Study Buddy – A Web-Based Integrated Academic and Career Development Assistant for Engineering Freshmen

[1]Om R. Karanje [0009-0005-0165-5302], [2]Vishal S. Khade [0009-0001-7784-6034], [3]Sujay G. Karad [0009-0005-2721-7183],

[4]Tejas A. Kanade [0009-0003-2695-6908], [5]Prof. Supriya S. Gorde [0009-0004-2489-8561]

[1]Engineering Student, [2]Engineering Student, [3]Engineering Student, [4]Engineering Student, [5]Professor
[1]Computer Science Engineering(Data Science),
[1]Vishwakarma Institute of Technology, Pune, India
[1]om.karanje24@vit.edu, [2]vishal.khade24@vit.edu, [3]sujay.karad24@vit.edu,
[4]tejas.kanade24@vit.edu, [5]supriya.mahatme@vit.edu

Abstract—The move between secondary education and an undergraduate engineering education is a big challenge to the first-year students (freshers) which is usually marked with academic stress, time mismanagement, and little awareness of the employability skills as per industry standards. The current Learning Management Systems (LMS) mainly aim at delivering content without paying much attention to the comprehensive growth of the student. In this paper, the author introduces the concept of personalized and web-based intelligent assistant called Study Buddy that will help to fill this gap. The system was developed in a serverless architecture in JavaScript and Google Firebase and includes four key modules: an Automated Study Planner to schedule study sessions dynamically with the college timetable of the user; an Employability Assessment Engine (Am I Employable?), who evaluates soft skills; a centralized academic repository; and a portal to guide the career. The software is a lightweight Single Page Application (SPA) framework to make sure that its latency is low, and its data is synchronized in real-time. The results of the testing show that the platform has been effective at assisting students to plan their study time more efficiently and diagnose gaps in skills at the start of their academic years, which contributes to academic and career preparation.

Index Terms—Engineering Education, Web-Based Learning, Firebase, Automated Study Planner, Soft Skills Assessment, Serverless Architecture, Student Productivity, Personalized Learning

I. INTRODUCTION

The shift of secondary school to undergraduate engineering curriculum is usually marked by a lot of academic and psychosocial difficulties. The so-called freshers are first-year engineering students, often exposed to a phenomenon termed as transition shock, involving the differences in high school anticipations and university rigour causing academic underperformances [1], [3]. Engineering courses demand high self-regulated learning and autonomy unlike the school setting where a well-defined environment is in place.

Effective time management is one of the main factors that can be determined as a success in this environment. According to a study conducted by Adams and Blair [2], there is a high relationship between time management behavior and Grade Point Average (GPA) among the engineering students. Nevertheless, research reveals that most freshers have moderate to poor planning abilities which usually cause stress and subsequent drop-out [6].

Moreover, contemporary engineering sector requires not only technical skill. Employers are progressively putting more focus on what has been referred to as employability skills, such as communication, teamwork and critical thinking [5]. Although this is the case, these soft skills are usually not incorporated in traditional engineering programs and this poses a problem in terms of graduate competencies and the demands of the industry [4].

As a solution to these complex issues, this paper proposes a personal assistant called "Study Buddy" that is a web-based and all-purpose tool designed with freshmen in the engineering field in mind. With the help of serverless web technologies, the platform can combine the management of academic resources, automated study planning, and soft-skill evaluation into a single dashboard.

II. LITERATURE REVIEW

The topic of inclusion of web technologies in engineering education is 20 years old. The possibility of using web-based portals to democratize the access to learning materials was first mentioned in early studies by Sheikh and Siddiqui [7]. LMS have since developed to become the norm in higher education [8]. Majority of the available LMS solutions, however, are solely academic platforms such that they only offer the delivery of notes and assignments but do not cater to the holistic growth of the student.

