

# AI-Driven Personalized Learning Platform

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



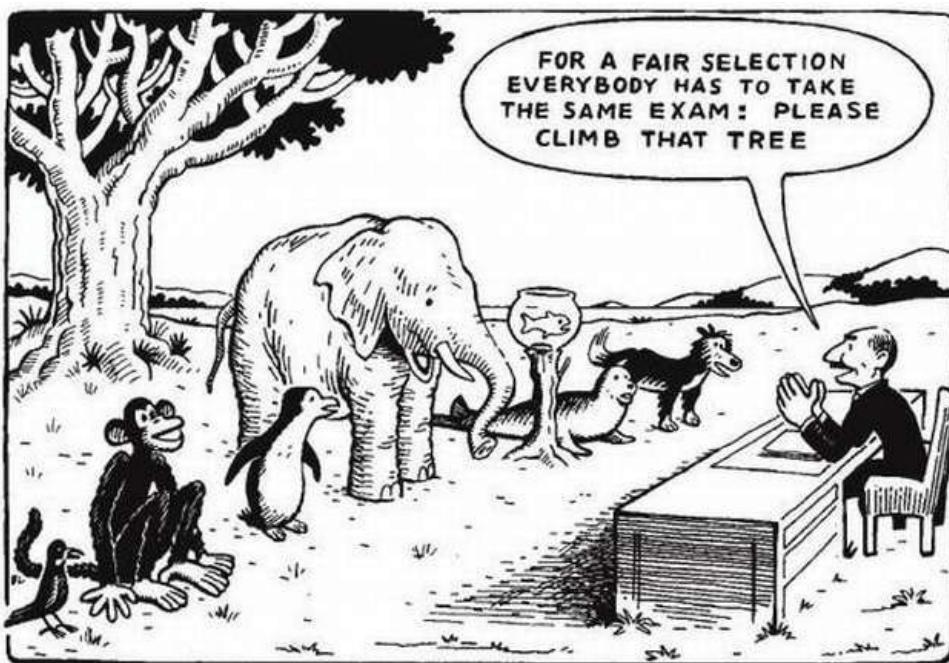
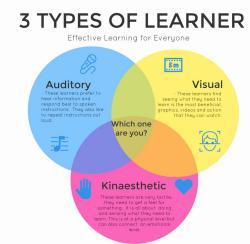
SD-14

# 01

## Problem Being Addressed

# Problem

- Traditional and online learning use a **one-size-fits-all approach** that ignores different learning styles.
- **Lack of personalization** leads to disengagement, poor retention, and frustration.
- Students need a **learning system that adapts** to their unique preferences and challenges.



# 02

## Issues

### Issue 1 - Personalized Content Generation

6

#### **Issue:**

- Traditional learning materials (PDFs, videos, lectures) do not dynamically adjust to individual needs.
- Manually personalizing content for each learner is time-consuming and unrealistic at scale.

#### **Solution:**

- Use **AI that adapts content** (prompt engineering + RAG) to transform learning materials:
  - Convert text-based materials (ebooks, PDFs) into customized learning modules.
  - Generate interactive learning modules (quizzes, flashcards, video explanations).
  - Personalize examples and quiz questions based on the user's interests

## Issue 2 - AI Accuracy & Reliability

### Issue:

- AI-generated content must be accurate, relevant, and aligned with the subject matter

### Solution:

- RAG (Retrieval-Augmented Generation) AI approach:**

- AI retrieves verified sources before generating responses
- Real-time validation using educational databases and reference textbooks.
- Continuous human feedback and user reporting to improve AI accuracy.

The RAG process



## Issue 3 - Scalability & Performance

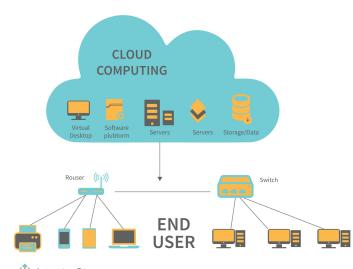
### Issue:

- Processing large amounts of data (ebooks, user inputs, learning history) requires efficient computing power.

### Solution:

- Cloud-based architecture** to handle data and scale efficiently.

