# Team HAP

# Michigan State University

Artificial Intelligence Training Course

Project Plan

Spring 2024


**HAP Sponsors**

Angela Endres

Steve Neubecker

**Michigan State Capstone Team**

Ashley Arciniega

Joey Morrison

Advait Paliwal

Caleb Story

Vetri Vijay

# Table of Contents

## Executive Summary

Health Alliance Plan (HAP) is a pioneering Michigan-based non-profit health insurance provider and subsidiary of the esteemed Henry Ford Health Systems. With a dedicated workforce of 1,100 employees serving a vast community of 430,000 insured members, HAP stands as a beacon of personalized care and customer-centric service. At HAP, the commitment goes beyond just providing insurance; it's about facilitating a seamless healthcare journey for every member. With 90% of premium revenue channeled directly into funding healthcare services, HAP sets the gold standard for utilizing resources to benefit its members. The mission is clear: to make the complex health care system easy for members to use.

HAP recognizes the imperative of remaining at the forefront of technological advancement to enhance customer value. Considering the transformative potential of AI in optimizing operational efficiencies, HAP integrates this cutting-edge technology into its workforce. In a world where AI often evokes apprehension due to misconceptions propagated by media narratives, HAP envisions a future where AI serves as a catalyst for positive change, enhancing productivity and efficiency without displacing jobs.

To actualize this vision, HAP proposes the implementation of a comprehensive AI training course tailored specifically for its workforce. This initiative bridges the gap between perception and reality, equipping employees with a nuanced understanding of AI's real-world applications. This course offers concise, user-friendly content presented through an accessible interface.

Central to the training program's effectiveness is the integration of AI chatbot technology and interactive avatars, providing a dynamic learning experience that fosters engagement and comprehension. Through hands-on interaction with AI-driven tools, employees gain practical insights into how AI can augment existing workflows, streamlining processes and enhancing decision-making Capabilities.

By championing AI literacy within its workforce, HAP not only future-proofs its operations but also cultivates a culture of innovation and adaptability. With each employee empowered to harness the potential of AI, HAP is primed to redefine the standards of excellence in healthcare provision, setting a precedent for industry-wide innovation and customer-centricity.

## Functional Specifications

HAP is looking to teach managerial staff an overview of what AI is and what it can be used for. They believe that a web application is the best way to do this. This web app is a training course that takes roughly 30 minutes to complete and covers concepts such as LLM basics, vector databases, machine learning/neural network basics, AI applications, and popular LLMs. This content is at a complexity where anyone of any technical knowledge can understand it and is presented to the user through an AI-generated avatar.

This application is used to help employees at HAP understand what AI is and how it can be used by them. Many of the employees at HAP do not have a technical background, so all the content is presented in a way such that anyone of any level of technical knowledge can follow along and understand the course. This course is also intended to be short. Since we are only looking for a surface level of understanding, the entire course takes roughly 30 minutes to complete. The main objectives of this course are for users to understand AI basics at a high level, interact with an AI-generated avatar with voice, and ensure the course only takes 30 minutes.

After taking this course, the user not only is able to use AI to increase productivity but also uses it as a tool to solve everyday problems. Generative AI can be used to create documents and email templates so the user can focus on the content they want to fill it with. LLMs like ChatGPT can also be interacted with by the user to ask questions and receive more tailored feedback when compared to regular internet searches. Users can ask questions about new technologies they're learning and pick up on the context as opposed to an internet browser that takes all searches at face value.

Content is presented in many test formats as well as through an AI avatar "professor". The avatar can be interacted with by the user to ask questions about the course content and get individualized feedback. The user learns the content to answer a series of questions to test their retention of the content.

# Design Specifications

## Overview

HAP's AI training course is inclusive and available to all employees. Equipped with the same comprehensive features, including a structured curriculum comprising five modules, five mini-quizzes, a conclusive final quiz, a dedicated resource page connecting to external materials, a games page, and a news page. Designed with the aim of enhancing the understanding of artificial intelligence among non-technical staff, this web-application is optimized for compatibility with computer and mobile screens.