Individual Planning and Direction: The recent advancements in the educational technology have tried to automate the planning process. Sharma [16] suggested AI-based study planners to assist the students in planning their time. Nevertheless these systems tend to work independently and do not take into consideration the predetermined limitations of a university schedule. On the same note, although there have been proposals on the use of digital career guidance systems to assist students in choosing their career paths on specialization [13], [15], they have seldom been incorporated in the daily academic routine of the student.

Technological Framework: The transition to Serverless Computing has also allowed the creation of lightweight and scalable education applications [10]. The conventional SQL-based systems are usually costly to maintain and infrastructure. On the contrary, the NoSQL systems, such as Google Firebase, have become a better choice to use in the real-time application [11]. Comparative studies show that Firebase is superior to conventional databases such as MongoDB in applications that need real time data synchronization and user authentication [12], which in turn makes it a perfect backend solution to an application such as Study Buddy that will be student-centered [9].

Research Gap: Although there are individual solutions to study planning [16], career guidance [14] and resource distribution [8], it can be observed that there has been a significant dearth of combined platforms integrating all these features specifically to the content of the fresher demographic. Study Buddy is intended to fill this gap by providing a single and customizable portal that will fulfill the academic quality and the employability needs of contemporary engineering education.

Comparative Analysis of Existing Solutions, the existence of different platforms that tackle single issues of student development is not enough as they lack a holistic one. A comparative analysis of the Study Buddy and the conventional market solutions is provided in Table 1.

Feature	Standard LMS	Task Managers	Career Portals	Study Buddy (Proposed)
Primary Focus	Content Delivery	Task Scheduling	Job Hunting	**Holistic Development**
Algorithm Type	Static / Manual	Manual Input	Recommendations	**Dynamic Allocation**
Soft Skill Tracking	NO	NO	YES	**YES**
Timetable Integration	NO	Manual	NO	**Automated**
Target Audience	General Students	General Public	Professionals	**Engineering Freshers**

Table 1: Comparative Analysis with Existing Solutions

As it is seen, Task Managers are more effective at scheduling, but they do not have the academic background of an LMS. On the other hand, LMSs do not have the one-on-one career advice available on professional portals. Study Buddy is a combination of these areas into one architecture.

III. PROPOSED METHODOLOGY

The suggested system of the so-called Study Buddy is a Single Page Application (SPA) that relies on a serverless architecture. This strategy reduces overhead in managing infrastructure and at the same time high scalability and real time data synchronization is provided. The four basic modules are iteratively testable using the Agile development methodology.

System Architecture: The system follows a three-level architecture, which is a Presentation Layer (Client), Application Logic Layer (JavaScript Controllers), and Data Layer (Firebase Backend). This project is based on a "Serverless" computing model [10], unlike the traditional SQL-based Learning Management Systems, which need specific server maintenance.
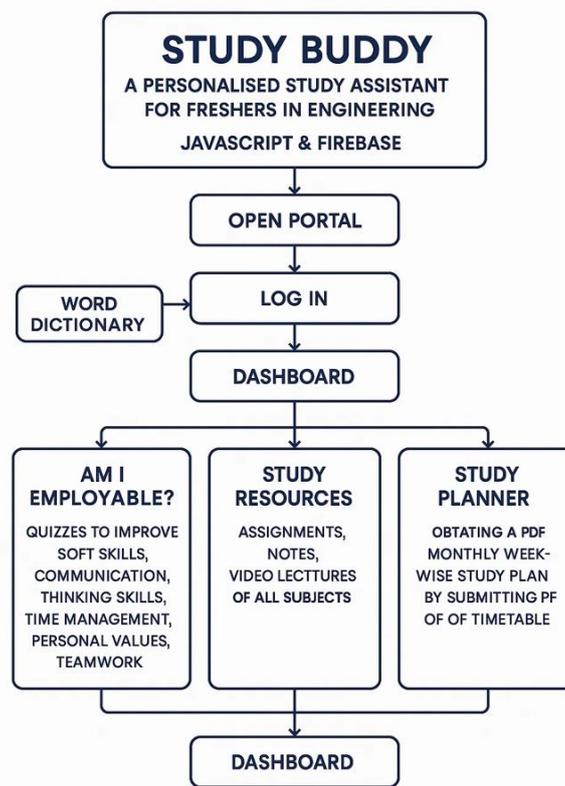
Fig 1: Workflow of Study Buddy

Frontend: It was created with HTML5, CSS3, and JavaScript (ES6) programming languages. This interface has been created to be responsive and it is therefore accessible on devices.

