



BACHELOR PROJECT

AI-Enhanced Comprehensive Learning Platform



Contents

Exploration of AI-Powered Tutoring Systems and Personalized Learning Solutions.....	3
Objective	3
Research Methodology.....	3
Literature Review.....	3
Technology Assessment.....	3
State of the art.....	4
Expected Outcomes	4
Automated Content Design and Tailoring.....	4
Engagement through Interactive Tools	4
Tools	4
Scientific papers	5
Research Landscape of Adaptive Learning in Education: A Bibliometric Study on Research Publications from 2000 to 2022.....	5
A Review of Agent-Based Programming for Multi-Agent Systems.....	5
QuizBot: A Dialogue-based Adaptive Learning System for Factual Knowledge.....	6
Fine-tuning GPT-3 Using Python to Create a Virtual Mental Health Assistant Bot	7
Building agents (AI tutor and personalized learning) with Flowise	8
Personalized learning inside the webapp	9
Model Comprehension.....	9
Data Collection	10
Learning Experience Personalization	10
Evaluation and Iterative Development	10
Documentation	10
Developing a GUI for AI Interactions in Django	11
Overview	11
Design Considerations	11
Implementation Steps.....	11
Step 1: Define the Layout	11
Step 2: Explore Existing Templates and Work	11
Step 3: Customize Your Django Templates.....	11
Step 4: Implement AJAX for Dynamic Content Loading	12
Step 5: Testing and Refinement.....	12

Conclusion.....	12
Architecture	13
Activity Diagram of system.....	14
AI Chat Tutor Assistant	15
AI Course Content Generator	16
AI Interaction games	17
User Profile Agent	18
PDF based AI agents that to the lessons:	19
Graphical use interface	20
Home page	20
Log in	20
Sign in	21

Bachelor project

Exploration of AI-Powered Tutoring Systems and Personalized Learning Solutions

Objective

The initial phase of our project is dedicated to a thorough examination of the current landscape of AI-powered tutoring systems and personalized learning solutions. Our goal is to investigate existing models, dissect their strengths and weaknesses, compare their methodologies, and delve into their architecture and implementation strategies. This exploration is crucial for laying the foundational knowledge required to develop an innovative approach to AI tutoring. By addressing identified gaps and incorporating cutting-edge technologies, we aim to enhance the efficiency, inclusivity, and adaptability of educational environments.

Research Methodology

To achieve our objectives, we will employ a multi-faceted research methodology comprising the following components:

Literature Review

- **Objective:** To conduct an exhaustive literature review aimed at identifying critical peer-reviewed articles, case studies, and white papers that detail the latest advancements and challenges in AI tutoring and personalized learning systems.
- **Approach:** For the literature review approach, we will meticulously analyze scientific publications and established AI models that are integral to the domain of AI-driven tutoring and personalized learning. Our investigation will specifically target peer-reviewed scholarly articles and case studies that report on the practical application and efficacy of current AI educational technologies.

Technology Assessment

- **Objective:** To evaluate the underlying technologies and algorithms of existing models, with a particular focus on their adaptability to diverse learning styles, scalability across various educational contexts, and effectiveness in engaging users.
- **Approach:** Analyze the technical documentation, developer guides, and research papers related to prominent AI tutoring platforms. Special attention will be given to AI methodologies such as machine learning, natural language processing, and adaptive learning technologies that facilitate personalized education.

State of the art

Expected Outcomes

The culmination of this phase will result in a comprehensive understanding of the state-of-the-art in AI-powered tutoring and personalized learning. By synthesizing insights from our literature review, technology assessment, and market analysis, we will identify key gaps and opportunities in the current ecosystem. This knowledge will serve as the cornerstone for the next phase of our project, where we will design and propose an innovative AI tutoring model that aims to surpass the limitations of existing solutions and significantly enhance the learning experience.

This project capitalizes on AI to elevate the quality of education provided, evaluation methods, and administrative efficiency. It crafts a dynamic, student-centered learning space by harnessing:

Automated Content Design and Tailoring

- AI is utilized to automate the development of lesson plans and create personalized educational resources, analyzing curriculum requirements and student performance to produce the most relevant material.

Engagement through Interactive Tools

- AI-enhanced tools are deployed to foster interactive learning, offering quizzes and dynamic presentations that adjust in real-time to student responses, ensuring a highly engaging and responsive educational experience.

The integration of these tools is streamlined and specific to meet distinct educational needs. The project relies on data-driven strategies to assess tool impact and refine learning journeys. Constantly evolving with the AI and edtech field, the project prioritizes continual enhancements for optimal learning results.

The project upholds a seamless, learner-adjusted, and data-oriented use of AI resources to transform education, striving for a personalized and inclusive academic experience.

Tools

For accessing the tools mentioned (link on tool name):

- [GitHub Copilot](#)
- [OpenAI's ChatGPT](#)
- [Smart Sparrow](#),
- [Knewton](#),
- [ClassPoint](#),
- [DreamBox Learning](#),
- [ExamSoft](#)

Scientific papers

Study the Impact of Adaptive Learning on education: does it ensure student interaction and provide optimal learning outcomes?

Research Landscape of Adaptive Learning in Education: A Bibliometric Study on Research Publications from 2000 to 2022

The study offers a detailed bibliometric analysis on adaptive learning from 2000 to 2022, highlighting rapid development within this research area. Key findings include: A significant contribution from authors like Qiao J. F., Han H. G., and Song Q., with China leading in publications.

Core journals identified, notably IEEE Transactions on Neural Networks and Learning Systems, emphasize the field's growth.

Major research topics emerged: deep learning applications in education, development of adaptive learning models, intelligent tutoring systems, and advanced modeling technologies for learning.

Technological advancements are driving the field's evolution, focusing on feature extraction, adaptation models, and computational modeling as current research frontiers.

This analysis underscores adaptive learning's interdisciplinary nature, bridging computer science and educational technology, and highlights the crucial role of emerging technologies in education.

