NYU | TANDON SCHOOL OF ENGINEERING

Building Adaptive Learning: AI Tools for Educational Access in Computer Science Education

By Adonis Garcia

4/1/2026

Abstract

Quality education should not depend on where you are born or how much money your family has. Yet hundreds of millions of students worldwide lack access to the kind of personalized support that makes learning actually work. This paper examines two AI-powered tools I am developing to address different pieces of this problem. The first is a Computer Science learning platform that generates code examples tailored to what individual students care about. Instead of teaching everyone the same way, it adapts. A student interested in music gets examples about playlist manipulation. A student passionate about climate gets examples analyzing environmental data. The concept stays the same, but the entry point changes. The second tool is a question-and-answer system for NYU's Introduction to Engineering course that gives students instant, accurate answers based on actual course materials while helping instructors see where students struggle most. Both projects are in active development. I am currently conducting surveys, focus groups, and research to understand what students actually need as I continue building. This paper documents the technical design of both systems, examines what they could realistically do for educational access, and honestly addresses their limitations. Technology alone will not solve educational inequality. But tools designed thoughtfully, tested carefully, and deployed with equity in mind can expand access to quality learning for students who currently get left behind.

Personal Connection

I grew up in the South Bronx. My mom worked multiple jobs to keep me in school, and I got into NYU through HEOP, a program for students who would not otherwise have access to schools like this. I know what it feels like to be brilliant but overlooked, to have potential but lack the resources to reach it. Over three years in GLASS, my understanding of educational inequality evolved from abstract awareness to direct experience. Working as a teaching assistant, I watched students struggle not because they were not smart but because teaching methods did not adapt to them. I saw myself in those students. Mentoring first-generation students through the Academic Achievement Program, I witnessed how information barriers create compound disadvantages. Students did not know what questions to ask, who to talk to, or what resources existed. Studying abroad in Abu Dhabi twice, I observed how educational systems work differently across the world but face similar challenges. I met students from across the Middle East, Africa, and Asia whose paths to higher education looked nothing like mine but whose determination felt familiar. These experiences shaped how I am building these tools: with the students I have watched struggle in mind, with my own journey as context, designing for equity from the start rather than as an afterthought.

Table of Contents

I. Introduction

Fall 2023. I was in office hours when I noticed her staring at the same problem for the third time. She kept saying she did not get it, but I could tell she was capable. The issue was not her ability to understand loops. It was the textbook example. It did not connect to anything she cared about.

I asked what she was interested in. When I rewrote the same concept using a context that actually mattered to her, she got it immediately. Not in an hour. Not after multiple explanations. Immediately.

I have watched this happen dozens of times as a teaching assistant and mentor. Brilliant students stuck on concepts they are capable of understanding, not because the material is too hard, but because the way it is taught does not match how they think. The problem is not the students. It is that education does not adapt.

I grew up in the South Bronx. My mom worked multiple jobs to keep me in school, and I got into NYU through HEOP, a program for students who would not otherwise have access to schools like this. I have maintained a 4.0 while working as a TA, mentoring underrepresented students, studying abroad in Abu Dhabi, and building machine learning systems at internships. Every piece of that path showed me the same thing: access to quality education is not just about getting into a school or having textbooks. It is about having support that meets you where you are.

Educational inequality shows up everywhere once you start looking. Between countries where students lack pathways to higher education. Between neighborhoods where kids get different opportunities based on zip codes. Even within elite universities like NYU, where some students know how to navigate systems and others do not. Growing up, I watched brilliant students get left behind not because they could not learn, but because the system could not adapt to how they learned. Now I see the same patterns play out differently. Students struggle to find information about their courses, spend hours confused about concepts that could be explained differently, give up on CS because they do not see how it connects to their lives.

The gap extends globally. In developing countries, there is a massive shortage of qualified teachers who can provide individualized attention. Even in well-funded institutions, students from underrepresented backgrounds face information barriers that their peers navigate easily. The problem is not that students cannot learn. It is that education is not adapting to meet them where they are.

This paper examines two AI-powered educational tools I am developing to address different dimensions of educational access. The first is a personalized Computer Science learning platform that generates code examples and explanations tailored to individual student interests. Instead of teaching everyone loops the same way, it adapts. If you are into music, you get examples about playlist manipulation. If you care about social justice, you get examples analyzing protest data. The second is an intelligent Q&A system for NYU's Introduction to Engineering course that uses RAG (Retrieval-Augmented Generation) architecture to give students instant, accurate answers to their questions while showing instructors what concepts students struggle with most.

Both projects are in active development. I have the technical architecture planned and working prototypes in testing, but I am conducting user testing, gathering feedback, and iterating on the design based on what students actually need. That research process is teaching me as much as the technical implementation will.