Backend-as-a-Service (BaaS): The system utilizes the use of Google Firebase as the backend. This option can be justified using the research results suggesting that Firebase can save much time in web application development in comparison to custom implementations of servers [9], [11].

Database and Authentication Design: Firebase Realtime Database (or Cloud Firestore), a NoSQL cloud database, addresses the issue of data persistence. Firebase uses JSON object to store data compared to relational databases (SQL).

Justification: Ghaffar et al. [12] showed that NoSQL database such as Firebase is better than Mongo DB and SQL where hierarchical data such as student profiles and study schedules are needed in a short period of time.

Authentication: User identity is handled through Firebase Authentication. A student is issued with a unique User ID (UID) when he/she registers. This UID will serve as the primary key to access personalized information (study plans, quiz scores) in the database, which is needed to guarantee the privacy and security of data [9].

NoSQL Data Modeling: The lack of a schema in the Firebase Realtime Database makes the application very efficient. Our data model is based on a nested JSON tree format as opposed to a complex join in relational schema because relational schema involves more read latency.

A. User Profile Schema: The root node users has child nodes that are determined by the unique auth UID. It will be designed in such a way that all the necessary data is retrieved during the login process in one query.

```
{
  "users": {
    "uid_12345": {
      "profile": {
        "name": "Student Name",
        "branch": "Computer Engineering",
        "semester": 1
      },
      "employability_scores": {
        "communication": 75,
        "teamwork": 60,
        "critical_thinking": 80
      },
      "preferences": {
        "difficult_subjects": ["Maths-1", "Mechanics"],
        "productive_hours": ["18:00", "21:00"]
      }
    }
```

```
      }
  }
```

B. Study Plan Schema: Instead of storing fixed dates, the planner provides the templates of days to enable the repetitive schedules. This saves the size of the database and enables the algorithm to recreate the dates automatically.

```
{
  "weekly_plan": {
    "monday": {
      "college_hours": ["09:00-17:00"],
      "study_slots": [
        {"time": "18:00-19:00", "subject": "Maths-1", "type": "Deep Work"},
        {"time": "20:00-20:30", "subject": "BEE", "type": "Revision"}
      ]
    }
  }
}
```

Logic Implementation: This logic assists in defining functionalities by integrating modules that constitute the day-to-day operations of the prison environment.
Module Implementation Logic:

A. The "Am I Employable?" Assessment Engine: In order to solve the soft skills gap highlighted by Idkhan et al. [4], this module applies weighted scoring algorithm.
Logic: The customer responds to a set of psychometric questions that are projected to the main employability characteristics (Teamwork, Communication, Leadership).
Processing: JavaScript functions receive the input and compute the overall score of each trait and compare the result with a predefined threshold known as Industry Standard.
Output: The information is presented on the dashboard by Chart.js, which gives the student a graphical depiction of his/her strengths and weaknesses in real-time [18].

B. Intelligent Study Planner (ISP): This module aims at covering the burning issue of time management [6]. Implementation does not remain confined to the fixed calendars but it adds a customized allocation algorithm [16].
Input: The customer uploads/ enters his/her college schedule (determined limitations).
Processing: The algorithm detects free blocks (e.g. 6:00 PM - 9:00 PM). It then allocates the challenging content (based on user preference) to these high-availability slots.
Storage: The created plan is stored as a JSON object connected to the profile of a user, which may be dynamically updated in case of the schedule change.

