iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81844

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

AI-Based Roadmap Generator

Mr. Abhishek Badiger[1], Mr. Sumit Deokar[2], Mr. Subodh Dashmukhe[3], Mr. Soham Poshiya[4], Mr. Jitendra Garud[5]
Computer Department, DYPIEMR, D. Y. Patil Educational Complex, Akurdi 411044, Pune, Maharashtra

Abstract: These days, more people are logging in to learn new things. Platforms everywhere offer videos, texts, books - a mix of tools at your fingertips. Easier access opens doors, sure. Still, plenty find themselves wandering without clear direction. Starting feels impossible for them. Picking useful subjects? That part trips them up too. Staying on track week after week - another hurdle they face. Without clear direction, attention fades fast, curiosity dims. Enter a tool built smart: artificial intelligence shapes custom paths forward. Step by step, study plans come together more easily. A tailored route unfolds - clean, straightforward. Built around what one person actually requires. Tools like machine learning shape how things work behind the scenes. Language understanding tech plays a role too. Big model systems help make sense of inputs. Information flows in: objectives show up first. What someone already knows gets recorded next. Pace matters. So does free time. All these pieces feed into the analysis that follows after. Later on, a custom study schedule takes shape. Based on what's gathered, a straightforward path unfolds for learning. Simple to move through, this path stays flexible. Unlike old-style approaches, it shifts as needed. Progress gets monitored along the way. Topics adjust. So do deadlines. Practice tips come next, shaped to fit your pace. Broken into steps, the path forward feels less crowded. Every step points at a target you can name. Tiny checkpoints appear along the way. Staying on track gets easier when progress shows clearly. Clicking through, you meet tools that respond as you go. What you do returns to you, right away. Skills get checked while progress moves forward. Discussion spots open up along the way. Quick support shows up through an AI helper. Learning paths shift based on what fits next. Missing pieces in understanding come into view. Videos pop up, then articles, followed by quizzes and hands-on work. What you complete stays saved without extra steps. A strong outline of skills takes shape here. For schoolwork or job paths, that brings clear benefits. Flexibility marks how it works - simple to use, built around you. Each person finds their own pace without pressure. Steps become clearer, effort feels less scattered. Confidence grows quietly along the way.

Keywords: Artificial Intelligence, Machine Learning, Natural Language Processing, Large Language Models, Personalized Learning, Adaptive Systems, Retrieval-Augmented Generation, Recommendation Systems, Learning Path Design, Educational Technology, Intelligent Tutoring, Progress Monitoring, Skill Evaluation, Chatbots, E-Learning.

I. INTRODUCTION

Digital education platforms are growing very fast. They make learning easy to access. Many people can now study online. But there is still a problem. Learners do not get proper guidance. They also do not get personal support. Most e-learning platforms use the same method for everyone. They give the same content to all users. They follow the same order. They do not check the learner's background. They do not adjust for learning speed. They also ignore personal goals. This creates confusion. It slows down learning. It also reduces course completion. Traditional systems use fixed courses. These courses do not change. They are not flexible. Research shows that adaptive learning works better. Intelligent tutoring systems also support this idea. These systems increase engagement. They also improve learning results [8], [9]. Most tools fail to adjust instantly. Yet they miss tailored experiences too. To tackle that, recommendation engines step in. By watching what users do, these systems offer fitting choices. One way they work? Through collaborative filtering. One way to go about it? Matrix factorization. Preferences become clearer through methods like these [10], [15]. Beyond that, neural nets step in - uncovering hidden structures in information [5]. When sequences matter, different models take over. Helping where order plays a role. Learning comes in steps. That matters when teaching [7]. Text machines have gotten smarter over time. These days, most rely on transformers. Understanding words? Much easier now. BERT makes sense of meaning [3]. GPT builds sentences that flow naturally [2]. Yet limitations exist in such systems. Wrong responses might appear now and then. To lower that risk, Retrieval-Augmented Generation steps in - pulling outside information into the mix [4]. Different smart approaches help too. As learners move forward, knowledge tracing keeps an eye on their path. Feedback shapes how reinforcement learning improves the system. Performance guides changes in the approach, seen in studies [6] and [12]. Built on such concepts is this work's core idea - a roadmap maker powered by artificial intelligence. Natural language processing joins big language engines, suggestion logic, plus info search techniques inside it. Input comes first: what someone wants to achieve, their current ability, along with hours they can commit.

After that, a detailed learning path takes shape, piece by piece. Clear direction shows up right away, built in layers. Each part fits the person using it. One method blends creating new material with adjusting what exists. Modern tools for learning get better because of this mix. Simple ways to learn stand out, shaped around individual needs.

II. EASE OF USE

A. User Interface Design

A single screen holds everything you need. Clicking through takes almost no time at all. Pick what you want to learn first, then adjust how deep you go. Some spaces guide your typing with hints nearby. No clutter shows up around the edges. Choosing where to start feels obvious after one look. Each box waits quietly until touched. Steps appear only when needed, never before. Starting out feels clear, almost natural. Right away, new people get what to do because guidance unfolds piece by piece. This path cuts through confusion without fuss. Each move follows the last like footsteps on a quiet road. Thinking less means paying attention more. The way things work stays in the background, unseen but steady. Learning takes center stage when mechanics fade away.