## Dashboard

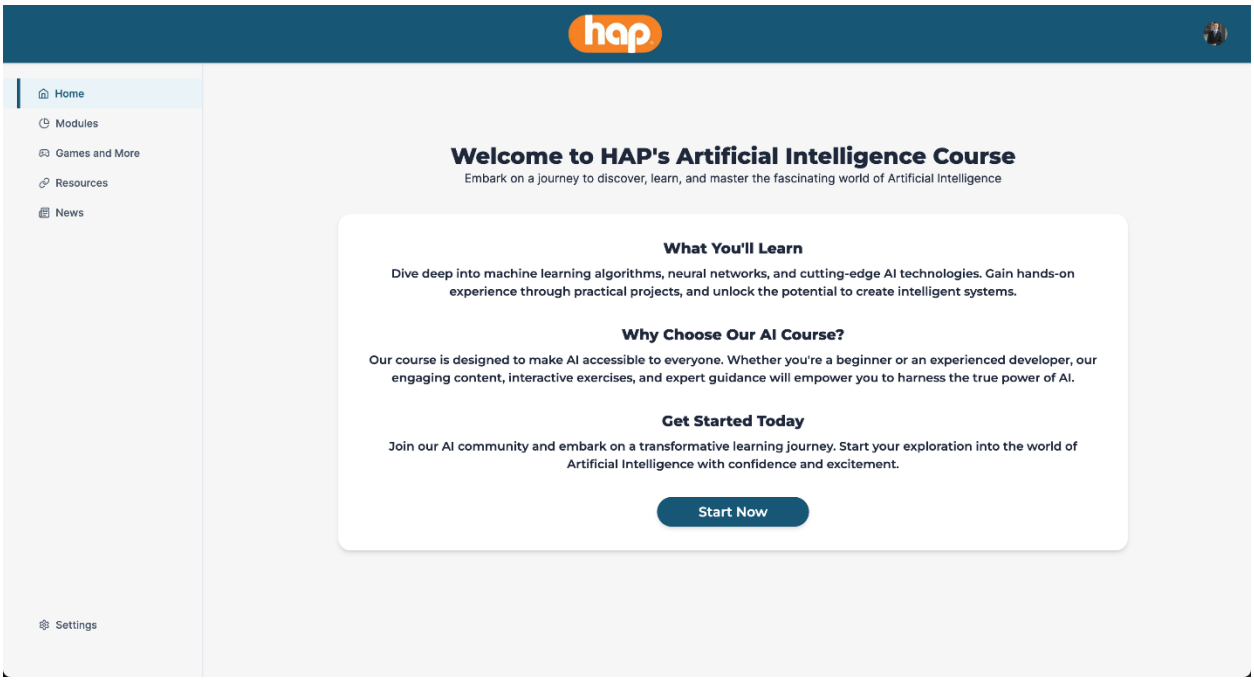


Figure 1 : Dashboard Mockup

The dashboard serves as our primary landing page, providing an overview of available modules and your progress within each. The sidebar navigation bar offers users the flexibility to access individual module descriptions, quizzes, or supplementary external resources.

## Modules

Each module within HAP's AI training course is thoughtfully organized into digestible chapters, aimed at facilitating effective learning. Within each chapter, users encounter detailed information supplemented with common questions strategically placed to

reinforce understanding. Additionally, to foster engagement and support comprehension, every module incorporates a dedicated chat section where users can seek clarification or discuss concepts further. To gauge comprehension and retention, a mini quiz is included at the end of each module, providing users with an opportunity to assess their learning outcomes. This structured approach ensures that learners can progress systematically through the course content while actively engaging with the material at every stage.

The figures shown below are screen mockups for the web-application. Figure 2, shown below, shows the introductory page for Module 1.