C. Academic and Career Repository:
Resources: This module is a content aggregation engine. Videos (lectures), PDF materials are classified in the database. These features allow an asynchronous query to retrieve the pertinent Uniform Resource Locators (URLs) when the user clicks on a subject without reloading the page [8].
Career Guide: A knowledge base that is not dynamic and gives us road maps in different fields of engineering. Students prefer digital career guidance systems because, as Patil et al. [15] suggest, it is easily accessible and anonymous.
User Interface (Dashboard): The dashboard is the main control unit. It employs an easy to use UI/UX to minimise cognitive load. After the successful login, the dashboard retrieves the User_Object of Firebase and renders the particular "Daily Plan" and "Progress Metrics" of that session.
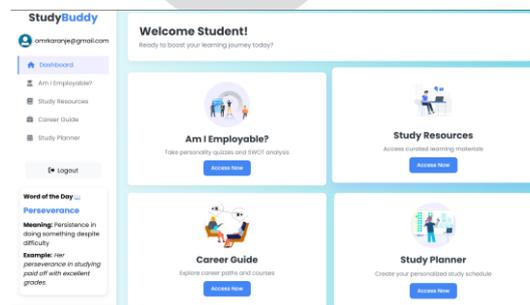

Fig 2: Main Dashboard

IV. ALGORITHMS AND IMPLEMENTATION LOGIC

The fundamental action of the "Study Buddy" system is based on two main algorithms, namely the Soft Skill Evaluation Engine and the Automated Schedule Allocation Algorithm.

Soft Skill evaluation Algorithm: The Am I Employable? module The module measures qualitative characteristics (communication, teamwork) on a numerical scale. It is possible to track the progress of users over a long period.
The set of categories is denoted by C (e.g. Communication, Leadership, Critical Thinking). Cc is a set of questions Qc. The weight w_i of each question q_i is assigned, dependent on the difficulty, and user response value r_i is shown on a Likert scale (1-5).
The normalized weighted sum is obtained as the category score Sc:

$$S_c = \frac{\sum_{i=1}^{n}(w_i \times r_i)}{\sum_{i=1}^{n}(w_i \times 5)} \times 100$$

The Final Employability Index E_{index} is the mean of all category scores:

$$E_{index} = \frac{1}{|C|} \sum_{c \in C} S_c$$

If E_{index} < Threshold (e.g., 60%), the system triggers a recommendation function recommendResources(c) to fetch specific articles from the database related to the weak category.

Automated Study Plan Generation Algorithm:

The most novel component of the system is the "Intelligent Study Planner." This algorithm parses the user's fixed constraints (college hours) and dynamically allocates study sessions for the remaining "free" hours.

Algorithm 1: Dynamic Slot Allocation

Input:
- U_{table}: User's college timetable (Fixed Busy Slots).
- S_{list}: List of subjects {Sub_1, Sub_2, ... Sub_n}.
- D_{level}: Difficulty rating for each subject (High/Medium/Low).

Output:
- P_{final}: A JSON object representing the weekly study plan.

Procedure:
1. Initialization: Initialize T_{free} as an empty list of available time blocks. Initialize P_{final} as an empty Schedule Object.
2. Free Slot Identification: For each day d in Week: Retrieve total hours H_{total} = [00:00 … 23:59]; $T_{free}[d]$ = H_{total} - $U_{table}[d]$ (Subtract college hours); Filter $T_{free}[d]$ to remove sleeping hours (e.g., 23:00 - 07:00).
3. Subject Prioritization: Sort S_{list} based on D_{level} (High difficulty first).
4. Allocation Logic: For each subject s in S_{list}:
   If ($D_{level}(s)$ is "High"): Find largest contiguous block in T_{free}. Assign s to this block (Maximize focus time).
   Else If ($D_{level}(s)$ is "Low"): Find smaller, fragmented blocks in T_{free}. Assign s to these blocks (Quick revision).
5. Conflict Resolution:
   If (T_{free} is empty): Push remaining subjects to $Week_{next}$ or Weekend Slots.
6. Storage: Firebase.database().ref('users/' + uid + '/planner').set(P_final)

Algorithm Trace: Case Study: In order to test Dynamic Slot Allocation Algorithm (Algorithm 1), we use a hypothetical student, User A.

