

Eduadapt:

Anai-drivenadaptivelearningplatform for Personalized Academic Support

D.Abhinaya¹,B.Sindhuja²,G.padmpravallika³,P.Avila⁴

Under the Guidance of S.Chandra Shekher, Assistant Professor, Department of AI&DS

Vignan'sInstituteofManagementandTechnologyforWomen,Hyderabad, Telangana,India

ABSTRACT

Eduadapt Is An Artificially Intelligent, Learner-Centric Educational System Conceived To Resolve The Shortcomings Inherent In Conventional E-Learning Environments That Deliverrigid, Non-Adaptive Content Uniformly To All Users. The Platform Concentrates On Academically Demanding Subjects, Particularly Formal Languages And Automata Theory (Flat), Design And Analysis Of Algorithms (Daa), And Quantitative Aptitude. Although Widely-Used Platforms Such As Coursera, Udemy, And Byju's Offer Structured Learning Materials, Theyremain Largely Static And Devoid Of Meaningful Personalization. Independently Operating Ai Utilities Such As Chatgpt, Scholarcy, And Quillbot Similarly Fall Short By Functioning Outside The Boundaries Of Structured Academic Curricula. Eduadapt Remedies These Deficiencies By Embedding Large Language Models (Llms) And Natural Language Processing (Nlp) Within A Cohesive,Curriculum-Integrated Educational Ecosystem. Its Feature Set Encompasses On-Demand Personalizedtutoring,Dynamicvisualrepresentationsofcomputationalabstractionsincludingdfaandnfstate Diagrams, Ai-Assisted Summarization Of Study Materials, And An Analytics-Driven Assessment Dashboard That Monitors Performance And Highlights Knowledge Gaps. By Unifying These Capabilities Within A Single Platform Aligned With Academic Syllabi, Eduadapt Fosters Improved Conceptual Retention, Accelerates Revision Workflows, And Delivers Tailored Academic Guidance. Rigorous Testing Across Functional, Performance, Boundary, And Adversarial Scenarios Validates The System's Operational Reliability And Accuracy, Underscoring Its Capacity To Measurably Enhance Student Learning Outcomes.

Keywords: Adaptive Learning, Artificial Intelligence, Large Language Models (llms), Natural Language Processing (NLP), E-Learning, Personalized Education, Automata Visualization, Performance Analytics, Intelligent Tutoring System.

I. INTRODUCTION

The accelerating evolution of digital learning technologies has fundamentally altered the modes through which students seek, access, and internalize academic knowledge.Despitethisprogress,amajority of prevailing e-learning solutions continue to operate under a one-dimensional content delivery model, presenting the same instructional material to every learner without accounting for differences in comprehension pace, skill level, or cognitive preference. Prominent platforms including Coursera, Udemy, and Byju's deliver well-structured academic content, yet they remain constrained by a static pedagogical framework that offers little responsiveness to individual learner needs. Concurrently, AI-driven tools such as chatgpt, Scholarcy, and quillbot, while effective within their respectivenichesofconversationaltutoringand

Document summarization, operate as isolated utilities lacking any formal integration with academic assessment ecosystems or structured curricula.

To overcome these systemic limitations, eduadapt has been architected as an intelligent, AI-augmented educationalplatformthatharnessesthepoweroflarge Language Models (llms) and Natural Language Processing (NLP) within an integrated academic environment. Operating as an autonomous intelligent tutor, the platform is capable of formulating personalized concept explanations, resolving learner queriesinrealtime,generatingcondensedsummaries of academic content, and rendering dynamic visual representations for technically abstract subjects such as Formal Languages and Automata Theory (FLAT), Design and Analysis of Algorithms (DAA), and Quantitative Aptitude.

The imperative for such a platform emerges from the escalatingcomplexityofmodernacademicdisciplines combined with an increasingly diverse student population exhibiting varied learning characteristics. Conventionalinstructionalmodelsanddigitallearning toolsfrequentlydemandsubstantialmanual

Pedagogical effort inherently lacks the capability for real-time responsiveness. Eduadapt addresses this void by offering a scalable, intuitive, and technically robust ecosystem that consolidates AI-powered tutoring, conceptual visualization, automated summarization, and longitudinal performance analytics into one unified system. Concurrent breakthroughs in artificial intelligence and machine learning have further enabled the construction of advanced models capable of real-time, semantically accurate content generation.