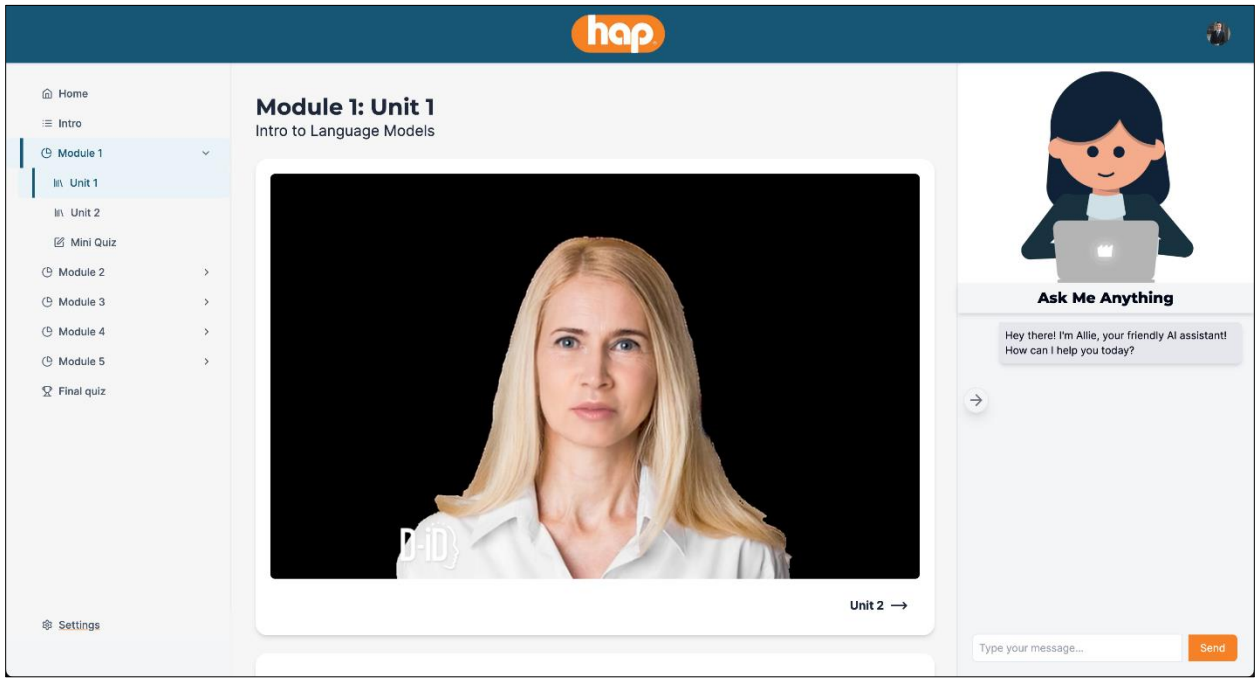


Figure 2 : Module

Figure 3, shown below, shows an example of what a chapter looks like. Lastly, figure 4, also shown below, shows an example of what a module quiz looks like. The module mini quizzes are comprehensive consists of ten questions focusing on just the module it is on. The final quiz is cumulative, focusing on the entire course.