B. Automated Resume and Profile Onboarding

Most of the heavy lifting happens behind the scenes. A step-by-step format leads the way forward. Filling out long forms becomes unnecessary. Simple answers are enough to move ahead. Starting off, there's a mix of domain choice plus how skilled someone is. Pick from menus that drop down, each with set answers only. Ease comes through these ready-made picks. Fewer mistakes happen because choices are limited on purpose. Speed increases since typing isn't needed most times. People who aren't tech-focused still get it right the first try.

C. Personalized and Guided Preparation Flow

Starting at the beginning, the setup builds a route shaped by what each person shares. Too many options never appear - clarity stays intact because of that. A sequence unfolds instead, one where steps link naturally ahead. Broken into stages like early, middle, and later phases, progress feels smooth. Logic threads through each part without breaking. Understanding grows because the journey moves forward in order.

D. Interpretable and Actionable Feedback Presentation

Right off the start, the layout feels clean and simple to go through. Step by step, everything shows up marked just right, tied together in order. No clutter sneaks in, nothing spills across like spilled ink on paper. Sections split the path logically, guiding eyes forward without hiccups. What comes next pops out naturally, making moves feel smooth when putting them into practice.

E. Low Friction Iteration and Progress Tracking

Any moment works for tweaking what you entered - new roadmaps pop up fast, no tangled procedures. Trying another goal? Switching fields? That opens doors to fresh ways of learning, reshaping your route on the fly. Simple moves keep people coming back, making steady growth feel natural.

III. LITERATURE REVIEW

A. Personalised Learning Path Systems

Learning paths shaped around individual students take into account what they already know, how they prefer to learn, yet move at a speed comfortable for them. Research indicates such responsive setups boost attention along with outcomes when set beside rigid formats [8], [9]. Information gathered from student activity helps reshape lessons over time. Still, plenty of today's platforms struggle to shift promptly if a person's habits evolve. That slows how well they work in actual classrooms. For this reason, smarter approaches driven by live feedback could meet shifting demands within online education more effectively.

B. LLM-Based and RAG-Augmented Generation

Learning platforms often rely on recommendation tools to tailor materials and shape individual study routes. While older techniques like collaborative filtering and matrix factorization help map user interests [10], [15], newer neural networks uncover deeper behavioral links, lifting forecast quality [5]. Progress continues with sequence-sensitive designs that treat lesson order as meaningful - key in teaching contexts [7]. Still, even advanced options usually run on fixed data collections. Because of this, few can build fresh material on demand or shift quickly when student demands evolve.

C. Genetic Algorithms and Optimisation Approaches

Modern language systems now rely heavily on transformer designs, shifting how machines process words. Though built differently, each version builds on earlier ideas to handle text more naturally. Instead of just linking phrases, the first transformer [1] captured deeper connections across sentences. By focusing on both direction and position, BERT grasps intent in queries much more accurately [3]. In contrast, GPT takes a generative path, producing responses that flow like everyday speech [2]. When applied carefully, such models help shape custom learning paths based on individual needs. Yet one key problem remains unresolved. Wrong answers may come from big language tools now and then. When schools rely on them, trust becomes an issue - accuracy matters too much to ignore. These flaws show up just when you expect correct facts most.

D. Knowledge Tracing and Graph-Based Methods

Over time, knowledge tracing tracks shifts in a student's grasp of material, forming a core piece of adaptive education tools. Instead of relying on set paths, models like Deep Knowledge Tracing use deep learning to forecast how well a student will do later while guiding what they should study next [6]. Feedback loops refine choices - reinforcement learning tweaks the route as new responses come in [12]. As progress unfolds, personalization stays active, offering fluid experiences rather than rigid structures.

E. Retrieval-Augmented Generation (RAG)

Instead of relying only on internal knowledge, some systems pull facts from outside sources before responding. By searching through stored data, they locate useful information linked to the query. After finding matching content, the model weaves it into its answer. Using meaning-based matching, these tools rank documents by relevance. Vector representations help measure how closely ideas align. The final output builds upon both retrieved text and learned patterns. Such methods aim to reduce errors found in responses made from memory alone. Because it relies on actual data, mistakes happen less often while accuracy increases. Learning paths that emerge from this method rest on confirmed facts. For these reasons, the process gains trust and fits well within serious teaching environments.

IV. METHODOLOGY

Starting off, the system builds a learning path using artificial intelligence through separate connected stages. Instead of one single block, it splits tasks into pieces like reading user details, linking abilities, shaping requests, producing content via advanced language tools, then boosting suggestions. Models based on transformers handle meaning and order in data [1], [3], whereas smart modules support decisions across different sizes of demand [24]. Inputs from users - such as field of interest, current proficiency, aims, and hours free - are gathered, verified, grouped into clear subject clusters. Next, a tailored request forms and moves to a powerful text generator, returning a staged plan for study. That result gets cleaned up so it reads smoothly. On top, another piece slots in useful materials for learning. At last, users see the full path laid out. Because it's built in pieces, the setup bends without breaking, grows when needed, yet stays steady through changes.

