

AI-Driven Personalized Learning Platform

Adam Hisel, Carter Parks, Morgan Prieskorn, Zac
Mueterthies, Ryan Johnson



SD-14

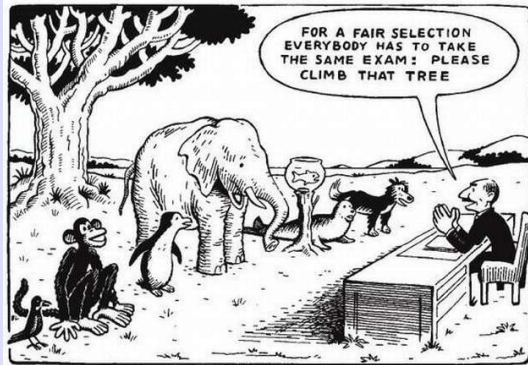
01

Topic

Topic

3

- Goal: Personalized supplemental learning
- Solution: AI-powered content
- **“Learn the way you want.” - LINK-X**

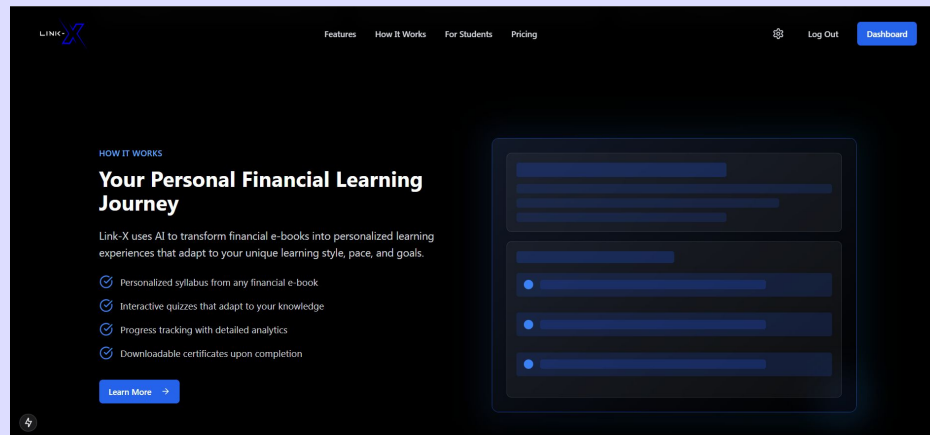
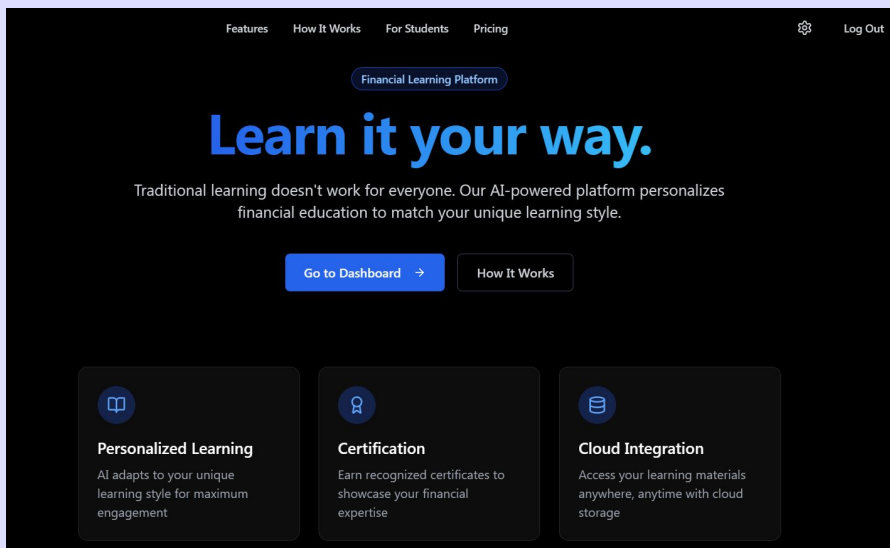


