CLASS ROUTINE DESIGN SYSTEM AND EXAM HALL INVIGILATION MANAGEMENT

RATUL PROSAD

MASTER OF ENGINEERING IN INFORMATION AND COMMUNICATION ENGINEERING

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

CLASS ROUTINE DESIGN SYSTEM AND EXAM HALL INVIGILATION MANAGEMENT

RATUL PROSAD

This Project Report submitted in partial fulfillment of the requirements
for the award of the degree of
Master of Engineering in Information and Communication Engineering

Department of Information and Communication Engineering
NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

MARCH 2021

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project and in our opinion, this project is adequate in terms of scope and quality for the award of the degree of Master of Engineering in Department of Information and Communication Engineering of Noakhali Science and Technology University.

_____

Name of Supervisor: DR. MD. ASHIKUR RAHMAN KHAN

Position: PROFESSOR & CHAIRMAN

Date:

_____

Name of Co-Supervisor: ZAYED-US-SALEHIN

Position: ASSISTANT PROFESSOR

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this project is my own except for quotations and summaries which have been duly acknowledged. This project report has not been accepted for any degree and is not concurrently submitted for the award of other degrees.

_____

Name: RATUL PROSAD

ID Number: ASH1811ME121M

Date:

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement, and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a ME program is only the start of a life-long learning experience. I appreciate his consistent support from the first day I applied to the graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my naive mistakes, and his commitment to my future carrier. I also would like to express very special thanks to my co-supervisor Zayed-Us-Salehin for his suggestions and co-operations throughout the study. I also sincerely thanks you for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to all my lab mates and members of the staff of the Information and Communication Engineering Department, NSTU, who helped me in many ways and made my stay at NSTU pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream, and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

This Project name is “Class Routine Design System and Exam Hall Invigilation Management”. This is a web based application. In Class Routine Design System mainly, focusses on creating automatic class routines with teacher’s requirements. Most universities handle this allocation process with a manual procedure. The manual procedure like this gives way to various challenges and is inclined to mistakes. A better approach to reliably schedule class routine is to utilize a computer assisted web-based system. This project has mainly two parts or two panel, Admin panel & User panel. In admin panel we get some information like courses information, teacher’s information, room’s information etc. We can Update, Delete & Add this information’s. There have teacher’s requirement for creating class routine and semester wise course offer distribution for teachers. We can update and view all this field in admin panel. After that we made a class routine using all of this field. In another panel is user panel. In this part we get all of information about course, teacher’s and rooms. In this panel we can see all this information & routine. Update, add and delete option in this panel also included. Both of this panel, we get routine separate by days, teacher’s and semester wise. We create this application for genetic algorithm, python. We create this application for reduce the complexity of routine creation manually, to reduce time to solve a problem and to reduce routine version. In Exam Hall Invigilation Management, proposes an improved algorithm to achieve automatic examination arrangement for invigilator based on greedy method, the algorithm can support the priorities better, set maximum frequency of invigilating and satisfy other personalization examination arrangement requirements, and also it can configure to allocate any numbers of invigilators in different examination halls in such a way that each invigilator will get equal amount of duties. This algorithm has written and implemented in java script language.

Keywords: Routine Allocation System, Web Based Application, Invigilator, Examination.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NP	Non-deterministic Polynomial
GA	Genetic Algorithm
CSS	Cascading Style Sheet
PMX	Partially Matched Crossover
OX	Order Crossover
SDLC	Software Development Life Cycle
UCTP	University Course Timetabling Problem
CSCL	Computer Supported Collaborative Learning
SA	Simulated Annealing
TS	Tabu Search
ER	Entity Relationship
LSA	Local Search Algorithm
API	Application Programming Interface
AGSS	Auto Generate System based on expert System
PSO	Particle Swam Optimization
SDLC	Software Development Life Cycle
GAA	Greedy Approach Algorithm
DB	Database
DBMS	Database Management System
ACS	Ant Colony Search
FL	Fuzzy Logic

UML	Unified Modeling Language
OMG	Object Management Group
HTML	Hyper Text Markup Language
HDD	Hard Disk Drive
MTV	Model Template View
DSF	Django Software Foundation
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Fifth-generation computers work with Artificial Intelligence methods for decision-making optimization and many other uses. Now that technology has evolved really quickly for a few days, it's having a significant effect on our everyday lives. The class routine allocation and examination hall surveillance project is a software application and an online network to build an automatic framework with the requirements of teachers and students. It is an information-oriented medium. Creating such platforms is very useful for teachers to check for the ID of the instructor, their classification, subjects, and students with this class, semester, and class time per week. So, we will describe here the term such as: motivation of the study, problem statement, objectives, scope and significance of project.

The most common issue in instructive founded is the timetable issue. It is considered as Non-deterministic Polynomial (NP) issue. There are a diverse number of timetable issues within the writing based on the sort of established. There are three primary classifications of timetable displayed as takes after:

Subject timetable: It is concerned with the planning for the speaker per semester at the college. It considers around minimizing the confliction of teacher classes and common understudy gather.

School timetable: It is concerned with the week by week lesson planning and dodges any instructor to meet more than one course at the same time.

Examination timetable: It is concerned with the planning of the exams for the college subjects and maintains a strategic distance from covering of subjects.

Greatest College employments manual way to plan timetable at that point enters it into the E - enroll framework since, this manual prepare is disappointing and time expending, and there was a prerequisite to apply cutting edge computing methods to naturally create the timetable. In this paper, Hereditary Calculation has been utilized for the examination and optimization of schedule planning framework. Each instructive institution in Bangladesh have schedules for the administration of classes and time distributed to them. As of now these schedules are physically made by a specific office within the institution. The parameters like teachers' time, number of periods, lesson time, etc. for making schedules are too physically kept in intellect whereas making these schedules. Due to the nonattendance of such mechanized framework for making and keeping up the course schedules, Schedule Allotment Framework is required to be designed.

1.2 MOTIVATION OF THE STUDY

The Reason of Creating this computer program extend is to completely computerize. This computer program is competent of appearing room, time, instructor title, subject title of these schedule. That's exceptionally accommodating for upkeep course plan making. All the information such as room, time, instructor title, subject title will be put away as a track record. The framework will be able to appear the upgraded schedule naturally .At whatever point the schedule overhauls, at that point the past adaptation consequently will be spared in database and the modern form will be online.

The last examination understudies of invigilator framework incorporates to some staff to be arranged of invigilator framework obligations that takes after by scholarly administration division. Invigilation's obligations of all staff attempt to spread by the scholarly administration division. Invigilation obligations are a set of necessity they have. Examination in invigilator framework is planning for challenging the assignment has continuously been in it. The various components of the invigilator framework is getting for the invigilation committee thought. The invigilator

framework such as teachers do checking guaranteeing for duties doled out and their own subjects don't invigilate for teachers. Within the writing survey, the issue of the paper examination timetable is talking about and the issue is found. The instructive examination of timetabling issue such as the room and timeslot of framework exams does not wrap up here. The exam time's part of the room timetable has produced for the institution regularly after. Issue of the invigilator is no datasets are accessible as due to the reality from the investigate community. It is less consideration. No dataset are accessible is dataset contains in this invigilator for modeled some time recently and talked about. Invigilator framework is noteworthy of able minimizing within the advancement. The invigilation plans have time went through and association on the preparation based on committee. Within the maker of handle an invigilation plans such as unwavering quality, ensures consistency and coherence for long term semesters. The Ultimate Examination Unit of the administration of any institution has full control over the ultimate examination to be able based in Scholarly Administration Division. In examination of operation room is the modifications of the number will be incredibly decreased the invigilators of last utilizing list. The imperatives of optimized for the speakers inclinations with made of the number and controlled of the information. Moreover, the framework could be a standalone creates by utilizing a few of program such as Java Script, Django web framework.

1.3 PROBLEM STATEMENT

Most instructive teach physically make their schedules by employing a human asset to work on it. A few educate make such administration framework for their individual prerequisite and utilize as it were whereas others do it physically. Routine Allocation Framework will offer assistance instructive educate to make and keep up the schedules in a more compelling and productive way.

Invigilator framework for examination is challenging errand due to a number of components that cannot be maintained a strategic distance from. Variables of a number is cannot to be maintained a strategic distance from amid due the errand of challenging. The think this work is burden them. Another reason is the number of invigilation's obligations between the crevices between isn't fulfilled. Invigilator framework is the moment issues articulations of any examination. Amid the draft

planning for timeslot of the one room or more than are doled out in this issues. The invigilators will be a lot of imperatives time and room that got to be watched invigilators out some time recently the final draft. The invigilators is a relegating to examination rooms for included the third problems. This issue has received because it less consideration than examination timetabling issue. The greatest number of invigilation obligations based on invigilators surpass is the another issue explanation in invigilator framework. At slightest two or three day hole ought to be in planning invigilator allude to proposal from them. The final step is to organize invigilator, as figure 1.1 shows.

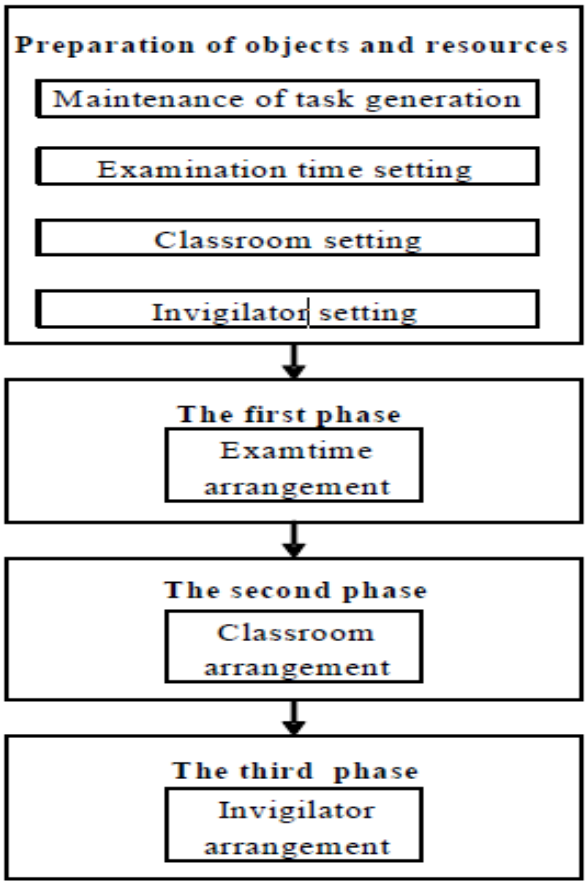

Figure 1.1: Flow Chart of Invigilation Scheduling.

1.4 OBJECTIVES

Physically Schedule creating such as class routine and exam hall invigilation management is so much challenging. Since we confront the numerous issue in this framework. We are reaching to fathom it by programmed schedule allotment framework:

i. To Diminish the Complexity of Schedule Creation Manually
ii. To create adaptable and intelligently invigilation planning framework that empowers programmed era of examinations plans.
iii. To Make a User-Friendly Interface
iv. To Decrease Time to Unravel a Problem
v. To create a straightforward easy-to-use web based application which a person with small or no thought of program application can associated with an awesome bargain of comprehension.
vi. To create a web based system that permits clients to form and keep up the course schedules

1.5 SCOPE

Lesson Schedule Allotment and Exam Lobby Invigilation Administration Framework benefits all the instructive teach that right now employments manual creation and support of the lesson schedules and exam corridor. Clients can effectively make and keep up lesson schedules and exam corridor utilizing the system's basic client interface. Within the scholarly administration division must as it were one client can control, screen and start the invigilator framework such as examination timetable. It too settle the issue of manual staffing handle into a controlled and closely checked framework. It can be utilized by every educational teach interested to make class routines employing a computerized framework. The framework may be a standalone creates by utilizing the few of computer program such as Python, Java Script, and Django web framework.

1.6 PROJECT SIGNIFICANCE

Schedules and exam corridors concur understudies to quick wrap up common put errands that are required of both the instructor and understudies. There are colossal benefits of our extend framework and benefits get both understudies and instructors. Benefits of this venture is that a look alternative for understudies and instructors, course cancel choice for instructors, additional look alternative for purge room, recognize cover lesson, can include additional lesson, can distinguish teacher's id, their assignment. From this framework will be planning to offer assistance the scholastic administration division within the portion isolation of obligations in invigilator framework. This result of this framework will be given to the invigilator are fulfilled. In this way, this will move forward the invigilator are cheerful in their obligations. At last have a report print choice that understudy or educator can print the record. The framework recipients are understudies and teachers.

1.7 CONCLUSION

Lastly, in this chapter, we discussed how we can develop an application by writing the topics such as: project purpose, significance, system description and so on.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Auto generation system is very important to maintain a smart time table. So, we discussed about our plan in the chapter by gathering, collecting and studying more papers. Classroom technologies, definition scheduling invigilators, timetable invigilator is very important topic for our literature review. A good automated timetabling survey is given in [1]. Sandhu [2] gives in his thesis a strong literature survey on timetabling techniques. In timetabling generations, he presented different techniques and algorithms used.

GA is one of the most common optimal scheduling solutions [3, 4]. GA is a natural selection and evolution-based heuristic algorithm [5]. GA is a technique of search and optimization inspired by nature. Class Routine Allocation system Control is applied. Ehsan, Zahra, and Mohammad [6] analyzed routine scheduling as a real method at Iran's Islamic Azad University. To fulfill all the constraints of automatically producing the timetable, they used hybrid algorithms. Chinnasri et al. [7] contrasted and assessed Partially Matched Crossover (PMX), Cycle Crossover (CX) and Order Crossover (OX) operators using GA for the University Course Timetabling Problem (UCTP). At Rangsit University, Thailand, they investigate real UCTP to classify the different probability of mutation, crossover with different number of generations

By using the genetic algorithm combined with heuristic quest, Lukas et al [8] suggested a method for solving this time table problem. The genetic algorithm's function is to decide the sequence in one group of all courses to be scheduled, while the role of heuristic search is to determine the time slots used to schedule the courses. They applied their approach to timetabling at Peliata Harapan University.

The method of Exam Hall Invigilation Management includes user Id management, instructor Id, teacher identification, and assignment of rooms to invigilate [9]. The allocation of rooms during the analysis can be automated using the algorithm of greedy approach [10]. Using the greedy approach algorithm [11], the authors proposed a method for scheduling the exam hall invigilation time table. The authors here realized user registration, security authentication, test, declaration, sourcing, etc. [12]. This paper also explains automated algorithm development and addresses machine security [13].

The system will increase patient throughput and use radiological services effectively here [14]. Here are automated the various examination system tasks that are performed manually, such as time table generation, searing allotment, invigilator allotment. But the moderator chooses the dates and rooms at which the faculties are assigned during invigilator allotment [15]. The authors suggested automating the operation of the examination cell where students would not have to queue up to fill in the different forms and follow the procedures to gain access to hall tickets and different examination notices [16].