A Review of Agent-Based Programming for Multi-Agent Systems

The document "A Review of Agent-Based Programming for Multi-Agent Systems" presents a systematic examination of agent-based programming (ABP) languages, their applications, extensions, and comparative analyses. It emphasizes the role of intelligent, autonomous agents in symbolic artificial intelligence, where agents reactively or proactively decide on actions based on available world information. The paper highlights the diverse range of agent programming languages (APLs), including both established and emerging languages, and discusses their applications across various domains.

Key insights include:

- **Agent Programming Languages (APLs):** The review identifies several APLs, focusing on both well-maintained veteran languages and newer entrants. The Belief-Desire-Intention (BDI) model emerges as a popular framework among the discussed APLs, which are often implemented in Java for cross-platform compatibility.
- **Extensions and Comparisons:** The paper discusses significant extensions to existing APLs that enhance their functionality or apply them to new, specific scenarios. It also addresses

the challenge of qualitatively and quantitatively comparing these languages, given their diverse underlying models of agency.

- **Applications:** A wide range of applications is covered, illustrating the utility of APLs in domains like electricity markets, robotics, and the Internet of Things (IoT). The Multi-Agent Programming Contest (MAPC) is highlighted as a platform for testing and comparing APLs in complex scenarios.

The paper underscores the importance of bridging the gap between theoretical agent concepts and practical programming paradigms to encourage wider adoption of APLs. It suggests that future research should focus on addressing usability issues, including improving documentation, providing more real-world examples, and enhancing the development of benchmarks for APL evaluation.

This document serves as a comprehensive resource for understanding the current state of agent-based programming, identifying future research directions, and appreciating the practical applications of agent technologies in solving complex, distributed problems.

QuizBot: A Dialogue-based Adaptive Learning System for Factual Knowledge

Advances in conversational AI have the potential to enable more engaging and effective ways to teach factual knowledge. To investigate this hypothesis, we created QuizBot, a dialogue-based agent that helps students learn factual knowledge in science, safety, and English vocabulary. We evaluated QuizBot with 76 students through two within-subject studies against a flashcard app, the traditional medium for learning factual knowledge. Though both systems used the same algorithm for sequencing materials, QuizBot led to students recognizing (and recalling) over 20% more correct answers than when students used the flashcard app. Using a conversational agent is more time consuming to practice with, but in a second study, of their own volition, students spent 2.6x more time learning with QuizBot than with flashcards and reported preferring it strongly for casual learning. Our results in this second study showed QuizBot yielded improved learning gains over flashcards on recall. These results suggest that educational chatbot systems may have beneficial use, particularly for learning outside of traditional settings.

Fine-tuning GPT-3 Using Python to Create a Virtual Mental Health Assistant Bot

The article on Better Programming guides readers through the process of fine-tuning GPT-3 to develop a virtual mental health assistant. It provides a detailed walkthrough of the procedure, including preparing and formatting the training data, choosing the right fine-tuning parameters, and running the fine-tuning process. The piece emphasizes the importance of fine-tuning for creating a more helpful and contextually aware conversational agent tailored to specific use cases like mental health support. It also touches on the ethical considerations and importance of privacy when handling sensitive topics.

Important: paper links are linked to the title

Building agents (AI tutor and personalized learning) **with Flowise**

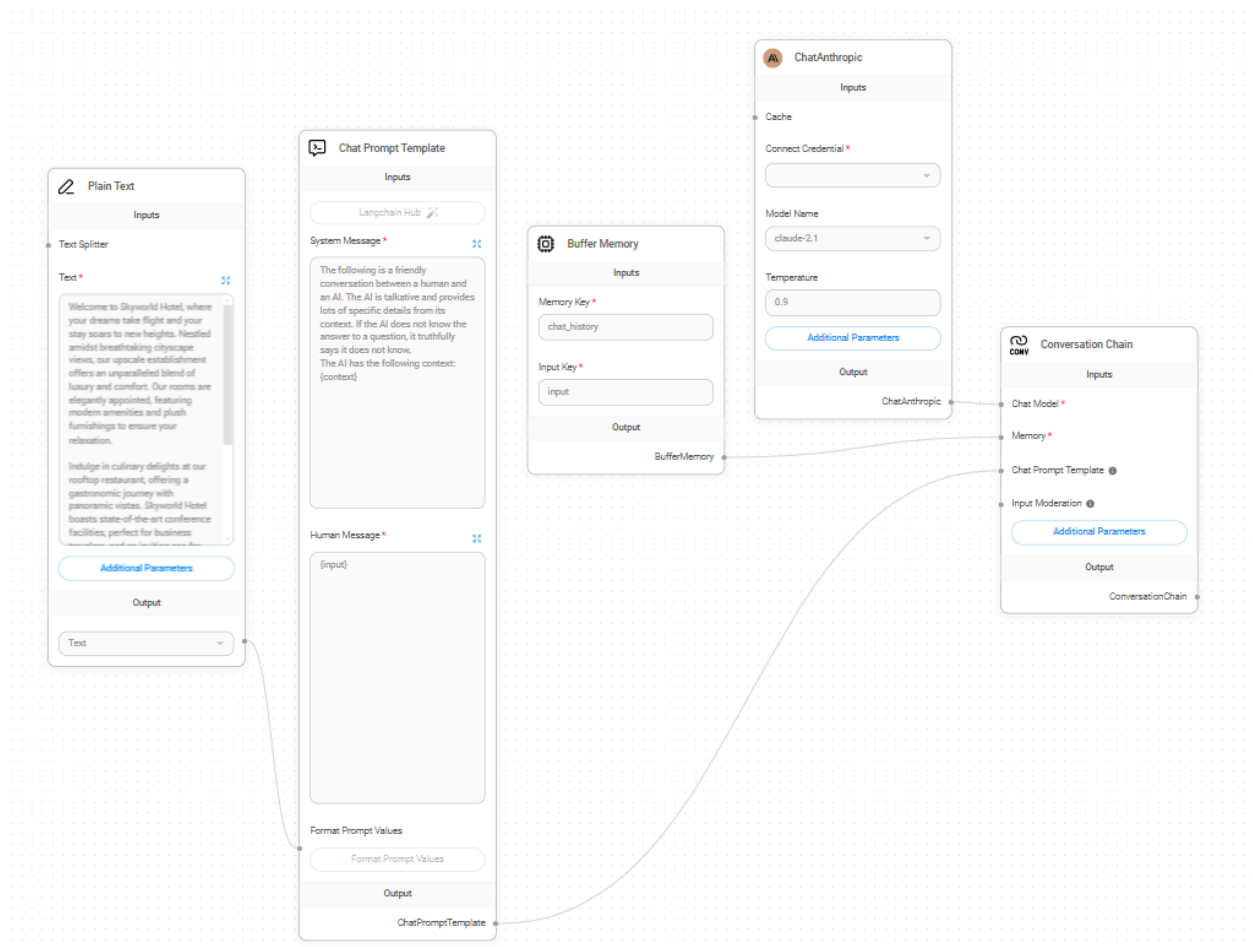
Flowise is an advanced platform that empowers developers to create custom Large Language Model (LLM) applications through an intuitive drag-and-drop interface. It facilitates seamless integration into existing applications via APIs, SDKs, and an embedded chat widget. This platform supports a variety of open-source LLMs, including HuggingFace and Orca, and enables self-hosting on major cloud services like AWS, Azure, and GCP, making it a versatile tool for developing AI-driven solutions.