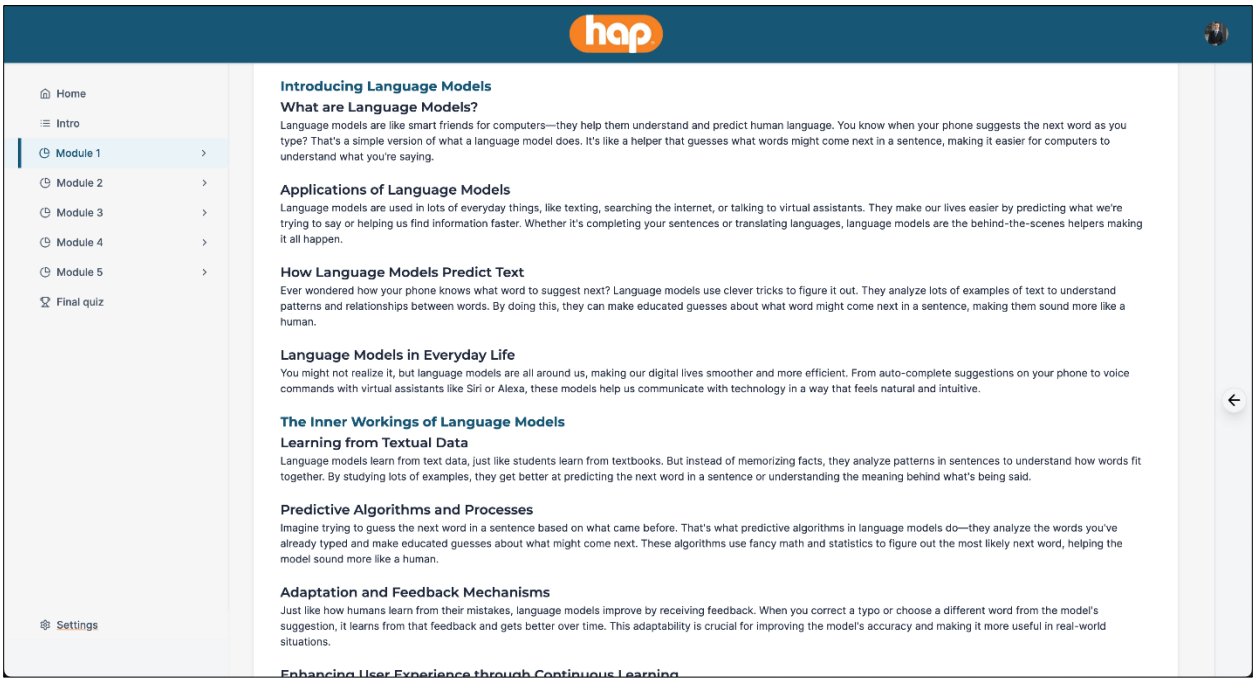


Figure 3: Chapter

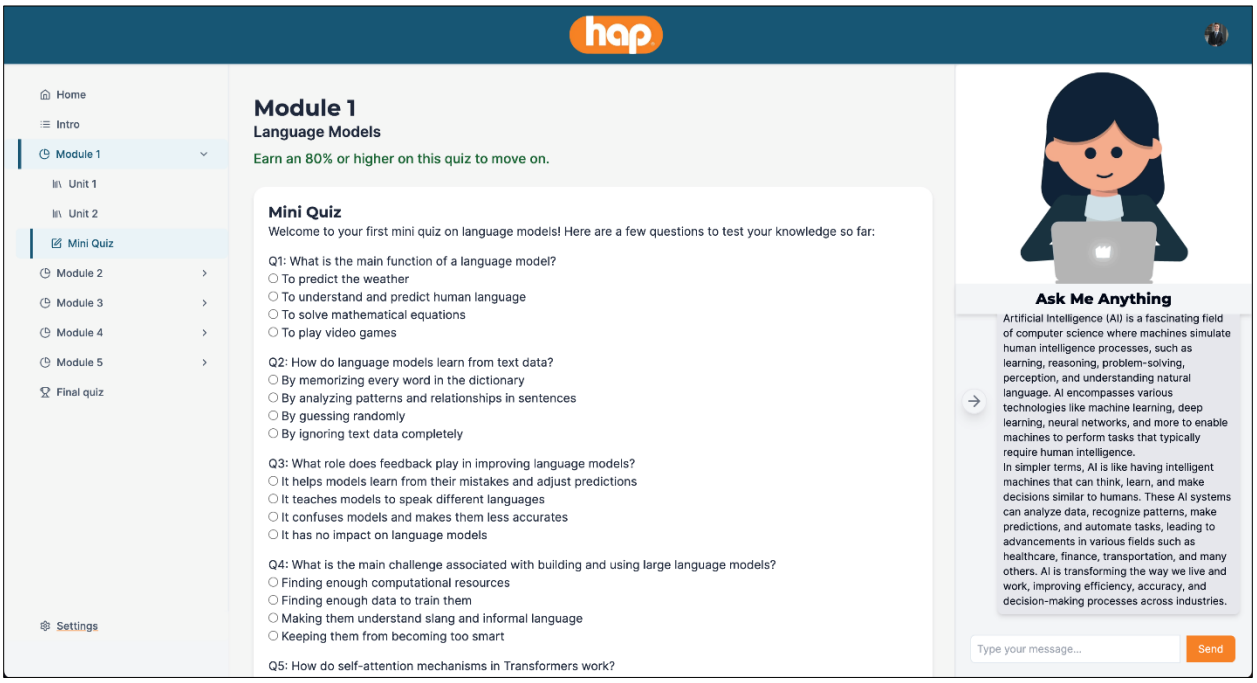


Figure 4:  Quiz

## Chat Integration

The chat feature is intended to give users hands-on experience of what it's like interacting with artificial intelligence. Here, you can interact with our AI bot Allie. She can answer any of your questions on artificial intelligence. Figure 5, shown below, shows the text chat feature.

