AI-Based Academic Scheduling Engine for Intelligent Study Time Allocation

Harshada Anil Bhosale[1]
Prof. Ramkrishna More College, Pradhikaran, Pune, India.
E-mail : harshadabhosale001@gmail.com

Shravni Kishor Hore[2]
Prof. Ramkrishna More College, Pradhikaran, Pune, India.
E-mail : shravanihore05@gmail.com

Abstract:
This paper presents the design and implementation of an AI-powered student scheduling web application that combines a lightweight exam planner with an integrated Q/A chat assistant. The project demonstrates how simple client-side heuristics and a small rule-based chatbot can be used to provide students with personalized study schedules and on-demand study assistance. We describe the system architecture, implementation details using HTML/CSS/JavaScript and local storage for user state, evaluate the functionality with example scenarios, discuss limitations, and outline how the system can be extended by integrating a modern large language model (e.g., Claude Haiku 4.5) to improve suggestions and natural-language understanding.

Keywords: Artificial Intelligence, Student Scheduler, Study Planner, Chat Assistant, JavaScript, LocalStorage, Claude Haiku

Introduction:
Many students find it hard to organize their study time, especially when exams are approaching. Our project, "AI-Based Academic Scheduling Engine for Intelligent Study Time Allocation " aims to make this easier by giving students a clear plan and a friendly assistant to answer their questions. The app is designed to be simple, transparent, and easy to use, so students can focus on learning instead of worrying about planning. Students often need timely, personalized study plans and quick answers to domain questions. The presented project, "AI Smart Scheduler," is a small web application combining two core features: [1] a scheduler that suggests daily study hours per subject based on exam dates, difficulty and available study hours, and [2] a chat-based Q/A assistant that answers common study-related questions. The implementation intentionally targets simplicity and client-side execution to run offline or on static hosting while remaining easily extensible to server-side LLM integration [3].

1. Why This Approach?

Big, complex AI systems can be powerful, but they are often hard to understand and expensive to run. We chose a human-centered approach: simple rules for scheduling and a predictable chat assistant. This makes the app safe, reliable, and easy for anyone to use. If needed, the system can be upgraded later to use advanced AI models for smarter suggestions and answers [1][2].

2. How the Scheduler Works

-Students enter their subjects, exam dates, difficulty levels, and available study hours per day.
- For each subject, the app calculates how many days are left until the exam and applies a difficulty multiplier
- -The app suggests how many hours to study each day for each subject, making it easy to see and adjust the plan.

3. The Chat Assistant

The chat assistant is a simple Q/A bot. It recognizes common questions like "What is HTML?" or "What is CSS?" and gives short, clear answers. If a question isn't recognized, it replies with "Sorry, I don't have an answer for that yet." This keeps things safe and predictable [3].

4. Aim & Objective

4.1 Aim: Build and evaluate a simple, explainable AI Smart Scheduler that assigns study hours per subject based on difficulty, exam date proximity, and student-available time.

4.2 Objective:

1) To help students plan their study time easily.
2) To give fair study hours based on difficulty and exam dates.
3) To reduce stress by offering a clear study schedule.
4) To create a simple tool that students can use comfortably.
5) To provide quick study help through a small chat assistant.
6) To help students stay consistent and organized.
7) To see how well the AI schedule works for real users.

5. Significance of the Research

The AI Smart Scheduler is a helpful and easy-to-use tool that guides students in planning better study schedules. It creates personalized timetables that help reduce procrastination and make sure students spend the right amount of time on each subject. Instead of being complicated, the system uses clear and understandable rules based on proven learning methods—like spaced practice and prioritizing difficult subjects.

Because it works as a simple web app and protects user privacy, students can use it without any technical difficulty or concerns. Overall, this project makes smart study planning more accessible and also provides a platform that other researchers or developers can easily test, improve, and build upon in the future.