- As our program gets larger with more educational dashboards the cloud can store that data on demand
- Allows for secure storage and traffic management to improve performance



# 03

## Tools and Practices

### Tools

10

Frontend Development:



PostgreSQL Database:



Hosting and Deployment:



Firebase

Containerization:



# Practices

11

Practices	Description
Agile Development	We follow an iterative and flexible development process (sprint plan), making adjustments based on feedback and testing.
CI/CD	Automated deployment and testing using Vercel & Docker, ensuring stable and frequent updates.
Learning Persona	Understanding how each user learns best; based on performance and feedback.
Prompt Engineering	Used to refine AI-generated quizzes, explanations, and personalized content formats.
RAG	Used to process ebooks and generate personalized content with accurate, structured learning paths (same Info-different styles).

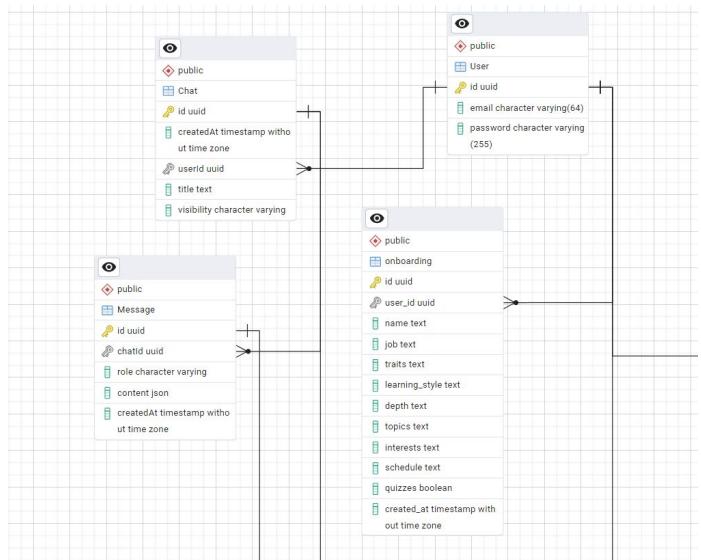
# 04

## Current Progress and Challenges

# Current Progress (Database)

13

- **Operational Tables:**
  - User
  - Chat
  - Message
  - Onboarding
- API is set up to successfully handle data posting and retrieval, ensuring seamless communication between the frontend and backend.  
(Authentication, chats, persona)



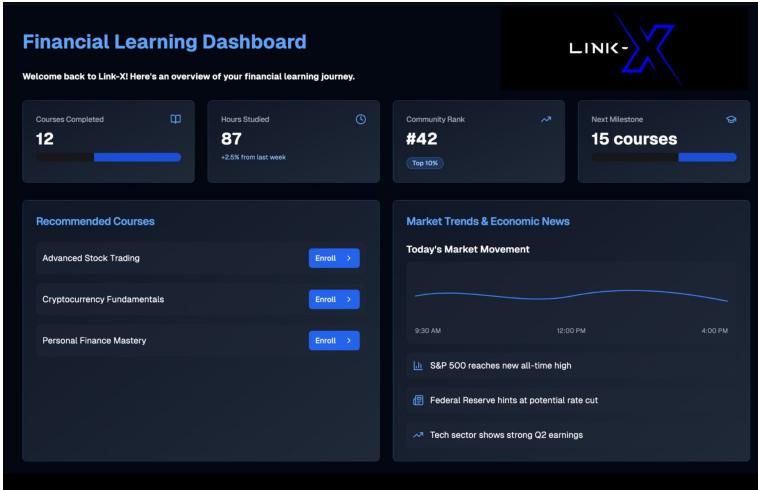
# Current Progress (UI)

14

- **Major Pages Implemented:** We have successfully developed four of our key pages.
- **Standardized Design:** Our goal is to align these pages with industry best practices for usability and aesthetics.
- **Client Collaboration:** We are actively working with the client to refine the design based on their vision and feedback.
- **Design Tools:** We are leveraging V0 by Vercel to streamline the design process and ensure consistency.