2.2 CLASSROOM TECHNOLOGIES

In developing countries, most of today's universities classrooms have (laptop) computers, smart boards and Internet facilities [17]. For example, if a teacher needs to take a note after noticing the success of a certain student while walking around the classroom, before he or she can enter the material, the teacher can have to take a mobile phone out of his or her pocket, unlock the screen, find the note-taking application icon, and add a new note [17].

The teacher has to pay attention to the touchscreen interaction during the whole process and precious time is lost to helping students. Although several research studies on the design of classroom technologies concentrate on promoting learning activities[18,19,20,21,22], only a few research explorations also concern the creation of experiences that require users to have limited attention resources (the teacher or the students). In the field of Computer-Supported Collaborative Learning, a number of associated studies on instructor routines are documented (CSCL). For example, Greiffenhagen [23] studied the work of teachers during collective learning exercises

to find out how teachers participate in the learning of pupils. Prieto et al.[24] presented a study on the 'orchestration load' of teachers (the mental load needed to promote CSCL activities in daily work) using mobile eye-tracking technology in both a laboratory and a field environment, finding that the increased load for teachers may be difficult for CSCL practice to facilitate and control CSCL activities [24].

In the fields of educational science and sociology, other similar research on the routines of teachers are documented. Some research examine the routines of teachers out of class. For example, in 'proactive teaching,' which takes place before or after class while the teacher is alone, Yinger [25] has studied the routines of teachers. Different fields, such as School Administration, Artificial Intelligence, Mathematics or Operational Research [26], can assert the resolution of the timetable problems just called class routine or examination hall invigilation. Perhaps we need to focus on simulation techniques imported from fields as diverse as physics or biology to solve the problem [27]. In a given time period, an inspection takes place, using a set of rooms and a set of watchmen [28]. For a given space, each invigilator is allocated, although the same space may be supervised by many invigilators.

The primary objective of the issue of examination timetabling is that no student takes more than one examination at any time. This dispute can be perceived as a challenging limitation and must be eliminated [28].

2.3 TIMETABLE GENERATION ALGORITHMS

There are numerous timetabling and scheduling issues studied by other researchers to address scheduling issues such as class routine or examination hall. Several researchers have suggested different approaches to address the issue of timetabling and scheduling.

Constraint Based Methods: In order to satisfy hard and soft constraints, a number of different variables and values such as halls, subjects, staff and students have to be allocated while hard constraints are given priority. We may infer that the purpose of the above approach is to minimize the violation of soft constraints and to fulfill the difficult constraints [29].

Sequential Methods: In this approach, the problem of timetabling is treated as graph issues and then the events are ordered using domain-specific heuristics and then the events are sequentially allocated to valid time slots such that no constraints for each time slot are violated [29].

Cluster Methods: The group in which the problem is split into a number of events/groups is specified so that hard constraints and soft constraints are met [29].

Metaheuristic Methods: Stimulated Annealing (SA), Tabu Search (TS) and Local Search Algorithm (LSA) [29].

LSA simply checks if the new solution is better than the one already established. Using a neighbor generation process, it iterates a number of times, generating a solution based on the already existing one [29].

TS This algorithm maintains a list of Tabu (a list of moves that are not permitted). To reduce the minimum time spent on a local basis. It is added to the list after each move to stop the algorithm going back. After spending a tenure, Tabu Quest is a shot term memory, the transfer is removed from the list to ensure that the Tabu list will not impede the optimal solution [29].

2.4 AUTO GENERATE SYSTEM BASED ON EXPERT SYSTEM

In this paper, we thought that the Auto Generate System Based on Expert System (AGSS) gave the timetabling committee the opportunity to organize the details. To obtain fast computational capabilities, we have used various optimization methods, such as Particle Swam Optimization (PSO), Genetic Algorithm (GA), Greedy Approach Algorithm (GAA), Tabu Search (TS), Ant Colony System (ACS), Fuzzy Logic (FL) and Stimulated Annealing (SA). The expert method is used by considering the ability to mimic the extraction and integration of data between expertise and human experts [30].

2.5 DEFINITION SCHEDULING INVIGILATORS

The primary issue in a series of events is the surveillance method or timetabling. Date, time, place, Id staff and subjects are an example of this issue. No individual or resources are reported at the same time and one or more than the location is expected for [31]. System form is the goal of optimizing the total and minimizing the output period for a given set according to customer demands. The process of assigning tasks to resources from a collection is the concept taken from the website by scheduling invigilators [31]. For example, the times and rooms would be allocated to several numbers of vigilantes on the inspection schedule. Normally, the academic workers would be selected as applicants for invigilate of exams. They will be responsible for regulating in the examinations, tracking invigilate another want to do. All [31] must be monitored in order to obtain the high standard of exams so that a suitable invigilator timetable is controlled.

The timetabling has four categories such as transportation framework, staff planning, instructive timetabling and sports timetabling. In conclusion, the invigilator framework must contain such as difficult and delicate limitations from one institution to another with exceptionally incredibly. The detailed of invigilators with framework are not fulfilled and make great curiously way better framework invigilation based on an overview [31]. They therefore proposed that any vigilance duty should have a gap of at least two or three days. If the timetable officer is open to any ideas for improving the existing timetable, the suggestion is motivated to provide, in addition to the original restrictions, an extra restriction [31].

2.6 ALGORITHMS

We implemented the genetic algorithm in the Class Routine Allocation Management System. It uses biology-inspired techniques that involve selection, chromosomes, crossover, and mutation to solve a given issue [32]. When implementing it in a specific programming language, GA has general steps to take, as shown in Figure 2.1. The Genetic Algorithm operators are:

Population Initialization: The success of this algorithm depends on the quality of the original population and on the size of the population [33].

Selection: The 1st chromosome with a higher fitness value is more likely than the others to be chosen for reproduction [33].

Crossover: The next step is to combine two or more chromosomes to produce new offspring after agreeing to pick the parent chromosomes to be selected [34].

Mutation: after crossover to produce new offspring, mutation can be described as small as well as random alteration in an individual [35]. The mutation is key to the convergence of a genetic algorithm, as opposed to crossover [35].

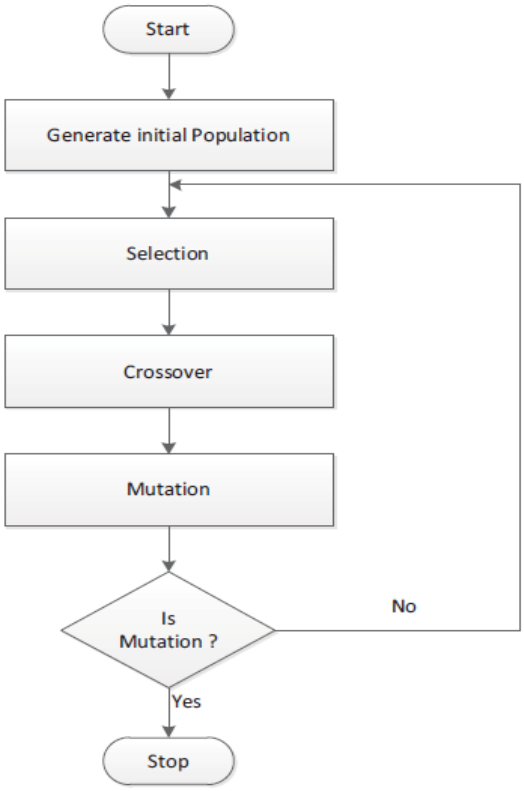

Figure 2.1: Working of a Genetic Algorithm.

A population of candidate solutions called chromosomes will be generated by the algorithm. This approach will be repeated until a person with a fitness value of 1 is assessed, which leads to a clash-free test schedule being produced.

$$fitness\ (i) = 1/(n+1)$$

$$where\ \ i = individual, n = number\ of\ clashes$$

The Unified Modeling Language (UML) proposed by the Object Management Group (OMG) in 1997 is considered to be the mainstream (OMG) language in 1997 and is considered to be the standard object-oriented research and design language [36].

In the Exam Hall Invigilation Management Scheme, on the other hand, we used greedy algorithms. All results are stored in the database so that they can be accessed, printed, edited or retrieved later by registered users [37], as the case may be.

2.7 CONCLUSION

Literature review means, to study a lot of research paper that similar to our project topic. We mentioned those papers and references for better study and future scope.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

Methodology is a contextual framework for research, a coherent and logical scheme based on views, beliefs, and values that guides the choices researchers or other users. It includes the theoretical review of the body of methods and principles associated with a branch of knowledge in such a way that the methodologies used from various disciplines vary depending on their historical growth. This creates a continuum of methodologies that reach through conflicting understandings of how best to understand knowledge and truth. In overarching theories and methods, this situates methodologies. In our project the schedule is divided into five stages. Issue finding, theory building, system creation, experimentation and discussion and assessment are the flows of the phases.

3.2 MODEL

In a production method, some data and information needs to be obtained from users in order to create a system. It is a very essential part of the Software Development Life Cycle (SDLC). We can easily understand what details or specifications we need to construct a class routine from this requirement collection component, such as hardware requirements, software requirements, etc. And we get simple ideas as well as a functional specification of a device from this research section. The main aim of our system is to establish a timetable. In our project we follow the agile model.

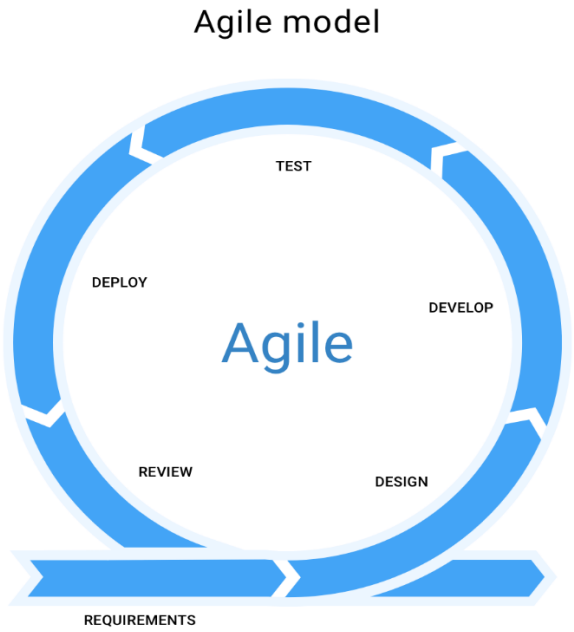

Figure 3.1: Agile Model.

When all stages of production are complete, we review the progress made towards the completion of the specifications. We present our suggestions for solving the problems that occurred during the previous stages and take our solutions into account. Afterwards, a new phase of the agile software development lifecycle begins, either with a new iteration or by progressing to the next phase. For all our programs, we use agile paradigms and consistently produce excellent results.

3.3 DESIGN

One of the components of the Django web application is architecture. We use architecture to grasp the whole process quickly. We use both the front end and the back end in our system. In front-end design we use HTML, CSS and Java Script for building a user friendly interface. To easily understand our process and easily find user requirements, we built a navigation bar. We use the Python programming language and Django web platform in the back-end portion for a user-friendly interface. We try to keep our design simple and easy to use.

3.3.1 Front End Design

Markup language standardized automated management framework. We gather a specific vision from our nearest customer for a simple graphical user interface and their understanding when we develop our web application. We gave Bellow the name of our language used and its explanation.

i. HTML
ii. CSS
iii. Java Script

HTML

i. HTML stands for Hyper Text Markup Language.
ii. HTML describes the layout of Web pages by using markup.
iii. The construction blocks of HTML pages are HTML elements.
iv. The tags reflect the HTML elements.
v. HTML tags mark content sections, such as "heading" "paragraph" "table" etc.

CSS

i. Simply change the style to make a global change, and all the elements on all the web pages will be modified automatically.
ii. For multiple HTML pages, we can write CSS once and then reuse the same layer. For each HTML feature, we can define a style and add it to as many Web pages as we like.
iii. HTML attributes are being discontinued now, and using CSS is preferred. It's a smart idea to start using CSS to make it compliant with future browsers on all HTML sites.
iv. CSS defines how HTML elements are to be shown on a computer, document, or in other media.
v. CSS is used in a clear and simple way to monitor the layout of a web document.

JavaScript

i. JavaScript is an interpreted, lightweight programming language.
ii. It is intended for the creation of network-centric applications.
iii. It works in tandem with and complements Java.
iv. Since JavaScript is compatible with HTML, it is very simple to use.
v. It's free to use and cross-platform.
vi. JavaScript is now used in game development, mobile app development, and desktop app development.

3.3.2 Back End Design

The back-end is the most important aspect of our program. Two algorithms are used to build the final scheduling system, which offers a special automatic management system. We use the Django database for web application storage and proofing of results. It is given bellow as follows:

a. Python
b. Django

Python

i. Python has a small number of keywords, a straightforward structure, and a well-defined syntax.
ii. Python code is easier to read and understand because it is more clearly described.
iii. On UNIX, Windows, and Macintosh, the majority of Python's library is very portable and cross-platform compatible.
iv. Python has an interactive mode that enables interactive testing and debugging of code snippets.
v. Python facilitates the development and porting of graphical user interfaces to a variety of system calls, libraries, and operating systems, including Windows MFC, Macintosh, and Unix's X Window system.

Django Web Framework

i. A lightweight and standalone web server for development and testing.
ii. A form serialization and validation system that can translate between HTML forms and values suitable for storage in the database.
iii. A template system that utilizes the concept of inheritance borrowed from object-oriented programming.
iv. A caching framework that can use any of several cache methods.
v. Support for middleware classes that can intervene at various stages of request processing and carry out custom functions.
vi. An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signals.

3.4 ALGORITHMS

One of the tough NP issues is having a plan. To find the best solution, the problem can be solved using a search algorithm, but it only works in simple cases. Finding a considerably good solution will take a while with more complex inputs and criteria, or it may be impossible.

3.4.1 Genetic Algorithm

In class routine allocation system we implemented GA. This is where the genetic algorithm comes in. GA's overall approach will be divided into key modules in this section and clarified accordingly.

The genetic algorithm is an approach that can simulate the phase of evolution and its action in nature through the quest for global optimization. It implies that in a type of vector that is known as the chromosome, any single key that might be possible is instructed. A representative of a gene is any one of the elements of the vector. The entire set of chromosomes consists of a population and it is estimated on the basis of fitness function. In order to calculate the "fitness" of a chromosome, a fitness value is added. In a random way, the first populations of the genetic process are created. The genetic algorithm is implemented by the operators to build the next generation out of

the generation currently used: mutation, crossover and reproduction. The genetic algorithm omits chromosomes that have lower fitness. It also maintains those who are extremely fit. The genetic algorithm transforms a set of primary individuals into high-quality individuals and each of these individual acts as a solution to the problem that was supposed to be solved. Each one of the individuals listed is known as a chromosome. Some genes are made up of them, which are predetermined.

