PROJECT REPORT

Automated Class Routine and Exam Schedule Management System(ACRESMS)

SHABBIR AHAMMED

ROLL: ASH1711ME121M

SESSION: 2016-2017

MASTERS OF SCIENCE (INFORMATION AND COMMUNICATION ENGINEERING)

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

DECEMBER 2020

SUPERVISOR'S DECLARATION

I hereby declared that I've checked this thesis/project in my opinion, this thesis/project is adequate in terms of scope and quality for award of the degree of Masters of Science in Information and Communication Engineering.

Professor Dr. Md. Ashikur Rahman Khan

Chairman,

Dept. of Information and Communication Engineering

Noakhali Science & Technology University

Sonapur – 3814, Noakhali, Bangladesh

STUDENT'S DECLARATION

This project is submitted to the dept. of Information and Communication Engineering of Noakhali Science & Technology University in partial fulfillment of the requirements for having the Masters of Engineering in Information and Communication Engineering (ICE). So, I hereby declare that this project report has not been submitted elsewhere for the requirement of any kind of degree, diploma or publication.

Shabbir Ahammed

Dept. of ICE, NSTU

Class Roll No.: ASH1711ME121M

Session: 2016-17

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

I am very grateful to my project guide Professor Dr. Mohammad Ashikur Rahman Khan for the guidance, inspiration and constructive suggestions that were very helpful for me in the preparation of this project.

ABSTRACT

Every educational institution needs class routine and exam schedule for its activities. If making this class routine and exam schedule will generate automatically, it will be very amazing. Automated Class Routine and Exam Schedule Management System(ACRESMS) is very important in any educational institutions. The authority can create class routine and exam schedule by clicking this system, generate class routine, update routine, generate exam schedule, update exam schedule and deleted it easily. This system saves time and reduce paper based work. We will divide the system into different segments such as teacher's information, course information, classroom details, teacher details, time management, session, years, teacher summery etc. Every segments have contained different database for clear concept. We use here MySQL database to implement the system. The purpose of the project is to automated making ACRESMS which is user friendly, simple and fast. This system will be helpful for any institutions because there are few paper based task, student can get class schedules and exam schedule easily and admin can handle this system without any problem because it will be online based system. This system reduces complexity of class room, teacher availability problem, examination hall room, invigilator availability and paper based complexity. This system will be user friendly system.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Every educational institution in Bangladesh has schedules for the management of classes and exam for allocation of time. Currently these tasks are manually created by a particular department in the institution. The parameters like teachers', room, time, number of periods, class time, exam time etc. for creating class and exam schedule are also manually kept in mind while creating these tasks. Also the schedule keeps changing with time. Hence making their management even more difficult due to change in parameters value/state. If the above harsh process of creating and maintaining the schedule could be done using an automated system, then it would help the department of an institution to loosen the burden and increase the efficiency. Due to the absence of such automated system for creating and maintaining the class routines and exam schedule, Automated Class Routine and Exam Schedule Management System(ACRESMS) is required to be implemented. In ACRESMS, we consider a sequential and organized database. Database contains very well decorated data that is inserted by the user. It is very user friendly, user can search data from database and see very quickly.

1.2 PROBLEM STATEMENT

Most educational institutions manually create their schedules (both class and exam) by using a human resource to work on it. Some institutions create such management system for their personal requirement and use only while others do it manually. Unfortunately, there is no such generic system at all for creating and maintaining schedules (both class and exam) which can be used by all the educational institutions. Automated Class Routine and Exam Schedule Management System(ACRESMS) will help educational institutions to create and maintain the class routines and exam schedules in a more effective and efficient way. In this system, user can see the teachers taking which courses in a week and how much credit he taken. It is very helpful for a teacher or faculty member.

1.3 OBJECTIVES

Objectives of the project are given below

i. To generate class routine and exam schedules in an efficient way
ii. To enhance high accuracy level. All operation would be done correctly.
iii. To make highly reliable system.
iv. To provide easiest way to operate

1.4 SCOPE of WORK

Automated Routine and Exam Schedule Management System(ACRESMS) benefits all the educational institutions that currently uses manual creation and maintenance of the class routines and exam schedule. Users can easily create and maintain class routines and exam schedules using the system's simple user interface. It also resolves the issue of manual staffing process into a controlled and closely monitored system. It can be used by every educational institution.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Literature review is very important because it provides foundation of knowledge on topic. It Identify the inconstancies: gaps in research, conflicts in previous studies, open questions left from other research and need for additional research (justifying current research).

2.2 RELATED WORK

There are some of system already implemented like this system and has lot of limitation. Some of these given below

Class Routine and Examination timetabling is a well-researched and important educational timetabling problem [1]. It has been studied many times in other institutions since the mid-1960s. Although there is some novel research that focused on the modeling

and formulation of Examination Timetabling Problem (ETP), a large number of researchers are interested in the applications of heuristics to solve the problem

A related work by investigated a new graph coloring constructive hyper-heuristic for solving examination timetabling problems using hierarchical hybridizations of four low level graph coloring heuristics [2]. The solution lists all examinations by the level of difficulty of assignment and ensures the most difficult exam to be scheduled is scheduled first. This method was considered simple and yet efficient approach but suffers setbacks in handling higher level conflicts in examination scheduling. Graph coloring has the limitation which neglects hard constraints such as students cannot take two examinations consecutively by formulating the basic examination timetabling problem as a Quadratic Assignment Problem. Aldy model assumes that the number of examinations is equal to the number of time periods and a hybrid algorithm based on a combination of GRASP (The Greedy Randomized Adaptive Search Procedure) and SA-TS (Simulated Annealing and Tabu Search) was introduced to get an optimal solution. This solution appears to be an efficient heuristic algorithm for solving examination scheduling problem for its ability to obtain the optimal or the best known solutions within reasonable computation time. One of the major limitations of this solution is its inability to handle a very large size problem and has not really been applied to real world problem.

In 2001, Dimopoulou and Miliotis tackled examination scheduling problem such as restricted availability of classrooms and the increased flexibility of the student's choices of courses by implementing an integer programming (IP) model that assigns courses to time slots and rooms. The quality of the schedule produced depends on the relative position of the courses assigned to the available time periods, a condition that the IP model attempts to satisfy by constructing groups of courses that are assigned to groups of time periods. The system was adopted by Athens University of Economics and Business for use and with success. The approach is said to be theoretical development and do not analyze an actual computer-based implementation. Sagir and Ozturk, 2010 decided to address the issue of assigning invigilators to examination who may be faculty lecturers or graduate students in educational institutions with a computer based system. They adopted Analytic Network Process model for estimating exam and objective weights. Recently,

developed a new binary model for examination time tabling problem, the model was presented in two directional novelties [3]. The first one is that a course can be offered more than once in a semester. If a course is requested by a few students, then it is enough to be offered once. If the number of students requesting a course is more than the maximum number of students who are allowed to attend a single class, then the course is multi-offered. The second novelty is that sharing a room for two simultaneous exams is allowed. Also, the model considers some hard and soft constraints, and the objective function is set in such a way that soft constraints are satisfied as much as possible. Due to wide assortment of applications, The QAP has been extensively used to formulate other practical problems such as economic problems, Scheduling problems, Timetabling problems, Facility layout problems.

2.3 SUMMARY

From these work that are they work, we found some limitation. At that time, they didn't make web based system and didn't use database language. So it is very difficult to making class routine and exam routine easily or automatically. Our proposed system solves the previous system complexity. Automated Routine and Exam Schedule Management System will be helpful for both teachers and students. It reduces the paper based complexity. So it will be attractive for teachers specially who generate class or exam schedule for students.

CHAPTER 3

METHODOLGY

3.1 INTRODUCTION

Research methodology is very essential for a researcher. This part contains the overall procedure of the research such as how can solve the problem in an efficient way? How can search data from database? How can design database table in a proper way? What is the activity of data? Which software model needed? etc. Research Methodology provides with a comprehensive overview of a broad range of research paradigms and methodologies, with ontological and epistemological underpinnings, as well as associated methods and techniques, in order to inform the design of methodologically sound research proposals and to develop interdisciplinary methodological literacy as future researchers.

The System includes:

- Generate class routine.
- Generate summary of the courses of the individual's teacher
- Print class routine
- Generate exam routine
- Print exam routine

3.2 SOFTWARE DEVELOPMENT MODEL

In this system, we use v-model for software development process. Every time we do some task and test the task that is working properly or not.
The various phases of the V-model are as follows [4]:

3.2.1 V-Model – Design

Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.

Figure 3.1 shows the overall mechanism of this model.

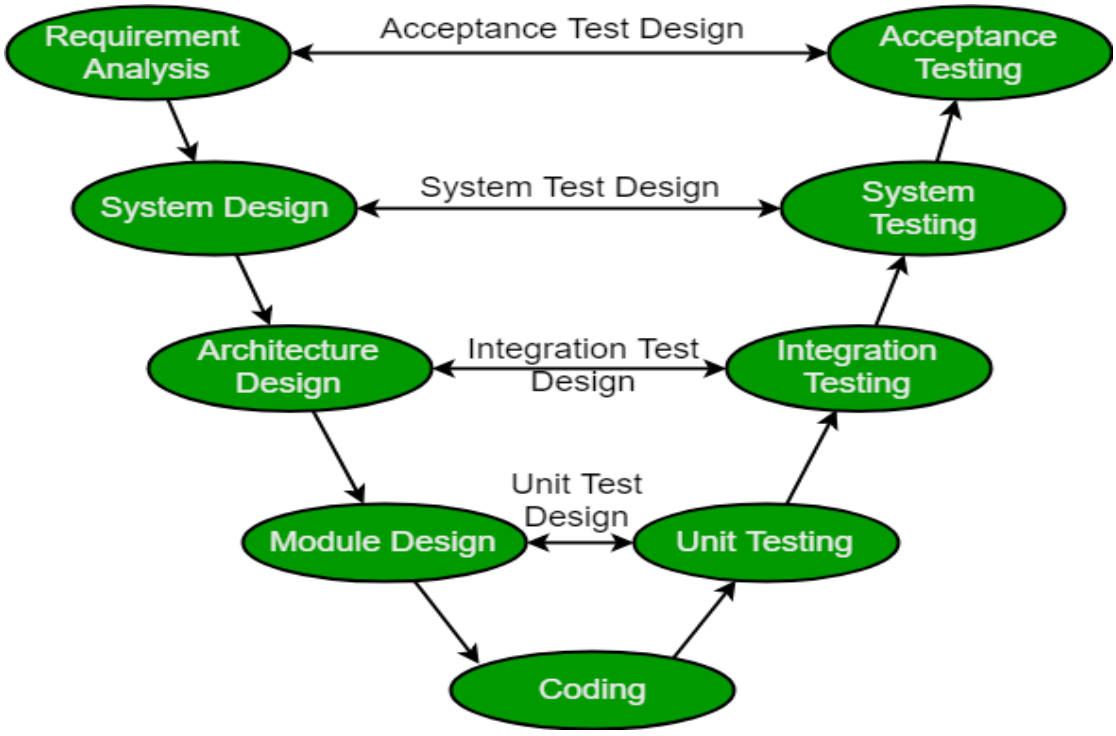


Figure 3.1: V-model

3.2.2 V-Model - Verification Phases

There are several Verification phases in the V-Model, each of these are explained in detail below.

