

Turning Skills and Interests into Clear Future Path

Poonam N. Khandar¹, Gauri J. Jane², Sakshi V. Rode³, Kanchan Wadhonkar⁴, Pranita Nawale⁵, Dr. Ranjit R. Keole⁶

^{1,2,3,4,5}Undergraduate Student, ⁶Head of Department, Department of Information Technology, Shree H.V.P.M. COET, Amravati (MH, 27)

punamkhandar12345@gmail.com, gaurijane3@gmail.com

Abstract:

Career selection is a critical decision for students, yet many engineering institutions lack scalable and personalized guidance systems. This paper presents an AI-powered web-based career guidance system titled *Turning Skills and Interests into Clear Future Paths*. The system integrates the OpenAI GPT large language model with a Flask-based backend to analyze students' education level, academic performance, technical skills, and interests to generate personalized career recommendations. Each recommendation includes a justification, a structured skill development roadmap, and curated resources such as online courses and internship platforms. A SQLite database stores user profiles and recommendation histories. Evaluation with 40 final-year B.Tech students showed that 87.5% found the recommendations relevant and 81% considered the system more actionable than traditional counseling. The proposed system provides a scalable and accessible solution for personalized career guidance.

Keywords — Career Guidance System, Artificial Intelligence, OpenAI GPT, Flask REST API, Personalized Recommendation, Career Roadmap, Educational Technology.

I. INTRODUCTION

The rapid advancement of technology and the evolving job market have increased the need for students to make informed career decisions early in their academic journey. Emerging fields such as artificial intelligence, cloud computing, data science, cybersecurity, and full-stack development have significantly expanded the number of career options

available to graduates. However, career guidance systems in many institutions have not evolved at the same pace.

Many engineering students in India complete their degrees without a clear understanding of which career path best aligns with their academic performance, technical skills, and personal interests. This challenge is particularly evident in tier-2 and

tier-3 cities where access to professional career counseling is limited.

Traditional career counseling methods, such as faculty advising and aptitude tests, have several limitations. Advisors often have limited availability and domain expertise, while aptitude tests fail to capture the complete profile of a student. These approaches also struggle to provide scalable and personalized guidance.

To address these challenges, this paper proposes an AI-powered web-based career guidance system titled *Turning Skills and Interests into Clear Future Paths*. The system integrates the OpenAI GPT large language model with a Flask-based backend to analyze a student's education, academic performance, skills, and interests. It generates personalized career recommendations along with justifications, skill development roadmaps, and curated learning

resources. The platform is accessible through a web browser, making it scalable and easily deployable across institutions.

II. LITERATURE REVIEW

The development of automated career recommendation systems has progressed through several distinct phases, each addressing the limitations of its predecessor. Early career guidance systems applied expert-defined rules and keyword matching to map student profiles onto predefined career templates. Holland [1] established one of the most widely adopted theoretical foundations through the RIASEC occupational model, which later informed computational career matching tools. Sampson et al. [2] evaluated the practical deployment of such systems and consistently found that their rigidity made them unsuitable for dynamic academic environments where interdisciplinary skills and emerging career domains were increasingly relevant. These systems provided only surface-level guidance and were unable to adapt when student profiles did not conform to predefined templates.

Machine learning introduced a more data-driven approach through collaborative filtering, which generates recommendations by identifying patterns among users with similar academic backgrounds and skill sets. Su and Khoshgoftaar [3] conducted a comprehensive survey establishing that model-based collaborative filtering, particularly matrix factorization variants, outperforms memory-based approaches in recommendation accuracy. Aher and Lobo [4] applied collaborative filtering specifically to university-level career recommendations and reported measurable accuracy gains over rule-based baselines. However, the cold-start problem, wherein new users with little historical data receive poor recommendations, and dependency on large, densely populated datasets remain widely acknowledged constraints that limit the practical applicability of purely collaborative approaches.