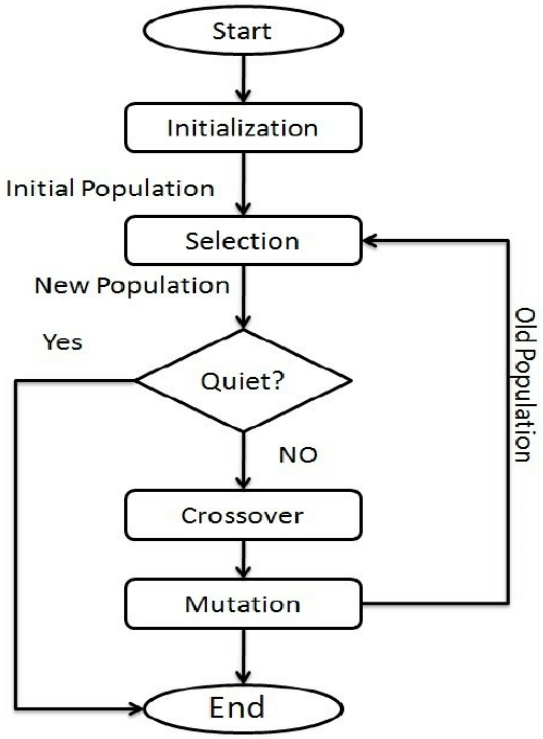

Figure 3.2: Genetic Algorithm Flowchart.

Sequentially, three key operators of genetics become conceivable to each of the individuals by the unique probabilities at the time each of the generations takes place. Figure 3.7 shows that the variables listed in the following affect the effectiveness of the algorithm: the agency of the individuals, the choice of the fitness function and the values of the GA parameters (the fitness value threshold, the population size, and mutation and crossover rate). These criteria may, by their significance, be divided into several classes.

Objects of Schedule

Professors: The professor's class has an ID and the professor's name. It also includes a list of courses that are taught by a professor.

Students Group: There is an ID and the name of the student group and the number of students in the Student Group class (size of group). It also includes a list of classes attended by the group.

Classroom: The class has an ID and the name of the classroom, as well as the number of seats and equipment details for the room (computers). If there are computers in the classroom, a computer for each seat is supposed to exist. Internally and automatically, IDs are created.

Course Code: There is an ID and the name of the course in the class.

Class: The Course Class includes a reference to the course to which the class belongs, a reference to the teacher who teaches, and a list of groups of students attending the class. It also stores how many seats in the classroom are required (sum of student groups' sizes), whether the class needs computers in the classroom, and the length of the class (in hours).

Chromosome

When we deal with a genetic algorithm, the first thing we should think is how to represent our solution in such a way that it is feasible for genetic operations like crossover and mutation. We should also understand how to specify how nice our solution is. In other words, our solution should be able to quantify its fitness value.

Representation: For a class routine schedule, we can reflect the chromosome. Well, for every hour (we assume that time is in granules of one hour), for every bed, every day, we need a slot (time-space slot). We also believe that classes should not start until 9 a.m. and should be finished before or at 5 p.m. (9 hours total), and working days are

Sunday through Thursday (5 days total). We may use a std::vector with a 12* 5* number of rooms in size. Std::list should be the slot since we allow multiple classes during the execution of our algorithm during the same time-space slot. There is an additional hash map which is used from the address of the object of the class to obtain the first time-space slot at which a class starts (its vector position). There is a different entry in the vector for each hour of a class, but in the hash map, there is only one entry per class. For example, if a class begins at 9:00 am and lasts one hour, it has entries in the slots of 1pm, 2pm, and 3pm.

Fitness: Now we need to give a chromosome fitness weight. As we said previously, to measure the fitness of a class schedule, only difficult criteria are used. Here's how we do this:

a. There can be 0 to 5 points in each class.
b. We raise the score if a class uses a spare classroom.
c. We raise the class score if a class needs computers and it is with them in the classroom, or it does not need them.
d. If a class is placed in a classroom with ample seats available, guess what, we'll increase its ranking.
e. If an instructor at the time has no other students, we raise the score of the class once again.
f. The last thing we check is if there is another class at the same time in each of the student groups that attend the class, and if they don't, we raise the class score.
g. If a class breaks a rule at every time-space slot it occupies, its score for that rule is not increased.
h. A class schedule's cumulative score is the number of points of all grades.
i. The fitness value is expressed as the maximum score and the maximum score is the number of classes.

Crossover: In order to form two new chromosomes, crossover is the fusion of genes between two chromosomes. For each row of the chromosome, crossover happens separately and crossover pairs are produced randomly for the entire population.

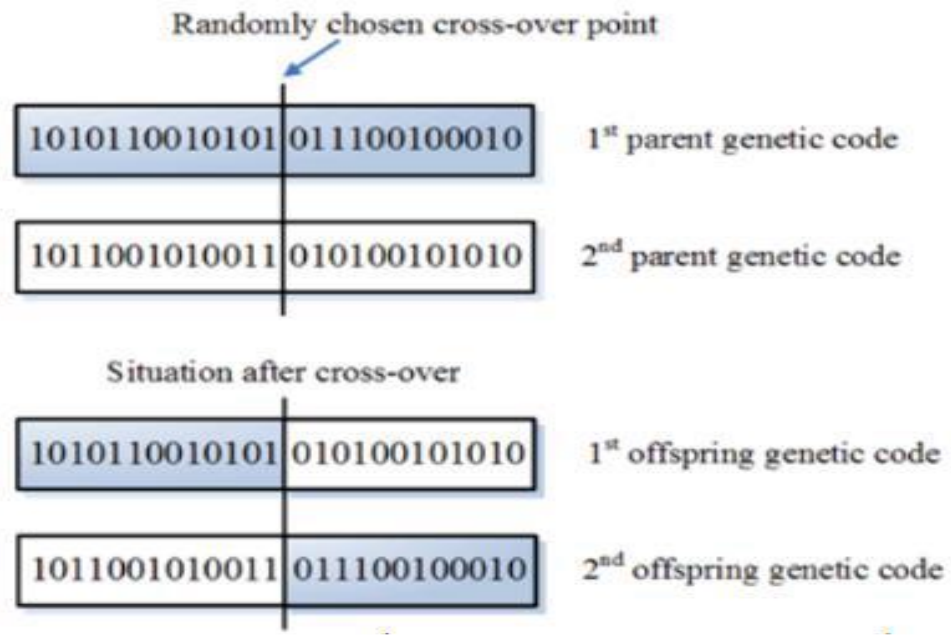

Figure 3.3: Genetic Code of the Parents and Offspring Before and After the Crossover.

In the hash maps of two parents, a crossover operation blends data and then generates a slot vector according to the content of the new hash map. In sections of random scale, a crossover 'splits' hash maps of both parents. The number of parts is determined by the number (plus one) of crossover points in the parameters of the chromosome. It then alternately copies the parent components to the new chromosome and forms a new slot vector.

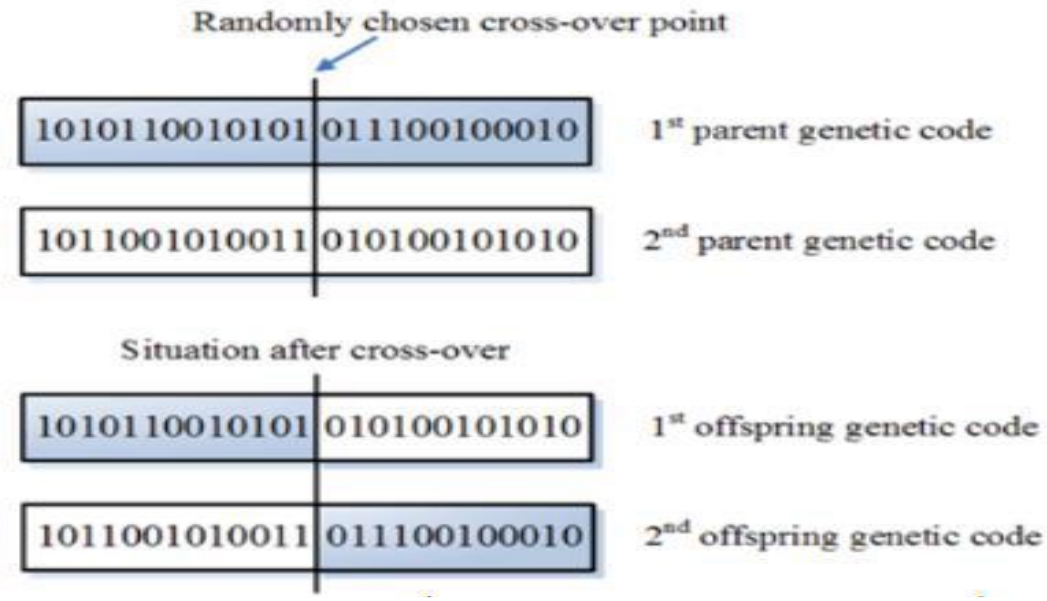

Figure 3.4: Genetic Code of the Parents and Offspring.

Mutation A mutation operation is very straightforward. It only randomly takes a class and moves it to another slot selected at random. In a single operation, the number of classes that will be transferred is determined by the mutation size in the parameters of the chromosome.

Algorithm

1. No schedule has more than one class at the same time.

$$\forall(i,j), i \neq j, \Delta \left(S_i \neq, S_{j,} \Delta\, TS_i \neq TS_j\,, where\ R_{i,} \neq R_{j,}\right)$$

$$\forall(i,j), i \neq j, \Delta \left(L_i \neq, L_{j,} \Delta\, TS_i \neq TS_j\right)$$

2. No teacher can teach more than one class at the same times.
3. No teacher can be in two classes as the same period.
4. Consider room availability and the number of the days, time slots, sessions, and register for class routine:

$$\forall i, class\ routine\ (S_i) \leq capacity\ (R_{i)},\ \ where\ class\ routine\ (y) \in i\ , y \in i, and\ capacity\ (y) \in i, y \in R)$$

3.4.2 Greedy Algorithm

The Greedy algorithm is designed to solve a given problem optimally. Decisions are taken from the specified solution domain in the greedy algorithm method. The nearest solution that seems to have an optimal solution is selected as greedy. Greedy algorithms try to find an efficient, localized solution that can ultimately lead to globally optimized solutions. The algorithm step is with the following description:

Step 1: To determine the adequacy of teacher resources.

Step 2: Check each invigilator's largest number of invigilating, whether there is too large, if it is true, it will be amended automatically to the difference of the number of the exam arrangement period permitted minus the number of examiner hours. At the same time, the total number of invigilators and the number of exam events under variety priority can be gained.

Step 3: From the row of test results list in order, if it has been completed by hand-ranked, then go to step8; if subscript overflow, indicating that exam arrangement task is completed, go to step 9.

Step 4: Get the necessary number of teachers according to the result.

Step 5: Start loop, until the num = 0, then go to step 8.

Step 6: Stating the lack of teacher resources provided by this priority, the arrange teachers whose time of be a invigilator has not yet reached its maximum in this examination task in this priority, write invigilator teacher's information, update the number of invigilating of this teachers and the set of exam arrangement times of examiner/ invigilator. However, these teachers may have the task of this test has been

examiner or examiners exam period, or with the test task, some classes of the same department, so the conflict information will be added.

Step 7: While stating that the priority to provide adequate resources for teachers, then selected num teacher randomly in this precedence, check the conflict, randomly again according to the number of teachers in conflict, if still exists conflict, then keep the last results, write information of invigilators, update the number of invigilating of this teachers and the set of exam arrangement times of examiner/ invigilator, and add the conflict information.

Step 8: i + +, go to Step 3.

Step 9: Extract the first conflict information from clash, according to the following manner.

1) Selected a teacher randomly from exam arrangement when the following occurs:
a) Conflict with other teachers;
b) The teacher has no invigilation tasks;
c) The teacher has other examination or invigilation tasks in this time.
d) The relationship between the set of this teacher's examiner / invigilator test arrangement period and the set of conflict teacher's examiner / invigilator test period is the subset of the universe.
e) Although do not conform to previous, but the set of difference between the set of extracted teacher's examiner / invigilator test arrangement period and the set of conflict teacher's examiner / invigilator test period is this teacher's examination time.

2) Find the first invigilation period in the set of difference between the set of extracted teacher's examiner / invigilator test arrangement period and the set of conflict teacher's examiner / invigilator test period, and swap with the invigilation task in conflict teacher's post-conflict period, and update the information accordingly.

Step 10: Remove the first conflict information have been processed in Clash, if the Clash is not empty, go to step 9, otherwise end the algorithm and exit.

In step 7, the method of selecting teacher randomly on the priority level is relatively simple, use random function and get a random value first, and then calculate to get the number of extracted teacher's pk (pk ∈ P), as in

$$k = randnum + \sum_{m-1}^{j-1} (Level_m\_Teachers\_Count)$$

3.5 REQUIREMENTS

An important guideline for a project is requirements. Developing clear and complete specifications, particularly for an IT-related project, may mean the difference between success and failure.The criteria are the skills, strengths, and characteristics that must be included in the final project deliverable. As more information about the project become apparent, specifications will need to be refined or created later in the project definition process. Clear conditions, on the other hand, are crucial to project progress before work starts. Confusion, duplication of effort, rework, and wasted time result from undefined criteria.

Users

The full system is mainly dependent on user requirements. The student and the teacher are consumers of our method. They have access to the final routine, day-wise routine, section-wise routine and the task of the teacher's invigilation after logging into the system. They are also available to display those details. This include information on classes, information on spaces, information on teachers and the designation of teachers. They will change this framework to post about it.

i. Login user panel.
ii. View Information.
iii. View final routine.
iv. View day wise routine.
v. View section wise routine.
vi. View semester wise routine.

vii. View session wise routine.
viii. View teacher’s Id.
ix. View teacher’s designation.
x. View teacher invigilation schedule.

Teachers

Teachers are able to reach the user panel in this scheduling scheme. But we need teacher specifications to build a schedule. So the administrator needs to complete the teacher's requirements form after logging into the admin screen. It is very necessary for a schedule to be made.

i. Sign into the admin panel.
ii. Requirement form for teacher.

Admin

The administrator is a special user of this device. All those fields are regulated or handled by them. Like data, requirements, schedule creation, etc. The admin will add, edit and delete course details, instructor information and room information after signing in. They are also set teacher’s requirements and create course offering and course distribution for teachers. All these data can be viewed and final routine, day-wise routine, section-wise routine, and teacher surveillance routine can be viewed. They also have access to the database renewal option.

i. Admin panel login.
ii. Control all data (add, edit, delete).
iii. Set standards for teachers.
iv. Handle the expectations of educators.
v. Creating the delivery of course offerings for teachers.
vi. Handle deals for course (add, edit, delete).
vii. Display all the data (courses, teachers, rooms, course offerings etc.).

viii. Display all routines (class routine and teacher's invigilation management schedule).

