  - fill the details correctly | | Successfully added, then redirected to show modules page | Same as expected | Same as expected |
| 10 | Update module | - Click on the action button in show module page named as update. | | Update form for module appears.  - fill the details correctly. | Same as expected | Same as expected |
| 11 | Delete module | - Click delete button | | - Module records get successfully deleted | Same as expected | Same as expected |
| 12 | Show Students | - Navigate to student link through slider | | List of student enrolled in the course appears | Same as expected | Same as expected |
| 13 | Add student | - Navigate to the add student button appeared at top left side of the card.  - Save student after entering  correct details and selecting course via drop-down | | Add student form appears.  After successfully adding student, successfully added message appears | Same as expected | Same as expected |
| 14 | Update student | - Click on the update button and fill the correct data in the form. | | Student records gets successfully updated | Same as expected | Same as expected |
| 15 | Delete Student | - Delete student by using delete button in action field | | Student gets successfully deleted | Same as expected | Same as expected |
| 16 | Show staffs | - Navigate by using staff link in slider | | List of staff in tabular form with add staff button in the top left side of page. | Same as expected | Same as expected |
| 17 | Add staff | - Go to add staff by using  add staff button.  - fill the details correctly and select module for staff | | Add staff form appears.  Valid data gets successfully added in database, user get redirected to show staff page. | Same as expected | Same as expected |
| 18 | Update Staff | - Update staff via update button in staff list.  - Fill new value correctly and save | | Staff records get successfully updated | Same as expected | Same as expected |
| 19 | Delete Staff | - Delete staff via use of delete button in action column | | Staff records get successfully deleted. | Same as expected | Same as expected |
| 20 | Show assignment | - Navigate to show assignment through  -> assignment link in the slidebar. | | - Assignment list of modules appears.  - Add assignment at top and delete and update button in action field. | Same as expected | Same as expected |
| 21 | Add assignment | - Click to add assignment button.  - Fill the correct data in form by selecting module. | | Form to add student appears, Valid data gets successfully added | Same as expected | Same as expected |
| 22 | Update assignment | - Update assignment details through update button in action bar  - Fill data to update. | | Update form for assignment appears. User entered data get updated | Same as expected | Same as expected |
| 23 | Delete assignment | - Click delete assignment | | Selected assignment get successfully deleted | Same as expected | Same as expected |
| 24 | Logout | - Click on the logout link | | User gets successfully logout | Same as expected | Same as expected |
| **The test below is related to student information portal** | | | | | | |
| 25 | Student login to dashboard | - Navigate to student login by ->  Student login link in university website.  - Enter details correctly | | Valid users gets into dashboard | Same as expected | Same as expected |
| 26 | Incorrect login | - Enter incorrect credentials | | Incorrect username password message appears | Same as expected | As expected |
| After successfully login | | | | | | |
| 27 | View Courses | - Navigate to -> courses link in the sidebar | | List of course that student is enrolled appears | Same as expected | Same as expected |
| 28 | View  Modules | - Navigate to modules page by clicking the course name which is highlighted in orange colour. | | List of modules in that courses appears. | Same as expected | Same as expected |
| 29 | View module details | - Click to any module to see its details | | The details of selected modules appears with show assignment button. | Same as expected | As expected |
| 30 | Show assignment | - Navigate to assignment list by clicking on the specific module name | | Assignment in that modules is displayed | Same as expected | Same as expected |
| 31 | Change Password | - Navigate -> change password.  - Enter old password correctly and add new password | | Change password form appears.  - Password get successfully changed if old password gets matched and redirects to student dashboard | Same as expected | Same as expected |
| 32 | Incorrect old password | - Enter Incorrect old password | | New password doesnot get updated due to incorrect old password | Same as expected | Same as expected |

# 7 System Evaluation

This section is for the evaluation of the system. A set of tasks will be assigned to a certain target audience, followed by a survey. The findings of that survey should indicate whether the system is ready for release.

Measurement of the final system against its initial performance goals, as well as continuous testing to ensure that the system continues to satisfy those goals, are all part of system evaluation.

## 7.1 Usability Evaluation Strategy

The goal of usability testing is to see how well people can understand and use a product to achieve their objectives. It also refers to the degree to which people are satisfied with the procedure. Practitioners collect this data using a variety of approaches, including surveying users about an existing site or planning a new one.

This section documents the review or the trial of the pilot system.