Eduadapt additionally incorporates a dedicated assessment and analytics dashboard that maintains records of learner performance trajectories, surfaces areas of conceptual weakness, and delivers intelligent, adaptive feedback continuously over time. The confluence of tutoring, visualization, summarization, and analytics within a single coherent framework transforms eduadapt into a holistic and curriculum-aligned learning environment, one that demonstrably elevates academic comprehension and contributes to improved overall scholastic outcomes.

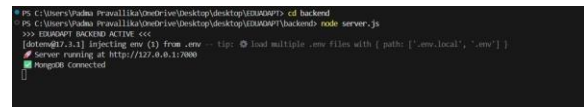
II. MATERIAL AND METHOD

A. System overview and architecture

Eduadapt is constructed upon a full-stack web development paradigm employing javascript (Node.js) on the server side, HTML and CSS for the client-side interface, and mongodb as the primary document-oriented data store. Integration with the openai API empowers the system's core AI Tutor module to dynamically generate personalised instructional content. The overall system architecture is organized into six distinct yet interoperable functional units: User Management, Learning Material Management, ai tutor, articles summarizer, Quiz and Assessment, and Performance Analytics.

B. Technology stack

The server-side infrastructure is built on Node.js utilizing the Express.js framework to expose all restful API endpoints governing user authentication, AI tutor communication, quiz orchestration, and student progress monitoring. Mongodb, interfaced through the Mongoose Object Document Mapper (ODM), manages the persistent, flexible storage of student profiles, academic content, assessment questions, evaluation outcomes, and usage analytics. The client-side interface employs HTML, CSS, and vanilla javascript, relying on the native Browser Fetch API for non-blocking asynchronous server communication. Development and debugging activities were performed within the Visual Studio Code integrated development environment.



```
PS C:\Users\paula\Downloads> cd backend
PS C:\Users\paula\Downloads\backend> code server.js
>>> npm start
[dotenv@2.3.3] injecting env ({}): tip: load multiple .env files with { path: ['.env.local', '.env'] }
Server running at http://127.0.0.1:7000
MongoDB Connected
```

Fig.1. backend server initialization and database Connection

C. Ai tutor module

The AI Tutor module channels the openai API to furnish context-aware, curriculum-specific academic explanations in real-time. Upon a student's submission of a query through the frontend, the system formulates a structured instructional prompt and dispatches it via an http post request to the express.js backend. The backend relays this prompt to the openai api, which returns a richly detailed response encompassing conceptual definitions, illustrative examples, and distilled learning objectives. This explanation is concurrently archived in mongodb for subsequent analytics processing and transmitted back to the frontend interface for immediate display. The architecture guarantees that every learner interaction is captured and feeds into the evolving performance analysis pipeline.

Fig.2. eduadapt ai tutor interaction interface



D. Articles summarizer module

The articles summarizer module empowers learners to submit voluminous academic texts or study notes for automated condensation. Upon receiving the submitted content, the system processes it through NLP-driven text analysis algorithms to produce semantically dense summaries that isolate the most critical concepts, thereby substantially reducing the cognitive burden associated with revision activities. The module communicates with the backend API endpoint, which applies natural language processing routines to distil and restructure the input into an accessible, condensed format.

E. Quiz and assessment module

The Quiz and Assessment module governs the complete lifecycle of automated quiz delivery and evaluation. When a learner selects a subject topic, the system first queries the database for an existing question set; in its absence, a fresh quiz is dynamically synthesized. Questions are presented to the learner in a sequential manner with interim responses buffered until all items are completed. The scoring engine subsequently evaluates the collected answers against

The verified solution set stored in the database, computing a final performance score. All quiz outcomes are persisted within the Progress collection to support downstream analytics and adaptive feedback generation.