Hardware

Table 3.1: Hardware Requirements

Computer Processor	Intel Pentium
Hard Disk (HDD)	More than 20 GB
Ram	More than 512 MB
Motherboard	Anyone
Internet Connections	Mobile Networks or Broadband or WiFi
Monitor	Any

Software

Table 3.2: Software Requirements.

Browsers	Google Chrome, Firefox, Edge, Opera Mini, UC Browsers, Safari

3.6 DIAGRAMS AND SYSTEM OVERVIEW

Recognizing the effect, it is necessary to choose instruments, technologies and software for design and implementation phases. Research more than one potential technology in certain design decision cases and take the best performance for the best outcome.

3.6.1 Diagrams

We used here mainly three diagrams. These are: general block diagram, ER diagram, Class diagram, General block diagram.

General Block Diagram

Therefore, developing and implementing a web-based automated framework to make a timetable system simpler is inevitable. We are planning to use Genetic algorithm and Greedy algorithm for allocating and scheduling purpose. In order to meet limitations associated with scheduling, the genetic algorithm and Greedy algorithm have been used. The Iterated by Algorithm semester schedule will also be configured. The proceeding goes on:

a. Collect and interpret information obtained from the Scheduling Planner.
b. To preserve data, a Django web framework database will be developed.
c. Draw potential diagrams for various scenarios. (Flow charts, diagrams for ER diagrams, Class diagrams).
d. Through running all the test steps, the system will test: black box testing, white box testing and software testing.

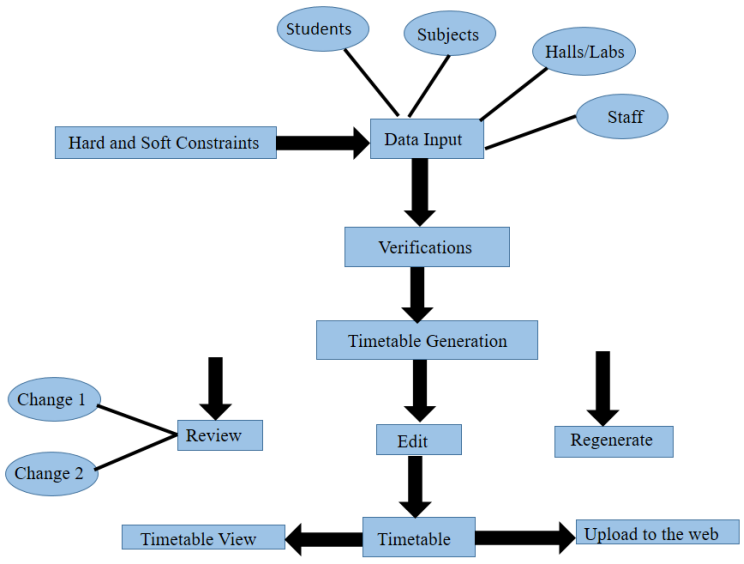

Figure 3.5: General Block Diagram.

Entity Relationship Diagram

The relationships of entity sets stored in a database are illustrated by an entity relationship diagram (ERD). An entity is an object, a data component in this context. A series of similar entities is an entity set. Such entities which have characteristics which define their properties. An ER diagram illustrates the relationship between sets of individuals. A collection of similar entities is an entity set and these entities may have attributes. In DBMS terminology, an object is a table or a table attribute in the database, so the ER diagram demonstrates the complete logical structure of a database by showing the relationship between tables and their attributes. To grasp this idea, let's take a look our ER diagram. Our ER diagram described our full project and connected all important attributes. It also described not only project system but also whole information and contents. Three basic concepts are based on this model:

i. Entities
ii. Attributes
iii. Relationships

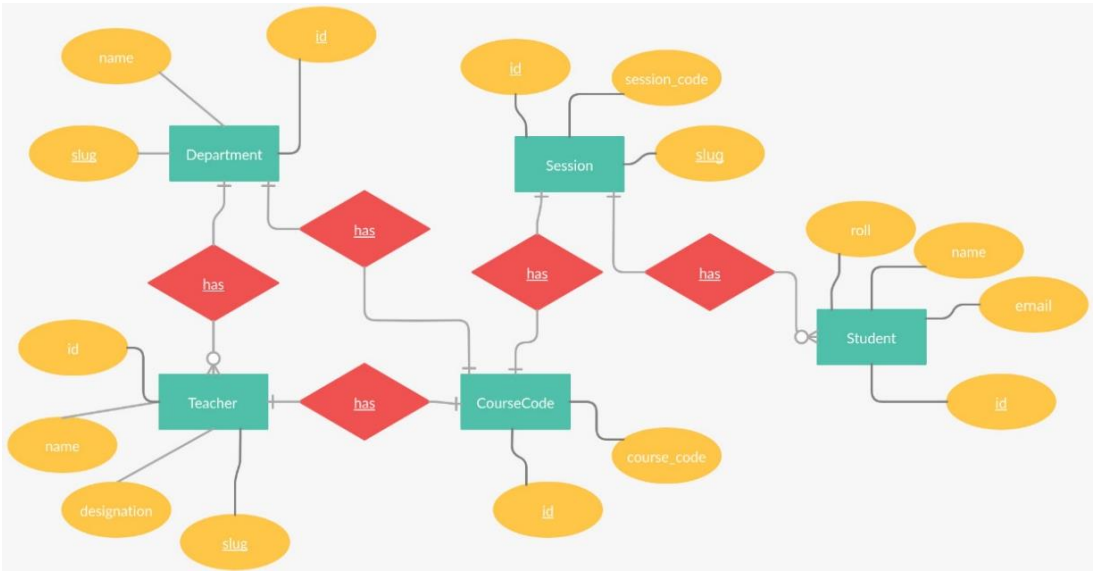

Figure 3.6: ER diagram.

The location, person, object, event or concept that stores data in the database may be an entity. An attribute and a specific key must be present in the characteristics of entities. Each entity consists of certain 'attributes' representing that entity. In the following diagram, for example, some entities are: Department, Teacher, Course Code, Session, Student. Attribute is either an entity type or a relation type single valued property. In this diagram, entities has some several attributes name, id, designation, session _code, email. Here are some geometric shapes in ER diagram. These are called diamond, rectangle, ellipse, lines.

Class Diagram

A class diagram in the Unified Modeling Language (UML) in software engineering is a type of static structure diagram that defines a system's structure by showing the classes of the system, their attributes, operations (or methods), and object relationships. A class diagram consisted of:

i. Class name
ii. A set of classes
iii. Attributes
iv. Operations
v. A set of relationships between classes

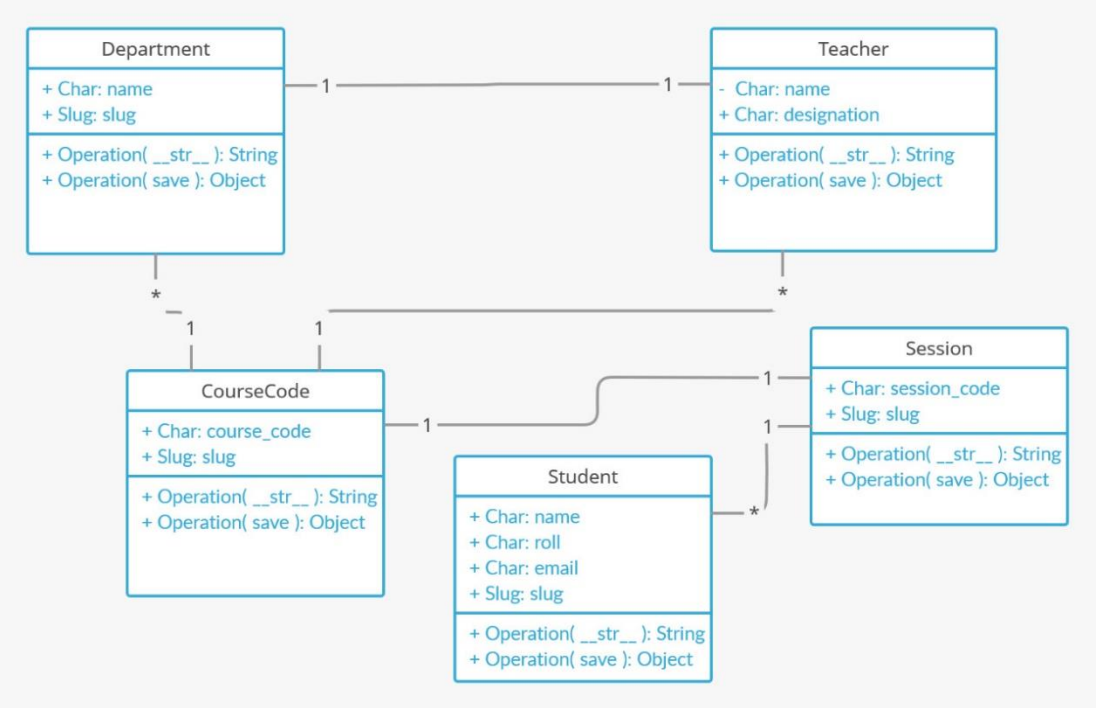

Figure 3.7: Class Diagram.

A class is an object's blueprint that can share the same relationships, attributes, operations, & semantics. A class, including its name, attributes, and operations in separate compartments, is made as a rectangle. In our class diagram, there are some classes, these are, Department, Teacher, Course Code, Student, Session. An attribute is the name of a class property that defines the object being modeled. This component is located just beneath the name-compartment in the class diagram. Such as: name, designation, email, session code.

3.6.2 System Overview

The full scheduling scheme is shown here. The designed application is a web application and must be acceptable for all users who will be using the framework. Users are able to use a uniform platform to access the program. It is, in addition, easy to manage. It will be tailored to the workload and it is also possible to improve protection. Additionally, more features are easier to add. This model displays the system architecture of the schedule system using the Greedy Approach Algorithm and Genetic Algorithm. This model is built using the agile model.

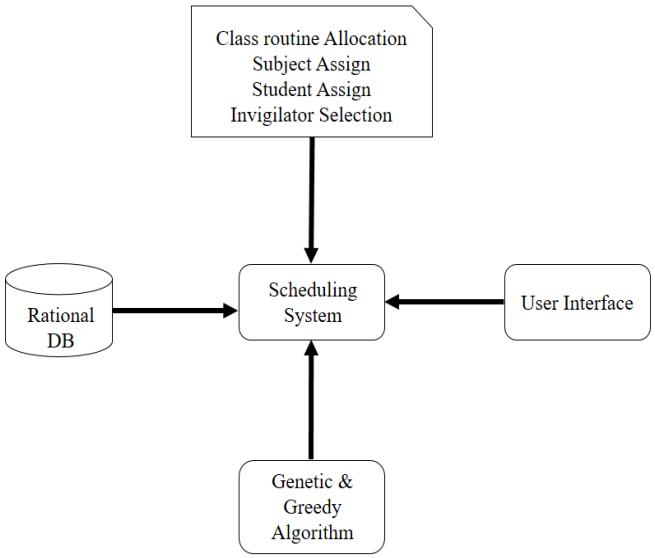

Figure 3.8: System Architecture.

Login Process

The diagram of the flow chart is generally used to identify the basic procedure. In this flow chart, we used Django web framework login process, by using email, password system. User give the input name and password. If the system is valid then, logged into the system and finally end the system. If the user name and password is not valid then it will show the error system. User authentication is an important part of most applications and can typically be seen as the mechanism that allows the system to verify the identity of user connecting to a resource on the network. In our case, our web app applies to these.

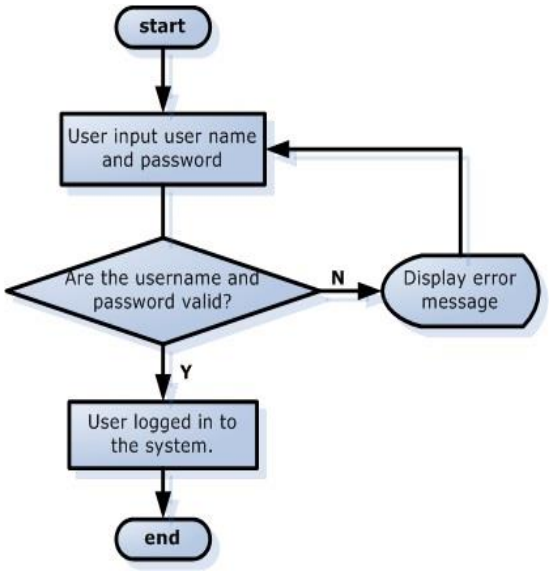

Figure 3.9: User Login Process.

3.7 IMPLEMENTATION

We have to collect various types of software or application such as text editor to render the web program, such as: Sublime Text, a browser such as: Chrome, Firefox and used to install Python executing servers such as: Django. And it is just right to be installed in our source code management software.

3.7.1 Database

Database is the management system that is used to store and distribute the data to the users on their request to save our project or system data. A database is a method for storing data. We use the Django web framework in our system to store our data.

a. DBMS Installation
b. DBMS Setup
c. Running on a machine
d. Creating Databases
e. Table Construction Database

3.7.2 Front End and Back End Design

Front-end and back-end are also in our system. There is a fine and user-friendly interface in the front-end design. Any menus and shortcut buttons or options are available to build a schedule quickly. We have a navigation bar in the front-end layout to quickly understand the process and easily construct a schedule and display the schedule. Both the user panel and the admin panel have a navigation bar. In our application, we can build a schedule for the front-end design by simply clicking a mouse.

3.7.3 Interactions

Interaction implementation is really necessary for the development of an application or website development. It is a part of web applications and websites being created. And implementation is a component of an application or system's creation. It is important to ensure that our framework or system communicates with user activities in our web application (students, teachers, administrators). It's very important for users to be interactive. We've made our application for users very interactive. In our system, we have introduced a sensitive application management system.

3.8 CONCLUSION

We now understand the design process, hardware, software, and the working algorithm of the device from the above research. The user receives first time registration information and login. The software used an algorithm-based approach to count and maximize the best degree of fitness. The entire smart and automated scheme is processed by the module.

CHAPTER 4

TESTING AND MAINTENANCE

4.1 INTRODUCTION

Testing and servicing is the study of software with user requirements and machine specifications. At the stage level in the life cycle of creating software or device level, testing is performed in the program code. In this chapter we discussed about various testing and our manual system for testing as test exercises.

4.2 TESTING

This is the process of system testing and getting results. We use testing for usability, testing of databases and testing of systems.

Usability Testing Usability testing is user interface testing. That is to assume that this is the way the device communicates with users. Finding out the interaction of the user in a device and handling versatility in control is the main focus of usability research.