Business Requirement Analysis

This is the first phase in the development cycle where the product requirements are understood from the customer's perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.

System Design

Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.

Architectural Design

Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the

final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD).

The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.

Module Design

In this phase, the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements.

The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

3.2.3 Validation Phases

The different Validation Phases in a V-Model are explained in detail below.

Unit Testing

Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

Integration Testing

Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

System Testing

System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during this system test execution.

Acceptance Testing

Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

The major drawback of the V shape model is difficult to add or change requirements later in the development process and V shape model is not appropriate for complex project.

3.3 TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

One of the purposes of the testing is to validate and verify the system [5][6].

Verification: It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet [4].

Validation: It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements [4].

3.3.1 Levels of testing

There are many techniques for testing any system. Here we mention four levels of testing procedure. These are unit testing, verification and validation, integration testing and system testing.

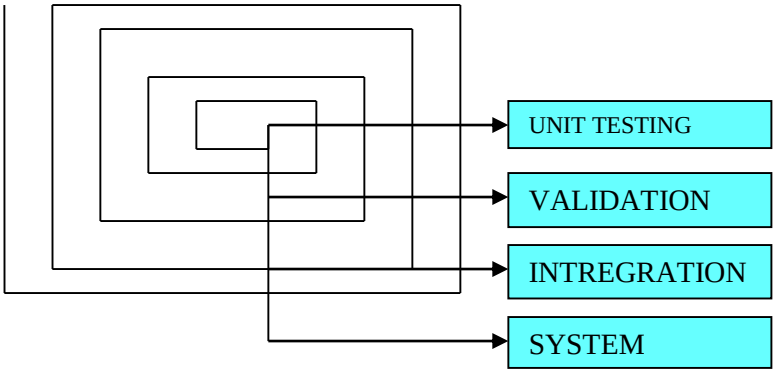


Figure 3.2: Levels of testing

Unit testing

This is the first testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

Verification and validation

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

Integration testing

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

System testing

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

3.4 TECHNOLOGIES

There are two important parts are needed for implementing Automated Class Routine and Exam Schedule Management System(ACRESMS). These are development tools and server requirements. According this tools, we will develop our system easily and accurately.

3.4.1 Development tools

For development tools we need these elements that are given below

- Development environment: Windows, Linux
- Design tools : Adobe Photoshop, Microsoft Paint.
- Front end : HTML5 [7], CSS3 [8], Twitter Bootstrap, JavaScript[9], jQuery etc.
- Back end : PHP[10], MySQL [11]
- Framework : Laravel
- IDE : Notepad++, Sublime Text Editor
- Compatible browsers: : Mozilla 43+, Chrome: 38+, IE 50+

3.4.2 Server requirements

For server site we need these requirements that are given below

- Operating System : Windows, Linux
- Web Server : Apache/NGINX
- PHP Version : Minimum 5.4
- MySQL version : Minimum 5.0
- RAM : Minimum 4 GB
- Storage : Minimum 5 GB
- Bandwidth : Minimum 4 GB/month

3.5 SCREENSHOTS

Here mention some screenshots of the AREMS. These screenshots are taken from Automated Class Routine and Exam Schedule Management System(ACRESM). These are the total overview of the system.

3.5.1 Admin login

Only authentic user can access the system by inserting user name and password

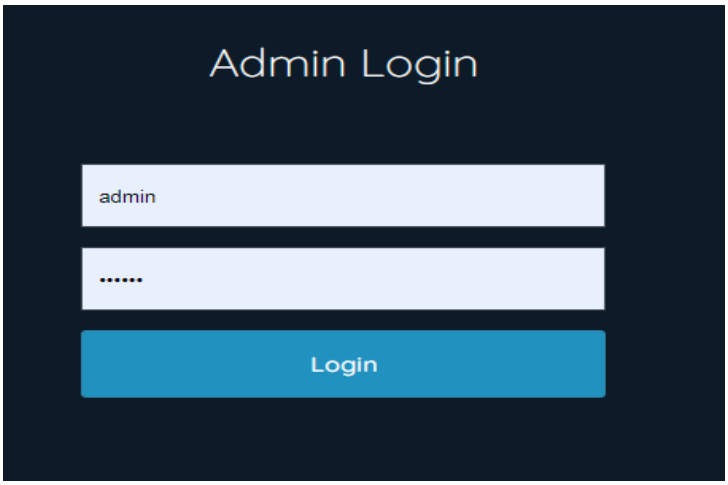


Figure 3.3: Admin Login

3.5.2 Dashboard

After login the system user see this page that is include the total overview of the system. User can see the Class routine, Exam Schedule, Department Info, Session, Class Room Info, Teacher Info, Course Info and Class Schedules menu. User can add, delete, update information from here. Every section has a sub section. User can operate anything that

needed by using these subsections. The sub section includes generate class routine, show routine, teacher summary, print routine, generate exam schedule, show schedule, print schedules, add teacher, update teacher information, add department, update department, add class room, update class room, add session, add schedules etc.

Figure 3.4: Dashboard

3.5.3 Generate Class Routine

From this section user can select day and submit it then he/she can generate routine on a particular day. User can select Sunday, Monday, Tuesday, Wednesday, Thursday and make a schedule for each particular day. For this, user can easily choose a day and update it without any problem.

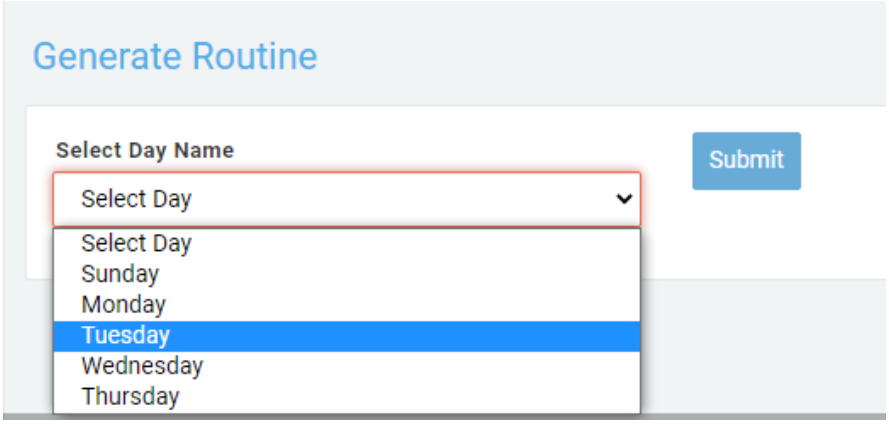


Figure 3.5: Day Selection Form

3.5.4 Generate Routine

In this section, user can add routine for a particular time slot and class room. If user clicking "Add New", user can able to select some tab that are given in figure 3.7. In this figure, here two dimensions: one side for time Schedules and another side is Class Room. After clicking submit button then routine will be generated. If need any change, user will able to update information by clicking edit button.

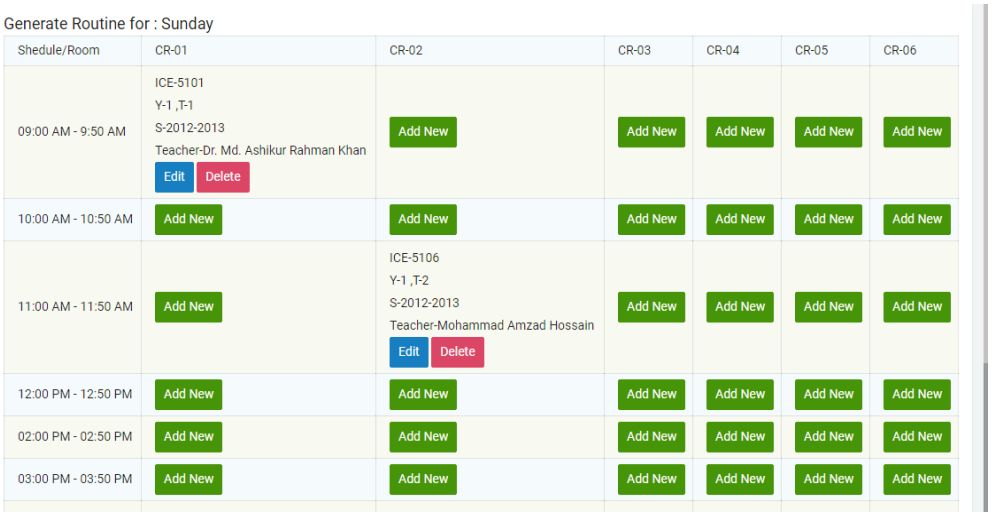


Figure 3.6: Generate routine

When user click Add New option then this page will be generated, after completing the page requirement then routine schedule generated for a fixed time slot and room. Here user select degree level, session, year, term, course, teacher. After selecting that correct information, then user click the submit button. After that routine will be generate for that particular time slot. In this way, user can add all the schedules in his/ her sweet will.

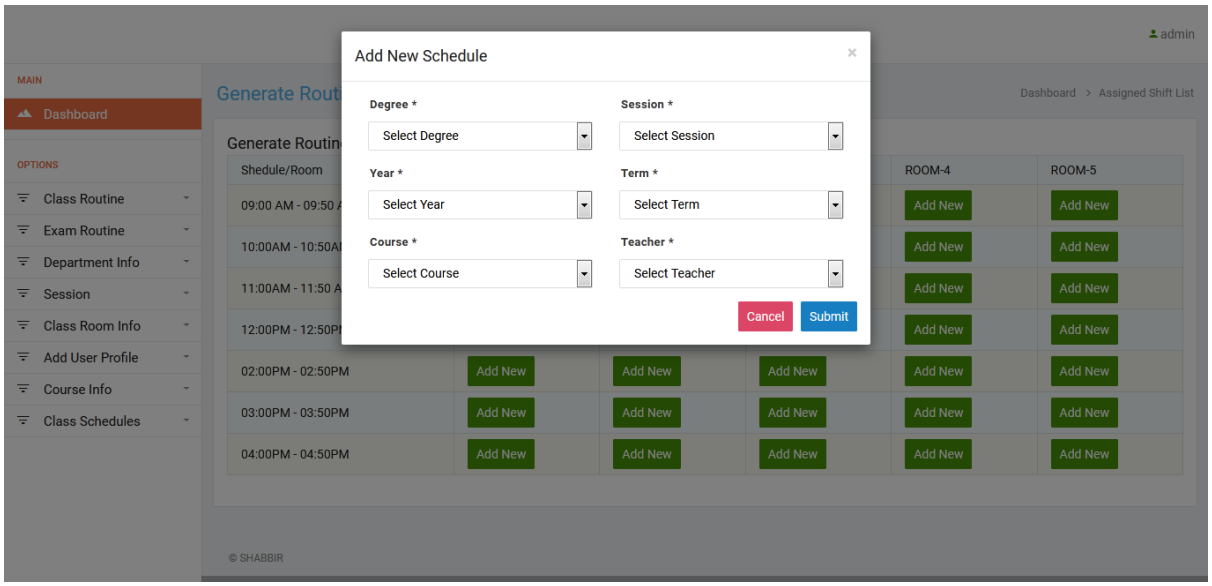


Figure 3.7: Add New operation

3.5.5 Show Class Routine

The final routine is here. Here user can see the total routine that he/she generated later. There are days, degree level, session, year, term, time schedule. Degree level describes honors or masters, day describes Sunday, Monday, Tuesday, Wednesday and Thursday, session describes students are continuing which admission year, year and term describes the students where they are in. Time schedule describes class start and ending time or class duration. The inner cell is continuing information of class. Here is available course code, class room number and teacher short name.

