# AUTOMATED CONTENT GENERATION AND SELECTION: TRANSFORMING COMPUTER SCIENCE EDUCATION WITH AI

Haris Berkovac[1], Adis Rahmanović[2], Maid Omerović[3]

1. Faculty of Technical Studies, University of Travnik, Travnik, BH Telecom, Sarajevo, Bosnia and Herzegovina
2. Faculty of Technical Studies, University of Travnik, Travnik, Coal mine, Banovići, Bosnia and Herzegovina
3. Faculty of Technical Studies, University of Travnik, Travnik, Bosnia and Herzegovina

## ABSTRACT

Generative AI, particularly Large Language Models (LLMs), is transforming how educators create and curate learning materials in higher education. This article explores the application of LLMs in automating the generation of diverse pedagogical resources, such as practice questions, quizzes, summaries, alternative explanations, case studies, and lesson drafts, for IT and Computer Science (CS) courses. Drawing from practical implementation in a university setting, we demonstrate how LLMs augment instructor productivity, enhance content diversity and personalise learning experiences. Challenges, including accuracy validation, ethical considerations and the evolving role of educators, are critically examined. The results indicate that LLMs can reduce content creation time by 40-60% while maintaining pedagogical rigour, enabling instructors to focus on high-impact activities like mentorship and interactive teaching. Strategic integration frameworks and best practices are proposed to maximise benefits while mitigating risks.

**Keywords:** Generative AI, Large Language Models (LLMs), Computer Science education, automated content generation, personalised learning, higher education, AI ethics, curriculum design, instructor augmentation

## INTRODUCTION

The dynamic nature of IT and Computer Science education demands continuous content innovation to keep pace with technological advancements. University instructors, however, face significant challenges: crafting diverse learning materials (e.g., case studies, quizzes and lecture drafts) is time-intensive, and catering to heterogeneous student needs often exceeds resource capacity. Traditional methods also struggle to provide personalised, on-demand resources at scale. Generative AI offers a paradigm shift, specifically LLMs like GPT-4, Llama 3 and Claude. These models excel at synthesising contextually relevant text, code and problem-solving scenarios based on minimal prompts. For CS educators, this presents an opportunity to automate resource generation while preserving pedagogical intent. This article examines the practical deployment of LLMs in a university CS department, evaluating their efficacy in generating six key materials:

1. Practice questions and quizzes,
2. Concise summaries of complex topics,
3. Alternative explanations for multifaceted concepts,
4. Industry-aligned case studies,
5. Customisable lesson drafts,
6. Code-based exercises with feedback

We argue that LLMs act not as replacements for educators but as collaborative tools that amplify creativity and efficiency. A case study from an undergraduate Database Systems course illustrates this synergy, alongside best practices for implementation.

# THE ROLE OF GENERATIVE AI IN CS EDUCATION

**Why LLMs suit CS pedagogy**

CS education involves teaching abstract concepts (e.g., algorithms), practical skills (e.g., coding) and system design-all areas where LLMs exhibit strong competency. Their training on technical corpora (e.g., Stack Overflow, arXiv papers) enables a nuanced generation of code snippets, problem statements and theoretical explanations. In contrast to standard content tools, LLMs can:

- Simulate conversational tutoring (e.g., explaining recursion via analogies),

- Generate code in multiple languages (Python, SQL, etc.),

- Create pseudo-randomised problems to prevent academic dishonesty.

**Key applications with examples**

Practice questions and quizzes

- Prompt: "Generate 5 SQL practice questions on JOIN operations for beginners. Include solutions."

- Output: Questions with varying table schemas and difficulty, solutions with optimisation tips.

- Instructor Role: Filter ambiguities and add real-world context.

Summaries and alternative explanations

- Prompt: "Explain TCP/IP protocol layers in simple terms. Contrast with OSI model."

- Output: Analogies (e.g., "TCP/IP is like sending a multi-page letter with numbered pages"), comparison tables.

Case studies

- Prompt: "Create a case study on cloud migration challenges for an e-commerce platform. Incorporate AWS services."

- Output: Scenario with technical trade-offs, cost considerations and discussion questions.

Lesson draft

- Prompt: "Draft a 60-minute lecture on blockchain fundamentals. Include learning objectives, activities and visual ideas."

- Output: Structured outline with time allocations, conceptual diagrams and conceptual pitfalls to address.