A. System Architecture Overview

From the start, user details like field of interest, current expertise, objectives, and schedule enter through a simple screen. Once gathered, these pieces go through checks and clean-up steps before moving forward. Instead of jumping straight into results, the system links each goal to key abilities needed. A smart model builds prompts only after sorting raw answers into clear themes. Because transformers handle pattern recognition well, they guide how ideas connect across stages. With strong sequence logic baked in, it manages flow just like today's top language tools do. Each decision adapts smoothly thanks to built-in learning modules that adjust on their own. Efficiency grows because parts work alone yet fit together when combined. At every turn, structure supports speed without losing accuracy.

Starting off, someone puts together a clear instruction for a big language tool. That message goes out to the machine brain waiting on the other side. Step by step, it builds a path showing how to learn something new. What comes back gets cleaned up so ideas flow in order. From there, extra materials get attached - things like videos or readings that fit well. These extras slide into place without breaking rhythm. In the end, everything reaches the person who asked. Each piece works alone but fits together when needed. Parts can shift easily if demands change later.

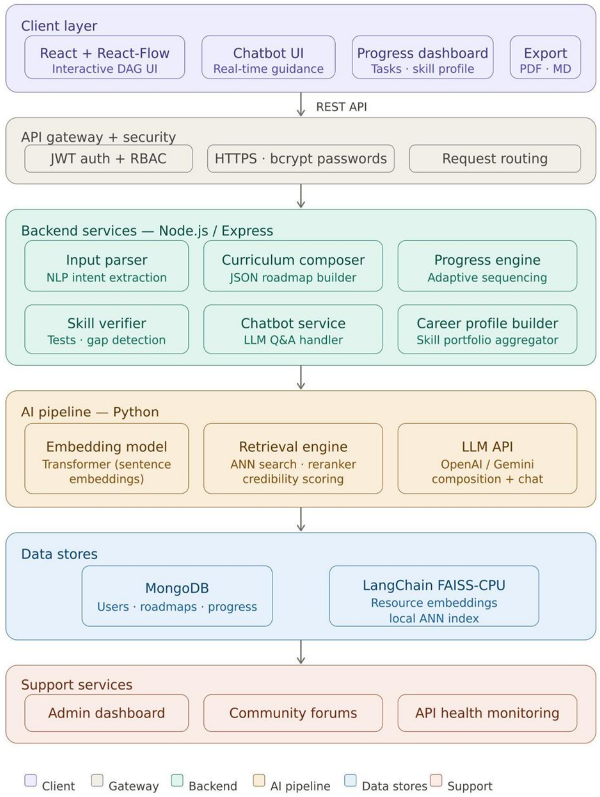


Fig 1. System Diagram

B. *Input Acquisition and Preprocessing*

From details like your target skills, free hours each week, what you already know, how you pick up new ideas best, along with any extra topics you add - this flow pulls together personal info. Once gathered, every piece gets reviewed for errors, tidied up, turned into a common format.

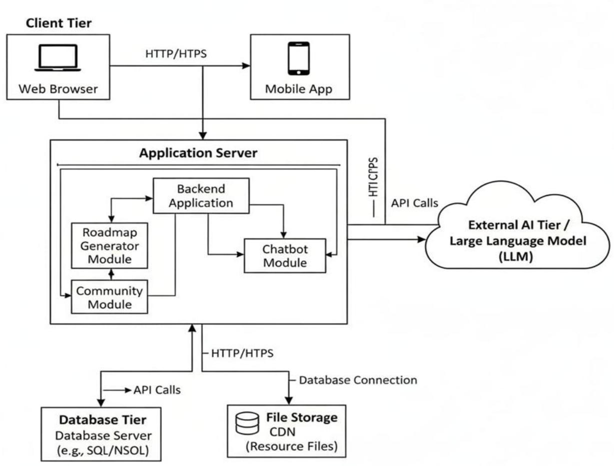


Fig 2. User input diagram

Consistency stays strong through the whole setup because of it. Downstream parts receive clean, uniform information ready to work with. That is how reliability spreads forward.

C. Prompt Generation

Out of sorted inputs and matched abilities comes a shaped prompt. Built with care, it steers the model to deliver responses that make sense at a glance. How the request is built shapes how the large language machine answers. When GPT systems get thought-out cues, their words fit the situation better [2]. That detail matters more than it might seem. A solid setup leads to steps that flow naturally, line up right, then feel simple to move through.

D. Roadmap Generation Using LLM

A clear prompt goes into a large language model, leading it to build a detailed learning plan piece by piece. Because of transformer designs, the system grasps meaning across text stretches, shaping outputs that flow well [1], [2]. Still, at times these models invent answers or state false details. To lower such chances, the setup pulls facts from outside resources before responding - this method is known as Retrieval-Augmented Generation (RAG) [4]. With responses tied to trusted references, results become sharper, making each suggested path stronger and better grounded.

