ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/1342-1347
Pragada Harika Sravani *et. al.*, / International Journal of Engineering & Science Research

Student Team Maker

Dr R Dinesh Kumar, Pragada Harika Sravani, Jampula Mythri
[1] Associate Professor, Department Of Cse, Bhoj Reddy Engineering College For Women, India.
[2,3]B. Tech Students, Department Of Cse, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

In colleges and universities, creating student study groups and assigning the right faculty guide is important for better learning and support. This project helps automate that process.

Student data like names, roll numbers, marks, CGPA, and any backlogs are taken from an Excel sheet and stored in a database. Students who have passed all exams are grouped based on their performance, with each batch having 4 students. If there are any extra students, they are added to a smaller final batch. Students with backlogs are placed in separate groups to focus on their needs.

Faculty guides are then assigned to each batch. Only experienced faculty (with at least 5 years of experience) and those who specialize in the required subject area are selected. Faculty preferences are also considered while making the match.

This system makes the team formation process faster, fairer, and more effective, helping students get the right guidance and improving academic and optimizing faculty support. The manual processes traditionally used for forming study groups and assigning faculty mentors often result in inefficiencies, inaccuracies, and inconsistencies. These issues can lead to unbalanced academic performance within groups, misalignment between mentor expertise and student needs, and an overall lack of equitable resource distribution. The increasing size of student populations and the growing complexity of academic requirements necessitate a systematic approach to these tasks. This project introduces an automated solution to address these challenges by using data-driven methodologies to create balanced study groups and allocate faculty guides based on their qualifications and domain expertise. The system is designed to streamline administrative workflows, reduce errors, and ensure fairness in the distribution of students and faculty resources. By automating repetitive tasks and incorporating intelligent algorithms, the proposed solution enables institutions to focus on delivering high-quality education and mentorship.

1. INTRODUCTION

In academic institutions, grouping students effectively into study batches plays a vital role in fostering collaborative learning environments

2-REQUIREMENT ANALYSIS

Functional Requirements

- Group Formation Logic:The system must allow for grouping students based on specified

criteria

- User Interface:Provide a user-friendly interface for educators to input criteria and view group assignments.
- Data Input**:** System must accept the data from excel sheet.

Non-Functional Requirements

Performance: The system should respond quickly to user inputs and complete group formations within a reasonable time frame.

Scalability**:-**The system must handle varying numbers of students and adapt to different educational contexts without performance degradation

Reliability**:-**The system must maintain high availability and recover quickly from failures.

Software Resources

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system 14 should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity. The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

Hardware Resources

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should be what the system does and not how it should be implemented. Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

3-DESIGN

The system design for the project, Automated Study Group Formation and Mentor Allocation, involves a structured architecture to ensure the seamless operation of data processing, group formation, and mentor assignment. Below is an overview of the system design, broken down into its major components:

3.1 SYSTEM ARCHITECTURE

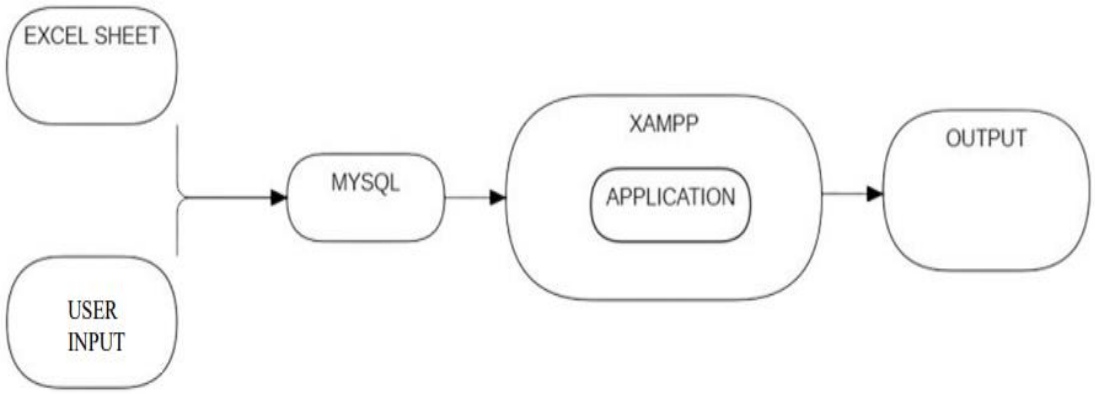


Figure 3.1: Architecture Diagram

4-IMPLEMENTATION METHODOLOGY

The implementation methodology for Team Generator can be divided into several phases, each focusing on specific tasks to achieve the goal of efficient student group formation and faculty guide allocation. Below is a detailed step-by-step methodology:

Data Extraction and Preprocessing Objective:

Extract student data from an Excel sheet, validate it, and store it in a database for processing.

Tasks:

• Extract Student Data:

Extract essential student data from an Excel sheet, including names, roll numbers, semester-wise marks, CGPA, and backlogs (if applicable).