Routine for : Information and Communication Engineering Department

Days	Degree level	Session	Year, Term	09:00 AM - 09:50 AM	10:00AM - 10:50AM	11:00AM - 11:50 AM	12:00PM - 12:50PM	02:00PM - 02:50PM	03:00PM - 03:50PM	04:00PM - 04:50PM
Sunday	Honours	2012-2013	Y-1, T-1	-	-	ICE-1101 (CL-ROOM-2) AR	-	-	-	-
	Masters	2012-2013	Y-1, T-1	ICE-5101 (CL-ROOM-1) MARK	-	-	-	-	-	-
Monday	Honours	2012-2013	Y-1, T-1	-	-	-	-	-	-	-
	Masters	2012-2013	Y-1, T-1	ICE-5104 (CL-ROOM-1) ZUS	-	-	-	-	-	-
Tuesday	Honours	2012-2013	Y-1, T-1	-	-	-	-	-	-	-
	Masters	2012-2013	Y-1, T-1	-	-	ICE-5110 (CL-ROOM-2) ZUS	-	-	-	-
Wednesday	Honours	2012-2013	Y-1, T-1	-	-	-	-	-	-	ICE-1101 (CL-ROOM-5) SJS
	Masters	2012-2013	Y-1, T-1	-	-	-	-	-	-	-
Thursday	Honours	2012-2013	Y-1, T-1	-	HUM-1105 (CL-ROOM-4) AA	-	-	-	-	-
	Masters	2012-2013	Y-1, T-1	ICE-5104 (CL-ROOM-1) MAH	-	-	-	-	-	-

Figure 3.8: Class Routine

3.5.6 Generate Exam Schedule

For using time slot and room, user can make examination schedule. Here needs room, date, time slot and teacher, year, term, course code. By using these data users can generate exam schedule very smartly. Here mainly generated by the base of date. After submitting user can also add another schedule in the same cell of the table. If user insert inappropriate data, user will update information easily or delete it easily. User can generate same cell more than one schedule.

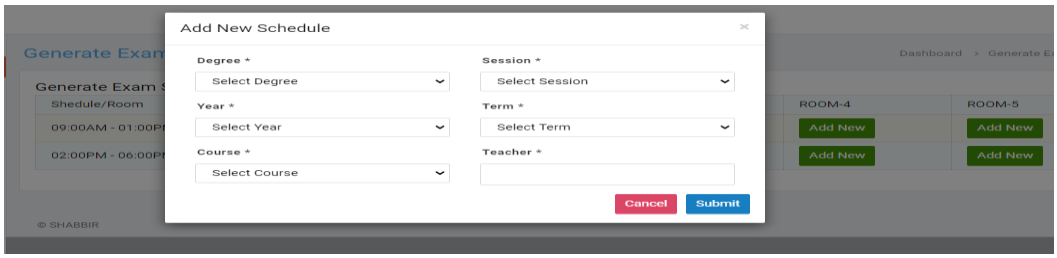


Figure 3.9: Generate Exam Schedule

3.5.7 Show Exam Schedule

This is the final table for exam schedule. Here shows total routine from date to date. There are two dimensions: date and time. Exam schedule table containing date and time schedule. Each cell contains course code, invigilator, and room number. User can print schedule easily.

EXAM SCHEDULE

Routine from : 25-11-2020 To 30-11-2020

Date	09:00AM - 01:00PM	02:00PM - 06:00PM
25-11-20	ICE-5102 (MARK, AR) [ROOM-1]	ICE-1101 (MAH, SJS) [ROOM-3]
26-11-20	ICE-5101 (MARK, MAH) [ROOM-1]	HUM-1105 (AR, SJS) [ROOM-3]
30-11-20	ICE-2203 (MARK, MAH) [ROOM-1]	ICE-1101 (MARK, SJS) [ROOM-4]

NB: MARK = Dr. Md. Ashikur Rahman Khan, AR = Abidur Rahaman, MAH = Mohammad Amzad Hossain, SJS = Sultana Jahan Soheli, ZUS = Zayed Us Salehin, NN = Nishu Nath, AA = Apurbo Adhikary

Figure 3.10: Exam Schedule

3.5.8 Teacher summery

Here is teacher summery of the class in a week. A teacher can get his/her summery by clicking name. This table include SL, Day, Time, Room Name, Course title, Course code, credit hour. User can see the total summery of teachers how much credit he/she takes, when class will be conducted, which room is allotted, which courses class will have held and course code etc. by using single click. It helps a teacher to get his/her schedule easily and keep print copy easily.

TEACHER SUMMERY

Teacher Name : Dr. Md. Ashikur Rahman Khan

SL	Day	Schedule Time	Room Name	Course Title	Course Code	Credit
1	Sunday	09:00 AM - 09:50 AM	ROOM-1	Advanced Database Systems	ICE-5101	03
2	Monday	04:00PM - 04:50PM	ROOM-2	Advanced Digital Communication	ICE-5107	03
3	Tuesday	03:00PM - 03:50PM	ROOM-5	Neural Network	ICE-5110	03
4	Wednesday	09:00 AM - 09:50 AM	ROOM-4	Big data analytics	ICE-5103	03
					Total Credit	12

Figure 3.11: Teacher summery

3.5.9 Add department

The user adds all the department of the institutions by clicking this form. Suppose user can add Information and Communication Engineering, Computer Science and Telecommunication Engineering, Applied Chemistry and Chemical Engineering, Electrical and Electronic Engineering(EEE), Fisheries and Marine Science, Pharmacy, Microbiology, Applied Mathematics, Food Technology and Nutrition Science, Environmental Science and Disaster Management, Biotechnology and Genetic Engineering, Agriculture, Statistics, Oceanography, Biochemistry and Molecular Biology, Zoology etc.

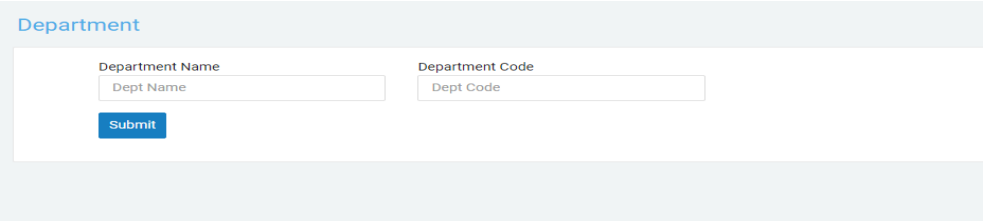


Figure 3.12: Add Department

3.5.10 All Department

After adding department, the user sees all department that he added before by clicking this link. If the user insert inappropriate data, he can update or delete by using edit and delete button. User can see the department code from dept code column.

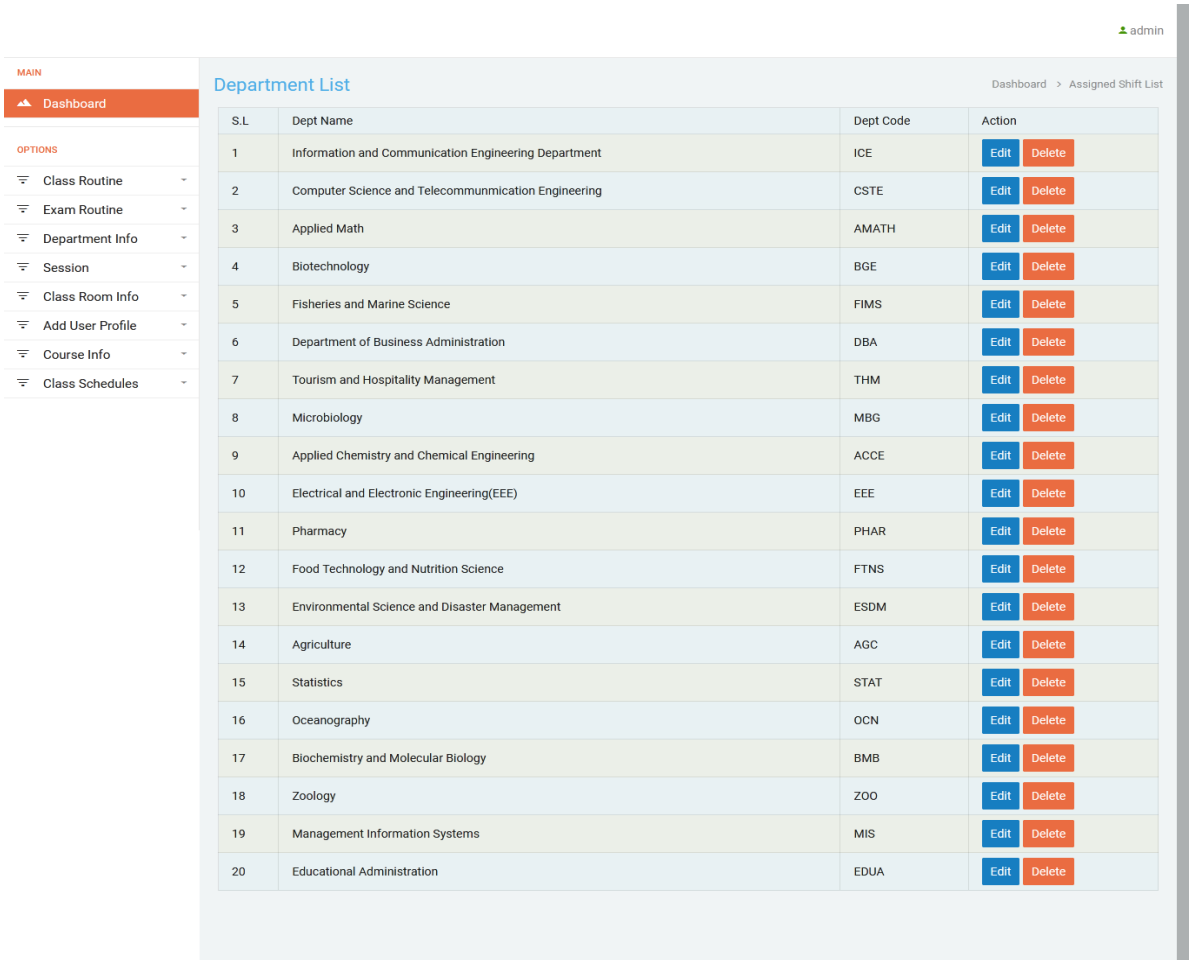


Figure 3.13: All Department List

3.5.11 Add session

User can add the current student's admission session suppose 2012-2013, 2013-2014, 2015-2016 etc. for both honors and master's degree. It is needed for generating Class routine and Exam schedule.