This paper documents the technical design of both systems and examines what they could do for educational equity. Can AI meaningfully expand access to quality education? What are the limitations? How do you design these systems so they help rather than hurt? These are not just academic questions. They are questions I am asking myself as I build, as I watch students struggle, as I think about the education gap I came from and the one I am trying to help close.

The paper weaves together technical documentation with reflection on what I have learned about education, access, and the role technology can play over three years in the GLASS program. My thesis: AI can meaningfully expand educational access when designed with equity and context in mind, but only if we are honest about limitations and infrastructure requirements.

II. Background and Literature Review

The Scope of Educational Inequality

The statistics tell part of the story. According to UNESCO's 2024 Global Education Monitoring Report, 251 million children and youth remain out of school worldwide, a figure that has barely improved since 2015 (UNESCO, 2024). The disparities are stark: 33% of school-aged children in low-income countries are out of school compared to just 3% in high-income countries. More than half of all out-of-school children live in sub-Saharan Africa. Updated estimates suggest the true figure may be closer to 272 million when accounting for upper secondary students (UNESCO Institute for Statistics, 2025).

Even among those enrolled, learning outcomes vary dramatically. The World Bank's State of Global Learning Poverty report found that COVID-19 school closures sharply increased learning poverty, defined as children unable to read a simple text by age 10, from 57% pre-pandemic to an estimated 70% in low- and middle-income countries (World Bank et al., 2022). Nine in ten children in low-income countries are in learning poverty. The generation affected risks losing $21 trillion in potential lifetime earnings.

In the United States, students from low-income families face significant barriers in STEM fields. A landmark National Center for Education Statistics report found that 48% of bachelor's degree students who entered STEM fields left within six years, with students from low-income backgrounds leaving at higher rates (Chen, 2013). More recent data from the National Student Clearinghouse shows that only 8% of graduates from low-income high schools earned STEM degrees within six years, compared to 16% from higher-income schools (National Student Clearinghouse Research Center, 2019). The 2024 Pell Institute report documents that 79% of 18-to-24-year-olds from the highest income quartile enrolled in postsecondary education versus just 44% from the lowest quartile (Cahalan et al., 2024).

Teacher quality and availability matters enormously, particularly in under-resourced schools and developing countries. UNESCO's Global Report on Teachers found an urgent need for 44 million primary and secondary teachers worldwide by 2030, with sub-Saharan Africa alone needing 15 million new teachers (UNESCO, 2024). Low-income countries average 52 pupils per class compared to a global average of 26, and about 26% of primary and 39% of secondary teachers in these countries lack minimum qualifications (UNESCO, 2022).

Information access creates another barrier, particularly for first-generation college students. Rachel Gable's research on the "hidden curriculum" at elite universities reveals how unspoken academic norms and institutional expectations catch first-generation students off guard (Gable, 2021). A systematic review by Ives and Castillo-Montoya (2020) found that most literature positions first-generation students against normative learning approaches, while NCES data shows first-generation, low-income students were nearly four times more likely to leave higher education after the first year compared to their peers (Cataldi et al., 2018).

For underrepresented students in STEM specifically, the challenges compound. Research on stereotype threat, pioneered by Steele and Aronson (1995), has generated over 300 follow-up studies documenting how awareness of negative stereotypes can impair academic performance. McGee (2021) argues that what is often labeled "impostor syndrome" among Black STEM doctoral students frequently masks structural racism. These are not individual failings but systemic issues that technology alone cannot solve but might help address.

Current Educational Technology Landscape

EdTech has exploded over the past decade, but most solutions were not designed with underserved populations in mind from the start. Khan Academy provides free video lectures and practice problems, democratizing access to content. Research shows meaningful effects: a large-scale study of approximately 200,000 students found that increasing Khan Academy usage by 30 minutes per week was associated with 0.09-0.18 standard deviation gains on MAP Growth assessments (Eames et al., 2026). A randomized controlled trial of nearly 11,000 students found year-long mastery learning interventions improved math scores by 0.12-0.22 standard deviations (Oreopoulos et al., 2024). However, these platforms often replicate the same one-size-fits-all approach that fails students in traditional classrooms.

Intelligent tutoring systems represent more sophisticated attempts at personalization. Meta-analyses consistently demonstrate their effectiveness. Ma et al. (2014) analyzed 107 effect sizes and found ITS associated with greater achievement versus teacher-led instruction ($g = 0.42$)

and no significant difference from human tutoring. Kulik and Fletcher (2016) found the median effect raised test scores 0.66 standard deviations, equivalent to moving from the 50th to 75th percentile. Carnegie Learning's MATHia (formerly Cognitive Tutor) has been validated in a RAND Corporation randomized trial across 147 schools and 18,000+ students, showing positive effects of approximately 0.2 standard deviations and meeting ESSA Tier 1 "Strong" evidence standards (Pane et al., 2014).