Fig.3.eduadaptpstudent performanceanalyticsinterface

F. Performanceanalyticsmodule

The Performance Analytics module ingests quiz evaluation data to systematically identify gaps in student understanding. It sustains longitudinal academic progress records, compiles granular performance reports, and generates targeted improvement directives tailored to individual learner profiles. The interactive dashboard leverages Chart.js line chart visualizations to render performance trends overtime, supporting both students and administrators in tracking academic development. The embedded predictive analytics engine applies pattern recognition algorithms to historical quiz performance data, surfacing recurring conceptual deficiencies that may otherwise go unnoticed.

G. Flatvisualizationmodule

The visualization module is purpose-built to facilitate learner comprehension of computationally abstract concepts inherent in formal languages and automata Theory. A backend NFA-to-DFA conversion engine executes the full subset construction algorithm to derive an equivalent deterministic finite automaton from any given non-deterministic specification. The conversion pipeline produces intermediate construction steps, composite DFA state mappings, complete transition tables, start state annotations, and final accept state designations, all of which are transmitted to the frontend for graphical animation. This progressive, step-by-step visual rendering substantially demystifies abstract automata theory concepts for students encountering them for the first time.

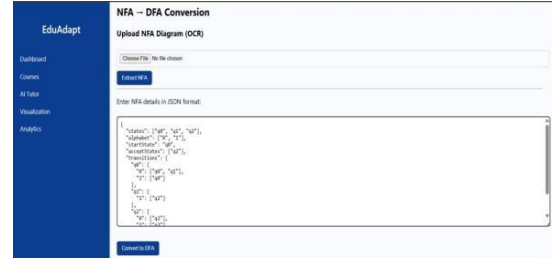


Fig.4.eduadaptnfatodfavisualizationconversion Interface

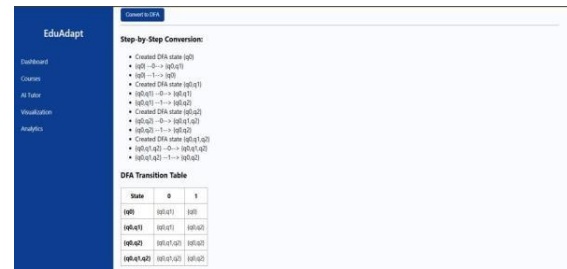


Fig.5.eduadaptpstep-by-stepnfatodfaconversion and Transition Table Output

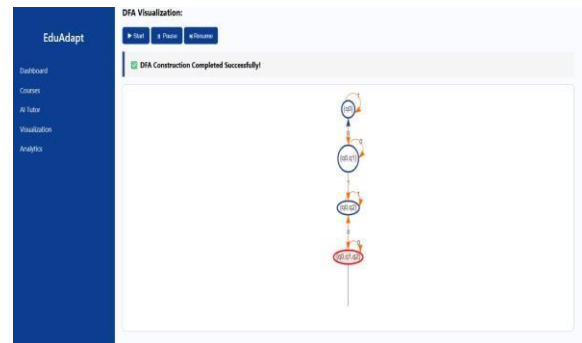


Fig.6.eduadaptdfavisualizationgeneratedfromnfa Conversion

H. Hardwareandsoftwarerequirements

Eduadapt is a resource-efficient, browser-based platform imposing minimal infrastructure demands. Operationally, the system requires an Intel Core i3 processor or equivalent, a minimum of 4 GB RAM with 8 GB recommended for optimal responsiveness, a reliable internet connection for API interactions, and at least 500 MB of available local storage. On the software front, the platform necessitates a Node.js runtime environment, a running mongodb server instance, and any contemporary web browser. The development environment was configured in Visual Studio Code supplemented with javascript, Node.js, and mongodb extensions.

III. LITERATURE REVIEW

An exhaustive survey of prevailing digital learning platforms and AI-assisted educational tools uncovers several persistent deficiencies that collectively



Motivate the design of educational systems. The analysis below examines key representative systems and systematically catalogues their respective shortcomings.

Mainstream e-learning services including Coursera, Udemy, and Byju's deliver curated academic content through a combination of lecture videos, reading materials, and assessment quizzes. Despite the quality of their offerings, these services adhere to a uniform, non-differentiating pedagogical model wherein every learner receives identical instructional material irrespective of their individual pace or demonstrated comprehension. Personalization is minimal at best, and learners are largely left to independently discern their own knowledge deficiencies without any intelligent guidance or adaptive course correction [1].