02

Design & Implementation

Styling And Feel

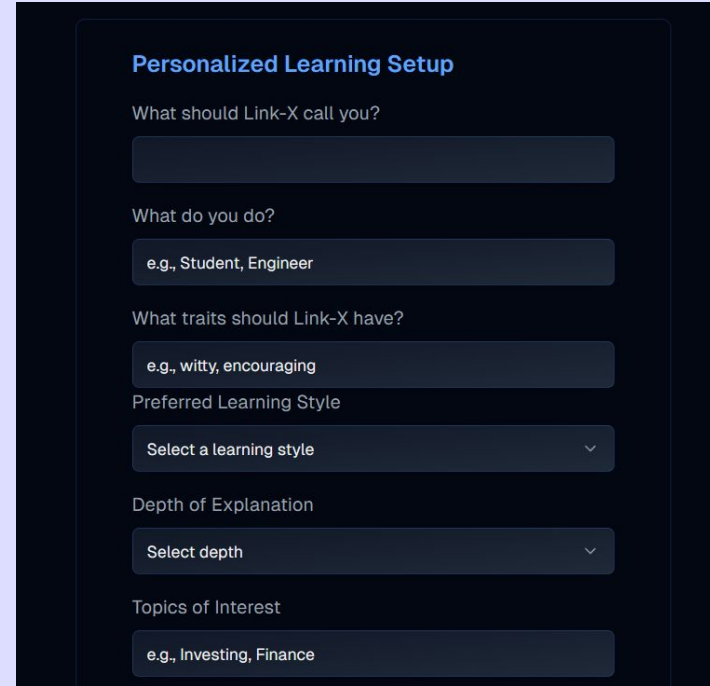
- **Universal Styling**
- **Business / Educational Feel**



Solution (Quick Overview)

6

- Gather Persona (Onboarding)
- Course Topic and Expertise Level
- Generate Course Outline
- Content with Persona
- Interactive (smart) chat



Personalized Learning Setup

What should Link-X call you?

What do you do?

e.g., Student, Engineer

What traits should Link-X have?

e.g., witty, encouraging

Preferred Learning Style

Select a learning style

Depth of Explanation

Select depth

Topics of Interest

e.g., Investing, Finance

Prompt 1

Issue:

- Gather Learning Subject
- Based on knowledge/experience

User Input:

- e.g., "I'm a sophomore in finance and I want to learn about investing"

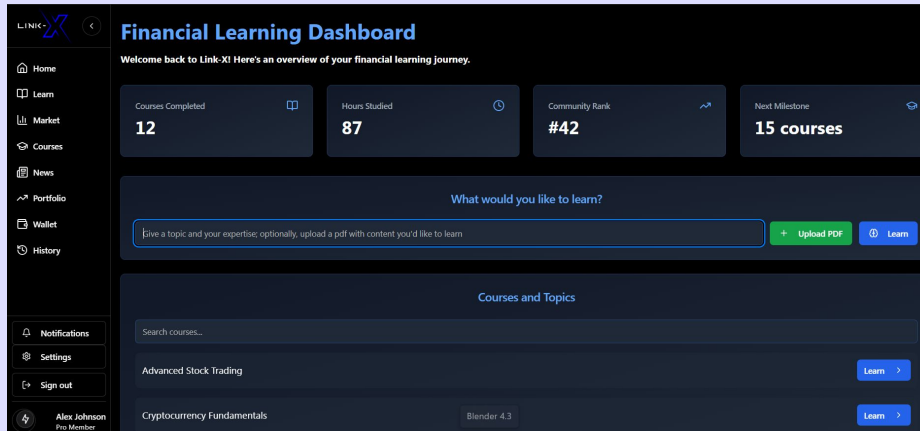
```
data = request.form or request.get_json()
question = data.get("question")
if not question:
    return jsonify({"error": "Missing question"}), 400

# Use OpenAI to extract topic and expertise
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
response = client.chat.completions.create(
    model="gpt-3.5-turbo",
    messages=[
        {
            "role": "system",
            "content": (
                "You are an education assistant. Extract a topic and the user's level of expertise from the question. "
                "Reply ONLY with a JSON object containing 'topic' and 'expertise' (one of: beginner, intermediate, advanced)."
            )
        },
        {"role": "user", "content": question}
    ]
)

import json
parsed = json.loads(response.choices[0].message.content)
topic = parsed.get("topic")
expertise = parsed.get("expertise")
if not topic or not expertise:
    return jsonify({"error": "Invalid GPT response"}), 400

# Generate course outline using the provided topic and expertise
outline = generate_course_outline(topic, expertise)
```