In the context of building AI assistants or tutors for a bachelor project, Flowise offers a unique approach to enhance chatbot capabilities beyond what standard LLMs provide. For instance, while LLM-based chatbots are powerful, they have limitations, such as the inability to perform web searches or complex mathematical reasoning effectively. Flowise addresses these limitations by allowing the integration of specialized nodes into the chatbot's workflow, such as a ConversationalAgent node for orchestration, a Buffer Memory node for context retention, and a Chat OpenAI node equipped with GPT models for advanced conversational abilities. Additionally, tool nodes like Calculator for arithmetic tasks and SERP API for internet queries can be integrated, significantly expanding the chatbot's functionality.

This system enables the creation of sophisticated AI assistants capable of conducting quizzes, facilitating educational content, and providing personalized tutoring experiences. By leveraging the modularity of Flowise, developers can construct AI agents that not only respond to queries but also remember conversation histories, access the internet for real-time information retrieval, and perform specific tasks requested by users. This approach allows for the development of a highly interactive and responsive educational tool, tailor-made to the requirements of a bachelor project.

For a detailed guide on setting up your first AI agent with Flowise, including step-by-step instructions on configuring nodes and integrating API keys, the Just Ship AI tutorial offers a practical walkthrough. This resource can serve as a valuable reference for incorporating advanced AI functionalities into your project, ensuring a comprehensive and engaging user experience.

Github: <https://github.com/FlowiseAI/Flowise>



Personalized learning inside the webapp

Integrating a machine learning model to personalize learning experiences in web applications offers a tailored approach to education, accommodating individual learning speeds, styles, and levels. This document outlines a structured method for implementing such a model, exemplified by a model found on Hugging Face or similar platforms, focusing on categorizing users into different learning levels: beginner, intermediate, and advanced.

Model Comprehension

Firstly, it's imperative to fully understand the chosen model's capabilities, inputs, and outputs. Key aspects include:

- **Inputs:** Identify the data the model requires, which might encompass user responses, interaction behaviors, or stated preferences.
- **Outputs:** Comprehend how the model's outputs correlate with user learning preferences and abilities, facilitating their alignment with predefined learning levels.

Data Collection

Gather initial data through:

- **Onboarding Quizzes/Surveys:** Craft these tools to get the user's knowledge base and learning preferences.
- **Activity Analysis:** Monitor and analyze how users engage with the platform, focusing on preferred content types and interaction patterns.

Learning Experience Personalization

- **Adaptive Content:** Modify content difficulty and format based on the model's analysis, ensuring it meets the learner's current level and preferences.
- **Feedback Mechanisms:** Utilize quizzes and exercises to validate the model's level placement, adjusting as necessary to optimize learning.

Evaluation and Iterative Development

- **A/B Testing:** Conduct comparative studies between personalized and static learning paths to assess the effectiveness of personalization.
- **Iterative Refinement:** Use outcomes and feedback to continuously refine the integration, potentially retraining the model with specific user data to enhance its predictive accuracy.

Documentation

- **Guidance Materials:** Provide comprehensive guides explaining the personalized learning system's operation and maximizing its benefits.

Developing a GUI for AI Interactions in Django

Overview

The GUI serves as the bridge between your project's AI capabilities and its users, enabling intuitive interactions and display of the AI-generated content. By following best practices in Django template creation and seeking inspiration from existing projects, you can create a compelling user experience that showcases the innovative features of your AI-driven application.

Design Considerations

- **User Experience (UX):** Prioritize simplicity and clarity in your design to facilitate ease of use.
- **Responsive Design:** Ensure your GUI is accessible across various devices and screen sizes.
- **Interactivity:** Incorporate interactive elements that engage users and encourage them to explore AI-generated content.

Implementation Steps

Step 1: Define the Layout

1. Sketch the layout of your GUI, identifying key components such as input fields for user prompts, areas to display AI-generated responses, and navigation elements.
2. Consider using Bootstrap or another CSS framework to expedite development and ensure a responsive design.

Step 2: Explore Existing Templates and Work

1. Look for Django template repositories or open-source projects for inspiration. Websites like GitHub, GitLab, or Django Packages can be valuable resources.
2. Evaluate templates and projects that have integrated AI or chat functionalities to understand how they structure their GUI and manage user interactions.

Step 3: Customize Your Django Templates

1. Based on your layout and inspiration from existing work, create or modify templates in your Django app. These templates should include:
 - **Base Template:** Defines the overall HTML structure (header, footer, etc.) and includes links to CSS and JavaScript files.
 - **Home Template:** The main interface for user interaction with the GPT model, including form fields for input and a section to display responses.
2. Utilize Django's template inheritance feature to maintain consistency across your application and reduce redundancy.

