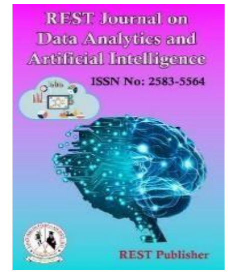
REST Journal on Data Analytics and Artificial Intelligence
Vol: 4(1), March 2025
REST Publisher; ISSN: 2583-5564
Website: http://restpublisher.com/journals/jdaai/
DOI: https://doi.org/10.46632/jdaai/4/1/4



Dynamic Student Group and Faculty Assignment Platform

Dhruv Lunawath, Samitha Reddy Yella, Singireddy Ashritha Reddy, Pardeep Kumar
School of Engineering Anurag University, Hyderabad, India.
**Corresponding Author Email: dhruvlunawath@gmail.com*

Abstract. *The Dynamic Student Group and Faculty Assignment Platform is an innovative solution designed to streamline the process of organizing student groups and assigning faculty members in academic settings. This platform utilizes intelligent algorithms to efficiently allocate students into groups based on various criteria such as skill set, course requirements, and learning preferences, while also optimizing faculty assignments according to their expertise, availability, and teaching load. The system allows for real-time adjustments, making it highly adaptable to changes in class sizes, course schedules, and faculty availability. This platform offers a user-friendly interface where students and faculty can access relevant information, including group members, course details, and upcoming assignments. It supports seamless collaboration among students through integrated communication tools, while also providing faculty with detailed reports on student progress, group dynamics, and engagement. The platform's dynamic nature enables it to handle fluctuations in group configurations, course offerings, and instructor assignments, ensuring that educational goals are consistently met.The system employs data-driven analytics to continuously improve the grouping and assignment process, taking into account performance metrics and feedback to refine future assignments. This enhances the overall educational experience, fosters better collaboration, and ensures that resources are allocated efficiently. By leveraging technology, the Dynamic Student Group and Faculty Assignment Platform significantly reduces administrative overhead, eliminates manual processes, and fosters an environment of flexibility and productivity in educational institutions. This leads to better alignment between students' needs and faculty expertise, ultimately promoting academic success and fostering a collaborative learning atmosphere.*
Keywords*: Dynamic assignment, Student grouping, Faculty allocation, Optimization, Scheduling, Resource management.*

1. INTRODUCTION

In today's rapidly evolving educational landscape, fostering effective collaboration among students is essential for developing critical 21st-century skills such as teamwork, communication, problem-solving, and adaptability [1]. Group-based learning has been widely recognized as an effective pedagogical approach that enhances student engagement, encourages knowledge sharing, and promotes a deeper understanding of complex topics [2]. However, the process of forming student groups plays a crucial role in determining the overall success of collaborative learning. Poorly structured groups can lead to disengagement, unequal workload distribution, and ineffective teamwork, ultimately impacting learning outcomes and student satisfaction [3].

Traditional methods of forming student groups, such as random allocation, self-selection, or manual instructor assignment, often fall short of ensuring an optimal balance within teams [4]. These methods may overlook critical factors such as individual strengths, academic performance, learning styles, and social compatibility, leading to imbalanced groups [5]. For example, self-selected groups may result in students working with familiar peers rather than forming diverse teams that encourage cross-disciplinary learning and exposure to different perspectives. Similarly, manually assigning groups can be a tedious and subjective process for instructors, increasing their administrative workload while failing to ensure an equitable distribution of skills and competencies among team members [6]. Consequently, there is a growing need for data-driven, automated approaches that can systematically optimize student grouping while reducing the burden on educators.

One promising method for achieving this is K-Means clustering, a widely used algorithm in unsupervised machine learning that excels at grouping data based on inherent similarities. K-Means clustering is an iterative algorithm that partitions a dataset into k distinct clusters, where each cluster represents a group of students with similar characteristics. The process begins by selecting k initial centroids (the central points of each cluster), followed by iteratively assigning each data point (student) to the nearest centroid based on a chosen distance metric, such as Euclidean distance. After all points are assigned, the algorithm recalculates the centroids as the mean of all points in each cluster. This iterative process continues until convergence is achieved, meaning that the centroids stabilize and data points no longer change clusters significantly. K-Means is particularly advantageous in educational settings due to its simplicity, computational efficiency, and effectiveness in handling large datasets with multiple attributes.
By applying K-Means clustering to student data—including academic performance, technical skills, learning preferences, and past collaborative experiences—educators can generate optimized student groups that are both balanced and diverse. This ensures that each team has a heterogeneous mix of students, fostering peer learning, reducing knowledge gaps, and encouraging a fair distribution of work. Additionally, the use of clustering-based grouping techniques removes biases associated with manual group formation and ensures a more systematic, objective, and scalable approach to student team assignments [7].