7



Prompt 2

Issue:

- When generating a course, first need an outline of the modules

Prompt:

- "You are an AI Assistant ...
 - user has provided a topic: {topic} and their expertise on the subject: {expertise} ...
 - *pdf possibly included* (FAISS & RAG)
 - task is to retrieve all relevant content based on their expertise and summarize it ...
 - For each chapter:
 - Provide a concise title (3-7 words)
 - Include an array of relevant metadata or key points
 - return as JSON"

```
{  
  "chapters": [  
    {  
      "chapterTitle": "string",  
      "metadata": [  
        "string",  
        "string",  
        ...  
      ]  
    },  
    ...  
  ]  
}
```


FAISS

Issue:

- Convert PDFs into a Knowledge-Base(KB)

FAISS:

- FaceBook AI Similarity Search
 - PDF divided into chunks & stored as vector embeddings
 - Generate references & citations
 - Store in index.faiss & index.pkl

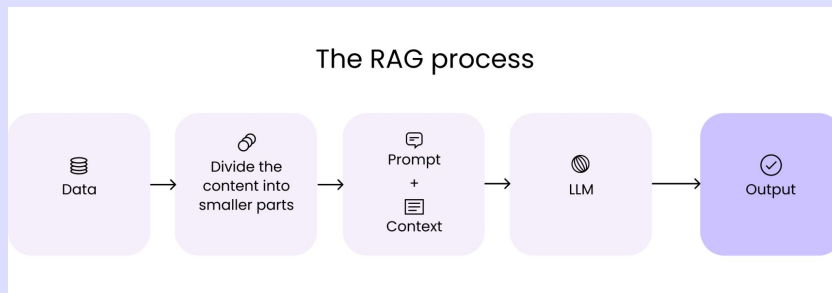
RAG

Issue:

- Use vector embeddings for course creation
- Extract relevant chunks

RAG:

- Retrieval-Augmented Generation
 - LLM performs retrievals from FAISS KB
 - Generates relevant knowledge based on prompt & KB



Prompt 3

Issue:

- For each module, the content needs to be generated by the LLM using Persona & RAG

Prompt:

- “You are a helpful and friendly AI tutor. {full_persona}{expertise}...”
 - *full_persona*: Name, traits, learning style, interests, etc
 - *expertise*: Description of knowledge level
 - Now explain this topic: {module_content}”
 - *module_content*: Title & key points

```
expertise_map = {  
  "beginner": "They prefer simple, clear explanations suitable for someone new to the topic.",  
  "intermediate": "They have some prior experience and prefer moderate technical depth.",  
  "advanced": "They want in-depth explanations with technical language.",  
}
```

Prompt 4

Issue:

- Questions asked in the AI chat should be relevant to the current course

Prompt:

- "You are an AI assistant ...
 - access to a knowledge base on a topic ...
 - provided the following query: {user_query} ...
 - task is to:
 - query is relevant → respond with a concise answer (w/ RAG & Persona)
 - query is **not** relevant → 'Query is not relevant to the Course. Please try again.'
- Output should only provide one of these two outcomes based on the relevance to content"

Prompts Put Together (Frontend)

13

Prompt 3:
Prompt 1:

I am a Freshmen in Computer Science

Prompt 4:

Definition of art

Hey Adam! Ready to flex those
break it down like a tactical play

Artificial Intelligence, often abbreviated as AI, refers to the simulation of human intelligence in machines that are programmed to think and learn like humans. These machines can be programmed to perform tasks that typically require human intelligence, such as visual perception, natural language processing, and decision-making.

1. **Learning:** Just like a basketball player improves through experience, AI systems learn from data. The more data they are fed, the better they become at performing tasks.

2. **Reasoning:** Think of this as a chess player. Similarly, AI systems can be programmed to make logical decisions based on the information they are given.