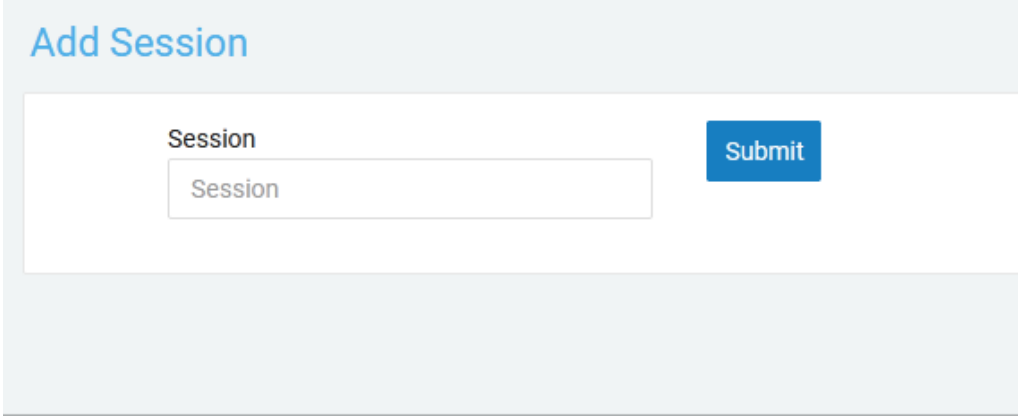


Figure 3.14: Add session

3.5.12 All Session

This section contains total session list of the session. User can edit, update or delete, if needed. This table contains three columns: SL, session and action. Action contains two buttons Edit and Delete

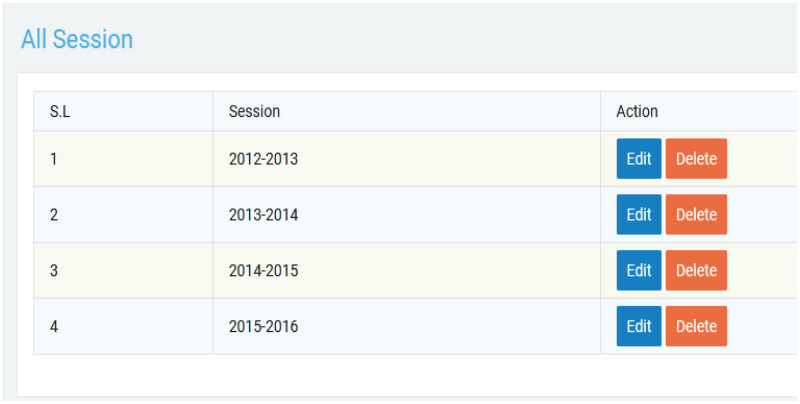


Figure 3.15: All session

3.5.13 Add Class room

User adds classroom using this form. If they want, they can include all the class room in each department. User inserts department name, Class room name, class room no, class room location and building name. Every department has its own class room, so they include their class room information according to their requirement.

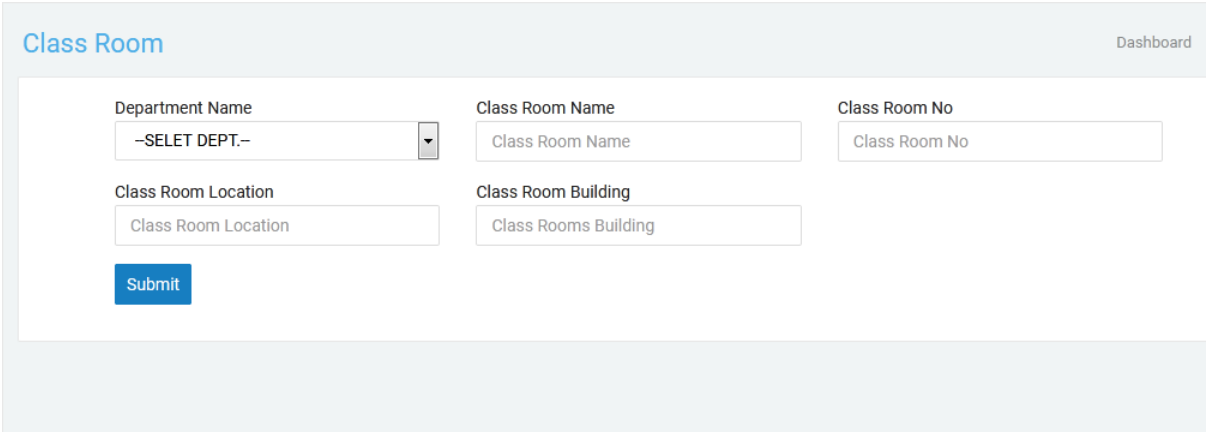


Figure 3.16: Add Class Room

3.5.14 All Class Room

In this list user can see all the class which are added by the user. User can change anything by clicking edit or delete button. In this table, user can see the overall class room information of the institution. The table may contain the departmental class room of Information and Communication Engineering, Computer Science and Telecommunication Engineering, Applied Chemistry and Chemical Engineering, Electrical and Electronic Engineering(EEE), Fisheries and Marine Science, Pharmacy, Microbiology, Applied Mathematics, Food Technology and Nutrition Science etc.

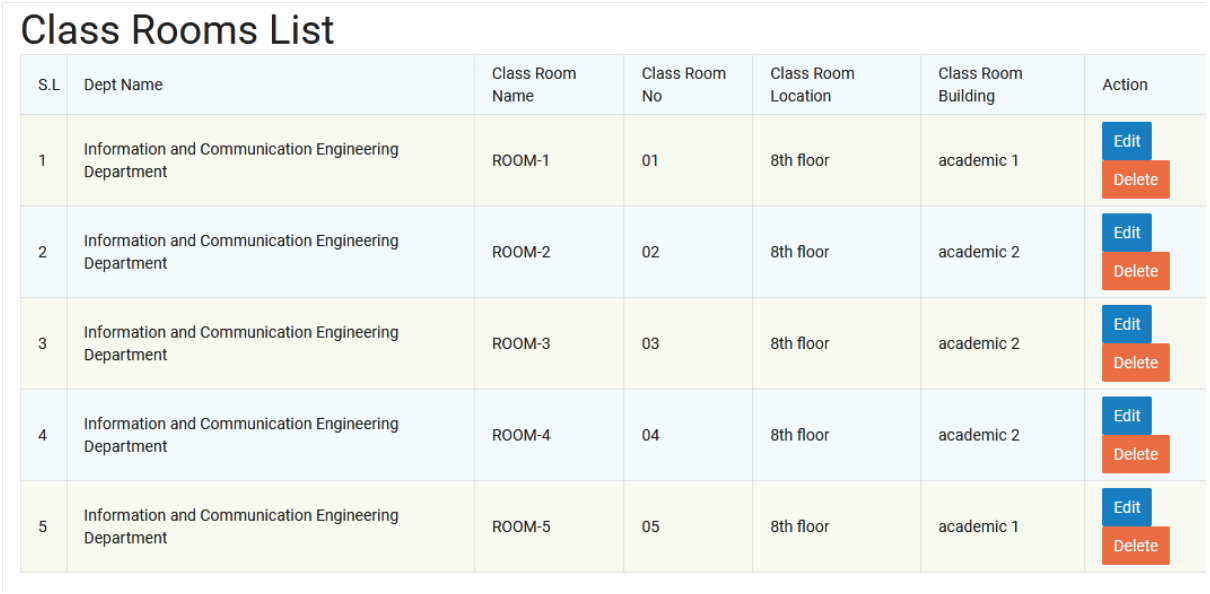


Figure 3.17: Class Room List

3.5.15 Add Teacher

User can add teacher from different department by using this form. This form contains teacher name, Contact number, email, Short name, designation, department name. After submitting this information then teacher information will be added in the database. In this

way, user can add different departments teacher easily. This information is needed for both class routine and exam schedules generation.

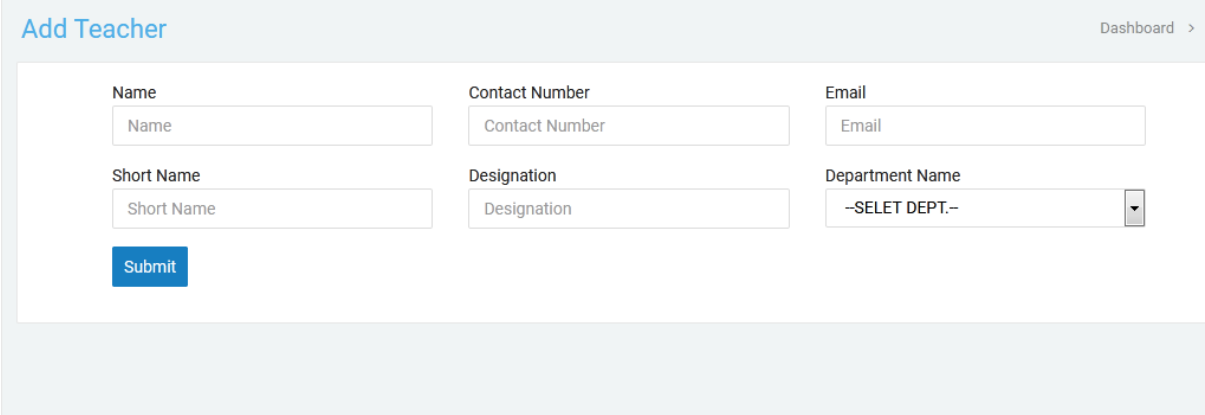


Figure 3.18: Add teacher

3.5.16 Teacher list

In this figure, user see the total teacher list from different department. User can update or delete information easily. User can see the S.L, Name, short name, contact number, email, designation and department in a table.

