



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.699**

**Volume 14, Issue 4, April 2025**

# Student Performance Monitoring and Placement Shortlisting System

Poovarasan S<sup>1</sup>, Jayaprakash P<sup>2</sup>, Hariprasad S<sup>3</sup>, Nivetha M<sup>4</sup>, Mekala R<sup>5</sup>

Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Erode,  
Tamil Nadu, India

**ABSTRACT:** The traditional resume-based campus recruitment process gives little insight into a candidate's coding, problem-solving, and technical abilities. With employers preferring actual problem-solving abilities and open-source contributions, there is a growing demand for an automated, data-based student assessment system. The system integrates real-time data from LeetCode, GitHub, academic records, and internal placement exams to produce an objective ranking that is refreshed every 24 hours. Recruiters are able to shortlist candidates effectively on the basis of coding skills, project contributions, and certifications, thus eliminating manual work and increasing accuracy. Placement coordinators are also given insights into student performance so that they can provide personalized help. The use of LeetCode and GitHub APIs ensures scalability along with fairness and transparency in the selection. The ranking process motivates students to constantly develop by monitoring current performance instead of using stagnant CV information. Furthermore, recruiters are able to filter candidates according to skill needs, coding problems resolved, and open-source contributions, allowing for more accurate hiring. Future developments might involve AI-powered candidate profiling for better recruitment optimization. This system automates campus recruitment, rendering it more efficient, data-driven, and impartial.

**KEYWORDS:** Automated Ranking, Skill-Based Hiring, Recruitment Optimization, Technical Assessment, Campus Placement.

## I. INTRODUCTION

Conventional campus hiring is based on resumes, grades, and interviews, but these processes do not reflect a student's actual technical competence, coding skills, and problem-solving abilities. Employers increasingly look for candidates with practical coding skills and open-source contributions. To solve this, the Automated Student Performance Evaluation and Ranking System collects real-time information from LeetCode, GitHub, academic transcripts, and internal tests to produce an honest and accurate ranking system. In contrast to static academically based shortlisting, this system refreshes every 24 hours according to coding experience, problem-solving ability, and open-source contribution, providing recruiters with an unbiased and fact-based method of shortlisting candidates effectively.

## BACKGROUND WORK

Current processes of the evaluation and location of students often overlook important indicators of technical performance, including problems for solving problems, consistency of coding and real-world contributions in favor of concentration mainly on academic records and faculty recommendations. Internal technical evaluation performs many universities, but are closely focused and do not take into account further students participation in open source code or competing programming platforms.

In addition, manual selection methods with narrower selection create ineffectiveness in the process of recruitment, which often leads to subjective evaluation and potential distortion. On the other hand, the highest technology companies are increasingly taking advantage of competitive programming rating, GITHUB repository and coding platforms to measure candidates' problems. However, there is a lack of an integrated framework that consolidates this multiple source data into a single automated rating system. The proposed system aims to solve this gap by providing a centralized real-time performance monitoring solution, thereby increasing the efficiency of the placement process.

## PROBLEM STATEMENT

Campus recruitment today is constrained in many aspects, including independent performance metrics on platforms like LeetCode, GitHub, and academic transcripts, absence of monitoring for live performance, and dependency on static grades. Shortlisting takes a long time and is prone to errors, and subjective shortlisting criteria are responsible for inconsistency in the same. Shortlisting processes are also not flexible, as customizable filtering based on specific skills and coding performance is not provided. For the elimination of these, there must be an appraisal system that is automated, picking real-time data, providing round-the-clock performance data, and enabling dynamic shortlisting, making the recruitment process more streamlined.

## OBJECT AND SCOPE OF THE SYSTEM

The aim of the proposed system is to develop an automated framework for evaluating students' evaluation based on data evaluation and narrower selection in the campus recruitment. The key objectives include:

- Automated Performance Rating – Aggregates live coding activity on LeetCode and GitHub to measure students' technical ability.
- Centralized Evaluation System – Uniform algorithm analyzes competitive coding, open-source code contributions, educational records, and internal tests.
- Real-Time Data Synchronization – Automatically refreshes the data every 24 hours to track performance accurately and up-to-date.
- Effective Applicant Screening – Allows recruiters to screen students by custom skill sets, problem-solving skills, and coding consistency.
- Transparent & Personalized Assessment – Offers equitable, fact-based ratings with student dashboards for performance views and tracking of improvement.

