

Using Agentic Workflows to Build AI-Integrated Assignments for Introductory Psychology

CS 461 Capstone I

Soham Das - 934460103

Padraic James Bergin - 933967013

Aaron Rasmussen - 934486920

Elliott Shim - 933969578

Activities: All activity write-ups, both teamwork related and individually completed, will be at the bottom of the document.

Aaron Rasmussen -

- Planning Activity: Impact-Effort Matrix (team activity)
- Requirements Activity: User Story Mapping (team activity)
- Creative Activity: Literature Review (individual activity)
- Personal Statement: I am responsible for creating the website/interface the users will be interacting with. I will be focusing on getting the application to function using Flask. I have very little web development experience outside of OSU coursework, so I am looking forward to learning new technologies and applying them into the real world. I feel like although the bulk of the project is focused on AI, having the opportunity to work on the front end of the project will give me a unique experience relative to my group mates. As the remainder of the group will have their hands full with the back end, I will appreciate being able to see and clearly point out my contributions to the project.

Soham Das -

- **Planning Activity:** Impact-Effort Matrix (team activity)
- **Requirements Activity:** User Story Mapping (team activity)
- **Creative Activity:** Literature Review (individual activity)
- **Personal Statement:** I'm going to be responsible for leading the team through implementing multi-agent architectures for our use case. I'd love to continue doing extensive research and implementation of utilizing the graph based workflow to find creative and efficient solutions to our problem. I have always wanted to lead a team of students through a practical application of our skills, and this is the perfect opportunity to do so. I will focus on conducting thorough research in order to create a balance between strategic oversight and hands-on involvement.

Padraic Bergin -

- Planning Activity: Impact-Effort Matrix (team activity)
- Requirements Activity: User Story Mapping (team activity)
- Creative Activity: Literature Review (individual activity)
- Personal Statement: I have found interest in developing the grading agent for our program. This task will allow me to work with prompt engineering to formulate an optimal communication method for OpenAI's LLM. This feature of the program is vital in tying all of the components together to get a response from ChatGPT. The graded response will need further processing to form the logic of our interactive tutor.

Elliott Shim -

- Planning Activity: Impact-Effort Matrix (team activity)
- Requirements Activity: User Story Mapping (team activity)
- Creative Activity: Literature Review
- Personal Statement: I am going to be responsible for the database, including initialization and management of it. I will also be responsible for the initial translation of the data from the textbook into useable data for the RAG application. This includes creating a vector-database as well as developing tools that allow for the accessing of that data through the use of LangChain agents. I will make sure that the system is both intuitive and reliable.

Product Requirements Document

Problem Description

Traditional classroom environments often struggle to deliver immediate, personalized feedback, which is a critical component for effective learning. In these settings, instructors might find it challenging to address individual student needs promptly, leading to a one-size-fits-all approach that can overlook diverse learning styles and paces. This lack of tailored feedback can result in decreased student engagement and higher failure rates. Research indicates that in conventional lecture-based courses, approximately 33.8% of students fail, whereas active learning environments, which emphasize student participation and immediate feedback, see a reduced failure rate of about 21.8% [1]. The disparity underscores the importance of personalized feedback in enhancing student performance and retention. To address this issue, we will be integrating AI-driven tutoring systems that are capable of providing real-time, individualized feedback that could bridge the gap left by traditional methods, which will yield a more responsive and effective learning experience for students.

Scope

Our project focuses on creating an AI-driven learning assistant that leverages Retrieval-Augmented Generation (RAG) to deliver personalized, on-demand educational experiences to students. The primary scope is limited to enhancing engagement and learning outcomes for students in introductory psychology courses. The AI assistant will engage with students to assess their understanding, deliver targeted information, provide feedback tailored to their needs, and automatically grade questions and answers. The core functionalities of the product include:

- Real-time interaction with students to answer questions and clarify concepts.
- Continuous assessment of student understanding to provide personalized responses.
- Support for various learning styles by adapting information delivery methods based on student preferences.

Stretch goals include implementing data-driven insights for educators, enabling them to track student progress, and suggesting adjustments to course material based on areas where students commonly need clarification.