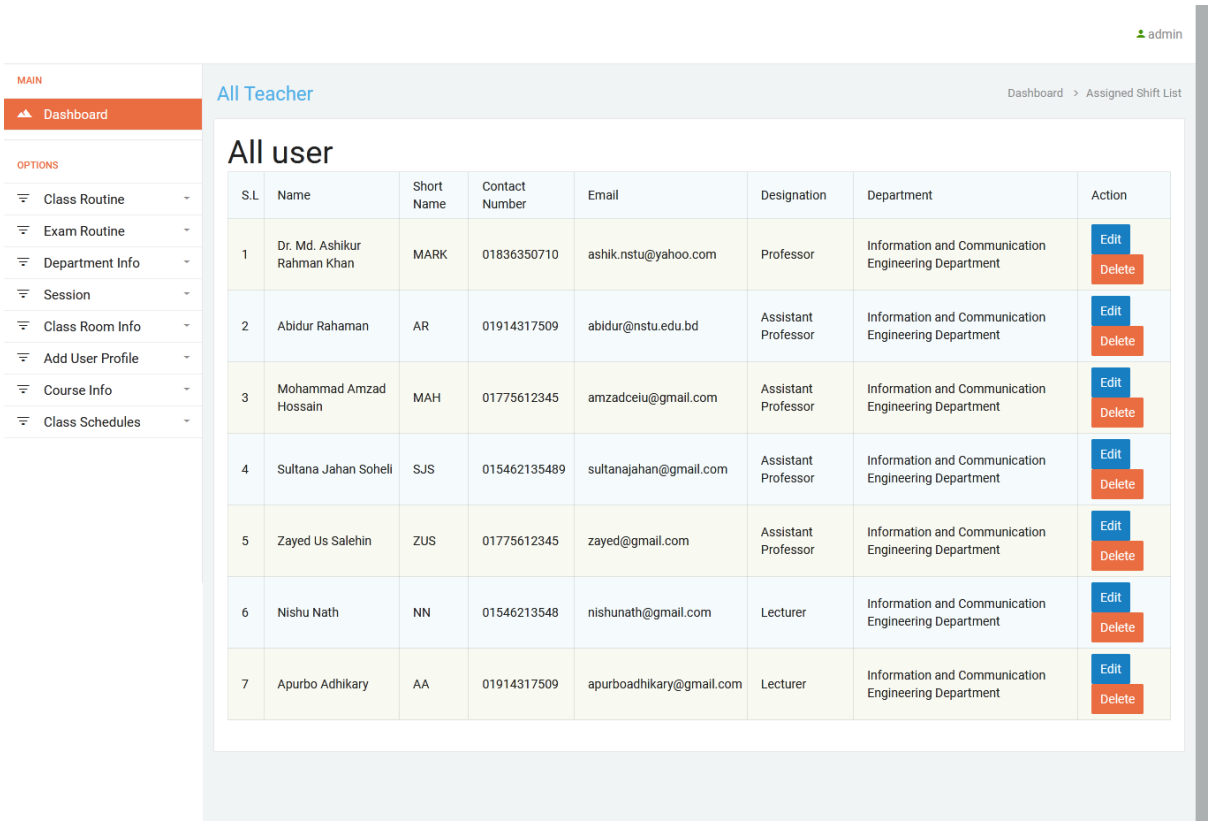


Figure 3.19: Teacher list

3.5.17 Course Information

This form contains department name, course name, course code, credit hour, degree level, year, course term. User can insert information by using this form easily. Suppose user can select department of Information and Communication Engineering, course name is Digital signal processing, course code is ICE-4101, credit hour is 03, degree level is honors, year is 4 and term is one then click the submit, this information will be added to the database. In this way, user can add the total course of the department for generating class routine and exam routine.

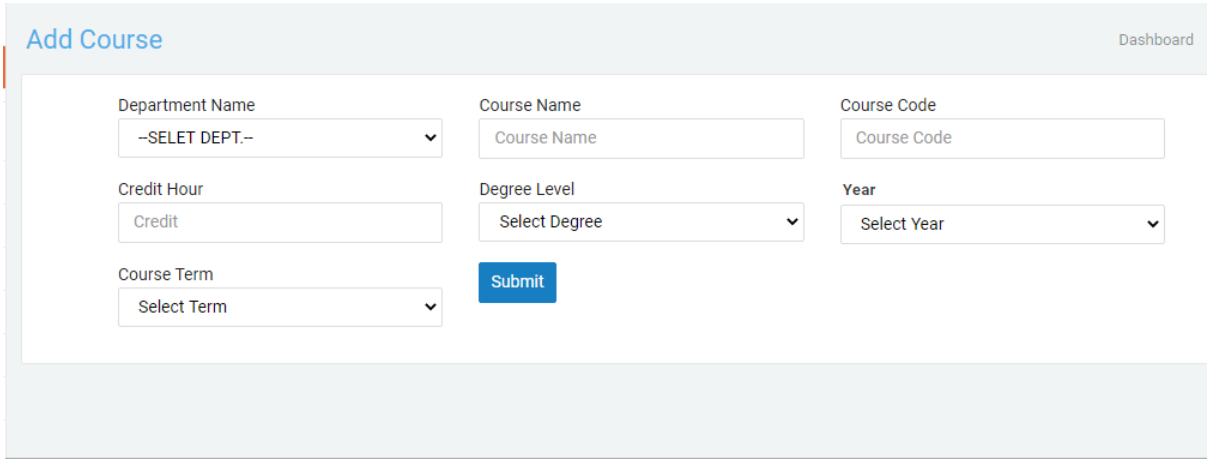


Figure 3.20: Add Course

3.5.18 Courses list

This table contains the all the courses of the department. This table contains nine columns: S.L, Department name, course name, course code, course credit, course year, course term, degree level and action. In action column, user can edit or delete information easily.

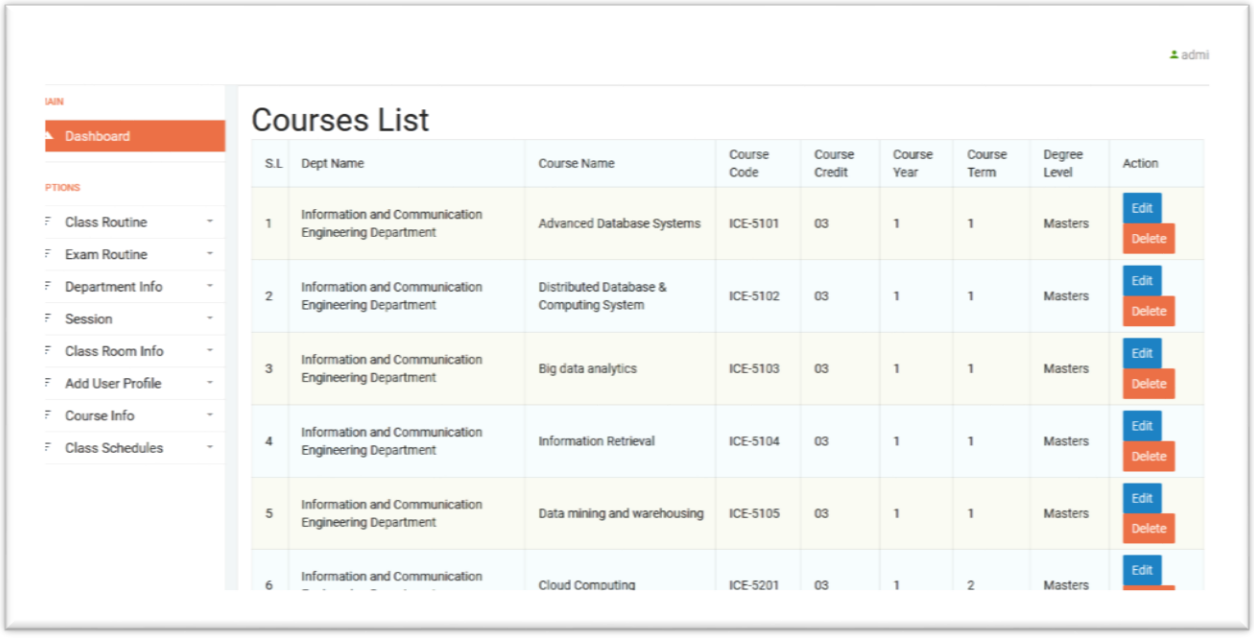


Figure 3.21: All Courses

3.4.19 Schedule

This form is for adding time schedules of class or exam. Here is one drop down button which indicates, the schedules for class or exam. Schedules time indicates the time of class or exam. Here class schedules duration is 50 minutes for each class and exam schedules is for 4 hours for each 3 credit hour class.

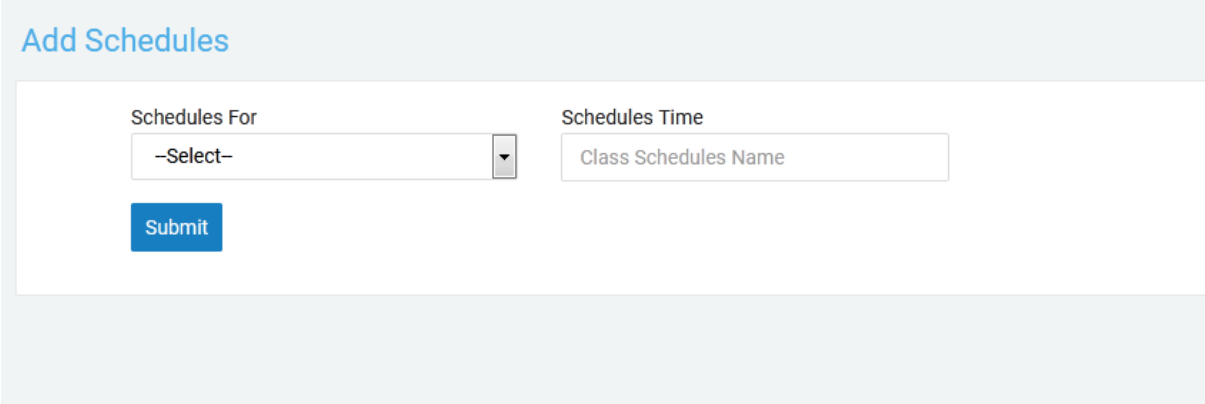


Figure 3.22: Add Schedule Time

3.5.20 All schedule

This table is for total schedules of class or exam. There are four columns including S.L, schedules for, schedules time and action. Action columns have two button: edit and delete that means user can delete or edit schedules time. Schedules time for exam is four-hour duration like 9.00 AM-1.00 PM and for class is fifty minutes' duration like 9.00 AM-9.50 AM.