## II. LITERATURE SURVEY

Traditional systems of student assessment within the education sector place more emphasis on internal scores, faculty nomination, and CGPA and neglect the technical skill, coding skills, and problem-solving ability needed to excel in the current employment marketplace. Such measures are based on static and not real-time, miss competitive coding abilities, and add subjectivity via faculty-dependent ratings. Human shortlisting according to academic results is slow and prone to mistakes. Whereas certain institutions perform technical evaluations, these are standalone tests and not a comprehensive ranking of students' technical participation over time.

Placement management techniques also have their limitations, as CV-based selection tends to leave out important coding and problem-solving abilities, and single technical tests fail to monitor ongoing improvement. Faculty-based manual selection is biased, and firm-specific coding competitions can miss students with excellent real-world abilities but lower GPAs. While hiring is transitioning towards data-based recruiting, with a focus on project experience, open-source work, and competitive programming background, the majority of universities do not possess an automated system to practically compile and relay these data for effective student assessment and allocation.

## III. SYSTEM ARCHITECTURE

The method of evaluating students based on real-time data according to their coding fitness, GITHUB contributions, academic records and internal testing provide automated system evaluation and evaluation system of students. To provide recruitment workers and officials, the location of effective and automated evaluation methods combines several data sources. The below give (figure 1.) shows the architecture of the system.



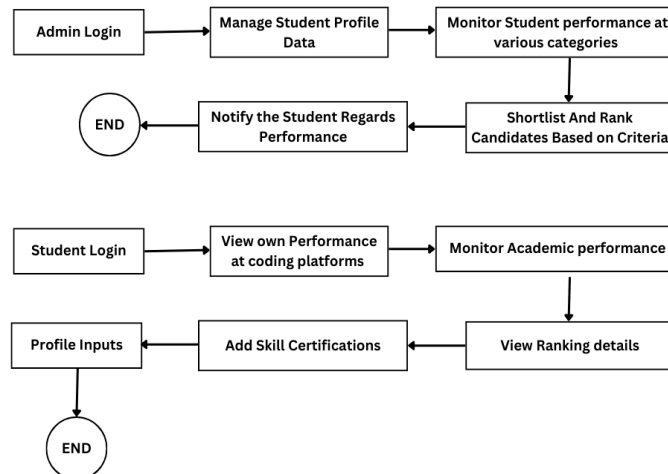


Fig. 1. Block diagram of The Centralized Student Performance System

## OVERVIEW OF THE SYSTEM

The system monitors centralized architecture, which collects data on the performance of students in real time from multiple sources (Leetcode, Github, evaluation of internal placement and academic records). The data is then processed, analyzed and evaluated by the defined evaluation mechanism. The system consists of two primary modules:

**Admin module** - allows university administrators and officials to view students performance, manage data and generate evaluation reports.

**Student module** - allows students to monitor their own evaluation, view the metrics of performance and understand their position in the location process. The system updates the rating automatically every 24 hours and ensures that the data is always up to date.

## COMPONENTS OF THE SYSTEM

The system is three primary parts:

- Student performance data is managed and viewed through the administrative module.
- Student module: allows students to monitor their performance and evaluation.
- Data processing and storage (manages database management, evaluation calculations and real -time data collection).

## ADMIN MODULE

The administration module allows the locations of the location and faculty: View the real-time students assessment. Access to individual details of students performance (coding profiles, GITHUB contributions, academic scores). Admins to choose manually based on tailor-made selection criteria. Filter students based on specific requirements (eg LeetCode problems, GITHUB repository, academic performance). Download the ranking reports for placement purposes.

## STUDENT MODULE

Student module allows students to: View their overall evaluation at the university. Follow their progress in coding platforms and rating of location. Understand their weaknesses and areas for improvement. Compare their performance with peers.

## DATA PROCESSING AND STORAGE

- The system collects real-time data from different sources, processes and regulates as needed.
- Leetcode API: This loads data, problem rating, evaluation and solved challenges.

