Design of Class Routine and Exam Hall Invigilation System based on Genetic Algorithm and Greedy Approach

Ratul Prosad a, Md. Ashikur Rahman Khan a† and Ishtiaq Ahammad b#

a Department of Information and Communication Engineering, Noakhali Science and Technology University, Noakhali, Chattogram, Bangladesh.
b Department of Computer Science and Engineering, Prime University, Dhaka, Bangladesh.

Authors’ contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2022/v13i330316

Open Peer Review History:

This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here:
https://www.sdiarticle5.com/review-history/86671

ABSTRACT

A classroom routine is nothing more than a well-practiced respond to a teacher's instruction. Most universities handle this allocation process with a manual procedure. The manual procedure gives various challenges and is inclined to mistakes. A better approach to reliably schedule class routine is to utilize a computer assisted web-based system. Therefore, in this work, focus is given on creating automatic class routines with teacher’s requirements. This work mainly consists of two parts, named Admin panel and 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. A class routine is then created based on these fields. In user panel, we get all of information about courses, associated teacher and rooms. In this panel we can see all this information & routine. Routine can be constructed by days, teacher's and semester wise. We create this application by utilizing genetic algorithm, and implemented by using Python language. Exam Hall Invigilation is another aspect which still most universities handle manually. Therefore, an automatic Exam Hall Invigilation Management System is also developed in this work. We propose an improved algorithm to achieve automatic examination arrangement for invigilator based on

1Professor and Chairman;
2Lecturer;
3Corresponding author: E-mail: ashik@nstu.edu.bd;
The system can configure to allocate any numbers of invigilators in different examination halls in such a way that each invigilator will get equal number of duties. This algorithm has written and implemented in Java-script language.

Keywords: Class routine; routine allocation system; web based application; exam hall invigilation system; genetic algorithm; greedy approach.

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 are 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. 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.

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 [1]. 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 these 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 are a reallocating 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 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 Fig. 1.

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.
2. RELATED WORK

Classroom technologies, definition scheduling invigilators, timetable invigilator is very important topic for our literature review. A good automated timetabling survey is given in [2]. Sandhu [3] 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 [4-5]. GA is a natural selection and evolution-based heuristic algorithm [6]. Class Routine Allocation System Control is applied. Ehsan, Zahra, and Mohammad [7] 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. [8] 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 [9] suggested a method for solving this time table problem. The method of Exam Hall Invigilation Management includes user Id management, instructor Id, teacher identification, and assignment of rooms to invigilate [10]. The allocation of rooms during the analysis can be automated using the algorithm of greedy approach [11]. Using the greedy approach algorithm [12], 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. [13]. This paper also explains automated algorithm development and addresses machine security [14].

The system will increase patient throughput and use radiological services effectively here [15]. 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 [16]. 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 [17].

2.1 Classroom Technologies

In developing countries, most of today's universities classrooms have (laptop) computers, smart boards and Internet facilities [18]. Although several research studies on the design of classroom technologies concentrate on promoting learning activities [19-23], only a few research explorations also concern the creation of experiences that require users to have limited attention resources (the teacher or the students). For example, Greiffenhagen [24] studied the work of teachers during collective learning exercises to find out how teachers participate in the learning of pupils. Prieto et al. [25] 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. In 'proactive teaching,' which takes place before or after class while the teacher is alone, Yinger [26] has studied the routines of teachers. Different fields, such as School Administration, Artificial Intelligence, Mathematics or Operational Research [27], 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 [28]. In a given time period, an inspection takes place, using a set of rooms And a set of watchmen [29].

2.2 Timetable Generation Algorithms

Constraint Based Methods: We may infer that the purpose of the above approach is to minimize the violation of soft constraints and to fulfill the difficult constraints [30].

Sequential Methods: In this approach, the problem of tabling is treated as graph issues.

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.

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

2.3 Auto Generate System Based on Expert System

To obtain fast computational capabilities, we have used various optimization methods, such as Particle Swarm 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 [31]. The primary issue in a series of events is the surveillance method or tabulating. 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 [32].

2.4 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 [33]. The Genetic Algorithm operators are: population initialization [34], selection [35], crossover [36], and mutation. 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 [37]. 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 [38].

3. METHODOLOGY

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 [39]. In a production method, some data and information need to be obtained from users in order to create a system. It is a very essential part of the Software Development Life Cycle (SDLC). The main aim of our system is to establish a timetable. In our project we follow the agile model.

3.1 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 JavaScript 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.

### 3.2 Algorithms

#### 3.2.1 Genetic algorithms

The genetic algorithm is implemented by the operators to build the next generation out of the generation currently used: mutation, crossover and reproduction.