Database Testing Database verification is a test to ensure the accuracy and quality of information in a system. In our database, both front-end and back-end access is possible. There is the risk of storing incorrect data that our application is designed to run using the internet.

System Testing System testing is the testing of an application's entire system. It is internal application work. In this system test, we find out whether or not our application works correctly. We got the output results after testing the whole system.

4.2.1 Testing Features and Strategy

As the changes made in the system to add new functionality or adapt the current functionality, a software function can be strong. All characteristics are supposed to have features designed to be useful, intuitive and effective. In reality, for testing that feature consistent with the cycle of that announcement, a new test set is shaped. In each construction of that release, the enormously important and usually used new features must be methodically tested and reversion testing should also be performed relevant to those areas. A test strategy is a plan which defines the software development cycle's testing method. It is designed to notify project supervisors, testers, and users of some important testing process topics. Based on development design documents, they are created.

Test Approach

The test approach is the application of a project's test plan, describing how available testing would be approved.

Black Box Testing Black box testing is also called functional testing, which ignores a system or component's internal mechanism and focuses on the outputs generated in response to selected inputs and conditions of execution. We decided to perform equivalence partitioning and Boundary partitioning for this system, value analysis.

White Box Testing White box testing is a form of software testing in which the tester is aware of the internal configuration and implementation of the object being evaluated. The tester selects inputs through the code to practice paths and decides the necessary outputs. Know-how for programming and the awareness of implementation is fundamental.

Testing Schedule This section will explain the timetable for testing.

i. Specify test signals.
ii. Stipulate proceedings transmitted entirely by article.

iii. Obligatory approximation time for each research mission.
iv. Plan activities and test signs absolutely for testing.
v. Stipulate its retro usage for each test supply.

4.2.2 Testing Environment

Testing environment is an arrangement of software and hardware for the testing teams to perform test cases. In other words, it checks the implementation of hardware, software and network configurations. The main component to be set up for the test environment consists of:

i. Framework and software
ii. Research results
iii. Server database
iv. Front end running setting
v. Client method of service
vi. Browser
vii. Hardware requires the operating system server
viii. Network
ix. Needed documentation

4.2.3 Test Cases

A test case is a set of conditions or variables under which a tester decides whether the test method meets or works properly. The method of designing test cases may also help to recognize issues with an application's specifications or design.

4.2.4 Test Arrange and Test Exercises

A better software tester by learning programming principles are discussed below:

A better understanding of the effect of code change: For black-box testers, test case design is a challenge. You need to define the collection of conditions for evaluating the program in a limited time, without understanding how the code is written. You should speak to developers and learn how the logic is applied if you have a clear understanding of programming. This will help you concentrate on the most risky areas for your practical and regression studies.

An understanding of the boundaries and layers of the application: Program is built in numerous layers – trade layer, benefit layer, database layer, etc. Knowing how code layers connected with each other can assist you plan way better integration tests.

An awareness of the release process: Developers build branches, address bugs, and merge corrections to a master branch in software release flows. You may recognize what kind of problems can occur as a result of version control if you are familiar with this method.

The capacity to talk developer's dialect: Specialized dialect is the dialect engineer's talk, and knowing the wording can increment the quality of communication with designers.

Gain developers believe and pick up proficient notoriety: On the off chance that you talk developers' dialect, they will have more regard for you as a talented analyzer. This may open them up to share with you their concerns approximately the code, unsafe spots, and refactoring thoughts – something they might not have done already, considering you'd not get it sufficient.

Automating functional tests: Automating functional tests is a common use of coding for testers. Like Selenium Web Driver, which is the most common framework, most

of the automation frameworks require you to write code. Writing test automation that is maintainable, stable, and accurate is not a simple job, as automation experts say. So, the stronger your programming abilities, the greater your chances of creating solid automation.

Handling errands that aren't viable to total physically: Separated from computerizing utilitarian scenarios, testing applications may moreover require dealing with different errands which are not down to earth or now and then not indeed conceivable to handle physically. There are a couple of examples here:

i. Generate data from tests.
ii. Read DB, Flat Files, or XML data.
iii. Log file parse.
iv. Extract and review data about results.
v. Configure the test environment.
vi. Monitor the status of the application.

This is our manual work procedure. We have considered five departments for our project. As a sample, below tables are given:

Table 4.1: Manual Class Routine.

Day	Year, Term	9.00-10.15	10.20-11.35	11.40-12.55		1.40-2.55PM	3.00-4.00PM	4.00-5.00PM
Sunday	Y-4, T-1		ICE-4105 TZK 812	ICE-5117 MARK 812	BREAK	ICE-4101 MARK 812		
	Y-3,T-1		ICE 3105 MMR 810	ICE-3109 SJS 810		ICE-3107 TZK 810		
	Y-2, T-1		ICE-2104 LAB (ZUS) 808			ICE-2106 LAB(NN) 808		
	Y-1,T-1		PHY-1103 MBH 809	ICE-1105 NN 809		ENG 1109 809	ICE-1101 MU 809	
Monday	Y-4, T-1	ICE-4104 LAB (SJS) 812	ICE-4107 ZUS 812			ICE-4103 SJS 812		
	Y-3, T-1	ICE-3104 LAB(AA +TZK) 813		MATH 3113 HMSA 812		ICE 3108 LAB AA+TZK 808		
	Y-2, T-1	ICE-2109 MBH 810	ICE-2101 AA 810	MATH 2113 SJ 810		ICE-2105 NN 810	ICE-2105 NN 810	
	Y-1, T-1	ICE-1101 MU 809	ICE-1105 NN 809	HUM-1105 MMIS 809		PHY 1103 TZK 809	ICE-1103 AA 809	
Tuesday	Y-4, T-1		ICE-4109 TZK 812	ICE-4105 AA 812		ICE-5117 ARK	ICE-4106 LAB(TZ	

							K+AA) 813	
	Y-3, T-1	ICE-3106 LAB (MMR+ NN) 808				ICE 3103 TZK 810	ICE-3109 SJS 810	ICE 3105 NN 810
	Y-2, T-1	ICE-2107 SJS 810	ICE-2103 ZUS 810	ICE-2109 MBH 810				
	Y-1, T-1	ICE-1103 MBH 809	MATH-1111 MMR 809	HUM-1105 DYS 809		BANGLA MRD 809		
Wednesday	Y-4,T-1		ICE-4101 MARK 812	ICE-4107 ZUS 812		ICE-5117 MARK 812		
	Y-3,T-1		ICE-3101 MU 810	ICE-3101 MMR 810		ICE-3107 AA 810		
	Y-2,T-1		ICE-2107 SJS 809			ICE-2101 MBH 809	ICE-2110 LAB (MBH) 813	
				ICE-1104 LAB (MBH+A A) 813			ICE-1108 LAB MARK 808	
Thursday	T-4,T-1	ICE-4108 LAB (ZUS) 813	EEE-2103 MARK					
	Y-3, T-1	ICE-3102 LAB	MATH 3113 [10.00-12.00] HMSA 812			ICE-3103		

		(MMR+ MU) 812					
	Y-2, T-1		MATH-2113 [11.00-1.00] SJ 810		ICE-2103 ZUS 810	ICE-2102 LAB (AA+MBH) 813	
	Y-1, T-1		MATH-1111 MMR 809	ENG-1109 809	BANGLA MRD 809	ICE-1102 LAB (MU) 808	

Table 4.2: Term Final Exam Routine.

Session: 2017-18 (Y-3, T-1)
2018-19 (Y-2, T-1)
2019-20 (Y-1, T-1)

Date	Course Code (10 AM-2 PM)
14.03.21	MATH-3113
15.03.21	ICE-1101
18.03.21	MATH-2113
21.03.21	ICE-3109, ICE-1105
23.03.21	ICE-2101
25.03.21	ICE-3107, ENG-1109
29.03.21	ICE-2109
31.03.21	ICE-1103, ICE-3105
05.04.21	ICE-2105
06.04.21	ICE-3103
07.04.21	MATH-1111
11.04.21	ICE-2103
13.04.21	PHY-1103, ICE-3101
15.04.21	ICE-2107
18.04.21	HUM-1105
22.04.21	BANG-1101

Table 4.3: Hall Invigilators List.

Dept.: ICE

Session: 2015-16

Term final Examination

Hall invigilators list

Date	Day	Invigilators List
13.12.2020	Sunday	Zayed-Us-Salehin, Sultana Jahan Soheli, Nishu Nath, Md. Bipul Hossain, Main Uddin.
20.12.2020	Sunday	Sultana Jahan Soheli, Apurba Adhikary, Tanvir Zaman Khan, Nishu Nath, Md. Bipul Hossain, Main Uddin,
27.12.2020	Sunday	Md. Ashikur Rahman Khan, Zayed-Us-Salehin, Sultana Jahan Soheli, Nishu Nath, Md. Bipul Hossain, Main Uddin.
03.01.2021	Sunday	Zayed-Us-Salehin, Sultana Jahan Soheli, Nishu Nath, Md. Bipul Hossain, Main Uddin.

13.12.2020	Sunday	Md. Ashikur Rahman Khan, Chief Invigilator
20.12.2020		

4.3 CODING MAINTENANCE AND DIALECT USED

When composing code we are continually making changes, including usefulness that we require or fair expelling the one we don’t require any longer. Learning a programming dialect and knowing how the calculations work can be troublesome but composing clean code is frequently indeed more complicated

4.3.1 Maintenance Plan

Corrective maintenance, adaptive maintenance, perfective maintenance and preventive maintenance can be enforced by the Schedule Management System.

Corrective Maintenance We will get the copy of the application from the server in the event of errors occurring in the application and hold it to correct the errors occurring. Then we'll replace the application on the server with the corrected application. As soon as we get informed, the errors will be corrected.

Adaptive Maintenance When the system has to adjust to the changing demand and environment, adaptive maintenance will be performed. This would be done to meet the users' evolving requirements.

Perfective Maintenance When the user asks any functionality to be installed, perfect maintenance will be maintained in the situation. This will be done to strengthen services in the system.

Preventive Maintenance Preventive maintenance will be introduced so that the device and its data will not be accessible by hackers or unauthorized users.

4.3.2 Dialect Used

Clean code is a collection of concepts that aim to render code:

a. Readable Reading
b. Maintainable Inside
c. Extendable

We need to understand the features offered by the correct language in order to write Pythonic code, features that enable us to easily understand and improve readability as well. Some of the features we can use to accomplish this are Decorators, Managers of context, Dunder approaches.

Decorators: Decorators can offer assistance us to maintain a strategic distance from duplication and partitioned rationale in little reusable pieces.

Managers of Context: Context managers help us work with those tasks that include conditions before and after a task is done, such as reading or writing files, opening them before they are processed, and closing them until they are no longer required.

Dunder approaches: Dunder Strategies or Enchantment Strategies offer assistance us to compose way better our code, improving the classes so that the code remains basic and justifiable but with higher usefulness.

4.4 CONCLUSION

A significant aspect of an errorless system is checking. We have absolutely completed all the testing stages for that. We have to carefully manage the device for a long-lasting operable module. For the right facilities, the web based device still requires extra care.

CHAPTER 5

RESULTS & DISCUSSION

5.1 INTRODUCTION

Web-based application is a secure method for schedule management system. Similar data and details have been checked. Some of the systems look interesting in the development of smart systems to be actually implemented. It is possible to further develop or merge existing systems, which helps to make the system more user-friendly, stable, and fast. For consumers, it is a less time-consuming and low-cost system that is very efficient. The screen that appears when the app opens is displayed. It is the one that is login screen in which the user has to log in to the Django site with his credentials. Originally only the admin can access the device.

5.2 USER INTERFACE

The design of the user interface is not just about buttons and menus; it is about the interaction between the user and the application or computer and, in certain ways, via that device, it is about the interaction between several users. This implies that the design of the user interface is not about how a product looks, but rather how it operates.

5.2.1 Dashboard

Upon effective login, the application's dashboard is shown where the user has a range of resources available. It can be added to the database using this feature when a new item is implemented in a department. We can easily choose the department from

the drop-down list that was given. The dashboard is seen in the picture below in the figure.

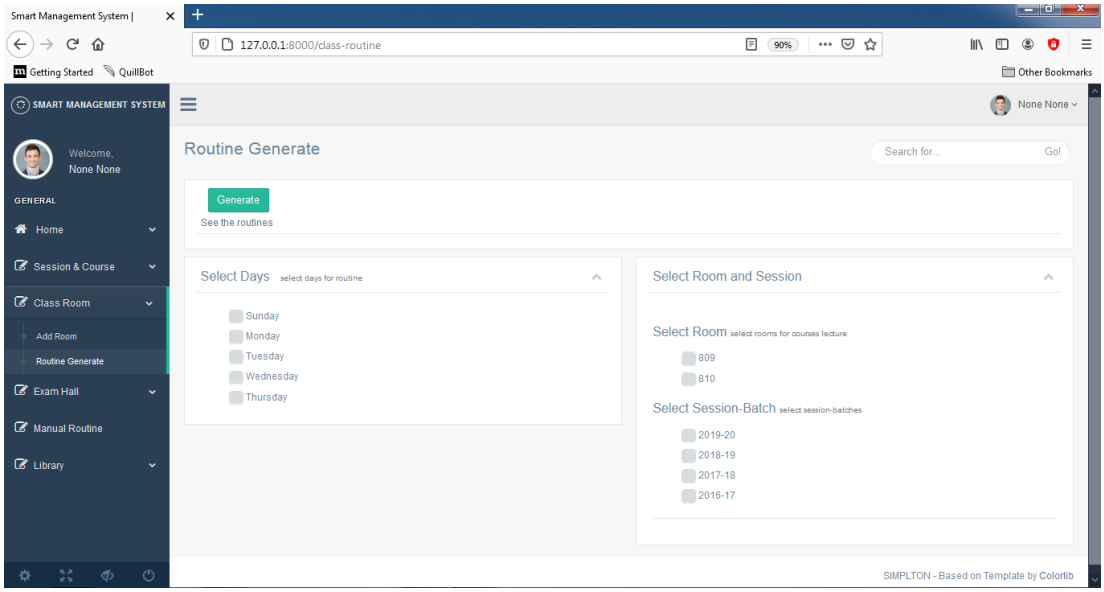

Figure 5.1: Dashboard for Class Routine Allocation System.

