International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Future Era: An LLM-Driven Career Planning Platform with Structured Roadmap Generation and Progress Tracking

Dr. Madala Guru Brahmam, M. Priyanka, D. Nakshatra Maanasa, M. Likhitha, K. Annapoorna
School of Computer Science and Engineering VIT-AP University, Amaravati, India

Abstract — Career planning can be challenging when a learner has a goal but is unsure which skills to develop, which resources to use, or how to organise the work over time. Relevant infor-mation is often spread across learning platforms, documentation sites, progress trackers, and job portals. This paper presents FutureEra, a web-based career-planning platform designed to bring these activities into one workflow. Users can enter a career goal and profile details, receive a structured roadmap generated with a large language model, explore milestone-related learning resources, record daily activity, compare career paths, use peer-benchmarking and community features, and interact with an AI mentor. The platform is implemented using React and Vite on the frontend, Node.js and Express.js on the backend, and MongoDB for persistent storage. Its generative features use the Groq API with an OpenAI GPT-OSS model selected through the server environment. The application also provides account management, password recovery, progress summaries, and PDF export. Generated roadmaps and estimates are presented as plan-ning suggestions, not as guarantees of salary, completion time, or employment. This paper describes the motivation, related work, system architecture, implementation, current limitations, and an evaluation plan. No quantitative performance or user-study results are claimed because verified experimental measurements are not available for this study.

Keywords— Career Planning, Generative AI, Large Lan-guage Models, Personalized Learning, Progress Tracking, Web Application

I. INTRODUCTION

Choosing a career is only the first step in a much longer process. A student may decide to work towards a role such as software engineer or data analyst, yet still need to identify the skills required, decide what to learn first, estimate the time involved, and find suitable learning material. Similar difficulties arise for career changers and job seekers who are trying to turn a broad goal into a practical plan. When guidance, courses, progress records, and job listings are spread across separate services, users must connect the information and maintain the plan themselves.

Generative artificial intelligence can produce guidance from a user's stated goal and profile. However, a conversational response alone may not be sufficient for sustained planning. Users also need a way to organise recommendations into milestones, revisit their progress, compare alternatives, and find resources related to each stage. Generated information can also contain inaccurate assumptions; therefore, estimates should be presented carefully and should not be treated as certain outcomes.

FutureEra addresses this practical need by combining AI-assisted roadmap generation with learning-resource discovery, milestone completion, daily check-ins, career-path compari-son, peer benchmarking, community interaction, and an AI mentor interface. The platform is intended to support users as they plan and review their next steps. It does not make career decisions on their behalf.

The main contributions of this work are:

- A unified workflow that connects a user-selected ca-reer goal with a structured roadmap, learning resources, progress tracking, and job-resource links.
- A full-stack implementation using React, Express.js, MongoDB, and the Groq API with a configured OpenAI GPT-OSS model for generative features.
- Supporting features for career-path comparison, peer benchmarking, community interaction, daily activity tracking, PDF export, and account recovery.
- A description of the implementation's limitations and a practical evaluation plan for assessing functionality, usability, and roadmap quality.

II. LITERATURE REVIEW

Research on career guidance has explored decision-support systems, recommendation methods, occupational matching,

© 2026 IJSRET

and approaches that consider a learner's interests or personal characteristics. K. et al. present AI Career Twin, a human–AI collaborative framework focused on skill-gap analysis and personalised career guidance [1]. Pan et al. examine career-prospect evaluation using multiple indicators and multi-objective optimisation [2]. Huang and Cheng discuss a stu-dent career-planning decision-support system based on the combination of academic and behavioural data [3]. These studies address different ways of using personal or contextual information to support career-related decisions.

Other work focuses on selecting or ranking possible career paths. Amin et al. propose a decision-support framework for career-path prediction and vocational mobility among col-lege graduates [4]. Samanta et al. describe a hybrid career-recommendation approach that combines ensemble learning with fuzzy-logic-based plausibility assessment [5].

