
**CSE7301 University Project
Review-4**

SUMMER-TERM TIMETABLE GENERATION

Batch Number: CSE-G44

Name of the HoD: Dr.Asif

Roll Number

Student Name

**Mohammed H.B Under the
Supervision of,**

20211CSE0716

Meda Sai Srihitha

20211CSE0729

S Pavani

20211CSE0736

T Vamshika

**Mr. Jerrin Joe Francis Assistant Professor
School of Computer Science and
Engineering Presidency University**

Name of the Program: CSE

Name of the Program Project Coordinator: Mr. Jerrin Joe Francis



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013



Name of the School Project Coordinators: Dr. Sampath A K / Dr. Abdul Khadar A / Mr. Md Ziaur Rahman



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013



Content

- Introduction
- Literature Review
- Proposed Method
- Objectives
- Methodology
- Timeline of the Project
- Expected Outcomes
- Algorithm & Pseudocode
- Output
- Conclusion

○ Software Components

○ References



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013



Introduction

- The Automatic Timetable Generator is a software used to generate timetable automatically. Currently timetable is managed manually. It will help to manage all the periods automatically and also will be helpful for faculty to get timetable in their phone by using application.
- Maximum and minimum workload for a Faculty for a day, week and month will be specified for the efficient generation of timetable.
- When selecting a faculty as substitute it allows to view timetable of that faculty for ensure that the faculty is free at that particular period.

-
- It is a comprehensive timetable management solutions for colleges which help to overcome the challenges in manually setting the timetable. By using this software it will be very easy for faculty to get timetable in their phones.

LITERATURE REVIEW

SL. NO	TITLE PAPER/AUTHOR/PUBLISHER/YEAR	OF THE ADVANTAGES	LIMITATIONS
--------	--------------------------------------	-------------------------	-------------

1

Tavakkol, M., & Parsa, M. *A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences.* Computers & Industrial Engineering, Volume 157, 2021, Article 107327.

The hybrid genetic algorithm enhances scheduling efficiency by incorporating faculty preferences.

It may require high computational resources for complex timetabling problems.



2	Rong, Q., & Lee, K. <i>Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms</i> . Journal of Scheduling, Volume 25, Issue 1, 2022, Pages 57-72.	Multi-objective optimization algorithms provide flexibility to address diverse scheduling criteria.	They often require extensive fine-tuning for specific scenarios.
---	--	---	--

LITERATURE REVIEW(Contd..)

SL.	TITLE OF THE PAPER/AUTHOR/PUBLISHER/YEAR	ADVANTAGES	LIMITATIONS NO
-----	---	------------	----------------



- 3 **Hassan, M., & Khalil, M.**
*An Intelligent Course
Scheduling System Using
Machine Learning
Techniques.*
Journal of Educational
Computing Research,
Volume 61, Issue 3, 2023,
Pages 445-465.

Machine learning
techniques enable
adaptive and automated
course scheduling.

They depend
heavily on large
and accurate
training datasets.

4	Wang, X., & Xu, H. A Novel Memetic Algorithm for Solving University Timetabling Problems. Expert Systems with Applications, Volume 178, 2021, Article 115018.	Combines global and local search strategies for efficient timetabling optimization.	Memetic algorithms may require significant computational resources for complex problems.
---	---	---	--

LITERATURE REVIEW(Contd..)

SL.	TITLE OF THE	ADVANTAGES	LIMITATIONS	NO
	PAPER/AUTHOR/PUBLISHER/YEAR			

Pillay, N., & Qu, R. *An Evolutionary Algorithm for the Multi-Criteria University Timetabling Problem.*

Applied Soft Computing, Volume 115, 2022, Article 108163.

Effectively balances multiple criteria, such as faculty preferences and resource availability.

May require extensive parameter tuning for effective implementation.

6	<p>Nguyen, T. T., & Le, M. T. <i>A Deep Reinforcement Learning-Based Approach for Automated Course Scheduling.</i> IEEE Access, Volume 9, 2021, Pages 115765115778.</p>	<p>Utilizes deep reinforcement learning to dynamical ly adapt and improve scheduling .</p>	<p>Requires large amounts of training data and computational power to achieve optimal results.</p>
---	--	--	--

Proposed Method

The proposed method for the Automatic Timetable Generator involves developing a system that automates timetable creation and management by considering faculty availability, workload limits, and class requirements. It uses a constraint satisfaction approach with algorithms like Genetic for optimal scheduling, ensuring no overlapping periods and balanced workloads. The system enables faculty to view timetables. A mobile app integration ensures real-time access for users, and the system automates updates for absentees or schedule changes, providing an efficient and scalable solution for timetable management.