Step 4: Implement AJAX for Dynamic Content Loading

1. To enhance user experience, use AJAX to submit user prompts and display AI-generated responses without reloading the page.
2. Update your Django views to handle AJAX requests and return JSON responses that include the AI-generated content.

Step 5: Testing and Refinement

1. Test the GUI across different devices and browsers to ensure compatibility and responsiveness.
2. Gather user feedback on the GUI's usability and make necessary adjustments to improve the interface and interaction design.

Conclusion

By carefully designing and implementing a GUI for your Django project, you can significantly enhance the way users interact with your AI-driven features. Drawing inspiration from existing templates and projects, combined with thoughtful customization and testing, will lead to a user-friendly interface that effectively showcases the capabilities of your self-improving GPT model.

Architecture

Architecture for an AI-Enhanced Comprehensive Learning Platform, designed to revolutionize the educational experience by integrating advanced AI to offer personalized learning pathways, real-time tutoring, and a suite of interactive tools aimed at enhancing student engagement and learning outcomes.

User Registration and Login: The platform provides a seamless user registration and login process. New users are prompted to register by entering their email, password, full name ensuring a secure and personalized experience from the outset. Upon registration, users undergo an initial skill assessment, answering questions related to their proficiency in various computer languages (e.g., Python, Java) and overall coding skills (ranging from Beginner to Engineer). This critical step allows the AI to tailor the learning pathway to each student's unique needs and skill levels.

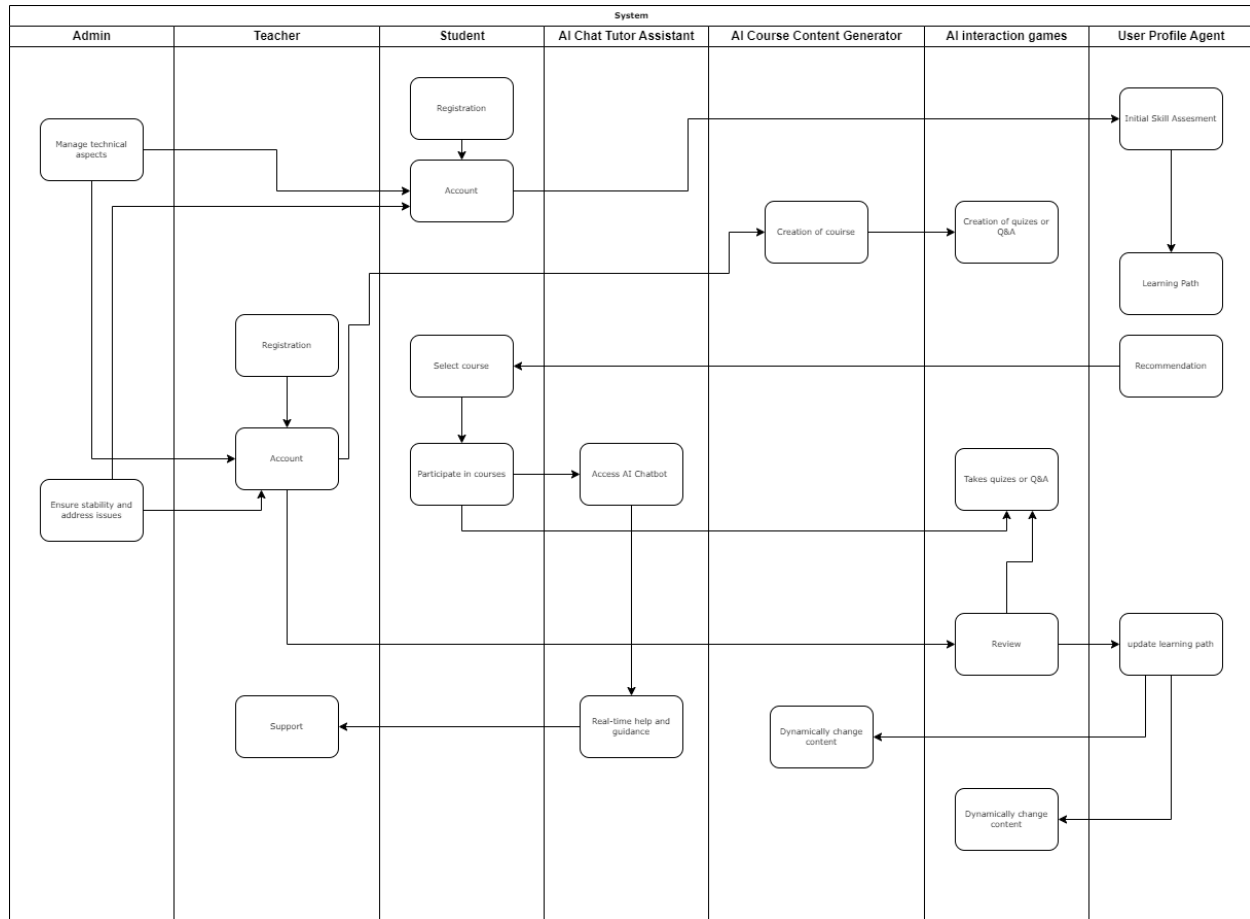
Skill Assessment: Before accessing the main features of the web application, new users complete a skill assessment questionnaire. Questions gauge the user's proficiency in several programming languages and their coding experience level. This information is pivotal for customizing the learning experience, ensuring that content is appropriately challenging and relevant.

Dashboard and Features: Post-login, the dashboard serves as the central hub for accessing the platform's diverse features:

- **Courses:** Students can enroll in and participate in various courses tailored to their learning paths.
- **Profile:** Information of user:
 - Teacher:** Courses created
 - User:** Courses and Recent games
- **Quizzes:** Interactive quizzes allow students to assess their understanding and retention of course material.
- **Q&A Section:** A collaborative space for students to ask questions and receive answers, fostering a community of learning.
- **AI Tutor:** The intelligent tutoring interface offers real-time assistance, homework help, and simulates test environments for practice and assessment, providing explanations and guidance at the student's level.
- **Personalized Learning Pathways:** Powered by AI, the platform analyzes each student's strengths, weaknesses, learning styles, and pace, curating a customized learning plan that adapts dynamically to progress and feedback.

