IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

Skill-Setu: An AI-Powered Student Skill Assessment and Career Recommendation System

Ms. Namrata Indrasing Girase[1], Prof. Manoj V. Nikum[2]

Research Scholar MCA, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[1]

HOD MCA & Assistant Professor, Shri Jaykumar Rawal Institute of Technology, Dondaicha, India[2]

Abstract: The transition from academic education to employment in the technology sector is challenging because students may not know which practical skills are expected for a particular entry-level role. Existing job portals, course platforms and general-purpose conversational assistants often address only part of this problem. This paper presents Skill-Setu, a web-based career guidance system designed to combine student skill assessment, internship matching, skill-gap identification, personalized learning roadmaps and AI-assisted career guidance in one platform. The application uses Python Flask for backend request handling, Supabase PostgreSQL for persistent data storage and access control, and the Groq API for generating structured AI responses. Internship matching is calculated using deterministic set-intersection logic so that students can see both matched and missing skills, while AI is used to provide qualitative explanations and learning suggestions. A five-phase roadmap supports structured preparation and task tracking. The project report describes unit, integration, security and workflow testing, including database-level Row Level Security checks. The reported implementation demonstrates an integrated approach to explainable skill matching and personalized career preparation. The system is intended as a guidance and progress-tracking aid; it does not guarantee employment.

Keywords: Career guidance, skill assessment, internship recommendation, skill-gap analysis, learning roadmap, Flask, Supabase PostgreSQL, Groq API, Row Level Security.

I. INTRODUCTION

The technology industry changes rapidly as software development practices, cloud services, automation and artificial intelligence evolve. Employers commonly expect graduates to demonstrate practical competence with programming languages, databases, frameworks and development tools. University courses provide important theoretical foundations, but students may still need guidance in identifying the additional skills required for a particular job role.

Students often search across job portals, online courses, discussion forums and separate assessment tools. These sources can provide useful information, but they may not compare a student's existing skills with a target role in a consistent and actionable way. A list of vacancies may state required skills without explaining which requirements a student already satisfies or what should be learned next. Similarly, generic course lists may not account for a learner's prior knowledge.

Skill-Setu is proposed as a unified web application that connects a student's education, technical skills, projects and career preferences with internship opportunities and target-role requirements. It combines deterministic computation for match percentages with AI-generated summaries, explanations and roadmaps. This separation is important: the numerical match is calculated by a reproducible algorithm, while the language model provides supporting qualitative guidance.

II. PROBLEM STATEMENT

Students preparing for entry-level technology roles face fragmented career-preparation workflows, limited personalized skill-gap information, unclear recommendation criteria and learning paths that may not reflect their current abilities. In addition, applications that store student profiles and use third-party AI services must protect personal records and API credentials. The problem addressed by this work is to design a lightweight, secure and user-friendly platform that assesses student profiles, compares skills with internship and role requirements, recommends structured learning activities and provides AI-assisted career guidance in a single system.

III. OBJECTIVES

Provide secure student registration, login and profile management.

Analyze a student's education, skills, projects and career interests to produce a structured profile assessment.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

Calculate transparent internship match percentages and identify matched and missing skills.

Compare student skills with benchmark requirements for selected career roles.

Generate multi-phase learning roadmaps with priorities, estimated durations and task-completion tracking.

Provide a dashboard and AI Career Assistant for ongoing career-preparation support.

Protect user-specific data through PostgreSQL Row Level Security and keep sensitive API credentials on the server.

IV. BACKGROUND AND RELATED WORK

Career preparation is commonly supported by three broad categories of digital platforms. Commercial job portals provide access to vacancies and employer networks, but their recommendation criteria may not expose a detailed skill-by-skill explanation. Online course marketplaces provide instructional content, yet their course sequences are often predefined and may not adapt to a student's existing knowledge. General-purpose conversational AI systems can answer open-ended career questions, but they do not necessarily provide persistent profile storage, a curated internship dataset or built-in progress tracking.

Skill-Setu combines selected capabilities from these categories. It uses a relational database to retain student information and learning progress, a deterministic algorithm for explainable matching, and an AI service for natural-language assessment and advice. Its design also applies established software engineering practices, including modular backend routes, environment-variable configuration and database-level access policies.

The project report identifies Flask, PostgreSQL Row Level Security, Supabase and Groq documentation, along with standard software engineering and database textbooks, as the technical foundations for the implementation. The comparative discussion in this paper is based on the project report's analysis; it should not be interpreted as a new independent benchmark of commercial platforms.