Objectives

- **Automation of Timetable Creation:** To automate the generation of timetables based on faculty availability, workload constraints, and class requirements, reducing manual effort.
- **Improved Faculty Workload Distribution:** To ensure fair and balanced workload distribution for faculty across days, weeks, and months.
- **Real-Time Accessibility:** To enable faculty and administrators to access timetables and notifications in real-time through a mobile application.
- **User-Friendly Interface:** To design a simple and intuitive interface for faculty and administrators to interact with the system effectively.

Methodology

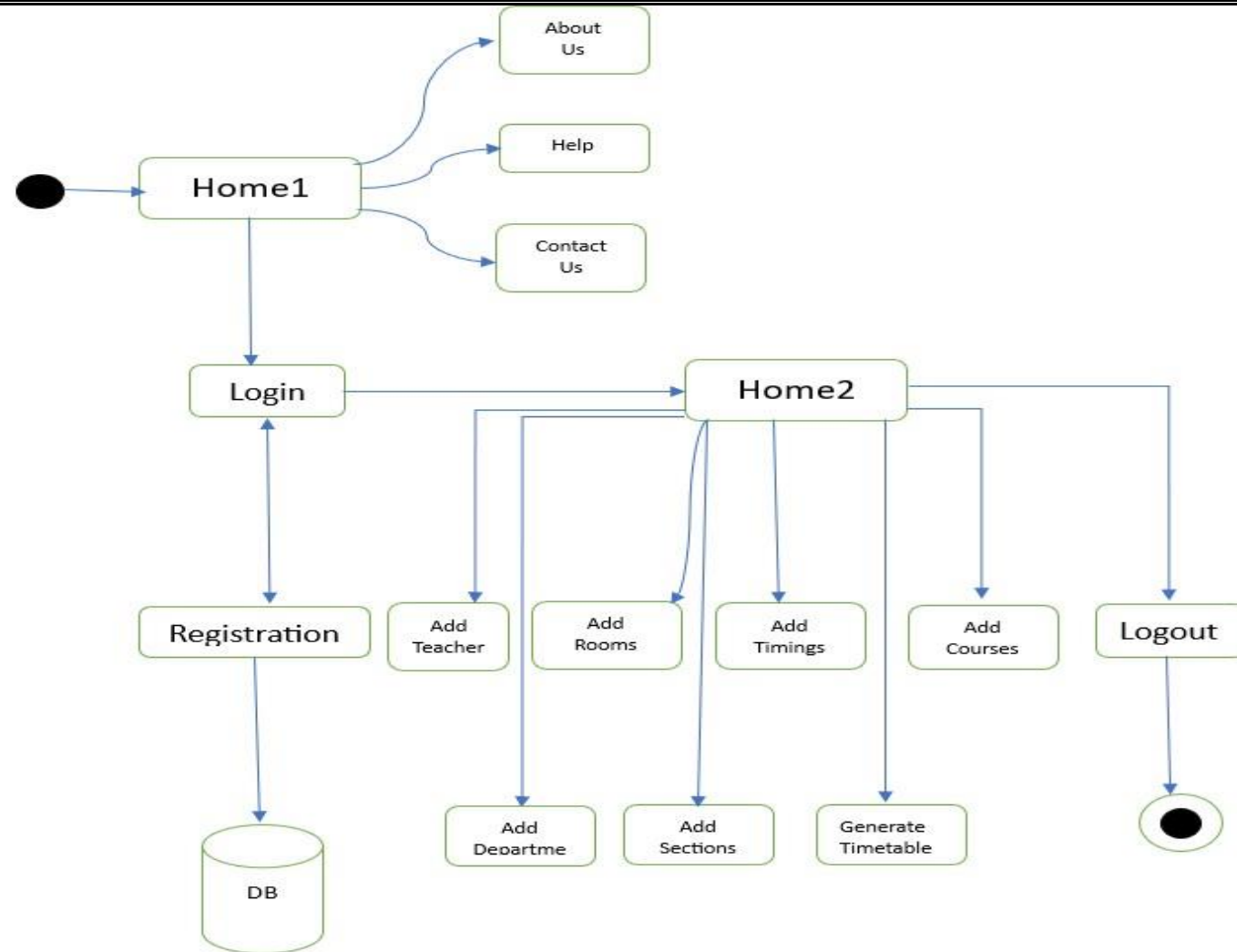
The Automatic Timetable Generator involves designing a Python-based system that uses constraint satisfaction and optimization algorithms, such as Genetic algorithm , to generate timetables based on faculty availability, workload limits, and class requirements, while a user-friendly interface allows faculty to view timetables, apply for leave, and manage substitutes. The system integrates mobile access for real-time updates and notifications and dynamically adjusts schedules to accommodate absences or changes, ensuring efficient and automated timetable management.