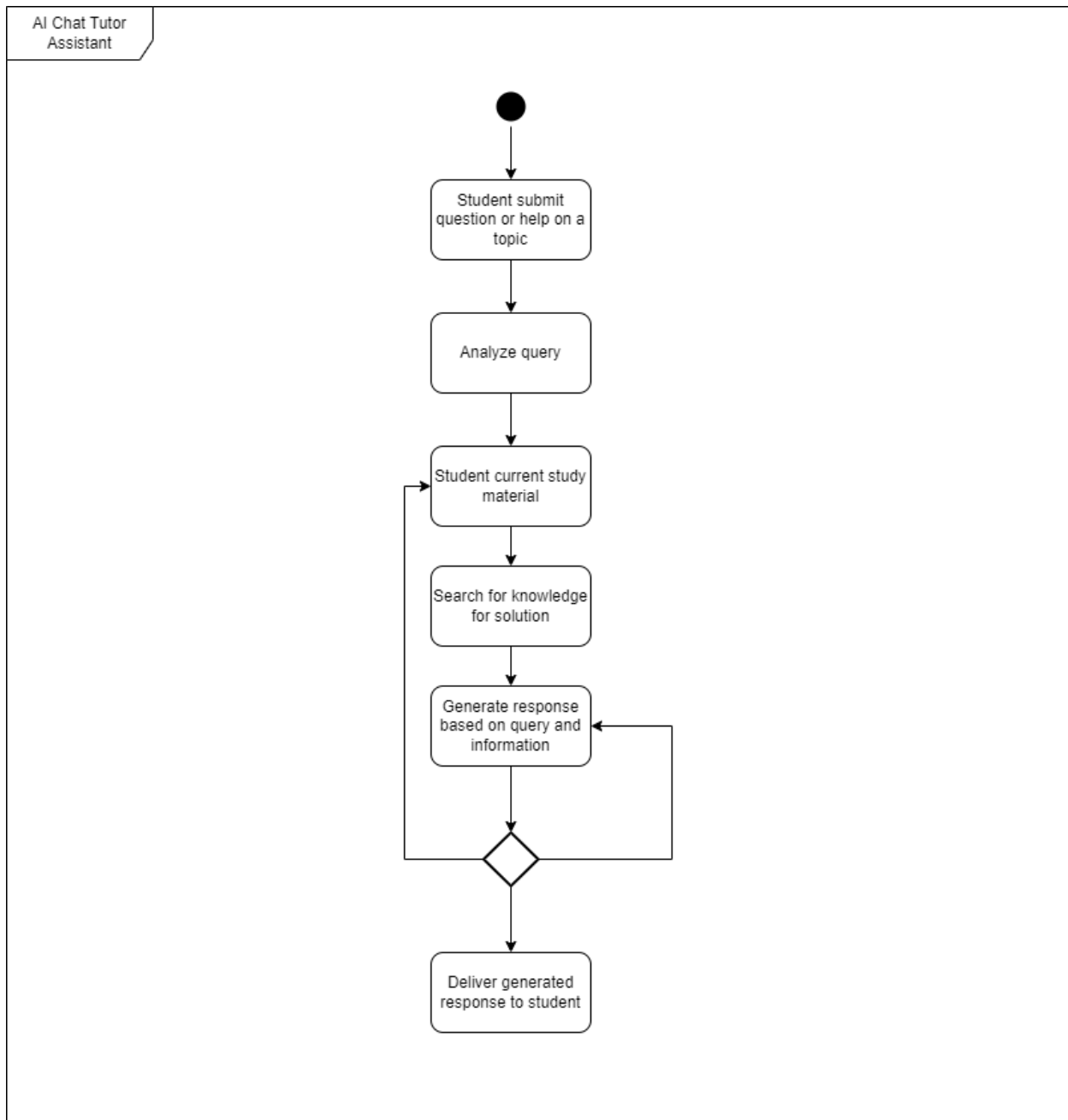
Core Objective and Features: At its core, the platform aims to address the adaptive learning challenge, questioning how AI-driven personalization affects student interaction and learning outcomes. By integrating personalized learning pathways with an intelligent tutoring interface, the platform aspires to scale adaptability and efficiency, making it an indispensable tool for after-school learning, homework assistance, and exam preparation.