The project will initially be scoped to support psychology courses at Oregon State University. However, pending successful implementation and positive feedback from students and faculty, the scope may be expanded to accommodate other disciplines or courses that could benefit from AI-driven, personalized learning tools. The long-term potential includes offering the product to other departments within OSU or educational institutions in the Pacific Northwest region.

Use Cases

- An instructor will be able to assign learning exercises to students.
- A student can self-assess their performance on each exercise based on the specific criteria set by the instructor.
- An instructor will be able to review and evaluate each student's self-assessment for a given exercise.
- A student will be able to view their past exercise history to track progress in their learning, while instructors can monitor student development over time.

Purpose and Vision (Background)

Through this project, we aim to offer an AI-powered assistant that psychology students at Oregon State University can use daily to deepen their understanding and reinforce their knowledge. For instructors, the tool will offer insights into student performance trends, allowing them to make data-informed adjustments to their teaching strategies and provide individualized feedback where needed.

We strive to ensure that our software empowers both students and educators by delivering tailored learning experiences that cater to each student's unique strengths, learning style, and progress. With this approach, our tool is designed to enhance the educational process in a way that is flexible, accessible, and highly responsive to each learner's needs.

Stakeholders

- OSU psychology students
- OSU teachers
- Joseph Slade (Project Partner/Leader)
- Kirsten Winters
- Alex Ulbrich
- Xiaoyan "Sean" Yang (Fall 2024 TA)
- Team Members (Aaron, Elliott, Soham, Padraic)

Our team is guided by Joseph Slade, who has shared his vision of the project and given us the freedom to develop it in our own way. By offering feedback and fostering discussion, Joe helps us understand his vision while allowing us the autonomy to bring it to life.

Weekly meetings with Joe allow us to have an hour of thought provoking discussion. We are able to showcase learning objectives and set new ones for the next meeting. In addition to these in person meetings, Discord allows us to share resources and ideas at any time.

OSU psychology students will be introduced to our application in the Winter term of 2024. These intro level students will test our application and allow for us to make modifications to improve the applications performance. Our goal is to have a beneficial outcome for these students.

Our team is committed to developing a robust application where we have all applied our strongest skills. We are open to utilizing the newest technologies to speed up development. We agreed that the more we can get done the further our scope can reach.

Preliminary Context

Assumptions

- The product we are going to be developing will be a website that is hosted on AWS and will be usable by anyone.
- We will utilize a purpose-made database for the storage of user credentials and information.
- We will have the Psychology 2e textbook with its questions and answers within a vector-database at launch.
- We will have a basic tutorial and introduction available to new users.
- We will have graphs that track the progress of students for instructors to view, as well as heat maps that show areas that students are struggling with.
- The final point of development will be represented by the launching of the website.
- We will be able to communicate with the psychology instructors on an as needed basis.
- We will have launched the completed website by the end of the Spring 2025 term, but will have a beta launched before the start of the Winter 2025 term.
- We assume users are familiar with basic navigation of websites, the ability to input data into an online form, and the ability to communicate with a chatbot.
- The deadline for the product completion is the 7th week of the Spring 2025 term.

Constraints

The main deployment of our application will be web-based, with the primary logic (all the LLM agents in our workflow) being in Python. However, we will need to consider discrepancies in screen sizes, such as if a user is on a tablet, laptop, or a mobile device.

Additionally, while it's a good trait to be ambitious about features that our product will provide, it's also necessary to understand that time is a huge constraint. With this in mind, our primary focus should be to first deliver a barebones product - one of our team activities is to create a priority matrix of all the tasks that we have in our backlog. This prioritization will be written

Another important consideration in terms of our constraints are the tools that we have access to - whether or not we have the correct appropriation of funds to be able to afford certain tools. One example of this is LangSmith - a tool that we'll be using to monitor the number of tokens that we expend, along with a detailed overview of traces that are run in our workflow. This is an essential monitoring tool that we'll be using to gather data, and we'll need to make sure that we have the funds to operate the dashboard.

Some other constraints that we have is determining which application specific tools we need to use. Currently, we are in the process of determining a structured plan of every framework, library, language, and service we'll need, and luckily, most of them are free.