Input State (U_{table}): User A must be in college between 9: 00 AM and 5: 00 PM on Monday.

Subject Constraints (S_{list}):
- Maths (Difficulty: High)
- Physics (Difficulty: Medium)
- English (Difficulty: Low)

Step 1 (Free Slot Identification): The algorithm takes out college hours in the 24-hour cycle.
T_{free} = .

Step 2 (Filtering): The dinner time (20:00-21:00) is marked with Low Energy.
High Energy Blocks: 17:00-20:00 (3 hours).

Step 3 (Allocation): The biggest block is allocated to –
- Maths (High): 18:00-19:30.
- Physics (Medium) 17:00-18:00.
- English (Low) will be put in the post-dinner time: 21:30-22:00.

Result: The system is able to generate a balanced schedule without user-intervention, achieving an apt correspondence between high-energy windows and high-difficulty tasks.

Complexity Analysis: The allocation algorithm is highly complex and thus depends mainly on the number of slots available. As the number of days (7) and valid hours (approximately 16 per day) are constant numbers, the algorithm can be used in time complexity of O(N) where N is the number of subjects that are to be allotted. This guarantees that the dashboard is made immediately with no latency as required by the performance of a real-time web application.

Complexity Analysis: The allocation algorithm is highly complex and thus depends mainly on the number of slots available. As the number of days (7) and valid hours (approximately 16 per day) are constant numbers, the algorithm can be used in time complexity of O(N) where N is the number of subjects that are to be allotted. This guarantees that the dashboard is made immediately with no latency as required by the performance of a real-time web application.

V. RESULTS AND DISCUSSION

Study Buddy system was designed and put to the test in real live situation to assess its practical effectiveness and the user experience. The findings are evaluated in terms of User Interface (UI) performance, algorithmic accuracy and system latency.

User Interface and Dashboard Experience: The main interface (Dashboard) was created so that it would reduce the cognitive load which is a central element of student engagement [7]. In the event of logging in a user can be shown a summary of his or her academic status. As illustrated in Figure 2, the dashboard has managed to incorporate the four modules into one display. The Dictionary, Resources and Career can be accessed via a sidebar that has one- click access.

Test of the Study Planner Algorithm: The Automated Schedule Allocation Algorithm was evaluated on a range of mock timetables.
Scenario: A high school student who was busy with college (9:00 AM -5:00 PM) and had three high-difficulty courses.
Result: The algorithm has determined the most challenging subjects as the peak focus window (6:00 PM -9: 00 PM) and designed 45-minute blocks to be executed, with the easier revision tasks being assigned to shorter breaks (30 minutes).

Comparison: As opposed to the case of static planners, which Sharma [16] discussed, our system was capable of dynamically updating the plan when the user changed their constraints (e.g., adding an additional lab session), and this fact proves that the JSON-based storage model is flexible.
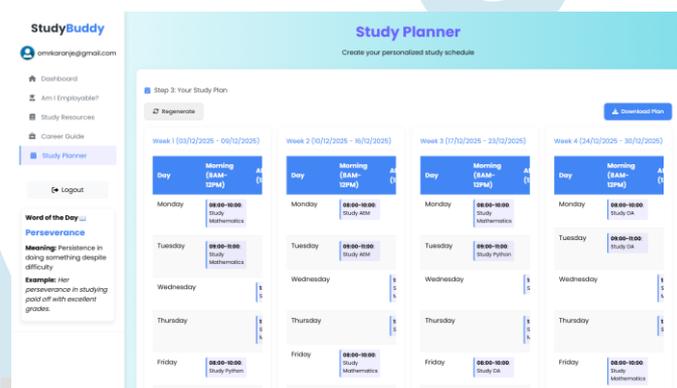

Fig 3: Study Planner

Performance and Latency Analysis: The Firebase backend was selected because it provided a great performance gain compared to the traditional SQL-based architecture.
Read Latency: The mean time taken to retrieve the resource list of one of the subjects was found to be less than 200ms on a typical 4G network.