The emergence of large language models like ChatGPT has changed what is possible. Students already use these tools to explain concepts, debug code, and generate examples. A systematic scoping review by Yan et al. (2024) identified 53 use cases for LLMs in education across nine categories, though highlighting low technological readiness and ethical concerns around transparency and privacy. However, general-purpose LLMs are not designed for education. They have no memory of what a student knows, cannot assess understanding, sometimes generate incorrect information, and were not built with pedagogical principles in mind.

The gap I see is this: most adaptive systems focus on adjusting difficulty or pacing. Few genuinely personalize the content itself to connect with individual student interests and contexts. Research on culturally relevant pedagogy demonstrates this matters. Scott et al. (2021) found that less than 60% of CS teachers surveyed felt equipped for culturally relevant pedagogical practices. Studies consistently show that students learn better when they see themselves reflected in examples and when content connects to their lived experiences (Leonard & Sentance, 2021; Morales-Chicas et al., 2019).

Technical and Theoretical Foundations

Large language models work by predicting the next word in a sequence based on patterns learned from massive text datasets. Models like Claude (which I am using) and GPT-4 have been fine-tuned on instruction-following tasks, making them better at responding to specific requests like "explain this concept" or "generate a code example." For education, the key capability is that these models can generate novel explanations and examples on demand. Unlike fixed educational content, an LLM can take a concept (loops in programming) and a context (climate data) and generate an appropriate example it was not explicitly trained on.

Recent research demonstrates the potential. A landmark Harvard randomized controlled trial found that students learned significantly more in less time with an AI tutor compared to active learning instruction, while also reporting feeling more engaged and motivated (Kestin et al., 2025). A systematic review of AI-driven intelligent tutoring systems in K-12 education found generally positive effects on learning, though effects were mitigated when compared to non-intelligent tutoring systems (npj Science of Learning, 2025).

RAG (Retrieval-Augmented Generation) architecture addresses a key LLM limitation: they do not have access to specific, up-to-date information like course materials. The foundational RAG paper by Lewis et al. (2020) demonstrated that combining parametric and non-parametric memory achieves state-of-the-art performance on open-domain question answering tasks. For a course Q&A system, this means: student asks question, system finds relevant lecture notes or textbook sections, LLM generates answer grounded in that specific material, includes citations so students can verify. Li et al. (2025) conducted the first comprehensive survey of RAG specifically in education, finding that RAG overcomes the primary LLM barrier in education: hallucinations.

From an educational theory perspective, this work builds on research into personalized learning and how people construct knowledge. Schema theory, originating with Bartlett (1932) and Piaget (1952), suggests that learning happens by connecting new information to existing mental frameworks. When examples use contexts students already understand (music, sports, social issues they care about), they provide better anchors for new concepts. Vygotsky's Zone of Proximal Development (1978) is directly applicable to AI-enhanced education, where AI systems can serve as the "more knowledgeable other" providing scaffolded support. Self-Determination Theory (Ryan & Deci, 2000) identifies autonomy, competence, and relatedness as the three innate psychological needs driving intrinsic motivation, all of which can be supported by well-designed adaptive learning tools.

The pedagogical principle underlying the CS platform is this: the goal is not to make learning easier but to make it more accessible. A student who sees a loop explained through basketball statistics is not learning something different than a student who sees it through poetry analysis. They are learning the same concept, just through an entry point that makes sense to them.

III. Technical Implementation: CS Learning Platform

System Architecture and Design Rationale

The CS learning platform aims to generate personalized code examples and explanations for core computer science concepts. The architecture reflects both technical requirements and pedagogical goals informed by the research on personalized learning effectiveness.

Frontend: React provides a clean, responsive interface. Students interact with a simple dashboard that profiles their interests through an initial survey (what topics do you care about? what hobbies do you have? what problems interest you?), then presents concepts they can explore. CodeMirror integration provides a syntax-highlighted code editor where examples appear and students can modify them. This interactive element is crucial: constructionist learning theory suggests that learning is most effective when learners construct tangible, shareable artifacts (Papert, 1980).

Backend: Node.js server handles routing and business logic. PostgreSQL database stores user profiles (interests, background), progress tracking (which concepts they have explored, which examples they found helpful), and system logs for analysis. The backend manages communication with Claude API (Anthropic's LLM), handling prompt construction, rate limiting, and response parsing.

The core functionality flow: (1) Student selects a CS concept (loops, functions, recursion, data structures, etc.). (2) System retrieves their interest profile from database. (3) Backend constructs a carefully engineered prompt that includes the concept to explain, the student's interests and background, and pedagogical requirements. (4) Claude API generates contextualized explanation and code example. (5) Frontend displays example with interactive editor. (6) Student can ask follow-up questions, request different examples, or mark examples as helpful. (7) System logs interactions for evaluation.

The platform currently covers 8-10 core CS concepts that form the foundation of introductory programming: variables and data types, conditionals, loops, functions, lists/arrays, dictionaries/objects, basic algorithms (searching, sorting), and file I/O. These concepts appear in virtually every introductory CS curriculum.