![Genetic Algorithm Flowchart](image)

**Fig. 2. Genetic Algorithm Flowchart**

The Algorithm

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

\[ \forall (i,j), i \neq j, \Delta (S_i \neq S_j \land T_{S_i} \neq T_{S_j}, \text{where } R_i \neq R_j) \]

\[ \forall (i,j), i \neq j, \Delta (L_i \neq L_j \land T_{S_i} \neq T_{S_j}) \]

2. No teacher can teach more than one class at the same time.

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:

\[ \forall i, \text{class routine } (S_i) \leq \text{capacity } (R_i), \text{ where class routine } (y) \in i, y \in i, \text{ and capacity } (y) \in i, y \in R \]
3.2.2 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 step 8: 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 these 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.

a) 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

\[ k = \text{randnum} + \sum_{i=1}^{j-1} (\text{Level}_m\_\text{Teachers}\_\text{Count}) \]
3.3 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. Our main requirements are: users, teachers, admin, hardware, and software. 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.

3.4 Diagrams & System Overview
We used here mainly three diagrams. These are: general block diagram, ER diagram, Class diagram, General block diagram.

General Block Diagram
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.

Fig. 3. General Block Diagram.

Entity Relationship Diagram
Three basic concepts are based on this model:

i. Entities
ii. Attributes
iii. Relationships

Fig. 4. 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, and Student. Attribute is either an entity types or a relation type single valued property. In this diagram, entities have 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 consisted of:

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

![Class Diagram](image)

**Fig. 5. 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.

**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.
4. TESTING ON MANUAL PROCEDURE

Testing and servicing are the study of software with user requirements and machine specifications [40]. 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.

4.1 Test Arrange and Test Exercises

A better software tester by learning programming principles is 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 riskiest 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. Here we shown a complete class routine design with a full semester. We have considered five departments for exam hall invigilation management system to maintain our whole project. As a sample, below tables are given for test exercises:
<table>
<thead>
<tr>
<th>Day</th>
<th>Year, Term</th>
<th>9.00-10.15</th>
<th>10.20-11.35</th>
<th>11.40-12.55</th>
<th>BREAK</th>
<th>1.40-2.55PM</th>
<th>3.00-4.00PM</th>
<th>4.00-5.00PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>Y-4, T-1</td>
<td>ICE-4105 TZK 812</td>
<td>ICE-5117 MARK 812</td>
<td>ICE-4101 MARK 812</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-3, T-1</td>
<td>ICE 3105 MMR 810</td>
<td>ICE-3109 SJS 810</td>
<td>ICE-3107 TZK 810</td>
<td>ICE-2106 LAB(NN) 808</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-2, T-1</td>
<td>ICE-2104 LAB (ZUS) 808</td>
<td>MATH 3113 HMSA 812</td>
<td>ICE 3108 LAB AA+TZK 808</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-1, T-1</td>
<td>PHY-1103 MBH 809</td>
<td>ICE-1105 NN 809</td>
<td>ENG 1109 809</td>
<td>ICE-1101 MU 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>Y-4, T-1</td>
<td>ICE-4104 LAB (SJS) 812</td>
<td>ICE-4107 ZUS 812</td>
<td>ICE-4103 SJS 812</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-3, T-1</td>
<td>ICE-3104 LAB(AA+TZK) 813</td>
<td>MATH 2113 SJ 810</td>
<td>ICE-2105 NN 810</td>
<td>ICE-1103 AA 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-2, T-1</td>
<td>ICE-2109 MBH 810</td>
<td>MATH-1111 MMR 809</td>
<td>HUM-1105 MMIS 809</td>
<td>ICE-1103 AA 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-1, T-1</td>
<td>ICE-1101 MU 809</td>
<td>ICE-1105 AA 809</td>
<td>HUM-1105 MMIS 809</td>
<td>ICE-1103 AA 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>Y-4, T-1</td>
<td>ICE-4109 TZK 812</td>
<td>ICE-4105 AA 812</td>
<td>ICE-5117 ARK 813</td>
<td>ICE-4106 LAB(TZK+AA) 813</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-3, T-1</td>
<td>ICE-3106 LAB (MMR+NN) 808</td>
<td>MATH-1111 MMR 809</td>
<td>ICE-3103 TZK 810</td>
<td>ICE-3109 SJS 810</td>
<td>ICE-3105 NN 810</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-2, T-1</td>
<td>ICE-2107 SJS 810</td>
<td>ICE-2103 ZUS 810</td>
<td>ICE-2109 MBH 810</td>
<td>BANGLA MRD 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-1, T-1</td>
<td>ICE-1103 MBH 809</td>
<td>MATH-1111 MMR 809</td>
<td>HUM-1105 MMIS 809</td>
<td>BANGLA MRD 809</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td>Y-4, T-1</td>
<td>ICE-4101 MARK 812</td>
<td>ICE-4107 ZUS 812</td>
<td>ICE-5117 MARK 812</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-3, T-1</td>
<td>ICE-3101 MU 810</td>
<td>ICE-3101 MMR 810</td>
<td>ICE-3107 AA 810</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-2, T-1</td>
<td>ICE-2107 SJS 809</td>
<td>ICE-2101 MMR 810</td>
<td>ICE-2101 MBH 809</td>
<td>ICE-2110 LAB (MBH) 813</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ICE-1104 LAB (MBH+AA) 813</td>
<td>ICE-1108 LAB MARK 808</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>T-4, T-1</td>
<td>ICE-4108 LAB (ZUS) 813</td>
<td>EEE-2103 MARK</td>
<td>ICE-3103</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-3, T-1</td>
<td>ICE-3102 LAB (MMR+MU) 812</td>
<td>MATH 3113 [10.00-12.00] HMSA 812</td>
<td>ICE-3103</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-2, T-1</td>
<td>MATH-2113 [11.00-1.00] SJ 810</td>
<td>MATH-2113 [10.00-12.00] HMSA 812</td>
<td>ICE-2103 ZUS 810</td>
<td>ICE-2102 LAB (AA+MBH) 813</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y-1, T-1</td>
<td>MATH-1111 MMR 809</td>
<td>ENG-1109 809</td>
<td>BANGLA MRD 809</td>
<td>ICE-1102 LAB (MU) 808</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Term Final Exam Routine
Session: 2017-18 (Y-3, T-1)
2018-19 (Y-2, T-1)
2019-20 (Y-1, T-1)