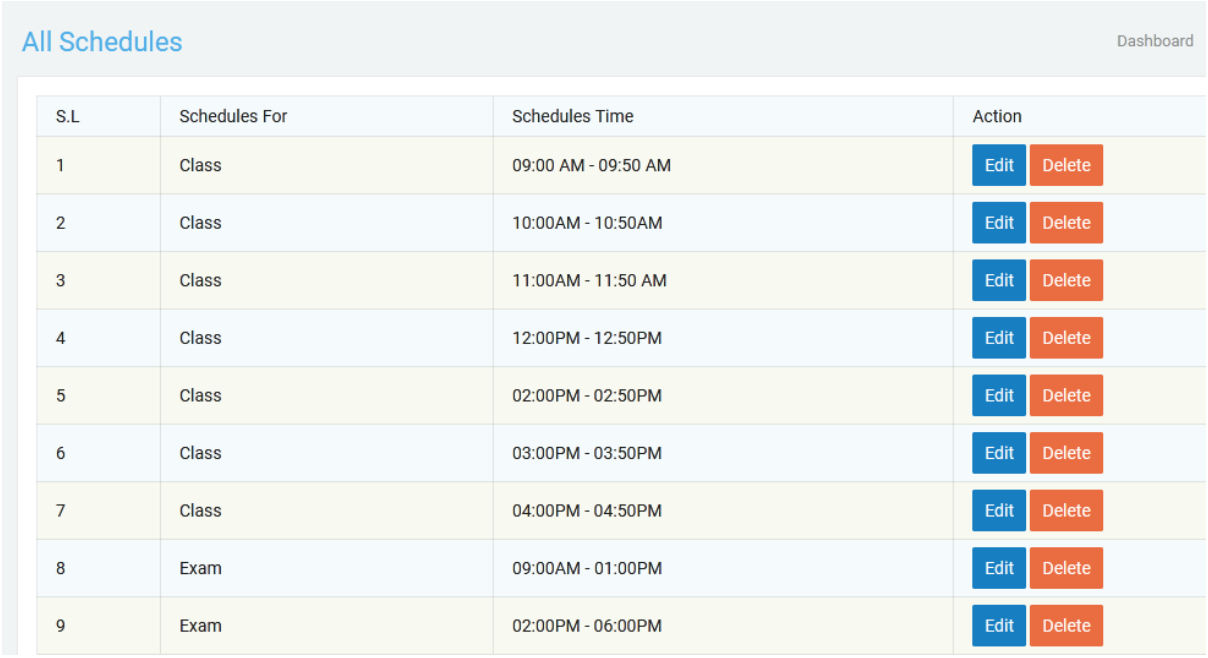


Figure 3.23: All Schedule Time

3.6 SUMMARY

We maintained research methodology in a proper way. We insert data to the database sequentially that's why we can search data from the database accurately. We generate class routine, exam schedule, print class routine, print exam schedule, print teacher summery in a systemic way without any problem.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In the paper based method teachers can make examination schedules and class routine to write on a paper. It very time consuming matter and it is very difficult to manage class room allocation, teacher allocation. But Automated Class Routine and Exam Schedule Management System(ACRESMS) can removes all the such kind of complexity.

4.2 COMPARE BETWEEN OLD SYSTEM

This routine is old routine. This format is generated by paper based hand writing method. It takes more time. There is no degree level and session columns. So it is very difficult to understand. When generate routine in analogue or hand writing method sometimes we see same class, different batch, same time has a class. So it is a great complexity. So it is very tough to identify.

Day	Year, Term	8:00-8:50	9:00-9:50	10:00-10:50	11:00-11:50	12:00-12:50		2:00-2:50	3:00-3:50	4:00-4:50
Sunday	Y-1, T-1			ICE-1103(Cl-2) MARK		ICE-1101(Cl-1) SR		HUM-1105(Cl-3) SI		
	Y-2, T-1		ICE-2105(Cl-2) AU	ICE-2101(Cl-1) AH	ICE-2109(Cl-1) SR			ICE-2103(Cl-1) ZUS	ICE-2110(Cl-1) Gr. SR	
	Y-2, T-2		ICE-2201(Cl-1) SJS		ICE-2204(Cl-3) Gr. ZUS+AU	ICE-2203 (Cl-2) ZUS		ICE-2209(Cl-2) SJS	ICE-2204(Cl-1)Gr. ZUS+AU	
Monday	Y-1, T-1		ENG-1109(Cl-3) IJN	PHY-1103(Cl-3) MARK		ICE-1101(Cl-1) SR		ICE-1105(Cl-2) MR		
	Y-2, T-1		ICE-2101(Cl-1) AH	ICE-2107(Cl-1) SJS	MATH-2113(Cl-3) JH			ICE-2103(Cl-1) ZUS	ICE-2106(Cl-1) Gr. AU	
	Y-2, T-2		ICE-2209(Cl-2) SJS	ICE-2205(Cl-2) ZUS	ICE-2206(Cl-2) Gr. ZUS				MATH-2211(Cl-1) MNU	
Tuesday	Y-1, T-1		PHY-1103(Cl-2) MARK		ICE-1103(Cl-2) MARK		BREAK	HUM-1105(Cl-1) TI		
	Y-2, T-1			ICE-2107(Cl-1) SJS	ICE-2103(Cl-1) ZUS	ICE-2101(Cl-1) AH			ICE-2104(Cl-1) Gr. ZUS	
	Y-2, T-2		MATH-2211(Cl-1) MNU	ICE-2203(Cl-2) AU		ICE-2207(Cl-1) SR		ICE-2203(Cl-2) ZUS	ICE-2201(Cl-2) Gr. SJS	
Wednesday	Y-1, T-1									
	Y-2, T-1		ICE-2105(Cl-2) AU	ICE-2101(Cl-2) AH	ICE-2109(Cl-1) SR	MATH-2113(Cl-1) JH		ICE-2107(Cl-1) SJS	ICE-2110(Cl-1) Gr. SR	
	Y-2, T-2		ICE-2201(Cl-1) SJS	ICE-2205(Cl-1) ZUS	ICE-2206(Cl-2) Gr. ZUS			ICE-2207(Cl-2) SR	ICE-2209(Cl-2) SJS	
Thursday	Y-1, T-1		ENG-1109(Cl-3) IJN	ICE-1105(Cl-3) MR		ICE-1101(Cl-1) SR		ICE-1103(Cl-3) MARK		
	Y-2, T-1		ICE-2106(Cl-1) Gr. AU		ICE-2105(Cl-2) AU	ICE-2107(Cl-2) SJS		ICE-2104(Cl-1) Gr. ZUS		ICE-2109(Cl-1) SR
	Y-2, T-2		ICE-2201(Cl-2) SJS	ICE-2205(Cl-2) ZUS	ICE-2207(Cl-1) SR			ICE-2207(Cl-2) Gr. SR		

Figure 4.1: Old Routine Format

In automated class routine, there is degree level and session column are added. It is automated system just user insert some data and it is automatically generated. This system saves lot of time. And here few paper based work. Paper is needed when user want to print the routine. It solved all the complexity of paper based or handwriting method.

ROUTINE

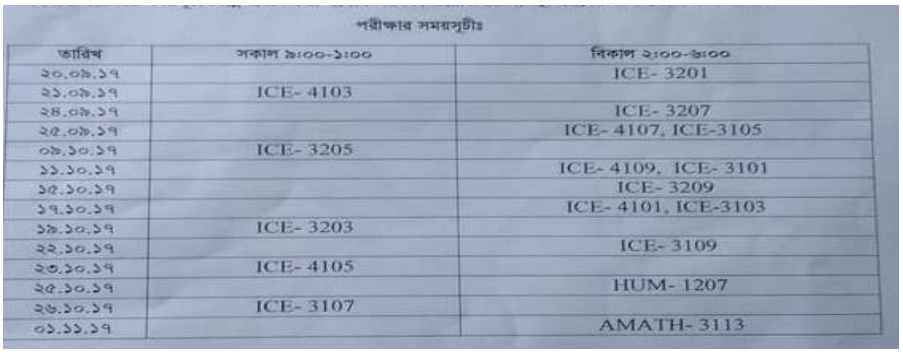
Days	Degree Level	Session	Year, Term	09:00 AM - 09:50 AM	10:00AM - 10:50AM	11:00AM - 11:50 AM	12:00PM - 12:50PM	02:00PM - 02:50PM	03:00PM - 03:50PM	04:00PM - 04:50PM
Sunday	Honours	2012-2013	Y-1, T-1			ICE-1101 (CL-ROOM-2) AR				
	Masters	2012-2013	Y-1, T-1	ICE-5101 (CL-ROOM-1) MARK						
Monday	Honours	2012-2013	Y-1, T-1							
	Masters	2012-2013	Y-1, T-1	ICE-5104 (CL-ROOM-1) ZUS						
Tuesday	Honours	2012-2013	Y-1, T-1							
	Masters	2012-2013	Y-1, T-1			ICE-5110 (CL-ROOM-2) ZUS				
Wednesday	Honours	2012-2013	Y-1, T-1							
	Masters	2012-2013	Y-1, T-1							
Thursday	Honours	2012-2013	Y-1, T-1		HUM-1105 (CL-ROOM-4) AA					
	Masters	2012-2013	Y-1, T-1	ICE-5104 (CL-ROOM-1) MAH						

NB: MARK = Dr. Md. Ashikur Rahman Khan, AR = Abidur Rahaman, MAH = Mohammad Amzad Hossain, SJS = Sultana Jahan Soheli, ZUS = Zayed Us Salehin, NN = Nishu Nath, AA = Apurbo Adhikary

Print Routine

Figure 4.2: Automated routine

Now we discuss the exam schedule, in old system, the full system is generated by user by using paper and it is very difficult to generate because there is no exam room description, time, batch and session complexity. It takes more time. There is no teacher information. Old examination schedule is given in figure 24



পরীক্ষার সময়সূচী

তারিখ	সকাল ৯:০০-১:০০	বিকাল ২:০০-৬:০০
২০.০৯.১৭		ICE- 3201
২১.০৯.১৭	ICE- 4103	
২৪.০৯.১৭		ICE- 3207
২৫.০৯.১৭		ICE- 4107, ICE-3105
০৯.১০.১৭	ICE- 3205	
১১.১০.১৭		ICE- 4109, ICE- 3101
১৫.১০.১৭		ICE- 3209
১৭.১০.১৭		ICE- 4101, ICE-3103
১৯.১০.১৭	ICE- 3203	
২২.১০.১৭		ICE- 3109
২৩.১০.১৭	ICE- 4105	
২৫.১০.১৭		HUM- 1207
২৬.১০.১৭	ICE- 3107	
০১.১১.১৭		AMATH- 3113

Figure 4.3: Old exam Schedule

But in the new system, we just remove the old systems limitation. here we include teacher information and room number. So it is very clear view then old system. it saves much time. It is very easy to understand. New exam schedule is given in figure

EXAM SCHEDULE

Schedules from : 01-10-2020 To 10-10-2020

Date	09:00AM - 01:00PM	02:00PM - 06:00PM
04-10-20	ICE-3105 (AR, SJS)	ICE-5105 (AR, ZUS)
05-10-20	ICE-1101 (MARK, MAH)	ICE-5102 (SJS, ZUS)
06-10-20	ICE-4201 (AR, MAH)	-

NB: MARK = Dr. Md. Ashikur Rahman Khan, AR = Abidur Rahaman, MAH = Mohammad Amzad Hossain, SJS = Sultana Jahan Soheli, ZUS = Zayed Us Salehin, NN = Nishu Nath, AA = Apurbo Adhikary

Figure 4.4: New Exam Schedule

4.3 SUMMARY

In this section, we just compare the present system and Automated Routine and Exam Schedule Management System. We ensure that our system will be more attractive and helpful than present system. It saves time and solved various complexities that happened in present or running system

CHAPTER 5

CONCLUSIONS

In Automated Class Routine and Exam Schedule Management System, we implemented best system. In class routine section, we include class room info, teacher info, teacher summery, session, schedules, course info etc. All the data are stored in a database sequentially and query the data very quickly. In exam schedule section, we include course info, room number, teacher info in an organized way. All the data are contained in the database table. User can print both class routine and exam schedule easily. Searching data from database is very easy. User can operate the system without any problem. Data in the database are stored in an organized process. So it is very helpful for maintaining the system. Database table creating process is very well decorated. Same type data stored in the same table. It increases searching efficiency.