Prompt Engineering and Personalization

The quality of generated examples depends entirely on prompt engineering. A systematic survey by Sahoo et al. (2024) documents 39 distinct prompting methods, and research specifically on prompt engineering in education shows that techniques like chain-of-thought prompting can achieve near-perfect alignment with human evaluators (Cohn et al., 2024). I am iterating on prompts that reliably produce code that is both pedagogically sound and actually functional. Early versions generated code that compiled but taught bad practices (poor variable names, no error handling, unnecessary complexity). Current prompts include explicit pedagogical constraints.

Personalization happens at multiple levels. Surface level: using examples from domains students care about (music, sports, social justice, science). Deeper level: adjusting explanation style based on their background (first-time programmer vs. has some experience), incorporating their specific goals (building websites vs. analyzing data). This approach aligns with the four-phase model of interest development proposed by Hidi and Renninger (2006), which emphasizes the importance of triggered situational interest evolving into maintained engagement.

The interest profiling survey asks: What topics interest you? What are your hobbies? Why are you learning programming? What do you want to build? How would you describe your current skill level? This information gets stored and referenced when generating examples. A student interested in climate change and social justice might see loop examples that analyze carbon emission trends or demographic data. A student into music production might see the same concept through audio file processing or playlist manipulation.

Current Progress and Challenges

I have a functional prototype with core features implemented. Students can select concepts, receive personalized examples, interact with the code editor. The database structure exists. API integration works. What I am focused on now is making sure this actually helps students learn.

Current research phase includes: focus groups with 5-6 students to watch them use the prototype and gather feedback, surveys about what types of examples resonate and what feels forced or unhelpful, technical testing of the prompt engineering, and conversations with CS educators about pedagogical soundness.

Technical challenges I am working through include generating pedagogically sound code, balancing personalization depth with API costs, building meaningful assessment, managing latency, and making examples genuinely diverse. Claude is good at generating functional code, but "functional" is not the same as "good for learning." Early examples sometimes included advanced syntax or concepts students had not learned yet. I am refining prompts to constrain complexity while maintaining realism. Ben-Ari (2001) argues that in CS education, models must be explicitly taught and must precede abstractions, a principle I am incorporating into prompt design.

Design Decisions Informed by Research

The focus groups are revealing what works and what does not. Students respond positively to examples that connect to their interests, but the connection needs to be genuine. Forcing a contrived example creates more distance than a well-chosen generic example. Students can tell when personalization is superficial, echoing findings from culturally relevant pedagogy research (Leonard & Sentance, 2021).

Students value seeing multiple examples of the same concept. The first example introduces the idea. The second reinforces it. The third makes the pattern clear. Students want to modify examples and see what happens. The interactive code editor matters. Learning happens through experimentation, not just reading. Explanations matter as much as code. Students need to understand not just what the code does but why you would use this approach.

IV. Technical Implementation: Q&A System

System Architecture and Problem Context

The Q&A system addresses a different dimension of educational access: information barriers. Students in large introductory courses often do not know where to find answers to basic questions. Office hours get crowded. Course materials are scattered across multiple platforms. Students from first-generation or underrepresented backgrounds may feel uncomfortable asking questions they think they "should" already know. Research shows these students face compound disadvantages navigating the "hidden curriculum" of higher education (Gable, 2021; Stebleton & Soria, 2012).

The system aims to provide instant, accurate answers grounded in actual course materials while giving instructors insight into what students struggle with most. It is designed for NYU's Introduction to Engineering (EG 1004), a required first-year course with 200+ students per semester.

Architecture is more complex than the CS platform because it needs to handle multiple user roles and process large amounts of course material. Frontend: Next.js provides server-side rendering and role-based access control. Students see a clean question interface. Instructors see an admin dashboard with analytics showing common questions, accuracy metrics, and concept gaps. Backend: FastAPI (Python) handles requests quickly. It manages the document processing pipeline, interfaces with vector database, and constructs prompts for response generation. Vector Database: Pinecone stores embeddings of course materials. LLM: Claude API generates responses based on retrieved context, following best practices from RAG system design (Lewis et al., 2020; Gao et al., 2024).

Document Processing Pipeline

The system needs to answer questions based on course materials: lecture slides, textbook chapters, assignment descriptions, lab manuals, past exam questions. The challenge is converting these documents into a format the system can search semantically.

Pipeline steps include: (1) Document Ingestion where course materials get uploaded and parsed. (2) Chunking where large documents are broken into meaningful chunks that respect section boundaries. (3) Embedding Generation where each chunk gets converted to a vector embedding. (4) Vector Storage where embeddings get stored in Pinecone with metadata tags. (5) Updates when course materials change, requiring re-processing of affected documents. This pipeline follows established patterns from RAG system research (Gao et al., 2024).