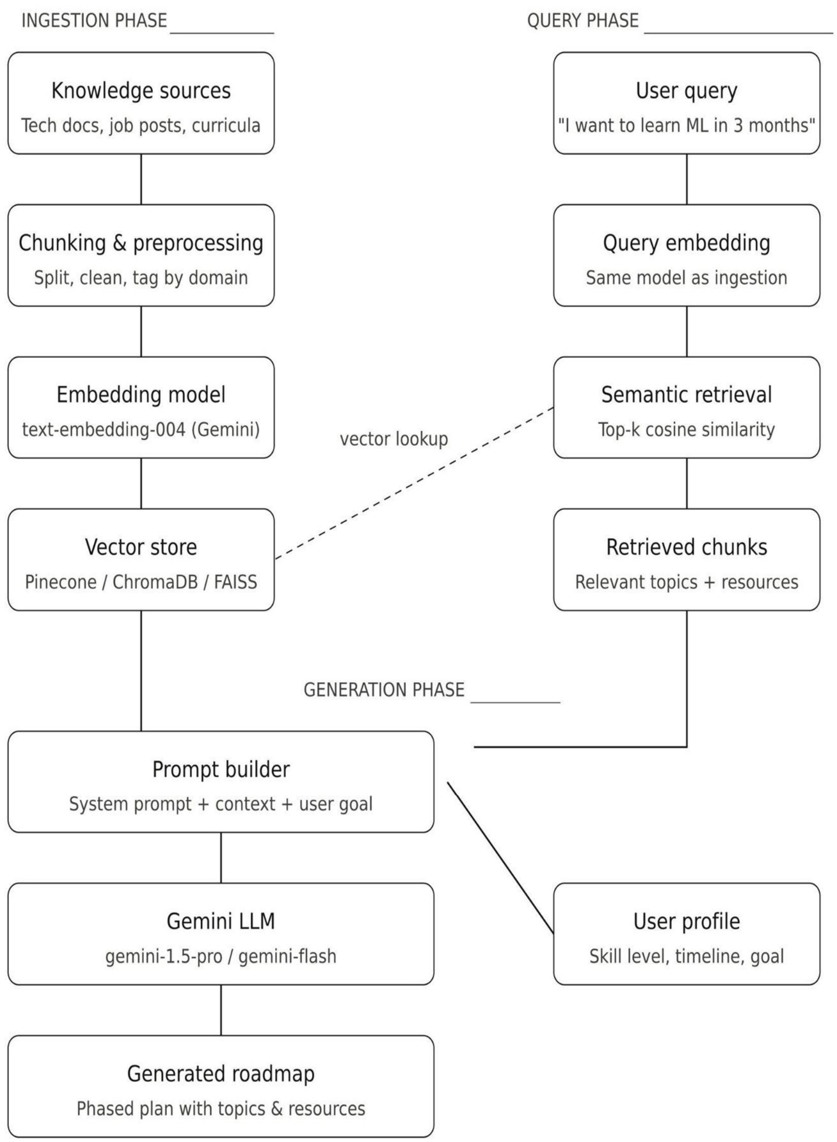


Fig 3. Rag Pipeline Diagram

E. Workflow of the System

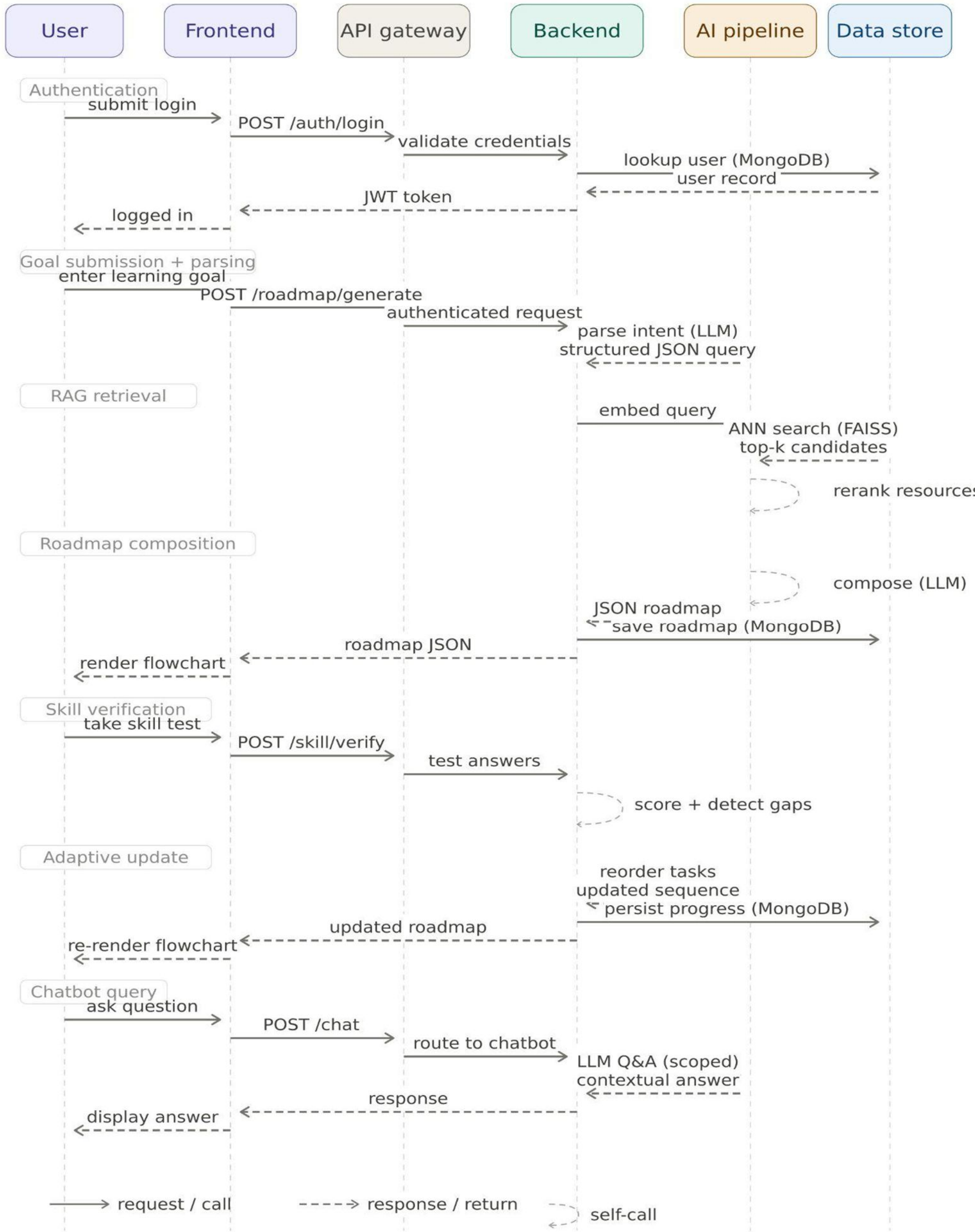


Fig 4. Sequence Diagram

Most of the time, pulling facts from a reliable source makes answers better. Instead of guessing, the model checks an outside database before responding. That way, what gets created ties back to real data people can trust. Using live sources keeps details sharp. Information flows in right when needed. Answers shaped this way rest on stronger ground.

F. Post-Processing and Formatting

Out of the gate, messy text from the model gets cleaned up. One step at a time, pieces are broken down so each part stands alone. Sections appear where they make sense, shaped by how things connect. Duplicates fade out, extra bits vanish without warning. Structure rises quietly, guiding the eye before you even notice. Clarity shows up not because it's promised, but because clutter left the room. Reading feels smoother now, using less effort than before. People move through it easier, almost like second nature.

G. Recommendation Module

The module suggests relevant learning resources for each user. It uses different recommendation techniques to improve personalization. Collaborative filtering and matrix factorization are used as basic methods for personalization One way the system works is by showing users learning materials they might like. Personalization gets a boost through varied approaches behind the scenes. Instead of just one method, it combines several strategies quietly. For instance, patterns in user behavior guide suggestions using known models [10], [15]. A deeper layer comes alive when neural networks map subtle connections between choices [5]. Learning patterns get sharper when deep methods shape the suggestions [17]. Instead of explicit ratings, progress comes from silent cues - what users do speaks louder, thanks to Bayesian ranking tricks [19]. As steps unfold one after another, sequence models track that flow, matching materials to movement through topics [7].

H. Adaptive Learning Mechanism

When users move forward, the roadmap shifts too - feedback shapes each change. Over time, what a person knows gets mapped out, helping forecast next steps [6]. As performance shows up in data, adjustments follow, guided by patterns spotted along the way [12], [23]. With every twist, the path bends closer to the learner's rhythm. Personal touches grow stronger here. Efficiency rises without drawing attention to it.

I. Testing and Validation

Across fields like building websites and working with data, tests were run. From those starting out to people getting more practice, checks happened too.

Checking each output took place to confirm it made sense, stayed clear, kept on topic. Out of every test run came organized steps people can actually use - each one built like a path you could walk without getting lost.

V. RESULT ANALYSIS

The proposed AI-Based Roadmap Generator was evaluated for its ability to generate personalized, adaptive, and structured learning paths across users with varying goals, skill levels, and time constraints.