Moreover, integrating machine learning-driven student grouping has broader implications for the future of education. With advancements in artificial intelligence (AI) and educational data mining (EDM), adaptive grouping mechanisms can be developed to dynamically restructure groups based on real-time performance metrics, student feedback, and behavioral analysis [8]. These intelligent systems can continuously refine grouping strategies to enhance engagement, collaboration, and learning efficiency. By leveraging data analytics and machine learning, educators can shift their focus from administrative tasks to mentoring and instructional guidance, ultimately improving the quality of education [9].

In this research, we explore the application of K-Means clustering in the automated student grouping process [10]. The study aims to evaluate the effectiveness of this approach by analyzing key performance metrics, student feedback, and project outcomes. By demonstrating the advantages of a data-driven, AI-powered student grouping system, this research highlights the potential of machine learning in transforming educational methodologies and fostering a more inclusive, efficient, and engaging learning environment [11].

2. LITERATURE SURVEY

The challenge of optimizing the assignment of students to groups and faculty to courses has been an ongoing concern in the field of education. Over the years, numerous studies have proposed various methods for improving these processes, particularly focusing on dynamic adaptation, group diversity, and resource allocation. This literature survey explores key developments in the area of dynamic student grouping and faculty assignment, examining algorithms, optimization models, and collaboration platforms.

Student Group Formation Algorithms: One of the primary challenges in dynamic student grouping is ensuring that groups are balanced in terms of academic performance, skills, and personal characteristics. Traditional methods of grouping often fail to consider diverse factors such as social compatibility and personal learning styles. To address this issue, researchers have explored various algorithms for optimal group formation.

For example, Suleiman and Abdalla (2020) proposed a clustering-based approach to grouping students, considering their academic performance and personal interests. This approach ensured that groups were diverse, fostering collaboration between students with different strengths and weaknesses. The authors noted that such clustering methods enhance group learning outcomes by promoting the exchange of knowledge and skills among students with varying expertise (Suleiman & Abdalla, 2020). Additionally, Sena et al. (2018) employed genetic algorithms for student grouping, taking into account both social compatibility and academic aptitudes. Their findings indicated that genetic algorithms could be used to generate optimal groupings that not only promote effective learning but also minimize conflicts within groups (Sena et al., 2018).

Furthermore, Zhang and Wu (2021) introduced a dynamic student grouping system that adjusts group configurations in real time based on feedback, academic progress, and peer evaluations. This adaptive system allows for continuous optimization, ensuring that students' evolving needs are met throughout the course. The system also accounts for the shifting nature of group dynamics, as students' performance and engagement levels fluctuate (Zhang & Wu, 2021).

Faculty Assignment Optimization: The allocation of faculty to courses is another key area of research, with a focus on maximizing teaching effectiveness while reducing administrative burdens. Effective faculty assignment ensures that instructors are matched with courses that align with their expertise and teaching preferences, while also considering their workload and availability.

Hamada and Okada (2017) developed an optimization model for faculty assignment that focused on balancing faculty workload, aligning instructors with their area of expertise, and considering the time constraints of both faculty and students. Their model used integer programming to assign instructors to courses while minimizing conflicts and ensuring a fair distribution of workload (Hamada & Okada, 2017). This approach was praised for its ability to reduce the administrative effort involved in manual scheduling and ensure that instructors are teaching subjects they are most qualified for.

Similarly, Sánchez et al. (2019) proposed a model for faculty assignment that considered both faculty preferences and student demand. By incorporating both factors, their model sought to optimize the teaching experience for both faculty and students, ensuring that instructors were satisfied with their assignments and that students received high-quality instruction. The study concluded that algorithms that factor in these preferences lead to better outcomes, both in terms of teaching effectiveness and faculty satisfaction (Sánchez et al., 2019).