Query Processing and Response Generation

When a student asks a question, the system performs: Query Embedding (the question gets embedded using the same model as documents), Hybrid Search (combining semantic similarity, keyword matching, and metadata filtering), Context Ranking (retrieving top 5-10 most relevant chunks), Prompt Construction (selected chunks become context for Claude), Response

Generation (Claude generates an answer grounded in course material with citations), Confidence Scoring (estimating confidence based on retrieval scores), and Feedback Loop (students can mark answers as helpful or not helpful).

Admin Dashboard and Analytics

The instructor-facing dashboard provides insights into student learning: Common Questions Analysis using HDBSCAN clustering on question embeddings to identify patterns, Concept Gap Identification by tracking which questions get low confidence scores, Usage Patterns showing when students ask questions and which topics generate most queries, and Accuracy Tracking by combining student feedback and manual instructor review.

Current Progress and Design Decisions

I have the architecture implemented and functional and am working on the document processing pipeline. Current focus includes pilot testing with EG 1004 course staff, evaluating chunking strategies, building instructor trust through safety measures, and cost analysis for scale deployment.

User research reveals key design requirements. Students want answers fast. A 10-second response time is too long when they are stuck. Students need to know when the system is uncertain. A wrong answer confidently delivered is worse than expressing uncertainty. Instructors worry about students over-relying on the system instead of engaging with material deeply. The system should reduce office hour load for basic questions while freeing instructors to focus on deeper conceptual discussions.

V. Evaluation Methodology

Both projects are in the middle of development, so I do not have results yet. But I am designing evaluation methodology now to ensure I collect the right data as development progresses. This approach follows best practices from educational technology research, which emphasizes mixed methods combining quantitative metrics with rich qualitative data (Rodriguez-Segura, 2022).

CS Learning Platform Evaluation Plan

Research Questions: Does personalization improve engagement compared to generic examples? Do students learn concepts better with contextualized examples? Which types of personalization work best? What are the limitations and failure modes?

Planned Methodology: Pilot Study (Spring 2026) recruiting 15-20 students from NYU's introductory CS courses. Students will use the platform for 3-4 weeks while learning core concepts. I will track engagement metrics (time spent, examples requested, return rate), learning assessment (pre/post conceptual tests, practice problem performance), qualitative data (interviews, observation sessions), and technical analysis (code correctness, pedagogical quality, cost per student).

Q&A System Evaluation Plan

Research Questions: Do students actually use the system? Are answers accurate and grounded in course material? Does it change student behavior? Does it help instructors identify learning gaps?

Planned Methodology: Deployment (Spring 2026) partnering with EG 1004 instructors to deploy system for all 200+ students. I will track usage metrics, accuracy assessment through instructor rating of sampled questions, impact analysis through surveys and office hour attendance patterns, and instructor perspective through interviews about dashboard usefulness.

Evaluation Challenges and Limitations

I need to be honest about what I can and cannot conclude from this research. Small Sample Sizes: With 15-20 pilot users for the CS platform, I will not have statistical power for strong quantitative claims. Selection Bias: Students who volunteer to test new tools are likely more motivated than average. Confounding Variables: Students use many learning resources simultaneously. Short Timeframe: A few weeks shows initial reactions but not long-term effectiveness. Resource Constraints: As a student building this, I have limited time and budget for comprehensive evaluation.

VI. Broader Impact and Scalability

The point of building these tools is not just to create two successful projects. It is to explore whether AI can meaningfully address educational inequality and what it would actually take to deploy these approaches in under-resourced contexts.

How These Tools Address Inequity

Scaling Personalized Support: In an ideal world, every student has a tutor who knows their interests, adapts explanations to their learning style, and has infinite patience. That world does not exist. With 44 million additional teachers needed globally by 2030 (UNESCO, 2024), AI does not replace human teachers, but it can scale personalized support in ways humans physically cannot. A student in a rural school with one overworked CS teacher could access the same adaptive examples as a student at an elite university. Meta-analyses of technology-facilitated personalized learning show medium effect sizes on achievement, with interventions that adapt to learners' level showing the highest effects (Major & Francis, 2021; Zheng et al., 2022).

Reducing Information Barriers: The Q&A system addresses asymmetry of knowledge navigation. Students who know how to ask questions, where to look, who to talk to have an advantage over those who do not. A system that makes information accessible levels that playing field. It is available 24/7, does not judge "stupid questions," and helps students who feel uncomfortable asking instructors.

Democratizing Access: Both tools are software, which means they scale differently than human resources. The marginal cost of one more student using the platform is minimal after initial development. In theory, these could be deployed globally at relatively low cost per student.

Cost Analysis and Practical Deployment

Theory is one thing. Practice is another. Development costs include my time, API costs during development (~$50-100), and database hosting (~$90/month total). Per-student operating costs are estimated at ~$2-5 per student per semester for the CS Platform and ~$1-3 per student per semester for the Q&A System. For a class of 200 students, the Q&A system would cost maybe $200-600 per semester. That is less than many textbooks.

