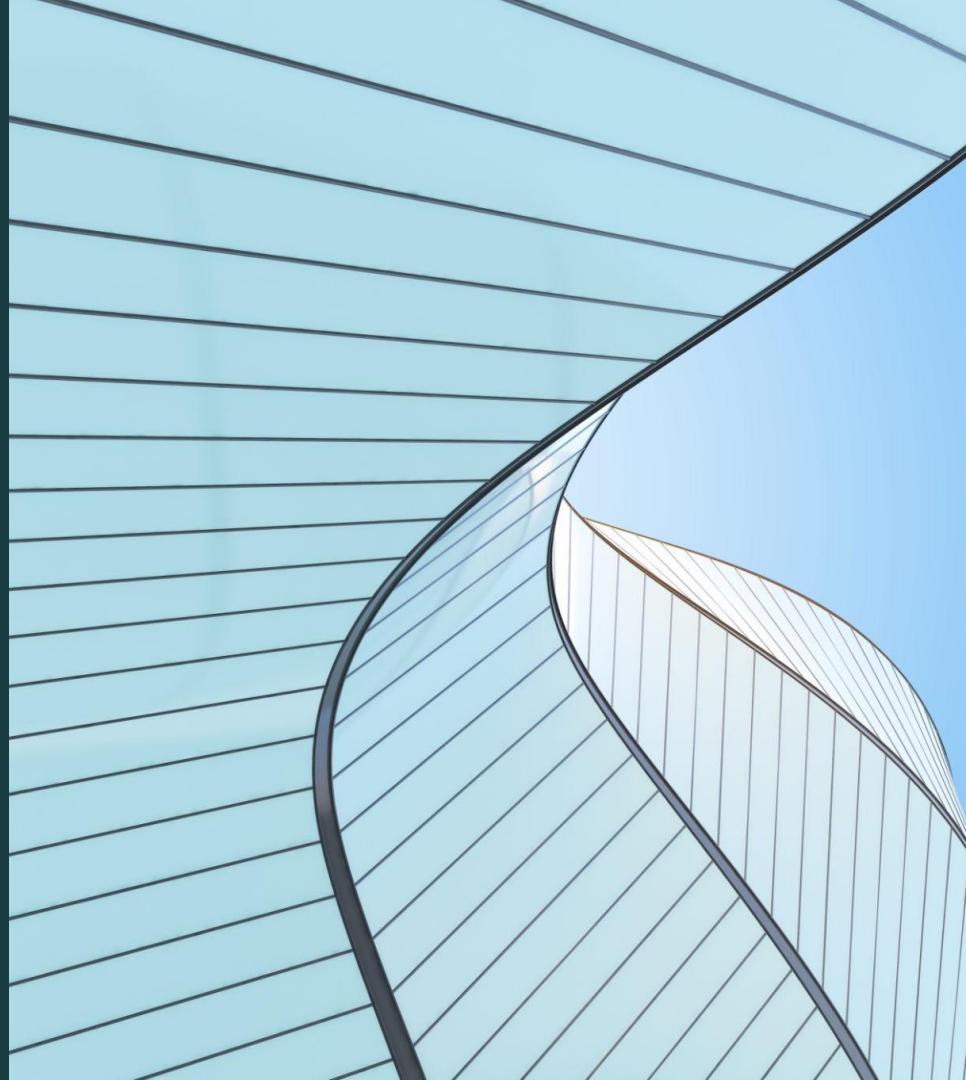


Intern/Mentor Presentation

Artificial Intelligence: Applications to Pedagogical Course Content

Esha Hegde



“The measure of intelligence is the
ability to change.”

– Albert Einstein

My Journey

Why:

- **Interest:** Took a few courses (online!) on Python, data structures, etc; found the AI field very intriguing.
- **Diving Deeper:** AI evidently can be applied to different fields such as healthcare, finance, government, etc.

Why not use it to revolutionize education?

How:

- **Internship Focus:** Conduct market research, help develop product ideas and pitch strategies, test AI generated courses + contribute to software, and learn software concepts through reverse engineering existing code.
- **Goal:** Understand how AI-generated courses compare to human-designed curricula and optimize them.



Background History

1960s-1980s



Computer-Assisted Instruction

Computer-Assisted Instruction and early Intelligent Tutoring Systems; content manually created, limited adaptivity

1990s-2000s



Adaptive Learning

Rule-based systems evolved; adaptive learning began using student data for personalized pacing.