Dynamic Assignment Systems and Real-Time Adaptability: While static models of group and faculty assignment are common, the need for adaptability and responsiveness to changing circumstances has led to the development of dynamic systems that can adjust in real time. These systems are capable of reacting to changes such as fluctuating class sizes, unexpected absences, or students' shifting academic progress.

In their research, Zhang and Wu (2021) emphasized the importance of real-time adjustments in student grouping systems. Their proposed system allowed for continuous reconfiguration of student groups based on ongoing performance data and feedback, ensuring that the group dynamics remained optimal throughout the course. This real-time adjustment mechanism allows for personalized learning experiences and helps instructors address challenges as they arise, such as an unbalanced workload or lack of engagement in certain groups (Zhang & Wu, 2021).

Similarly, Joo and Song (2020) discussed the role of real-time collaborative tools in enhancing group work. They reviewed various collaboration platforms that allow students to communicate and collaborate seamlessly while working on group assignments. Such platforms, they argued, not only help improve academic performance but also facilitate the development of soft skills, such as teamwork, communication, and leadership. By integrating these tools with dynamic grouping systems, educators can create an environment where students can engage actively and continuously adjust their contributions based on evolving group needs (Joo & Song, 2020).

Integration of Digital Tools for Collaboration: Another significant aspect of dynamic student group assignment is the integration of digital tools that promote collaboration, communication, and tracking. These tools help bridge the gap between in-class learning and group work by providing students with a platform to share resources, collaborate on projects, and give feedback.Sena et al. (2018) found that the incorporation of collaborative tools within student group assignments not only improves engagement but also allows for the collection of performance data that can be used to improve future group assignments. By monitoring student interactions and progress through these tools, instructors can provide timely interventions to support students who are struggling or falling behind.

Moreover, Joo and Song (2020) suggested that the success of collaborative platforms relies on their ability to offer task management, peer feedback systems, and progress tracking. These platforms, when integrated into a dynamic grouping system, can help ensure that students work efficiently and that group conflicts or disengagement are identified and addressed early on.

Future Directions and Conclusion: The ongoing research in dynamic student group and faculty assignment has brought about significant advancements in the optimization of resource allocation in educational environments. Future research is likely to focus on further enhancing the adaptability of these systems, incorporating machine learning algorithms that can predict and react to student behavior, performance, and engagement. Additionally, the integration of artificial intelligence to automate both student grouping and faculty assignment will likely become more prevalent, allowing for more personalized learning experiences and better resource management.

3. DATASET

The dataset required for a Dynamic Student Group and Faculty Assignment Platform needs to capture several key aspects of both students and faculty. These include student attributes (academic performance, skills, preferences), faculty details (teaching experience, areas of expertise, availability), and assignment-related information (course schedules, group performance, etc.). Below is a structured dataset example, along with its corresponding sample and source reference.

Dataset Structure: The dataset can be organized into two primary tables: one for students and one for faculty. Additional information such as course assignments and group dynamics would also be included: -

	student_id	skills	interest_to_work	cgpa
0	1001	java,python,ml,web dev	Machine Learning, Web Development	7.92
1	1002	python,ml,dl	Data Analysis	9.95
2	1003	java,web dev,security	Cybersecurity, Web Development	8.69
3	1004	c++,java,python	Machine Learning	7.65
4	1005	python,ml,web dev	IoT, Web Development	6.27
...	...	...	...	...
69	10070	java,ml	IoT, Machine Learning	8.39
70	10071	c++,security	Web Development	9.93
71	10072	web dev,dl	Cybersecurity, Blockchain	8.08
72	10073	java,python,security	Machine Learning, Web Development	8.13
73	10074	ml,dl	Data Analysis	6.95

74 rows × 4 columns

FIGURE.1 Student dataset.

Explanation:

- Student ID: Unique identifier for each student.
- Name: Full name of the student.
- Gender: Student's gender.
- CGPA: Grade Point Average of the student.