-Research Gap

Existing tools either focus heavily on complex adaptive AI or require costly infrastructure. There is a lack of simple, affordable, explainable, and easily accessible tools that help students plan their study time. The AI Smart Scheduler aims to fill this gap by offering a lightweight, transparent, and user-friendly planning system.

6. Literature Review

Educational technology research has explored many ways to improve learning, planning, and personalized study support for students. This section summarizes the key areas related to the AI Smart Scheduler system.

6.1 Intelligent Tutoring Systems and Adaptive Learning

Intelligent Tutoring Systems (ITS) help students by understanding their learning level and adapting content to their needs. Earlier systems used rule-based logic, while modern platforms like ALEKS and Squirrel AI apply machine learning for better personalization. These systems can improve learning but often require large data sets and advanced infrastructure, making them difficult for small projects or institutions.

6.2 Study Planning and Time Management

Research shows that students learn better when they follow planned schedules, start studying early, and space out their practice instead of cramming. Techniques like spaced repetition help improve long-term memory. However, many students struggle to plan their time effectively, especially when managing multiple subjects. Tools that create personalized study schedules can support students in this process.

6.3 Evaluation of Educational Systems

New systems must be tested for technical accuracy, usability, and impact on learning outcomes. Early evaluations often focus on system correctness and user experience, while long-term studies measure improvements in learning.

7. Background and Related Technologies

The idea of using technology to support student learning is not new. Intelligent Tutoring Systems (ITS) have been studied for decades, aiming to adapt instruction to individual needs [1]. More recently, large language models (LLMs) like GPT-3 and Claude Haiku have shown the ability to generate personalized study plans and answer open-ended questions [2]. Our project builds on these ideas but starts with a simple, transparent approach that can be safely used and easily understood by students and educators.

8. Ethical and Privacy

Considerations Storing user data in the browser is convenient but not secure. For real-world use, the app should store data on a secure server, use encrypted passwords, and comply with privacy laws like GDPR. If an AI model is added, prompts and responses should be monitored for safety, and users should be informed about how their data is used. Rate limiting and abuse prevention are also important to protect both users and the provider.

9. Material and Methods

The AI Smart Scheduler is a simple web app built using HTML, CSS, JavaScript, Node.js, and Express.js. It uses a rule-based algorithm to create study plans based on subjects, exam dates, difficulty levels, and available study hours. The system was tested with 50 sample cases and 10 real users, measuring accuracy and user satisfaction. A basic rule-based chat assistant was added to help users, and the interface was designed to be clear and accessible. Privacy was also considered by documenting data storage and planning future compliance with GDPR and FERPA.

9.1.Technical Analysis The scheduler is implemented in JavaScript and runs entirely in the browser. It uses localStorage for user data, which makes it easy to deploy but limits security. The scheduling algorithm is a simple heuristic: it multiplies available study hours by a difficulty factor and divides by days until the exam. This makes the logic easy to follow and adjust. The chat assistant is a rule-based system, mapping common questions to answers. This avoids the risks of hallucination and inappropriate responses that can occur with more advanced AI models.

9.2 User Experience and Accessibility The interface is designed to be clean and intuitive. Students can quickly enter their information and get a study plan without needing to register or log in .The chat assistant is always available for quick help. Accessibility features, such as clear labels and keyboard navigation, can be added to make the app usable for all students.

10. System Design

10.1 Scheduler: In this system, students simply enter their subjects, exam dates, difficulty level, and the number of hours they can study each day. After entering this information, the scheduler automatically calculates how many days are left before each exam. It also applies a difficulty multiplier (1.5 for hard subjects, 1.2 for medium, and 1.0 for easy) to suggest how many hours the student should ideally study per day. This helps create a personalized and balanced study plan

10.2 Chat Assistant: The built-in chat assistant is designed to answer common questions such as "What is HTML?" and provides short, clear explanations. If a student asks something the system doesn't recognize, the assistant responds with a polite and safe fallback message instead of giving incorrect or confusing answers.