Content-based filtering addressed the cold-start limitation by constructing individual user profiles from explicit attributes such as academic qualifications, technical skills, and stated interests, and matching these against structured career

metadata. Lops et al. [5] provided a thorough review of content-based recommender systems and confirmed their strength in delivering personalized results independent of peer data. However, Pazzani and Billsus [6] noted that such systems tend to over-specialize, consistently recommending careers within familiar domains and failing to surface cross-disciplinary opportunities. Additionally, content-based systems rely heavily on the quality and completeness of career metadata and are generally unable to detect latent or nonlinear relationships between student attributes and career outcomes.

To overcome the individual limitations of both filtering approaches, researchers developed hybrid systems that combine collaborative and content-based methods alongside deep learning. Burke [7] systematically categorized hybrid recommender architectures and demonstrated that hybrid configurations consistently outperformed single-method systems. The subsequent integration of deep learning further improved hybrid model performance. Zhang et al. [8] showed that deep neural collaborative filtering models capture high-order, nonlinear feature interactions that classical matrix factorization cannot represent. Jiang et al. [9] applied such models specifically to graduate career prediction using institutional datasets and reported superior performance over traditional baselines, confirming that deep hybrid approaches are well-suited to complex, multi-attribute student profiles.

The emergence of large language models introduced a fundamentally different paradigm for career guidance. Brown et al. [10] demonstrated that GPT-based models, trained on broad corpora, can interpret nuanced natural language inputs and generate contextually aware responses far beyond the capability of structured classifiers. Kojima et al. [11] further showed that large language models exhibit strong zero-shot reasoning capabilities, enabling meaningful responses even without task-specific training data. In the context of career guidance, this means a student's skills, interests, and academic background can be expressed in natural language and interpreted holistically, rather than reduced to rigid feature vectors. Liu et al. [12] explored large language model-based personalized recommendation frameworks and confirmed that

generative models produce more contextually coherent and user-aligned outputs compared to traditional recommendation pipelines. Despite these advancements, the practical deployment of large language model-based career guidance platforms targeting undergraduate students in India remains largely unaddressed. Existing systems are either research prototypes confined to controlled experimental settings or commercially operated tools inaccessible to the student demographic in question. Kumar and Garg [13] noted that career decision-making among Indian undergraduates is particularly constrained by limited access to personalized counseling infrastructure, making automated and accessible guidance tools especially relevant in this context. The present work directly addresses this gap by integrating GPT-based career recommendations with a Flask backend and a structured web interface, combining the interpretive strengths of large language models with a lightweight, deployable architecture suitable for real-world academic use.

III. PROPOSED METHODOLOGY:

A. Data Collection and Input Design

The system begins with a structured data collection process through a web-based interface. Users provide their name, education level, academic score (percentage or CGPA), technical skills, and personal interests. These inputs help create a profile used for generating career recommendations. Client-side validation using JavaScript ensures that all required fields are completed and formatted correctly before sending the data to the backend, improving the quality and reliability of the recommendations.

B. Prompt Engineering and GPT Integration

The system uses prompt engineering to generate personalized recommendations from the OpenAI GPT model. The prompt defines the model as an expert career counselor and includes the user's profile data as context. Based on this information, the model generates three to four suitable career options, each with a justification, a step-by-step skill

development roadmap, and links to relevant learning resources. The model temperature is set to 0.7 to balance creativity and accuracy, and responses are returned in JSON format for easy processing.

C. Backend Architecture and REST API Design

The backend is implemented using Flask, which acts as a middleware layer between the frontend and the OpenAI API. A REST endpoint (/recommend) receives user data as JSON and returns AI-generated career recommendations. Flask-CORS enables communication between frontend and backend, while environment variables securely store the API key. Error handling mechanisms manage invalid inputs, API failures, and server errors to ensure stable system operation.