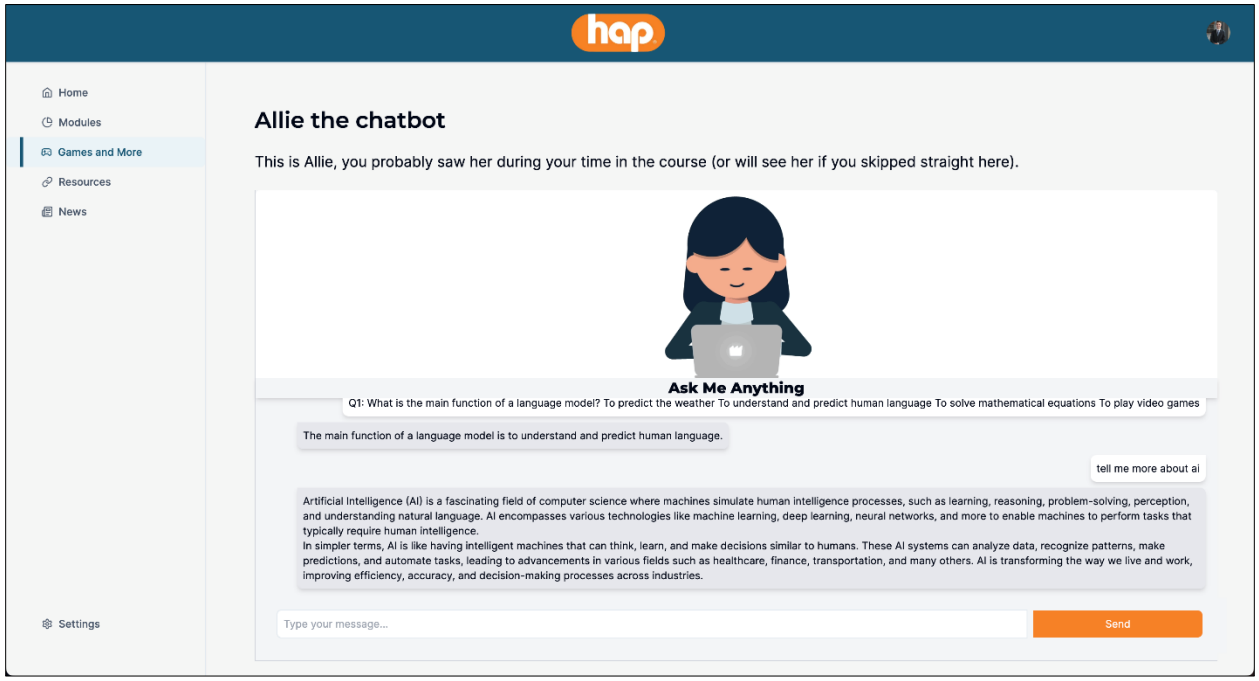


Figure 5: Chat

## Interactive Sections

In addition to the core learning course, there are some interactive sections that can be used by the employees. One of which is the Games and More. This section allows for the user to further gain an understanding of AI through games and activities. The user has the choice of playing a game of Hangman or Memory with AI related terms, taking an interactive tutorial of the AI chatbot, and using an object detection model on the user's uploaded images as shown in Figure 6.

Another section that we have in addition to the learning course is AI news. In this section, the user is able to see all the most recent news relating to artificial intelligence. The user is able to search for specific terms to further filter the responses. Figure 7 shows the news page