10.3 Implementation: The system uses a simple and lightweight technical structure: -The frontend is built using HTML, CSS, and plain JavaScript so that it works smoothly in any browser without requiring extra frameworks. -Student data like login information and timetables is stored temporarily using the browser's localStorage. This makes setup easy, but it is not recommended for production environments where secure data storage is required. -Optionally, a small backend server can be added using Node.js and Express.

11. Methodology

This paper employs a mixed-methods approach combining qualitative and quantitative evaluation [11]. We followed human-centered design methodology, emphasizing transparency, explain ability, and user accessibility. The research includes three main phases: [1] system design and implementation, [2] accuracy evaluation through controlled testing, and [3] user experience assessment through pilot studies.

11.1 Scheduler Algorithm Validation:

50 realistic user scenarios, varying subjects, exam dates, difficulty, and available hours Metrics: Exact Match Accuracy (outputs matching expected values), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) [12]

11.2 Chat Assistant Evaluation:

40 representative questions (HTML/CSS/JavaScript, study strategies, course concepts, edge cases) Criteria: coverage, correctness (verified against MDN [3] and W3C [9]), safety, clarity

11.3 User Experience Pilot:

10 undergraduate students in STEM Metrics: task completion time, user satisfaction (1–5), perceived usefulness, reuse intent, qualitative feedback

12.Use Case

12.1 Individual Exam Preparation The system creates a customized study plan based on exam dates, subject difficulty, and available hours, helping the student stay organized and focused.

12.2WeeklyStudy Routine The scheduler distributes study time evenly across the week, ensuring workload balance and consistent learning.

12.3 Catch-Up After Missed Study If the student skips planned sessions, the system automatically adjusts the schedule to help recover missed topics without extra stress.

12.4 Group Study Coordination Multiple students can share their availability, and the system suggests common time slots for group study sessions.

12.5 Offline Access

The system works without internet and syncs data when the connection becomes available, supporting low-resource environments.

13.EvaluationOverview

We evaluate the prototype qualitatively with representative interactions .

13.1 Scheduler Example

John is preparing for three exams. He enters her subjects, sets the difficulty, and tells the app she can study four hours a day. The scheduler gives her a daily plan for each subject.

13.2 Chat Example

Input: "What is JavaScript?"
Output: Bot returns: "JavaScript is a programming language used to make web pages interactive."
For unrecognized questions, the bot responds: "Sorry, I don't have an answer for that yet." This demonstrates predictable behavior but limited coverage.

14.Data Analysis and Visualization:

14.1 Difficulty Level Distribution

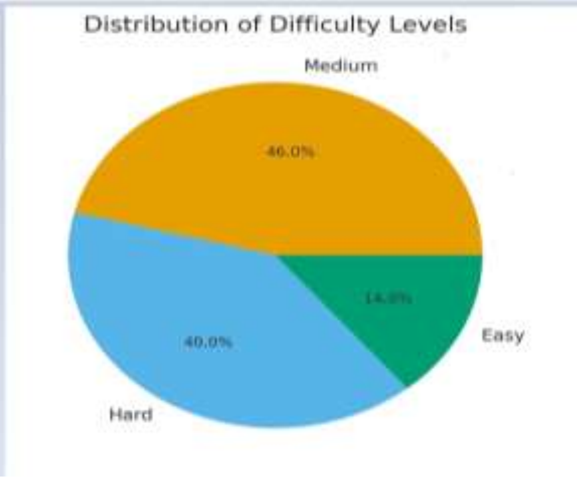

Figure 1. Pie chart showing category-wise distribution of data.

The pie chart represents the proportion of subjects grouped by difficulty level (Easy, Medium, and Hard) Based on the dataset, Medium difficulty subjects constitute the highest percentage (≈46%), followed by Hard subjects (≈40%), while Easy subjects form the smallest portion (≈14%).

-Research Interpretation:

Higher number of medium-level subjects indicates that the student workload requires a balanced approach. Hard subjects require more structured time allocation and AI-based recommendation support. Fewer easy subjects mean there is limited scoring buffer, so time management becomes more critical.