# UI Components



User Dashboard

**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

Depth of Explanation

Topics of Interest  
 e.g., Investing, Finance

Interests, Values, or Preferences for Personalization  
 e.g., Basketball, Video Games

Preferred Study Schedule

Include quizzes for progress tracking

User Onboarding

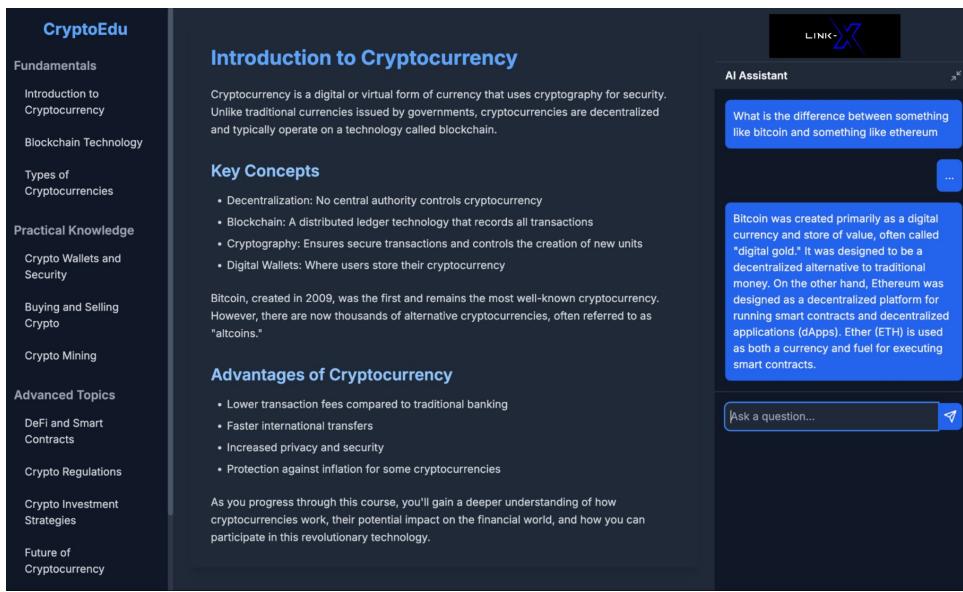
# UI Components



Modular Navbar Component

# Course Screen Sketch

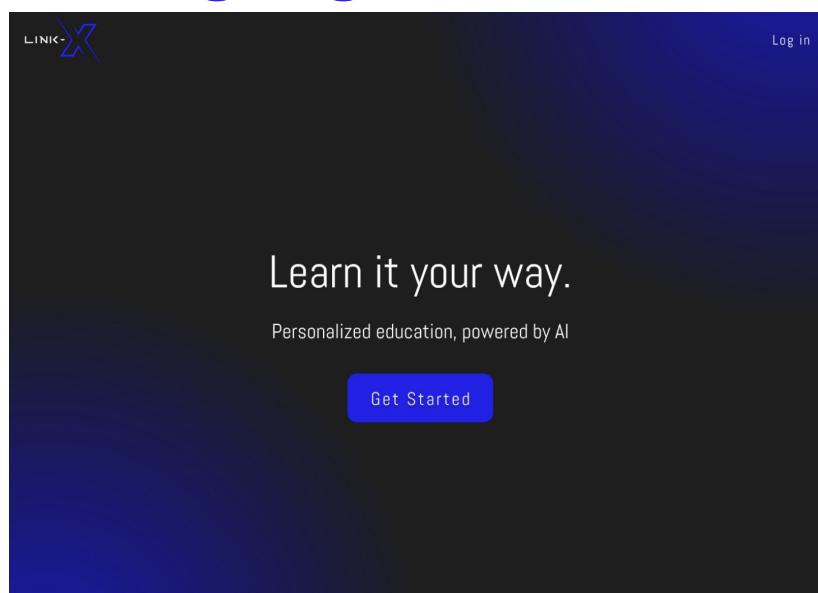
17



The course screen sketch is a dark-themed interface for a crypto education platform. On the left, a sidebar titled "CryptoEdu" lists categories: Fundamentals (Introduction to Cryptocurrency, Blockchain Technology), Practical Knowledge (Crypto Wallets and Security, Buying and Selling Crypto, Crypto Mining), and Advanced Topics (DeFi and Smart Contracts, Crypto Regulations, Crypto Investment Strategies, Future of Cryptocurrency). The main content area features a title "Introduction to Cryptocurrency" with a sub-section "Key Concepts" containing bullet points about decentralization, blockchain, cryptography, and digital wallets. Below this is a section titled "Advantages of Cryptocurrency" with a list of benefits like lower fees and faster transfers. A sidebar on the right titled "AI Assistant" shows a question "What is the difference between something like bitcoin and something like ethereum" and a detailed answer about the differences between Bitcoin and Ethereum. At the bottom right is a "Get Started" button.

# Landing Page Screen Sketch

18



The landing page screen sketch has a dark blue gradient background. At the top left is the "LINK-X" logo. At the top right is a "Log in" button. In the center, the tagline "Learn it your way." is displayed above the sub-tagline "Personalized education, powered by AI". At the bottom center is a large blue "Get Started" button.

# Challenges (Codebase & Design)

- **Adapting to the Existing Codebase:** Understanding the structure, dependencies, and best practices for working within it.