Akbar et al. apply the Simple Additive Weighting method in a web-based decision-support system for programming-career choices [6]. Milla´n et al. discuss a pre-university vocational-guidance platform based on the Ikigai methodology [7]. Together, these approaches illustrate how structured criteria and user-related information can be used to help people explore career options.

Career guidance also involves helping users interpret al-ternatives and connect their interests or profiles with oppor-tunities. Gati and Slama discuss support for making career decisions when moving from a set of options to a considered choice [8]. Sarsenbay et al. present a multidimensional psy-chometric recommender system for student career guidance [9]. Dai and Lin describe a dynamic career-planning decision-support system based on decision trees and reinforcement learning [10]. These works provide relevant perspectives on decision support, recommendation, and the use of multiple factors in career planning.

FutureEra has a related but distinct implementation focus. Its main workflow begins with a career goal entered by the user, generates a time-ordered roadmap, and uses the saved milestones to support resource discovery, progress tracking, comparison, and reflection. The platform is not presented as a new career-prediction model, nor does it claim to determine a user's ideal occupation. Instead, it combines an LLM-generated planning structure with application features that help users review and manage their progress. The cited approaches motivate the broader career-guidance problem; they should not be interpreted as direct evidence that FutureEra produces more accurate recommendations.

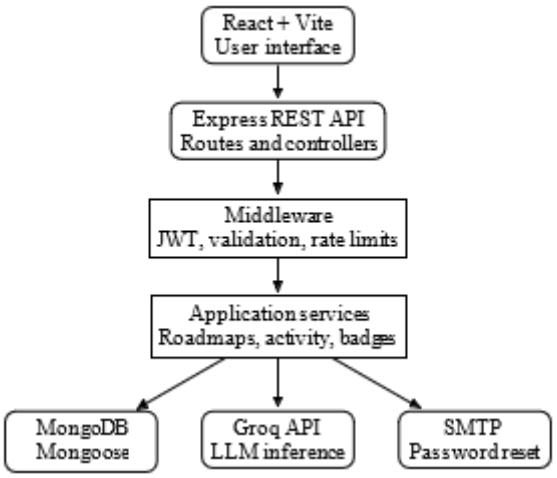


Fig. 1. High-level architecture of FutureEra. The backend coordinates requests between the client, persistent storage, the language-model service, and email delivery.

III. SYSTEM DESIGN AND IMPLEMENTATION

1. Problem Definition and Target Users

FutureEra is intended for students, recent graduates, career changers, job seekers, and lifelong learners who want a more organised way to work towards a career goal. The problem addressed is the gap between identifying a goal and arranging the learning activities that may help a user pursue it.

The system accepts a career goal and relevant profile details, generates a roadmap, and allows the user to work through its milestones. The generated roadmap can contain a suggested duration, estimated cost, expected annual income, weekly workload, risk level, assumptions, and month-by-month tasks. These values are generated estimates. They can vary with location, experience, market conditions, and the information supplied to the model, and should be checked against reliable sources before being used for important decisions.

2. System Architecture

FutureEra follows a client–server architecture. The frontend is built with React and Vite and communicates with a backend REST API. The backend uses Node.js and Express.js to handle requests, apply authentication and validation middleware, co-ordinate application logic, call the AI service when required, and persist application data in MongoDB through Mongoose. The high-level architecture is shown in Fig. 1. The backend coordinates communication between the client, application services, persistent storage, model inference, and password-

© 2026 IJSRET

reset email delivery.

The main roadmap-generation flow is as follows:

- The user enters a career goal and profile information.
- The frontend sends the information to the backend API.
- The backend validates the request and forwards the relevant details to the language-model service.
- The service requests a structured roadmap from the configured model through the Groq API.
- The returned content is processed and checked against the expected response structure.
- The application makes the accepted roadmap available for the user to review and track.

Structural validation checks whether a response has the expected fields and data types. It does not establish that every recommendation or estimate is factually correct.