Activity Diagram of system



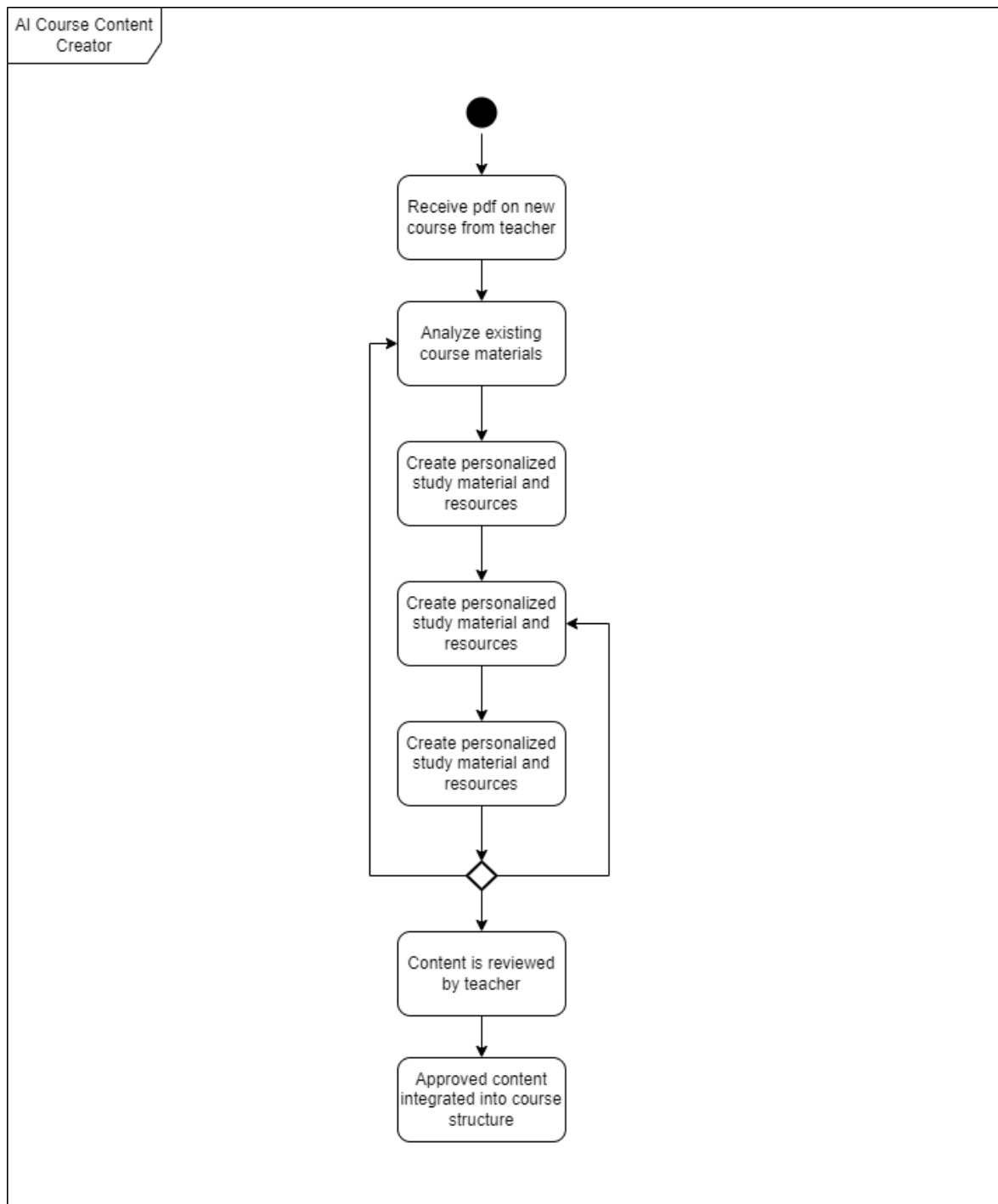
AI Chat Tutor Assistant

This agent acts as a virtual tutor, available to students for real-time academic assistance. It uses advanced natural language understanding to interpret student queries and provide explanations, hints, and educational support. Powered by GPT-3.5 Turbo, it can generate contextually relevant and nuanced responses to a wide array of subjects, assisting students as they navigate through course materials.



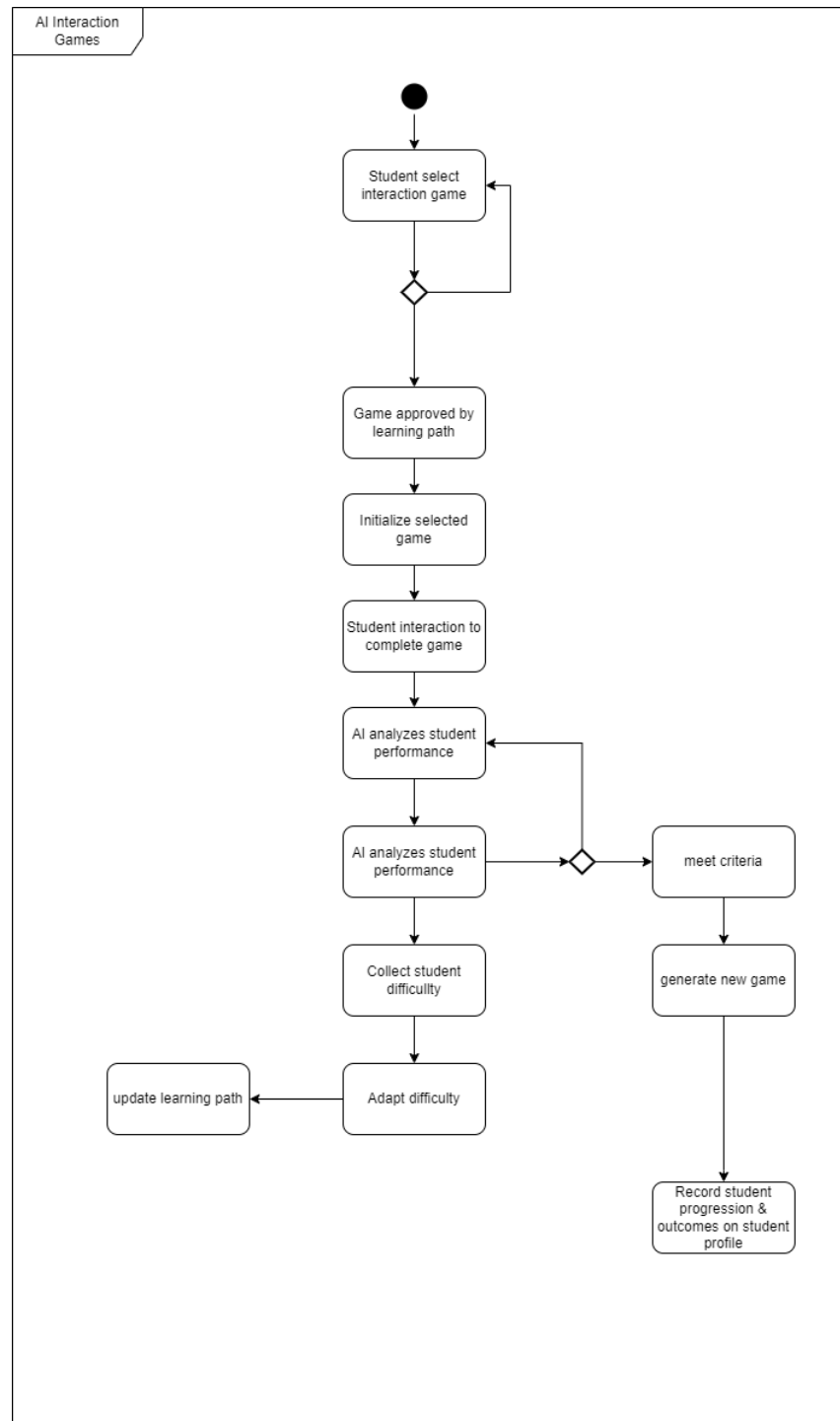
AI Course Content Generator

This agent is responsible for the automated creation of educational content. Leveraging GPT-3.5 Turbo's capabilities, it can analyze curriculum standards and generate course materials, including lecture notes, reading assignments, and interactive learning modules. Its content production is adaptive, considering the varying educational needs and learning objectives outlined by educators.



