AI Curriculum Planner: Adaptive Academic Advising for Simulated Students

1. Graph Schema Design
We modeled the university curriculum as a directed graph where:
- Nodes = courses
- Edges = prerequisite relationships

The curriculum includes 13 courses categorized under foundational, intermediate, and specialized tracks (AI, Security, Data Science):

Course	Prerequisites
Intro CS	None
Math I	None
Math II	Math I
Data Structures	Intro CS
Algorithms	Data Structures, Math II
AI	Algorithms
ML	AI
NLP	AI
Data Science	Algorithms
Security	Algorithms
Blockchain	Security
Deep Learning	ML
Ethics	None

Implemented using NetworkX in Python for flexibility and visualization.
The graph ensures no student can take a course without completing its prerequisites.

2. Student Simulation
We simulate a cohort of 100 students. Each student has:
- A unique student ID
- A GPA randomly generated between 2.0 and 4.0
- A set of completed courses, respecting the graph's prerequisite logic
- Grades (on a 1.0–4.0 scale) for passed courses
- A randomly assigned interest domain: "AI", "Security", or "Data Science"

Simulation ensures that each student has a valid and realistic academic history that follows prerequisite rules.
Example Generated Student:


```
{
  "id": 0,
  "completed_courses": ["Math I", "Math II", "Intro CS"],
  "grades": {"Math I": 2.34, "Math II": 3.85, "Intro CS": 3.12},
  "gpa": 3.31,
  "interest": "Data Science"
}
```


3. Personalization Strategy
We implemented a heuristic-based recommendation system (similar to Reinforcement Learning but faster for prototyping).

Key Logic:
- Eligibility Check: Course is recommended only if prerequisites are met.
- Interest Matching: Courses aligned with the student's interest are prioritized.

Constraints:
- Max 3–5 courses per term
- No duplicate or invalid courses
- Failed courses can be retaken in a future term (not implemented in this basic version)

Output:
For each student, the system recommends top courses for the next term:

"recommended": ["Data Structures"]

4. Evaluation and Visualization

While not using full Reinforcement Learning due to time constraints, this prototype:
- Accurately enforces course logic
- Aligns output with student interests
- Demonstrates potential for future Q-learning or DQN extension

Graph visualization was performed using matplotlib + networkx for curriculum mapping.

5. Future Work

- Add Reinforcement Learning to improve long-term planning.
- Include graduation tracking and timeline visualization.
- Integrate Neo4j for large-scale querying of graph data.
- Add advisor simulation and feedback loops.

Author

Haidy Hamed
Candidate for CIS-2025-20 Research Internship Track
GitHub: https://github.com/HaidyyHamed/AI-Academic-Advisor.git
Email: haidyhamed263@gmail.com