Infrastructure requirements include reliable internet connection, device access, and payment infrastructure for API costs. Here is where deployment in under-resourced contexts gets complicated. The digital divide is real: research shows academic outcomes range from 19.6% for households with no Internet access to 68.0% for those with multiple access locations (Pierce et al., 2024). UNESCO reports that 2.2 billion individuals under 25 lack Internet access at home (UNESCO, 2025). Payment infrastructure for ongoing API costs requires institutional support or external funding.

Framework for Deployment in Under-Resourced Communities

If someone wanted to actually use these tools in communities that need them most, here is what it would take: Partnership with Local Organizations (do not parachute in with technology), Teacher Training (these tools augment but do not replace human teachers), Adaptation for Context (language translation, cultural relevance, curriculum alignment), Sustainable Funding Model (institutional funding, grants, freemium models, tech partnerships), Offline Functionality (pre-generating common examples, caching content), and Measurement and Iteration (deploy small, measure impact, iterate based on feedback). Rodriguez-Segura's (2022) systematic review of 81 EdTech studies in developing countries found that self-led learning and instructional improvements are most effective, while access alone is insufficient.

Limitations and Ethical Considerations

I need to be honest about what these tools cannot do and what risks they create. The Digital Divide is Real: Building AI education tools while many students lack computers or internet access might actually widen inequality. Risk of Reducing Human Interaction: Education is fundamentally social. Over-reliance on AI tools could reduce the human connection that makes education meaningful.

Data Privacy: Both systems collect sensitive data about what students know and struggle with. Research shows students feel surveilled and have low awareness of institutional data practices (Jones et al., 2020). Strong privacy protection, clear consent processes, and policies about data access are essential. AI Bias: Baker and Hawn's (2022) foundational review documents algorithmic bias across race, gender, nationality, socioeconomic status, and disability in educational AI systems. LLMs trained on internet data may inadvertently reinforce stereotypes or exclude certain perspectives. Constant monitoring and testing with diverse user groups is necessary.

Accuracy and Hallucination: LLMs sometimes generate plausible-sounding but incorrect information. In education, wrong answers can be worse than no answers. The system needs strong safeguards: citation requirements, confidence scoring, human oversight, and clear communication to students that AI can make mistakes. Educator Oversight is Essential: These tools work best when teachers use them to inform instruction, not replace it.

What Success Actually Looks Like

Success is not "AI solves educational inequality." That is not happening. Success is: students who previously struggled with CS finding ways into the field, teachers spending less time answering basic questions and more time on deeper learning, under-resourced schools having access to adaptive learning support, data that helps educators understand what students need, students feeling more confident asking questions, and gradual improvements in learning outcomes for underrepresented students. The goal is to expand access at the margins, to provide support where it did not exist before.

VII. GLASS Journey Reflection

I wrote an essay my freshman year about Dominican students at NYU. The observation that stuck with me: every Dominican student I met at Tandon had received at least part of their education in the United States. I could not find a single student who had been educated entirely in the Dominican Republic and still gotten into NYU. That statistic felt personal because my mom is Dominican, because I grew up seeing how education gaps play out, because I knew brilliant people who never got chances they deserved.

That essay was exploratory. I did not know what to do about it. Three years later, I am building tools that try to address pieces of that problem. The journey between those two points is what GLASS made possible.

How Understanding Evolved

Freshman year, I thought the problem was simple: some countries have worse schools, so we need to fix the schools. Sophomore year as a TA, I started seeing it differently. The students I taught were not failing because they were not smart. They were failing because the way we taught did not match how they learned. Some students needed visual explanations. Others needed to talk through concepts. Most needed someone to just explain things differently than the textbook did.

That is when I realized the problem is not just access to schools. It is access to the kind of personalized support that makes education actually work.

Studying abroad in Abu Dhabi (twice) showed me how educational systems work in dramatically different contexts. I took courses with students from across the Middle East, Africa, Europe, Asia. The technical knowledge we were all learning was the same, but the paths we took to get there were wildly different. Students from international schools had different preparation than students from local schools. Students from wealthy families had private tutors. Students like me had figured it out through a mix of luck and determination.

The universal pattern: students succeed when they have support that adapts to their needs. The difference between educational systems is not just resources, it is flexibility.

How Experience Shaped Design

The TA work directly influenced how I am building these tools. I noticed that students asked the same basic questions every semester, taking up huge portions of office hours while students with deeper questions waited. That observation led to the Q&A system. I noticed that the example that clicked for one student made no sense to another. That led to the personalized learning platform.

As a mentor in the Academic Achievement Program, I worked with first-generation students and underrepresented students navigating NYU. Many struggled not with the coursework itself but with knowing what resources existed, what questions to ask, who to talk to. The students who succeeded were not necessarily smarter. They had better information navigation skills or more cultural capital. That is a solvable problem.