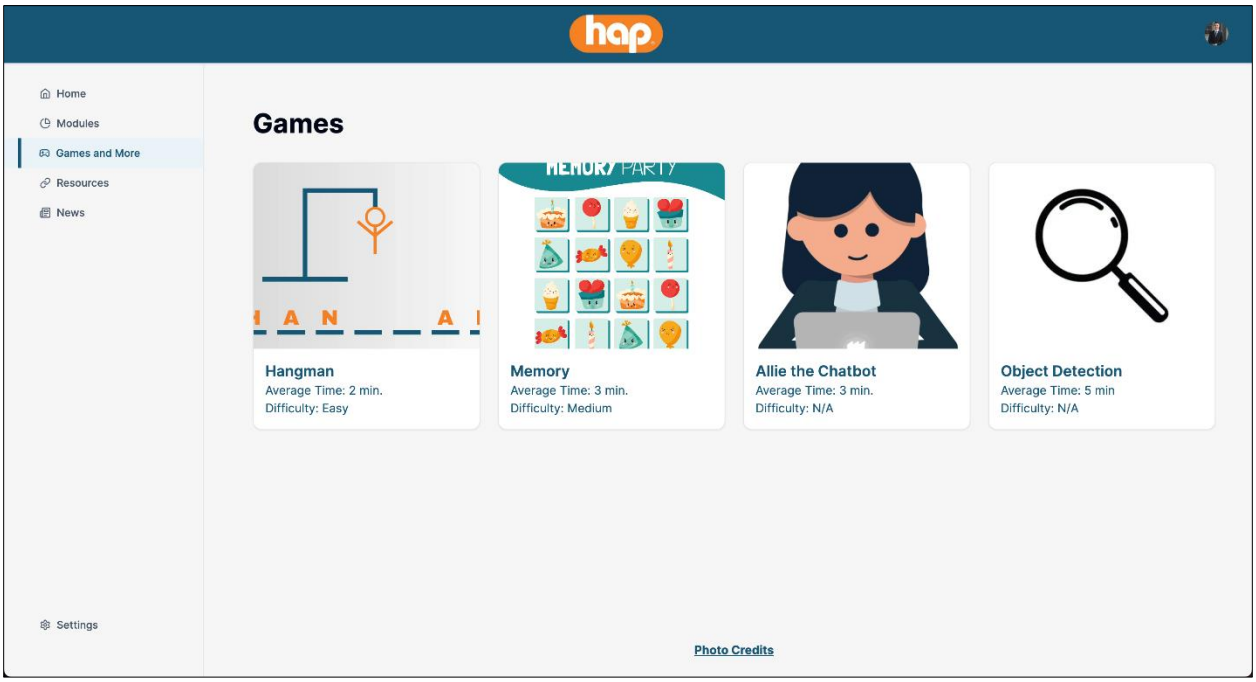


Figure 6: Games

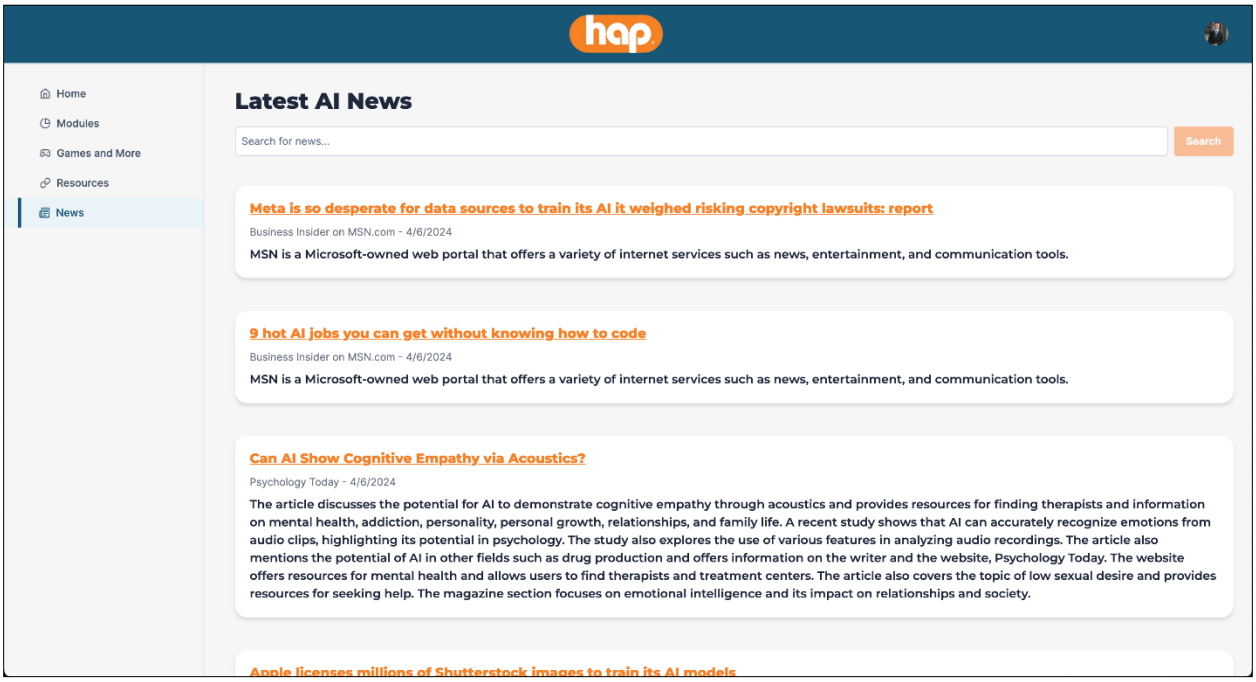


Figure 7: News

## Technical Specifications

### System Architecture

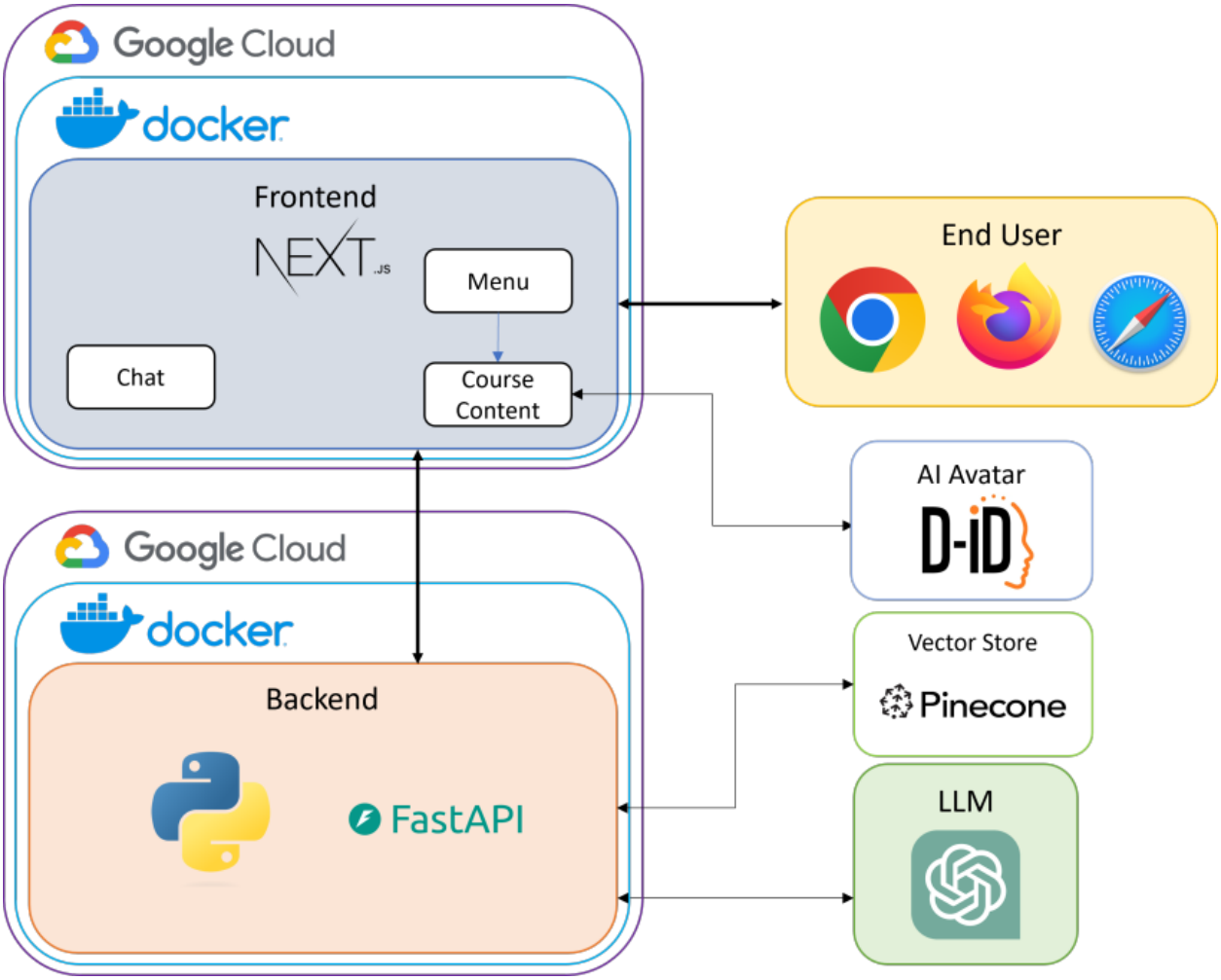


Figure 7: System Architecture

## System Components

### Software

Docker

- Containerization: Used for packaging the application and its dependencies into containers. Ensures consistency across multiple development, testing, and production environments, facilitating easy deployment and scaling.

FastAPI

- API Interface: Utilized to create a high-performance, asynchronous API interface. Enables the development of RESTful APIs with automatic interactive documentation, validation, and serialization, enhancing the efficiency and reliability of the system's backend services.

Google Cloud Platform

- Cloud Run: Serves as the primary compute platform for the HAP project, offering fully managed, serverless, and event-driven capabilities. It enables the deployment of containerized applications, ensuring scalability and seamless integration with other GCP services.
- Secrets Manager: Used to securely store and manage sensitive information such as API keys, ensuring that these secrets are encrypted, access-controlled, and auditable.

MongoDB

- Database: Serves as the primary database for the project, offering a flexible and scalable NoSQL database solution. It is used to store and manage structured and unstructured data efficiently.