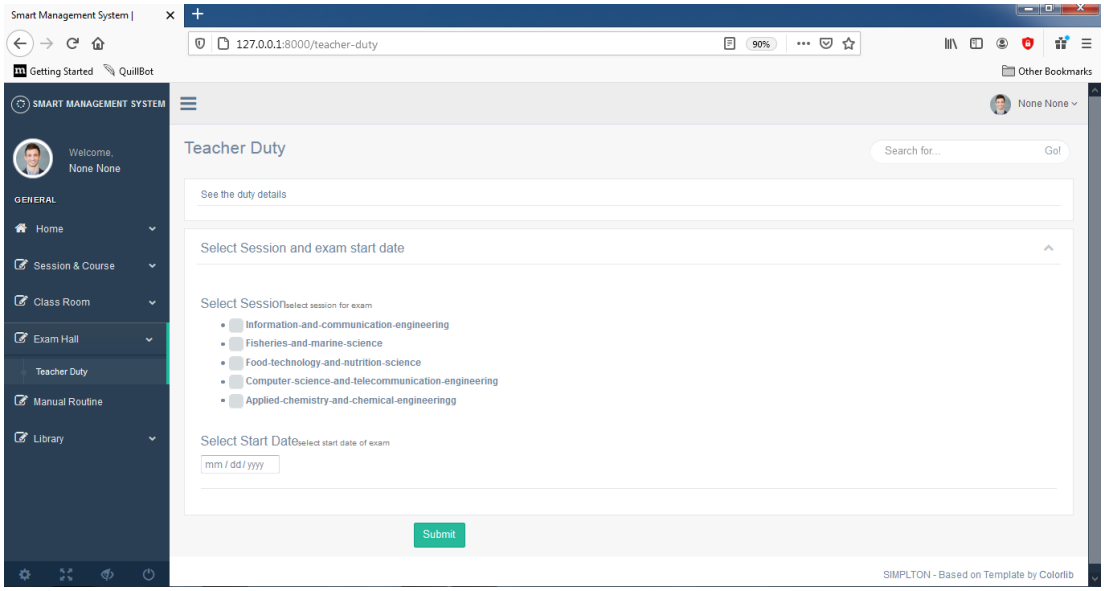

Figure 5.2: Dashboard for Automatic Exam Hall Invigilation Management System.

All the data that remains is removed from the database. For each detailed item, by a simple click on the 'Delete Selected Items' button. The rest of the terms are the

same. There is no way to recover this information, so an alert is shown before the data is erased. In order to reduce the data burden on the database, this feature has been applied to the application.

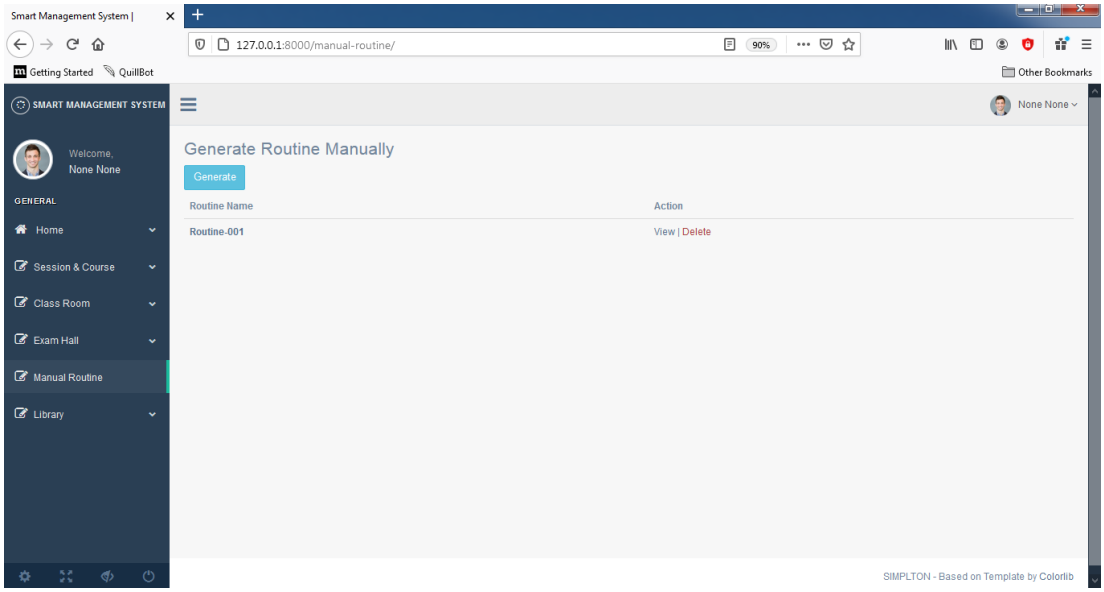

Figure 5.3: Dashboard for Manual Exam Hall Invigilation Management System.

5.2.2 Edit and Preference Items

Universities face problems with a few significant courses losing out on getting a room allocated during the manual class routine allocation and examination hall invigilation process. Therefore, an option to turn on/off preferences for a schedule has been allowed for the user. For a schedule, when the choice is switched on, it is sure to get a classroom and it will also be the first to get one. If several courses are activated for preference, a classroom is first assigned to the first one, followed by the second, third and so on in the order in which preference has been activated (row by row). In the Django web framework, how all course codes, rooms, time slots, departments, sessions, teachers are shown and how the option can be switched on or off can be clearly seen.

5.3 FINAL OUTLOOK

We can select course codes, rooms, time slots, departments, sessions, teacher's Id and view the latter by clicking on the "Generate" button provided and see the expected routine and see the expected duty details. The major components of this design are Input Design, Database Design, Table and File Design, Data Design, Output Design, process Design and Program Design. After that, the user interface ended with what we needed to shape the framework. Via various kinds of device tests, we checked our system. All of the test results are good for our device testing, and the performance is right. The findings obtained in this study are summarized as follows: automation efficiency, cost-effectiveness, human error elimination and ease of manipulation. A method that has been carried out by manual inspection and schedule generation so far. Detection by marker-controlled catchment segmentation of collisions of other scheduling slots is useful for accuracy in the automated method. In the analysis, we concentrate on the most accurate and effective technique for the problem of time tabling. Therefore, each sub-system of each algorithm needs to have an estimated degree of accuracy and performance. We may achieve a high degree of precision, but it takes time. Usage of the identification stage for the search result level categorization for our automated framework.

Generation: 28 / Fitness: 0.8506172839506173

sunday

Room	9:00	10:20	11:40	1:40	3:00	4:00
809	ICE-1105 [N.N]	BANGLA 1111 [M.R.D]	ICE-2105 [N.N]	BANGLA 1111 [M.R.D]	ICE-2107 [S.J.S]	
810	ICE 4109 [T.Z.K]		ENG-1109 [D.D]		ICE-4103 [S.J.S]	ICE-3109 [S.J.S]

monday

Room	9:00	10:20	11:40	1:40	3:00	4:00
809	ICE-2101 [M.B.H]			ICE-3105 [N.N]		MATH-2113 [S.J]
810	ICE-4107 [Z]	EEE-2103 [M.A.R.K]	ICE-4107 [Z]	PHY 1103 [T.Z.K]		ICE-4103 [S.J.S]

Figure 5.4: Automatic Class Routine System (1).

tuesday

Room	9:00	10:20	11:40	1:40	3:00	4:00
809	ICE-3107 [A.A]	ICE-1101 [M.U]	ICE-1105 [N.N]	ICE-3101 [M.M.R]	BANGLA 1111 [M.R.D]	
810	ICE-3109 [S.J.S]	ICE-2105 [N.N]	ICE-5117 [M.A.R.K]		ICE-4107 [Z]	

wednesday

Room	9:00	10:20	11:40	1:40	3:00	4:00
809	ICE-3105 [N.N]	ICE-2105 [N.N]	MATH 3113 [H.S.A]	ICE-2105 [N.N]	MATH 3113 [H.S.A]	MATH 3113 [H.S.A]
810	MATH-2113 [S.J]	ICE-2107 [S.J.S]	ICE-1105 [N.N]	PHY 1103 [T.Z.K]		EEE-2103 [M.A.R.K]

thursday

Room	9:00	10:20	11:40	1:40	3:00	4:00
809	ICE-5117 [M.A.R.K]	HUM-1105 [D.Y.S]			ICE-1103 [M.B.H]	
810	ICE-2107 [S.J.S]	ICE-3105 [N.N]	ICE-3107 [A.A]		EEE-2103 [M.A.R.K]	ICE-1103 [M.B.H]

Figure 5.5: Automatic Class Routine System (2).

Total

4-4-2021	Cheif Invigilator: 5 Invigilator: 8
8-4-2021	Cheif Invigilator: 5 Invigilator: 8
12-4-2021	Cheif Invigilator: 5 Invigilator: 9
18-4-2021	Cheif Invigilator: 5 Invigilator: 8
22-4-2021	Cheif Invigilator: 5 Invigilator: 6
26-4-2021	Cheif Invigilator: 5 Invigilator: 8
2-5-2021	Cheif Invigilator: 5 Invigilator: 8
6-5-2021	Cheif Invigilator: 5 Invigilator: 8
10-5-2021	Cheif Invigilator: 5 Invigilator: 8
16-5-2021	Cheif Invigilator: 5 Invigilator: 5
20-5-2021	Cheif Invigilator: 5 Invigilator: 6
24-5-2021	Cheif Invigilator: 5 Invigilator: 8
30-5-2021	Cheif Invigilator: 5 Invigilator: 8
3-6-2021	Cheif Invigilator: 5 Invigilator: 7
7-6-2021	Cheif Invigilator: 5 Invigilator: 6
13-6-2021	Cheif Invigilator: 5 Invigilator: 5

Figure 5.6: Counting Chief Invigilators and Invigilators.

Teacher Duty in Exam

Date	Day	Courses	Time	Chief Invigilator	Invigilator
4-4-2021	Sunday	[['ACCE 2101', 25], ['CSTE 2101', 25], ['FIMS 2101', 20], ['FTNS 2101', 20], ['ICE-2105', 25]]	2:00pm - 6:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Dr. Newaz Mohammed Bahadur', 'Dr. Mohammed Yusuf Miah', 'Dr. Humayun Kabir', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Ruhul Kabir', 'Zayed-Us-Salehin', 'Sultana Jahan Soheli']
8-4-2021	Thursday	[['ACCE 3201', 25], ['CSTE 3201', 25], ['FIMS 3201', 19], ['FTNS 3201', 20], ['ICE-3109', 25]]	9:00am - 1:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Md. Saiful Alam', 'Juganta Kumar Roy', 'Md. Javed Hossain', 'Dr. Nahid Akter', 'Dr. Mohammad Belal Hossain', 'Md. Tazul Islam', 'Apurba Adhikary', 'Tanvir Zaman Khan']
12-4-2021	Monday	[['ICE-4105', 25], ['ICE-1105', 25], ['ACCE 2103', 25], ['CSTE 2103', 25], ['FIMS 2102', 20]]	2:00pm - 6:00pm	['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar']	['Nishu Nath', 'Md. Bipul Hossain', 'Main Uddin', 'Debashismoy Dutta', 'Nahid Sultana', 'Sukanta Bhowmik', 'Abul Kalam Azad', 'Hasnat Riaz', 'Dr. Md. Anisuzzaman']

Figure 5.7: Automatic Exam Hall Invigilation System (1).

18-4-2021	Sunday	[['FTNS 2103', 20], ['ICE-2109', 25], ['ACCE 3203', 25], ['CSTE 3203', 25], ['FIMS 3203', 19]]	9:00am - 1:00pm	['Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar']	['Md. Abdullah Al Mamun', 'H.M. Shahadat Ali', 'Salma Jahan', 'Sadia Afroz', 'Sanchita Dewanjee', 'Ratnadip Kuri', 'A Q M Sala Uddin Pathan', 'Dr. Robiul Hasan']
22-4-2021	Thursday	[['FTNS 3203', 20], ['ICE-3105', 25], ['PHY 1103', 25], ['ACCE 2105', 25], ['CSTE 2105', 25]]	2:00pm - 6:00pm	['Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi']	['Mohammad Rahanur Alam', 'Md. Muhaiminul Islam Selim', 'Dilruba Yesmin Smrity', 'Monju Rani Das', 'Dr. Mohammed Yusuf Miah', 'Dr. Md. Asadun Nabi']
26-4-2021	Monday	[['FIMS 2103', 20], ['ICE-4103', 25], ['FTNS 2105', 20], ['ICE-2107', 25], ['ACCE 3205', 25]]	9:00am - 1:00pm	['Professor Md. Jahangir Sarkar', 'Md. Ashikur Rahman Khan', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur']	['Mahabubur Rahman', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Tanjina Rahman', 'Sultana Jahan Soheli', 'Apurba Adhikary', 'Md. Saiful Alam', 'Juganta Kumar Roy']
2-5-2021	Sunday	[['CSTE 3205', 25], ['FIMS 3205', 19], ['FTNS 3205', 20], ['ICE-3101', 25], ['ICE-1103', 25]]	2:00pm - 6:00pm	['Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan']	['Md. Javed Hossain', 'Dr. Nahid Akter', 'Dr. Shyamal Kumar Paul', 'Marjia Sultana', 'Tanvir Zaman Khan', 'Nishu Nath', 'Md. Bipul Hossain', 'Main Uddin']

Figure 5.8: Automatic Exam Hall Invigilation System (2).

6-5-2021	Thursday	[['ACCE 2107', 25], ['CSTE 2107', 25], ['FIMS 2104', 20], ['FTNS 2107', 20], ['ICE-2101', 25]]	9:00am - 1:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Nahid Sultana', 'Sukanta Bhowmik', 'Abul Kalam Azad', 'Hasnat Riaz', 'Tasnim Sultana', 'Towhid Hasan', 'Debashismoy Dutta', 'H.M. Shahadat Ali']
10-5-2021	Monday	[['ACCE 3207', 25], ['CSTE 3207', 25], ['ICE 4109', 25], ['FIMS 3207', 19], ['FTNS 3207', 20]]	2:00pm - 6:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun']	['Sadia Afroz', 'Sanchita Dewanjee', 'Ratnadip Kuri', 'A Q M Sala Uddin Pathan', 'Salma Jahan', 'Md. Muhaiminul Islam Selim', 'Smiriti Chakrabarty', 'Syeda Saima Alam']
16-5-2021	Sunday	[['ICE-3107', 25], ['MATH-1111', 25], ['FIMS 2105', 20], ['FTNS 2109', 20], ['MATH-2113', 25]]	9:00am - 1:00pm	['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Dilruba Yesmin Smrity', 'Monju Rani Das', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Sultana Jahan Soheli']
20-5-2021	Thursday	[['ACCE 3209', 25], ['CSTE 3209', 25], ['FIMS 3209', 19], ['FTNS 3209', 20], ['MATH 3113', 25]]	2:00pm - 6:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Dr. Mohammed Yusuf Miah', 'Dr. Md. Asadun Nabi', 'Dr. Mohammad Belal Hossain', 'Md. Tazul Islam', 'Apurba Adhikary', 'Tanvir Zaman Khan']

Figure 5.9: Automatic Exam Hall Invigilation System (3).

