

PSCS190-Examination Timetable Generation

Batch Number: ISR-G03

Roll Number	Student Name
Susheeth G	20211ISR0036
Ritish N	20211ISR0047
Vidyashree BN	20211ISR0039
Mithali S Anand	20211ISR0082
Tejashwini BA	20211ISR0040

**Under the Supervision of,
Dr.Murali Parameswaran
Professor / Associate Professor / Assistant Professor
School of Computer Science and Engineering
Presidency University**

Name of the Program: B.TECH. INFORMATION SCIENCE &ENGINEERING (AI AND ROBOTICS)

Name of the HoD: Dr. Zafar Ali Khan

Name of the Program Project Coordinator: Dr.Afroz Pasha

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



Content

- Problem Statement
- Analysis of Problem Statement
- Timeline of the Project
- References



Problem Statement Number: PSCS190

Organization: Presidency University
Category : Software

Problem Description: The aim of the project is to generate a timetable given information such as course registration information, elective information, etc. The project will involve defining constraints, domains and variables before slotting the examinations in the respective slots. Further constraints, such as faculty availability for tasks like invigilation, student and faculty exhaustion (Eg. Not allocating more than 3 hours of examinations or invigilation in a day), etc. are also to be considered while generating the timetable. The final deliverable for this project will aid in the smooth and fair conduct of examinations in Presidency University.

Difficulty Level: Complex



Github Link

- The Github link provided should have public access permission.
- Github Link
- <https://github.com/susheeth1/Timetable-scheduling-problem-for-university-examinations>



Analysis of Problem Statement

Technology Stack Components:

- Frontend (User Interface)

The frontend should allow users (administrators, faculty, students) to interact with the system, input data (course registrations, faculty availability), and view the generated timetable.

- Backend (Business Logic and APIs)

The backend will handle the core logic of timetable generation, including constraint management, scheduling algorithms, and data processing.

- Database (Data Storage)

The database stores data about students, courses, faculty, rooms, and schedules. For this problem, both relational and non-relational databases can be considered, depending on your specific needs.

- Scheduling Algorithms and Optimization

The core of this project is the scheduling algorithm. You can use various techniques to solve the Constraint Satisfaction Problem (CSP), ensuring that exams are scheduled optimally without conflicts.

- APIs (Communication Between Frontend and Backend)

- DevOps and Deployment

Once the system is built, you'll need to deploy it to ensure it's accessible to users and runs reliably.

- Version Control

Git: Use Git for version control. Host your repository on GitHub or GitLab to manage collaboration, track issues, and ensure proper versioning.

Analysis of Problem Statement (contd...)

Software and Hardware Requirements:

Software Requirements:

Operating System: Windows, macOS, or Linux

Programming Language: Python, Java, or similar for algorithm development

Database: MySQL, PostgreSQL, or equivalent for storing course and exam data

IDE: Visual Studio Code, PyCharm, or Eclipse for development

Optimization Tools: Constraint satisfaction libraries (e.g., Google OR-Tools, PuLP) for scheduling algorithms

User Interface: Web framework (React, Angular, or similar) or desktop interface for displaying timetables

Version Control: Git for code management

Deployment Platform: Cloud-based server (e.g., AWS, Azure) or local server for hosting the application

Hardware Requirements:

Processor: Intel i5 or higher

RAM: 8 GB or more

Storage: 256 GB SSD or higher

Internet Connection: For accessing cloud services or hosting platform

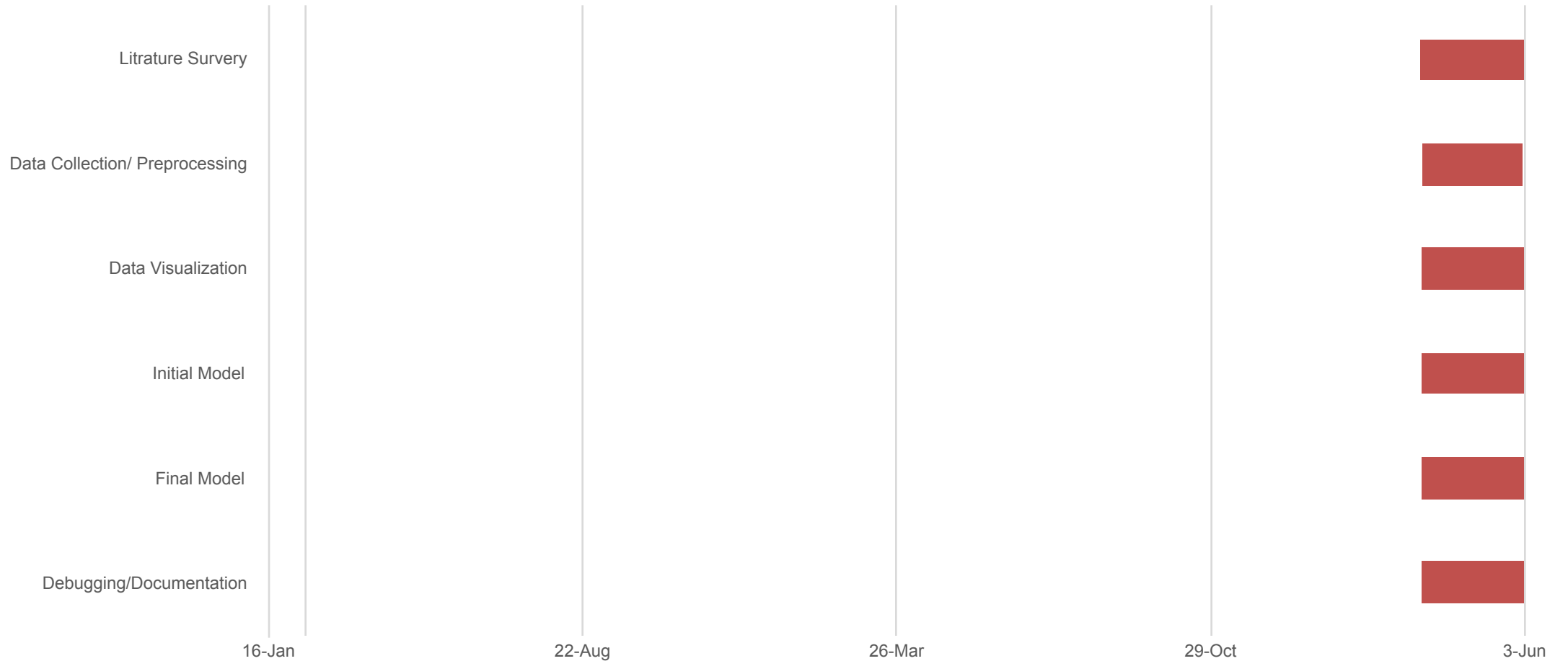
Display: Standard monitor or laptop display for development and UI testing

Analysis of Problem Statement (contd...)

Constraints of the problem statement

- Allocate exams into available time slots, ensuring no overlap for students registered in multiple courses.
- Ensure elective exams do not clash with core course exams for students registered in those electives.
- Schedule exams while adhering to faculty availability for invigilation duties.
- Avoid scheduling faculty for multiple invigilation tasks in overlapping time slots.
- Limit the total examination hours for students and faculty to a maximum of 3 hours per day to avoid exhaustion.
- Ensure sufficient gaps between exams to prevent fatigue for both students and faculty.
- Ensure equitable distribution of exam schedules to avoid consecutive heavy exam days for students.
- Balance faculty workload for invigilation, preventing back-to-back sessions.
- Consider the availability of exam halls or venues while slotting exams.

Timeline of the Project (Gantt Chart)



References (IEEE Paper format)

- **Agbolade, S. J., Ayinla, B. I., & Odeniyi, L. A. (2024, June). Optimisation of University Examination Timetable Using Hybridised Genetic and Greedy Algorithms. *International Journal of Computer (IJC)*, 51(1), 1-16.**
- **Rashmi, K. R., & Abhishek, M. B. (2021, June). Automated University Timetable Generation Using Prediction Algorithm. *International Research Journal of Engineering and Technology (IRJET)*, 8(6), 2345-2350.**
- Burke, E., Elliman, D., & Weare, R. (1994). A Genetic Algorithm for University Timetabling. In AISB Workshop on Evolutionary Computing, Leeds.



Thank
You!



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