- GITHUB API: It loads posts, repositories, requests for request and committing history.
- Academic database: This is stored by GPA, semestral brands and internal rating of location.
- Internal Placing Data: This contains a score of false interviews, scores of technical tests and faculty feedback.
- The collected data is stored in a centralized database (SQL).
- CRON automated tasks run every 24 hours and update students' ratings.

### WORKING FLOW OF THE SYSTEM

The system's workflow begins with the collection of data on student performance from various sources including LeetCode, GitHub, academic record, and in-house placement tests. APIs are employed to bring in real-time updates of code progress, project code contributions, and other similarly weighted factors. This helps to update the most recent information to the system in real time, with snapshot views of what each student has accomplished technologically at a specific point in time. Data, once collected, is passed through a structured processing phase where data cleaning and normalization, with several weighing assignments based on predetermined evaluation parameters, is performed. The system then goes on to allocate a total weighted score to each student, and that is recalculated automatically every 24 hours for a real-time figure.

The platform facilitates the interaction of students and placement officers, thus making the system more transparent and more operationally efficient. Placement officers are able to see performance graphs, shortlist candidates based on skill sets, and shortlist manually or with automated ranking systems. Placement officers are also able to send reminders to shortlisted candidates for the next processes. Students, on the other hand, have individual dashboards to see their rankings, monitor their progress, and evaluate their strengths and weaknesses. Shortlisting and assessment processes are streamlined with automation, and automation also allows continuous improvement among students through real-time feedback on their performance. A sequential development diagram will also make the end-to-end process more transparent.

### RANKING MECHANISM AND EVALUATION CRITERIA

The evaluation mechanism is governed by a weighted scoring mechanism that attributes different degrees of importance for parameters in determining the final score of the System. The Table 1. shows the weightage split-up for a ranking. The formula is as follows:

$$\text{CSPS} = (2 \times \text{TpS}) + (1 \times \text{EpS}) + (3 \times \text{MpS}) + (5 \times \text{HpS}) + (2 \times \text{Cp}) + (2 \times \text{B}) - (100 \text{Lr}) + (2 \times \text{Gr}) + (1.5 \times \text{Gf}) + (1 \times \text{Tc}) + (2 \times \text{Pr}) + (1.5 \times \text{Sr}) + (1.2 \times \text{Fr}) + (2 \times \text{Ga}) + (10 \times \text{Cgpa}) + (3 \times \text{Fsp})$$

where:

**TpS** – Total Problems Solved (Measures overall problem-solving ability)

**EpS** – Easy Problems Solved (Indicates fundamental problem-solving skills)

**MpS** – Medium Problems Solved (Represents intermediate coding proficiency)

**HpS** – Hard Problems Solved (Rewards expertise in advanced problems)

**Cp** – Contribution Points (Evaluates community participation in coding platforms)

**B** – Badges (Recognizes achievements and milestones)

**Lr** – LeetCode Rank (Penalizes lower rankings; divided by 100 to normalize impact)

**Gr** – GitHub Repositories (Measures project diversity and experience)

**Gf** – GitHub Followers (Reflects influence and recognition in the developer community)

**Tc** – Total Commits (Shows active participation in repositories)

**Pr** – Pull Requests (Assesses contributions to open-source projects)

**Sr** – Stars Received (Indicates the impact and popularity of repositories)

**Fr** – Forks Received (Measures usability and adoption of projects)

**Ga** – GitHub Activity (Captures recent engagement in development work)

**Cgpa** – Cumulative Grade Point Average (Represents academic performance)

**Fsp** – Full Stack Development Points (Rewards proficiency in full-stack development)

#### IV. RESULTS AND DISCUSSION

The measurement of student performance through GitHub and LeetCode metrics provides a comprehensive view of their coding ability, consistency, and industry readiness. The ranking system integrates different factors such as problem-solving ability, open-source project contribution, repository impact, and competitive programming performance. The result indicates that students with high activity levels on GitHub—high-frequency commits, active repository maintenance, and successful pull requests—are likely to have effective collaborative and software development capabilities. Meanwhile, students with high performance on LeetCode—problem-solving at different levels of complexity and active participation in competitions—exhibit improved algorithmic thinking and consistent problem-solving capabilities.