2000s



Natural Language Processing

Advances in Natural Language Processing (NLP) enabled automated text generation and dynamic lesson content.

2020s-present



Large Language Models

Generative AI (e.g., large language models) can create full courses aligned to standards, reducing manual effort.

Ongoing Challenges

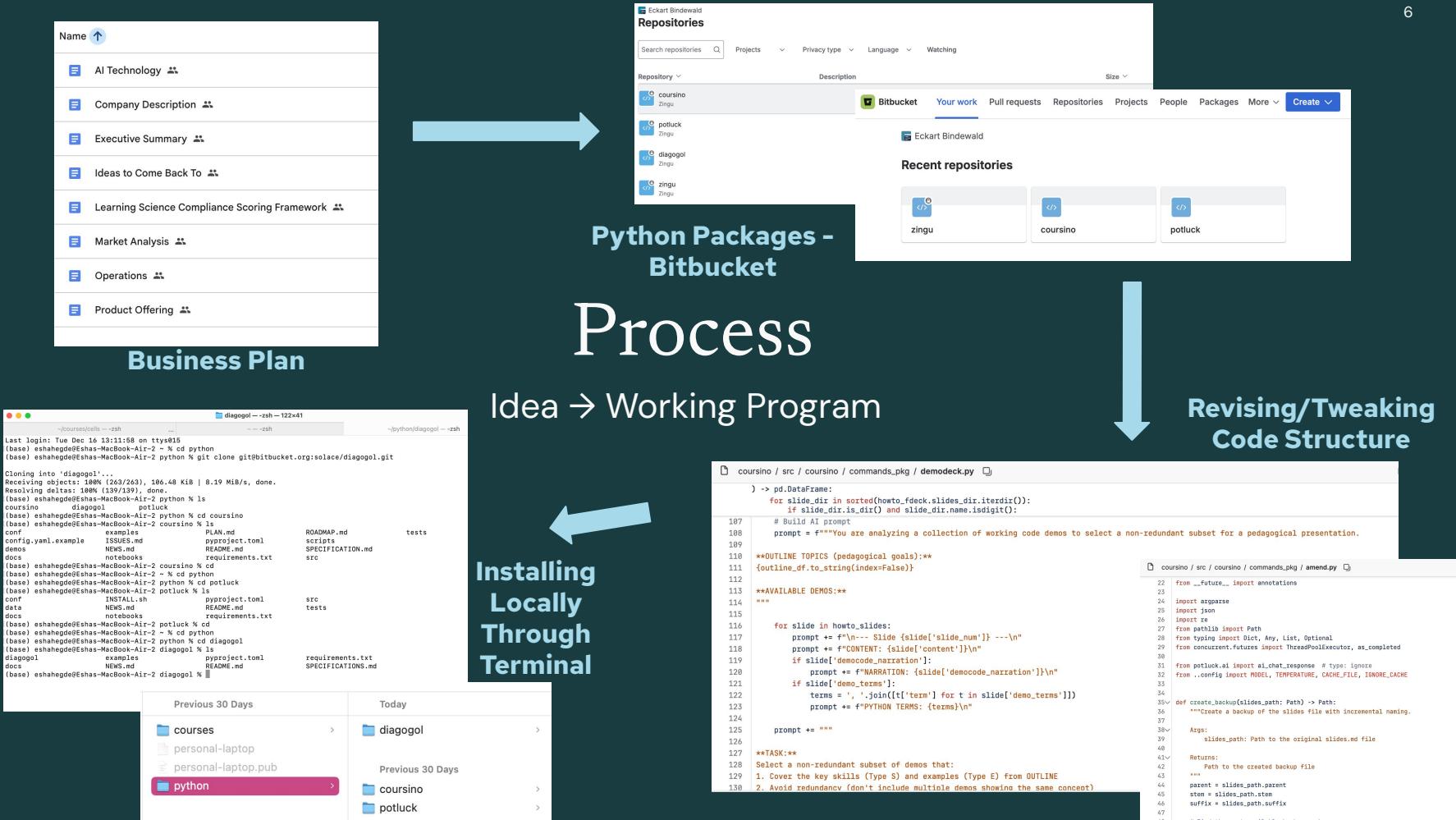
Accuracy, bias, instructional quality, and integration with teacher-led strategies.