< AI Assistant

Tell me a little more about AI perception

AI perception refers to the ability of artificial intelligence systems to understand and interpret their environment through various sensory inputs, such as images, video, sound, and text. This involves using computer vision, natural language processing, and other technologies to analyze and interpret data from the real world. AI perception enables machines to perceive and understand the world around them in a manner similar to how humans do, allowing them to make sense of their surroundings, recognize patterns, and extract insights from complex data. Computer vision, which is a key component of AI perception, enables machines to interpret and understand visual information, such as images and videos, and use that information to make decisions or perform tasks.

AI Intelligence? Let's

revolutionizing how we use human intelligence and reasoning (using

AI systems improve algorithms that allow them to perform tasks and actions to

for decision-making and make decisions.

+ Upload PDF

Learn

Final Learning Page

14

LINK-X

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

Definition of artificial intelligence

Brief history of AI development

Importance and applications of AI

TYPES OF ARTIFICIAL INTELLIGENCE

Narrow AI vs. General AI

Reactive machines, limited memory, and rule-based systems

Examples of each type in real-world applications

MACHINE LEARNING BASICS

Definition and significance of machine learning

Supervised, unsupervised, and reinforcement learning

Common algorithms used in ML

DEEP LEARNING EXPLAINED

Overview of deep learning and its applications

Exit to Dashboard

Definition of artificial intelligence (AI)

Hey Adam! Ready to flex those mental muscles and dive into the all-star game of Artificial Intelligence? Let's break it down like a tactical playbook!

Artificial Intelligence, often abbreviated as AI, is like the MVP of the tech world, and it's revolutionizing how we interact with technology. In essence, AI refers to computer systems or machines that mimic human intelligence processes. These processes include learning (acquiring information and rules for using it), reasoning (using rules to reach approximate or definite conclusions), and self-correction.

- Learning:** Just like a basketball player fine-tunes their jump shot through practice, AI systems improve through experience. This is where machine learning (ML) comes into play, which involves algorithms that allow systems to learn from data. The system receives inputs, like a coach's feedback, and adjusts its actions to improve results.
- Reasoning:** Think of this as the playbook of strategies that a quarterback memorizes for decision-making on the field. Similarly, AI systems apply rules and logical reasoning to process information and make decisions.
- Perception:** AI's ability to interpret inputs from the world, akin to how athletes use their senses during a game. For example, computer vision allows AI to 'see' and understand images, which is like a player visualizing the field or court.
- Language Understanding:** Just like understanding the communication among teammates, AI has Natural Language Processing (NLP) to understand and generate human language. This enables AI to interface with users in a natural and meaningful way.

AI can be broadly classified into two types: **Narrow AI** and **General AI**.

Narrow AI: This is like a specialized athlete, highly skilled in one area, such as AI systems that excel at playing chess or recognizing your favorite football team.