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

Table 3. Hall Invigilators List
Dept.: ICE
Session: 2015-16
Term final Examination
Hall invigilators list

<table>
<thead>
<tr>
<th>Date</th>
<th>Day</th>
<th>Invigilators List</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.12.2020</td>
<td>Sunday</td>
<td>Sultana Jahan Soheli, Apurba Adhikary, Tanvir Zaman Khan, Nishu Nath, Md. Bipul Hossain, Main Uddin,</td>
</tr>
<tr>
<td>03.01.2021</td>
<td>Sunday</td>
<td>Zayed-Us-Salehin, Sultana Jahan Soheli, Nishu Nath, Md. Bipul Hossain, Main Uddin,</td>
</tr>
<tr>
<td>13.12.2020</td>
<td>Sunday</td>
<td>Md. Ashikur Rahman Khan, Chief Invigilator</td>
</tr>
</tbody>
</table>

5. IMPLEMENTATION AND ANALYSIS

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. Django is an elevated Python web framework which promotes fast development and simple, practical design. It's constructed by professional developers to handle a lot of the headaches of web development so that we can concentrate on developing our app instead of reinventing the wheel. Originally only the admin can access the device.
<table>
<thead>
<tr>
<th>Day</th>
<th>Room</th>
<th>9:00</th>
<th>10:20</th>
<th>11:40</th>
<th>1:40</th>
<th>3:00</th>
<th>4:00</th>
</tr>
</thead>
</table>

**Fig. 7. Automatic Class Routine System (1)**

<table>
<thead>
<tr>
<th>Day</th>
<th>Room</th>
<th>9:00</th>
<th>10:20</th>
<th>11:40</th>
<th>1:40</th>
<th>3:00</th>
<th>4:00</th>
</tr>
</thead>
</table>

**Fig. 8. Automatic Class Routine System (2)**

**Fig. 9. Counting Chief Invigilators and Invigilators**

<table>
<thead>
<tr>
<th>Room</th>
<th>Time</th>
<th>Chief Invigilator</th>
<th>Invigilator</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICE-3106 [Q.G]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>8 Invigilators</td>
</tr>
<tr>
<td>MAH-2113 [M.J.]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-2101 [M.B.H]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-3105 [M.N.]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-4107 [Q.G]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>EEE-2102 [M.A.R.H]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-1116 [M.R.D]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>BANGLA 1111 [M.R.D]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-2105 [M.R.D]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>BANGLA 1111 [M.R.D]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-2107 [M.J.S]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-3709 [M.J.S]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-4103 [M.J.S]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-3709 [M.J.S]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
<tr>
<td>ICE-4107 [Q.G]</td>
<td>8:00</td>
<td>5 Invigilators</td>
<td>5 Invigilators</td>
</tr>
</tbody>
</table>
5.1 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. 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. 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.

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:
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.

6. 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.

6.1 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.

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

COMPETING INTERESTS
Authors have declared that no competing interests exist.

REFERENCES
3. Sandhu KS. Automating class schedule generation in the context of a university timetabling information system (Doctoral dissertation, Griffith University);2003.
10. Lu H, Hu Y. 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;2012.


31. Ekanayake TW. Intelligent timetable sheduler by using genetic algorithm.


© 2022 Prosad et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle5.com/review-history/86671