Conversational AI platforms like chatGPT furnish on-demand academic explanations and real-time doubt resolution through contextually aware dialogue. These tools prove valuable for targeted topic clarification; however, they function as standalone utilities disconnected from any formal academic structure. Their lack of integration with assessment mechanisms, progress monitoring systems, and curriculum-aligned content renders them supplementary aids rather than self-contained, end-to-end educational solutions [2,3].

Dedicated text condensation services such as Scholarcy and Quillbot are engineered to distil lengthy documents into compact, digestible summaries, thereby streamlining revision processes. However, their design confines them to a single isolated function: summarization. These tools maintain no awareness of learner progress, are incapable of identifying conceptual weak areas, and offer no mechanism for aligning summarized content with broader pedagogical objectives, effectively making them point solutions rather than components of a dynamic learning ecosystem [4].

Standard learning management and assessment systems facilitate the delivery of online evaluations and knowledge tests. However, such platforms typically employ fixed-difficulty assessments that remain unresponsive to real-time shifts in learner performance. Score reporting is generally superficial, devoid of interpretive analytics, trend analysis, or actionable personalized recommendations, significantly limiting their utility for targeted academic improvement [5, 6].

Traditional classroom instruction and manual mentoring foster direct pedagogical interaction but inherently lack the capability to deliver real-time adaptive feedback or interactive visualizations for abstract subject matter. The unavailability of dynamic digital representations for computationally complex

domains such as automata theory constitutes a

Significant instructional barrier, preventing learners from building intuitive conceptual models [7].

The foregoing survey establishes that while individual existing tools provide partial utility, none offers a comprehensively integrated solution that merges AI tutoring, interactive computational visualization, automated content summarization, and data-driven performance analytics within a single curriculum-aligned platform. Eduadapt is intentionally designed to occupy this unaddressed space by synthesizing all these dimensions into one cohesive adaptive learning system.

Fig.7.systemarchitecture

B. Uml design

IV. PROPOSED SYSTEM ARCHITECTURE

The Eduadapt platform is architecturally grounded in a modular, multi-tier design philosophy that cleanly partitions frontend user interaction, server-side business logic, AI service integration, and persistent data management into discrete, independently maintainable layers. This separation of concerns promotes system scalability, facilitates targeted maintenance, and enables the efficient handling of simultaneous user sessions.

A. System architecture

The overall system is structured across three architectural tiers: the Client Layer, the Application Server Layer, and the Data Persistence Layer. The Client Layer is realised using HTML, CSS, and javascript, delivering the complete user interface inclusive of authentication flows, course navigation, AI tutor querying, note summarization, quiz participation, and performance visualization screens. The Application Server Layer is constructed on Node.js with the Express.js framework and encapsulates all core system logic, including RESTful routing, authentication management, AI prompt engineering, quiz evaluation logic, and the NFA-to-DFA conversion algorithm. The Data Persistence Layer uses mongodb paired with Mongoose ODM schemas to store and retrieve structured records for students, academic modules, quiz repositories, results archives, and progress logs.



The architectural design of *eduardapt* is formally captured through a comprehensive suite of UML modelling artefacts. The use case diagram delineates the interaction boundaries between two primary actors — the Student and the System Administrator — and the *eduardapt* system, specifying use cases including System Login, Note Upload and Summarization, AI tutor query submission, quiz attempt, Progress and Analytics Review, FLAT Automata Visualization, User Account Administration, Learning Material Upload, and Quiz Monitoring. The Class Diagram articulates the object-oriented structural blueprint of the system, defining a User base class extended by Admin and Student subclasses, each associated with domain entities including learning material, Quiz, performance report, llm tutor module, visualization module, and summarizer module.

The Sequence Diagram traces the chronological message flow for the AI tutoring interaction: the student issues a query via the frontend, which constructs and forwards an http post request to the backend; the backend assembles a structured LLM prompt and transmits it to the openai api; the model returns a comprehensive explanation; the backend persists the interaction record in mongodb and returns an HTTP 200 OK response with the explanation to the client for rendering. The Activity Diagram charts the complete quiz participation workflow, beginning from user login and quiz retrieval or dynamic generation, progressing through sequential question presentation, answer collection, submission, and automated evaluation, culminating in score computation, weak area identification, and report generation.