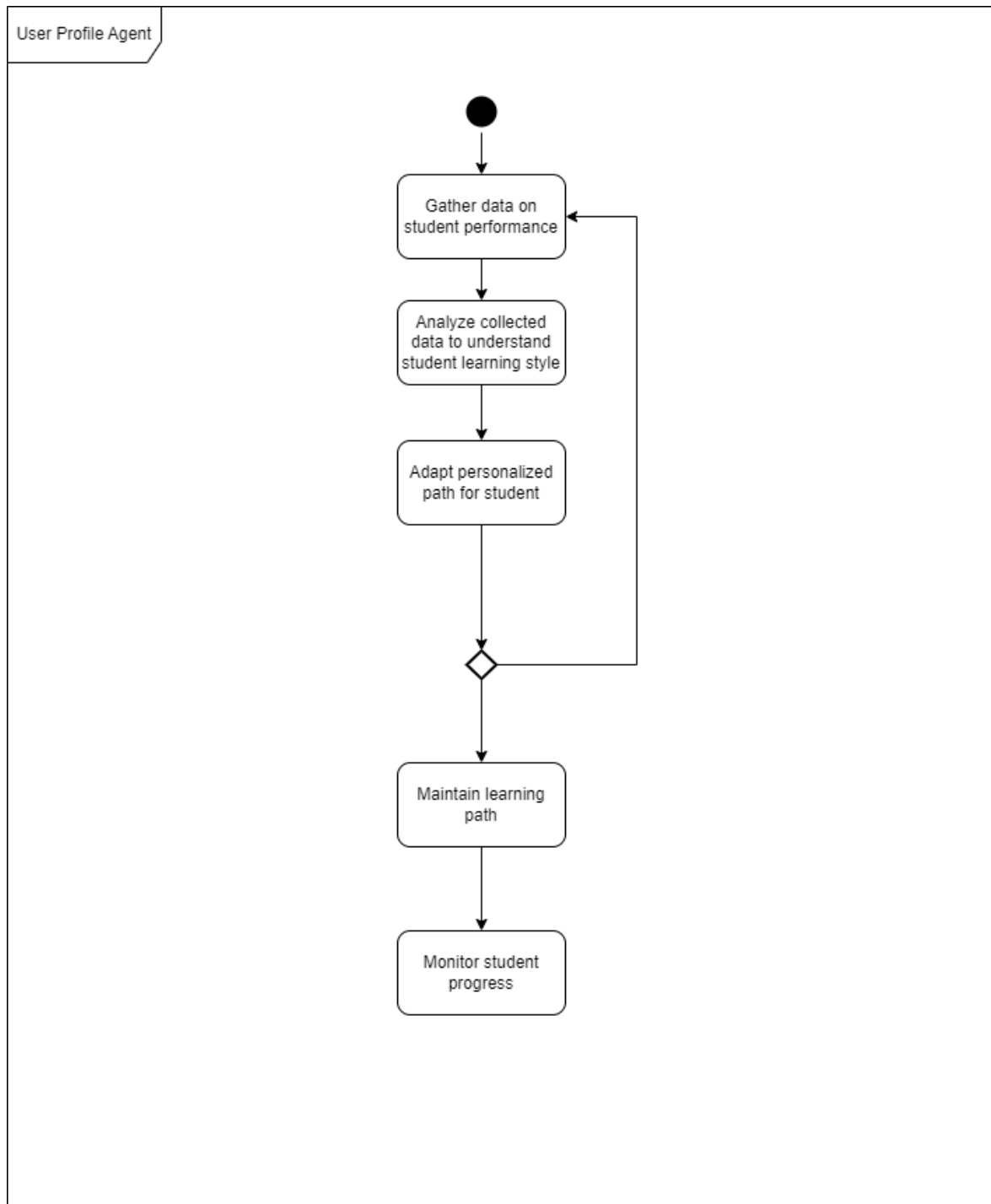
AI Interaction games

The AI interaction games agent employs GPT-3.5 Turbo to create engaging and educational games that adapt to the student's skill level. The games are designed to reinforce learning objectives in an interactive format, providing instant feedback and dynamically adjusting challenges to suit the student's evolving competencies.



User Profile Agent

This agent specializes in personalization by analyzing students' performance data, learning preferences, and engagement metrics. Using GPT-3.5 Turbo's pattern recognition and predictive modeling, it can tailor learning pathways for each student, making recommendations for courses, suggesting suitable learning resources, and adjusting the difficulty of material as the student progresses.



PDF based AI agents that to the lessons:

In the architecture of the AI-Enhanced Comprehensive Learning Platform, a pivotal feature is the integration of an AI Assistant capable of transforming PDF content into structured lessons tailored to individual students. This AI Assistant employs advanced natural language processing (NLP) and machine learning algorithms to analyze educational materials provided in PDF format. It then intelligently segments this content into discrete, digestible lessons that align with the curriculum's scope and sequence. By evaluating the student's progress, preferences, and performance data, the AI dynamically adjusts the content complexity and topics to ensure personalized learning paths. This approach not only facilitates a more engaging and effective learning experience but also allows for the automated creation of a diverse and adaptive curriculum. The AI Assistant's capacity to convert static PDF documents into interactive, personalized learning experiences represents a significant leap forward in educational technology, offering scalable, personalized education solutions that meet the unique needs of each learner.