### 7.1.1 Target Audience Selection/Sample Size Selection

The target audience for the course management system are the one having the some of the technical background and the people who have some computer literacy. This system is made on the basis of different roles like corporate, student, staff and for the admin.

On the group of four people the, each performing the same task will be conducted where the two members will be focusing on TA technique and the other two will be doing PEW. The advantage of using both strategies is that the disadvantages of one are usually offset by the benefits of the other.

The result of this technique will be used for the improvising and changing the necessary part in the system.

### 7.1.2 Trail Active Task List Creation/Design

The system that has been built is to be checked to maintain its uniformity and to compare it with the result, the two method TA and PEW has been implemented on same set of tasks. This is generally done to know how the system is carried out by the users either in the easy way or uneasy and the assessment of how well the system complies with usability requirements in general.

Experimental instruction,

THINK ALOUD – Experimental Instructions for Candidates

Its really great that you took this test which will help us in improving the quality of the system software that we are developing by finding the exiting problem or bugs in the current system.

We want you to think aloud which means express everything that you are thinking while testing the software. Everything will be recorded while you are testing the software and used for the further development of the software by fixing if there are issue.

Thank You!

Best Regard,

Ritesh Koirala

|  |  |  |
| --- | --- | --- |
| Task | Feature Tested | Instruction |
| Login Admin | Login Algorithm | Enter [admin@gmail.com](mailto:admin@gmail.com) as a username.  Enter admin as a password.  Click login to login in the system. |
| Open/edit Course record | View record/  Update record | Click on the hamburger icon on right side.  Then, click on course.  Then, select update button on the left side to update the record. |
| Add Course | Add records | Click on the hamburger icon on right side.  Then, click on add course.  Fill up all the required field.  Click save and save.  Refresh the page.  Then you will be able to see the added record there |
| Delete course | Delete record | Go on the course again,  Then, select delete button on the left side to delete the record. |
| Open/edit module record | View record/  Update record | Click on the orange highlighted email in course.  Then, select update button on the left side to update the record. |
| Add module | Add records | While viewing module, on right side above title there is add course button, click there.  Fill up all the required field.  Click save and save.  Refresh the page and go to module.  Then you will be able to see the added record there. |
| Delete module | Delete record | Then, in module select delete button on the left side to delete the record. |
| Open/edit Student record | View record/  Update record | Click on the hamburger icon on right side.  Then, click on Student.  Then, select update button on the left side to update the record. |
| Add Student | Add records | While viewing student, on right side above title there is add student button, click there.  Fill up all the required field.  Click save and save.  Refresh the page.  Then you will be able to see the added record there. |
| Delete student | Delete record | Then, in student select delete button on the left side to delete the record. |
| Open/edit Staff record | View record/  Update record | Click on the hamburger icon on right side.  Then, click on staff.  Then, select update button on the left side to update the record. |
| Add staff | Add records | While viewing module, on right side above title there is add staff button, click there.  Fill up all the required field.  Click save and save.  Refresh the page and go to staff.  Then you will be able to see the added record there. |
| Delete staff | Delete record | Then, in staff select delete button on the left side to delete the record. |
| Open/edit assignment record | View record/  Update record | Click on the hamburger icon on right side.  Then, click on assignment.  Then, select update button on the left side to update the record. |
| Add assignment | Add records | While viewing staff, on right side above title there is add assignment button, click there.  Fill up all the required field.  Click save and save.  Refresh the page and go to assignment.  Then you will be able to see the added record there. |
| Delete assignment | Delete record | Then, in assignment select delete button on the left side to delete the record. |

## 7.2 Initial System Pilot Trials/Results

#### 7.2.3.1 Thinking Aloud (Observations) and Results

A direct observation technique for user testing that entails getting participants to speak out while carrying out a task. Users are asked to describe what they are doing, thinking, and feeling at any given time. This approach is particularly useful for figuring out what consumers expect from a system and pinpointing its confusing features.

However, the system was created to be learned so that records management would benefit in the long run, thus an early reaction of this kind is not concerning unless it is persistent and lasts for a long time.

Since the user was unable to complete all the task and she found that the layout was really fast which proves the consistency of the software is good.

The feedback was taken in the stressed way and the software will be updated as the response received.

And, the software system fault which were discovered by the TA technique is listed below in section Amendments part.

#### 7.2.3.2 Post Event Walk-through (Questions) and Results

The PEW method demonstrated that the users' overall experience was positive. They acknowledged that they initially felt overwhelmed as well, but they also commented on how quickly and easily they became familiar to the surroundings.