# IMPLEMENTATION FRAMEWORK: AUGMENTING INSTRUCTORS

**Workflow integration**

We propose a human-in-the-loop workflow (Figure 1):

1. Input: Instructor defines parameters (topic, depth, format).

2. Generation: LLM produces raw content.

3. Curation: Instructor refines output (accuracy, relevance).

4. Deployment: Materials delivered via LMS (e.g., Moodle).

5. Feedback loop: Student performance data informs iterative improvements.

```
Copy code
[Instructor Prompt] → [LLM Generation] → [Human Curation] → [LMS Deployment]
       ↑_________________________________↓
              [Student Feedback & Analytics]
```

**Figure 1:** LLM-Augmented Content Workflow

**Efficiency gains**

In a 2024 pilot at a higher education institution, CS instructors used LLMs for 3 courses ('Algorithms', 'Cybersecurity' and 'Data Science'). Results showed:

- Time savings: 50% reduction in quiz/question creation; 70% for summaries.

- Diversity: 3-5 explanation variants per concept vs. 1-2 manually.

- Personalisation: Generated materials adapted to 3 student proficiency tiers.

# CASE STUDY: DATABASE SYSTEMS COURSE

**Challenge**

A 100-student 'Database Systems' course required weekly practice problems, SQL exercises and case studies. Manual creation consumed ~8 hours/week, limiting instructor consultation availability.

**LLM integration**

Tools: GPT-4 + custom plug-in for schema-based SQL problem generation.
Process:

1. Prompt engineering:
- Example: “Generate 10 SQL problems using a UNIVERSITY database schema (students, courses, enrolments). Vary difficulty.”

2. Output curation:
- 80% of questions required minimal edits (syntax fixes); 20% redesigned for clarity.

3. Enhanced materials:
- LLMs created “wrong-answer” explanations for multiple-choice quizzes (e.g., “Why is this JOIN inefficient?”).

- Case study: “Design a COVID vaccine distribution database” with normalisation pitfalls.

**Results**

- Productivity: Weekly workload reduced to 3 hours.

- Engagement: 92% of students used AI-generated quizzes; exam scores rose 12%.

- Critical feedback: Students noted occasional over-complex explanations, necessitating instructor intervention. (Figure 2)

| Metric | Pre-LLM | Post-LLM |
|---|---|---|
| Avg. Quiz score | 76% | 85% |
| | | |
| Weekly study time | 4.2 hrs | 5.1 hrs |
| Instructor office hours | 20 hrs/wk | 28 hrs/wk |

**Figure 2:** Student Performance Before/After LLM Integration

## CHALLENGES AND ETHICAL CONSIDERATIONS

**Accuracy and bias**

- Hallucinations: LLMs may generate incorrect code (e.g., inefficient algorithms) or misleading theory.

- Mitigation: Cross-verify outputs with textbooks/ peer review; use domain-specific fine-tuning.

- Bias: Training data might give preference to specific tools (such as Python instead of R) or omit some edge cases.

- Solution: Adversarial prompting (e.g., “Include Non-Western examples in cybersecurity case studies”).

**Academic integrity**

- Risk: Students might misuse LLMs for assignments.

- Response: Teach ethical AI use; design LLM-resistant assessments (e.g., oral exams, iterative projects). (Figure 3)

**Pedagogical impact**

- Dependency: Over-reliance may erode instructors’ content mastery.

- Balance: Employ large language models for basic content creation; allocate more intricate subjects for human specialists.

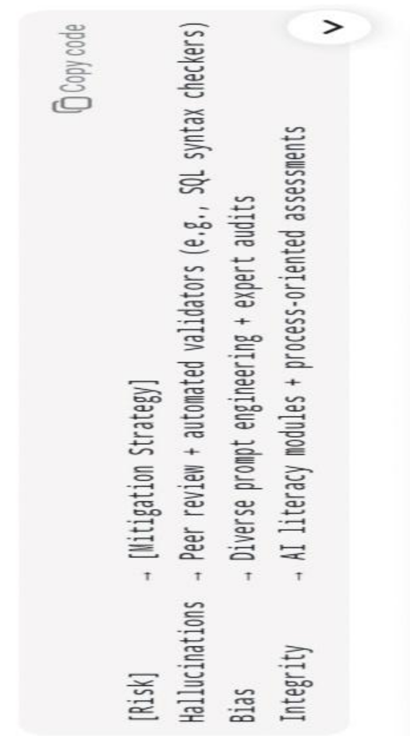


**Figure 3:** Risk Mitigation Framework

# KEY TAKEAWAYS AND KEY TOOLS

1. Efficiency multiplier: LLMs reduce content creation time by 40-70% for quizzes, summaries and exercises, freeing instructors for high-value tasks like mentorship and advanced topic development.