My internships at Polydelta AI and Bank of America taught me how to actually build production ML systems. The technical skills, the system design thinking, the understanding of what is feasible at scale. GLASS gave me the mission and the framework for thinking about global challenges. The internships gave me the technical capabilities to actually build solutions.

Where This Goes Next

After graduation, I want to keep working on this problem. Not necessarily these exact tools, but the broader question of how technology can expand educational access. I am interested in roles where I can build AI systems for social impact, where technical excellence serves a mission beyond profit.

In the short term, I plan to: complete these projects and properly evaluate them, open-source the code so others can adapt and improve, write about what I learned for other students interested in education technology, and stay connected to communities working on educational equity.

Long term, I want to be part of building the next generation of educational tools. Tools designed from the start for students who get left behind by traditional systems. Tools that scale personalized support. Tools that actually close gaps instead of widening them.

The dream is still what it was freshman year: make quality education more accessible to students in places like the Dominican Republic. I am just approaching it more practically now, building specific tools, measuring what works, being honest about limitations, and learning from people actually doing this work.

What GLASS Made Possible

Without GLASS, I probably still would have cared about education equity. I definitely would have learned to code. But I would not have had: the framework for thinking about global challenges and sustainable development goals, the funding to study abroad twice and see how education works in different contexts, the cohort of peers also working on making the world better through STEM, the push to connect my technical skills to social impact, the space to develop these projects as independent studies instead of just taking more classes, and the expectation that my work should matter beyond just getting a degree.

GLASS did not give me the mission. The mission came from my background, my mom's sacrifices, watching students struggle, caring about education gaps. But GLASS gave me the tools, the support, and the community to turn that caring into action.

VIII. Conclusion

Educational inequality is not one problem. It is thousands of interconnected barriers that keep students from accessing quality learning. With 251 million children out of school globally (UNESCO, 2024), 70% of children in low- and middle-income countries unable to read a simple text by age 10 (World Bank et al., 2022), and 44 million teachers needed by 2030 (UNESCO, 2024), the scale of the challenge is immense. These two projects address small pieces of that massive challenge. The CS learning platform tries to make personalized instruction scalable. The Q&A system tries to make information access equitable. Neither solves educational inequality. Both might help at the margins.

The technical contributions are straightforward: architectures for personalized learning systems using LLMs, frameworks for course Q&A systems using RAG, evaluation methodologies for measuring impact. Future work can build on these implementations, improve them, adapt them to other contexts.

The broader contribution is showing what is possible when you design educational technology with equity in mind from the start. Not "build a tool then figure out how underserved populations might use it," but "what do students facing barriers actually need, and how could technology provide that?"

I am still figuring out what these tools can do. The evaluation will reveal strengths, weaknesses, failure modes. I expect to find that some things work better than I hoped and others work worse. That is the process. Build, test, learn, iterate.

What I believe after three years of GLASS, after building these projects, after teaching and mentoring and studying abroad and interning: technology alone does not solve social problems. But technology designed thoughtfully, deployed carefully, evaluated honestly, and improved continuously can expand access to things that matter. Education matters. Access matters. Personalization matters.

The work continues. These projects are beginning, not end points. The code will get better. The evaluation will teach me what I got wrong. Other people will build better versions. That is how progress happens.

For now, I am focused on finishing these tools, testing them properly, and learning as much as I can about what actually helps students learn. Everything else follows from that.

References

Baker, R. S., & Hawn, A. (2022). Algorithmic bias in education. *International Journal of Artificial Intelligence in Education, 32*, 1052-1092. https://doi.org/10.1007/s40593-021-00285-9

Bartlett, F. C. (1932). *Remembering: A study in experimental and social psychology*. Cambridge University Press.

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching, 20*(1), 45-73.

Cahalan, M., Vaughan III, T., et al. (2024). *Indicators of higher education equity in the United States: 2024 report*. Pell Institute for the Study of Opportunity in Higher Education.

Cataldi, E. F., Bennett, C. T., & Chen, X. (2018). *First-generation students: College access, persistence, and postbachelor's outcomes* (NCES 2018-421). National Center for Education Statistics.

Chen, X. (2013). *STEM attrition: College students' paths into and out of STEM fields* (NCES 2014-001). National Center for Education Statistics.

Cohn, C., Hutchins, N., Le, T., & Biswas, G. (2024). A chain-of-thought prompting approach with LLMs for evaluating students' formative assessment responses in science. *Proceedings of the 38th AAAI Conference on Artificial Intelligence, 38*(21), 23182-23190.

Eames, T., Brunskill, E., Yamkovenko, B., Weatherholtz, K., & Oreopoulos, P. (2026). [Khan Academy effectiveness study]. *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.2507708123

Gable, R. (2021). *The hidden curriculum: First generation students at legacy universities*. Princeton University Press.