Architecture



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013





Software Components

Technology Stack Components:

HTML

CSS

Python

Django








JavaScript

Timeline of the Project



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013



	REVIEW - 0	REVIEW - 1	REVIEW - 2	REVIEW - 3	Final Viva-Voce
	29-Jan-2025 To 31-Jan-2025	18-Feb-2025 To 21-Feb-2025	17-Mar-2025 To 21-Mar-2025	16-Apr-2025 To 19-Apr-2025	10-May-2025 To 17-May-2025
PLANNING					
REQUIREMENT ANALYSIS					
DESIGN					
CODING					
TESTING					
DEPLOYMENT					
PAPER PUBLICATION					

Expected Outcomes

- **Time Efficiency:** Significant reduction in the time required to create and manage timetables.
- **Error Reduction:** Elimination of manual errors in scheduling, such as overlapping periods or exceeding workload limits.
- **Improved Accessibility:** Real-time access to timetables and notifications for faculty and administrators through a mobile application.
- **Balanced Workloads:** Equitable distribution of teaching workloads among faculty members, ensuring compliance with specified limits.
- **Scalability and Adaptability:** A scalable solution capable of handling larger institutions and adapting to evolving scheduling requirements.

-
- **User Satisfaction:** Increased satisfaction among faculty and administrators due to the system's efficiency, reliability, and ease of use.

Algorithm

1. Start
2. Navigate to Home1
 - a. If the user selects About Us:
 - i. Display details about the system and its purpose. Validate and insert into the database.
 - ii. Accept teacher details (ID, name, department, etc.).
 - b. If the user selects Help:
 - i. Add Rooms: details (room ID, name, capacity, etc.).
 - ii. Display a help page or FAQs related to system usage.
 - iii. Validate and insert into the database.
 - c. If the user selects Contact Us:
 - i. Add Timings: details (start time, end time, session type, etc.).
 - ii. Display contact details for technical support or system queries.
 - iii. Accept timing details (start time, end time, session type, etc.).
 - d. Allow navigation to the Login page. ii. Validate and insert into the database.
3. Login Page
 - a. Accept user credentials (username and password). i. Accept course details (course ID, name, semester, etc.). ii. Validate and insert into the database.
 - b. Verify credentials with the database.
5. Navigate to Home2

e. Add Department:

c. If

valid:

- i. Accept department details (department ID, name, head of department, etc.).
- i. Redirect to Home2. ii. Validate and insert into the database.
- d. Else, display "Invalid credentials".
- f. Add Sections:
- 4. Registration i. Accept section details (section ID, course ID, etc.). ii. Validate and insert into the database. a. If the user is new:
- g. Generate Timetable:
- i. Accept registration details (name, email, password, etc.). i. Fetch teacher, room, course, and timing details from the database.
- ii. Insert details into the database. ii. Generate a timetable based on availability and constraints.
- iii. Display "Registration successful".
- 6. Logout

7. End

Pseudocode

Start

// Navigation to Home1

Display options: About Us, Help, Contact Us,
Login If User selects "About Us": Display
system details and its purpose If User selects

"Help": Display help page or FAQs If User selects
"Contact Us": Display contact details for technical support
If User selects "Login":
 Navigate to Login Page
// Login Process
 Accept username and password Verify
credentials with database If valid:
Redirect to Home2 Else:
 Display "Invalid credentials"

// Registration Process

Accept registration details (name, email, password, etc.)

Insert registration details into the database

Display "Registration successful"

// Navigate to Home2 (Post Login)

Display options: Add Teacher, Add Rooms, Add Timings, Add Courses, Add Department, Add Sections, Generate Timetable

If User selects "Add Teacher":

Accept teacher details (ID, name, department, etc.)