C. Module architecture

Eduadapt is decomposed into six cohesive functional modules. The User Management Module governs the registration and authentication of students and administrators, enforces bcrypt-based password hashing, manages JWT session tokens, and implements role-differentiated access control. The Learning Material Management Module provides administrators with tools to upload, classify, and curate academic resources including course notes and quiz question banks.

The *llm tutor module* maintains a live integration with the openai API to supply instantaneous, curriculum-referenced conceptual responses to learner queries. The Article Summarizer Module accepts submitted academic content and returns NLP-generated condensed summaries. The Quiz and Assessment Module handles question set retrieval or real-time generation, orchestrates the sequential delivery of questions, records learner responses, and computes

evaluation scores. Finally, the Performance Analytics Module aggregates assessment outcomes, identifies

Conceptual weak zones, generates longitudinal progress reports, and delivers individualized remediation recommendations.

V. EXPERIMENTAL SETUP AND RESULTS

A. Testing methodology

A multi-dimensional testing protocol was employed to comprehensively evaluate and adapt, encompassing functional verification, performance benchmarking, boundary and edge case analysis, and adversarial negative testing. This holistic evaluation strategy was purposefully designed to confirm module-level behavioural correctness, gauge system stability across varied operational load profiles, verify graceful handling of extreme inputs, and validate robust failure management under erroneous or malicious conditions.

B. Functional testing

Functional testing was conducted to verify conformance of each module to its formal specification. All principal modules, encompassing user authentication, tutor interaction, article

All functional modules demonstrated full compliance with their specified requirements, and the end-to-end learning workflow executed successfully throughout the testing phase.

C. Performance testing

Performance testing assessed the platform's responsiveness and stability across sustained and concurrent usage scenarios. The tutor module

Test case id	Module	Expected output	Status
TC-01	User management	Account created successfully	Pass
TC-02	User management	Authentication successful	Pass
TC-03	User management	Access denied with error prompt	Pass
TC-04	Tutor	Detailed academic response delivered	Pass
TC-05	Tutor	Validation warning displayed	Pass
TC-06	Summarizer	Brief, accurate summary produced	Pass
TC-07	Quiz module	Quiz rendered on screen	Pass
TC-08	Quiz module	Final score computed correctly	Pass
TC-09	Performance analytics	Deficient areas highlighted	Pass
TC-10	Progress tracking	Data persisted in database	Pass

Summarization, quiz evaluation, and academic progress monitoring, were subjected to both isolated unit testing and integrated end-to-end workflow testing with the help of Render. The outcomes of key functional test cases are tabulated below.

Consistently generated responses within acceptable latency threshold even under simultaneous multi-user query loads. Backend database operations involving concurrent quiz submissions executed without measurable degradation. System behaviour remained stable across multi-session parallel access, and the Article Summarizer handled extended academic text inputs without performance degradation or errors. All performance benchmarks were satisfied, affirming reliable system operation under realistic deployment conditions.

D. Boundary and negative testing

Boundary testing established that the platform handles extreme input conditions gracefully. Null input to the summarizer and quiz modules triggered appropriate client-side validation alerts; abnormally lengthy tutor queries were processed without system failure; quiz submissions absent of any selected answers produced descriptive validation errors; and maximum-length input fields were accepted within established constraint limits. Negative testing confirmed correct error response behaviour for malformed API requests, unauthorised access attempts to protected dashboard views, transient database connectivity interruptions, and submission of structurally corrupted quiz data. Across all adversarial scenarios, the system responded with informative error messages and sustained operational continuity without service interruption.

E. System interface results

End-to-end empirical validation of the deployed eduadapt system confirmed successful and accurate operation across all primary functional domains. Server initialization proceeded without errors, and stable MongoDB connectivity was established and maintained throughout testing. The Login and Registration interfaces correctly processed authentication and new user enrolment, with bcrypt-based password encryption verified. The Student Dashboard accurately surfaced learning progress metrics and analytics visualizations. The Course and Module selection interfaces retrieved and rendered the correct curriculum-aligned content. The Quiz Result and Attempt History screens faithfully recorded and presented performance data.