Gao, Y., et al. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint*, arXiv:2312.10997.

Hidi, S., & Renninger, K. A. (2006). The four-phase model of interest development. *Educational Psychologist, 41*(2), 111-127.

Ives, J., & Castillo-Montoya, M. (2020). First-generation college students as academic learners: A systematic review. *Review of Educational Research, 90*(2), 139-178.

Jones, K. M. L., Asher, A., Goben, A., et al. (2020). 'We're being tracked at all times': Student perspectives of their privacy in relation to learning analytics in higher education. *Journal of the Association for Information Science and Technology, 71*(9), 1044-1059.

Kestin, G., Miller, K., Klales, A., Milbourne, T., & Ponti, G. (2025). AI tutoring outperforms in-class active learning: An RCT introducing a novel research-based design in an authentic educational setting. *Scientific Reports, 15*, 17458.

Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems: A meta-analytic review. *Review of Educational Research, 86*(1), 42-78.

Leonard, H. C., & Sentance, S. (2021). Culturally-relevant and responsive pedagogy in computing: A quick scoping review. *International Journal of Computer Science Education in Schools, 5*(2), 3-13.

Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459-9474.

Li, Z., Wang, Z., Wang, W., Hung, K., Xie, H., & Wang, F. L. (2025). Retrieval-augmented generation for educational application: A systematic survey. *Computers and Education: Artificial Intelligence, 8*, 100417.

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology, 106*(4), 901-918.

Major, L., & Francis, G. (2021). The effectiveness of technology-supported personalised learning in low- and middle-income countries: A meta-analysis. *British Journal of Educational Technology, 52*(5), 1935-1964.

McGee, E. O. (2021). Racism camouflaged as impostorism and the impact on Black STEM doctoral students. *Race Ethnicity and Education*. https://doi.org/10.1080/13613324.2021.1924137

Morales-Chicas, J., Castillo, M., Bernal, I., Ramos, P., & Guzman, B. L. (2019). Computing with relevance and purpose: A review of culturally relevant education in computing. *International Journal of Multicultural Education, 21*(1).

National Student Clearinghouse Research Center. (2019). *High school benchmarks: National college progression rates*.

Oreopoulos, P., Gibbs, C., Jenson, J., & Price, J. (2024). Khan Academy mastery learning intervention [RCT study]. Khan Academy Research.

Pane, J. F., Griffin, B. A., McCaffrey, D. F., & Karam, R. (2014). Effectiveness of Cognitive Tutor Algebra I at scale. *Educational Evaluation and Policy Analysis, 36*(2), 127-144.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Piaget, J. (1952). *The origins of intelligence in children*. International Universities Press.

Pierce, G. L., Cleary, P. F., & Mollah, G. K. (2024). The persistent educational digital divide and its impact on societal inequality. *PLoS ONE, 19*(4), e0286795.

Rodriguez-Segura, D. (2022). EdTech in developing countries: A review of the evidence. *The World Bank Research Observer, 37*(2), 171-203.

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist, 55*(1), 68-78.

Sahoo, P., et al. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint*, arXiv:2402.07927.

Scott, A., Martin, A., McAlear, F., & Koshy, S. (2021). *Culturally responsive-sustaining computer science education: A framework*. Kapor Center.

Steele, C. M., & Aronson, J. (1995). Stereotype threat and the intellectual test performance of African Americans. *Journal of Personality and Social Psychology, 69*(5), 797-811.

Stebleton, M. J., & Soria, K. M. (2012). Breaking down barriers: Academic obstacles of first-generation students at research universities. *The Learning Assistance Review, 17*(2), 7-20.

UNESCO. (2022). World Teachers' Day: UNESCO sounds the alarm on the global teacher shortage crisis. UNESCO Publishing.

UNESCO. (2024). *Global education monitoring report 2024/5*. UNESCO Publishing.

UNESCO. (2025). AI and education: Protecting the rights of learners. UNESCO Publishing.

UNESCO & International Task Force on Teachers for Education 2030. (2024). *Global report on teachers: Addressing teacher shortages and transforming the profession*. UNESCO Publishing.

UNESCO Institute for Statistics. (2025). Out-of-school rate. *Global Education Monitoring Report VIEW Platform*.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes* (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.). Harvard University Press.

World Bank, UNICEF, FCDO, USAID, Bill & Melinda Gates Foundation, & UNESCO. (2022). *The state of global learning poverty: 2022 update*. World Bank Group.

Yan, L., Sha, L., Zhao, L., et al. (2024). Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology, 55*(1), 90-112.

Zheng, L., Long, M., Zhong, L., & Gyasi, J. F. (2022). The effectiveness of technology-facilitated personalized learning on learning achievements and learning perceptions: A meta-analysis. *Education and Information Technologies, 27*(8), 11807-11830.