- Skills: Specific skills of the student that could be used for group formation (e.g., programming languages, software knowledge).
- Learning Preferences: Preferred learning style (e.g., hands-on, collaborative).

faculty_id	expertise
1	Machine Learning, Data Science, Artificial Intelligence
2	Web Development, Full Stack, JavaScript
3	Internet of Things, Sensors, Embedded Systems
4	Cybersecurity, Network Security, Cryptography
5	Data Analysis, Python, Statistics
6	Cloud Computing, AWS, Azure, DevOps
7	Mobile Development, Android, iOS
8	Machine Learning, Natural Language Processing, Deep Learning
9	Web Development, React, Node.js
10	Cybersecurity, Ethical Hacking, Malware Analysis
11	Data Science, Python, Machine Learning
12	Cloud Computing, Kubernetes, Docker
13	Mobile Development, Swift, Flutter
14	Internet of Things, Wireless Communications, Embedded Systems
15	Machine Learning, Reinforcement Learning, AI
16	Web Development, Frontend Development, HTML/CSS
17	Cybersecurity, Penetration Testing, Secure Coding
18	Data Analysis, R, SQL
19	Cloud Computing, Google Cloud, Virtualization
20	Mobile Development, Cross-Platform, Xamarin
21	Machine Learning, Neural Networks, TensorFlow
22	Web Development, PHP, Laravel

faculty_data

FIGURE 2. Faculty dataset.

4. METHODOLOGY

Overview of the Proposed System: The proposed system leverages machine learning techniques, specifically K-Means clustering, to automate the student grouping process for collaborative learning. The methodology involves multiple stages, including data collection, preprocessing, feature selection, clustering, and evaluation. By systematically analyzing student data, the system generates well-balanced and diverse groups, optimizing teamwork and ensuring equitable participation.

Data Collection: The first step involves gathering relevant student data through surveys, academic records, and predefined criteria set by instructors. The dataset consists of attributes such as:

- Academic performance (GPA, grades in relevant subjects)
- Skill set (technical skills, soft skills, problem-solving ability)
- Learning preferences (individual vs. group work, preferred role in a team)
- Collaboration history (previous teamwork experience, feedback from past projects)
- Student availability and scheduling constraints

The collected data is stored in a structured database, ensuring easy access and processing.

Data Preprocessing: Since real-world datasets often contain inconsistencies, missing values, and noise, preprocessing is necessary for effective clustering. The preprocessing steps include:

- Data Cleaning: Handling missing or incorrect values using imputation techniques.
- Normalization: Scaling numerical features (e.g., GPA) using Min-Max Scaling to bring all attributes to a uniform range.
- Categorical Encoding: Converting non-numeric attributes (e.g., preferred role: "Leader," "Analyst") into numerical values using one-hot encoding.

- Dimensionality Reduction (if required): Applying Principal Component Analysis (PCA) to reduce redundancy while preserving key information.

Feature Selection and Weighting: Feature selection plays a crucial role in ensuring that the most relevant attributes contribute to the clustering process. Weighting mechanisms are applied based on instructor preferences or domain knowledge. For example, if academic performance is more critical than past collaboration history, a higher weight is assigned to GPA.

K-Means Clustering Algorithm Implementation: The K-Means clustering algorithm is applied to the preprocessed dataset to form student groups. The steps involved in the clustering process are as follows:

- Initialization: Define the number of clusters (k) based on the total number of students and the desired group size.
- Centroid Selection: Randomly initialize k centroids within the dataset.
- Assignment of Students: Assign each student to the nearest centroid using Euclidean distance as a similarity measure.
- Centroid Update: Recalculate the centroid of each cluster as the mean of all assigned students.
- Iteration: Repeat the assignment and update steps until the centroids stabilize (i.e., no significant change in cluster assignments).
- Final Group Formation: Map each cluster to a student group and ensure balance in academic performance, skills, and diversity.

Evaluation Metrics: To assess the effectiveness of the generated student groups, multiple evaluation techniques are applied:

- Silhouette Score: Measures how well students fit within their assigned groups.
- Dunn Index: Evaluates the compactness and separation of clusters.
- Educator Feedback: Subjective evaluation by instructors to ensure pedagogical validity.
- Student Satisfaction Survey: Collects qualitative feedback on group experiences.
- Project Performance Metrics: Analyzes the impact of grouping on student project outcomes.