AI Assistant

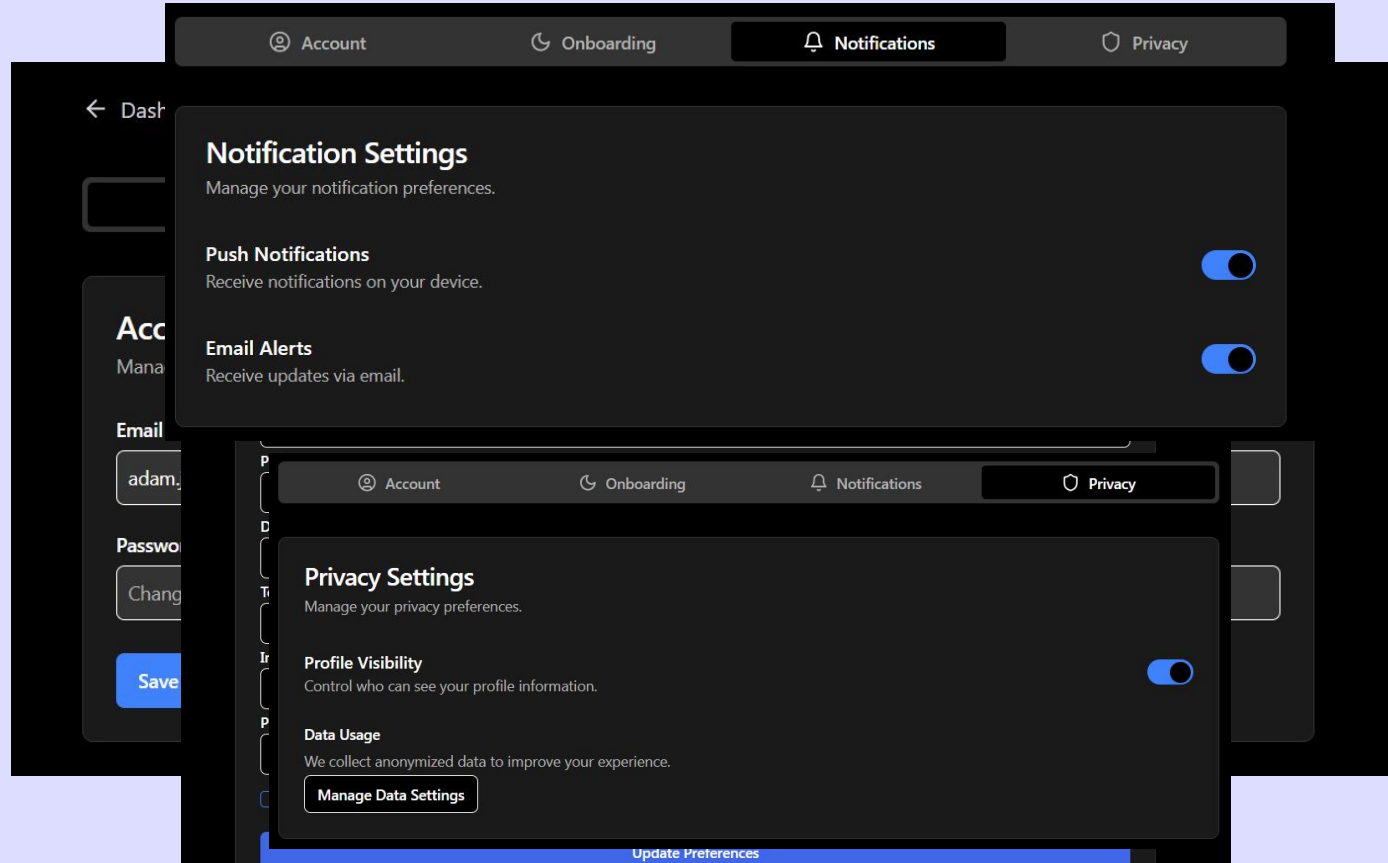
Tell me a little more about AI perception

AI perception refers to the ability of artificial intelligence systems to understand and interpret their environment through various sensory inputs, such as images, video, sound, and text. This involves using computer vision, natural language processing, and other technologies to analyze and interpret data from the real world. AI perception enables machines to perceive and understand the world around them in a manner similar to how humans do, allowing them to make sense of their surroundings, recognize patterns, and extract insights from complex data. Computer vision, which is a key component of AI perception, enables machines to interpret and analyze visual information, such as images and videos. This technology is used in various applications, such as facial recognition, object detection, autonomous vehicles, and medical imaging. Natural language processing (NLP) is another

Ask a question...