Use libraries such as PhpSpreadsheet (in PHP) to handle the extraction of data from the Excel file. • Data Validation: o Ensure data integrity by checking for missing or incomplete entries (e.g., missing CGPA or semester marks). o For students with pending examinations, track the number of backlogs in a x

5-TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement. Testing for a Multilevel Data Concealing Technique that integrates Steganography and Visual Cryptography is crucial to ensure its functionality, security, and reliability. The testing process involves several stages, including unit testing, integration testing, and security testing.

Unit Testing

During This first round of testing, the program is submitted to assessments that focus on specific units

or components of the software to determine whether each one is fully functional. In this phase, a unit can refer to a function, individual program or even a procedure, and White box testing method is usually used to get the job done.

One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It quite common for software developers to perform unit tests before delivering software to testers for formal testing.

Integration Testing

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they properly integrated, it will affect the functionality of the software program. In order to run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on the way in which the units are defined.

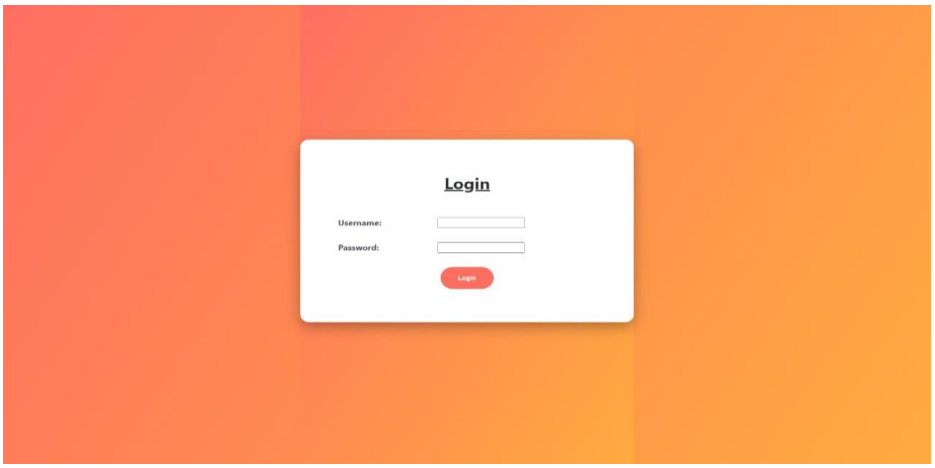
System Testing

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System

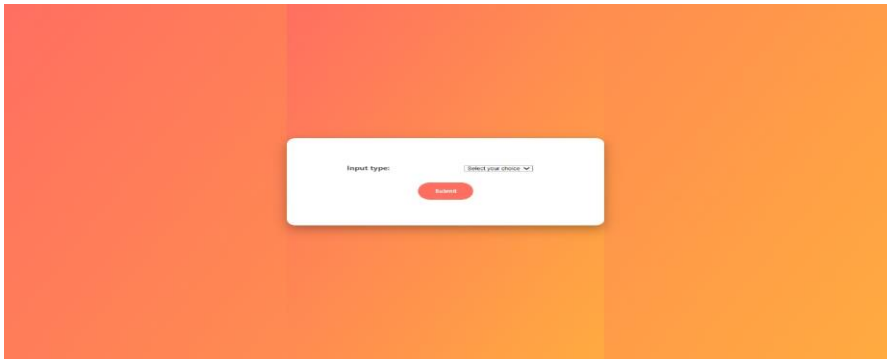
Acceptance Testing

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production. The extensiveness of these tests is just another reason why bringing software testers in early is important. When a program is more thoroughly tested, a greater number of bugs will be detected; this ultimately results in higher quality software.