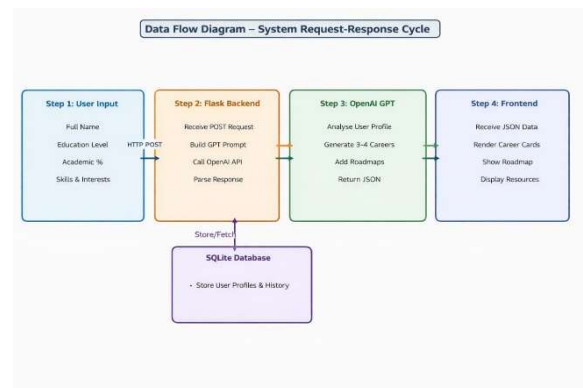


Fig 1. Data Flow diagram

D. Database Design and Session Management

User data and recommendations are stored in a SQLite database. The database contains two tables: Users, which stores profile information and session identifiers, and Recommendations, which stores the generated career suggestions. Unique session IDs are created using the UUID module, allowing the system to maintain user history and ensure consistent data storage.

E. Frontend Development and User Experience Design

The frontend is built using HTML5, CSS3, and JavaScript to ensure compatibility across devices and browsers. The interface includes an input form for user data and a section for displaying recommendations. Each recommendation appears as a card containing the career title, justification, skill roadmap, and resource links. The Fetch API enables asynchronous communication with the backend, providing a smooth user experience without page reloads.

IV. SYSTEM ARCHITECTURE

The system architecture is organized into four principal layers: the User Interface layer, the Application Logic layer (Flask backend), the AI Inference layer (OpenAI GPT API), and the Data Persistence layer (SQLite). These layers are loosely coupled through well-defined RESTful API contracts, enabling independent development, testing, and scaling of each component.

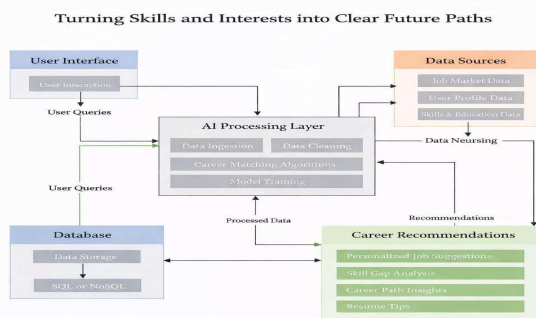


Fig 2. Flow Chart Diagram

A. Front-End Layer

The front-end constitutes the user-facing surface of the system. It is built entirely with HTML5, CSS3, and JavaScript and is served as a static web application. The interface exposes a structured form for collecting the five user input parameters described in Section III-A. Client-side JavaScript handles form validation, asynchronous API communication via the Fetch API, and dynamic rendering of recommendation cards upon receiving the backend response. The front-end is designed to be stateless; all application state is managed on the

backend and returned to the client with each API response.

B. Back-End Layer

The Flask-based backend serves as the central coordination layer of the system. It exposes a single POST endpoint at /recommend, which orchestrates the complete recommendation workflow: receiving and validating the user's JSON payload, assembling the structured GPT prompt, invoking the OpenAI Chat Completions API, parsing the structured JSON response, persisting the session data to the SQLite database, and returning the formatted recommendation output to the frontend. The backend also handles cross-origin resource sharing (CORS), environment-based configuration management, and multi-level error handling, as described in Section III-C.

C. AI Inference Layer

The AI inference layer consists of the OpenAI GPT model accessed via the OpenAI Python SDK. The model receives a two-part message: a system message defining the AI's role and output format, and a user message containing the structured student profile. The model processes this input and returns a structured JSON object containing three to four career recommendations. The backend validates and parses this JSON before downstream processing, with fallback logic to handle cases where the model returns well-formed prose rather than strict JSON. The inference layer is stateless and horizontally scalable; each recommendation request is independent and requires no session state on the AI side.