Dependencies

We need access to:

- Vector Database for storing embedded text from psychology textbook - ChromaDB
- The LangChain family of libraries - this is the heart of our application as LangChain provides a framework for building multi-agent architectures to create a complex workflow.
 - LangSmith, LangGraph, and LLM integrations, including embedding functions - mostly will be using
- Flask for Python web integration.
- AWS EC2 for hosting our web application.

Market Assessment and Competition Analysis

The current educational landscape faces challenges in accommodating the diverse learning needs of students, and traditional models struggle to engage students in a personalized, on-demand manner. There is a growing demand for technology-driven educational tools that can provide individualized learning experiences outside classroom hours, especially in large, introductory courses. Our AI-assisted learning product aims to address this gap by offering a highly personalized, 24/7 accessible support tool for students in psychology courses, with future scalability for other subjects and institutions.

Competitor Analysis:

Several educational tools and AI-driven solutions attempt to enhance student engagement, but each falls short of providing a fully integrated solution with features that specifically address asynchronous, personalized learning with real-time feedback. Key competitors include:

- **Khan Academy and Coursera**
 - *Shortcomings:* Lacks real-time interaction and personalized responses to student questions. Content delivery is pre-recorded, limiting adaptability to student understanding.
 - *Target Market:* General education with a focus on self-paced, individual learning rather than personalized, real-time support in specific courses.
- **ChatGPT and Other General-Purpose Chatbots**
 - *Shortcomings:* Lacks direct integration with course-specific content and does not support targeted assessments or real-time tracking of student progress within a course context. Often requires significant instructor input to provide meaningful course-aligned guidance.
 - *Target Market:* Broadly used across sectors, not specialized for academic or course-specific contexts, and lacks insights into educational performance or course metrics.
- **Tutoring and Office Hours**
 - *Shortcomings:* Limited availability outside designated hours, lack of scalability, and depends on instructor or tutor availability. Not a practical solution for all students due to scheduling conflicts.
 - *Target Market:* Effective in one-on-one settings but lacks the scalability to support large student groups or offer on-demand access.
- **Self-Study Resources (e.g., textbooks, study guides)**
 - *Shortcomings:* Lack interactivity and adaptability to individual student needs. Self-study often depends on the student's motivation and self-assessment skills without guidance on where to focus.
 - *Target Market:* Effective for motivated self-learners but lacks adaptability and does not offer feedback or guidance based on student performance.

Summary of Competitive Advantages

Our AI-powered assistant is uniquely positioned to address these gaps by:

- Providing real-time, personalized assistance tailored to course material.
- Leveraging RAG (Retrieval-Augmented Generation) to offer accurate, contextually relevant responses.
- Offering a scalable solution that can be expanded beyond psychology to other courses and institutions over time.
- Supplying faculty with data insights to identify areas where students face challenges and adjust their teaching strategies accordingly.

By combining these elements, our product aims to provide a flexible, affordable, and comprehensive solution that improves student engagement and learning outcomes in a way that current tools do not fully address.

Target Demographics (User Persona)

We have 2 different users for this app: instructors and students.

Harold is a 56-year-old psychology professor at OSU and has been teaching the intro to psychology course for 4 years. Due to the large number of students attending the intro to psychology course, he is unable to host enough office hours to help all the students that need it. He is old-school and does not have much experience with working with computers.

Jim is a 35-year-old psychology professor at OSU and is very interested in tech, specifically in Large Language Models like ChatGPT. He has noticed that students in his intro to psychology class often are not engaged with the course, with many of them not showing up for lecture. He wants to find a way to change this, but he doesn't want to simply assign weekly quizzes or assign what he sees as busy work assignments.

Jason is a 18-year-old freshman student that has just started college and is enrolled in the intro to psychology course. He is having trouble maintaining his focus when reading the provided textbook and is falling behind because of it.

Leah is a 20-year-old psychology student that has switched majors to psychology. They are extremely motivated to get started on this new journey in Intro to psychology. After going to lecture for the week, they are looking to be more prepared for the weekly quiz. Opening up the textbook seems like a daunting task and after a few sentences they feel completely distracted. They wish they could just talk to someone about what they learned so far and see how it compares to the expected knowledge.