Settings Page

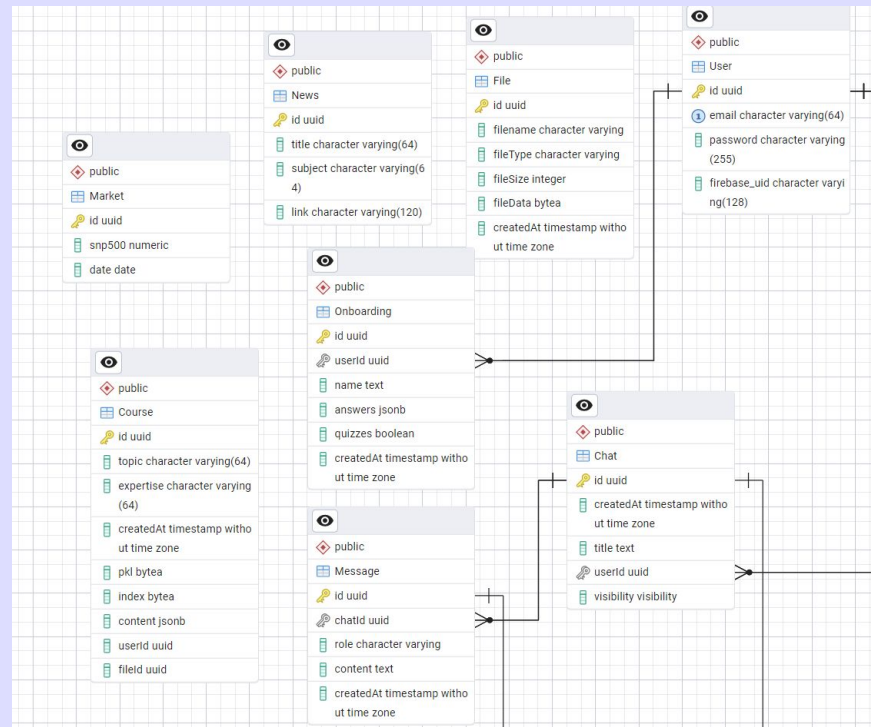
15



Database Current Progress

16

Identifier	Providers	Created ↓	Signed In	User UID
morgan4@gmail.com	✉	Apr 14, 2025	Apr 14, 2025	i6ztHuJGfkeiLYYGNTFGJabN...
morgan3@gmail.com	✉	Apr 14, 2025	Apr 14, 2025	QzqGwLGFvLVtGuF0TURO2dB...
test2@iastate.edu	✉	Apr 14, 2025	Apr 14, 2025	ttVVwsRHnCyrXEDoiMUnknj...
practice@iastate.edu	✉	Apr 14, 2025	Apr 14, 2025	AiCTlLiGu4VifNMu3mbKdGS5...
adamtest@iastate.edu	✉	Apr 14, 2025	Apr 14, 2025	M4AiOq7dPJgIXZUwnbWQAH...
test@iastate.edu	✉	Apr 14, 2025	Apr 14, 2025	eSCNfBV6Z0gh0Znj7MPMF4...
a@iastate.edu	✉	Apr 14, 2025	Apr 14, 2025	JnqLSIG8ZXcLNot1JRfV58rsH...
adamhise12@yahoo.c...	✉	Apr 14, 2025	Apr 14, 2025	mjV6sLpUHTaKuSYMjRP4cpcl...
carterp@iastate.edu	✉	Apr 13, 2025	Apr 14, 2025	rJxr50IDqeew9c1b5v5NIYQp...
demo@iastate.edu	✉	Apr 13, 2025	Apr 14, 2025	GaduYSgGfPZPANdfmkUrHAQ...
carter.p@iastate.edu	✉	Apr 13, 2025	Apr 13, 2025	rXat1AvoxSZLyJnuj3y5UIN0g...
ahisel@iastate.edu	✉	Apr 13, 2025	Apr 13, 2025	rpquqILfMzgzqf3ZuBQADS9NY...
parksc@iastate.edu	✉	Apr 13, 2025	Apr 13, 2025	Ay5r8dvU7iYzdPiY0nmbVEarG...




```

# Generate a course from a user question.
# Store the course in the Postgres database.
# Return the course id for use in the frontend.
@app.route('/create-course', methods=['POST'])
def learn_from_question():
    try:
        # Verify session cookie
        user = get_user_from_session()
        if "error" in user:
            return jsonify(user), 401

        # Get question from request body
        data = request.form or request.get_json()
        question = data.get("question")
        if not question:
            return jsonify({"error": "Missing question"}), 400

        # Use OpenAI to extract topic and expertise
        client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
        response = client.chat.completions.create(
            model="gpt-3.5-turbo",
            messages=[
                {
                    "role": "system",
                    "content": (
                        "You are an education assistant. Extract a topic and the user's level of expertise. "
                        "Reply ONLY with a JSON object containing 'topic' and 'expertise' (one of: beginner, intermediate, advanced)."
                    )
                },
                {"role": "user", "content": question}
            ],
        )
        import json
        parsed = json.loads(response.choices[0].message.content)
        topic = parsed.get("topic")
        expertise = parsed.get("expertise")
        if not topic or not expertise:
            return jsonify({"error": "Invalid GPT response"}), 400