We try to generate user friendly Automated Routine and Exam Schedule management system. We know, no systems will provide hundred percent accuracy. So the system may have some limitation. But in the future, if the user will find any mistake or mismatch information, our team will fix the problem

CHAPTER 6

REFERENCES

Some references are given below

[1] Qu, R., Burke, E. K., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling, 12(1), 55–89.

[2] Nasser R.Sabar·Masri Ayob·RongQu· Graham Kendall "A graph coloring constructive hyper-heuristic for examination timetabling problems" Springer Science + Business Media, LLC 2011

[3] Alireza R K and Mehrdad N K, "A new binary model for university examination timetabling: a case study" Komijan and Koupaei Journal of Industrial Engineering International 2012,8:28

[4] Software Engineering | SDLC V-Model-Definition[Online].Available:https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/. [Accessed: 03-Mar-2020].

[5] Ian Somerville, 2009. "Software Engineering-ninth edition", Ninth edition, PP-29 & 206.

[6] Roger S. Pressman, 2001. “Software Engineering”, Fifth edition, PP-434.

[7] Hypertext Markup Language.[Online].

Available:https://www.w3schools.com/html/default.asp, [Accessed: 10-Mar-2020].

[8] “Cascading Style Sheet attributes and its formula .[Online].Available: http://w3schools.com/css/default.asp. [Accessed: 05-Mar-2020].

[9] Javascript and its function activity.[Online]. Available:http://w3schools.com/js/default.asp. [Accessed: 15-Mar-2020].

[10] PHP language details .[online]. Available:http://w3schools.com/php/default.asp. [Accessed: 10-Mar-2020].

[11] Structures Query Language [online]. Available:http://w3schools.com/sql/default.asp. [Accessed: 25-april-2020].

CHAPTER 7

APPENDICES

Important project segments are given below

```
<?php
\App\Session::init();
\App\Session::checkLoginSession();
$levelSettings = new \App\level\Level();
$setting=new \App\settings\Setting();
$protocol = ((!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] != 'off') ||
$_SERVER['SERVER_PORT'] == 443) ? "https://" : "http://";
$hostURL = $protocol . $_SERVER['HTTP_HOST'].'/';
$url = $hostURL;
if ($hostURL == 'http://localhost/'){
   $url = $hostURL. explode("/",
$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI'])[1].'/';
}
```

```
?>

<!DOCTYPE html>

<html lang="en">

<!-- Mirrored from hencework.com/theme/doodle/full-width-light/index.html by HTTrack Website Copier/3.x [XR&CO'2014], Tue, 06 Nov 2018 08:10:40 GMT -->

<head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />

    <title><?=$setting->getTitle()?></title>

    <meta name="description" content="." />

    <!-- Favicon -->

    <link rel="shortcut icon" href="favicon.ico">

    <link rel="icon" href="favicon.ico" type="image/x-icon">

    <base href="<?php echo $url ?>">

    <link href="files/bower_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>


    <!-- Morris Charts CSS -->

    <!--    <link href="files/bower_components/morris.js/morris.css" rel="stylesheet" type="text/css"/>-->


    <!-- Data table CSS -->
```

```
    <!--    <link
href="files/bower_components/datatables/media/css/jquery.dataTables.min.css"
rel="stylesheet" type="text/css"/>-->

    <!--    <link
href="https://cdn.datatables.net/buttons/1.5.2/css/buttons.dataTables.min.css"
rel="stylesheet" type="text/css"/>-->


    <!--    <link href="files/bower_components/jquery-toast-
plugin/dist/jquery.toast.min.css" rel="stylesheet" type="text/css">-->


    <!--    <link href="files/bower_components/awesome-bootstrap-checkbox/awesome-
bootstrap-checkbox.css" rel="stylesheet" type="text/css">-->

    <!--    <link
href="files/bower_components/owl.carousel/dist/assets/owl.carousel.min.css"
rel="stylesheet" type="text/css">-->


    <!--    <link href="files/bower_components/awesome-bootstrap-checkbox/awesome-
bootstrap-checkbox.css" rel="stylesheet" type="text/css">-->

    <!-- select2 CSS -->

    <link href="files/bower_components/select2/dist/css/select2.min.css" rel="stylesheet"
type="text/css"/>

    <!--    dropify-->

    <link href="files/bower_components/dropify/dist/css/dropify.min.css"
rel="stylesheet" type="text/css"/>

    <!---->
```

```
    <!--    <link href="dist/css/simple-line-icons.css" rel="stylesheet" type="text/css"/>-->


    <!--    alertyfy-->

    <link rel="stylesheet"
href="//cdn.jsdelivr.net/npm/alertifyjs@1.11.2/build/css/alertify.min.css"/>

    <!-- Default theme -->

    <link rel="stylesheet"
href="//cdn.jsdelivr.net/npm/alertifyjs@1.11.2/build/css/themes/default.min.css"/>


    <!-- bootstrap-tagsinput CSS -->

    <!--    <link href="files/bower_components/bootstrap-tagsinput/dist/bootstrap-
tagsinput.css" rel="stylesheet" type="text/css"/>-->


    <!-- Bootstrap Daterangepicker CSS -->

    <link href="files/bower_components/bootstrap-daterangepicker/daterangepicker.css"
rel="stylesheet" type="text/css"/>


    <!-- Summernote css -->

    <link rel="stylesheet"
href="files/bower_components/summernote/dist/summernote.css" />

    <!-- Integration CSS -->

    <!-- Calendar CSS -->

    <!--    <link href="vendors/bower_components/fullcalendar/dist/fullcalendar.css"
rel="stylesheet" type="text/css"/>-->
```

```
    <!-- bootstrap-select CSS -->
    <!--    <link href="vendors/bower_components/bootstrap-select/dist/css/bootstrap-select.min.css" rel="stylesheet" type="text/css"/>-->

    <link href="dist/css/custom-final-visit-plan.css" rel="stylesheet" type="text/css">
    <link href="dist/css/visit-plan.css?ver=1.3" rel="stylesheet" type="text/css">

    <!-- Integration CSS -->
    <!-- Custom CSS -->
    <link href="dist/css/style.css?ver=1.4" rel="stylesheet" type="text/css">
    <link href="dist/css/custom-css.css?ver=1.2" rel="stylesheet" type="text/css">
    <style>

    </style>
</head>

<?php
$body_class='';
if (\App\Session::get('default_languageID')== 1) { // 1 =  English
    $body_class='english';
}elseif (\App\Session::get('default_languageID') == 2){ // 2 = Bangla
```

```
    $body_class='bangla';
}else{
    $body_class='english';// Default = English
}
?>



<body id="<?php echo $body_class?>">


<!-- /Preloader -->


<div class="wrapper theme-1-active pimary-color-red slide-nav-toggle">
    <!-- Top Menu Items -->
    <nav class="navbar navbar-inverse navbar-fixed-top">


        <div id="mobile_only_nav" class="mobile-only-nav pull-right" style="overflow:
visible">
            <ul class="nav navbar-right top-nav pull-right">


<!--                <li class="dropdown alert-drp">-->
<!--                    --><?php //$newstock=new \App\product\ProductInventory();
//                    $newstock=$newstock->newStockNotification();
```

```
//                ?>

<!--                --><?php //if(!empty($newstock)){?>

<!--                    <a href="view/stock/new-stock.php" title="New Stock" aria-expanded="true"><i class="zmdi zmdi-notifications top-nav-icon"></i><span class="top-nav-icon-badge">--><?php //echo $newstock ?><!--</span></a>-->

<!--                --><?php //}?>

<!--            </li>-->


            <li class="dropdown auth-drp">

                <a  href="#" class="dropdown-toggle pr-0" data-toggle="dropdown"><i class="zmdi zmdi-account txt-success"></i>  <?php echo \App\Session::get('user_name')?></a>

                <ul class="dropdown-menu user-auth-dropdown" data-dropdown-in="flipInX" data-dropdown-out="flipOutX" >

                    <li>

                        <a  href="view/user/profile-update.php" id=""><i class="zmdi zmdi-check-circle"></i><span>Update Profile</span></a>

                    </li>

                    <li>

                        <a  href="view/user/password.php" id="changePassword" ><i class="zmdi zmdi-sign-in"></i><span> Change Password</span></a>

                    </li>

                    <li>
```

```
                    <!--<a  href="javascript:void(0)" id="logout"
onclick="$('#logoutForm').submit()"><i class="zmdi zmdi-power"></i><span>Log
Out</span></a>-->

                    <a  href="view/logout.php" id="logout" ><i class="zmdi zmdi-
power"></i><span>Log Out</span></a>

                </li>

                <!--<form action="" id="logoutForm" method="post">-->

                <!--    <input type="hidden" name="logout">-->

                <!--</form>-->

              </ul>

            </li>

          </ul>

        </div>

    </nav>

    <!-- /Top Menu Items -->


    <!-- Left Sidebar Menu -->

    <div class="fixed-sidebar-left">

        <ul class="nav navbar-nav side-nav nicescroll-bar">

            <li class="navigation-header">

                <span><?php echo $levelSettings->levelName(1);?></span>

                <i class="zmdi zmdi-more"></i>

            </li>
```

```
<li>
    <a class="active" href="view/" data-toggle="collapse" data-target="#dashboard_dr"><div class="pull-left"><i class="zmdi zmdi-landscape mr-20"></i><span class="right-nav-text"><?php echo $levelSettings->levelName(2);?></span></div><div class="pull-right"></div><div class="clearfix"></div></a>
</li>

<li><hr class="light-grey-hr mb-10"/></li>
<li class="navigation-header">
    <span><?php echo $levelSettings->levelName(3);?></span>
    <i class="zmdi zmdi-more"></i>
</li>

<li>
    <a href="javascript:void(0);" data-toggle="collapse" data-target="#rtngenerate"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-text">Class Routine</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
    <ul id="rtngenerate" class="collapse collapse-level-1 two-col-list">
```

```
<li>
    <a href="view/routine/generate-routine.php">Generate Routine</a>
</li>

<li>
    <a href="view/routine/show-routine.php">Show Routine</a>
</li>
<li>
    <a href="view/routine/routineprint.php">show only table</a>
</li>
<li>
    <a href="view/routine/teacher-summery.php">Teacher Summery</a>
</li>
<li>
    <a href="view/routine/teacher-summery.php">Teacher Summery</a>
</li>

<!-- <li>
    <a href="view/routine/show-routine-multi2.php">show routine with consecutive merged different try</a>
</li> -->
```

```
            </ul>
        </li>
    <li>
            <a href="javascript:void(0);" data-toggle="collapse" data-
target="#hhh4"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-
20"></i><span class="right-nav-text">Exam Routine</span></div><div class="pull-
right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="hhh4" class="collapse collapse-level-1 two-col-list">

              <li>
                  <a href="view/routine/generate-exam-routine.php">Generate Exam
Routine</a>
              </li>
                <li>
                  <a href="view/routine/show-exam-routine.php">Show Exam
Routine</a>
              </li>

            </ul>
        </li>
       <li>
            <a href="javascript:void(0);" data-toggle="collapse" data-target="#hhh"><div
class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-
```

```
text">Department Info</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="hhh" class="collapse collapse-level-1 two-col-list">

                <li>
                    <a href="view/routine/adddepartment.php">Add Department</a>
                </li>
                <li>
                    <a href="view/routine/showdepartment.php">All Department</a>
                </li>


            </ul>
        </li>

 <li>
            <a href="javascript:void(0);" data-toggle="collapse" data-target="#hhhh"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-text">Session</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="hhhh" class="collapse collapse-level-1 two-col-list">

                <li>
```

```
                <a href="view/routine/addsession.php">Add Session</a>
            </li>
            <li>
                <a href="view/routine/showsession.php">All Session</a>
            </li>

        </ul>
    </li>
    <li>
        <a href="javascript:void(0);" data-toggle="collapse" data-target="#hh"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-text">Class Room Info</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
        <ul id="hh" class="collapse collapse-level-1 two-col-list">
            <li>
                <a href="view/routine/addclassroom.php">Add Class Room</a>
            </li>
            <li>
                <a href="view/routine/showclassroom.php">All Class Room</a>
            </li>

        </ul>
```

```
        </li>
    <li>
            <a href="javascript:void(0);" data-toggle="collapse" data-target="#h"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-text">Add User Profile</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="h" class="collapse collapse-level-1 two-col-list">

              <li>
                 <a href="view/routine/adduserprofileinfo.php">Add Teacher</a>
              </li>
              <li>
                 <a href="view/routine/showuserprofileinfo.php">All Teacher</a>
              </li>


            </ul>
        </li>


        <li>
            <a href="javascript:void(0);" data-toggle="collapse" data-target="#s"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-
```

```
text">Course Info</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="s" class="collapse collapse-level-1 two-col-list">

              <li>
                 <a href="view/routine/addcourse.php">Add Course</a>
              </li>
              <li>
                 <a href="view/routine/showcourse.php">All Course</a>
              </li>


            </ul>
          </li>

           <li>
              <a href="javascript:void(0);" data-toggle="collapse" data-target="#k"><div class="pull-left"><i class="zmdi zmdi-filter-list mr-20"></i><span class="right-nav-text">Class Schedules</span></div><div class="pull-right"><i class="zmdi zmdi-caret-down"></i></div><div class="clearfix"></div></a>
            <ul id="k" class="collapse collapse-level-1 two-col-list">

              <li>
```

```
                    <a href="view/routine/addclassschedules.php">Add Schedules</a>
                </li>
                <li>
                    <a href="view/routine/showclassshedules.php">All Schedules</a>
                </li>



            </ul>
        </li>



        <?php if(in_array(25,\App\Session::get('permissions')) ||
in_array(1,\App\Session::get('permissions'))  ||
in_array(33,\App\Session::get('permissions'))){?>

        <?php }?>



    </ul>
</div>
```

```
    <!-- /Left Sidebar Menu -->

    <!-- Right Sidebar Backdrop -->

    <div class="right-sidebar-backdrop"></div>

    <!-- /Right Sidebar Backdrop -->









<!-- Footer -->

<footer class="footer container-fluid pl-30 pr-30">

    <div class="row">

        <div class="col-sm-12">



        </div>

    </div>

</footer>

<!-- /Footer -->



</div>

<!-- /Main Content -->



</div>
```

```
<!-- /#wrapper -->


<!-- JavaScript -->


<!-- jQuery -->
<script src="files/bower_components/jquery/dist/jquery.min.js"></script>
<!--<script-->
<!--    src="https://code.jquery.com/jquery-2.1.4.min.js"-->
<!--    integrity="sha256-
8WqyJLuWKRBVhxXIL1jBDD7SDxU936oZkCnxQbWwJVw="-->
<!--    crossorigin="anonymous"></script>-->
<!-- Bootstrap Core JavaScript -->
<script src="files/bower_components/bootstrap/dist/js/bootstrap.min.js"></script>
<!-- Data table JavaScript -->
<!--<script
src="files/bower_components/datatables/media/js/jquery.dataTables.min.js"></script>--
>


<!-- Slimscroll JavaScript -->
<script src="dist/js/jquery.slimscroll.js"></script>


<!-- simpleWeather JavaScript -->
<!--<script src="files/bower_components/moment/min/moment.min.js"></script>-->
```

```
<!--<script
src="files/bower_components/simpleWeather/jquery.simpleWeather.min.js"></script>--
>

<!--<script src="files/dist/js/simpleweather-data.js"></script>-->


<!-- Progressbar Animation JavaScript -->

<!--<script
src="files/bower_components/waypoints/lib/jquery.waypoints.min.js"></script>-->

<!--<script
src="files/bower_components/jquery.counterup/jquery.counterup.min.js"></script>-->


<!-- Fancy Dropdown JS -->

<!--<script src="dist/js/dropdown-bootstrap-extended.js"></script>-->


<!-- Sparkline JavaScript -->

<!--<script src="files/jquery.sparkline/dist/jquery.sparkline.min.js"></script>-->


<!-- Owl JavaScript -->

<!--<script
src="files/bower_components/owl.carousel/dist/owl.carousel.min.js"></script>-->


<!-- ChartJS JavaScript -->

<script src="files/chart.js/Chart.min.js"></script>
```

```
<!-- Morris Charts JavaScript -->

<!--<script src="files/bower_components/raphael/raphael.min.js"></script>-->

<!--<script src="files/bower_components/morris.js/morris.min.js"></script>-->

<!--<script src="files/bower_components/jquery-toast-
plugin/dist/jquery.toast.min.js"></script>-->

<!---->

<!-- Switchery JavaScript -->

<!--<script src="files/bower_components/switchery/dist/switchery.min.js"></script>-->


<!-- Select2 JavaScript -->

<script src="files/bower_components/select2/dist/js/select2.full.min.js"></script>


<!-- Form Flie Upload Data JavaScript -->

<script src="dist/js/form-file-upload-data.js"></script>


<!--dropify-->

<script src="files/bower_components/dropify/dist/js/dropify.min.js"></script>


<!-- Data table JavaScript -->

<!--<script
src="files/bower_components/datatables/media/js/jquery.dataTables.min.js"></script>--
>

<!--<script src="dist/js/dataTables-data.js"></script>-->
```