Requirements

User Stories and Features (Functional Requirements)

User Story	Feature	Priority GitHub Issue	Dependencies
As a student, I want to have my understanding of material analyzed and presented to me.	Understanding Analysis	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A
As a student, I want to be able to have a user account that keeps track of my queries and performance.	User history	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A
As a student I want the chatbot to explain topics to me at a level appropriate to my personal aptitudes.	Personalized response	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A

As an instructor I want to receive query analysis to determine what subjects students are struggling with.	Query Analysis	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A
As an instructor I want to be able to set the prompt to determine the type of response the agent provides.	Customizable Prompts	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A
As an administrator, I want to be able to see the traces and analytics over time.	Administrative Trace Analysis	Should Have TBD https://github.com/Sodas0/Agentic-Workflows-AI	N/A

Non-Functional Requirements

- The app should be able to function correctly on multiple different web browsers.
- The app should have a simple design for students and instructors who might not be technically proficient. The entire user experience process must be accessible and user-friendly, accounting for users with disabilities.
- The AI models must maintain at least 90% accuracy in delivering content and assessing student performance to ensure quality feedback.
- The system should maintain an uptime of at least 99.9% to ensure consistent availability for students and instructors.
- The tool must comply with FERPA and other relevant regulations to protect student information and performance data.
- All user data must be stored and processed securely, completely eliminating any data leaks.
- User authentication should be managed with secure login mechanisms, including multi-factor authentication for instructors.
- Comprehensive documentation should be provided for both the application and AI model configurations to support future enhancements and maintenance.

Data Requirements

We successfully embed the textbook pdf into a vector database, and retrieve relevant documents relative to the user's prompting.

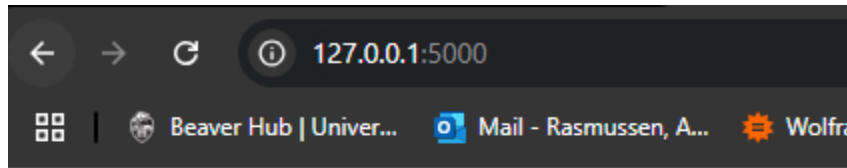
Integration Requirements

- Flask/HTML/CSS - *will be used for the app UI development / frontend*
- Purpose-Built Database - *will be used to store all the data*

User Interaction and Design

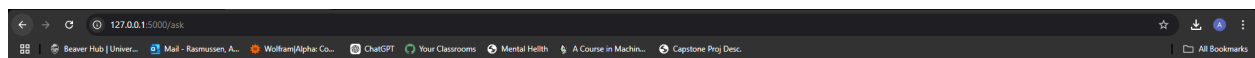
The following screen designs represent the most barebones and up-to-date user interface requirements for our application. We are yet to begin implementation. Our end goal is to have all of the Python logic displayed on the web in a chatbot-esque manner. This will be more refined in the future, and we will create a Figma mock-up of the final product once we solidify our front-end plans:

```
---GENERATE---
"Output from node 'generate':"
'___'
{ 'messages': [ 'The pursuit of happiness involves using our talents to '
                'benefit others or make the world a better place. Happy people '
                'tend to have more friends and social support networks. While '
                'money can buy happiness to some extent, the association '
                'between wealth and happiness is weaker within countries.']] }
```

Ask a Question:

Enter your question:



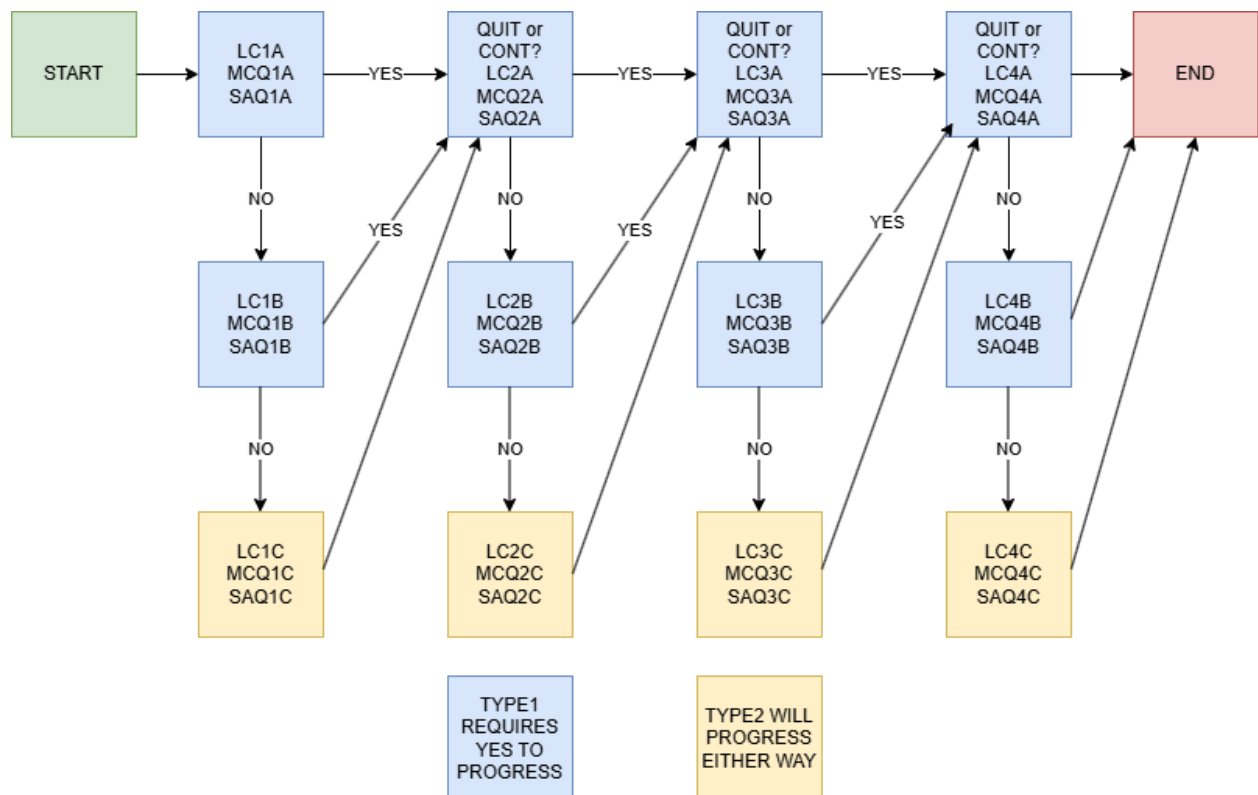
Question:

Who was Pavlov?

Answer:

Pavlov was a Russian scientist known for his research on classical conditioning, particularly through his experiments with dogs, where he discovered how organisms can learn to associate stimuli. He was originally a physiologist studying the digestive system before accidentally uncovering the learning process we now refer to as classical conditioning.

[Ask another question](#)



Real terminal output:

Question: What is Task Decomposition?

Student response: Task decomposition breaks complex tasks into smaller steps, using techniques like Chain of Thought to enhance reasoning and explore options.

Chat GPT Message: *As a grader, compare the student response to the question What is Task Decomposition? to the textbox definition as defined as "Task Decomposition is the process of breaking down complex tasks into smaller, manageable steps. It can be achieved through techniques like Chain of Thought (CoT) and Tree of Thoughts, which guide models to think step by step and explore multiple reasoning possibilities. This approach enhances model performance and provides clarity on the model's decision-making process.". Limit your evaluation to 50 words. At the end return a grade score 0-10. If the grade is below an 8, ask the student a few questions based on the textbook definition to get them on the right track to answer the question better next time. Student response: "Task decomposition breaks complex tasks into smaller steps, using techniques like Chain of Thought to enhance reasoning and explore options.". Structure your response to fit this template: {*

```
"evaluation": {  
  "summary": "", // brief evaluation text, 50 words max  
  "grade": 0,    // integer score from 0-10  
  "feedback": "", // any additional feedback on performance  
  "follow_up_questions": [] // questions to guide student if grade < 8  
}}
```

JSON object is valid!

Graded response (JSON OBJECT):

```
{  
  "evaluation": {  
    "summary": "The student identifies key aspects like breaking down tasks and using Chain of  
Thought but misses details on Tree of Thoughts and clarity on decision-making.",  
    "grade": 7,  
    "feedback": "Good start, but include all techniques mentioned and their impact on  
decision-making.",  
    "follow_up_questions": [  
      "Can you explain what Tree of Thoughts is and how it helps in task decomposition?",  
      "How does task decomposition enhance a model's performance and decision-making clarity?"  
    ]  
  }  
}
```

Milestones and Timeline

For all features, we will all be putting in effort to work on each main feature as we all have full stack knowledge.