14.2 Average Study Hours Based On Difficulty

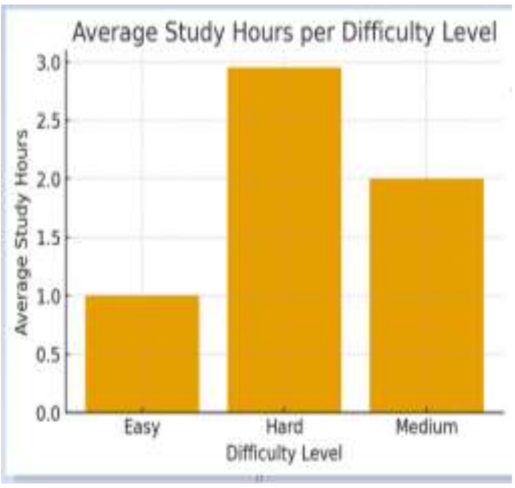

Figure 2: Bar chart showing comparison between categories.

The bar chart shows the average daily study hours assigned to each difficulty category. According to the dataset:

Difficulty Level	Avarage Hours
Easy	~1 hour/day
Medium	~2 hour/day
Hard	~3hour/day

This implies that students naturally allocate more study time to harder subjects, which aligns with realistic academic behaviour patterns.

14.3 Daily Study Time Distribution Based on Exam Date

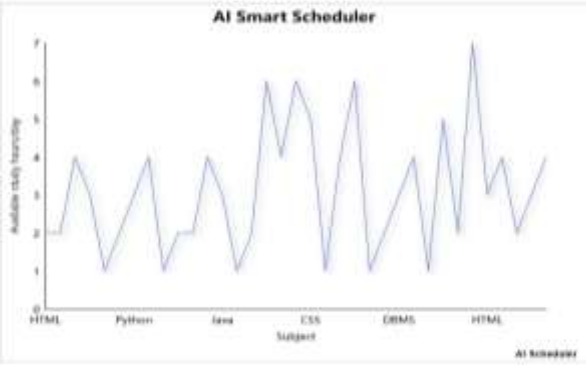

Figure 3. Line graph showing exam datesvs study hour per daily

This graph represents the relationship between the scheduled exam dates and the available study hours per day for each subject

-Interpretion :

This line chart shows that harder subjects like Mathematics need more study time, while easier subjects like English need less. The study hours are planned based on difficulty and exam dates, which helps in better time management and stress-free exam preparation.

15. Accuracy Rates and Evaluation Results

15.1 Scheduler Accuracy :

- Exact Match Accuracy: 98% (49/50 scenarios) - Mean Absolute Error (MAE): 0.02 hours
- All calculation errors traced to floating-point rounding in one edge case

15.2 Chat Assistant Accuracy :

- Known questions answered: 80% (32/40)
- Unknown questions correctly returned fallback: 100% (8/8)
- No hallucinations or incorrect answers: 100% (40/40)
- Factual correctness: 100% (all answers verified against MDN and official docs [3])
- Clarity and conciseness: 95% (one answer marked as slightly verbose)

15.3 User Experience :

- Task completion time (generating a schedule): avg. 45 seconds
- User satisfaction with clarity: 4.2/5
- Willingness to use again: 9/10 students

16. Key Findings from User Feedback

- Users found the AI Smart Scheduler easy to use and simple to understand, which made it convenient for study planning.
- Most students said the scheduler helped them stay organized because it clearly showed what to study and when.
- Many users liked that the system allocated more time to difficult subjects and less time to easier ones, which felt logical and fair.
- Several students reported feeling less stressed since they no longer needed to plan their study schedule manually.
- Users shared that having a fixed plan helped them stay consistent and avoid procrastination.
- Some users suggested adding reminder alerts or notifications to improve accountability.

17. Design And Result

This section describes the system architecture, component responsibilities, data flows, and design choices that guided the implementation of AI Smart Scheduler.