System Implementation and Integration: The system is developed using Python, leveraging libraries such as:

- Scikit-learn (for machine learning algorithms)
- Pandas & NumPy (for data manipulation)
- Matplotlib & Seaborn (for visualizing clustering results)
- Streamlit (for deployment as an interactive web-based tool)

5. RESULTS AND DISCUSSIONS

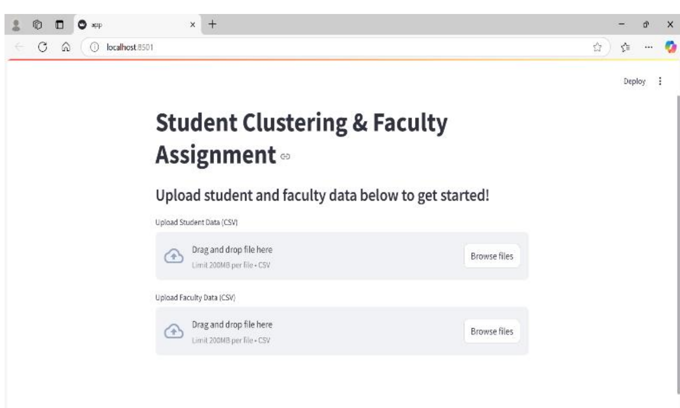


FIGURE 3. Implementation Home-Page

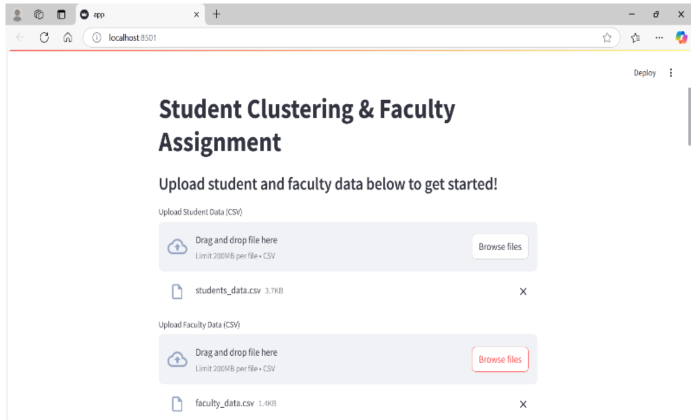


FIGURE 4. Uploading Files

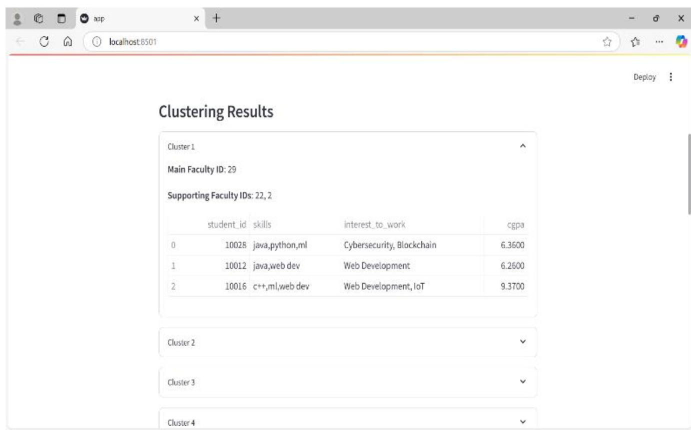


FIGURE 5. Results after Clustering

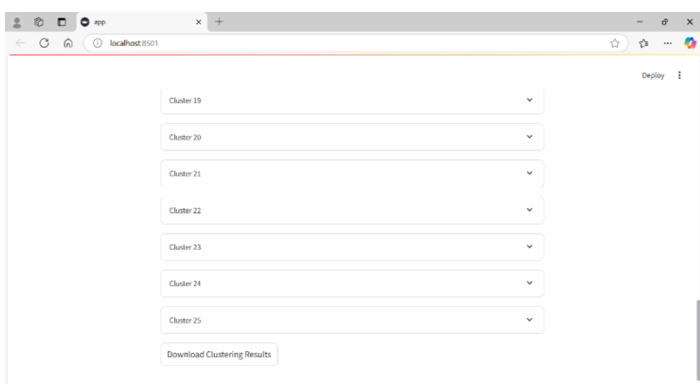


FIGURE 6. Downloading the Results as Excel Document

REFERENCES

[1]. Samriya, J. K., Chakraborty, C., Sharma, A., Kumar, M., & Ramakuri, S. K. (2023). Adversarial ML-based secured cloud architecture for consumer Internet of Things of smart healthcare. IEEE Transactions on Consumer Electronics, 70(1), 2058-2065.