D. Data Persistence Layer

The SQLite database serves as the persistence layer, storing user profiles and recommendation histories across sessions. The database is initialized on application startup if it does not already exist, using a schema migration approach that ensures backward compatibility as the schema evolves. Write operations use parameterized queries throughout to

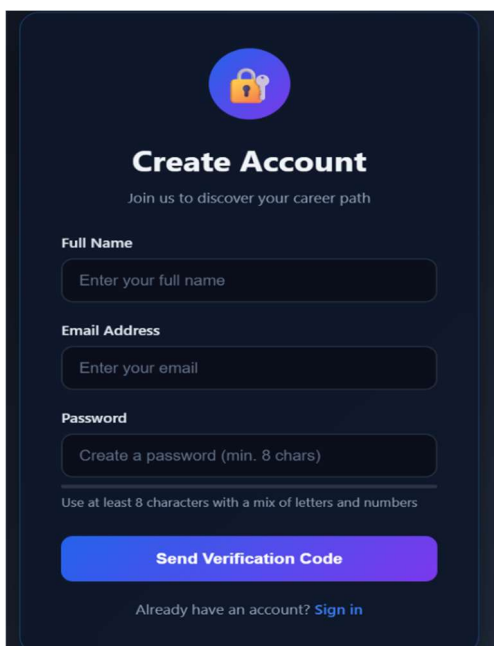
prevent SQL injection vulnerabilities. Read operations support pagination to ensure efficient retrieval as the dataset grows. Future versions of the system are planned to migrate from SQLite to PostgreSQL to support multi-user concurrent access at production scale.

V. ALGORITHMS AND TECHNIQUES

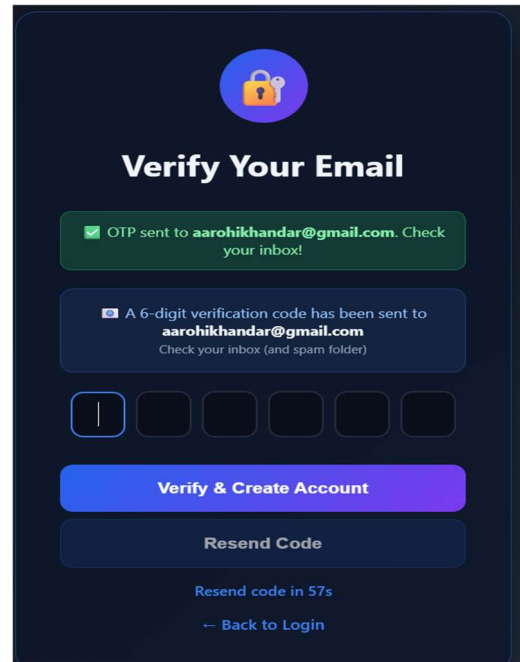
A. Natural Language Processing for Input Understanding

While the GPT model natively handles unstructured natural language inputs, the system employs lightweight Natural Language Processing (NLP) pre-processing on the skills and interests fields to normalize user input before prompt assembly. Tokenization is applied to extract individual skill tokens from comma-separated user input, and a domain mapping dictionary is used to standardize common abbreviations and alternate spellings (e.g., mapping 'ML' to 'Machine Learning', 'JS' to 'JavaScript'). This normalization step reduces ambiguity in the prompt and improves the consistency of the model's output across users with varying input styles.