Next.JS

- Web Framework: Employed as the web framework for the project, providing a robust platform for building server-rendered React applications. Ensures fast page loads, automatic code splitting, and streamlined data fetching, enhancing the user experience of the web application.

OpenAI

- GPT API: Integrated to provide advanced natural language processing capabilities. Enables the system to understand, generate, and interact with human-like text, forming the core of the HAP's conversational AI.

Pinecone

- Vector Store: Acts as the vector store for the project, providing a scalable and efficient solution for storing and querying large-scale vector embeddings. This is crucial for maintaining the memory and context of GPT interactions, enhancing the overall performance and relevance of the responses.

shadcn/ui

- UI Components: An open-source UI component library integrated to accelerate the development of the user interface. Provides a consistent and modern look and feel across the application, improving user engagement and satisfaction.

### Development Environments

GitLab

- Version Control: Serves as the version control system for the project, offering a single application for the entire DevOps lifecycle. Facilitates collaborative coding, continuous integration/continuous deployment (CI/CD), issue tracking, and code review, ensuring efficient and high-quality software development.

Trello

- Task Management: Serves as a web-based project management tool that uses a card-based system for organizing tasks and projects. It's designed to visually represent workflows, facilitating the ability to manage tasks, collaborate, and track progress.

Visual Studio Code

- Prototyping: Serves as a collaborative interface design tool used for creating user interfaces, web pages, and app prototypes. It operates entirely in the browser, allowing multiple users to work on the same design simultaneously, fostering real-time collaboration and feedback.

## Risk Analysis

### Avatar Cost

Difficulty: Medium

Description: The potential lack of a dedicated budget for the avatar is a significant concern due to the high costs usually associated with avatar services.

Mitigation: Interaction during Q&A sessions primarily utilize chat responses instead of more resource-intensive avatar responses.

### Technical Limitations of AI and LLMs

Difficulty: High

Description: Large Language Models (LLMs) might not always produce accurate or contextually appropriate responses, a crucial factor for the integrity of the service.

Mitigation: Continuous updates and training with pertinent data is implemented to enhance the AI model's accuracy and contextual relevance.

### Latency of Product

Difficulty: Medium

Description: Latency in response generation could adversely affect user experience, as swift and seamless interaction is anticipated in dynamic AI-driven interfaces.

Mitigation: Caching of the course content and the avatar is implemented to improve response times and overall user interaction.

### Content Relevance

Difficulty: High

Description: The rapid advancement in the field of AI presents a risk of course content becoming quickly outdated, with ongoing project involvement for updates potentially unfeasible.

Mitigation: Scheduled updates and the utilization of AI's adaptability in course material development is employed to maintain content relevancy and timeliness.

# Schedule

### Week 1 – 4: Jan 8 – Jan 29

- Develop front end and back end skeleton
- Choose Avatar and begin development
- Created rough draft of UI design

### Week 4: Jan 29 - Feb 2

- Connect all of the individual pieces (frontend, backend, cloud, and avatar)
- Have all the course content finalized and approved
- Finish fully functional prototype on Figma

### Week 5: Feb 5 - Feb 9

- Debug until the individual parts of the project communicate effectively.
- Develop the dashboard page
- Begin avatar integration

### Week 6: Feb 12 - Feb 16

- Make the backend call the OpenAI API and display on frontend
- Develop course modules
- Continue avatar integration

### Week 7: Feb 19 - Feb 23

- Link the avatar to the course content
- Begin the chat functionality

### Week 8: Feb 26 - Mar 1

- Finish the chat functionality
- Develop the final quiz site
- Add extra materials to resource page

### Week 9: Mar 4 - Mar 8

- Ensure that the UI/UX is smooth and consistent for the user
- Ensure all content is accurate and all quizzes are functional
- Begin games page

### Week 10: Mar 11 - Mar 15

- Finish the avatar's functionality with the frontend and course content
- Make sure that all aspects of the course are linked and communicate together.
- Add more games

### Week 11: Mar 18 - Mar 22

- Add final features to the avatar, chatbot, and course content.
- Polish the UX
- Create news page

- Improve games page

### Week 12: Mar 25 - Mar 29

- Continue debugging
- Wrap up stretch goals
- Improve news page, add scheduler

### Week 13: Apr 1 - Apr 5

- Beta Presentation
- Bug fixes
- Finalize stretch goals

### Week 13: Apr 8 - Apr 12

- Bug fixes

### Week 14: Apr 15 - Apr 19

- Bug fixes