Research Question

1. How can artificial intelligence (AI) be leveraged to generate and optimize educational content that aligns with established academic standards?
2. How does the accuracy and quality of this content compare to that of human-designed curricula?

Hypothesis

If artificial intelligence is utilized to generate educational course content based on established educational standards, then the resulting materials will demonstrate comparable or higher accuracy and instructional quality than human-designed curricula, requiring less time and manual effort to produce.





Process

Folder + Package Structure in Windsurf - IDE Editor

The screenshot shows the Windsurf IDE interface with the following structure:

```

    .
    ├── build.py
    ├── commands
    │   ├── __init__.py
    │   ├── amend.py
    │   ├── atomic.py
    │   ├── autoconfig.py
    │   ├── autoreview.py
    │   ├── backup.py
    │   └── build.py
    ├── config
    ├── courses
    ├── dependencies
    ├── docs
    ├── examples
    ├── filesdeck
    │   ├── accessibility_demo.py
    │   ├── import_demo.py
    │   └── tutechammon_syntax_demo.md
    ├──风帆规则
    ├──风帆测试规则
    └──风帆
        ├── coursino
        │   ├── __init__.py
        │   ├── commands
        │   │   ├── __init__.py
        │   │   ├── commands_pkgs
        │   │   │   ├── build.py
        │   │   │   └── find_snakefile.py
        │   │   └── __init__.py
        │   └── __main__.py
        └── __main__.py

```

Running/Testing Program - Generation

Code snippet from build.py:

```

19 def find_snakefile() -> Optional[Path]:
20     Search order:
21     1. Snakefile in current directory (user override)
22     2. Snakefile in coursino repo root (development)
23     3. Snakefile.prototype for single presentations
24
25     Returns:
26     ---- Path to Snakefile, or None if not found
27
28     # Check current directory (user override)
29     local_snakefile = Path.cwd() / "Snakefile"
30     if local_snakefile.exists():
31         return local_snakefile
32
33     # Find coursino package and look for Snakefile
34
35     import coursino
36     # coursino... __file__ is like: /path/to/site-packages/coursino/__init__.py
37     package_dir = Path(coursino.__file__).parent
38
39     # First try package data directory (for pip installs)
40     data_dir = package_dir / "data"
41     full_snakefile = data_dir / "Snakefile"
42     if full_snakefile.exists():
43         return full_snakefile
44
45     # Fallbacks: try repo root (for development installs)
46     repo_root = package_dir.parent.parent
47     full_snakefile = repo_root / "Snakefile"
48     if full_snakefile.exists():
49         return full_snakefile
50
51     prototype_snakefile = data_dir / "Snakefile.prototype"
52     if prototype_snakefile.exists():
53         # Only use prototype if we're in a presentation directory
54         if (Path.cwd() / "SYNOPSIS.md").exists():
55             return prototype_snakefile
56
57     # Fallbacks: try repo root (for development installs)
58     repo_root = package_dir.parent.parent
59     full_snakefile = repo_root / "Snakefile.prototype"
60     if full_snakefile.exists():
61         if (Path.cwd() / "SYNOPSIS.md").exists():
62             return full_snakefile
63
64     except Exception as e:
65         print(f"Error finding Snakefile: {e}")