- **Running the Backend Properly:** Initial setup issues, API key integration for OpenAI, and backend filing confusions.
- **Refactoring & Optimization:** Cleaning up and restructuring code to improve maintainability and efficiency.
- **Website Design:** Working with our clients to design a professional website that exceeds their expectations.



# Challenges (AI & Prompt Engineering)<sup>20</sup>

- **Limited Experience in ML & Prompt Engineering:** Learning how to optimize prompts and integrate AI effectively.
- **Configuration Hurdles:** Understanding how to fine-tune responses and manage OpenAI's API parameters.
- **Ongoing Research & Experimentation:** Finding the best strategies to implement prompt engineering within our system.



# 05

## Remaining Work Timeline

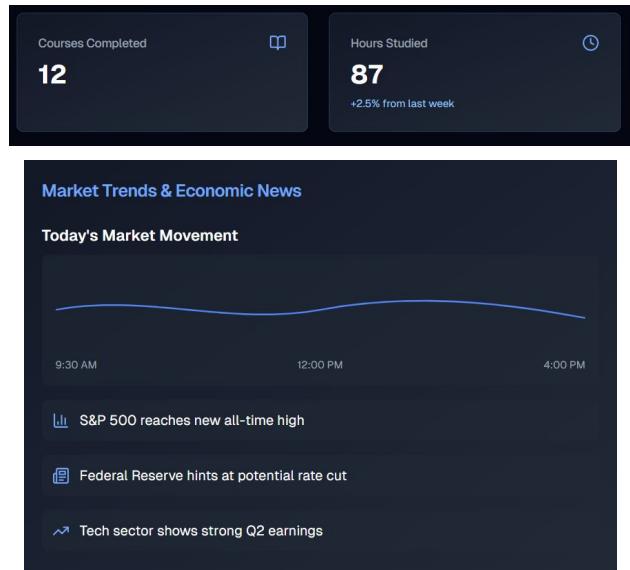
### Sprint Plan

22

Week Starting	Sprint Plan	
2/2/2025	Onboarding and understanding code base	
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	UI & Onboarding Complete
3/30/2025	Testing Personalization & Backend	
4/6/2025	Make Database Backend to save personalization progress	
4/13/2025	Testing RAG for Uploading pdfs	Have Working Personalizaton for any topic
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: Dashboard & Study Page<sup>23</sup>

- **Enhancing Dashboard Functionality:** Improve user experience by adding interactive elements, progress bars, and dynamic data displays.
- **Expanding Study Page Features:** Implement better organization, filtering options, and personalized study recommendations.
- **Seamless Integration:** Ensure all pages work cohesively with existing components for a smooth workflow.



# Future Plans: Universal Styling

24

- **Consistent UI/UX:** Apply a uniform design system across all pages for a professional and cohesive look.
- **Theme & Responsiveness:** Ensure accessibility across different devices and screen sizes.
- **Efficiency & Maintainability:** Use reusable styling components to simplify future development.