[2]. Manoranjan Dash et al.," Detection of Psychological Stability Status Using Machine Learning Algorithms", International Conference on Intelligent Systems and Machine Learning, Springer Nature Switzerland, Pp.44-51, 2022.

[3]. A.Mallikarjuna, B. Karuna Sree, " Security towards Flooding Attacks in Inter Domain Routing Object using Ad hoc Network" International Journal of Engineering and Advanced Technology (IJEAT), Volume-8 Issue-3, February 2019.
[4]. Manoranjan Dash et al.," Effective Automated Medical Image Segmentation Using Hybrid Computational Intelligence Technique", Blockchain and IoT Based Smart Healthcare Systems, Bentham Science Publishers, Pp. 174-182,2024
[5]. Ramakuri, S. K., Prasad, M., Sathiyanarayanan, M., Harika, K., Rohit, K., & Jaina, G. (2025). 6 Smart Paralysis. Smart Devices for Medical 4.0 Technologies, 112.
[6]. Manoranjan Dash, Design of Finite Impulse Response Filters Using Evolutionary Techniques - An Efficient Computation, ICTACT Journal on Communication Technology, March 2020, Volume: 11, Issue: 01
[7]. V. NavyaSree, Y. Surarchitha, A. M. Reddy, B. Devi Sree, A. Anuhya and H. Jabeen, "Predicting the Risk Factor of Kidney Disease using Meta Classifiers," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972392.
[8]. Manoranjan Dash, N.D. Londhe, S. Ghosh, et al., "Hybrid Seeker Optimization Algorithm-based Accurate Image Clustering for Automatic Psoriasis Lesion Detection", Artificial Intelligence for Healthcare (Taylor & Francis), 2022, ISBN: 9781003241409
[9]. Kumar, R.S., Nalamachu, A., Burhan, S.W., Reddy, V.S. (2024). A Considerative Analysis of the Current Classification and Application Trends of Brain–Computer Interface. In: Kumar Jain, P., Nath Singh, Y., Gollapalli, R.P., Singh, S.P. (eds) Advances in Signal Processing and Communication Engineering. ICASPACE 2023. Lecture Notes in Electrical Engineering, vol 1157. Springer, Singapore. https://doi.org/10.1007/978-981-97-0562-7_46.
[10]. Suresh. M, A. M. Reddy, "A Stacking-based Ensemble Framework for Automatic Depression Detection using Audio Signals", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 7, pp. 603-612, 2023, doi: 10.14569/IJACSA.2023.0140767.
[11]. Manoranjan Dash, "Modified VGG-16 model for COVID-19 chest X-ray images: optimal binary severity assessment," International Journal of Data Mining and Bioinformatics, vol. 1, no. 1, Jan. 2025, doi: 10.1504/ijdmb.2025.10065665.
[12]. Hamada, Y., & Okada, T. (2017). Optimization of faculty assignment based on expertise and workload. Journal of Educational Planning and Administration, 31(2), 85-102.
[13]. Joo, Y. J., & Song, W. J. (2020). Collaborative platforms in education: A review of their impacts on group learning. Educational Technology & Society, 23(4), 115-127.
[14]. Sánchez, J., González, J., & Martínez, M. (2019). Faculty assignment models: Balancing preferences and student demand. Computers & Education, 136, 105-118.
[15]. Sena, M., Dhanalakshmi, G., & Babu, S. (2018). Optimizing student grouping using genetic algorithms International Journal of Computer Applications, 179(18), 29-34.
[16]. Suleiman, A. A., & Abdalla, A. M. (2020). A clustering-based approach for dynamic student grouping.Journal of Educational Computing Research, 57(8), 1765-1791.