V. PROPOSED SYSTEM AND METHODOLOGY

A. System Architecture

Skill-Setu follows a three-tier Model-View-Controller-oriented architecture. The presentation layer uses server-rendered HTML templates, Bootstrap 5, JavaScript and Chart.js. The application layer is implemented using Python Flask and handles authentication, form processing, business logic, database operations and AI requests. The data layer uses Supabase PostgreSQL to store profiles, internships, saved internships, skill assessments, learning roadmaps and career-chat messages.

The Groq API is called from the backend. Student profile information is assembled into prompts and sent to the configured model to obtain structured outputs such as profile summaries, strengths, improvement areas, suitable roles and next steps. API credentials are read from environment variables rather than embedded in client-side code.

B. Student Profile and AI Assessment

The profile module captures information such as the student's name, education, college, graduation year, technical skills, interests, preferred role and experience level. The backend stores the profile in the profiles table and submits relevant profile information to the AI service. The returned assessment is expected to contain a summary, strengths, improvement areas, suitable roles and recommended next steps. The assessment is stored for later display and review.

C. Deterministic Internship Matching

Internship matching uses normalized skill sets. Let S_student represent the student's skills and S_required represent the skills required by an internship. The overlap and missing-skill sets are:

$$S_{overlap} = S_{student} \cap S_{required}$$

$$S_{missing} = S_{required} - S_{student}$$

The match percentage is calculated as:

$$\text{Match Percentage} = (|S_{overlap}| / |S_{required}|) \times 100$$

 This work is licensed under a Creative Commons Attribution 4.0 International License

The implementation described in the project report converts comma-separated skill strings to lowercase sets, removes surrounding whitespace and calculates the intersection and difference. If the required skill set is empty, the function returns a zero match. The interface can therefore show a numerical match together with the exact overlapping and missing skills. The AI service may explain the result in natural language, but it does not determine the numerical score.

D. Skill-Gap Analysis and Learning Roadmap

The skill-gap module compares a student's profile with predefined skill requirements for target roles such as Data Analyst, Python Developer, Frontend Developer, Backend Developer, Data Scientist, Machine Learning Engineer and Business Analyst. The comparison identifies skills that are already present and skills that require further learning.

For a selected target role, the platform generates a five-phase roadmap: (1) Fundamentals, (2) Core Concepts, (3) Tools and Visualization, (4) Portfolio Projects, and (5) Interview Preparation. Roadmap tasks include priority and estimated duration information. Students can mark tasks as completed, and the completion state is persisted in the learning_roadmaps table.

E. Career Readiness Indicator

The dashboard calculates a weighted readiness indicator using three components:

$$\textbf{Readiness Index} = (0.40 \times M_{skill}) + (0.40 \times C_{roadmap}) + (0.20 \times P_{profile})$$

Here, M_skill is the average match percentage across selected target roles, C_roadmap is the percentage of assigned roadmap tasks completed, and P_profile is the profile-completeness score. Each component is expressed on a 0–100 scale. This index is a progress-tracking measure and must not be treated as a prediction of employment or a guarantee of job placement.

VI. DATABASE DESIGN AND SECURITY

The database schema described in the project report contains six principal tables: profiles, internships, saved_internships, skill_assessments, learning_roadmaps and career_chat_messages. UUID identifiers are used for records. User-specific tables include a user_id field, while assessment and roadmap outputs are stored as JSONB data. Indexes are defined on user_id columns in frequently accessed user-specific tables to support query performance.

Row Level Security (RLS) is enabled on the relevant tables. Policies use the authenticated user's identity to restrict access to that user's records. For example, the profile policy applies the condition auth.uid() = user_id. The internships table has a policy that permits reading active internship records. This database-level enforcement provides an additional protection layer beyond checks performed in Flask route handlers.

The project also describes keeping Groq and database credentials in a server-side .env file, using environment variables, and excluding the .env file from version control. Input validation and template auto-escaping are used to reduce risks associated with invalid input and cross-site scripting. Production deployment should additionally use a strong randomly generated Flask secret, HTTPS, appropriately restricted database keys, secure cookie settings and reviewed RLS policies.

VII. IMPLEMENTATION TECHNOLOGIES

Technology	Purpose
Python and Flask	Backend routing, session handling, business logic and server-rendered pages.
Supabase PostgreSQL	Relational data storage, JSONB records, indexes and Row Level Security.
Groq API	AI-generated profile assessments, skill-gap explanations, roadmaps and chat responses.
HTML, CSS and Bootstrap 5	Responsive user interface and page layout.
JavaScript and Chart.js	Interactive behavior, dashboard charts and progress visualization.
python-dotenv	Loading configuration and credentials from environment variables.