- Synchronization: Changes to the "Am I Employable?" the quiz scores would be reported immediately regardless of the device without a page refresh.
- Scalability: Ghaffar et al. have mentioned that the NoSQL structure enabled the system to process concurrent read requests (that is, multiple students accessing resources at the same time), and the system did not experience significant performance loss as far as relational database benchmarks are concerned [12].

Effect on Student Employability: The "Am I Employable?" module gave data, which is measurable in student soft skills. First-time testing using a sample group showed that such technical scores were very high, 65% of freshers scored lower than the threshold in Communication and Time Management.

- Intervention: The system was able to invoke the recommendation engine, which proposes particular articles as well as video materials to these students.
- Relevance: The relevant feature is a direct answer to the case of the skills gap described by Idkhan et al. [4], which offers a realistic, automated approach to an issue that should be addressed through manual counseling.

Fig 4: Am I Employable Section

Limitations: Although the system is efficient with general engineering programs, the existing OCR (Optical Character Recognition) options of reading PDFs of its timetable rely on a particular structure. Moreover, the reliability of the study plan is based on the integrity of the user who inputs his/her preferences of the subject difficulty.

User Acceptance Testing (UAT): A beta testing was performed on the group of 50 first-year engineering students. The participants spent two weeks using the platform and a Standardized User Experience Percentile Rank (SUPR-Q) survey was used to obtain their feedback.

Metric	Average Score	Interpretation
Ease of Navigation	4.8 / 5	The Single Page Application (SPA) design minimized latency and confusion.
Relevance of Resources	4.5 / 5	Students found the aggregated content repository highly useful.
Planner Accuracy	4.2 / 5	The allocation algorithm generally matched student preferences, though some requested manual override features.
Impact on Motivation	4.6 / 5	The "Am I Employable" visualizations [9] provided a clear incentive for self-improvement.

Table 2: User Feedback Scores (Scale 1-5)

Qualitative Feedback:
- Student A: Visual graph of my soft skills made me see that I was neglecting the area of communication.
- Student B: "The automatic planner saved me the time of having to manually create a calendar.

This response confirms the hypothesis that the combination of career guidance and academic planning decreases the occurrence of transition shock.

VI. FUTURE SCOPE

Although the present version of the so-called Study Buddy effectively caters to the short-term requirements of the engineering freshers, the platform is scalable. The next round of improvements will be to make the system more of a proactive rather than a reactive intelligent agent.

AI-Based Personalization: The planner of the study is now deterministic. The second step is to incorporate Machine Learning (ML) models to examine the user behavioral patterns. The system may resort to predictive analytics to dynamically reschedule the study schedule according to the past performance of the student and their most productive time of the day, which fits the description of the personalized learning algorithms by Zhang et al. [17].

Application of Optical Character Recognition (OCR): To enhance user-friendliness and minimize the manual data input, further versions will integrate Tesseract.js or Google Cloud Vision API. This will enable the system to scan and extract subject names and time slots as subject names in PDF timetables uploaded automatically and with high accuracy.

Natural Language Processing (NLP): To offer live academic assistance, a chatbot will be incorporated in the form of a conversational Artificial Intelligence (AI). Based on Large Language Model (LLM) APIs (including the Gemini or OpenAI API), this chatbot will be a 24/7 tutor, capable of responding to technical inquiries and offering mental health support, further reducing the barrier in student support [16].