```

```

# Generate course outline using the provided topic and expertise
outline = generate_course_outline(topic, expertise)
db_session = Session()

# Get Postgres UID from Firebase UID
postgres_user = get_user_by_firebase_uid(db_session, user["uid"])
if not postgres_user:
    return jsonify({"error": "User not found"}), 404

# Create the new course record
new_course = create_course(
    db=db_session,
    user_id=str(postgres_user.id),
    topic=topic,
    expertise=expertise,
    content=outline, # Course outline (JSON)
    pk1=None,
    index=None,
    file_id=None
)
db_session.close()

# Return the new course ID to the client
return jsonify({"message": "Course created successfully", "courseId": str(new_course.id)}), 200

except Exception as e:
    print("Error in /create-course:", str(e))
    return jsonify({"error": str(e)}), 500

```

03

Challenges

Team Challenges

19

- Build times vary.
- Moving backend out of Nextjs.
- Shifting directives.
- Fast-moving work environment.
- Moving from frontend to backend.
- Working with AI.

```
morga@DESKTOP-6JUAQQT MINGW64 ~/OneDrive/Documents/Senior Design/LINK-X (ryan)
$ bash run_backend.sh
Step 1: Building Backend...
[+] Building 2476.8s (16/17)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.15kB
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [ 1/12] FROM docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96
=> => resolve docker.io/library/python:3.10-slim@sha256:65c843653048a3ba22c8d5083a022f44aef774974f0f7f70cbf8cee4e931ac96
=> [internal] load build context
=> => transferring context: 13.41MB
=> CACHED [ 2/12] WORKDIR /app
=> CACHED [ 3/12] COPY src/requirements.txt ./
=> CACHED [ 4/12] RUN pip install --upgrade pip
=> [ 5/12] RUN pip install --no-cache-dir -r requirements.txt
=> [ 6/12] COPY src/.env ./
=> [ 7/12] COPY src/firebaseKey.json ./
=> [ 8/12] COPY src/index.faiss /app/index.faiss
=> [ 9/12] COPY src/index.pkl /app/index.pkl
=> [10/12] COPY src /app/src
=> [11/12] RUN pip install -U Flask flask-cors types-flask-cors
=> [12/12] RUN pip install faiss-cpu
=> exporting to image
=> exporting layers
1 warning found (use docker --debug to expand):
```

04

Remaining Work Timeline

Sprint Plan

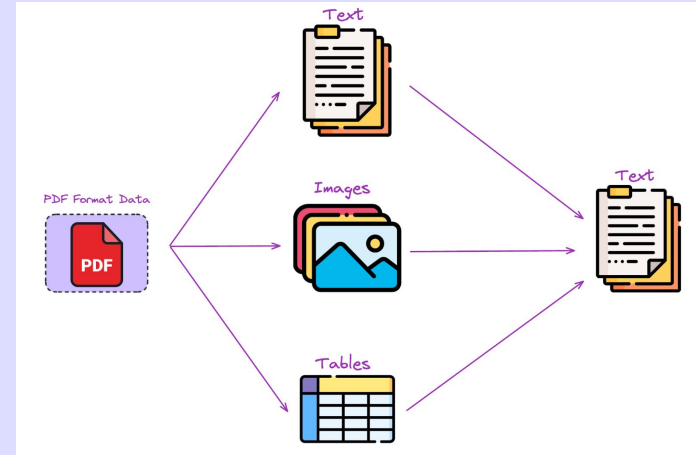
21

Week Starting	Sprint Plan	
2/2/2025	Onboarding and understanding code base	UI & Onboarding Complete
2/9/2025	Onboarding and understanding code base	
2/16/2025	Account creation and Authentication	
2/23/2025	Persona Creation and Database	
3/2/2025	Landing Page and Dashboard	
3/9/2025	Topics Page and Study Page	
3/16/2025	Spring Break	
3/23/2025	Testing Personalization & Backend	Have Working Personalization for any topic
3/30/2025	Testing Personalization & Backend	
4/6/2025	Make Database Backend to save personalization progress	
4/13/2025	Testing RAG for Uploading pdfs	
4/20/2025	Certificate Report for completion	
4/27/2025	Deployment	
5/4/2025	Extra ~ Analytics for Dashboard (user progress)	
5/11/2025	Finals Week	

