Documentation

Project Documentation: Intelligent Tutor

Project Summary

The intelligent tutor is a system that generates targeted assessments, grades and analyzes them, and uses the results to construct a dynamic learner profile. This profile can then be used by AI tutors to provide highly personalized learning experiences. The project addresses the lack of personalization in traditional learning systems and empowers learners to identify their strengths, gaps, and next steps. By supporting mastery learning, adaptive curricular design, and motivation through progress tracking, this project aims to make personalized, self-guided learning effective and scalable across disciplines.

Project Executives

Project Lead: Eric Fackelman - efackelm@charlotte.edu

Project Information

Background/Motivation:

Traditional education often follows a one-size-fits-all model where learners must follow rigid curricula designed for the average student. This leaves little room for personalization based on individual interests, aptitudes, and goals. Self-learners face the opposite problem: while resources are abundant online, they lack effective systems to assess progress, identify gaps, and adaptively choose what to learn next.

The project is motivated by the need to bridge this gap using AI. Generative AI can create tailored assessments, analyze performance, and build dynamic learner profiles. These profiles can serve as the foundation for adaptive tutoring systems that recommend the right next step at the right time, modulating difficulty and providing motivation. In the long term, this system could shift the role of teachers and institutions from delivering one-size-fits-all instruction to mentoring, motivating, and accrediting learners who follow personalized, AI-supported pathways.

Objectives/Goals:

Final Goal:
Launch a functional MVP that demonstrates the system's core value: generating assessments, building a learner profile, and providing personalized next-step recommendations.

Key Objectives:

1. Build Foundational Assessment System:
- Develop prompt templates to generate valid questions across multiple disciplines.
- Support common assessment formats (MCQ, short answer, fill-in-the-blank).

2. Develop Learner Profile Representation:
- Create a simple database schema that tracks concepts, mastery scores, and history.
- Implement profile updating logic after each assessment.

3. Grading & Analysis Pipeline:
- Automate grading for objective formats.
- Use LLM evaluation for free-response with confidence scoring.

4. Personalized Recommendation Engine:
- Recommend next concepts and learning paths based on mastery gaps.
- Generate explanations of why the learner should study a concept.

5. Build Core User Interface:
- Dashboard with mastery visualization (progress bars or graphs).
- Assessment-taking flow with instant feedback.

6. Integrate & Test System:
- Connect assessment generation, grading, learner profile, and UI into a single flow.
- Conduct pilot testing with students to validate core experience.

Methodology/Tasks:
Our methodology is to iterate quickly with a focus on the core loop: assessment → profile → recommendation.

1. **Backend Setup & Data Modeling**

- Task: Design learner profile schema (concepts, scores, timestamps).
- Task: Build API endpoints to serve assessments, results, and profile data.

2. **Assessment Generation & Grading**

- Task: Develop LLM prompts for multiple-choice, short-answer, and simple applied questions.
- Task: Implement auto-grading pipeline with hybrid rule-based + LLM grading.

3. **Learner Profile & Analytics**

- Task: Implement mastery score updates using simple weighted averages.
- Task: Build logic for spaced review & weak-concept targeting.

4. **Personalization Engine**

- Task: Generate next-step recommendations aligned with student goals.
- Task: Create rationales for why a topic is suggested.

5. **Frontend & UI Development**

- Task: Build dashboard to visualize mastery progress over time.
- Task: Implement “Take Assessment” and “View Next Steps” flows.

6. **Integration & Testing**

- Task: Deploy prototype system (e.g., Streamlit or React + FastAPI).
- Task: Conduct pilot testing with 5–10 learners to validate usability.

Scope:
The MVP will focus on supporting a few select disciplines (e.g., Python programming, introductory math, language learning, history) with a small set of assessment types. The system will showcase its ability to:

- Generate and grade assessments.
- Build and update a learner profile.
- Provide personalized, adaptive recommendations.

Future expansions will include:

- Support for more domains and complex assessments (essays, projects).
- More advanced knowledge tracing models (e.g., Deep Knowledge Tracing).
- Gamification (badges, challenges, leaderboards).
- Integration with institutional platforms for accreditation.
- Social and collaborative learning features.

Team Size:
5-10

GitHub:
https://github.com/Charlotte-AI-Research/intelligent-tutor

Getting started

If you're interested in preparing for the project's start, here's a comprehensive list of topics to learn up on. This is *not* expected, but it's here if anyone's interested.

Background knowledge

- Intelligent Tutoring Systems (big area of research in education technology)
    - In industry: Khanmigo, Cognitive tutor, startups
    - 2025 AI in Education Journal for latest research: https://iaied.org/journal
- Zone of proximal development (ZPD) and why it's important for student engagement
    - Modeling student knowledge allows our tutor to stay inside the ZPD
- Mastery learning and the evidence to support it being highly effective
- Cognitive load theory - germane, intrinsic, extraneous load
- Formative, summative, and diagnostic assessment
- Item Response Theory (how to assess knowledge)
- What are knowledge tracing algorithms

Technical

- Software engineering fundamentals (we need both frontend and backend so learn what interests you most)
- Git for version control
- External API use (OpenAI, HuggingFace, etc)
- Prompt engineering, context engineering

Additional Videos

Sal Khan - Why AI Could Save (Not Destroy) Education
https://youtu.be/hJP5GqnTrNo?si=ULPW0n5XroWbIyon

Veritasium - What Everyone Gets Wrong About AI and Learning
https://youtu.be/0xS68sl2D70?si=91VCorjnImuEbOpC

Roadmap

Papers

Important research questions to look into:

- How do pre-trained LMs perform in pedagogical settings (i.e. generating assessment, explaining concepts, generating examples, asking critical questions, etc)
- What NLP-based approaches to knowledge tracing are best suited to our goals?
- What kind of sycophantic bias exists in LM that will inhibit tutoring ability (i.e. not being critical enough when evaluating student response)

Benchmarking the Pedagogical Knowledge of Large Language Models
https://arxiv.org/abs/2506.18710

Exploring knowledge-tracing in student-tutor dialogues using LLMs
https://arxiv.org/pdf/2409.16490

- dialogue-focused KT framework
- significantly outperforms previous KT
-

Language Bottleneck Models: A Framework for Interpretable Knowledge Tracing and Beyond
https://arxiv.org/pdf/2506.16982

- compact NLP summary of learner state
- human-readable and interpretable states

Notes

Archive

WORKFLOW VISION:

user explains what they want to learn

system asks clarifying questions until it meets a threshold of clarity about learner goals

system generates a high-level composition of learner goals

user can accept or reject system-generated goal components

goal components map to knowledge components

knowledge components map to mastery score & knowledge assessment (text)

user picks knowledge component to learn

system presents entrance assessment

user completes assessment

system analyzes results, gives feedback, stores insights in learner profile

system suggests targeted practice on missing knowledge

High-level functionality:

User demonstrates knowledge through assessment
System tracks knowledge for targeted instruction & assessment

Canvas-native intelligent tutoring system

Agentic, learner-adaptive canvas page personalized to a student

Student can ‘tell the page’ what their learning goals are

The page updates itself with new learning content and assessments

Upon student submission, system can analyze attempt and update student profile

Then, system can generate new content and post to canvas

We can then give an LLM tutor access to the context of the student profile, allowing it to tailor instruction accordingly