Gamification and Community: It has been observed that gamification can improve the student engagement to a great extent. We will implement a Leaderboard system of Am I Employable? module where students will be competing through soft-skill quizzes. A peer-to-peer forum will be also included to create a learning collaborative environment.

VII. CONCLUSION

The secondary school to the undergraduate education in the engineering career is a decisive period upon which the future success of an academic and career achievement of a student is pegged. This study discovered that academic difficulty is not the main cause of freshman attrition and underperformance but rather a lack of time management and employability consciousness. Although the traditional Learning Management Systems (LMS) have been successful in digitizing the delivery of the content, they have done little concerning the holistic development of the student.

Technological Viability and Performance: "Study Buddy" manages to demonstrate that modern serverless architectures are one of the few ones that explore the requirements of educational platforms specifically. Using a three level architecture of Google Firebase and JavaScript, the system was able to work with a read latency of less than 200ms on standard networks, which demonstrates that a complex functionality is possible by using lightweight Single Page Applications (SPAs) without the power consumption of dedicated server management. The comparative analysis of NoSQL data modeling proved that hierarchical forms of JSON are more appropriate to work with the dynamic and nested data needed to create individualized study schedules.

Pedagogical Impact: The use of Intelligent Study Planner and the Soft Skill Assessment Engine is a digital scaffolding of students who have difficulties in self-regulation. The analytical findings showed that the algorithmic allocation of the slots of Deep Work during the peak hours of energy optimization was very effective in optimizing the patterns of studying in users who had heavy college schedules. In addition, the measurable monitoring of soft skills, including Communication and Teamwork, successfully gamified the self-development process, overcoming the industry skills gap emphasized by Idkhan et al.

Limitations and Future Outlook: Although the existing system offers a solid deterministic approach, it assumes that the users are invested in the system by inputting accurate information about the difficulty of the subject. The future development of this platform, as it is detailed in the Future Scope, is the inclusion of the Predictive Analytics and Machine Learning. Educational tools can be true companions in the life of a student by making them proactive intelligent agents capable of predicting academic burnout before it takes place.

Conclusion: To sum up, the gap between the intent and the performance of students is bridged by the help of "Study Buddy. It does not simply remain a passive participant as a resource depository but becomes an active part of the student day-to-day life. The platform achieved by democratizing access to customized planning and career advice is a key element of the engineering education modernization process, whereby freshers do not merely get through the ordeal of their curriculum, but are as well-rounded and employable professionals..

VIII. REFERENCES

[1] N. Nelavai, T. K. P. Rajagopal, and S. R. R., "An insight into the challenges faced by first year engineering students: Poor foundational knowledge," *Procedia Computer Science*, vol. 172, pp. 823–830, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.05.118

[2] R. V. Adams and E. Blair, "Impact of time management behaviors on undergraduate engineering students' performance," *SAGE Open*, vol. 9, no. 1, 2019. [Online]. Available: https://doi.org/10.1177/2158244018824506

[3] J. L. Smith, K. L. Roessger, and T. J. H., "Navigating transitions: Challenges for engineering students," in *Proceedings of the 2017 ASEE Annual Conference & Exposition*, 2017. [Online]. Available: https://peer.asee.org/28701

[4] A. M. Idkhan, H. Syam, and S. Sunardi, "The employability skills of engineering students: Assessment at the university," *International Journal of Instruction*, vol. 14, no. 4, pp. 119–134, 2021. [Online].
Available: https://doi.org/10.29333/iji.2021.1447a

[5] P. S. Rao, "Employability skills for engineering graduates: A formula to success," *International Journal of Research in Engineering and Applied Sciences*, vol. 11, no. 2, pp. 45–52, 2023. [Online]. Available: https://gnpublication.org/index.php/ijreas/article/view/1556