Feature	Description	End Date	Dependencies	Assigned To
Webpage	An interactive webpage that allows teachers to start an assignment and users to join an assignment. Users will be able to chat with the chatbot.	Week 8	Server host Database host Flask OpenAI	Aaron
Textbook to RAG	Prepares textbook to be accessed by RAG and Chatgpt	Week 8	Langchain	Soham
Grading Agent	Grades student responses against the retrieved facts from the textbook	Week 8	Langchain Database Textbook OpenAI	Padraic
Database for textbook	Stores text from textbook into a vector database. This database can be accessed through LangChain.	Week 7	LangChain	Elliott
User Analysis	Provides metrics for instructors and our team to understand how well students are learning from our tutor	Week 8	Specialized Database LangSmith Database Host Flask	Everyone
User Testing	Provides information regarding average user behaviors, outliers, vulnerabilities, and bugs.	Winter Term 2025	LangChain LangSmith Flask	Everyone

Goals and Success Metrics

Goal	Metric Baseline Target	Tracking Method
Exam Performance	>5% improvement in letter grade	Anonymous end of term grade aggregation. Provision of grades from a previous term, or a grade average.
User Satisfaction	>80% out of 100% satisfaction rating average	End of term student survey

Open Questions

- Is the app published to the app store?
- How are we going to measure the student's responses for comprehension?
- How are we going to secure user data and stay within FERPA guidelines?
- How do we handle user history to preserve compute time?
- Should users be able to use the product without logging in?

Out of Scope

We do not possess any cybersecurity specialists and may not be able to provide a completely secure service.

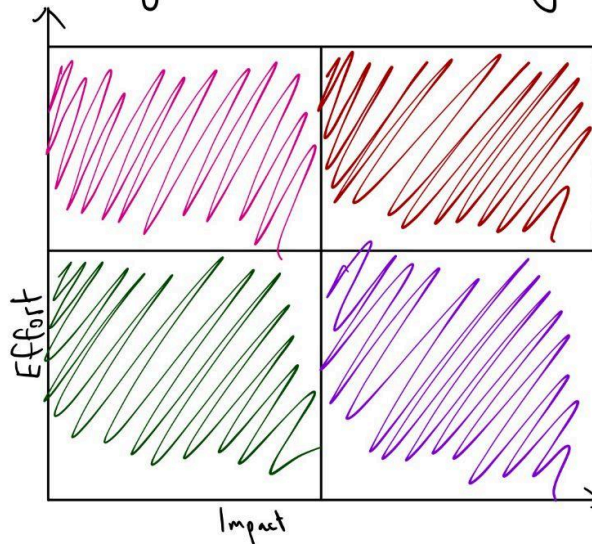
We will not be providing tutors for courses outside Intro to Psychology

We may expand the tutors capabilities to more generally tutor courses, provided we possess enough time.

Activity Write Ups

Team Activity: Prioritization Matrix (Impact-Effort):

Team Activity: Impact-Effort Matrix - Ranking our current backlog of tasks.



Key:

■ = High Impact, High Effort

■ = Low Impact, High Effort

■ = High Impact, Low Effort

■ = Low Impact, Low Effort

■ 1. Develop solution to large context windows that makes sure we don't overwhelm the model

■ 3. Implement a single agent RAG workflow with LangSmith tracing enabled

■ 5. Contact LangSmith about membership pricing.

■ 7. Learn how to customize tools for our agents to conditionally call.

■ 9. Create a CI/CD Pipeline for our application to ensure smooth sailing when it comes to deployment.

■ 2. Read the documentation pages for multi-agent architectures and follow one of the tutorials on LangGraph.

■ 4. Create the vector database for document retrieval

■ 6. Develop an agent to serve chapter content to our user.

■ 8. Research and implement a simple query analysis workflow.

■ 10. Research hosting platforms & pricing for our website deployment.