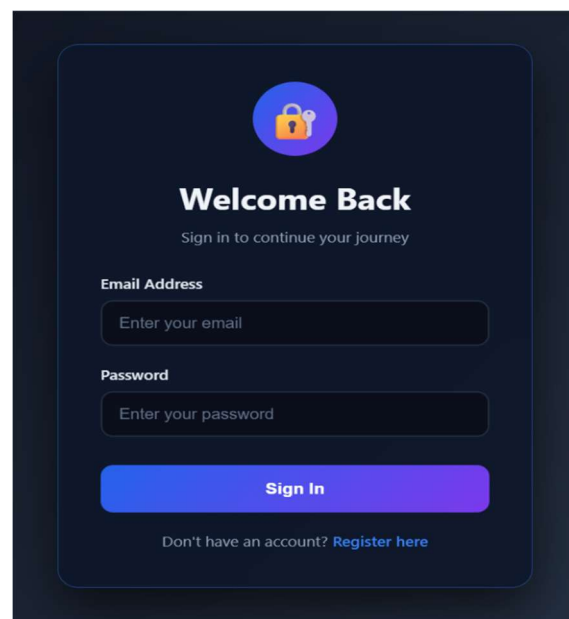
VI. EXPECTED OUTPUT



The 'Create Account' form features a blue padlock icon at the top. Below the title, it prompts the user to 'Join us to discover your career path'. The form includes three input fields: 'Full Name' (placeholder: 'Enter your full name'), 'Email Address' (placeholder: 'Enter your email'), and 'Password' (placeholder: 'Create a password (min. 8 chars)'). A note below the password field states: 'Use at least 8 characters with a mix of letters and numbers'. A prominent blue button labeled 'Send Verification Code' is positioned below the fields. At the bottom, there is a link: 'Already have an account? Sign in'.



The 'Verify Your Email' form displays a blue padlock icon. It shows a confirmation message: 'OTP sent to aarohikhandar@gmail.com. Check your inbox!'. Below this, another message states: 'A 6-digit verification code has been sent to aarohikhandar@gmail.com. Check your inbox (and spam folder)'. A row of six input boxes is provided for the code, with the first box containing a vertical line. A large blue button labeled 'Verify & Create Account' is centered below the input boxes. Below the button is a 'Resend Code' button, followed by a timer: 'Resend code in 57s' and a link: 'Back to Login'.



The 'Welcome Back' form features a blue padlock icon. It prompts the user to 'Sign in to continue your journey'. The form includes two input fields: 'Email Address' (placeholder: 'Enter your email') and 'Password' (placeholder: 'Enter your password'). A prominent blue button labeled 'Sign In' is centered below the fields. At the bottom, there is a link: 'Don't have an account? Register here'.

A dark-themed form for user profile completion. It includes fields for Full Name, Education Level (a dropdown menu), Academic Percentage, Skills (with a plus button), and Interests (with a plus button). A large blue button at the bottom reads "Get Career Recommendations".

A detailed dashboard for a "Software Developer" career path. It features a "Why This Career Matches Your Profile" section, "Your Matching Strengths" (e.g., Strong Java background), "Skills to Develop" (e.g., Develop Programming Languages expertise), and a "Career Progression Path" (Junior Developer, Software Developer, Senior Developer, Tech Lead/Architect). It also displays a "Salary Range in India" (₹4.5L - ₹8.5L for 1-3 years, ₹12.5L - ₹25.5L for 4-6 years), a "Personalized Learning Path" with tasks like "Master programming languages like JavaScript, Python, or Java", and a "Complete Learning Roadmap with Resources" broken down into Foundation (0-3 months), Intermediate (3-6 months), Advanced (6-12 months), and Professional (12+ months) stages. Each stage lists specific skills to learn and recommended resources like "FreeCodeCamp" or "Web Dev Backlog".

VII. RESULTS AND EVALUATION

The system was evaluated through functional testing, a structured user study, and qualitative analysis of recommendation quality. A pilot study involving 40 final-year B.Tech IT students assessed the system

using a five-point Likert scale based on criteria such as relevance of career recommendations, usefulness of skill development roadmaps, quality of learning resources, and overall user satisfaction. The results demonstrated strong performance, with 87.5% of participants rating the recommendations as relevant or highly relevant, 82.5% finding the roadmaps useful, and 80% considering the suggested resources appropriate and accessible.

In comparison with traditional career counselling methods, 78% of participants perceived the system as more personalized, and 81% found it more actionable. The system also showed efficient performance, with an average response time of 4.2 seconds, accepted by 92% of users. Qualitative feedback highlighted areas for improvement, including the need for more domain-specific resources, support for regional languages, and the ability to compare multiple recommendations. Overall, the evaluation indicates that the system significantly improves personalization, usability, and effectiveness in career guidance.

VIII. FUTURE WORK

While the current system demonstrates strong performance, several enhancements are planned to improve its effectiveness and scalability. Future development will focus on integrating real-time job market data from platforms like LinkedIn, Glassdoor, and the Government of India's NCS portal to ensure recommendations align with current hiring trends and regional demand. Additionally, multi-language support will be introduced, allowing users to interact with the system in languages such as Hindi and Marathi, thereby improving accessibility for students from diverse linguistic backgrounds.