<!---->

<!--<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/pdfmake.min.js"></script>-->

<!--<script src="https://cdnjs.cloudflare.com/ajax/libs/pdfmake/0.1.36/vfs_fonts.js"></script>-->

<!--<script src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.html5.min.js"></script>-->

<!--<script src="https://cdn.datatables.net/buttons/1.5.2/js/buttons.print.min.js"></script>-->

<!--alertyfy-->

<script src="//cdn.jsdelivr.net/npm/alertifyjs@1.11.2/build/alertify.min.js"></script>

<!-- Bootstrap Tagsinput JavaScript -->

<!--<script src="files/bower_components/bootstrap-tagsinput/dist/bootstrap-tagsinput.min.js"></script>-->

<!-- Moment JavaScript -->

<!--<script type="text/javascript" src="files/bower_components/moment/min/moment-with-locales.min.js"></script>-->

<!-- Bootstrap Daterangepicker JavaScript -->

<!--<script src="files/bower_components/bootstrap-daterangepicker/daterangepicker.js"></script>-->

<!-- Form Picker Init JavaScript -->

<!--<script src="dist/js/form-picker-data.js"></script>-->

<!-- Summernote Plugin JavaScript -->

<script src="files/bower_components/summernote/dist/summernote.min.js"></script>

<!-- Summernote Wysuhtml5 Init JavaScript -->

<script src="dist/js/summernote-data.js"></script>

<!-- Bootstrap Touchspin JavaScript -->

<script src="files/bower_components/bootstrap-touchspin/dist/jquery.bootstrap-touchspin.min.js"></script>

<!-- Form Advance Init JavaScript -->

<script src="dist/js/form-advance-data.js"></script>

<!-- Integration JS --->

<script src="dist/js/visit-plan.js"></script>

<!-- Noty CSS and JS -->

<!--<link rel="stylesheet" href="dist/css/noty.css">-->

<!--<script src="dist/js/noty.min.js"></script>-->

<!-- Tinymce JavaScript -->

<!--<script src="vendors/bower_components/tinymce/tinymce.min.js"></script>-->

```
<!-- Tinymce Wysuhtml5 Init JavaScript -->

<!--<script src="dist/js/tinymce-data.js"></script>-->


<!-- Calender JavaScripts -->

<!--<script src="vendors/bower_components/moment/min/moment.min.js"></script>-->

<!--<script src="vendors/jquery-ui.min.js"></script>-->

<!--<script
src="vendors/bower_components/fullcalendar/dist/fullcalendar.min.js"></script>-->

<!--<script src="dist/js/fullcalendar-data.js"></script>-->


<script src="vendors/bower_components/bootstrap-
validator/dist/validator.min.js"></script>

<!-- Bootstrap Select JavaScript -->

<script src="vendors/bower_components/bootstrap-select/dist/js/bootstrap-
select.min.js"></script>


<script>

//    $("#dataTable").DataTable();

</script>

<!-- Integration JS --->


<!-- Init JavaScript -->

<!--<script src="dist/js/init.js"></script>-->
```

```
<script>
    alertify.set('notifier','position', 'top-center');
</script>
<audio id="audio" src="http://www.soundjay.com/button/beep-07.wav" ></audio>
<script>
    function playSound() {
        var sound = document.getElementById("audio");
        sound.play();
    }
    $(document).ready(function() {
//        $('#datatable').DataTable({
//            language: {
//                searchPlaceholder: "Search data"
//            }
//        });
        menuOpen();

        $("#searchInput").on("keyup", function() {
            var value = $(this).val().toLowerCase();
            $("#mytable tr").filter(function() {
                $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
```

```
            });
        });
    })

    function menuOpen() {
        var folder=window.location.pathname.split('/')[3];
        console.log(folder)
        if(folder=='product' || folder=='brand' || folder=='category')
        {
            $('.product_ul').addClass('in');
        }
    }

    $('.alert').delay(5000).fadeOut(1000,function () {
        $(this).alert('close');
    });
</script>
</body>
</html>
```