The user stated their response that the overall system was good to use Due to the consistent usage of relevant icons throughout, the design is pleasant to the eye, consistent in layout, and very simple.

#### 7.2.3.3 List of Key System Amendments

As, the good response there were some of the errors or the problem which has been sated by the user while applying the techniques and they are given below in the table.

|  |  |
| --- | --- |
| Topic | Recommendation |
| Login | Make the login button work by pressing login. |
| Records | * webpage I am in should have been focused. * While creating the record there should be shown that which one is most required to fill the box by using \* sign. * There should be pop up while creating the records, updating and deleting record whether they surely want to do it or not. * Making some drop down button on some field will be better. |
| Refresh the screen | After successful process the content should be refreshed automatically. |

## 7.3 Final System Trial/Results

### 7.3.1 Usability Questionnaire Design

For this Questionnaire process a group of 10 people having basic knowledge of computer are taken and let them give the feedback on the course management system in the term of design, easiness and usability standard.

Questionnaire:

1. Whether the system was easy to use.
2. Whether it is right thing to use.
3. Whether it is straight forward to us.
4. Whether it is safe.
5. Whether it is worthy of spending time.
6. Whether it is good for using in study.
7. Whether the user feel like having the full control of it.
8. Whether it needs guideline.
9. Whether it needs instruction.

10.whether is it better than paper-based system.

11. whether the system has good response on it.

12. whether this system is preferable or not.

### 7.3.2 Final Questionnaire (Used in Trials)

CMS Client Evaluation Trial Questionnaire

We appreciate you taking the time to respond to this online survey, whose purpose is to get your thoughts on the course management system and what we can do to make it better.

There is some question which will take like 2 minutes to complete and we will keep your review in best regard and it will be kept within us. Please tick in the place which you think is applicable for the questionnaire presented there.

1. Graphics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Questionnaire | Strongly Agree | Agree | Disagree | Strongly Disagree |
| The website appears to be well-made. |  |  |  |  |
| The website is easy to understand for the use. |  |  |  |  |
| The logo and color are well presented. |  |  |  |  |

1. Navigation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Questionnaire | Strongly Agree | Agree | Disagree | Strongly Disagree |
| The navigation bar is easy to find, responsive and make a clear sense of understanding. |  |  |  |  |
| I found what was searching easily. |  |  |  |  |

1. Features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Questionnaire | Strongly Agree | Agree | Disagree | Strongly Disagree | Did Not Use |
| Do you find everything was straight forward? |  |  |  |  |  |
| There was no issue in the place given to complete the task. |  |  |  |  |  |

1. Task work

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Questionnaire | Strongly Agree | Agree | Disagree | Strongly Disagree | Did Not Use |
| Has easily been able to create, edit and delete every record. |  |  |  |  |  |
| Was easy and made simple to work. |  |  |  |  |  |
| Do you find something like you were not in control of some activity that you were performing? |  |  |  |  |  |

1. Any additional thing that was in test but we did not mention in above questionnaire?

|  |  |  |
| --- | --- | --- |
|  |  |  |

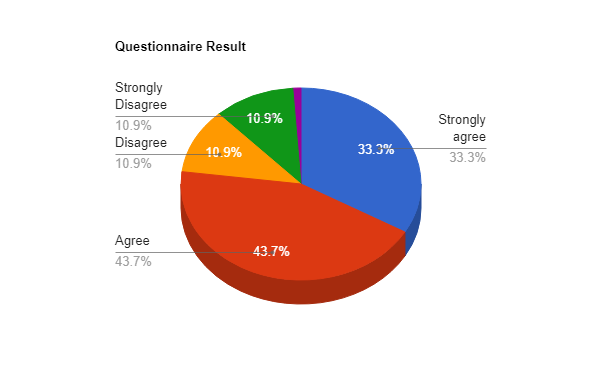
I appreciate your time. We sincerely value your opinions and will work to incorporate them into improving the course administration system for the university.

### 7.3.3 Questionnaire Results – Implications of Results

The result obtained from above questionnaire is:

|  |  |
| --- | --- |
| Marked | value |
| Strongly Agree | 33.3% |
| Agree | 43.7 |
| Disagree | 10.9% |
| Strongly Disagree | 10.9% |
| Did Not Use | 1.1% |

Graphical representation of result



According to the result we received from the questionnaire we have checked in the system we have built and made the necessary change needed to improve the course management system according to the response from users.