The Performance Analytics interface accurately identified conceptual weak areas from quiz data and rendered longitudinal progress charts via `chart.js`. The FLAT Visualization engine executed NFA-to-DFA conversions correctly, producing stepwise transition tables and graphical automata diagrams. The AI tutor delivered contextually precise, curriculum-aligned explanations across all test queries spanning FLAT, DAA, and quantitative aptitude subject domains. The

Management operations including content upload, modification, and monitoring with full functionality.

F. Discussion

The empirical validation outcomes confirm that EduAdapt effectively resolves the principal deficiencies identified in prevailing e-learning platforms. The convergence of AI tutoring, document summarization, computational visualization, and performance analytics within a single curriculum-aligned system yields a substantially more comprehensive and adaptive learning experience than any available isolated tool. The LLM-driven AI tutor proved capable of producing accurate, contextually nuanced, and pedagogically appropriate academic explanations, closely emulating the responsiveness of a human subject-matter expert.

The NFA-to-DFA visualization module directly addressed a well-documented instructional challenge in the teaching of Formal Languages and Automata Theory, where abstract computational constructs have historically resisted effective conveyance through static textual presentations alone. The interactive, step-by-step graphical rendering of automata

transformation processes furnishes a dynamic learning medium that conventional static e-learning platforms fundamentally cannot offer.

The performance analytics dashboard enables evidence-based pinpointing of individual conceptual deficiencies, facilitating precise and targeted academic intervention. This analytical capability — conspicuously absent from most existing digital learning solutions — fundamentally reframes assessment from a terminal summative exercise into an ongoing, formative, and adaptive pedagogical mechanism. Certain limitations were, however, identified. The system's current adaptive logic operates primarily on post-quiz data rather than enabling real-time within-session adaptations. Furthermore, assessment difficulty levels remain static and do not dynamically escalate in response to evolving learner performance during an active session. These identified constraints delineate concrete directions for future development.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

The EduAdapt Adaptive Learning Platform has been successfully designed, developed, and

validated to address the fundamental inadequacies of both traditional and contemporary e-learning environments. Through the cohesive integration of artificial intelligence, automated performance assessment, NLP-based summarization, interactive visualization, and multi-dimensional analytics within a unified platform, EduAdapt realizes a personalized,



Engaging, and academically rigorous learning experience.

The coordinated operation of the AI Tutor, Article Summarizer, Quiz and Assessment, and Performance Analytics modules ensures that learners benefit from immediate concept clarification, concise revision materials, structured knowledge evaluation, and data-driven academic feedback. The openai API integration underpins intelligent content generation capabilities, while mongodb delivers the scalable and efficient data persistence required to manage user profiles, academic content, assessment records, and progress data. Comprehensive testing across functional, performance, boundary, and adversarial scenarios validated that all system modules operate accurately and reliably under both normal and extreme conditions.

In totality, eduadapt represents a meaningful contribution to the advancement of digital academic education by offering a centralized, intelligent, and scalable learning environment. The platform establishes a solid architectural foundation for future feature development and demonstrates the practical feasibility of deploying LLM-integrated adaptive learning systems within authentic academic contexts.

B. Futurework

Several developmental enhancements are earmarked for forthcoming iterations of the platform. The foremost planned enhancement involves the engineering of a fully adaptive difficulty calibration algorithm that modifies the complexity of assessment questions dynamically and continuously based on real-time learner performance signals, thus delivering a genuinely personalized evaluative experience.

Expanded visualization functionality is envisioned for future releases, incorporating interactive DFA and NFA diagram construction interfaces, real-time automata simulation environments, and visualization support extended to encompass algorithm paradigms from the Design and Analysis of Algorithms curriculum. Security hardening measures, including jwt authentication reinforcement, refined role-based access policies, and advanced password protection mechanisms, are also planned.