A. Roadmap Generation Quality

Outputs from the suggested system formed the basis for examining how well roadmaps could be created. These were measured against basic versions made through Google Gemini alone, absent any organized prompt design. Focus fell on three points - how information was arranged, its connection to the topic, its extent of coverage. One aspect looked at clear organization within each output. Another considered whether details matched what was asked. A third judged if major elements appeared present

Table no. 1 Roadmap Generation Quality

Metric	Proposed System	Baseline
Structure	4.5 / 5	3.2 / 5
Relevance	4.3 / 5	3.5 / 5
Completeness	4.2 / 5	3.4 / 5

Starting from scratch, the initial outputs came through straightforward questions given to Google Gemini. Without any framework guiding their structure, these inputs followed a flat approach - no layers, no segmentation. Each step stayed untouched by systematic planning. Differences stood out plainly in what came back. Outputs come out clearer under the new setup. With a defined path forward, progress feels less scattered. The topic sequence now follows a logical flow. By using fixed prompt formats, alignment improves. User intentions link more naturally to results. Because it covers every key point needed. Topics follow one another in a way that makes sense.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

B. Personalization Analysis

Each test adjusted how the system responded to individuals. Various types of input shaped each run. Through it all, one aim stayed fixed. Sometimes sessions ran longer, sometimes shorter. Skill background shifted between trials. What came out in steps became clearer after watching closely. When

time is tight, the path shrinks. Shorter timelines mean fewer steps along the way. Key ideas take center stage in these cases. Extra details fade out. More hours allow a fuller picture to form. Additional pieces come into view then. Depth grows where time permits. For beginners, the journey begins with core ideas. Starting points matter most at this stage. Moving up, early material gets left behind. Focus shifts toward deeper subjects instead. From how it responds, you can tell the system pays attention to what users say. Because of this awareness, learning routes adjust naturally. When needed, changes fit who is using it.

C. Progress Tracking and Productivity Evaluation

The system was tested using simulated user data. This data shows user interaction. The system tracks progress using module completion. It also measures productivity. It uses completion rate, consistency, and time usage for this.

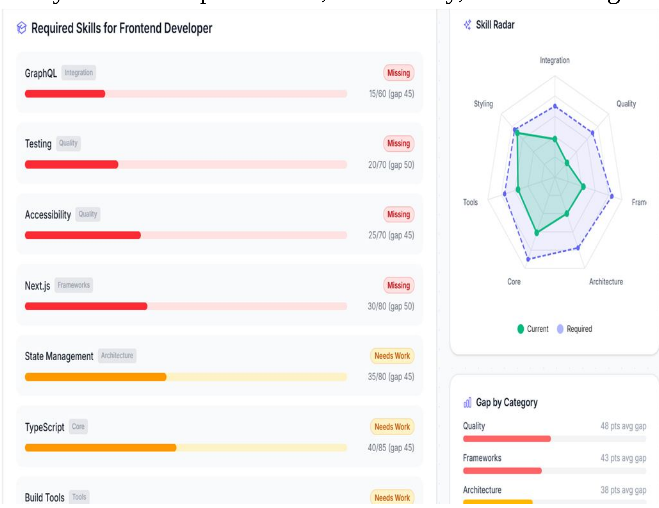


Fig 5. Progress tracking

The productivity score is computed as a weighted combination of completion rate, consistency, and time efficiency:

$S = 0.5C + 0.3K + 0.2T$

C stands for how much of the course someone finishes, figured by dividing done parts by all parts. Though it tracks progress, what matters is steady effort over weeks. Regular login patterns shape K, which counts how many days a person studies versus how many they could have.

When behavior spreads evenly across the timeline, values rise. Efficiency in timing shows up in T, drawn from real minutes spent compared to scheduled ones.If sessions match plans closely, scores adjust upward. Each piece feeds into the full picture without needing extra data points. Patterns emerge only when all three align over repeated cycles.Surprisingly, those who stayed active scored better across tasks.

Table no. 2 Production Value

User	Modules Completed	Progress (%)	Consistency (%)	Productiviy Score
A	8 / 10	80	85	82
B	5 / 10	50	60	56
C	9 / 10	90	88	87

When learners finished more modules, their outcomes improved noticeably. Notably, steady effort matched up closely with stronger performance numbers. This pattern supports the method used to measure progress.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

D. System Performance

Most of what we tested pointed to one thing first: how fast it replied. Was real work possible on it? Every check followed that path. Demands from the roadmap arrived in many forms. The moment each launched, it was marked with a time. Every time a question came in, it got recorded on the spot. About two or three seconds later, answers appeared. Once the input entered, the system worked through its steps before shaping the response. The moment unfolded smoothly, showing what was ready. Fast enough that delays never felt noticeable while things were happening. Most folks noticed they did not have to wait much at all. When delays start to build, stored data steps in just in time. A well-thought-out structure helps speed up progress as it moves forward. What lies ahead gets easier when pieces fall where they should.

E. User Engagement and Usability

Inside this system hides a useful feature. As you take routine ability checks, status changes show right beside them. When queries pop up, an automated helper responds instantly. Step by step, diagrams reveal what comes next. Things start clicking once folks see them in action. When users gave the interface a go, uncertainty faded fast. Clarity came easier - almost like stepping into a lesson that knows your pace [8].

F. Qualitative Comparison of Outputs

Out of the two results - basic versus new - one stands apart. Clearly, they are not the same.
Input: Learn UI/UX Design roadmap in 6 months.
Output: Created through the method outlined in the new planning framework
A fresh timeline takes shape, one that flows with clearer steps and moments woven together.