Validate details

Insert teacher details into database

If User selects "Add Rooms":

Accept room details (room ID, name, capacity, etc.)

Validate details

Insert room details into database

If User selects "Add Timings":

Accept timing details (start time, end time, session type, etc.)

Validate details

Insert timing details into database

If User selects "Add Courses":

Accept course details (course ID, name, semester, etc.)

Validate details

Insert course details into database



Pseudocode

If User selects "Add Department":

- Accept department details (department ID, name, head of department, etc.)

- Validate details

- Insert department details into database

- If User selects "Add Sections":

- Accept section details (section ID, course ID, etc.)

- Validate details

- Insert section details into database

If User selects "Generate Timetable":

- Fetch teacher, room, course, and timing details from the database

- Generate timetable based on availability and constraints

// Logout Process

- If User selects "Logout":

End session and return to Home1

End



**PRESIDENCY
UNIVERSITY**
Private University Estd. in Karnataka State by Act No. 41 of 2013



Hello ! Lets Get Started with TTGS

Welcome to TTGS!, a timetable management system for colleges and universities to schedule their organization's timetable. TTGS started off as a company that formed strong relationships with schools, colleges and universities to help them generate their timetable with ease and maximum efficiency thus saving the organization's valuable time. Get started by referring to the flow of the system below or Contact Us with any queries or difficulties.

[Get Started »](#)[Contact Us »](#)

1. Add Teachers

Navigate to the "Add Teachers" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Teacher" option.

2. Add Rooms

Navigate to the "Add Rooms" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Rooms" option.

3. Add Timings

Navigate to the "Add Timings" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Timings" option.

Output



4. Add Courses

Navigate to the "Add Courses" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Courses" option.

5. Add Departments

Navigate to the "Add Departments" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Departments" option.

6. Add Sections

Navigate to the "Add Sections" section and add the corresponding details. Incorrect Details can also be removed via the "Edit Sections" option.

7. Generate Timetable

Navigate to the "Generate Timetable" Section and click on Generate Timetable and wait patiently as our algorithm works its magic.

© Pavani



**PRESIDENCY
UNIVERSITY**
Private University. Estd. in Karnataka State by Act No. 41 of 2013



TTGS | Generated Timetable

CS101 (Computer Science)

Class #	Course	Venue(Block-Room)	Instructor	Class Timing
0	C1 Java	304	T6 Gandhalee Manohar	T5 Tuesday 3:30 - 4:30
1	C1 Java	401	T6 Gandhalee Manohar	M3 Monday 11:30 - 12:30
2	C1 Java	401	T6 Gandhalee Manohar	M2 Monday 10:30 - 11:30
3	C1 Java	306	T5 Shruti Shah	M3 Monday 11:30 - 12:30
4	C1 Java	304	T5 Shruti Shah	T3 Tuesday 12:30 - 1:30
5	C2 OS	306	T2 Bertilla Fernandes	M2 Monday 10:30 - 11:30

6	C2 OS	301	T1 Wilson Rao	T5 Tuesday 3:30 - 4:30
7	C2 OS	306	T1 Wilson Rao	T4 Tuesday 2:30 - 3:30
8	C2 OS	303	T2 Bertilla Fernandes	T4 Tuesday 2:30 - 3:30
9	C2 OS	304	T2 Bertilla Fernandes	M1 Monday 9:30 - 10:30

EC101 (ECE)

Class #	Course	Venue(Block- Room)	Instructor	Class Timing
30	C3 Networking	304	T1 Wilson Rao	Th2 Thursday 2:30 - 3:30
31	C3 Networking	401	T2 Bertilla Fernandes	T3 Tuesday 12:30 - 1:30
32	C4 Swift	303	T3 Priti Shelar	T5 Tuesday 3:30 - 4:30
33	C4 Swift	306	T4 Sunita Jena	M3 Monday 11:30 - 12:30

34	C5 Aerodynamics	401	T6 Gandhalee Manohar	M3 Monday 11:30 - 12:30
35	C5 Aerodynamics	306	T6 Gandhalee Manohar	T4 Tuesday 2:30 - 3:30

EE101 (EEE)