Cloud-based deployment via platforms such as Amazon Web Services or containerized orchestration using Docker is targeted to enhance system availability, horizontal scalability, and concurrent multiuser capacity. Gamification elements such as achievement recognition badges, competitive leaderboards, and milestone-based progress rewards are under consideration to boost learner motivation and sustained engagement. Prospective versions may also introduce multilingual interface support, voice-

driven query interaction, and sophisticated personalized learning pathway recommendations



Derived from machine learning analysis of longitudinal academic performance histories.

ACKNOWLEDGEMENT

The authors extend their deepest appreciation to Mr. S. Chandra Shekhar, Assistant Professor, Department of Artificial Intelligence and Data Science, Vignans Institute of Management and Technology for women, for his invaluable mentorship, sustained encouragement, and constructive guidance throughout every phase of this project. The authors also wish to acknowledge Dr. D. Shanthi, Professor and Head of Department, and Dr. A. Appa Rao Naidu, Professor and Principal, VMTW, for their institutional support and for making available the resources essential to the conduct of this research. Heartfelt gratitude is additionally extended to the authors' families and peer network for their unwavering moral support throughout this endeavour.

AUTHOR INFORMATION

D. Abhinaya, B. Sindhuja, G. Padma Pravalika, P. Avila — Department of Artificial Intelligence and Data Science, Vignans Institute of Management and Technology for women (an autonomous institution, affiliated to JNTUH, Hyderabad), Ghatkesar, Kondapur, Telangana 501301, India. Under the guidance of S. Chandra Shekhar, Assistant Professor, Department of AI&DS.

REFERENCES

- [1] Coursera Inc., "Coursera: Online Courses and Certifications," 2025. [Online]. Available: <https://www.coursera.org>
- [2] T. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (neurips)*, vol. 33, pp. 1877–1901, 2020.
- [3] Openai, "GPT-4 Technical Report," openai, 2023. [Online]. Available: <https://openai.com>
- [4] Scholarcy Ltd., "Scholarcy: AI-based Academic Summarization Tool," 2025.
- [5] 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.
- [6] G. Siemens, "Learning Analytics: The Emergence of a Discipline," *American Behavioral Scientist*, vol. 57, no. 10, pp. 1380–1400, 2013.
- [7] M. Woolf, *Building Intelligent Interactive Tutors:*

Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann, 2008.

[8] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.



- [10] MongoDB Inc., "mongodb Documentation," 2025. [Online]. Available: <https://www.mongodb.com/docs>
- [11] Node.js Foundation, "Node.js Documentation," 2025. [Online]. Available: <https://nodejs.org>
- [12] Express.js, "Express—Fast,unopinionated,minimalist web framework for Node.js," 2025. [Online]. Available: <https://expressjs.com>
- [13] Mongoose, "Mongoose ODM Documentation," 2025. [Online]. Available: <https://mongoosejs.com>
- [14] Udemy Inc., "Udemy: Online Learning Platform," 2025. [Online]. Available: <https://www.udemy.com>
- [15] Think and Learn Pvt. Ltd., "Byju's — The Learning App," 2025.
- [16] Quillbot Inc., "quillbot: AI Writing and Paraphrasing Tool," 2025.
- [17] Shanthi, Dr. D., G. Ashok, Chitrika Biswal, Sangem Udharika, Sri Varshini, and Gopireddi Sindhu. 2025. "Ai-Driven Adaptive It Training: A Personalized Learning Framework For Enhanced Knowledge Retention And Engagement". Metallurgical and Materials Engineering, May, 136-45. <https://metall-mater-eng.com/index.php/home/article/view/1567>
- [18] Shanthi, D., Aryan, S. R., Harshitha, K., & Malgireddy, S. (2023, December). Smart Helmet. In International Conference on Advances in Computational Intelligence (pp. 1-17). Cham: Springer Nature Switzerland
- [19] Shanthi, D., G. Narsimha, and R.K. Mohanthy. 2015. Human Intelligence vs. Artificial Intelligence. International Journal of Electronics Communication and Computer Engineering 6 (5): 30–34.
- [20] P. Endla, A. R, S. Suneel, A. P. Singh, P. A and D. Shanthi, "MedSensePathway: A Hybrid Framework for Real-Time Diagnosis of Malarial Parasites using Medical Imaging," 2025 9th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2025, pp. 1972-1978, doi:10.1109/ICECA66444.2025.11382939.