Multilingual Web Application for Learning and Assessment for Ethnic Minorities Students in Vietnam

Huynh Minh Tri

Le Quy Don High School for the Gifted, Da Nang, Vietnam

Email: trihuynh160107@gmail.com

ABSTRACT

This study presents the development of a multilingual web application aimed at enhancing accessibility and educational outcomes for ethnic minority students in Vietnam. By addressing linguistic barriers, the application offers adaptive learning resources, personalized competency assessments, and AI-driven content suggestions. Key features include multilingual learning materials, a reporting system for progress tracking, and adaptive tests with personalized feedback. Built using React.js, Python, and MongoDB, this scalable platform provides data-driven insights for teachers and administrators while empowering students and parents with tools for improved educational engagement.

Keywords: Web applications, multilingualism, minority students, adaptive learning, competency assessment, AI in education, educational technology.

1. INTRODUCTION

1.1 Research Objectives

The primary objectives of this research are centered around addressing critical challenges in education and leveraging innovative solutions to make a meaningful impact.

One of the key goals is to enhance accessibility to quality education for ethnic minority students in Vietnam. These communities often face significant barriers, including geographical isolation, linguistic diversity, and limited access to educational resources. By developing a multilingual platform and incorporating digital tools, this research aims to bridge these gaps and create an inclusive environment where every student can thrive, regardless of their background or circumstances.

Another important objective is to improve learning outcomes by employing adaptive assessments and multilingual resources. Adaptive assessments allow for personalized evaluations that cater to individual learning paces and styles, ensuring that students receive support tailored to their unique needs. Meanwhile, multilingual resources address language barriers, enabling students to engage with educational content in their native languages while gradually building proficiency in others. Together, these features foster deeper understanding, better retention, and enhanced academic performance.

Additionally, this research seeks to provide data-driven insights for educators to refine their teaching strategies. By analyzing patterns and trends in student performance, the system can generate actionable recommendations for educators, helping them identify areas where students may need extra guidance or resources. This evidence-based approach empowers teachers to make informed decisions and adopt strategies that are both effective and responsive to the diverse needs of their classrooms.

Through these objectives, the research aims to transform the educational landscape for ethnic minority students, promoting equity, engagement, and excellence in learning.

1.2 Context and Significance of the Issue

Ethnic minority students in Vietnam face significant educational challenges, primarily due to linguistic barriers. These obstacles contribute to lower educational attainment and limit social mobility. Addressing these barriers is crucial for fostering equality in education and unlocking the potential of this demographic. By leveraging modern web technologies and AI, this research aims to bridge the gap and promote inclusive learning environments.

1.3 Reason for Choosing the Topic

The choice of this topic is rooted in several compelling reasons. First and foremost, there exists an urgent need to address the persistent educational disparities faced by ethnic minorities. Many communities continue to struggle with unequal access to quality education, which perpetuates cycles of poverty and limits opportunities for personal and societal development. This challenge demands innovative solutions that go beyond traditional approaches.

Additionally, the increasing role of technology in transforming education cannot be overlooked. With the rapid advancements in digital tools and platforms, technology has become a powerful enabler, bridging

gaps in resources, geographical barriers, and linguistic differences. It offers the potential to create inclusive learning environments that cater to diverse needs and backgrounds.

Furthermore, the integration of artificial intelligence (AI) and multilingual platforms holds immense promise for revolutionizing both accessibility and the personalization of learning experiences. AI-driven systems can adapt to individual learning paces and styles, while multilingual platforms ensure that language is no longer a barrier to acquiring knowledge. These combined capabilities make it possible to empower ethnic minority students with tools tailored to their unique contexts, fostering greater engagement and meaningful learning outcomes.

This multifaceted approach underscores the importance of leveraging modern technology to address long-standing challenges in education, particularly for underrepresented and underserved communities.

2. THEORETICAL OVERVIEW

The theoretical foundation of this research combines principles from educational technology, linguistics, and adaptive learning:

Educational Technology: Educational technology focuses on improving learning outcomes through digital tools and platforms. Research has demonstrated that digital platforms not only enhance access to learning materials but also introduce innovative teaching methods, boosting engagement and learning motivation among students.

Linguistics: Linguistics plays a critical role in enabling students to access educational content in their native languages. Providing learning materials in familiar languages reduces psychological stress, enhances comprehension, and improves learning outcomes. This study emphasizes developing multilingual learning content to meet the needs of ethnic minority students.

Adaptive Learning: Adaptive learning is a methodology that personalizes the learning process based on individual needs and capabilities. Adaptive learning systems employ AI algorithms to analyze learner data, adjusting content, pace, and teaching methods to optimize learning outcomes. This approach not only helps students grasp knowledge more effectively but also encourages active participation in the learning process.

By combining these principles, this research creates a multilingual educational application that helps students overcome language barriers while enhancing overall learning experiences, promoting equity and efficiency in education.

3. METHODOLOGY

3.1 Application Development

The application was developed using a carefully selected technology stack to ensure functionality, scalability, and user-friendliness. For the frontend, React.js was chosen as the framework to create dynamic and responsive user interfaces. Its component-based architecture and extensive ecosystem

allowed for the development of a seamless and engaging user experience, catering to the needs of modern web applications.

On the backend, Python was utilized to handle server-side logic efficiently. Known for its readability and versatility, Python facilitated the creation of scalable and robust backend systems. Its wide array of libraries and frameworks ensured streamlined development processes and the ability to handle complex functionalities with ease.

For data storage, MongoDB served as the database solution, offering a flexible and schema-less structure that perfectly suited the application's evolving requirements. Its NoSQL nature allowed for efficient management of unstructured and semi-structured data, ensuring both scalability and speed. Together, these technologies formed a cohesive and powerful stack, driving the application’s performance and reliability.

3.2 Features and Functionalities

Multilingual Learning Materials: The platform offers educational resources in multiple languages to ensure that students from various linguistic backgrounds can access content in their native language. This approach makes learning more inclusive, allowing students to engage with materials more effectively and at their own pace.

Competency Assessment: Adaptive tests are designed to assess students' competencies in different subjects. These assessments adjust in difficulty based on the student’s performance, ensuring that the tests are personalized to accurately reflect each learner’s knowledge and skill level. Immediate, personalized feedback is provided to help students identify strengths and areas that need improvement.

Reporting System: A comprehensive reporting system enables teachers and administrators to track and monitor student progress over time. Detailed reports show how students are performing on various tasks, assessments, and learning modules, providing a clear picture of their development and any challenges they may be facing.

Display Learning Content: The platform showcases educational materials, such as lectures, tutorials, and interactive exercises, in a user-friendly interface. This content is accessible anytime, allowing students to explore topics in-depth and at their own pace.

Content Review: After engaging with learning materials, students can review the content they’ve learned through quizzes, practice tests, or summaries. This step ensures that key concepts are reinforced and understood before moving on to new material, improving retention and comprehension.

Test Results Reporting: Upon completing an assessment, students are presented with detailed test results. These reports include information on correct and incorrect answers, overall scores, and areas of strength. This transparency helps students understand their performance and where to focus their efforts next.

Results Analysis: Test results are thoroughly analyzed to provide valuable insights for educators and administrators. This analysis helps identify trends in student performance, highlight common challenges, and inform teaching strategies, ensuring that learning outcomes are continuously optimized.

3.3 Target Users

- **Students:** Access learning materials in native languages.
- **Teachers:** Track progress, manage multilingual tests.
- **Education Managers:** Monitor and evaluate educational quality.
- **Parents:** Monitor children's progress and support learning.

3.4 Benefits

- **Enhanced Accessibility:** Overcome language barriers.
- **Improved Learning Outcomes:** Adaptive assessments and feedback.
- **Data-Driven Insights:** Supports teaching strategies.

4. SYSTEM ARCHITECTURE

4.1 User Interface Design

Home Page: Displays platform information and login/registration options, emphasizing ease of access for users of diverse backgrounds.

Exercise Page: Presents test questions with a timer, real-time progress indicators, and immediate feedback.

Teacher Management Page: Enables educators to upload materials, create tests, and manage multilingual content efficiently.

Admin Page: Facilitates user account management, test oversight, and multilingual configurations.

4.2 Data Flow

The platform operates seamlessly through a structured workflow that integrates frontend interaction, backend processing, and AI analytics to provide a personalized and effective user experience.

The interaction begins at the frontend, where users engage with the platform through an intuitive and dynamic interface. Whether entering data, navigating content, or completing tasks, every user action triggers requests to the backend. This layer ensures a smooth and responsive interaction that meets user expectations for modern web applications.

Once a request reaches the backend, the server logic processes it efficiently. Using Python-based solutions, the backend retrieves or updates data in MongoDB, the platform’s chosen database. MongoDB’s schema-less design accommodates diverse and evolving data structures, while robust mechanisms in the backend ensure data integrity and consistency throughout these operations.

To elevate the learning experience further, the platform integrates advanced AI analytics. These modules analyze user performance data in real time, identifying patterns, strengths, and areas for improvement. Based on these insights, the AI system tailors content delivery to match individual learning needs and

preferences. This adaptive approach not only enhances learning effectiveness but also fosters greater engagement by making the experience more relevant and personalized for each user.

By combining these three layers, the platform achieves a harmonious balance between user-centric design, reliable data management, and cutting-edge intelligence, ensuring a transformative and impactful educational journey.

5. RESULTS AND DISCUSSION

5.1 Key Outcomes

Improved Accessibility: Students could engage with materials in their native languages, fostering confidence and understanding.

Enhanced Engagement: The adaptive learning model motivated consistent participation by offering challenges aligned with student proficiency levels.

Valuable Insights: Teachers and administrators gained actionable data to refine curricula and target learning interventions effectively.

5.2 Challenges

Localization Efforts: Translating content into multiple languages required significant effort. Collaboration with local educators helped address linguistic nuances.

Performance Optimization: Real-time adaptability demanded high computational efficiency. Optimization techniques, including caching and efficient algorithm design, ensured responsiveness.

6. SCALABILITY AND FUTURE DIRECTIONS

Additional Exam Preparation Features: Including modules for standardized test preparation.

Enhanced AI Integration: Further refining recommendation algorithms to deliver even more personalized learning experiences.

Global Application: Adapting the system for use in other regions facing similar linguistic challenges.

7. CONCLUSION

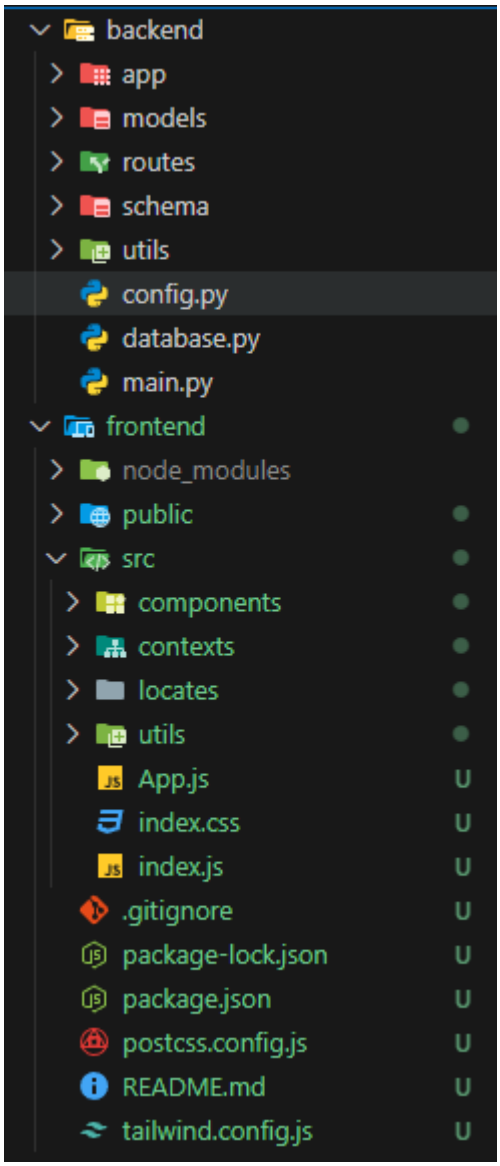
This research highlights the transformative potential of multilingual web applications in the field of education, particularly in addressing the needs of ethnic minority students in Vietnam. The integration of advanced technologies such as Artificial Intelligence (AI) and machine learning into the platform enables a more personalized and inclusive learning experience. By bridging language barriers and enhancing accessibility, the platform offers an opportunity to overcome systemic educational challenges faced by marginalized groups.