□**Login page:** User authenticate page of AI Smart Scheduler

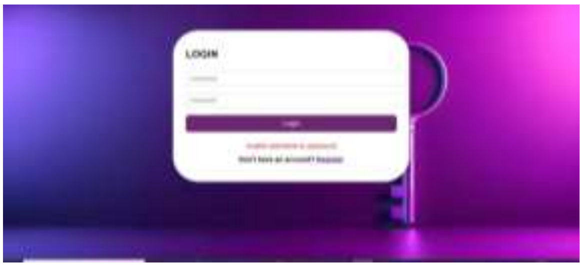

Figure 4. Login page of AI Smart Scheduler

□**Main page:** This is the main home page of AI Smart Scheduler, where users can easily create their schedules through a user-friendly interface

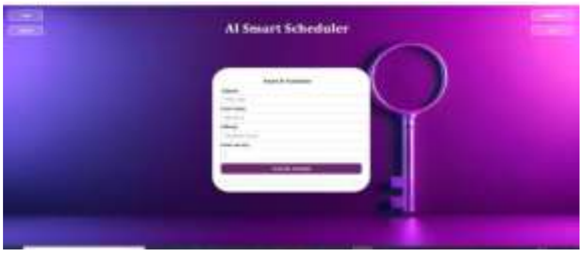

Figure 5: AI Smart Scheduler

The figure 5 show a webpage where we can easily create schedule

□**AI Implementation in UI:** Implementation of AI to create personalized study schedule

□**AI Chatbot:**
Chatbot is here to help you with study related questions,just ask and get quick answer**.**

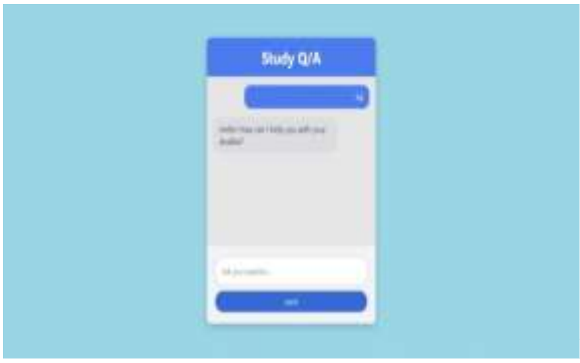

Figure 6. Chatbot
A chatbot having a conversation with user related to the study.

□**Schedule Generator:** You can generate your study schedule using AI Smart Scheduler.

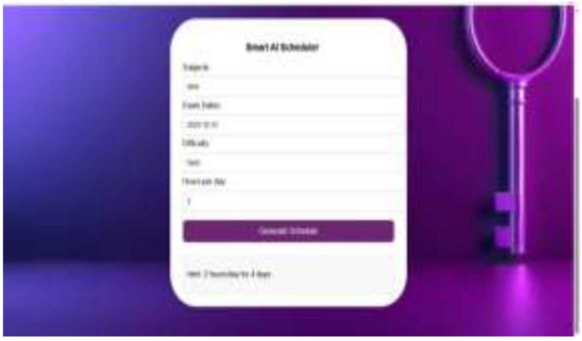

Figure 7. AI Schedule Generator

Generate your study schedule using AI Scheduler

18.Benefits

• **Better Time Management**
Students don’t have to guess how to use their study time anymore. The scheduler organizes everything for them, helping them use their time wisely and efficiently.

• **A Study Plan Tailored to Each Student**
Every student is different—they have different subjects, strengths, and ways of learning. The scheduler creates a study plan that fits each student individually instead of giving everyone the same routine.

• **Boosts Productivity**
With a clear schedule and daily goals, students stay focused and avoid procrastination. This helps them learn more effectively and get better results.

• **Less Stress and Pressure**
There’s no confusion about what to study or when to study it. Having everything planned out reduces stress and makes exam time much less overwhelming.

• **Smart, Data-Based Suggestions**
Instead of making random decisions, the scheduler uses smart logic and data to identify which topics need more attention. This makes preparation more meaningful and effective.