24-5-2021	Monday	[['ICE-4101', 25], ['HUM-1105', 25], ['FIMS 2106', 20], ['FTNS 2102', 20], ['ICE-2103', 25]]	9:00am - 1:00pm	['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Nishu Nath', 'Md. Bipul Hossain', 'Main Uddin', 'Debashismoy Dutta', 'Dr. Md. Anisuzzaman', 'Md. Abdullah Al Mamun', 'H.M. Shahadat Ali', 'Salma Jahan']
30-5-2021	Sunday	[['ACCE 3211', 25], ['CSTE 3211', 25], ['FIMS 3211', 19], ['FTNS 3202', 20], ['ICE 3103', 25]]	2:00pm - 6:00pm	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Md. Saiful Alam', 'Juganta Kumar Roy', 'Md. Javed Hossain', 'Dr. Nahid Akter', 'Dr. Robiul Hasan', 'Mohammad Rahanur Alam', 'Md. Muhaiminul Islam Selim', 'Dilruba Yesmin Smrity']
3-6-2021	Thursday	[['ENG-1109', 25], ['ICE-4107', 25], ['BANGLA 1111', 25], ['FIMS 2107', 20], ['FTNS 2104', 20]]	9:00am - 1:00pm	['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun']	['Monju Rani Das', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Sultana Jahan Soheli', 'Apurba Adhikary', 'Mahabubur Rahman', 'Tanjina Rahman']

Figure 5.10: Automatic Exam Hall Invigilation System (4).

7-6-2021	Monday	[['FTNS 3204', 20], ['FIMS 2108', 20], ['FTNS 2106', 20], ['ICE-5117', 25], ['FIMS 2109', 20]]	2:00pm - 6:00pm	['Md. Abdullah Al Mamun', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar']	['Marjia Sultana', 'Dr. Shyamal Kumar Paul', 'Towhid Hasan', 'Tanvir Zaman Khan', 'Nishu Nath', 'Tasnim Sultana']
13-6-2021	Sunday	[['FTNS 3206', 20], ['FIMS 2110', 20], ['FTNS 3208', 20], ['EEE-2103', 25], ['FTNS 3211', 20]]	9:00am - 1:00pm	['Md. Abdullah Al Mamun', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Abdullah Al Mamun']	['Syeda Saima Alam', 'Smiriti Chakrabarty', 'Md. Bipul Hossain', 'Main Uddin', 'Md. Tazul Islam']

Figure 5.11: Automatic Exam Hall Invigilation System (5).

This is our manual routine. If anyone needs to update the routine manually then, he or she can do it with manual system facility. As a sample below figures can be considered:

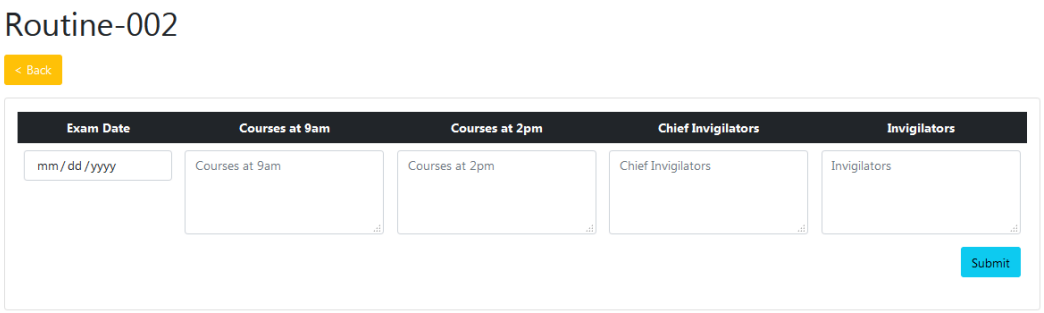

Figure 5.12: Creating Routine, Manual Exam Hall Invigilation System.

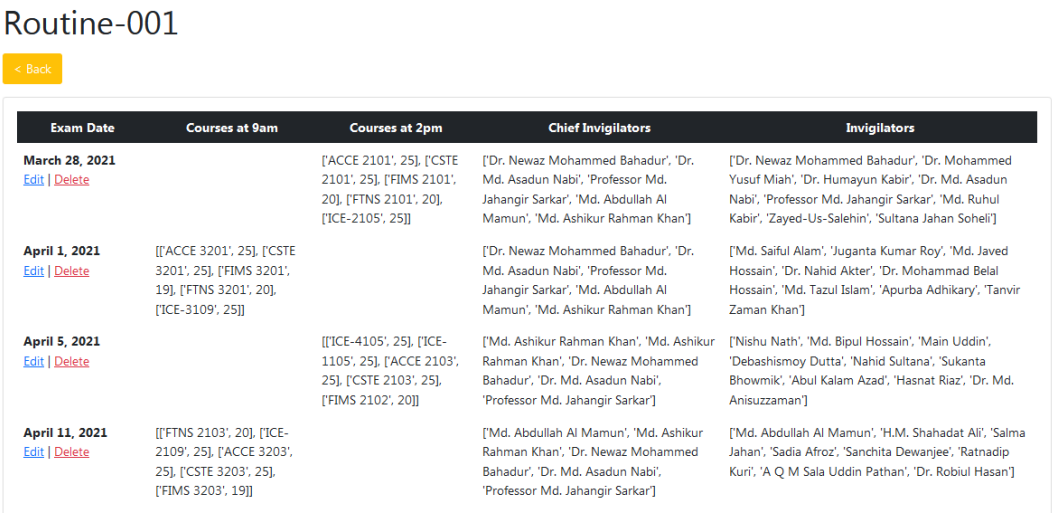

Figure 5.13: Manual Exam Hall Invigilation System (1).

April 15, 2021 Edit | Delete		[['FTNS 3203', 20], ['ICE-3105', 25], ['PHY 1103', 25], ['ACCE 2105', 25], ['CSTE 2105', 25]]	['Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi']	['Mohammad Rahanur Alam', 'Md. Muhaiminul Islam Selim', 'Dilruba Yesmin Smrity', 'Monju Rani Das', 'Dr. Mohammed Yusuf Miah', 'Dr. Md. Asadun Nabi']
April 19, 2021 Edit | Delete	[['FIMS 2103', 20], ['ICE-4103', 25], ['FTNS 2105', 20], ['ICE-2107', 25], ['ACCE 3205', 25]]		['Professor Md. Jahangir Sarkar', 'Md. Ashikur Rahman Khan', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Dr. Newaz Mohammed Bahadur']	['Mahabubur Rahman', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Tanjina Rahman', 'Sultana Jahan Soheli', 'Apurba Adhikary', 'Md. Saiful Alam', 'Juganta Kumar Roy']
April 25, 2021 Edit | Delete		[['CSTE 3205', 25], ['FIMS 3205', 19], ['FTNS 3205', 20], ['ICE-3101', 25], ['ICE-1103', 25]]	['Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan']	['Md. Javed Hossain', 'Dr. Nahid Akter', 'Dr. Shyamal Kumar Paul', 'Marjia Sultana', 'Tanvir Zaman Khan', 'Nishu Nath', 'Md. Bipul Hossain', 'Main Uddin']
April 29, 2021 Edit | Delete	[['ACCE 2107', 25], ['CSTE 2107', 25], ['FIMS 2104', 20], ['FTNS 2107', 20], ['ICE-2101', 25]]		['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Nahid Sultana', 'Sukanta Bhowmik', 'Abul Kalam Azad', 'Hasnat Riaz', 'Tasnim Sultana', 'Towhid Hasan', 'Debashismoy Dutta', 'H.M. Shahadat Ali']
May 3, 2021 Edit | Delete		[['ACCE 3207', 25], ['CSTE 3207', 25], ['ICE 4109', 25], ['FIMS 3207', 19], ['FTNS 3207', 20]]	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun']	['Sadia Afroz', 'Sanchita Dewanjee', 'Ratnadip Kuri', 'A Q M Sala Uddin Pathan', 'Salma Jahan', 'Md. Muhaiminul Islam Selim', 'Smiriti Chakrabarty', 'Syeda Saima Alam']
May 9, 2021 Edit | Delete	[['ICE-3107', 25], ['MATH-1111', 25], ['FIMS 2105', 20], ['FTNS 2109', 20], ['MATH-2113', 25]]		['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Dilruba Yesmin Smrity', 'Monju Rani Das', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Sultana Jahan Soheli']

Figure 5.14: Manual Exam Hall Invigilation System (2).

May 13, 2021 Edit | Delete		[['ACCE 3209', 25], ['CSTE 3209', 25], ['FIMS 3209', 19], ['FTNS 3209', 20], ['MATH 3113', 25]]	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Dr. Mohammed Yusuf Miah', 'Dr. Md. Asadun Nabi', 'Dr. Mohammad Belal Hossain', 'Md. Tazul Islam', 'Apurba Adhikary', 'Tanvir Zaman Khan']
May 17, 2021 Edit | Delete	[['ICE-4101', 25], ['HUM-1105', 25], ['FIMS 2106', 20], ['FTNS 2102', 20], ['ICE-2103', 25]]		['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Nishu Nath', 'Md. Bipul Hossain', 'Main Uddin', 'Debashismoy Dutta', 'Dr. Md. Anisuzzaman', 'Md. Abdullah Al Mamun', 'H.M. Shahadat Ali', 'Salma Jahan']
May 23, 2021 Edit | Delete		[['ACCE 3211', 25], ['CSTE 3211', 25], ['FIMS 3211', 19], ['FTNS 3202', 20], ['ICE 3103', 25]]	['Dr. Newaz Mohammed Bahadur', 'Dr. Md. Asadun Nabi', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan']	['Md. Saiful Alam', 'Juganta Kumar Roy', 'Md. Javed Hossain', 'Dr. Nahid Akter', 'Dr. Robiul Hasan', 'Mohammad Rahanur Alam', 'Md. Muhaiminul Islam Selim', 'Dilruba Yesmin Smrity']
May 27, 2021 Edit | Delete	[['ENG-1109', 25], ['ICE-4107', 25], ['BANGLA 1111', 25], ['FIMS 2107', 20], ['FTNS 2104', 20]]		['Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun']	['Monju Rani Das', 'Md. Masudur Rahman', 'Zayed-Us-Salehin', 'Sultana Jahan Soheli', 'Apurba Adhikary', 'Mahabubur Rahman', 'Tanjina Rahman']
May 31, 2021 Edit | Delete		[['FTNS 3204', 20], ['FIMS 2108', 20], ['FTNS 2106', 20], ['ICE-5117', 25], ['FIMS 2109', 20]]	['Md. Abdullah Al Mamun', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Professor Md. Jahangir Sarkar']	['Marjia Sultana', 'Dr. Shyamal Kumar Paul', 'Towhid Hasan', 'Tanvir Zaman Khan', 'Nishu Nath', 'Tasnim Sultana']
June 6, 2021 Edit | Delete	[['FTNS 3206', 20], ['FIMS 2110', 20], ['FTNS 3208', 20], ['EEE-2103', 25], ['FTNS 3211', 20]]		['Md. Abdullah Al Mamun', 'Professor Md. Jahangir Sarkar', 'Md. Abdullah Al Mamun', 'Md. Ashikur Rahman Khan', 'Md. Abdullah Al Mamun']	['Syeda Saima Alam', 'Smiriti Chakrabarty', 'Md. Bipul Hossain', 'Main Uddin', 'Md. Tazul Islam']

Figure 5.15: Manual Exam Hall Invigilation System (3).

5.4 CONCLUSION

Above result ensures us a good output. We made a smart output for our project. We discussed about automatic class routine allocation system, both automatic and manual exam hall invigilation system. Above we have shown all the exact outputs.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

The Scheduling System is a program that strengthens these Processes which have functions in the system may provide a database for storing documents and records. The system shown here offers a complete solution to the problem with the Class Routine Allocation method. Finally, it takes a few minutes to come up with a full solution of high quality to allocate a big improvement over a manual work. In exam hall invigilation management system, explanation about that how efficiently a greedy approach algorithm can be created. Here an algorithm is built to delegate invigilators to each examination room automatically, and all invigilators must have the same number of duties.

6.2 CONCLUSION

A continuously guided automated schedule management system to create an automatic handle and load. All such information is stored in a centralized database that can be accessed whenever appropriate. Now, this program enables users to:

a. Distribution of classrooms for most courses.
b. To adjust a schedule based on complex circumstances.
c. To display the empty schedule for the examination hall surveillance system at any given timeframe.
d. Use the classroom effectively.

6.3 SCOPE FOR FUTURE DEVELOPMENTS

The Web application is a modern technology that enables everyone to access the Internet from computers or other devices. We are introducing this software and we intend to upgrade to additional functionality someday. In our university, there is a roadmap for our modern routine to incorporate it. And add more functionality to the real life work of the student, as well. We're certainly converting it to Android users too. Some of the possible changes that can be achieved in future:

i. The routine management framework can be upgraded with the use of genetic algorithms to enable the allocation of laboratories for courses that require them.

ii. There will be an update that takes into account the personal preference and comfort of the respective faculty.

iii. For few departments or divisions or parts of any university, the greedy algorithm will be employed in the exam hall invigilation management system. So in the future, by connecting to the whole University's invigilator allocation facility, this algorithm can be updated.

REFERENCES

[1] Schaerf, A. (1999). A survey of automated timetabling. Artificial intelligence review, 13(2), 87-127.

[2] Sandhu, K. S. (2003). Automating class schedule generation in the context of a university timetabling information system (Doctoral dissertation, Griffith University).

[3] Ehrgott, M., & Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. OR-Spektrum, 22(4), 425-460.

[4] Szpigel, B. (1973). Optimal train scheduling on a single line railway.

[5] Saini, N. (2017). Review of selection methods in genetic algorithms. International Journal of Engineering and Computer Science, 6(12), 22261-22263.

[6] Alwashahi, M. (2015). Investigation and Optimization of Scheduling System in Sohar University using Genetic Algorithm (GA). International Journal of Computer Applications, 126(11).

[7] Chinnasri, W., Krootjohn, S., & Sureerattanan, N. (2012, April). Performance comparison of genetic algorithm's crossover operators on university course timetabling problem. In 2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT) (Vol. 2, pp. 781-786). IEEE.

[8] Lukas, S., Aribowo, A., & Muchri, M. (2009, August). Genetic algorithm and heuristic search for solving timetable problem case study: Universitas Pelita Harapan timetable. In 2009 Second International Conference on the Applications of Digital Information and Web Technologies (pp. 629-633). IEEE.

[9] Lu, H., & Hu, Y. (2012, August). The design and implementation of online examination system based on J2EE. In 2012 International Conference on Industrial Control and Electronics Engineering (pp. 93-95). IEEE.

[10] Indu Sharma, Anjali Singhal, Research on Online Examination System, International Journal of Engineering Technology, Management and Applied Sciences,Volume 2 Issue 3, August 2014.