3. AI-Assisted Career Roadmap Generation

The roadmap feature uses an OpenAI GPT-OSS model through the Groq API. The backend reads the model identifier from the GROQ_MODEL environment setting. The repository's example environment file specifies openai/gpt-oss-120b, while the backend configuration uses openai/gpt-oss-20b as its fallback when the setting is absent. Consequently, the model variant used at runtime depends on the deployment configuration. Before publication, the team should confirm the value configured in the deployed environment and update the repository documentation so that it describes the same model.
A user supplies a target career, optional profile details, and a target timeline. The requested output contains a goal title, estimated duration and cost, expected annual income, weekly workload, risk level, assumptions, and monthly milestones with tasks. The backend attempts to parse the model re-sponse as JSON. A salary-unit guard is applied before schema validation to handle a response that appears to express a monthly amount where an annual estimate is expected. In the implementation reviewed for this paper, the generation flow allows up to two attempts: a suspiciously low salary value on the first attempt leads to a retry, while the corresponding value on the second attempt is adjusted by multiplying it by twelve. The resulting object is then checked against a Zod schema for required fields, data types, ranges, milestones, and tasks. This salary adjustment is a heuristic and should not be understood as independent verification of the estimate.

These checks improve structural consistency, but they can-not establish that a salary, cost, or timeline is factually correct. A response may satisfy the schema and still contain an unre-alistic estimate. Users should verify important details against current official sources, employer information, and labour-market data relevant to their location.

Algorithm 1 summarises the two-attempt control flow described above. It distinguishes the salary-unit check from the subsequent Zod schema validation. The algorithm is a representation of the implementation logic, not a claim that the model itself has been trained or optimised by this project.

Algorithm 1 AI-Assisted Career Roadmap Generation and Validation

Require: Career goal g, profile details p, target timeline d, model m
Ensure: Validated roadmap R or generation error

$R \leftarrow$ null
$i \leftarrow 1$
while $i \leq 2$ **and** $R =$ null **do**
  Construct prompt using g, p, and d
  if $i = 2$ **then**
    Append the retry instruction for annual salary in INR and valid JSON
  end if
  Request JSON response from model m through Groq API
  if response is empty **then**
    $i \leftarrow i + 1$
    continue
  end if
  Parse response using JSON parser
  if JSON parsing fails **then**
    $i \leftarrow i + 1$
    continue
  end if
  Extract estimated salary s
  if $0 < s < 100000$ **then**
    if $i = 1$ **then**
      $i \leftarrow i + 1$
      continue
    else
      $s \leftarrow 12s$
      Update salary in response
    end if
  end if
  Validate response using Zod schema
  if schema validation succeeds **then**
    Set missing weekly workload to 20 hours
    $R \leftarrow$ validated roadmap
  else
    Keep R as null and proceed to the next attempt
  end if
  $i \leftarrow i + 1$
end while
if $R \neq$ null **then**
  Return R to application
else
  Return generation error
end if

The application also provides an AI mentor interface called "Evolve Me" for follow-up career-planning conversations. Its

© 2026 IJSRET

prompt can use saved information such as the user's career goal, path status, completed and remaining milestones, work-load, and roadmap estimates. Some progress-related values are read from stored application data and may not reflect the latest recalculated frontend indicators. The mentor is intended to provide contextual planning suggestions; it is not a prediction of a user's future performance or career outcome.

4. Learning Resources and Job Opportunities

The platform supports milestone-specific learning-resource discovery through external search links. For a selected mile-stone, the frontend uses the career goal and milestone descrip-tion to construct Google and YouTube search queries. This allows users to discover potentially relevant tutorials, courses, documentation, and practice materials. The application does not verify or curate the results returned by these external

5. Progress Tracking and Career-Path Comparison

Users can record milestone completion and daily check-ins. The application tracks activity streaks, displays activity history, and supports achievement badges. These features make recorded progress visible and are intended to encour-age consistent engagement. Recorded activity alone does not demonstrate that a skill has been mastered.