# Future Plans: Prompt Engineering for<sup>25</sup> Personas

- **Persona-Specific Responses:** Fine-tune AI behavior to align with different user needs and expectations.
- **Adaptive Learning:** Improve model accuracy based on user interactions and feedback.
- **Optimized API Usage:** Experiment with various prompt techniques to enhance response quality and efficiency.



# 06

## Demo

# THANK YOU



# AI-Driven Personalized Learning Platform

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

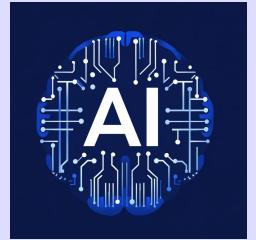


SD-14

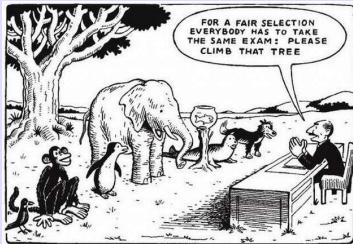
# 01

Topic

## Topic



- 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

The screenshot shows the homepage of the Link-X platform. At the top, there's a dark header with a navigation menu including 'Home', 'Learn', 'Market', 'Courses', 'News', 'Portfolio', 'Wallet', and 'History'. Below the header, a large banner features the text 'Learn it your way.' in a bold, white, sans-serif font. Underneath the banner, a subtext reads: 'Traditional learning doesn't work for everyone. Our AI-powered platform personalizes financial education to match your unique learning style.' Below this, there are three dark rectangular cards with white icons and text: 'Personalized Learning' (AI adapts to your unique learning style), 'Certification' (Earn recognized certificates to showcase your financial expertise), and 'Cloud Integration' (Access your learning materials anywhere, anytime with cloud storage). At the bottom of the page are two buttons: 'Go to Dashboard' and 'How It Works'.



The screenshot shows a 'How It Works' section titled 'Your Personal Financial Learning Journey'. It explains how AI transforms financial e-books into personalized learning experiences. A list of features includes: Personalized syllabus from any financial e-book, Interactive quizzes that adapt to your knowledge, Progress tracking with detailed analytics, and Downloadable certificates upon completion. There is also a 'View More' button.

# Solution (Quick Overview)

6

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

This screenshot shows a 'Personalized Learning Setup' form. It includes several input fields:

- 'What should Link-X call you?' with a text input field.
- 'What do you do?' with a text input field containing 'e.g., Student, Engineer'.
- 'What traits should Link-X have?' with a text input field containing 'e.g., witty, encouraging'.
- 'Preferred Learning Style' with a dropdown menu labeled 'Select a learning style'.
- 'Depth of Explanation' with a dropdown menu labeled 'Select depth'.
- 'Topics of Interest' with a text input field containing '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)
```

The screenshot shows a dark-themed dashboard with a sidebar on the left containing navigation links like Home, Learn, Market, Courses, News, Portfolio, Wallet, History, Notifications, Settings, Sign out, and a user profile for Alex Johnson (Pro Member). The main area displays a summary of the user's progress: 12 courses completed, 87 hours studied, #42 in the community rank, and 15 courses as the next milestone. Below this, there's a search bar asking "What would you like to learn?" followed by a "Search courses..." input field. Under "Courses and Topics", there are two sections: "Advanced Stock Trading" and "Cryptocurrency Fundamentals", each with a "Learn" button.

# 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

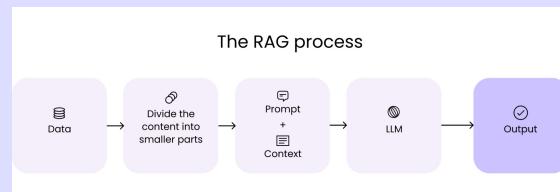
# RAG

## 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"

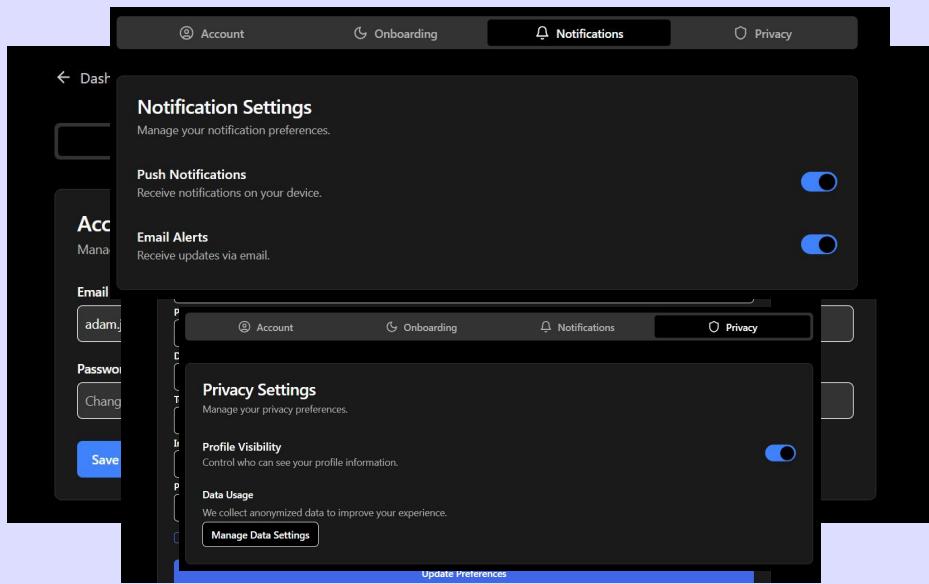
# Prompts Put Together (Frontend)

The screenshot shows a mobile application interface for an AI Assistant. At the top, there's a header bar with a back arrow and the title "AI Assistant". Below it is a blue input field containing the text "Tell me a little more about AI perception". The main area is a dark-themed card with white text. It starts with a heading "Definition of art" and a sub-section "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!". This is followed by a detailed explanation of AI perception, mentioning computer vision, natural language processing, and other technologies. The text continues with examples of learning and reasoning, and ends with a note about self-correction. On the right side of the card, there's a sidebar with the text "I Intelligence? Let's revolutionizing how we ic human intelligence reasoning (using AI systems improve algorithms that allow its actions to s for decision-making and make decisions.)". At the bottom right of the card, there are two buttons: a green one labeled "+ Upload PDF" and a blue one labeled "Learn". A red arrow points from the left margin of the main content area towards the first section of the AI perception text.

# Final Learning Page

The screenshot shows a final learning page with a sidebar on the left and a main content area on the right. The sidebar contains a navigation menu with sections like "INTRODUCTION TO ARTIFICIAL INTELLIGENCE", "TYPES OF ARTIFICIAL INTELLIGENCE", "MACHINE LEARNING BASICS", and "DEEP LEARNING EXPLAINED". The main content area has a heading "Definition of artificial intelligence (AI)". Below it is a section titled "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!". This is followed by a detailed explanation of AI perception, mentioning computer vision, natural language processing, and other technologies. The text continues with examples of learning and reasoning, and ends with a note about self-correction. On the right side of the main content area, there's a dark-themed card with white text, identical to the one in the AI Assistant screenshot above. At the bottom right of the main content area, there's a text input field with the placeholder "Ask a question..." and a blue send button with a white arrow icon.

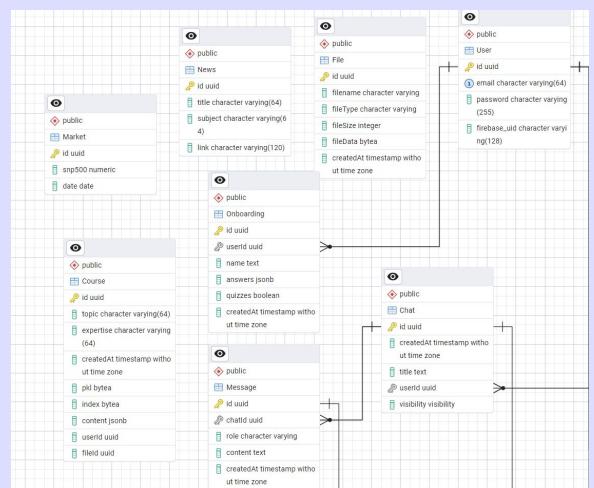
# Settings Page



# Database Current Progress

16

Identifier	Providers	Created	Signed In	User UID
morgan4@gmail.com	✉️	Apr 14, 2025	Apr 14, 2025	i6zIHuJGfkgeIYYGNtFGJabN...
morgan5@gmail.com	✉️	Apr 14, 2025	Apr 14, 2025	QzqGwLGFVLvGuF0TUR02db...
test2@iastate.edu	✉️	Apr 14, 2025	Apr 14, 2025	ttVvwsRHnCYRXEDoiMUnknj...
practice@iastate.edu	✉️	Apr 14, 2025	Apr 14, 2025	AiCTLIQ4u4VfNMu3mbkG0S5...
adamtest@iastate.edu	✉️	Apr 14, 2025	Apr 14, 2025	M4AiOq7dPjgXZUwnbWQAH...
test@iastate.edu	✉️	Apr 14, 2025	Apr 14, 2025	eSCNFBv6Z0gh0Znj7MPMF4...
a@iastate.edu	✉️	Apr 14, 2025	Apr 14, 2025	JnglSG6ZXcLN0tJRFV59rsH...
adamhisel12@yahoo.c...	✉️	Apr 14, 2025	Apr 14, 2025	mjV6sLpJHTaKuSYMjRP4cpcl...
carterp@iastate.edu	✉️	Apr 13, 2025	Apr 14, 2025	rJxr50IDqeew9c1b5v5NIVEQp...
demo@iastate.edu	✉️	Apr 13, 2025	Apr 14, 2025	GaduYsgGfPZPanDfmkUHAQ...
carter.p@iastate.edu	✉️	Apr 13, 2025	Apr 13, 2025	rXat1AvoxSZLyJnuy3y5UN0g...
ahisel@iastate.edu	✉️	Apr 13, 2025	Apr 13, 2025	rpgugqlLfMzgqf3ZuBQADS9NY...
parks@iastate.edu	✉️	Apr 13, 2025	Apr 13, 2025	Ay5rdvU7lyzdPiY0nmhVEar0...