Team Activity: User Story Map:

Ask a question		Prepare for an Exam		Evaluate class performance		Activities
Log in	Select a Course	Submit a prompt	Ask for re-explanation	View trace information	Continue lesson without understanding check	Steps
Enter username	Use dropdown to chose from available classes	Select the text entry box	Click the Re-explain Button	Press Analyze Button	When the bot asks a question, press SKIP	Details
Enter password	Press GO button	Type a question		Select Category		
Press login button		Press ENTER or click Submit		View Data analytic information		

Epic: User Authentication

As a user, I want to log in using my username and password so that I can access my personalized learning dashboard.

Epic: Question and Exam Preparation

As a user, I want to submit a question prompt to the system so that I can receive an answer or further explanation.

Epic: Course Selection and Navigation

As a user, I want to select a course from a dropdown list so that I can focus on the specific class content.

Epic: Performance Evaluation and Analysis

As an instructor, I want to analyze student performance data so that I can identify areas where students need additional support.

Individual Activity: Literature Review - Conducted by Soham Das. This is a summarization of my research and experimentation. I've read through dozens of documentation pages from LangChain and LangGraph, especially those pertaining to what I believe will be helpful for the entire team to know about when we start our main development sprints. I believe that diving headfirst into solving our problem is a suboptimal approach, and that we should take our time learning about the tools we'll be using in order to effectively write and maintain clean code when it comes time for the development of our project. My literature review notes can be found [here](#).

Individual Activity: Literature Review - Conducted by Elliott Shim.

I have extensively reviewed the LangChain and LangGraph documentation and have successfully adapted several implementation tutorials for use in our project. I have become very familiar with the process of retrieving data from a PDF and translating it into embedded data stored in a vector-database. I am also familiar with the methods of retrieving this data and pre-processing it using LangChain tools in order to serve the information stored in the vector-database (and in the chat-log memory) to agents as useable data. Finally, I have also reviewed documentation regarding some databases which may be used to store user data such as login credentials and form-data such as preferences for the way the chatbot speaks to a given user.

Individual Activity: Literature Review - Conducted by Aaron Rasmussen.

My responsibilities in this project are to set up the front end / web site that our users will interact with. We have decided that using Flask would be a good idea. Prior to working on this project I have never used Flask so I needed a general understanding of the framework and its capabilities. The following is a literature review of an article I read.

The article, 'Understanding Flask Framework: Installation, features & Expert Insights' provides a comprehensive overview of Flask, a micro web framework in Python that is widely used in web development and data science. It highlights the framework's versatility, explaining that while Flask is primarily associated with building web applications, it is also commonly employed for data science tasks, such as creating APIs and visualization tools that allow data scientists to present their findings in an interactive manner. The framework's minimalistic design is emphasized as a key factor in its popularity, providing developers with essential tools without overwhelming them with unnecessary complexity.

Overall, the article presents Flask as a powerful yet straightforward web framework that has gained popularity for its flexibility and simplicity. Its lightweight design, combined with an ecosystem of extensions, allows developers to use Flask for everything from small web projects to complex applications with specific needs like API creation, data visualization, and microservices. Flask's compatibility with data science and machine learning further enhances its versatility, making it an appealing choice for developers and data scientists alike. [3]

Individual Activity: Literature Review - Conducted by Padraic Bergin.

In reading, I have learned some introductory ideas about prompt engineering. Prompt engineering is the process in which human written language is adjusted when interacting with a Large Language Model. This is done to set the context for the model and to increase the likelihood that the question is answered correctly or the instructions are followed. A crucial step in this process is Chain-of-thought, this method uses known questions and answers to guide the LLM down the right path before asking it an unknown question. This is the current basis of the grading agent I have implemented so far. The RAG is able to pull the context to the question from the textbook and ChatGPT then has this information to reference when grading the students response. [4]

Works Cited:

- [1] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.
<https://doi.org/10.1073/pnas.1319030111>
- [2] LangChain Inc. (n.d.). LangGraph. Retrieved November 8, 2024, from <https://langchain-ai.github.io/langgraph/>
- [3] Ashray, July 23, 2024, Understanding Flask Framework: Installation, Features & Expert Insights. Retrieved November 6, 2024, from <https://www.analyticsvidhya.com/blog/2021/10/flask-python/>
- [4] Prompt Engineering Guide." Prompt Engineering Guide, Accessed 8 Nov. 2024, from <https://www.promptingguide.ai/>