```

Code snippet from critique_overall.json:

```

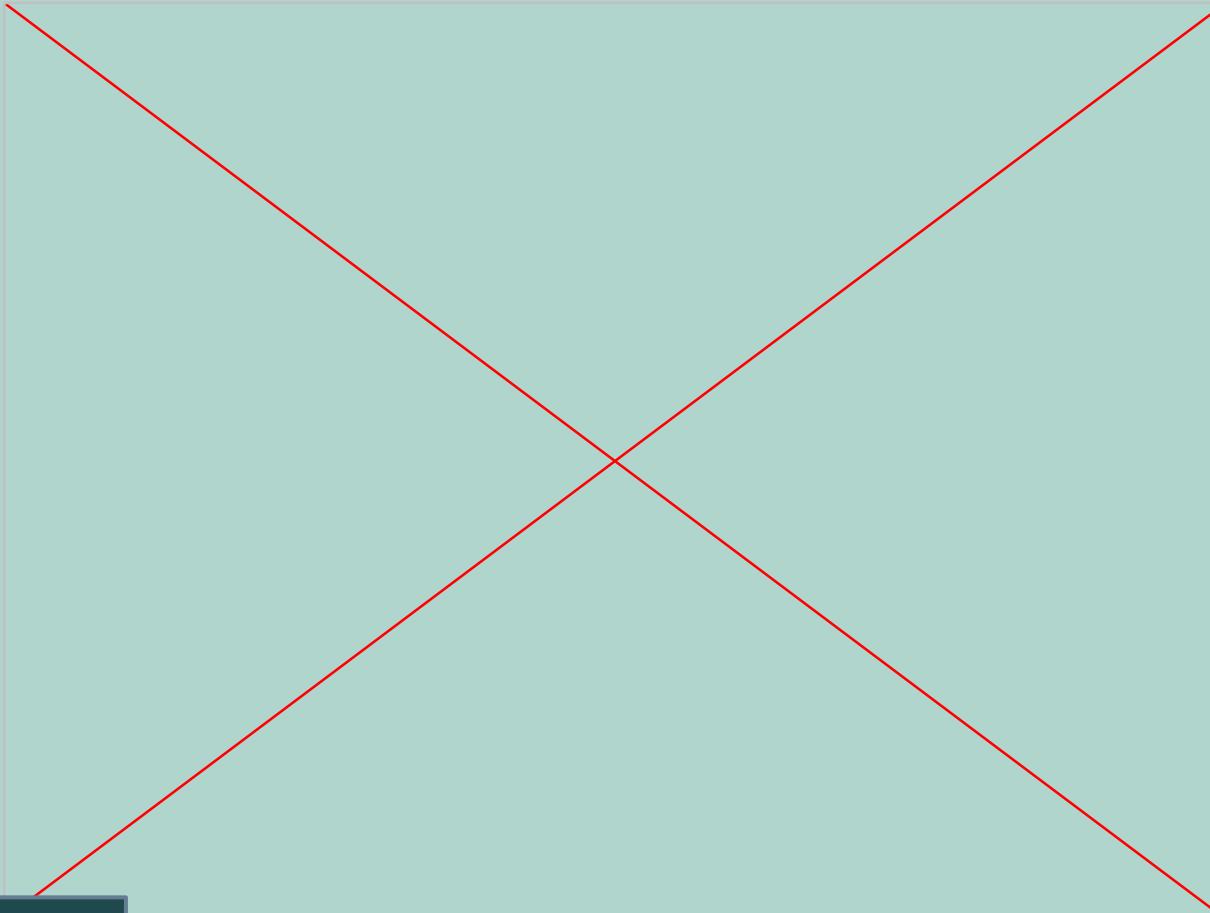
{
    "modules": [
        "modules/01_cells_foundations_and_energy"
    ],
    "lessons": [
        "lessons/01_cells_foundations_and_energy"
    ],
    "presentations": [
        "presentations/01_cells"
    ],
    "slides": [
        "slides/01"
    ],
    "deck": [
        "deck/01"
    ],
    "critique_overall": [
        "critique_overall.json"
    ],
    "dependencies": [
        "dependencies/tv"
    ],
    "outline_atomic": [
        "outline/tv"
    ],
    "outline_tv": [
        "outline/tv"
    ],
    "presentation_mp4": [
        "presentation.mp4"
    ],
    "synopsis_sm": [
        "synopsis_sm"
    ],
    "topics_tv": [
        "topics_tv"
    ],
    "antiSynopsis_sm": [
        "antiSynopsis_sm"
    ],
    "background_md": [
        "background_md.md"
    ],
    "background_md_tsv": [
        "background_md.tsv"
    ],
    "critique_overall_js": [
        "critique_overall.js"
    ],
    "dependencies_tsv": [
        "dependencies.tsv"
    ],
    "outline_atomic_tv": [
        "outline_atomic_tv"
    ],
    "outline_tv_tsv": [
        "outline_tv.tsv"
    ],
    "synopsis_tv": [
        "synopsis_tv"
    ],
    "antiSynopsis_md": [
        "antiSynopsis_md"
    ],
    "lessons_sm": [
        "lessons_sm"
    ],
    "synopsis_sm": [
        "synopsis_sm"
    ],
    "audience_yaml": [
        "audience.yaml"
    ],
    "courseyaml": [
        "courseyaml"
    ]
}

```

01 cells foundations and energy

- Introduce cellular respiration processes that convert biochemical energy into ATP, the cell's energy currency
- Review prokaryotic vs. eukaryotic cell structures and key organelles, with emphasis on mitochondria and energy roles
- Describe membrane structure and how water and other molecules move across it
- Build simple, stepwise models of the three stages of cellular respiration and relate each to relative ATP output
- Predict effects of changes like low oxygen or membrane damage on ATP production and cellular balance