Future Plans: Learn-X

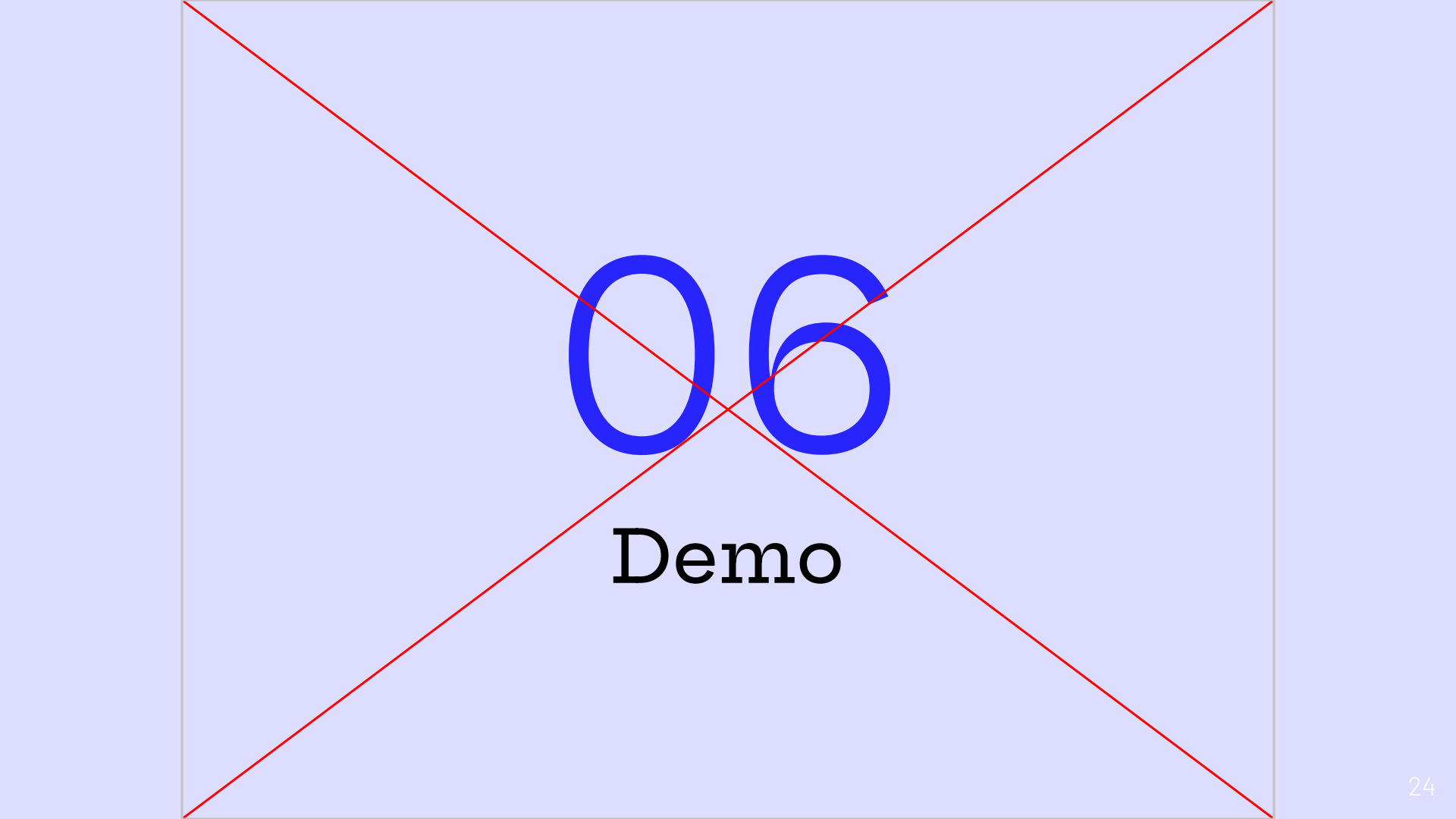
22

- **Integration With Canvas:** Pulling course content from canvas.
- **Analytics:** Allow for relationship between student and instructor. Course instructors can view students' progress.
- **Expanding Study Page Features:** Implement RAG and pdf content scraping. Connect chatbot to content.
- **Seamless Integration:** Ensure all pages work cohesively with existing components for a smooth workflow.



05

Demo



06

Demo

THANK YOU