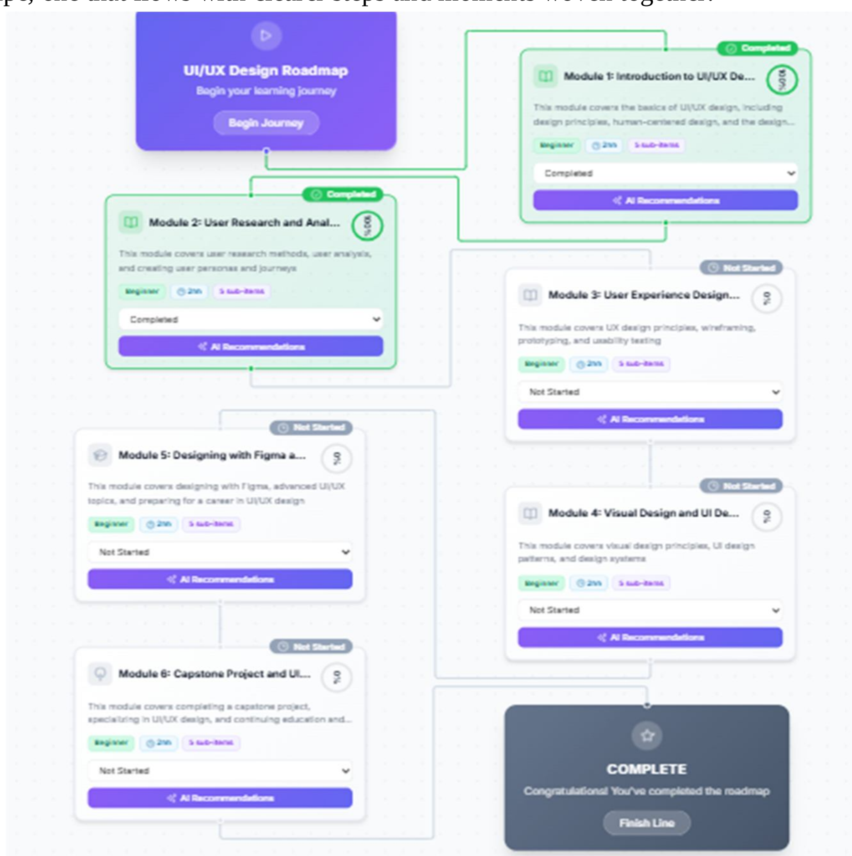


Fig 6. UI/UX Design Roadmap

This path shows what can happen when ideas build in phases, each feeding the next. Moments gain meaning through order, not just listing. Step by step, the method proves its worth without grand claims. Structure emerges quietly, shaped by timing and purpose.

VI. CONCLUSION AND FUTURE SCOPE

A fresh approach emerges here - using smart tools to shape how people learn. Instead of fixed routes found on most platforms, this tool shifts with each person's needs. Starting from where they stand, what they want, and which area draws them in, it builds steps forward. Words are understood by machines that trace meaning, while models craft next moves tailored precisely. Recommendations slot into place smoothly, forming a clear line through topics. Flow matters just as much as content, keeping interest alive without overwhelming. One field works like another in its eyes - the structure bends freely between coding, job growth, or picking up new abilities Still, it struggles sometimes when the input isn't clear, often producing awkward phrasing here or there. Right now, adjusting instantly based on how users react doesn't work very well.One step ahead might mean watching progress as it happens, shaping how topics flow by using smart maps of ideas instead of fixed sequences. Clarity could come from systems that show their reasoning, making choices easier to follow. When different areas of learning open up, customization finds more room to grow. Learning adjusts better when it listens, building improvements from real responses over time.

Overall, the system demonstrates considerable promise in advancing personalized learning by offering an intelligent, adaptive, and learner-centered solution.

VII. ACKNOWLEDGMENT

The authors would like to sincerely thank their project guide and faculty members for their constant support, helpful guidance, and thoughtful suggestions throughout this research work. Their encouragement and knowledge played an important role in shaping both the direction and quality of the study.They also feel grateful to their institution for providing the resources and a supportive environment that made this work possible. In addition, the authors truly appreciate the encouragement and useful feedback from their friends and peers.Finally, they would like to thank everyone who, in one way or another, contributed to the successful completion of this project.

REFERENCES

Here are the corrected, properly numbered references for your paper:

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998–6008.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.

[4] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in Proc. 26th International World Wide Web Conference (WWW), 2017, pp. 173–182.

[6] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2015.

[7] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in Proc. ACM Conference on Recommender Systems (RecSys), 2017.

[8] S. D'Mello and A. Graesser, "AutoTutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers," ACM Transactions on Interactive Intelligent Systems, vol. 2, no. 4, 2012.

[9] R. S. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," Journal of Educational Data Mining, vol. 1, no. 1, pp. 3–17, 2009.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, 2009.

[11] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2019.

[12] R. Sutton and A. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[14] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook. New York, NY, USA: Springer, 2015.