# CONCLUSION

The summarize of the client-driven system solution for the University of woodland is now have been ready and ready for being presenting (to the client) as the assignment of the university. This project took a total of 9 week to complete. The first week was comprised by finding and taking of the interview and then it was followed by the documentation of the requirement specification, system analysis and design and so on. the plan on how much implementation to be done was made beforehand and on the basis of that we went forward and the build note, testing and evaluation of the system that was built has been done according to that.

The web application is built on the basis of the information that we got from the interview. The university of woodland has 120 different courses and more than 5000 students enrolled in different course. So, the system built should go along with every student and have the smooth functionality so, there should not be any problem while using it. The capacity of the website is like more than 10000 at a time but their can be some sort of problem like slow response poor functionality which might be cause by high number of people are doing some heavy activity like upload download at the same time which might even cause some malfunction in system software resulting in crashing the system. But even so somewhat like 1000 people doing heavy activity can be handled by the website efficiently.

The project was done in the group of 4 people dividing the project in equal part and making it worth by giving the hard time and dedication. After the completion of the documentation part, we started the implementation of the documentation into the development of the web application. We have used the agile methodology to complete this project. The black box testing was done to test the system whether it was working fine or not and for that some of the steps were taken which are listed in above testing documentation and the output of the testing was as expected everything was working fine and while evaluating the result was algo good but there were some fault which has been again rebuilt after the response that we got from the tester so, we can say that our project is 80% - 90% good for working which is based on the evaluation and testing of the system software.

Since, we have got the 12-module total to complete in the university of woodland but due to the limited time we were only able to complete 5 modules so if there was more time than the first thing, we would have covered is the remaining modules. There are some of the modern concepts which are like very fascinating to keep in the website which are as follow:

1. Virtual tour of the college.
2. Health related survey every week.
3. New organized event detail as a notification.
4. Notice box.
5. University newspaper.
6. Knowledgeable games like quiz, sudoku, etc.
7. Weekly talent hunt page for writing.
8. Online library.
9. Online student support.

These are some features that are usually seen in many universities website that helps in the growth of the student and make the website so reliable that the student will get used to it and work as an advertisement item for the growth of university.

So, the agile methodology was used to organize and develop the project. The project went smoothly and organized as plan. The website of the university of woodland is working and functioning well and the documentation is completely done and presented and this conclude the client driven system software that was proposed by the university of woodland.

# 9 Project Management

## 9.1 Project Gantt chart:



## 9.2 Project Meeting Minutes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Meeting | Date | Time | Location | Duration |
| Meeting 1 | May 6 | 8:30 AM | Workshop 205 | 1 hour |
| Meeting 2 | May 16 | 8:00 AM | Workshop 203 | 1 hour |
| Meeting 3 | May 26 | 8:30 AM | Workshop 205 | 2 hour |
| Meeting 3 | June 8 | 9:00 AM | Workshop 205 | 2 hour |

## 9.3 Project Quality Plan/Strategy:

This section includes the main plan and strategy for accomplishment of project. The plan and strategy are dependent upon the aims and objective of this system build-up.

The plans and strategy to fulfill aims and objective of this project are:

- Effective requirement gathering by knowing current systems functionalities.

- Knowing the expectation of key stakeholders.

- Gathering disabilities of current system.

- Comparing with comparable system solutions.

- Following effective methodology of system development process.

- Proper engagement of group members for successful project.

- Knowing the accessibility and usability of the system.

- Building system by focusing on system requirements along with fulfilling stakeholders needs.

- Building system under allocated budget.