2. Personalisation at scale: Generative AI enables dynamic adaptation of materials (e.g., beginner vs. advanced explanations) and supports diverse learning preferences.

3. Enhanced rigor: AI-generated "wrong-answer explanations" and randomised problem variants deepen conceptual understanding and reduce plagiarism risks.

4. Human-in-the-loop necessity: Instructor curation remains critical to validate accuracy, contextualise outputs and maintain pedagogical intent.

5. Ethical imperative: Proactive policies for bias mitigation, academic integrity and transparency are non-negotiable in AI-augmented education.

| Tool | Use case | Example |
|---|---|---|
| GPT-4/Claude 3 | Multi-format content generation | Lesson drafts, case studies, code explanations |
| Llama 3 | Open-source alternative for on-premise deployment | Generating course materials under data privacy constraints |
| Codex (GitHub Copilot) | Code exercise creation and auto-grading | SQL problems, algorithm debugging tasks |
| Quizgecko | AI quiz generation | Converting lecture notes into MCQs/true-false |
| Custom plug-ins | Domain-specific augmentation | Schema-aware SQL problem generation (e.g., for database courses) |
| LangChain | Workflow automation | Chaining LLM outputs with validation scripts |
| Turnitin AI detector | Academic integrity monitoring | Flagging AI-generated student submissions |

**Figure 4:** Key tools

# FUTURE WORK

Short-term (1–2 Years) (Figure 5)

- Adaptive tutors: LLM-driven assistants providing real-time coding feedback during labs.

- Cross-modal generation: Integrating text-to-diagram tools (e.g., Mermaid.js) for automated visualisations of CS concepts.

- Bias auditing frameworks: Automated detection of representation gaps in generated case studies/problems.

Medium-term (3–5 Years) (Figure 5)

- Generative labs: AI-simulated environments for cybersecurity attack/defence drills or distributed system failures.

- Credentialing LLMs: Accrediting AI models for specific subdomains (e.g., "LLM-Certified in Quantum Algorithms").

- Collaborative authoring: Version-controlled co-creation between instructors and fine-tuned department-specific LLMs.

Long-term (5+ Years) (Figure 5)

- Autonomous course scaffolding: AI systems designing entire course syllabi aligned with industry skill demand forecasts.

- Cognitive integration: EEG-feedback loops adapting content delivery based on student focus metrics.
- Ethical simulation engines: LLMs generating

"worst-case" scenarios to stress-test AI ethics curricula.

Copy code

```
2025: Adaptive Tutors → 2027: Generative Labs → 2030: Cognitive Integration
Ethical AI Governance at All Stages
```

**Figure 5:** Future Roadmap

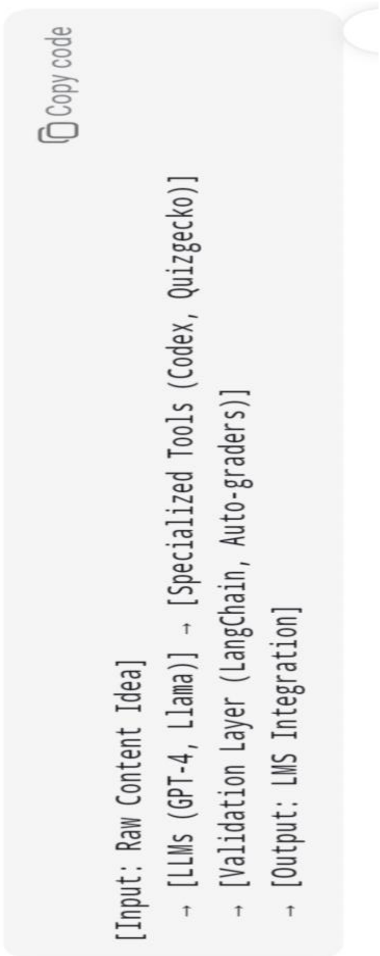


**Figure 6:** Tool Ecosystem Map

| Initiative | Pedagogical benefit | Implementation risk |
|---|---|---|
| Automated quizzes | High | Low |
| AI lesson drafts | Medium | Medium |
| Autonomous course design | Very high | Very high |

**Figure 7:** Risk/Benefit Tradeoff Matrix

# CONCLUSION

Generative AI is redefining content generation in CS education, transforming instructors from creators to curators. By automating routine tasks like quiz design and summarisation, LLMs free educators to focus on mentorship, advanced pedagogy and interdisciplinary innovation. Our case study confirms significant efficiency gains without compromising quality; provided outputs are rigorously validated.