VIII. TESTING AND RESULTS

The project report describes a multi-level validation process covering unit testing, integration testing, security checks, system workflows and usability. Unit tests focus on input validation and parsing of structured AI responses. Integration tests examine environment-variable loading, Supabase operations, Groq request-response handling and session continuity. Security testing includes attempts to access another user's records, checks for exposed credentials, SQL-injection-style input and malicious HTML or JavaScript content.

The report's test matrix records successful outcomes for the documented scenarios, including account registration and login, protected-route redirection, profile submission, internship matching, bookmarking, skill-gap generation, roadmap creation and task completion, career-assistant responses, readiness calculation, RLS isolation and safe rendering of untrusted text. One example uses student skills Python and SQL against required skills Python, SQL and Excel. Two of the three required skills overlap, giving approximately 66.7%, displayed as 67% in the reported test result.

The report also presents the following performance figures from its described test environment: high-speed broadband page load of 190 ms, simulated 4G page load of 340 ms and throttled 3G page load of 780 ms. It lists an average initial page-load/routing duration of 180 ms, a Supabase query duration of 90 ms and Groq inference duration of 380 ms. These values are reproduced as reported project measurements; the paper does not independently reproduce the experiments or establish that the same timings will apply to other hardware, networks, regions or API conditions.

The project report states that its test cases passed and describes zero authorization leaks in the tested scenarios. Such results indicate the outcomes recorded by the project team, but they are not a substitute for an independent security audit, broader penetration testing or evaluation with a larger group of student users.

IX. DISCUSSION

The main design contribution of Skill-Setu is the combination of transparent rule-based matching and AI-supported explanation within a single student workflow. Because match scores are computed from explicit sets, students can inspect why a recommendation received a particular score. The missing-skill list connects the recommendation to a practical learning action. The roadmap and dashboard then provide a mechanism for tracking progress over time.

The architecture also separates responsibilities: Flask coordinates application workflows, PostgreSQL stores persistent records and enforces access policies, and the AI API generates language-based assistance. This separation can make the system easier to maintain and test. However, the quality of recommendations depends on the accuracy and freshness of internship records, the completeness of student profiles and the quality of role-specific skill benchmarks. AI-generated suggestions should also be reviewed because language-model outputs may be incomplete or inaccurate.

X. LIMITATIONS AND FUTURE SCOPE

The project report identifies demo internship records as a current limitation and proposes connecting live job-board APIs, such as Adzuna, to provide current opportunities. It also proposes automated resume parsing using PDF extraction libraries, video tutorial links for roadmap milestones, learning activity heatmaps and a mobile-oriented release through a Progressive Web App or Flutter client.

Further development should evaluate recommendation quality using a larger and more diverse set of student profiles and real internship descriptions. Future experiments could measure precision and recall of skill extraction, student satisfaction, roadmap completion, response latency under load and the consistency of AI-generated advice. Role benchmarks should be periodically reviewed with current job descriptions and industry input. Accessibility, multilingual support and privacy-conscious data retention are also important areas for improvement.

XI. CONCLUSION

This paper presented Skill-Setu, an AI-powered student skill assessment and career recommendation system developed using Python Flask, Supabase PostgreSQL and the Groq API. The platform integrates student profile assessment, deterministic internship matching, target-role skill-gap analysis, phased learning roadmaps, dashboard-based progress tracking and an AI Career Assistant. Its use of set-based matching makes internship scores inspectable, while PostgreSQL Row Level Security supports isolation of user-specific records. The project report documents functional, integration and security testing, along with performance measurements from its test environment. Skill-Setu offers a unified approach to

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

career preparation, while further work is needed to connect live opportunities and evaluate the system with broader real-world student populations.

REFERENCES

[1]. R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill Education, 2020.
[2]. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
[3]. PostgreSQL Global Development Group, "Row Security Policies," PostgreSQL Documentation. [Online]. Available: https://www.postgresql.org/docs/current/ddl-rowsecurity.html
[4]. Supabase, "Supabase Documentation." [Online]. Available: https://supabase.com/docs
[5]. Groq, "Groq API Documentation." [Online]. Available: https://console.groq.com/docs
[6]. Bootstrap Team, "Bootstrap Documentation." [Online]. Available: https://getbootstrap.com/
[7]. A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY, USA: McGraw-Hill Education, 2020.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License