Class #	Course	Venue(Block- Room)	Instructor	Class Timing
36	C10 Circuit	304	T3 Priti Shelar	Th3 Thursday 9:30 - 10:30
37	C10 Circuit	401	T3 Priti Shelar	M4 Monday 12:30 - 1:30
38	C10 Circuit	401	T3 Priti Shelar	M2 Monday 10:30 - 11:30
39	C10 Circuit	303	T3 Priti Shelar	M1 Monday 9:30 - 10:30
40	C9 Wiring	401	T4 Sunita Jena	T1 Tuesday 10:30 - 11:30
41	C9 Wiring	401	T4 Sunita Jena	T2 Tuesday 11:30 - 12:30

42	C9 Wiring	306	T4 Sunita Jena	T3 Tuesday 12:30 - 1:30
43	C9 Wiring	303	T1 Wilson Rao	T4 Tuesday 2:30 - 3:30

Conclusion

The proposed work represents a genetic algorithm approach is very effective and useful on the lecture timetabling problems. Using the method we have described and shown a great potential for leading timetable in future which are fairer to students. The framework seems directly applicable to a very wide variety of other timetabling problem. For example, experimental result shows that a key aspect towards its success is the employment of the mutation operator described. The GA in timetabling framework has been shown to be successful on several real problem ‘University Department size’, and so it seem we can justify the expectation for it to work very well on other problems of similar size and nature . That is, there is no reason to suspect that there is anything particularly easy about the problem it was tested on. In comparison to other real problems. Much works remain to do to see how performances scale to larger and otherwise different kinds of timetabling problems.

Github Link

The Github link provided should have public access permission.

GitHub Link : <https://github.com/Srihitha12-eng/-Summer-term-Timetable-generation-.git>



References

- [1]. **Tavakkol, M., & Parsa, M.** (2021). A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences. *Computers & Industrial Engineering*, 157, 107327.
[<https://doi.org/10.1016/j.cie.2021.107327>]
- [2]. **Rong, Q., & Lee, K.** (2022). Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms. *Journal of Scheduling*, 25(1), 57-72.
[<https://doi.org/10.1007/s10951-021-00788-3>]
- [3]. **Hassan, M., & Khalil, M.** (2023). An Intelligent Course Scheduling System Using Machine Learning Techniques. *Journal of Educational Computing Research*, 61(3), 445-465.

[<https://doi.org/10.1177/07356331221122514>]

References

[4]. Wang, X., & Xu, H. (2021). A Novel Memetic Algorithm for Solving University Timetabling Problems. *Expert Systems with Applications*, Volume 178, Article 115018. [<https://doi.org/10.1016/j.eswa.2021.115018>]

[5]. Pillay, N., & Qu, R. (2022). An Evolutionary Algorithm for the Multi-Criteria University Timetabling Problem. *Applied Soft Computing*, Volume 115, Article 108163. [<https://doi.org/10.1016/j.asoc.2021.108163>]

[6]. **Nguyen, T. T., & Le, M. T. (2021).**A Deep Reinforcement Learning-Based Approach for Automated Course Scheduling. *IEEE Access*, Volume 9, 2021, Pages 115765-115778.

[<https://doi.org/10.1109/ACCESS.2021.3106042>]

- [7] **Chowdhary, A., Kakde, P., Dhoke, S., Ingle, S., Rushiya, R., & Gawande, D. (2014).** Timetable Generation System. *International Journal of Computer Science and Mobile Computing*, 3(2), 410–414. [<https://ijcsmc.com/docs/papers/February2014/V3I2201474.pdf>]
- [8] **Almeida, J., Figueira, J. R., Francisco, A. P., & Santos, D. (2023).** A Hybrid Meta-Heuristic for the Generation of Feasible Large-Scale Course Timetables Using Instance Decomposition. *arXiv preprint* arXiv:2310.20334. [<https://arxiv.org/abs/2310.20334>]

-
- [9] Gaikwad, M., Gaikwad, A., Chaudhary, M., Sawarkar, D., Bhargava, M., & Anaspure, V. (2022). Auto Timetable Generator. *International Research Journal of Modernization in Engineering Technology and Science*, 4(5), 2976–2981.
[https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/23679/final/fin_irjmets1653043540.pdf]
 - [10] Al-Khair, A. (2015). Algorithm Tools Proposed to Solve the Problem of Timing While Providing Teacher Availability. *International Research Journal of Modernization in Engineering Technology and Science*, 4(5), 2976–2981.

Thank You