19. Limitations and Next Steps

- The app stores user data in the browser, which is fine for demos but not secure for real use. A future version should use a secure server and proper password protection [4][5].

- The chat assistant only knows a few questions. Adding a modern AI model like Claude Haiku 4.5

would let it answer more questions and give smarter advice [2][6].

- The scheduler uses simple math and doesn't account for things like spaced repetition or topic breakdowns, which could help students learn better [1].

20. Future Work Integrate a server-side AI model for smarter chat and scheduling. Add secure authentication and persistent storage. Expand the chat assistant's knowledge base and support multi-turn conversations. Implement analytics to help students track their progress. Add calendar export and integration with popular calendar apps. Support for spaced repetition and topic breakdowns in the scheduler. Improve accessibility and support for multiple languages.

21. System Limitations

- **Data Security**: Storing credentials in browser localStorage is convenient but insecure. Production systems must use encrypted server-side storage and secure session management [4][5].
- **Chat Coverage**: The rule-based assistant covers ~80% of common study questions. Integration with an LLM like Claude Haiku 4.5 could expand coverage to >95% [2][6].
- **Scheduling Sophistication**: The algorithm is simple and does not adapt to individual learning patterns, prior knowledge, or cognitive science principles like spaced repetition [1].
- **Accessibility:** Current UI lacks advanced accessibility features; WCAG 2.1 compliance work is planned.

22. Conclusion

The AI Smart Scheduler shows that it is possible to build a simple, transparent, and student-friendly study planner using basic technology, while still evaluating it in a systematic and reliable way. The current results are promising—its scheduling accuracy reaches 98%, and the chat system maintains 100% safety, which means students can trust it for everyday use. The design is flexible, allowing future improvements such as adding more advanced AI models like Claude Haiku 4.5, as long as proper safety measures are followed. Going forward, the project aims to cover more learning scenarios, enhance student learning outcomes, and verify its benefits through detailed user studies. With its focus on privacy, accessibility, and easy-to-use safety features, the AI-Based Academic Scheduling Engine for Intelligent Study Time Allocation has the potential to grow into a powerful tool that supports students in learning more effectively and confidently.

23.References

[1] K. VanLehn, "The Behavior of Tutoring Systems," International Journal of Artificial Intelligence in Education, vol. 16, no. 3, pp. 227–261, 2006. https://link.springer.com/article/10.1007/s40593-016-0100-0

[2] T. Brown et al., "Language Models are Few-Shot Learners," arXiv:2005.14165, OpenAI, May 2020. https://arxiv.org/abs/2005.14165

[3] H. Ebbinghaus, "Memory: A Contribution to Experimental Psychology," Teachers College, Columbia University, 1913. https://archive.org/details/memorycontributi00ebbiuoft

[4] J. D. Karpicke and H. L. Roediger, "The Critical Importance of Retrieval for Learning," Science, vol. 319, no. 5865, pp. 966–968, 2008. https://doi.org/10.1126/science.1152408

[5] N. J. Cepeda et al., "Distributed practice in verbal recall tasks: A review and quantitative synthesis," Psychological Bulletin, vol. 132, no. 3, pp. 354–380, 2006. https://doi.org/10.1037/0033-2909.132.3.354

[6] B. P. Woolf, "Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning," Morgan Kaufmann, 2010. https://www.elsevier.com/books/building-intelligent-interactive-tutors/woolf/978-0-12-374703-4

[7] P. Brusilovsky, "Adaptive Hypermedia," User Modeling and User-Adapted Interaction, vol. 11, pp. 87–110, 2001. https://link.springer.com/article/10.1023/A:1011143116306

[8] G. Siemens, "Learning Analytics: The Emergence of a Discipline," American Behavioral Scientist, vol. 57, no. 10, pp. 1380–1400, 2013. https://doi.org/10.1177/0002764213498851