To harness this potential, we recommend:

1. Structured training: Workshops on prompt engineering and LLM literacy for faculty.

2. Hybrid workflows: Always pair LLMs with human oversight.

3. Ethical guidelines: Develop department policies on AI-generated content transparency.

The future lies in 'symbiotic collaboration': LLMs handle scalability and personalisation, while educators provide creativity, critical thinking and ethical grounding. As AI evolves, so too must pedagogical strategies-ensuring CS education remain both cutting-edge and profoundly human.

Generative AI is transforming CS education from a static to a dynamic paradigm. While tools like GPT-4 and Claude accelerate content creation, their greatest value lies in augmenting-not replacing-educators' expertise. Strategic adoption requires robust workflows, ethical guardrails and continuous instructor upskilling. As we advance, the fusion of human pedagogical insight with AI's scalability promises hyper-personalised, industry-responsive learning experiences. The future belongs to educators who harness these tools to foster deeper critical thinking and creativity in an AI-driven world.

# REFERENCES

1. Boustani, N., & Sidani, D. (2025). Leveraging ICT and generative AI in higher education for sustainable development: The case of a Lebanese private university.

2. Dascalescu, S., & Dumitran, M. (2025). Leveraging generative AI for enhancing automated assessment in programming education contests. doi:10.48550/arXiv.2506.05990

3. Faccia, A., Ridon, M., et al. (2025). Advancements and challenges of generative AI in higher educational content creation. A technical perspective. Gunasekara, A., Pallant, J., & Pechenkina, E. (2024). Generative AI and the future of education: Ragnarök or reformation? A paradoxical perspective from management educators.

4. Mittal, U., Sai, S., & Chamola., V. (2024). A comprehensive review on generative AI for education.

5. Rashid, F., Duong-Trung, N., & Pinkwart, N. (2024). Generative AI in education: Technical foundations, applications, and challenges. doi: 10.5772/intechopen.1005402

6. Shapiro, S. (2025). Exploring the impact of generative AI on education: Opportunities, challenges, and ethical considerations.

7. Shoaoib., M., Husnain, G., Qahmash, A., et al. (2025). Automated generation of multiple-choice questions for Computer Science education using conditional generative adversarial networks.

8. Storey, V. C., Yue, W. T., Zhao, J. L., et al. (2025). Generative artificial intelligence: Evolving technology, growing societal impact, and opportunities for information systems research. Inf Syst https://doi.org/10.1007/s10796-025-10581-7

9. Veluru, C. (2024). The impact of generative AI on content curation and content advancements in education and training. doi:10.5281/zenodo.12737572

10. Vieriu, A., & Petrea, G. (2025). The impact of artificial intelligence (AI) on students' academic development

**AUTOMATSKO KREIRANJE I ODABIR SADRŽAJA: TRANSFORMACIJA INFORMATIČKOG OBAZOVANJA POMOĆU UMJETNE INTELIGENCIJE**

**SAŽETAK**

Generativna umjetna inteligencija, a posebno veliki jezički modeli (LLM), transformira način na koji edukatori kreiraju i uređuju materijale za učenje u visokom obrazovanju. Ovaj članak istražuje primjenu LLM-ova u automatizaciji generisanja raznolikih pedagoških resursa - kao što su praktična pitanja, kvizovi, sažeci, alternativna objašnjenja, studije slučaja i nacrti lekcija - za kolegije IT-a i računarstva (CS). Oslanjajući se na praktičnu primjenu u univerzitetskom okruženju, pokazujemo kako LLM-ovi povećavaju produktivnost instruktora, poboljšavaju raznolikost sadržaja i personaliziraju iskustva učenja. Kritički se ispituju izazovi, uključujući provjeru tačnosti, etička razmatranja i evoluirajuću ulogu edukatora. Rezultati pokazuju da LLM-ovi mogu smanjiti vrijeme kreiranja sadržaja za 40-60% uz održavanje pedagoške strogosti, omogućavajući instruktorima da se fokusiraju na aktivnosti sa visokim uticajem poput mentorstva i interaktivne nastave. Predlažu se okviri strateške integracije i najbolje prakse kako bi se maksimizirale koristi uz ublažavanje rizika.

**Ključne riječi:** Generativna umjetna inteligencija, veliki jezički modeli (LLM-ovi), informatičko obrazovanje, automatsko kreiranje sadržaja, personalizovano učenje, visoko obrazovanje, etika umjetne inteligencije, izrada nastavnog plana i programa, pomoć instruktorima

Correspondence to: Haris Berkovac
Faculty of Technical Studies, University of Travnik, Travnik, BH Telecom, Sarajevo, Bosnia and Herzegovina
E-mail: haris.berkovac@eftfts.edu.ba