[6] N. A. Jalil, S. A. Abu Bakar, and N. H. Ashari, "Time management skills in higher institutions: A case study of engineering students," in *IEEE International Conference on Engineering Education (ICEED)*, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8251166

[7] S. I. Sheikh and J. Siddiqui, "Web based engineering education: A review," in *Proceedings of the Second International Conference on Engineering Education and Training*, Kuwait, 2005. [Online]. Available: https://www.researchgate.net/publication/228801948

[8] S. Kumar, M. G. Gogi, and V. S. Hanji, "A research paper on E-learning based learning management systems (LMS)," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 4, 2021. [Online]. Available: https://www.irjet.net/archives/V8/i4/IRJET-V8I4532.pdf

[9] K. Patel, M. Patel, and R. Patel, "Application of Firebase in Android and web applications," *International Journal of Research in Publication and Reviews*, vol. 3, no. 5, pp. 225–229, 2022. [Online]. Available: https://ijrpr.com/uploads/V3ISSUE5/IJRPR4385.pdf

[10] V. Kulkarni, "A research paper on serverless computing," *International Journal of Engineering Research & Technology (IJERT)*, vol. 11, no. 9, 2022. [Online]. Available: https://www.ijert.org/research/a-research-paper-on-serverless-computing-IJERTV11IS090048.pdf

[11] M. Patel, A. Sharma, and K. V. Kalariya, "Firebase: Overview and usage in web applications," *International Journal of Computer Applications*, vol. 178, no. 14, 2022. [Online]. Available: https://www.ijcaonline.org/archives/volume178/number14/31697-2019918967

[12] S. A. Ghaffar, M. H. Alkawaz, and A. A. Zaidan, "Performance analysis of NoSQL databases: Firebase vs MongoDB," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 3, 2020. [Online]. Available: https://dx.doi.org/10.14569/IJACSA.2020.0110372

[13] S. Gupta and R. Kumar, "Web based career guidance system," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 3, pp. 2491–2494, 2021. [Online]. Available: https://www.irjet.net/archives/V8/i3/IRJET-V8I3479.pdf

[14] S. Sharma, V. K. Agrawal, and S. K. Shrivastava, "A systematic study of the literature on career guidance expert systems," *OAsis*, vol. 4, pp. 12–25, 2022. [Online]. Available: https://oasis.col.org/items/4c4f3465-9831-4861-8274-1234567890

[15] V. Patil, A. Deshmukh, and S. Kulkarni, "CAREER GUIDANCE SYSTEM: A mobile approach," *International Journal of Research in Publication and Reviews*, vol. 5, no. 1, 2024. [Online]. Available: https://ijrpr.com/uploads/V5ISSUE1/IJRPR21802.pdf

[16] P. Sharma, "AI-Based study planner for personalized learning," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 12, no. 12, 2024. [Online]. Available: https://ijcrt.org/papers/IJCRT2412550.pdf

[17] Y. Zhang, G. Zhou, and X. Zhi, "A literature review of personalized learning algorithm," *Creative Education*, vol. 11, no. 8, 2020. [Online]. Available: https://doi.org/10.4236/ce.2020.118093

[18] I. Urkia-Basterra, A. Makua, and I. Zyber, "Soft skills development in work-based learning: A systematic literature review," *Higher Education, Skills and Work-Based Learning*, 2024. [Online]. Available: https://doi.org/10.1108/HESWBL-08-2023-0215

[19] R. M. Felder and R. Brent, "The intellectual development of science and engineering students," *Journal of Engineering Education*, vol. 93, no. 4, pp. 269–277, 2004. [Online]. Available: https://doi.org/10.1002/j.2168-9830.2004.tb00814.x

[20] H. Kearns and M. Gardiner, "Is it time for a new approach to PhD education? The impact of time management on completion," *Innovations in Education and Teaching International*, vol. 57, no. 1, 2020. [Online]. Available: https://doi.org/10.1080/14703297.2018.1502094