PDF for python lesson:

https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf

PDF for java lessons :

<https://www.cs.cmu.edu/afs/cs.cmu.edu/user/gchen/www/download/java/LearnJava.pdf>

PDF for JavaScript lessons:

https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf

PDF for C lessons:

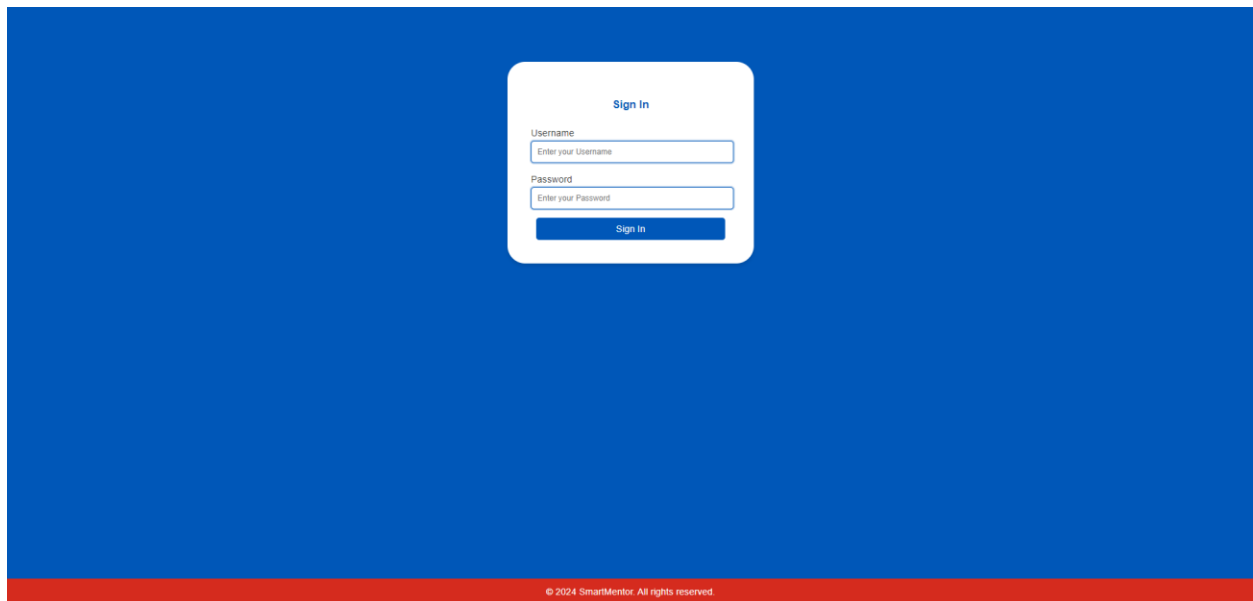
<https://vardhaman.org/wp-content/uploads/2021/03/CP.pdf>

Graphical use interface

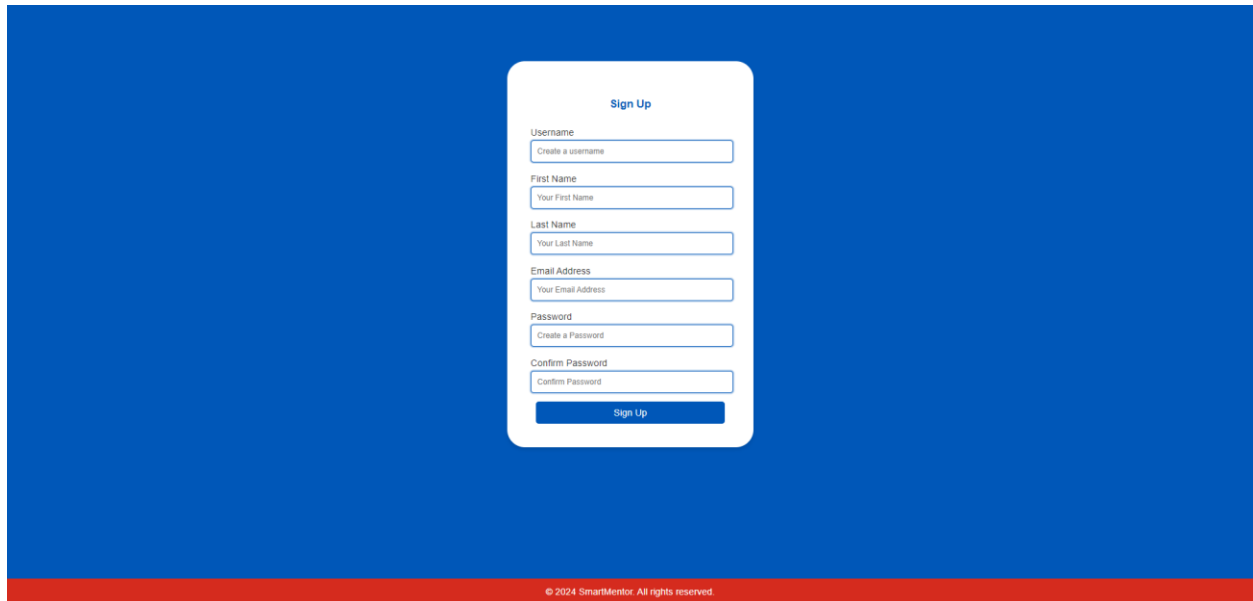
Home page



Log in



Sign in



The image shows a 'Sign Up' form centered on a solid blue background. The form is a white rounded rectangle with a blue border. At the top of the form, the text 'Sign Up' is displayed in blue. Below this, there are seven input fields, each with a label above it: 'Username' (with placeholder 'Create a username'), 'First Name' (with placeholder 'Your First Name'), 'Last Name' (with placeholder 'Your Last Name'), 'Email Address' (with placeholder 'Your Email Address'), 'Password' (with placeholder 'Create a Password'), and 'Confirm Password' (with placeholder 'Confirm Password'). At the bottom of the form is a blue button with the text 'Sign Up' in white. A red horizontal bar at the very bottom of the page contains the copyright text '© 2024 SmartMentor. All rights reserved.'

Sign Up

Username
Create a username

First Name
Your First Name

Last Name
Your Last Name

Email Address
Your Email Address

Password
Create a Password

Confirm Password
Confirm Password

Sign Up

© 2024 SmartMentor. All rights reserved.