## 9.4 Project Work log:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Group 2** | **Aayush Shrestha** | **Pragyan Kumar Chamlagain** | **Ritesh Koirala** | **Sungava Subedi** |
| **May 6-12, 2022** | 2.1.2.1 Interview - Simon White , Setting up Codding environment, Frontend of Login, In progress of Implementation of Course management system | 1. Introduction, 2 Requirements Engineering -> 2.1.1 Interview plans, 2.1.2 Interview findings: 2.1.2.2 Adam Blake,2.1.3 Other problem domain research 2.1.3.1 Comparable Software System Review 2.1.3.1.1 Records Management Systems 2.1.3.1.2 Student Records/Information Portal 2.1.3.1.3 Student Records/Information Applications (Mobile Format) 2.1.3.2 Development Relevant Legislation | 2.1.2 Interview findings: 2.1.2.4 Mark Williams, 3.1 Preliminary Design Stages,3.4 System Database Design 3.4.1 E-R Model 3.4.2 Attribute Listings | Interview Finding: 2.1.2.3 Raj patel, 2.2.2 Functional Requirements: 2.2.2.1.1 Record Management Systems, 2.1.3.1.2 Student Records/Information Portal 2.1.3.1.3 Woodlands University College Corporate Website,4 System Interface Designs -> 4.1 Draft Interface Designs (Records Management) |
| **May 13-19, 2022** | Testing of Course management system, Implementation of Course management system, module management system in progress. | 2.1.3.3 Academic Literature Review 2.1.3.4 User Group Questionnaires,2.2 Requirements Specification 2.2.1 Problem Domain Description 2.2.1.1 Existing Business Operation 2.2.1.1.1 Student Life-Cycle 2.2.1.1.2 Personal Tutorial Life-Cycle 2,2.1.1.3 Optional Process 1 2.2.1.1.4 Optional Process 2 | 3.3 Detailed Dynamic System Designs 3.3.1 Events Charts 3.3.2 Object Creation Charts 3.3.3 System Scenario Charts 3.3.4 Dynamic Diagrams | 2.2.3 Performance Requirements 2.2.3.1 Records Management 2.2.3.1.1 Speed 2.2.3.1.2 Capacity 2.2.3.1.3 Reliability 2.2.3.1.4 Usability 2.2.3.1.5 Accessibility 2.2.3.2 Student Records/Information Portal 2.2.3.2.1 Speed 2.2.3.2.2 Capacity 2.2.3.2.3 Reliability 2.2.3.2.4 Usability 2.2.3.2.5 Accessibility |
| **May 20-26** | Coding environment setup done (Spring boot, Java, mysql). Documentation: Introduction, Aims & Objectives, Requirement Elicitation in progress, SRS in progress. |  |  |  |
| **May 27-June 2** | Admin login,Implementation of course, modules management system in the system. designing home, about page for university website, Functional requirement documentation, System analysis and design in progress, Testing Module management system, SRS in progress | 2.2.1.2 Summary of existing business limitations requiring resolution and existing strengths. 4.4 Draft Interface Designs (Student Records/Information Portal) Student record management system in progress | 3.2 Detailed Static System Designs,Testing and Implementation of assignment management system | 2.2.3.3 Woodlands University College Corporate Website 2.2.3.3.1 Speed 2.2.3.3.2 Capacity 2.2.3.3.3 Reliability 2.2.3.3.4 Usability 2.2.3.3.5 Accessibility, 2.2.4 Design Constraints 2.2.4.1 Records Management Systems 2.2.4.2 Student Records/Information Portal 2.2.4.3 Woodlands University College Corporate Website, Implementation of staff management system in progress |
| **June 3-10** | Implementation of student, staff , assignment record management system in the system. Student login for student information portal. Staff login for staff portal. Implementation of student and staff portal in the system Frontend in process for the system. System interface and design. 5 System Build and Technical Notes, 6 Test Strategy. SRS is in progress | Implementation of student record management system ,4.7 Draft Interface Designs (Woodlands University College Corporate Website) 4.7.1 Wireframes 4.7.2 System Navigation Diagram 4.7.3 System Screen mock-ups 4.7.4 System Activity Event Diagrams 4.8 Design Revisions (Woodlands University College Corporate Website) 4.9 Heuristic Evaluation (Woodlands University College Corporate Website) | 7 System Evaluation, 7.1 Usabilitty Evaluation Strategy, Initial System Pilot Trials/Results, 7.3 Final System Trial/ Results, 8 Project conclusions, 9 Project Management | Implementation of Staff management system, 2.2.4 Design Constraints 2.2.4.1 Records Management Systems 2.2.4.2 Student Records/Information Portal 2.2.4.3 Woodlands University College Corporate Website 2.2.5 Commercial Constraints (Total Project) 2.2.6 Acceptance Tests (Total Project) |

# 10 References

*Reference: <https://www.lawcommission.gov.np/en/wp-content/uploads/2019/07/The-Act-Relating-to-Compulsory-and-Free-Education-2075-2018.pdf> Accessed Date: 2022/05/16*

***Bootstrap css:***

*Reference: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> Accessed Date: 2022/05/26*

***Data Protection Act***

*Reference: https://www.gov.uk/data-protection Accessed Date: 2022/05/14*