[15] J. L. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," Information Retrieval, vol. 5, no. 4, pp. 287–310, 2002.

[16] K. Zhou, H. Yang, F. Wu, and X. Xie, "Deep interest network for click-through rate prediction," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2018.

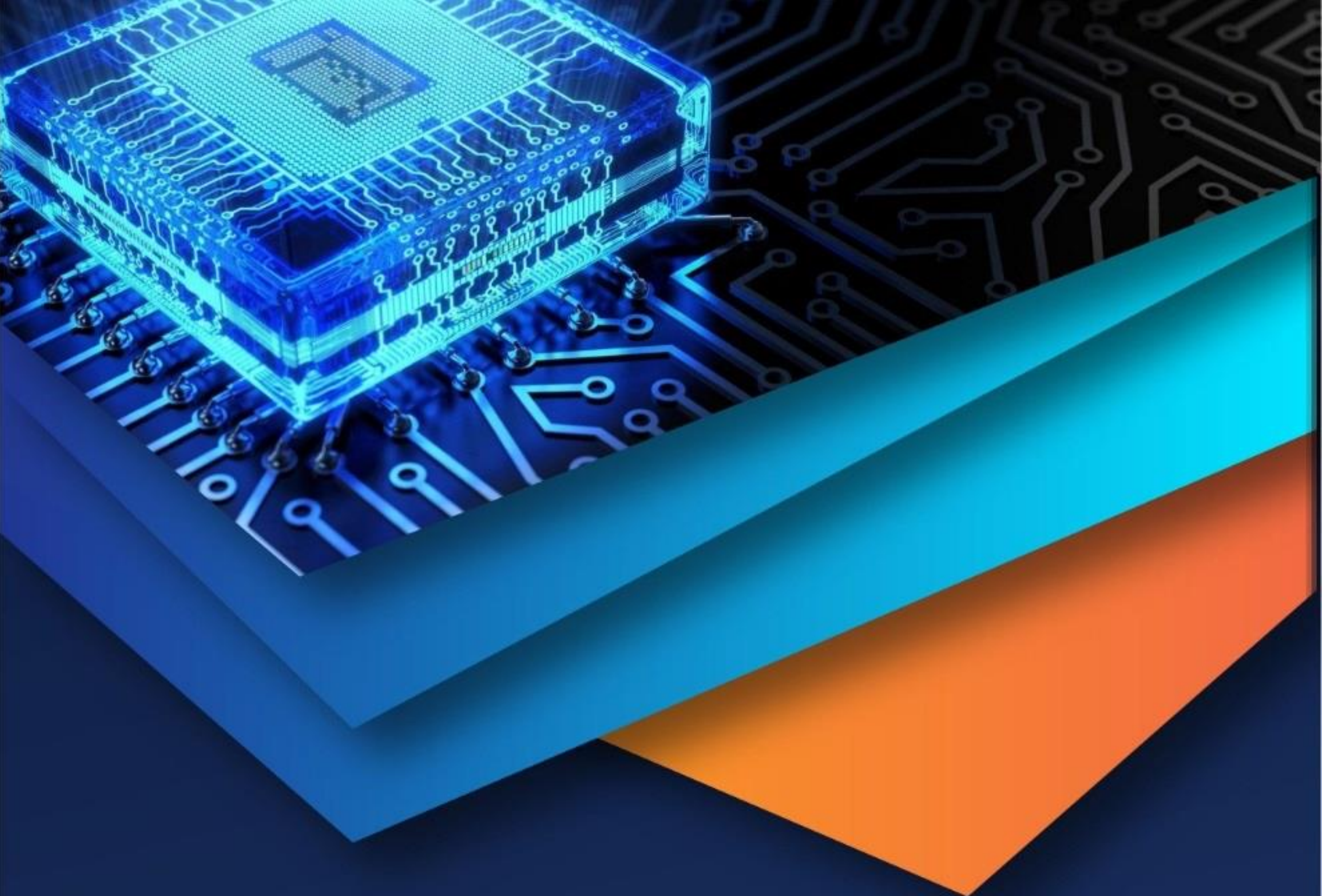
[17] H. Wang, F. Zhang, X. Xie, and M. Guo, "Deep learning based recommender system: A survey and new perspectives," ACM Computing Surveys, vol. 52, no. 1, 2019.

[18] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, no. 8, 2013.

[19] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in Proc. UAI, 2009.

[20] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in Proc. EMNLP, 2014.

[21] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in Proc. IEEE ASRU, 2013.
[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. ACM SIGKDD, 2016.
[23] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, 2016.
[24] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Pearson, 2010.
[25] L. Zhang, X. Zhao, and Z. Wang, "Personalized recommendation algorithm based on knowledge graph," IEEE Access, vol. 8, 2020.
[26] J. Merino-Campos, "Artificial Intelligence in Personalized Learning: A Review," 2025.
[27] C. Merino-Campos, "The Impact of Artificial Intelligence on Personalized Learning in Higher Education," Trends in Higher Education, vol. 4, no. 2, 2025.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY