



POLITECNICO MILANO 1863

TEACHme: Leveraging Large Language Models (LLM) for Second Language Conversational Practice in Human-Agent Interaction

Design Document

Version 3.0 – 01/06/2024

Alireza Yahyanejad (10886993)

Mahdi Valadan (10903871)

Table of Contents

1. Introduction	1
1.1 Project Overview.....	1
1.2 Objectives	1
1.3 Scope.....	1
2. System Architecture.....	1
2.1 High-Level Architecture	1
2.1.1 Presentation Tier	1
2.1.2 Logic Tier	1
2.1.3 Data Tier.....	2
2.2. High Level Components and Their Interactions	2
2.3 Component Description.....	2
2.3.1 Front-End Application.....	2
2.3.2 Reasons for choosing technologies in front-end	3
2.3.3 Large Language Model Integration.....	4
3. Detailed Design	4
3.1 Front-End Design	4
3.1.1 User Interface	4
3.1.2 User Experience (UX)	5
3.2 LLM Integration.....	5
3.2.1 Conversation Flow	5
3.2.2 Error Handling.....	5
4. Implementation Plan	5
4.1 Development Phases	5
4.2 Tools and Technologies	5
4.3 Testing Plan.....	6
5. User Interface Design.....	6
5.1 Mockup	6
5.1.1 Student Panel.....	6
5.1.2 Teacher Panel	7
5.1.3 Teacher Panel – Add Student	7
5.2 Real Application	8
5.2.1 Student Dashboard	8

5.2.2 Teacher Dashboard	9
5.2.3 Sign-up Page	10
5.2.4 Login Page	10
5.2.5 Creating New Content	11
5.2.6 Manage Students	11
5.2.7 Conversation Page	12
5.2.8 Feedback Page	13
6. Deployment Plan	15
6.1 Hosting and Infrastructure.....	15
6.2 Continuous Integration/Continuous Deployment (CI/CD)	15
7. Maintenance and Support	15
7.1 Monitoring	15
7.2 User Support	15

Table of Figures

Figure 1 High Level Components and Their Interactions	2
Figure 2 Mockup - Student Panel	6
Figure 3 Mockup - Teacher Panel	7
Figure 4 Mockup - Teacher Panel - Add Student	7
Figure 5 Student Dashboard - part 1	8
Figure 6 Student Dashboard - part 2	8
Figure 7 Teacher Dashboard - part 1	9
Figure 8 Teacher Dashboard - part 2	9
Figure 9 Sign-up Page	10
Figure 10 Login Page	10
Figure 11 Creating New Content	11
Figure 12 Manage Students	11
Figure 13 Conversation Page - part 1	12
Figure 14 Conversation Page - part 2	12
Figure 15 Conversation Page - part 3	13
Figure 16 Feedback Page - part 1	13
Figure 17 Feedback Page - part 2	14
Figure 18 Feedback Page - part 3	14
Figure 19 Feedback Page - part 4	15

1. Introduction

1.1 Project Overview

Second language acquisition poses several challenges due to cost and accessibility barriers associated with conversational practice, such as hiring native English speakers for practicing speaking skills. This project aims to leverage Large Language Models (LLMs) embedded in embodied agents to enhance conversational practice for Italian speakers learning English.

1.2 Objectives

- Develop a scalable and cost-effective solution for conversational practice in second language acquisition.
- Integrate LLMs into embodied agents to provide realistic and interactive language practice sessions.
- Create a user-friendly application interface using Next.js for seamless user experience.

1.3 Scope

The scope of this project includes:

- Development of a front-end application using Next.js.
- Integration of LLMs to facilitate conversational practice.
- Deployment on suitable platforms for accessibility by Italian speakers.

2. System Architecture

2.1 High-Level Architecture

The system architecture of the project follows a three-tier architecture model, which includes the Presentation Tier, Logic Tier, and Data Tier. This model ensures separation of concerns, scalability, and ease of maintenance.

2.1.1 Presentation Tier

The Presentation Tier, or front-end, is developed using Next.js. It is responsible for providing the user interface and handling user interactions. The key features of this tier include:

- User authentication and profile management
- An interactive chat interface for conversational practice
- Progress tracking and feedback mechanisms

The Presentation Tier communicates with the Logic Tier via API calls to retrieve and display data to the user.

2.1.2 Logic Tier

The Logic Tier, or back-end server, is built with Node.js and Express. It is responsible for implementing the business logic and application functionality. The main responsibilities of this tier include:

- Providing API endpoints for user management, chat sessions, and progress tracking
- Integrating with the Large Language Model (LLM) API for real-time conversation handling

- Processing user requests from the Presentation Tier and interacting with the Data Tier to fetch or store data

The Logic Tier serves as the intermediary between the Presentation Tier and the Data Tier, handling all business logic and data processing tasks.

2.1.3 Data Tier

The Data Tier consists of a database, such as MongoDB, responsible for storing and managing all data. This includes user information, session logs, and progress tracking data. The Data Tier's primary responsibilities are:

- Storing and managing data securely
- Providing data access and retrieval operations for the Logic Tier

The Data Tier directly interacts with the Logic Tier to handle all data-related operations, ensuring data consistency and integrity.

2.2. High Level Components and Their Interactions

The chosen hardware architecture for this web application is three-tier. The architecture contains three application layers. A tier to interface with the client, the second tier for the application level, and another is server for database management. This division is efficient because the server tier can make a permanent connection with the DBMS, which is less expensive. As well as, having a middle tier between the user and the server can guarantee more security to the access control of the database from the users. Figure 1 represent the high-level components with their interactions.

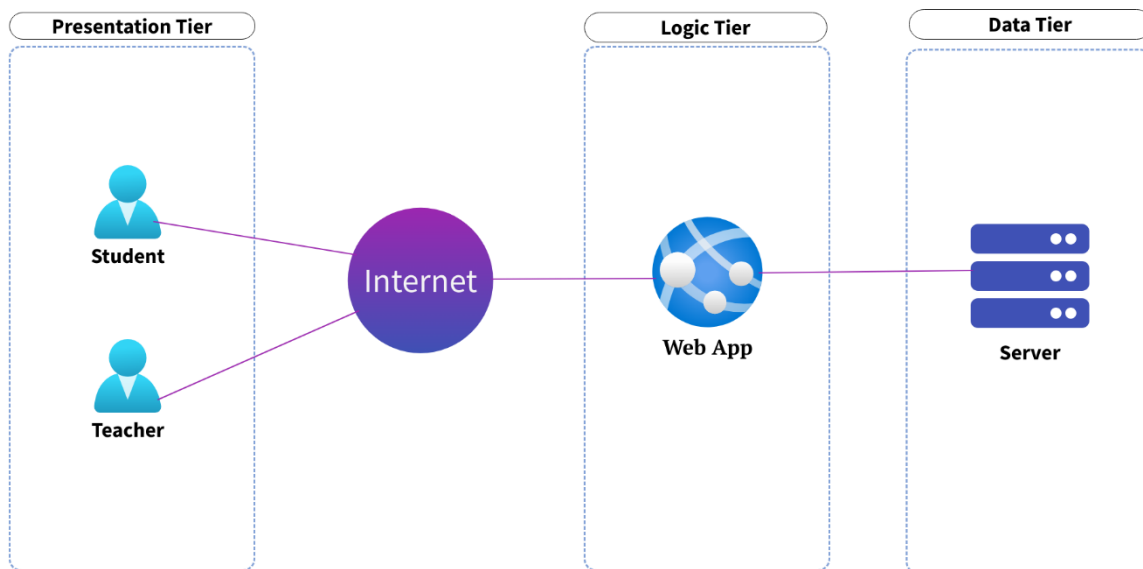


Figure 1 High Level Components and Their Interactions

2.3 Component Description

2.3.1 Front-End Application

Next.js is used as the framework. Its features are:

- User authentication and profile management.
- Interactive chat interface for conversational practice.
- Progress tracking and feedback mechanisms.
- An avatar with lip movements.

For the avatar the following technologies are used:

- ChatGPT for script creation.
- Azure Text-to-Speech for audio synthesis.
- Viseme for phoneme frame data generation.
- Canvas for rendering mouth animations.

2.3.2 Reasons for choosing technologies in front-end

Each technology was chosen for its unique strengths and capabilities that contribute to the overall success of the project.

2.3.2.1 Next.js as Frontend

Rationale: Next.js is a React-based framework known for its component-based architecture and ease of use, which facilitates the development of complex user interfaces. It supports Server-Side Rendering (SSR), improving performance and SEO, which is beneficial for web applications that require fast load times and high visibility. Additionally, Next.js supports Static Site Generation (SSG), enabling the creation of static pages that can be pre-rendered for even better performance and scalability.

Benefits: Improved performance and SEO are achieved through SSR. Development is eased with React's component-based approach. Scalability and flexibility are ensured with support for both SSR and SSG.

2.3.2.2 ChatGPT for script creation

Rationale: ChatGPT, powered by OpenAI's GPT-4 architecture, excels in generating human-like text based on input prompts, making it ideal for creating coherent and contextually appropriate scripts. It offers flexibility, as it can be fine-tuned and adapted to various styles and tones, making it versatile for different types of content. Additionally, automating script generation reduces the time and effort required to produce high-quality scripts, enabling rapid content creation and iteration.

Benefits: The use of ChatGPT results in high-quality, contextually relevant scripts. It provides customizable outputs to suit project-specific needs. Furthermore, it offers a scalable solution for generating large volumes of content.

2.3.2.3 Azure Text-to-Speech for audio synthesis

Rationale: Azure Text-to-Speech provides natural-sounding, high-fidelity speech synthesis, enhancing the overall quality of the audio output. The service offers various voices and languages, allowing for tailored audio that matches the desired character and tone of the content. Additionally, Azure's robust API makes it easy to integrate with other components of the project, streamlining the workflow.

Benefits: The use of Azure Text-to-Speech results in professional and realistic audio output. It offers a wide range of voice options and languages. Furthermore, it enables seamless integration with existing systems and services.

2.3.2.4 Viseme for phoneme frame data generation

Rationale: Viseme provides detailed phoneme frame data, which is crucial for synchronizing mouth movements with spoken audio, ensuring accurate lip-sync. Utilizing viseme data helps achieve more precise and natural mouth animations, improving the visual appeal and realism of the animations. Additionally, automating the generation of phoneme frame data reduces manual animation efforts and increases consistency across different content pieces.

Benefits: Using viseme data enhances lip-sync accuracy. It improves animation quality and realism. Furthermore, it saves time by automating the generation of animation data.

2.3.2.5 Canvas for rendering mouth animations

Rationale: The HTML5 Canvas API is highly versatile and can be used for a wide range of graphics and animations, including complex mouth movements. Canvas provides efficient rendering capabilities, ensuring smooth animations even for detailed mouth movements. Additionally, as a standard web technology, Canvas is widely supported across different browsers and platforms, ensuring broad accessibility.

Benefits: The use of the HTML5 Canvas API results in high-performance, smooth animations. It offers broad compatibility and support. Furthermore, it provides flexibility for creating detailed and dynamic animations.

2.3.3 Large Language Model Integration

OpenAI ChatGPT serves as the LLM provider, offering real-time conversational response generation. It adapts to the user's language proficiency level and manages context to maintain coherent conversations.

3. Detailed Design

3.1 Front-End Design

3.1.1 User Interface

According to section **Error! Reference source not found.**, application has parts:

3.1.1.1 Student Dashboard

Within the student dashboard, all conversations and feedback are readily available. Upon logging into their account, students gain access to this section of the application, where they can view the topic, difficulty, and level associated with each conversation and feedback.

3.1.1.2 Teacher Dashboard

Upon logging into their account, teachers are presented with an overview of all conversations and their respective students. Each conversation is accompanied by its topic, difficulty, level, and the students' email addresses.

3.1.1.3 Sign-up Page

Both students and teachers can easily create their accounts through the sign-up page by providing their name, email, and password.

3.1.1.4 Login Page

Users can log in to their accounts by entering their credentials. Incorrect username or password entries will prevent access to the account.

3.1.1.5 Creating New Content

Exclusive to teachers, this section allows them to create new content by selecting the user level, difficulty, student, duration, and topic. Upon clicking the "Create" button, the new content is generated.

3.1.1.6 Manage Students

Teachers can manage their students, including adding or removing them from their roster, within this section.

3.1.1.7 Conversation Page

In the conversation page, students can engage with topics previously created by their teacher. By clicking the "Start" button, students can begin conversing with the chatbot about the selected topic. They can also end the conversation at any time by clicking the "End Conversation" button.

3.1.1.8 Feedback Page

Within this section, students can review the details of their conversation, including overall feedback and specific details. The "Conversation Detail" section displays the topic, difficulty, user level, and teacher information. Students can access overall feedback and delve into conversation specifics such as content, feedback, synonyms, pronunciation, and AI responses.

3.1.2 User Experience (UX)

The application features intuitive navigation and a minimalistic design to enhance usability. It employs a responsive design to support various devices and includes interactive elements to engage users in conversational practice.

3.2 LLM Integration

3.2.1 Conversation Flow

The system initiates with an initial greeting and topic selection tailored to user preferences. It ensures conversational coherence through context-aware responses from the LLM. Additionally, it dynamically adjusts conversation complexity based on the user's proficiency level.

3.2.2 Error Handling

The system handles API errors gracefully, providing appropriate user feedback. It implements fallback mechanisms to address failed LLM responses, ensuring a seamless user experience even in the face of unexpected errors.

4. Implementation Plan

4.1 Development Phases

- Phase 1: Initial setup and basic front-end development
- Phase 2: Back-end development and API integration
- Phase 3: LLM integration and testing
- Phase 4: User interface refinement and final testing
- Phase 5: Deployment and user feedback collection

4.2 Tools and Technologies

The front-end of the system utilizes Next.js, React, and Tailwind CSS for an optimized user experience. MongoDB serves as the database backend, providing efficient data storage and retrieval. Integration with

the LLM is achieved through the OpenAI API, enabling seamless integration of natural language processing capabilities into the application.

4.3 Testing Plan

- Unit Testing: For individual components and functions.
- Integration Testing: Ensuring smooth interaction between front-end, back-end, and LLM.
- User Acceptance Testing: Collecting feedback from target users to refine the application.

5. User Interface Design

5.1 Mockup

5.1.1 Student Panel

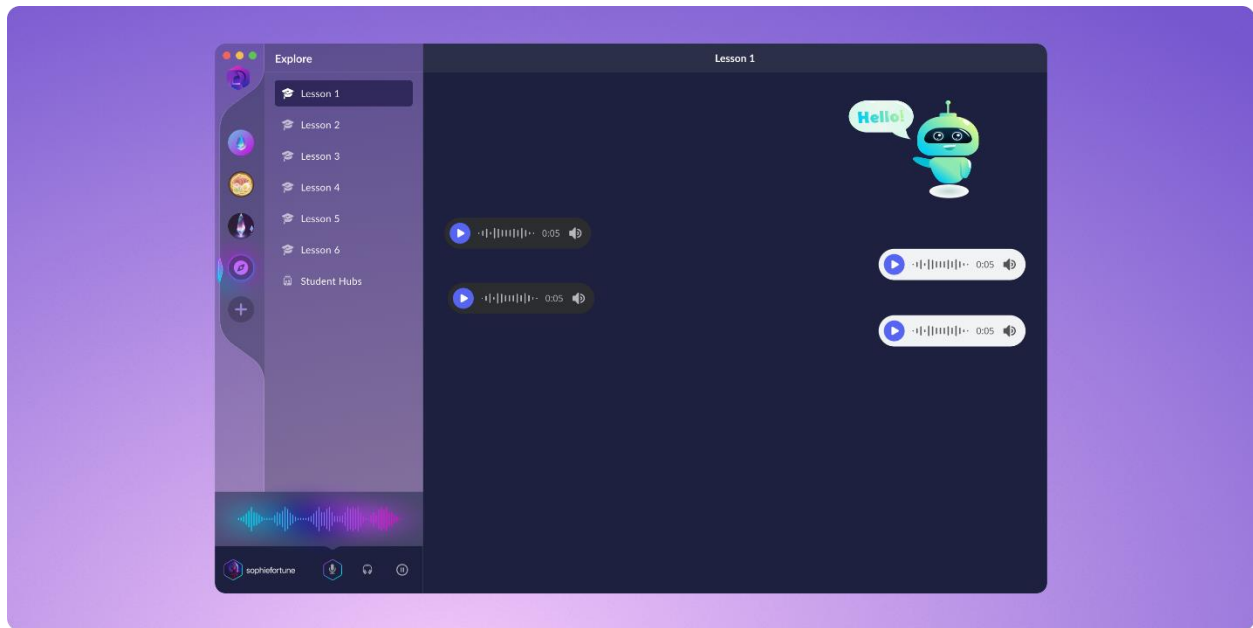


Figure 2 Mockup - Student Panel

5.1.2 Teacher Panel

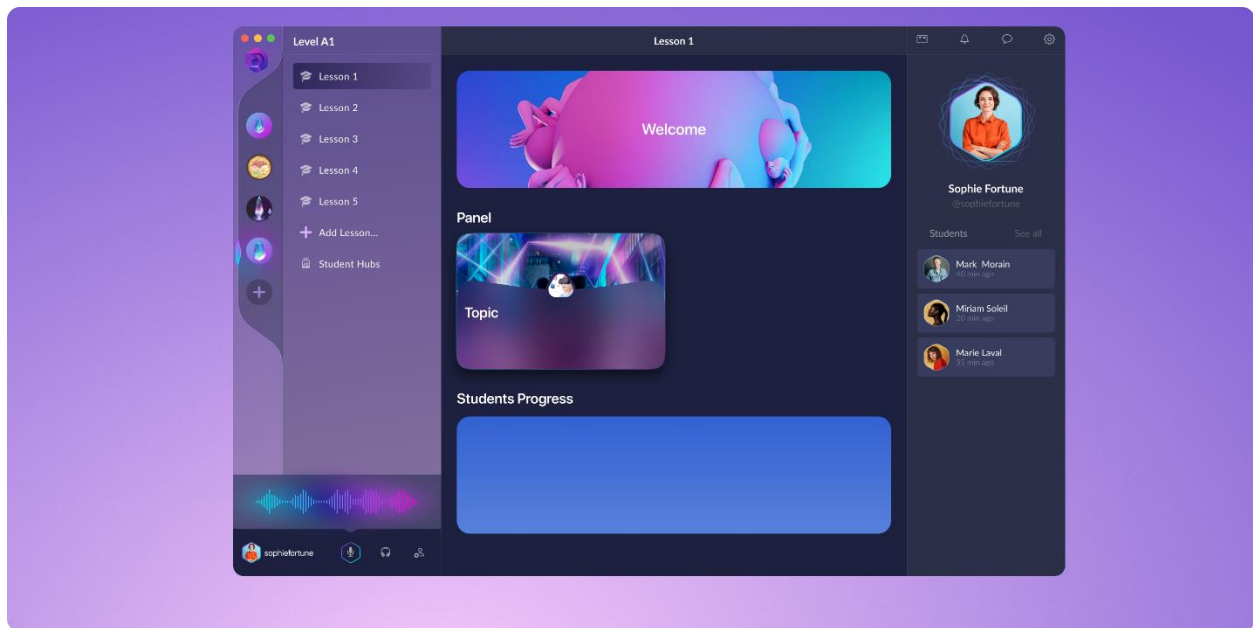


Figure 3 Mockup - Teacher Panel

5.1.3 Teacher Panel – Add Student

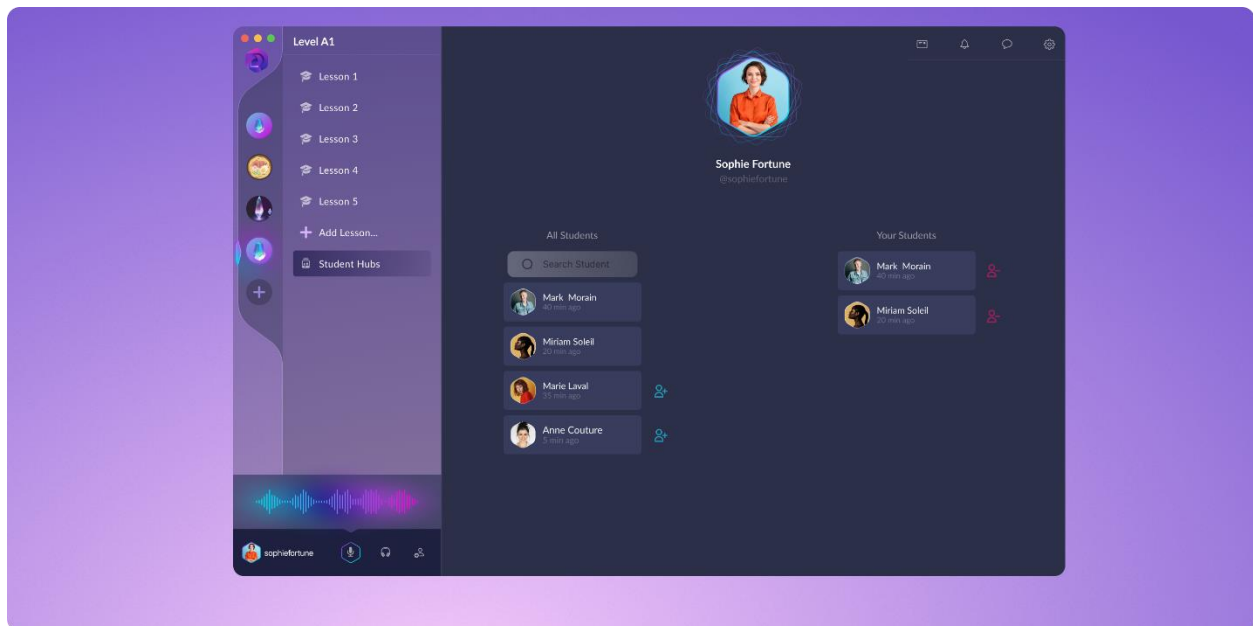


Figure 4 Mockup - Teacher Panel - Add Student

5.2 Real Application

5.2.1 Student Dashboard

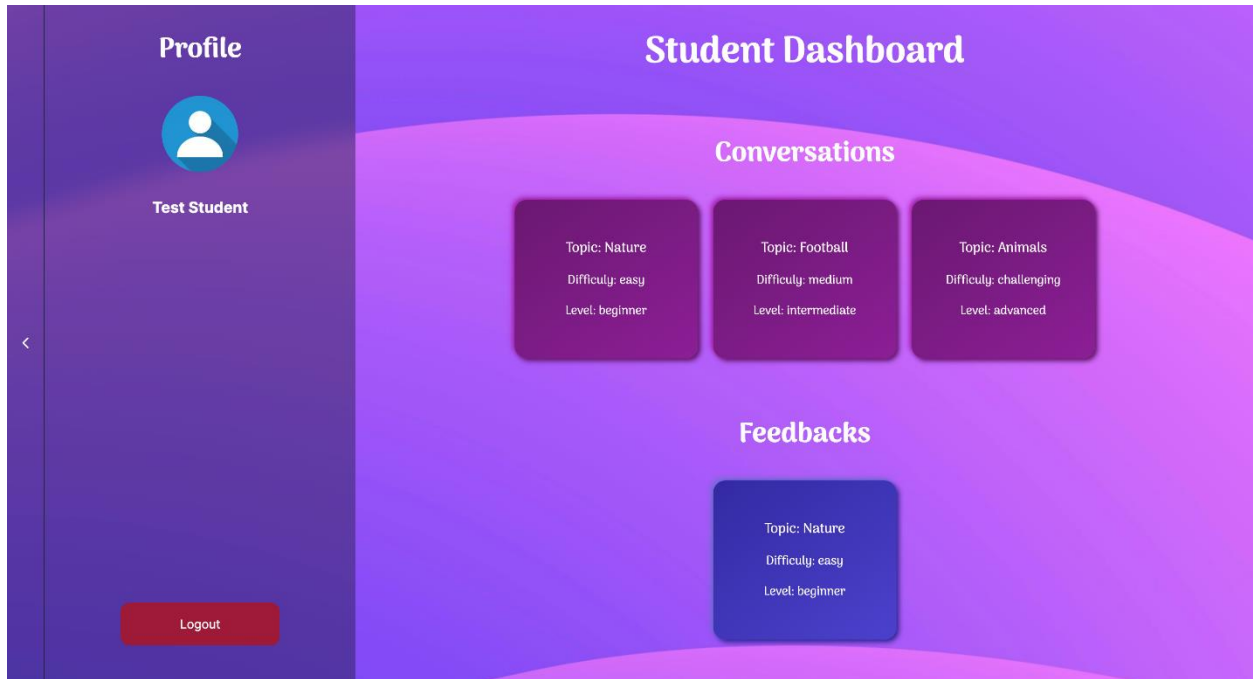


Figure 5 Student Dashboard - part 1

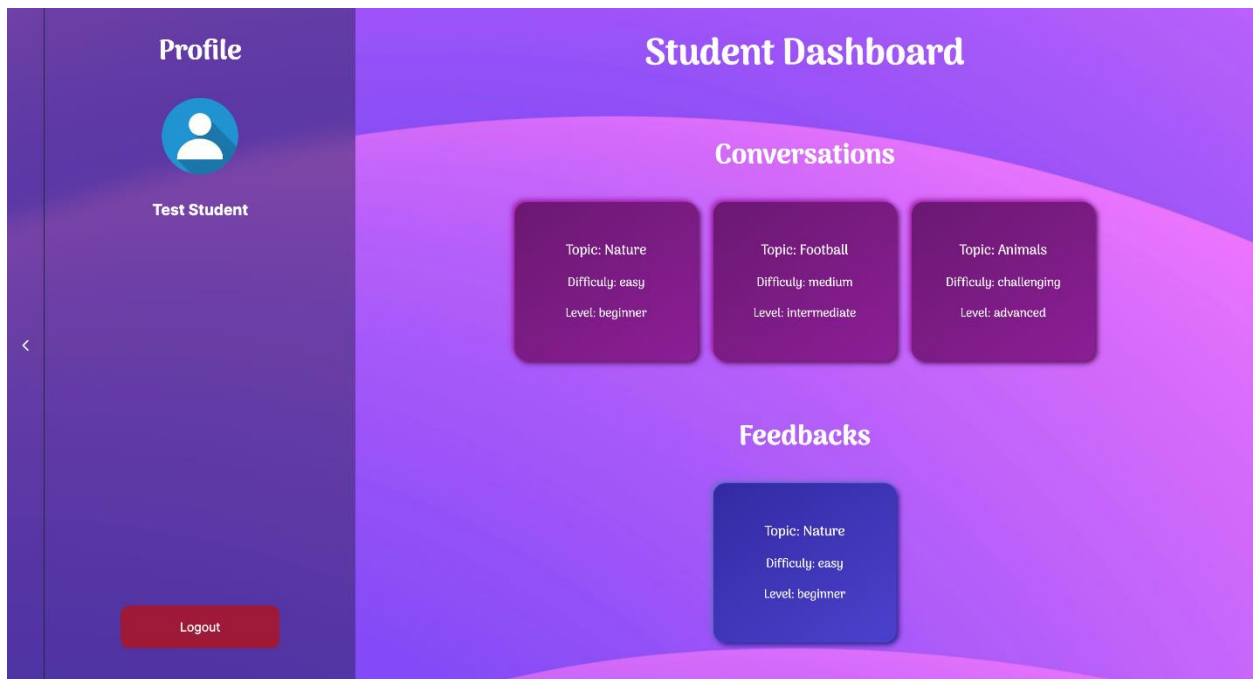


Figure 6 Student Dashboard - part 2

5.2.2 Teacher Dashboard

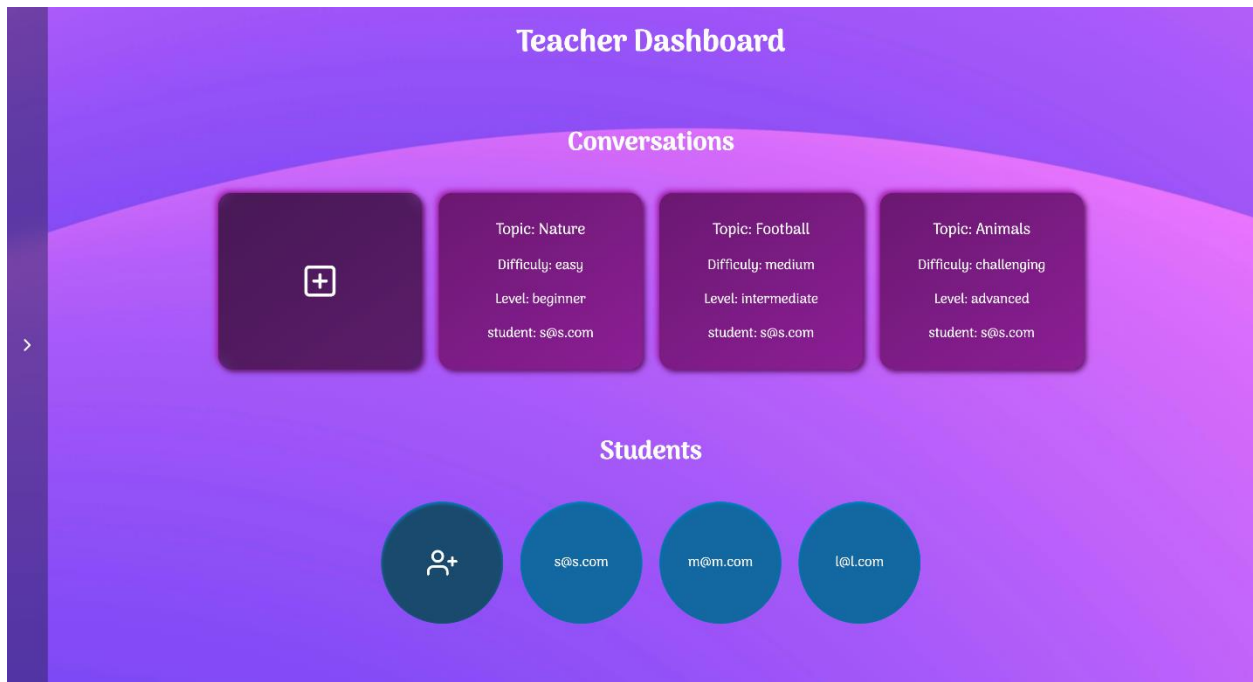


Figure 7 Teacher Dashboard - part 1

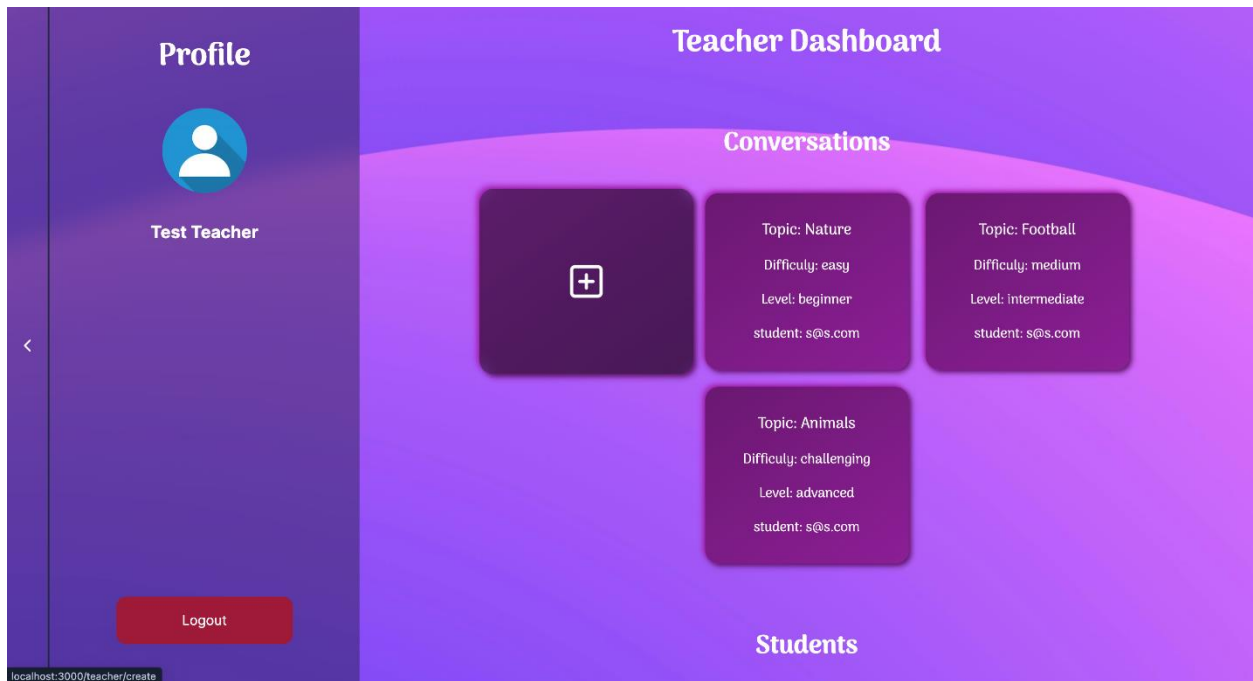
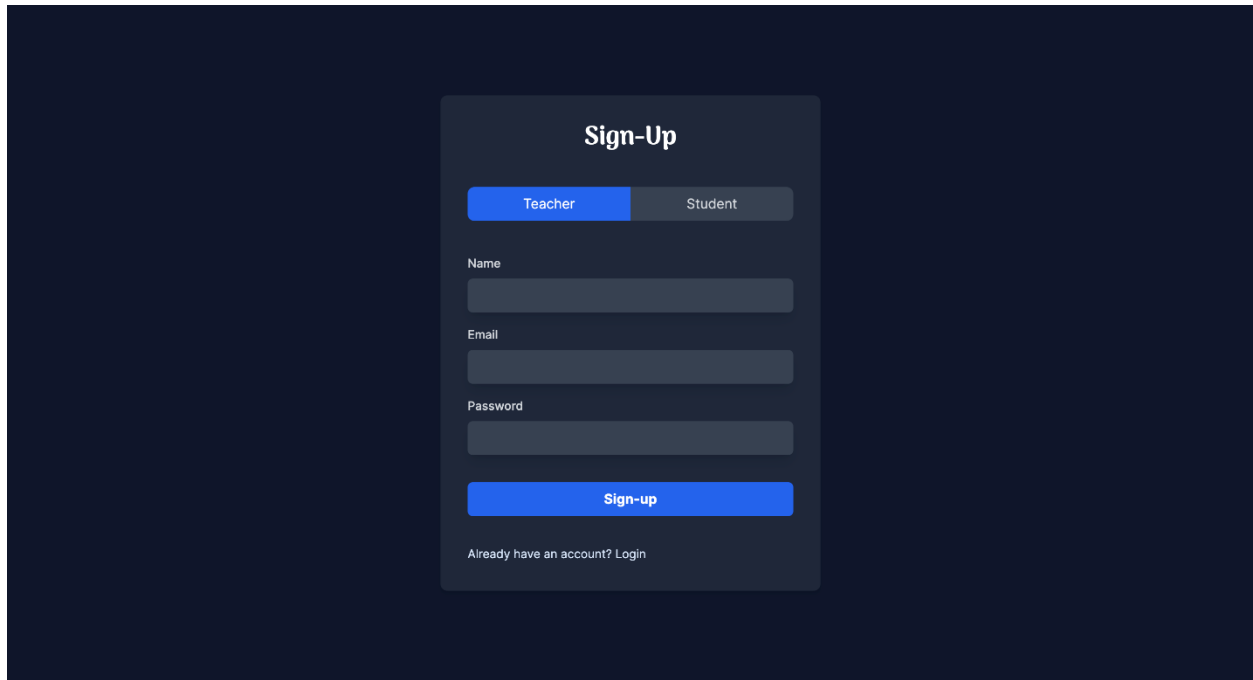


Figure 8 Teacher Dashboard - part 2

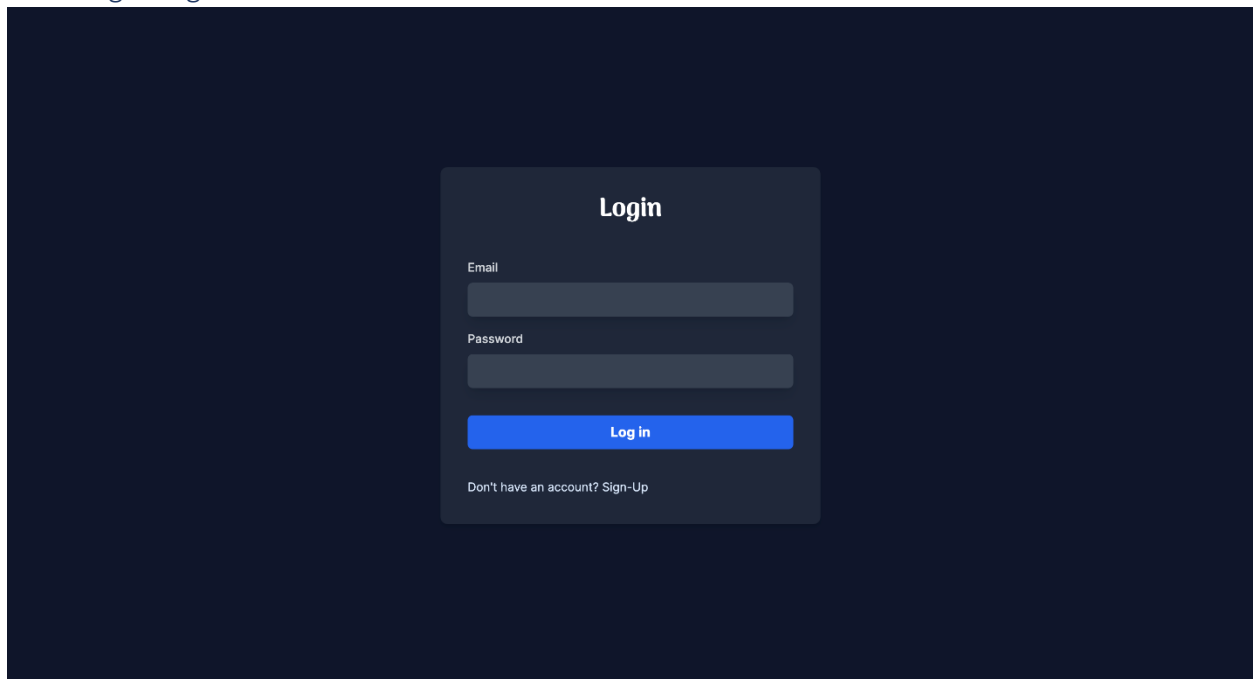
5.2.3 Sign-up Page



The image shows a dark-themed sign-up page. At the top center is a dark gray rounded rectangle containing the title "Sign-Up" in white. Below the title are two tabs: "Teacher" (highlighted in blue) and "Student" (dark gray). Under the tabs are three input fields labeled "Name", "Email", and "Password" in white text. Below these fields is a blue button with the text "Sign-up" in white. At the bottom of the form is a link that says "Already have an account? Login" in white text.

Figure 9 Sign-up Page

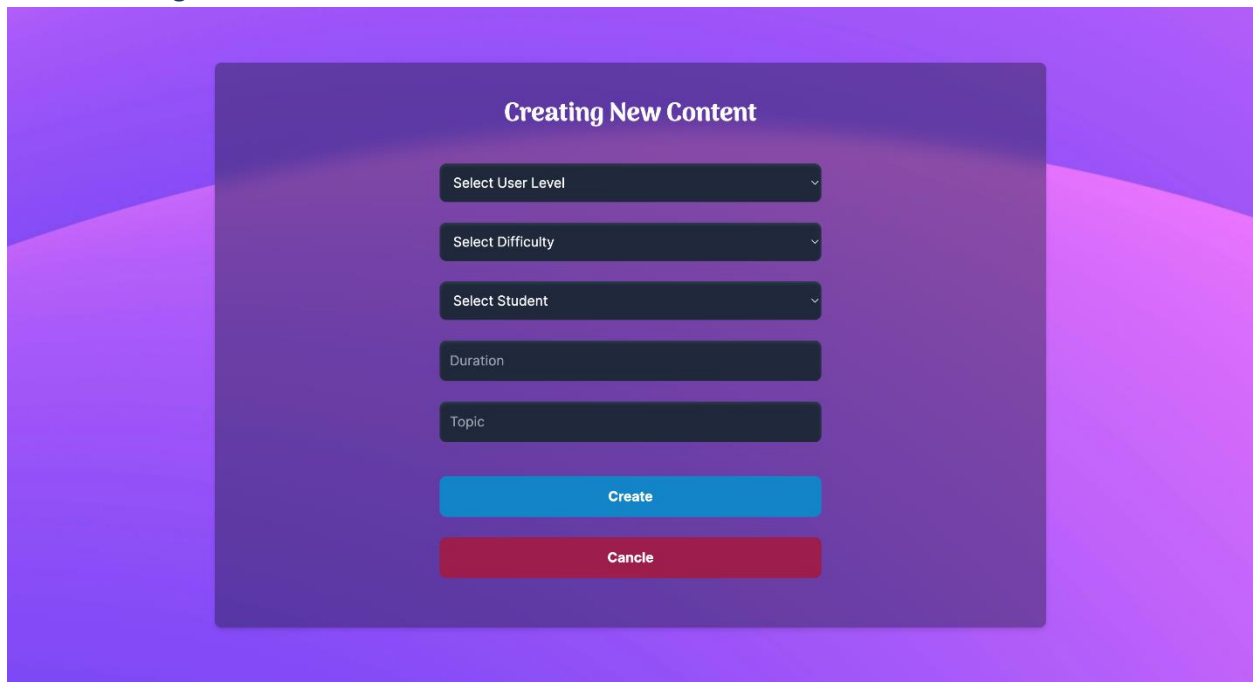
5.2.4 Login Page



The image shows a dark-themed login page. At the top center is a dark gray rounded rectangle containing the title "Login" in white. Below the title are two input fields labeled "Email" and "Password" in white text. Below these fields is a blue button with the text "Log in" in white. At the bottom of the form is a link that says "Don't have an account? Sign-Up" in white text.

Figure 10 Login Page

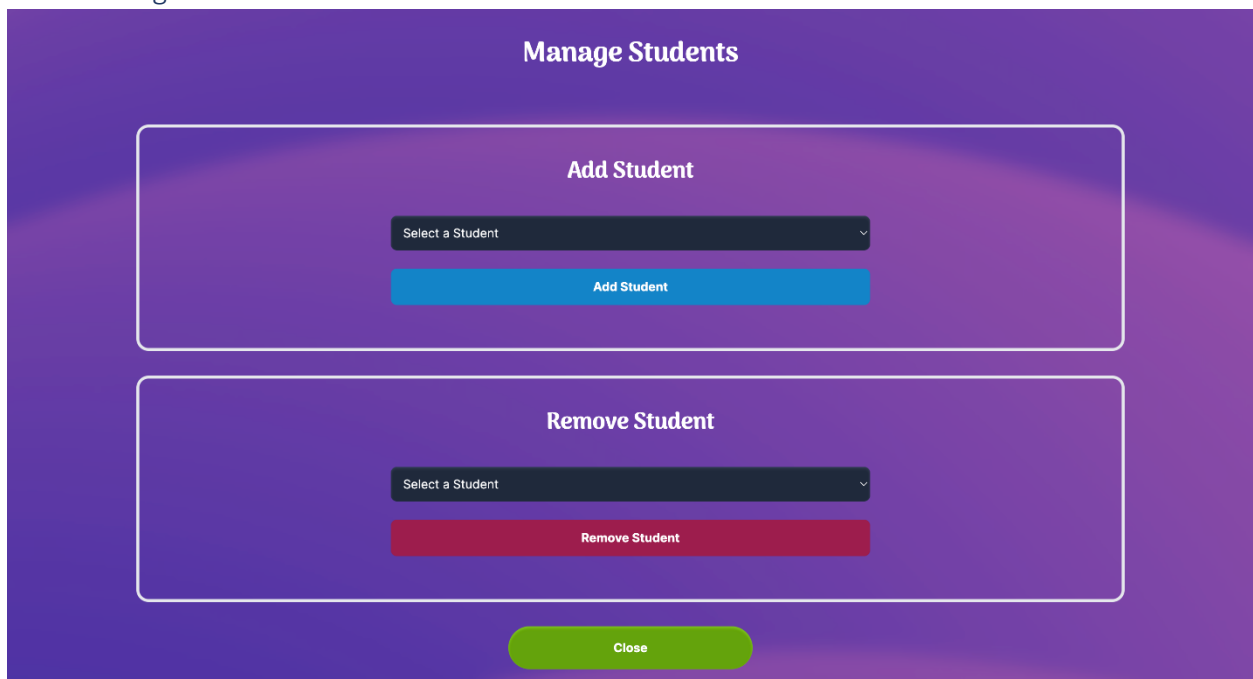
5.2.5 Creating New Content



A modal form titled "Creating New Content" with a dark purple background. It contains five input fields: "Select User Level", "Select Difficulty", "Select Student", "Duration", and "Topic". Each of the first three fields has a dropdown arrow on the right. Below the input fields are two buttons: a blue "Create" button and a red "Cancel" button.

Figure 11 Creating New Content

5.2.6 Manage Students



A modal form titled "Manage Students" with a dark purple background. It contains two sections, each with a title and a form:

- Add Student**: Includes a "Select a Student" dropdown and a blue "Add Student" button.
- Remove Student**: Includes a "Select a Student" dropdown and a red "Remove Student" button.

At the bottom of the modal is a green "Close" button.

Figure 12 Manage Students

5.2.7 Conversation Page

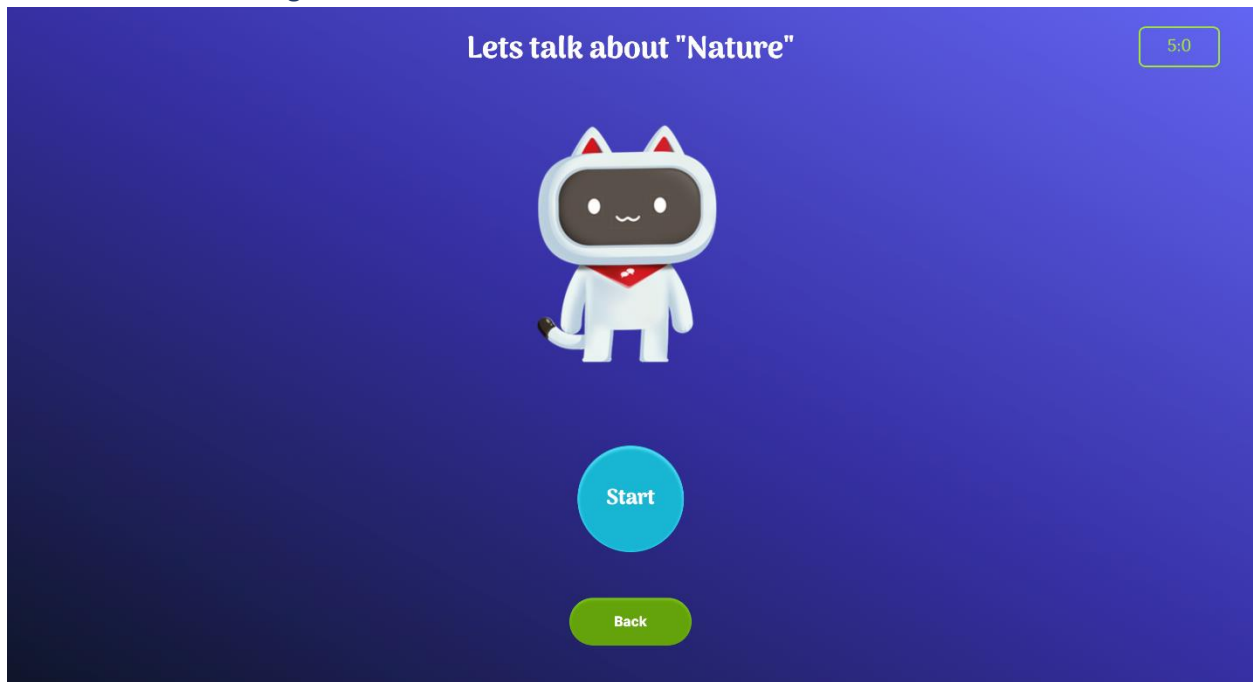


Figure 13 Conversation Page - part 1

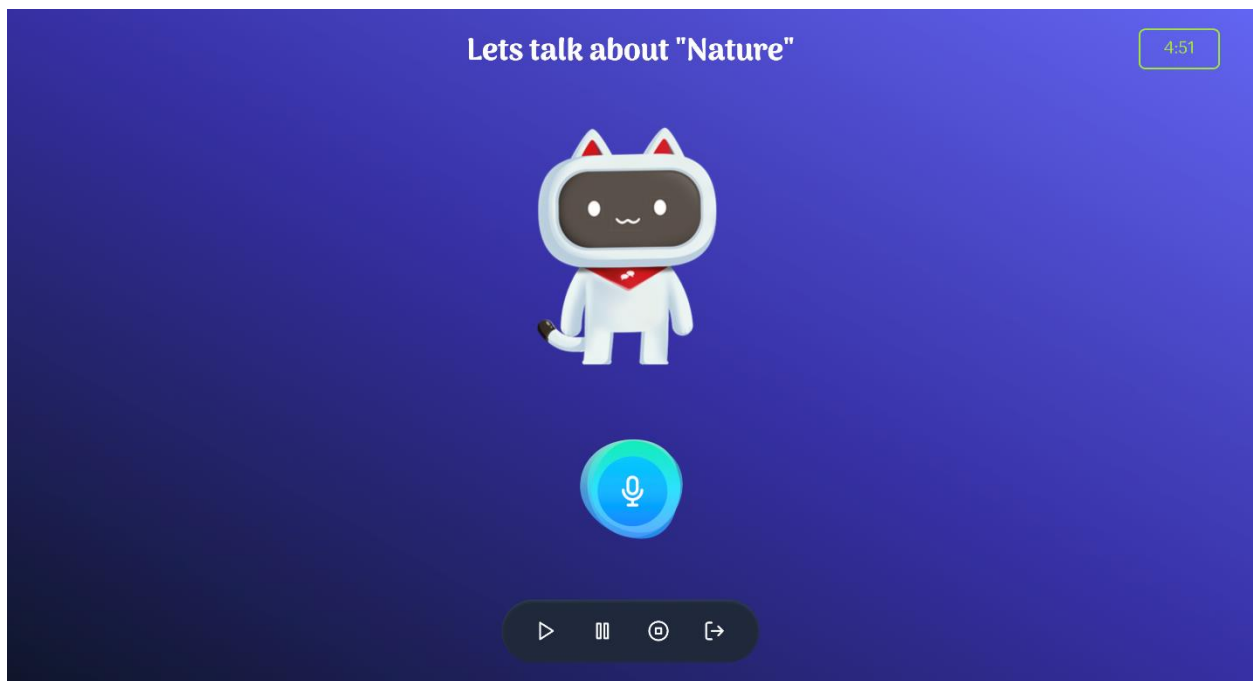


Figure 14 Conversation Page - part 2

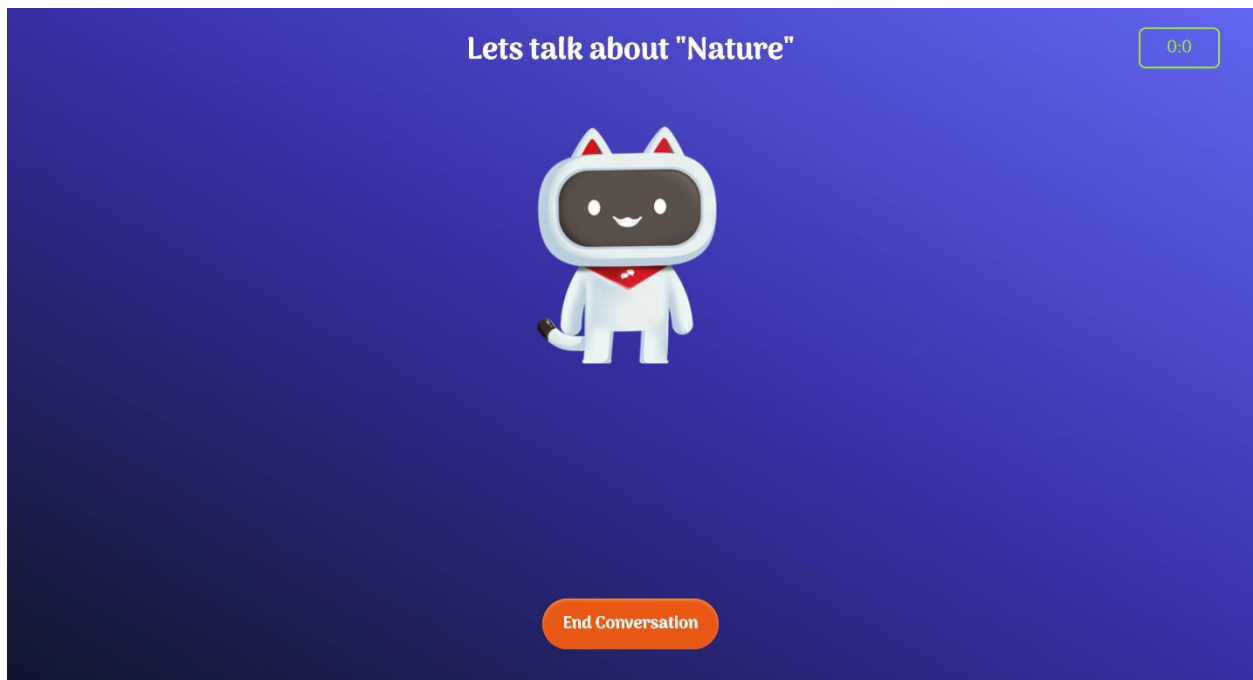


Figure 15 Conversation Page - part 3

5.2.8 Feedback Page

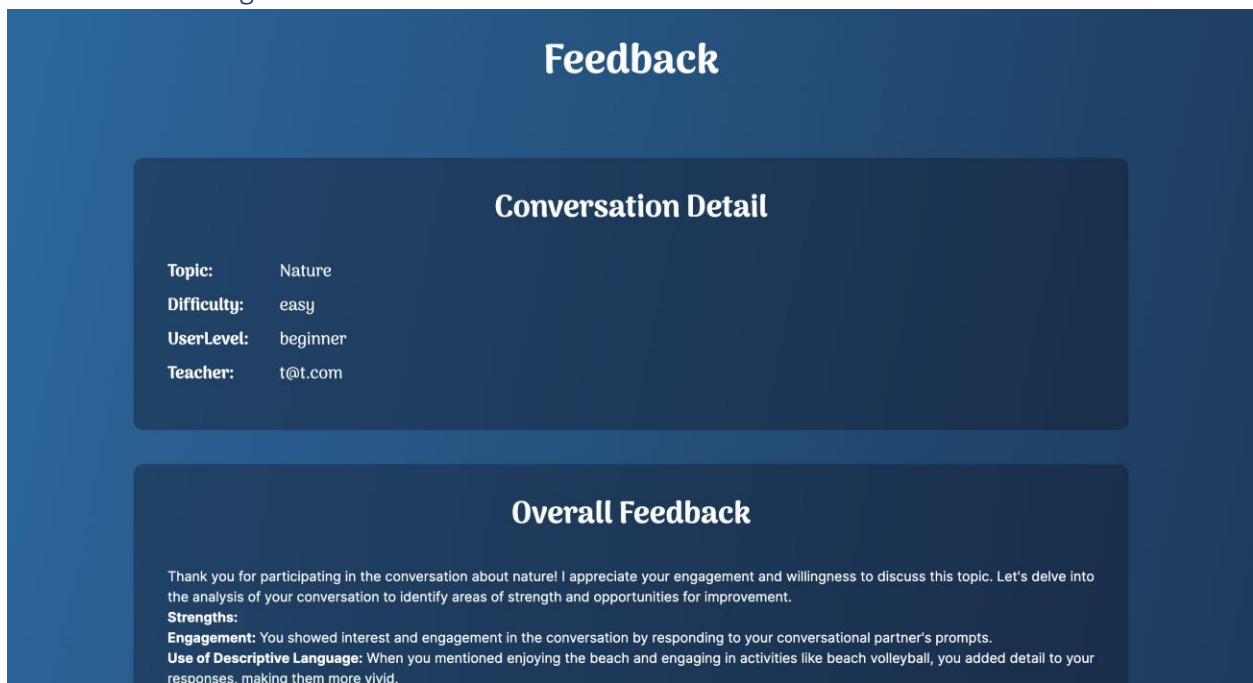


Figure 16 Feedback Page - part 1

Overall Feedback

Thank you for participating in the conversation about nature! I appreciate your engagement and willingness to discuss this topic. Let's delve into the analysis of your conversation to identify areas of strength and opportunities for improvement.

Strengths:

Engagement: You showed interest and engagement in the conversation by responding to your conversational partner's prompts.

Use of Descriptive Language: When you mentioned enjoying the beach and engaging in activities like beach volleyball, you added detail to your responses, making them more vivid.

Connection: You attempted to connect your responses to the topic of nature, even though there were some deviations.

Areas for Improvement:

Staying on Topic: There were moments where your responses shifted away from the topic of nature, like mentioning Siri or not elaborating on nature-related experiences. Try to maintain focus on the current subject to ensure a coherent conversation.

Completeness: Some of your responses were brief, making it challenging for your conversational partner to continue the discussion. Adding more details or examples can enrich the conversation.

Memory Recall: There were instances where you mentioned not remembering specific details about your experiences. Try to recall and share more specific anecdotes or feelings to deepen the conversation.

Actionable Suggestions:

Focus on Nature: When discussing a topic, ensure your responses relate directly to it. For instance, when talking about the beach, elaborate on how the sea or sand makes you feel connected to nature.