The application provides pace indicators and progress sum-maries using saved career-path and milestone information. These indicators are interface-level aids; their thresholds have not been empirically calibrated in the evaluation reported here, and they should not be interpreted as measures of skill mastery. Users can create multiple career paths and compare at-tributes such as estimated duration, cost, salary, and risk. The application also includes peer benchmarking and community interaction for users pursuing similar goals. These features support comparison and discussion, but generated estimates and peer comparisons should be treated as decision aids rather than definitive career rankings.

6. Account Management and Security

The application includes signup and login, JWT-based ac-cess and refresh-token functionality, password hashing with bcrypt, request validation using Zod, rate limiting on sensitive routes, security headers through Helmet, and a restricted cross-origin policy. The password-reset flow generates a time-limited token and sends a reset link through an email service using Nodemailer and Gmail SMTP.

These controls address common web-application risks when configured correctly, but their presence alone does not prove that an application is secure. Security testing, dependency maintenance, deployment review, and careful handling of environment variables remain necessary. The project should also ensure that sensitive tokens are stored and handled using appropriate protections; this paper does not claim that every security control has undergone an independent audit.

7. Technology Stack

Table I summarises the principal technologies used in the application.

Table 1: Main Technologies Used in Futureera

Technology	Purpose
React and Vite	Frontend interface and build workflow
React Router	Client-side navigation
Tailwind CSS	Responsive interface styling
Axios	HTTP communication with the backend
Node.js and Express.js	Backend API and request handling
MongoDB and Mon-goose	Persistent data storage and models
Groq API and OpenAI GPT-OSS	Roadmap generation and AI mentor
Zod	Request and response structure validation
JWT and bcryptjs	Token-based authentication and password hashing
Nodemailer	Password-reset email delivery
Recharts and jsPDF	Visualisations and PDF export

IV. IMPLEMENTATION STATUS AND EVALUATION PLAN

The current implementation integrates roadmap genera-tion, saved milestones, external learning-resource discovery, progress tracking, career-path comparison, peer benchmarking, community interaction, daily activity features, and AI mentor conversations into one application workflow. Their inclusion describes the implemented application features; it does not imply that every feature has undergone formal usability, per-formance, or security testing. At the current stage, the work primarily reports the system design and implementation rather than an experimentally validated career-guidance model.

No quantitative benchmark, controlled user study, or mea-sured generation-success rate is reported in this paper. Ac-cordingly, this work does not claim that FutureEra improves career outcomes, produces more accurate recommendations than other systems, or increases engagement by a measured amount. Those claims would require a separate evaluation. A reproducible evaluation could measure (i) JSON parsing and schema-validation success across a defined set of career goals, (ii) retry frequency and response latency, (iii) roadmap relevance, ordering, and feasibility assessed by independent reviewers, (iv) the plausibility of duration, cost, and income

© 2026 IJSRET

estimates when compared with current public sources, and (v) usability through representative task-completion activities and participant feedback. The number of prompts, model configuration, reviewer rubric, participant count, and testing conditions should be reported with any results.

A functional test plan should also cover registration, login, duplicate-account handling, password recovery, valid and invalid model responses, roadmap persistence, milestone completion, daily check-ins, streak updates, career-path comparison, and PDF export. These are proposed test cases; they should only be reported as passed after the corresponding tests have actually been executed.

V. DISCUSSION

FutureEra brings several parts of career planning into one interface. Rather than managing a plan, resource links, progress records, and job searches separately, a user can access these features through the same application. Saved milestones connect the generated roadmap with the user's recorded activity. Career-path comparison also gives users a way to examine more than one possible direction before deciding where to focus their effort.

The system has limitations. A language model may suggest an unsuitable resource or produce estimates that do not fit a user's circumstances. Schema validation checks structure, not factual correctness. External job links can become outdated, and activity streaks represent recorded activity rather than verified learning. Salary and timeline estimates may vary with location, experience, market conditions, and personal circumstances. Transparent assumptions, editable plans, and independent verification are therefore important when using the platform for career planning.