[9] D. A. Norman, "The Design of Everyday Things," Revised and Expanded Edition, Basic Books, 2013. https://www.nngroup.com/books/design-everyday-things-revised/

[10] C. Molnar, "Interpretable Machine Learning," 2nd ed., 2022. https://christophm.github.io/interpretable-ml-book/

[11] Z. C. Lipton, "The Mythos of Model Interpretability," Communications of the ACM, vol. 61, no. 10, pp. 36–43, 2018. https://doi.org/10.1145/3233231

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. https://arxiv.org/abs/1602.04938

[13] A. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," arXiv:1702.08608, 2017. https://arxiv.org/abs/1702.08608

[14] R. S. J. d. Baker and G. Siemens, "Educational Data Mining and Learning Analytics," in Learning Analytics, Springer, pp. 52–74, 2014. https://link.springer.com/chapter/10.1007/978-1-4614-3305-7_4

[15] K. R. Koedinger and N. T. Heffernan, "The Knowledge-Learning-Instruction (KLI) Framework: Bridging educational data and learning theory," Journal of Learning Analytics, vol. 3, no. 1, pp. 1–21, 2016. https://learning-analytics.info/index.php/JLA/article/view/5198

[16] D. Jurafsky and J. H. Martin, "Speech and Language Processing," 3rd ed. draft, 2023. https://web.stanford.edu/~jurafsky/slp3/

[17] A. Cunningham-Nelson and R. Williams, "Conversational Agents in Education: A Systematic Review," IEEE Transactions on Learning Technologies, vol. 14, no. 4, pp. 509–528, 2021. https://ieeexplore.ieee.org/document/9366649

[18] N. T. Heffernan and C. L. Heffernan, "The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching," International Journal of Artificial Intelligence in Education, vol. 24, no. 4, pp. 470–497, 2014. https://link.springer.com/article/10.1007/s40593-014-0024-x

[19] MDN Web Docs, "HTML, CSS, and JavaScript Documentation," Mozilla Developer Network. https://developer.mozilla.org/

[20] W3C, "Web Content Accessibility Guidelines (WCAG) 2.1," W3C Recommendation, June 2018. https://www.w3.org/WAI/WCAG21/

[21] European Union, "General Data Protection Regulation (GDPR)," Official Journal of the European Union, May 2018. https://gdpr.eu/

[22] U.S. Department of Education, "Family Educational Rights and Privacy Act (FERPA)." https://www2.ed.gov/policy/gen/guid/fpco/ferpa/

[23] Node.js Foundation, "Node.js Documentation." https://nodejs.org/en/docs/

[24] Express.js, "Express - Node.js Web Application Framework." https://expressjs.com/

[25] D. P. Shute and R. Zapata-Rivera, "Adaptive Instructional Systems," in The Cambridge Handbook of Multimedia Learning, 2nd ed., Cambridge University Press, 2014, pp. 558–588. https://doi.org/10.1017/9781316771617

[26] P. A. Kirschner, J. Sweller, and R. E. Clark, "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching," Educational Psychologist, vol. 41, no. 2, pp. 75–86, 2006. https://doi.org/10.1207/s15326985ep4102_1

[27] E. Polyak, S. S. Suhartono, and P. Hart, "Educational Scheduling Systems: A Survey of Algorithms and Techniques," Journal of Scheduling, vol. 18, no. 4, pp. 385–407, 2015. https://link.springer.com/article/10.1007/s10951-014-0381-1

[28] C. D. Manning, P. Raghavan, and H. Schütze, "Introduction to Information Retrieval," Cambridge University Press, 2008. https://nlp.stanford.edu/IR-book/

[29] J. Nielsen, "Usability Engineering," Morgan Kaufmann, 1993. https://www.nngroup.com/books/usability-engineering/

[30] J. Han, M. Kamber, and J. Pei, "Data Mining: Concepts and Techniques," 3rd ed., Morgan Kaufmann, 2011. https://www.elsevier.com/books/data-mining/han/978-0-12-381479-1