Elaborate and Share: Try to expand on your experiences or feelings related to nature. Describing specific moments or senses can make your responses more engaging and create a richer conversation.

Practice Recall: To enhance your conversational skills, practice recalling details of past experiences or preferences related to the topic at hand. This can help you engage more deeply in discussions.

Overall, your willingness to participate and share your thoughts is commendable. By focusing on staying on topic, providing more detailed responses, and improving memory recall in conversations, you can further enhance your communication skills. Keep up the enthusiasm for learning and engaging in conversations about various topics! If you have any questions or need further guidance, feel free to ask. Good luck with your future conversations!

Figure 17 Feedback Page - part 2

Details

Message 1:

Content: hello

Feedback:

Synonyms:

pronunciation:

AI Response: Hello! How are you today? Are you ready to talk about nature?

Message 2:

Content: yes let's go

Feedback:

Synonyms: go, proceed, advance, move

pronunciation:

AI Response: Great! Nature is so fascinating. What do you enjoy most about nature?

Message 3:

Figure 18 Feedback Page - part 3

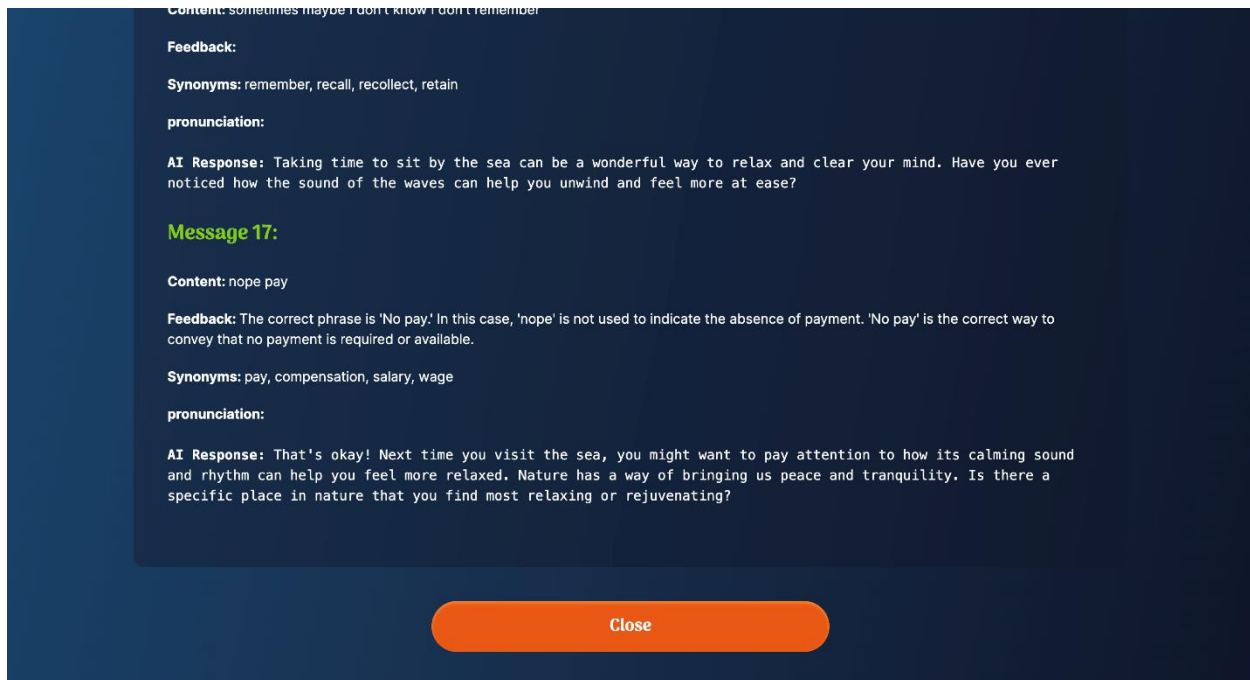


Figure 19 Feedback Page - part 4

6. Deployment Plan

6.1 Hosting and Infrastructure

- Front-End: Vercel
- Database: MongoDB Atlas

6.2 Continuous Integration/Continuous Deployment (CI/CD)

- Set up CI/CD pipelines for automated testing and deployment.
- Use GitHub Actions or Jenkins for continuous integration.

7. Maintenance and Support

7.1 Monitoring

- Implement monitoring tools to track application performance and user interactions.
- Regularly review logs and metrics for potential issues.

7.2 User Support

- Provide user support through email and in-app chat.
- Regularly update FAQs and documentation based on user feedback.