▶ 0:00 / 19:00



Example Video
Course Generated

Discoveries So Far

```

MODEL_BALANCED = params.get("model_balanced", "gpt-5-mini") # "cerebras:gpt-oss-120b" # "gpt-5-mini" # "gpt-5-mini" # "groq:qwen
params["model_balanced"] = MODEL_BALANCED
MODEL_HIGHEND = params.get("model_highend", "gpt-5") # "cerebras:gpt-oss-120b" # "gpt-5-mini" # "gpt-5-mini" # "groq:qwen/qwen3-
params["model_highend"] = MODEL_HIGHEND
MODEL_FASTCHEAP= params.get("model_fastcheap", "cerebras:gpt-oss-120b")
params["model_fastcheap"] = MODEL_FASTCHEAP
MODEL_VISION = params.get("model_vision", "gpt-5-mini") # AI model for vision tasks
params["model_vision"] = MODEL_VISION
MODEL_AUDIO = params.get("model_audio", "gpt-40-mini-tts")
params["model_audio"] = MODEL_AUDIO
MODEL_NANO = params.get("model.nano", "gpt-5-nano") # "cerebras:gpt-oss-120b" # "gpt-5-nano" # minimal model, mostluy for running
params["model.nano"] = MODEL_NANO
MODEL_WEBSEARCH = params.get("model_websearch", "perplexity:sonar")
params["model_websearch"] = MODEL_WEBSEARCH
MODEL_DIAGRAM = params.get("model_diagram", "gpt-5-mini") # "groq:qwen/qwen3-32b" # "gpt-5" # 40-mini"
params["model_diagram"] = MODEL_DIAGRAM
MODEL_REASON = params.get("model_reason", "gpt-5-mini")
params["model_reason"] = MODEL_REASON
MODEL_RESEARCH = params.get("model_research", "perplexity:sonar-deep-research") # "gpt-40-mini"
params["model_research"] = MODEL_RESEARCH
MODEL_CODE = params.get("model_code", "groq/openai/gpt-oss-120b") # "gpt-40-mini"
params["model_code"] = MODEL_CODE
CODE_ARCHETYPE = params.get("code_archetype", "aider") # aider or claude
params["code_archetype"] = CODE_ARCHETYPE

```

Model Configuration

Volume of Content Generated

Can produce full courses (lessons, quizzes, activities) in minutes/hours vs. days for humans.

Prompt Engineering

The AI program generates content based on user-provided prompts; more detailed and specific prompts result in more focused and high-quality course output.

Model Discrepancies

The AI program generates content based on user-provided prompts; more detailed and specific prompts result in more focused and high-quality course output.

Time & Effort Comparison

AI significantly reduces production effort, enabling faster iteration vs. human-designed content.

Next Steps

Software

- Finish programming in-house diagram generation package
- Revise and finalize course generation software for efficiency and accuracy
- Implement software to generate various courses

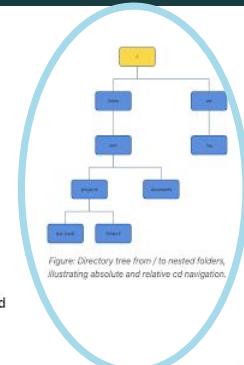
Business

- Bring courses to a user interface domain
- Learn more about the business aspect
- Test generated courses for instructional alignment and accuracy

Changing Directories with `cd`

- Use `cd` to move between directories in the Unix file system.
- Relative paths: `cd ..` moves up one level; `cd foldername` moves into a subdirectory.
- Absolute paths: `cd /home/user/documents` jumps directly to a specific location.
- Understanding relative vs. absolute paths is key for efficient navigation.
- Practice navigating nested directories to build command line confidence.

Figure: Directory tree from / to nested folders, illustrating absolute and relative cd navigation.



Current Diagram Example



Independent Research

- Earn GT credit
- Usually 10th grade -- open to all grades!
- Design and conduct **original research**
- Opportunity to **publish your work**

**Ideal if you enjoy
independence and exploring
your interests!**

Intern/Mentor

- Earn GT credit
- College- level research experience
- Grades 11 and 12 --
***transportation needed!**
- Work directly with **professional mentor**
- Gain real-world **experience** and build your professional network

Thank you!

Questions?