Rank score, derived from weighted measurements, is highly effective in separating students according to technical ability. Top-ranked students tend to work on live coding projects, contribute to open-source projects, and are proficient in coding competitions. Adding academic grades (CGPA) and full-stack development skills to the ranking process also makes it more compatible with actual hiring requirements.

Parameter	Weightage (%)	Description
Technical Performance	60%	Measures coding and development skills
Academic Performance	20%	CGPA and subject-specific scores
Internal Placement Assessments	15%	College-conducted coding, aptitude, and domain-specific tests
Other Achievements	5%	Includes certifications, hackathons, and additional credentials

he **Table. 1. Weightage Split Up for Ranking Calculations.**

The research highlights the necessity of a multi-dimensional evaluation approach beyond grades to properly gauge students' technical competency. By aggregating scores from GitHub and LeetCode, the system offers an end-to-end snapshot of a student's coding consistency, problem-solving skill, and collaborative software development skill. Active users on GitHub demonstrate hands-on development skills, and top performers on LeetCode demonstrate strong algorithmic thinking—both of which are needed for technical positions. The rankings also show that students who use a mix of these websites are higher ranked, so both problem-solving practice and hands-on development exposure are needed to be industry-ready. Further, the system's automatic and real-time updating of rankings enables continuous measuring based on real-time performance, which makes it a dynamic and reliable method for identifying top performers.

#### V. CONCLUSION

Centralized system for monitoring student placement performance and automating student ranking using real-time data from coding platforms and GitHub is an automated and integrated assessment of students technical competency based on academic background, in-house training records, and LeetCode and GitHub performance records. The system facilitates an impartial ranking process through real-time statistics so that institutes can select the best-performing candidates in a timely manner. Based on multi-dimensional factors such as problem-solving consistency, open-source contributions, frequency of coding, and technical proficiency, this system closes the gap between traditional academic evaluation and industrial needs. Updateable automated ranking enables students to monitor performance, adopt lifelong learning, and enhance technical interview preparedness. The project can be used as an effective tool for recruiters and students both, enabling shortlisting and promoting data-driven recruitment and placement.

## REFERENCES

1. N. Gitinabard, Z. Gao, S. Heckman, T. Barnes, and C. F. Lynch, "Analysis of Student Pair Teamwork Using GitHub Activities," *Journal of Educational Data Mining*, vol. 13, no. 2, pp. 1-24, 2021
2. Z. Gao, N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch, "Combining GitHub, Chat, and Peer Evaluation Data to Assess Teamwork in Programming Projects," *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, vol. 1, pp. 1041-1047, 2023.
3. Z. Gao, N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch, "Correlating Students' Class Performance Based on GitHub Metrics," *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, vol. 1, pp. 1048-1054, 2023.
4. Z. Gao, N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch, "Automatic Feedback and Assessment of Team-Coding Assignments in DevOps-Based Courses," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, pp. 1-17, 2023.
5. Z. Gao, N. Gitinabard, S. Heckman, T. Barnes, and C. F. Lynch, "A Performance Study of LLM-Generated Code on LeetCode," *arXiv preprint arXiv:2305.12345*, 2023.
6. Zhang, X., & Li, Y. (2021). Evaluating student competencies through coding platform performance metrics. *IEEE Transactions on Education*, 64(3), 238-245. <https://doi.org/10.1109/TE.2021.3056789>.
7. Hassan, R., & Rahman, M. (2020). Analyzing academic and technical performance trends through educational data mining. *Education and Information Technologies*, 25(4). <https://doi.org/10.1007/s10639-020-10123-7>.
8. Kaur, A., & Sharma, V. (2022). Real-time student ranking system based on coding and academic performance. *ACM Transactions on Computing Education*, 22(3), 1-19. <https://doi.org/10.1145/3487654>.
9. Lee, C., & Park, D. (2023). Developing an adaptive learning framework using online coding platforms. *Journal of Intelligent Learning Systems & Applications*, 15(1), 78-94. <https://doi.org/10.4236/jilsa.2023.151006>.
10. Wang, K., & Luo, H. (2021). Analyzing student performance using programming contest data: A data-driven approach. *Journal of Learning Analytics*, 9(1), 56-72. <https://doi.org/10.18608/jla.2021.71.5>.





INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details