VI. CONCLUSION AND FUTURE WORK

This paper presented FutureEra, an AI-assisted career-planning platform that connects roadmap generation with learning resources, milestone tracking, daily activity, career-path comparison, peer benchmarking, community interaction, an AI mentor, and job-resource links. The application uses a React-based frontend, an Express.js backend, MongoDB for persistence, and an OpenAI GPT-OSS model accessed through the Groq API. It also includes account-management features such as password hashing, token-based authentication, request validation, rate limiting, and password recovery.

Future work includes systematic functional and security testing, a user-centred usability study, and a rubric-based assessment of roadmap quality. Other useful improvements include checking external links, recording sources for factual career information, testing a wider range of user profiles and goals, and measuring response time under defined conditions. Such evidence is needed before drawing conclusions about the platform's effectiveness or its impact on career outcomes.

Acknowledgment
The authors thank Dr. Madala Guru Brahmam, Assistant Professor, Senior Grade 1, School of Computer Science and Engineering (SCOPE), VIT-AP University, for guidance and feedback during the project.

REFERENCES

1. V. K., S. S., S. Rao, S. Raghavendra, and D. D. S. Shaji, "AI Career Twin: A Human–AI Collaborative Framework for Dynamic Skill-Gap Analysis and Personalized Career Guidance," 2026. [Online]. Available: https://doi.org/10.5281/zenodo.23341915.
2. J. Pan, Y. Zhang, X. Song, and X. Qiu, "Comprehensive career prospect evaluation under Gen-AI-based multi-indicator weighting and multi-objective optimization," in Proc. SPIE, vol. 14326, Art. no. 143260K, 2026, doi: 10.1117/12.3120249.
3. X. Cheng and J. Huang, "Research on Adaptive Decision Support System for Student Career Planning Based on Multi-Source Academic and Behavioural Data Fusion," Decision Making: Applications in Man-agement and Engineering, vol. 9, no. 1, pp. 270–290, 2026.
4. M. Amin, M. S. Nazeer, K. Ullah, Y. Shang, and D. Pamucar, "A decision support framework for optimizing career path prediction and vocational mobility of college graduates," PeerJ Computer Science, vol. 11, Art. no. e3345, 2025, doi: 10.7717/peerj-cs.3345.
5. A. Samanta, S. Patnaik, S. Mohanty, R. R. Panda, and K. Nikhil, "A Hybrid Predictive Framework for Career Path Recommendation using Ensemble Learning and Fuzzy-Logic Based Plausibility Assess-ment," in Proc. 2025 13th International Conference on Intelligent Systems and Embedded Design (ISED), Dec. 2025, pp. 199–205, doi: 10.1109/ISED67359.2025.11404754.
6. M. A. Akbar, R. Rusli, Y. A. Wahid, D. F. Surianto, and F. Adiba, "Optimizing Career Choices in the World of Programming: A Web-Based Decision Support System with the Simple Additive Weighting (SAW) Method," Komputika: Jurnal Sistem Komputer, vol. 13, no. 2, pp. 223–233, 2024.

© 2026 IJSRET

7. A. Milla´n, J. Garc´ıa-Unanue, and M. Retamosa, “Which career should I choose? Application of a pre-university vocational guidance platform based on the Ikigai methodology,” in Proc. International Conference on Higher Education Advances, Jun. 2023, pp. 533–540.
8. I. Gati and M. Slama, “From Options to Choices: Helping Clients Make Better Career Decisions,” Canadian Journal of Career Development, vol. 25, no. 1, pp. 100–117, 2026, doi: 10.82396/cjcd.v25i1.3272.
9. S. Sarsenbay, I. Varlamis, C. Turan, B. Razhametov, and Y. Kazym, “Optimizing Academic Trajectories: A Multi-Dimensional Psychometric Recommender System for Student Career Guidance,” Informatics, vol. 13, no. 6, Art. no. 81, Jun. 2026.
10. J. Dai and L. Lin, “Dynamic Career Planning Decision Support System Based on Decision Trees and Reinforcement Learning,” in Proc. 2025 International Conference on Digital Society and Intelligent Computing, Nov. 2025, pp. 390–396.

© 2026 IJSRET