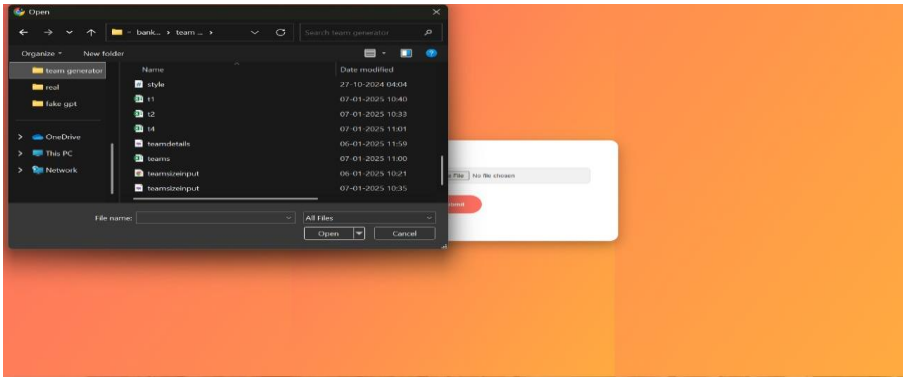
6-SCREENSHOTS



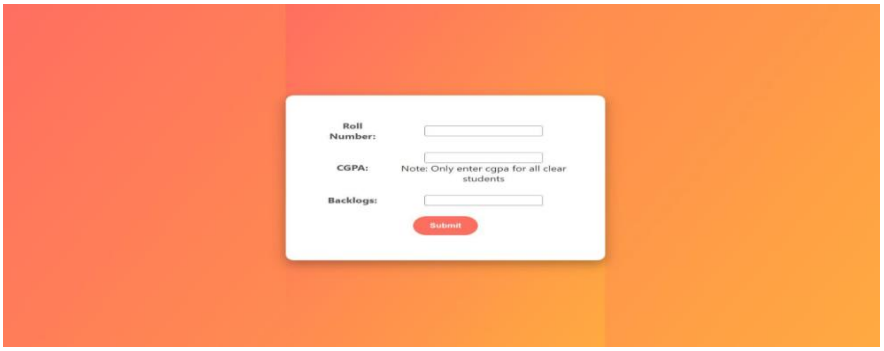
Screenshot 6.1 Login Page



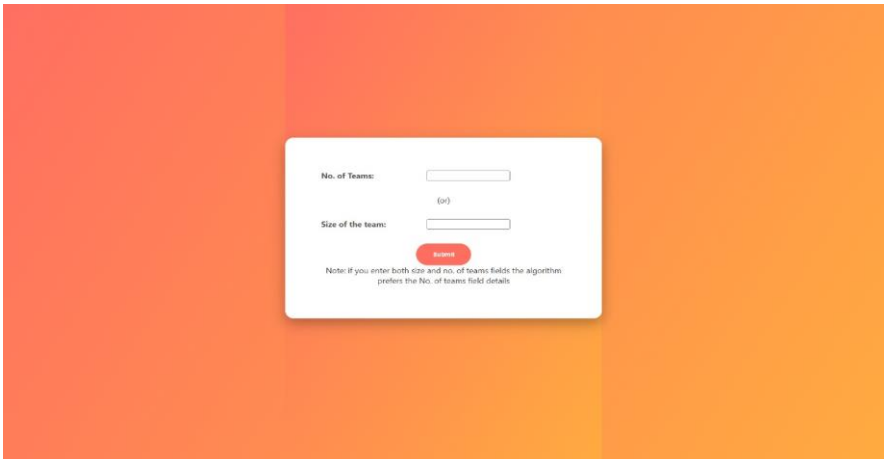
Screenshot 6.2 Input type selection page



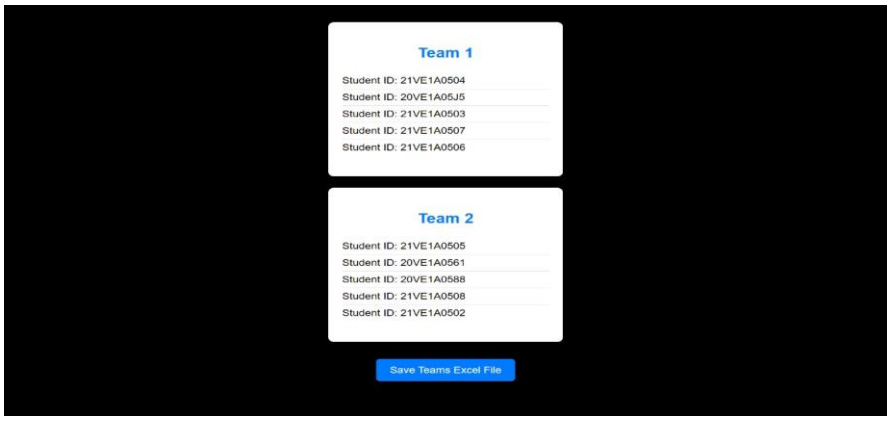
Screenshot 6.3 File uploading page



Screenshot 6.4 Student details input page



Screenshot 6.4 Team size input page



Screenshot 6.6 Final output page

7-CONCLUSION

This project efficiently automates student group formation and faculty mentor allocation, reducing manual workload and ensuring balanced, high-performing teams. It enhances academic collaboration, provides targeted support for students with backlogs, and strengthens mentorship by aligning faculty expertise with student needs—ultimately improving the overall quality of education.

REFERENCES

[1] J. Bayer, H. Bydzovska , J. Geryk, T. Obsivac , and L. Popelinsky , "Predicting dropout from social behaviour of students," Proc. 5th Int. Conf. Educ. Data Min., no. Dm, pp. 103–109, 2012.

[2] A. Pradeep, S. Das, and J.J. Kizhekkethottam ,"Students dropout actor prediction using EDM techniques," Proc. IEEE Int. Conf. Soft-Computing Netw. Secur. ICSNS 2015, 2015.

[3] B. Guo, R. Zhang, G. Xu, C. Shi, and L. Yang, "Predicting students performance in educational data mining ," Int. Symp . Educ. Technol. ISET 2015,
pp. 125–128, 2016.

[4] Francesca del Bonifro, Maurizio Gabbrielli, Giuseppe Lisanti and Stefano Zingaro, "Student Dropout Prediction", Artificial Intelligence in Education, pp.129-140, 2020.

[5] Vivek Sasikumar Iyer, Siddharth Chaudhury and Dr.M.Shobana , "Mooc Student Dropout Prediction Using Machine Learning Algorithms" , International Journal of Engineering Research in Computer Science and Engineering, Vol 8, Issue 5, May 2021.