```

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

    # 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=outline, # Course outline (JSON)
        pk=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

```

# 03

## Challenges

# Team Challenges

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

```
morgan@DESKTOP-6JU4QO: ~\OneDrive\Documents\Senior Design\LINK-X (ryan)
$ bash run backend.sh
Step 1: Building Backend...
[+] Building 246.8s (16/17)
--> [internal] load build definition from dockerfile
--> [internal] load .dockerignore
--> [internal] load metadata for docker.io/library/python:3.10-slim
--> [internal] load dockerfile
-->> transferring context: 2B
--> [ 1/12] FROM docker.io/library/python:3.10-slim@sha256:65c8436530d8a3ba27c8d5081a02f4daef776074f0ff7f7c6bf8ce4e931ac96
-->> resolve docker.io/library/python:3.10-slim@sha256:65c8436530d8a3ba27c8d5081a02f4daef776074f0ff7f7c6bf8ce4e931ac96
--> [internal] load build context
-->> transferring context: 13.41MB
--> [ 2/12] WORKDIR /app
--> [ 3/12] COPY src/requirements.txt .
--> [ 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.firebaseio.json .
--> [ 8/12] COPY src/index.faiiss /app/index.faiiss
--> [ 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 --debug to see all)

docker:desktop-linux
```

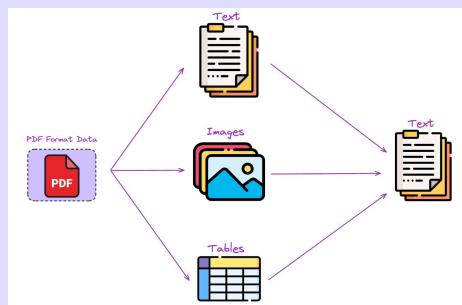
04  
Remaining Work Timeline

# Sprint Plan

Week Starting	Sprint Plan	
2/2/2025	Onboarding and understanding code base	
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	UI & Onboarding Complete
3/30/2025	Testing Personalization & Backend	
4/6/2025	Make Database Backend to save personalization progress	
4/13/2025	Testing RAG for Uploading pdfs	Have Working Personalization for any topic
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

- **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