[11] Manoj Kr. Mahto1 , Mr. Lokesh Kumar2 ,Exam Time Table Scheduling using Genetic Algorithm, International Journal of Enhanced Research in Management & Computer Applications, Vol. 4 Issue 8, August-2015

[12] Yong-Sheng, Z., Xiu-Mei, F., & Ai-Qin, B. (2015, November). The research and design of online examination system. In 2015 7th International Conference on Information Technology in Medicine and Education (ITME) (pp. 687-691). IEEE.

[13] Kotwal, D. V., Bhadke, S. R., Gunjal, A. S., & Biswas, P. (2016). Online examination system. international research Journal of Engineering and Technology, 3(1), 115-117.

[14] He, C. H. (2018). Tabu search based resource allocation in radiological examination process execution. Frontiers of Information Technology & Electronic Engineering, 19(3), 446-458.

[15] Yepuri, V. K., Pamu, G. C., Kodali, N., & Pradyumna, L. V. (2018). Examination Management Automation System. International Research Journal of Engineering and Technology (IRJET).

[16] Rakshit Patki1, Saranya Nair, Indira joshi, International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue: 03 , Mar 2019.

[17] An, P., Bakker, S., & Eggen, B. (2017). Understanding teachers' routines to inform classroom technology design. Education and Information Technologies, 22(4), 1347-1376.

[18] Bodén, M., Dekker, A., Viller, S., & Matthews, B. (2013, June). Augmenting play and learning in the primary classroom. In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 228-236).

[19] Charoenying, T. (2008, June). Accountable game designs for classroom learning. In Proceedings of the 7th international conference on Interaction design and children (pp. 1-5).

[20] da Silva, V. C., de Goes Jr, E. S., da Hora França, M., & Ambrósio, P. E. (2011, May). Words Game in an Educational Context: Augmented Reality Aplication. In 2011 XIII Symposium on Virtual Reality (pp. 128-133). IEEE.

[21] Sylla, C. (2013, June). Designing a tangible interface for collaborative storytelling to access' embodiment'and meaning making. In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 651-654).

[22] Zhang, Z., Shrubsole, P., & Janse, M. (2010, June). Learning environmental factors through playful interaction. In Proceedings of the 9th International Conference on Interaction Design and Children (pp. 166-173).

[23] Greiffenhagen, C. (2012). Making rounds: The routine work of the teacher during collaborative learning with computers. International Journal of Computer-Supported Collaborative Learning, 7(1), 11-42.

[24] Prieto, L. P., Sharma, K., Wen, Y., & Dillenbourg, P. (2015). The burden of facilitating collaboration: towards estimation of teacher orchestration load using eye-tracking measures. International Society of the Learning Sciences, Inc.[ISLS].

[25] Yinger, R. (1979). Routines in teacher planning. Theory into practice, 18(3), 163-169.

[26] Burke, E. K., Elliman, D. G., & Weare, R. (1993, June). Automated scheduling of university exams. In IEE Colloquium on Resource Scheduling for Large Scale Planning Systems (pp. 3-1). IET.

[27] Chunbao, Z., & Nu, T. (2012). An intelligent, interactive & efficient exam scheduling system (IIEESS v1. 0). Proceeding of the Practice and Theory of Automated Timetabling (PATAT), Norway, 437-450.

[28] Moreira, J. J. (2008). A system for automatic construction of exam timetable using genetic algorithms. Tékhne-Revista de Estudos Politécnicos, (9), 319-336.

[29] Abdelhalim, E. A., & El Khayat, G. A. (2016). A utilization-based genetic algorithm for solving the university timetabling problem (uga). Alexandria Engineering Journal, 55(2), 1395-1409.

[30] Ekanayake, T. W. INTELLIGENT TIMETABLE SHEDULER BY USING GENETIC ALGORITHM.

[31] Yusof, N. K. M. (2015). Invigilator System for UMP Examination Problem Based on Faculty of Computer System & Software Engineering (FSKKP) (Doctoral dissertation, UMP).

[32] Mc Ginley, B., Maher, J., O'Riordan, C., & Morgan, F. (2011). Maintaining healthy population diversity using adaptive crossover, mutation, and selection. IEEE Transactions on Evolutionary Computation, 15(5), 692-714.

[33] Aminu, A., Caesarendra, W., Haruna, U. S., Sani, A., Sa'id, M., Pamungkas, D. S., ... & Kurniawan, E. (2019, October). Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm. In 2019 2nd International Conference on Applied Engineering (ICAE) (pp. 1-5). IEEE.

[34] Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.

[35] Tutorialspoint 2016, Genetic Algorithms Tutorial ,2nd edition ed.,India : Tutorials Point.

[36] Norgren, E., & Jonasson, J. (2016). Investigating a Genetic Algorithm-Simulated Annealing Hybrid Applied to University Course Timetabling Problem: A Comparative Study Between Simulated Annealing Initialized with Genetic Algorithm, Genetic Algorithm and Simulated Annealing.

[37] Vasupongayya, S., Noodam, W., & Kongyong, P. (2013). Developing examination management system: senior capstone project, a case study. International Journal of Computer and Information Engineering, 7(7), 1046-1052.

APPENDICES

Coding

Class Routine Generate Form

```
# ---------- Class Routine Generate Form  ---------- #
class ClassRoutineForm(forms.Form):
        SESSION = [(str(s).lower(), str(s).title()) for s in
Session.objects.all()]
        ROOM = [ (str(r).lower(), str(r).upper())  for r in
ClassRoom.objects.filter(room_category="lecture")]

        DAYS = [
                ('sunday','Sunday'),
                ('monday','Monday'),
                ('tuesday','Tuesday'),
                ('wednesday','Wednesday'),
                ('thursday','Thursday'),
        ]

        session = forms.MultipleChoiceField(
                required=True,
                widget=forms.CheckboxSelectMultiple(
                                attrs={
                                        'type':'checkbox',
                                        'class':'flat',
                                }
                        ),
                choices=SESSION,
        )
        room = forms.MultipleChoiceField(
                required=True,
                widget = forms.CheckboxSelectMultiple(
                        attrs = {
                                'type':'checkbox',
                                'class':'flat',
                        }
                ),
                choices = ROOM,
        )

        days = forms.MultipleChoiceField(
                required=True,
                widget = forms.CheckboxSelectMultiple(
                        attrs ={
                                'type':'checkbox',
                                'class': 'flat',
                        }
                ),
                choices = DAYS,
        )

        def clean(self):
                session = self.cleaned_data.get('session')
                room = self.cleaned_data.get('room')
```

```
                days = self.cleaned_data.get('days')

        def deploy(self):
                session = self.cleaned_data.get('session')
                room = self.cleaned_data.get('room')
                days = self.cleaned_data.get('days')

                # calculating slots
                daily_class_time =
ClassTimeSlot.objects.all().count()
                total_slots = len(days)*daily_class_time

                s=[]

                # Generating all slots
                idx = 1
                # days = ['Sunday', 'Monday', 'Tuesday', 'Wednesday',
'Thursday']
                times = [ str(t) for t in
ClassTimeSlot.objects.all()]
                for day in days:
                        for time in times:
                                line = "Slot = "+str(idx)+","+
day+","+time
                                idx+=1;
                                s.append(line)

                # Generating all rooms
                total_rooms =
ClassRoom.objects.filter(Q(room_code__in=room)&
Q(room_category="lecture"))
                idx=1
                print(total_rooms)
                for r in total_rooms:
                        line = "Room = "+str(idx)+" , " +
str(r.room_code)+ " , " + "Medium"
                        idx+=1
                        s.append(line)

                # Generating all teacher
                idx=1
                teachers = list(set([ str(c.course_teacher) for c in
CourseCode.objects.filter(course_session__session_code__in=session,
department__name="Information and Communication Engineering") ]))
                hash_teachers = dict()
                for t in teachers:
                        line = "Instructor = "+str(idx)+" , " + t +
", 0 , 0"
                        hash_teachers[t]=idx
                        idx+=1
                        s.append(line)

                # Generating all courses
                idx=1
                courses = list(set([ str(c.course_code) for c in
CourseCode.objects.filter(course_session__session_code__in=session,
department__name="Information and Communication Engineering")  ]))
                for c in
CourseCode.objects.filter(course_session__session_code__in=session,d
epartment__name='Information and Communication Engineering'):
```

```
                        line = "Course = "+str(idx)+" , " +
c.course_code + ", Medium , A11, "+
str(hash_teachers[c.course_teacher.name])
                        idx+=1
                        s.append(line)

                # main portion
                slots = []
                rooms = []
                courses = []
                instructors = []

                for line in s:

                        if line.startswith("#"):
                        # ignore comments in input file
                                continue

                        lineSections = line.split("=")
                        if lineSections[0].strip() == "Slot":
                                slots.append(Slot(lineSections[1]))
                        elif lineSections[0].strip() == "Room":
                                rooms.append(Room(lineSections[1]))
                        elif lineSections[0].strip() == "Course":

        courses.append(Course(lineSections[1]))
                        elif lineSections[0].strip() == "Instructor":

        instructors.append(Instructor(lineSections[1]))

                algo = GeneticAlgorithm(days, slots, rooms, courses,
instructors)
                algo.initChromosomes(100)
                algo.execute()

                print(session)
```

Teacher Duty

```
# ---------- Teacher Duty Form  ---------- #
class TeacherDutyForm(forms.Form):
        DEPARTMENT = [(str(s).lower(), str(s).capitalize()) for s in
Department.objects.all()]

        department = forms.MultipleChoiceField(
                required=True,
                widget=forms.CheckboxSelectMultiple(
                        attrs={
                                'type':'checkbox',
                                'class':'flat',
                        }
                ),
                choices=DEPARTMENT,
        )
        start_date = forms.DateField(
                required=True,
                widget=forms.DateInput(
```

```
                                format=('%d-%m-%Y'),
                                attrs={
                                        'type':'date',
                                        'placeholder':'exam start
date',
                                }
                        ),
                )

        def clean(self):
                department = self.cleaned_data.get('department')
                start_date = self.cleaned_data.get('start_date')

        def deploy(self):
                department = self.cleaned_data.get('department')
                start_date = self.cleaned_data.get('start_date')



                # slipt date in day, month, year
                date = str(start_date).split('-')
                day = int(date[2])
                month = int(date[1])
                year = int(date[0])
                month_last = [31,28,31,30,31,30,31,31,30,31,30,31]

                department_list =
Department.objects.filter(slug__in=department)
                session="2014-15"

                total_course =
CourseCode.objects.filter(department__in=department_list).count()
                c_list = [None]*total_course
                idx=1

                for s in Session.objects.all():
                        for d in department_list:
                                cnt=idx
                                for c in
CourseCode.objects.filter(course_session=s,department__slug=d):

        c_list.insert(cnt,c.course_code)
                                        print(cnt, c, s)
                                        cnt+=idx
                                idx+=1

                courses =
CourseCode.objects.filter(course_code__in=c_list)

                # duty with routine! save in here
                exam_duty = dict()

                # teacher priority according to designation
                all_teachers = dict()

                for d in department_list:
                        professor = [ str(t.name) for t in
Teacher.objects.filter(department__slug=d,designation="professor")]
```

```
                        associate_professor = [ str(t.name) for t in
Teacher.objects.filter(department__slug=d,designation="associate
professor")]
                        assistant_professor = [ str(t.name) for t in
Teacher.objects.filter(department__slug=d,designation="assistant
professor")]
                        lecturer = [ str(t.name) for t in
Teacher.objects.filter(department__slug=d,designation="lecturer")]
                        priority=[0]
                        all teachers[str(d)] = priority + professor +
associate_professor + assistant_professor + lecturer

                print(all_teachers)
                # teacher priority according to designation
                # professor = [ str(t.name) for t in
Teacher.objects.filter(designation="professor")]
                # assistant_professor = [ str(t.name) for t in
Teacher.objects.filter(designation="assistant professor")]
                # lecturer = [ str(t.name) for t in
Teacher.objects.filter(designation="lecturer")]

                # all_teachers = professor + assistant_professor +
lecturer

                idx=1
                time=""
                new_date=""
                date_course = []
                date_teacher= []
                t_count=0
                for cc in c_list:
                        if cc==None:
                                continue

                        c = CourseCode.objects.get(course_code=cc)
                        t_count = all_teachers[c.department.slug][0]


                        pos_t = all_teachers[c.department.slug][0]+1

                        if
pos_t>=len(all_teachers[c.department.slug]):
                                pos_t=1
                                all_teachers[c.department.slug][0]=1
                        else:
                                all_teachers[c.department.slug][0]+=1

                        date_course.append(c.course_code)

        date_teacher.append(all_teachers[c.department.slug][pos_t])

                        if idx%len(department_list)==0:
                                if time=="9:00am - 1:00pm":
                                        time="2:00pm - 6:00pm"
                                else:
                                        time="9:00am - 1:00pm"
                                new_date = str(day)+"-"+str(month)+"-
"+str(year)
                                if idx>0:
```

```
                                        exam_duty[new_date]=
[date_course, time, date_teacher]
                                date_course=[]
                                date_teacher=[]

                        # exam_duty[c]=[new_date, time,
all_teachers[c.department.slug][pos_t]]

                                day+=4
                                if day>month_last[month-1]:
                                        if month==12:
                                                month=1
                                                year+=1
                                        else:
                                                month+=1
                                        day = day%month_last[month-1]

                                t_count+=1
                        idx+=1
                #exam teach duty in html

                #data filling in html
                module_dir = os.path.dirname(__file__)
                print(module_dir)
                file_path =
os.path.join(module_dir,'templates\\teacher_duty_details.html')

                #output path
                import datetime
                dt = str(datetime.datetime.now())
                dt , a  = dt.split('.')
                dt = dt.replace(" ","-")
                dt = dt.replace(":","-")

                output_path =
'home/templates/home/teacher_duty/'+dt+'.html'

                with open(file_path, 'r') as content_file:
                        htmlRepresentation = ""
                        htmlRepresentation += "<h1>Teacher Duty in
Exam</h1>"

                        htmlRepresentation += "<p> Sessions: "+
session[0] +"</p>"

                        for key in exam_duty.keys():
                                htmlRepresentation +='<div
class="row">'
                                htmlRepresentation+='<div class="col-
md-3"style="border: 1px solid #ddd;padding:6px;">'+ str(key) +
'</div>'

                                for item in exam_duty[key]:
                                        htmlRepresentation+='<div
class="col-md-3" style="border: 1px solid #ddd; padding:6px;">'+
str(item) + '</div>'

                                htmlRepresentation
+='</div>'
```

```
                        fileContents = content_file.read()
                        fileContents =
fileContents.replace('##PYTHON_CHANGE_THIS##', htmlRepresentation)

                        writeToFile = open(output_path,'w')
                        writeToFile.write(fileContents)
                        writeToFile.close()
```