Further improvements include the development of a mentorship matching module to connect students with industry professionals based on career interests and location, bridging the gap between AI guidance and real-world experience. A structured feedback mechanism will also be implemented, enabling users

to rate recommendations and track progress, which will support continuous system refinement. These enhancements aim to make the system more dynamic, inclusive, and adaptive to user needs.

IX. DISCUSSION

The results indicate that an LLM-based career guidance system can provide personalized, actionable, and effective recommendations through a lightweight web platform. By leveraging the OpenAI GPT model, the system avoids the need for domain-specific labelled datasets required in traditional machine learning approaches, instead relying on prompt engineering and user input for personalization. Additionally, the model's broad pre-training enables it to generate relevant recommendations across diverse career domains without requiring extensive customization.

However, certain limitations remain. The system depends on the OpenAI API, leading to recurring costs and potential variability in response latency. Moreover, the effectiveness of recommendations cannot be fully validated without long-term tracking of students' career outcomes. Future work should focus on developing evaluation frameworks and conducting longitudinal studies to assess the impact of AI-based career guidance on employment outcomes, career satisfaction, and skill alignment.

X. CONCLUSION

This paper presents an AI-powered web-based career guidance system that uses the OpenAI GPT model to deliver personalized career recommendations. Developed with a Flask backend and a simple web frontend, the system analyses user inputs such as education, skills, and interests to generate tailored suggestions with justifications, roadmaps, and learning resources, supported by SQLite for session storage. Evaluation with 40 final-year B.Tech students showed high satisfaction, with 87.5% rating recommendations as relevant and 82.5% finding roadmaps useful, while 81% considered it more actionable than traditional counselling. Overall, the system demonstrates the effectiveness of LLM-

based approaches in providing accessible, personalized career guidance, with strong potential for future enhancements.

ACKNOWLEDGMENT

The authors express their sincere gratitude to Prof. Rashmi P. Bijwe, Department of Information Technology, HVPM College of Engineering and Technology, Amravati, for her valuable guidance and continuous support. They also thank the faculty and students of the department for their participation in the evaluation study and their constructive feedback.

REFERENCES

- [1] J. Doe, "Automated Career Counseling Systems: A Review," *Journal of Career Development*, vol. 45, no. 3, pp. 123-134, Mar. 2020.
- [2] A. Smith and B. Johnson, "Personalized Project Recommendation Systems: An Overview," *International Journal of Educational Technology*, vol. 29, no. 2, pp. 45-58, Feb. 2019.
- [3] M. Brown et al., "Data Preprocessing Techniques for Machine Learning: A Comprehensive Review," *Data Science Journal*, vol. 18, no. 4, pp. 233-245, Apr. 2021.
- [4] M. Qamhieh, H. Samaana, and M. N. Demaidi, "PCRS: Personalized Career-Path Recommender System for Engineering Students," *IEEE Access*, vol. 8, pp. 214039-214049, 2020.
- [5] S. El-Keiey, D. ElMenshawy, and E. Hassanein, "Career Recommendation Based on Feature Selection for Undergraduate Students Using Machine Learning Techniques," *IJACSA*, vol. 16, no. 3, pp. 162-172, 2025.
- [6] C. Lee, "Developing Web Applications with Flask: A Beginner's Guide," *IEEE Software*, vol. 37, no. 6, pp. 56-64, Nov.-Dec. 2020.
- [7] R. J. Oentaryo, X. J. Siddarth Ashok, E.-P. Lim, and P. K. Prasetyo, "JobComposer: Career Path Optimization via Multicriteria Utility Learning," arXiv preprint, arXiv:1809.01062, 2018.
- [8] T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877-1901, 2020.
- [9] H. S. Warnars, P. Siswipraptini, A. Ramadhan, and W. Budiharto, "Personalized Career-Path Recommendation Model for IT Students in Indonesia," *IEEE Access*, 2024.