The development of this platform has demonstrated how digital solutions can be leveraged to foster more equitable educational opportunities, ensuring that students from diverse linguistic and cultural

backgrounds can fully participate in the learning process. Moving forward, the platform will undergo further enhancement to increase its functionality, improve user engagement, and ensure its scalability to reach a wider audience. By continually refining these features, the goal is to maximize the platform's educational impact, promote long-term learning outcomes, and contribute to the overall improvement of education for ethnic minority students across Vietnam.

8. APPENDICES

8.1 Structure of the project



backend/:

- This folder contains the **server-side logic** of the application.
- It handles tasks like managing the database, defining API routes, processing business logic, and responding to requests from the frontend.
- Essentially, it's where the core application logic and data handling occur.

frontend/:

- This folder contains the **client-side code** of the application.
- It focuses on building the user interface (UI) using ReactJS and handles how users interact with the application.

- It communicates with the backend to display or send data, ensuring a seamless user experience.

In the frontend folder, we have the App.js file. The App.js file is the main entry point for rendering and structuring the frontend application. It manages routing, context providers, and overall layout.

It provides:

- **Navigation** (with Routes and react-router-dom).
- **Dynamic layout** (e.g., conditionally showing Header).
- **Context management** (language support via LanguageProvider).

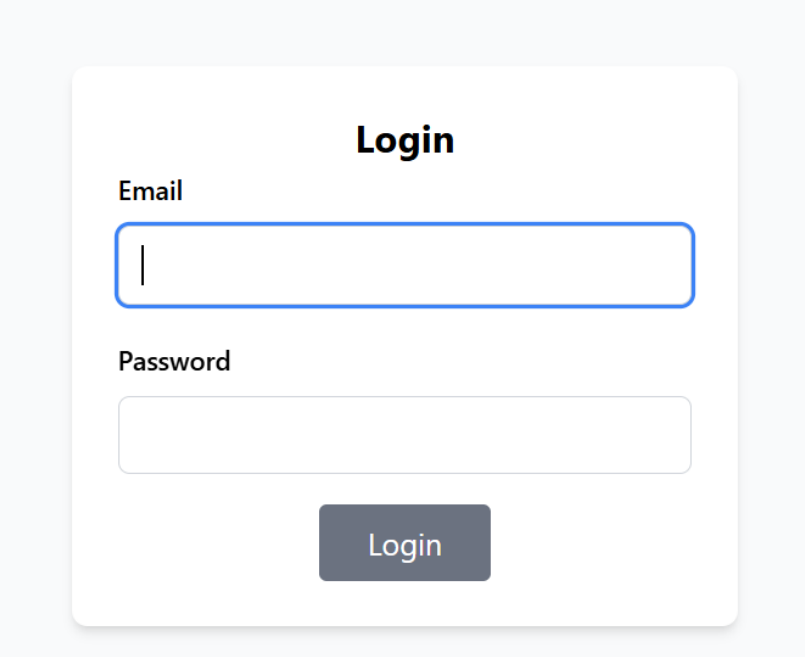
```jsx
import React from "react";
import {
  BrowserRouter as Router,
  Route,
  Routes,
  useLocation,
} from "react-router-dom";
import Header from "./components/Header";
import Login from "./components/Login";
import StudentDashboard from "./components/StudentDashboard";
import TeacherDashboard from "./components/TeacherDashboard";
import Quiz from "./components/Quiz";
import CreateQuiz from "./components/CreateQuiz";
import { LanguageProvider } from "./contexts/LanguageContext";
import Chatbot from "./components/Chatbot";

Tabnine | Edit | Test | Explain | Document | Ask
function AppContent() {
  const location = useLocation();
  const isLoginPage = location.pathname === "/";
  return (
    <div className="min-h-screen bg-gray-50">
      {!isLoginPage && <Header />}

      <main className="container mx-auto px-4 py-8">
        <Routes>
          <Route path="/" element={<Login />} />
          <Route path="/student-dashboard" element={<StudentDashboard />} />
          <Route path="/teacher-dashboard" element={<TeacherDashboard />} />
          <Route path="/quiz/:id" element={<Quiz />} />
          <Route path="/create-quiz" element={<CreateQuiz />} />
        </Routes>
        <Chatbot />
      </main>
    </div>
  );
}
Tabnine | Edit | Test | Explain | Document | Ask
function App() {
  return (
    <LanguageProvider>
      <Router>
        <AppContent />
      </Router>
    </LanguageProvider>
  );
}
export default App;
```

8.2 Main interfaces

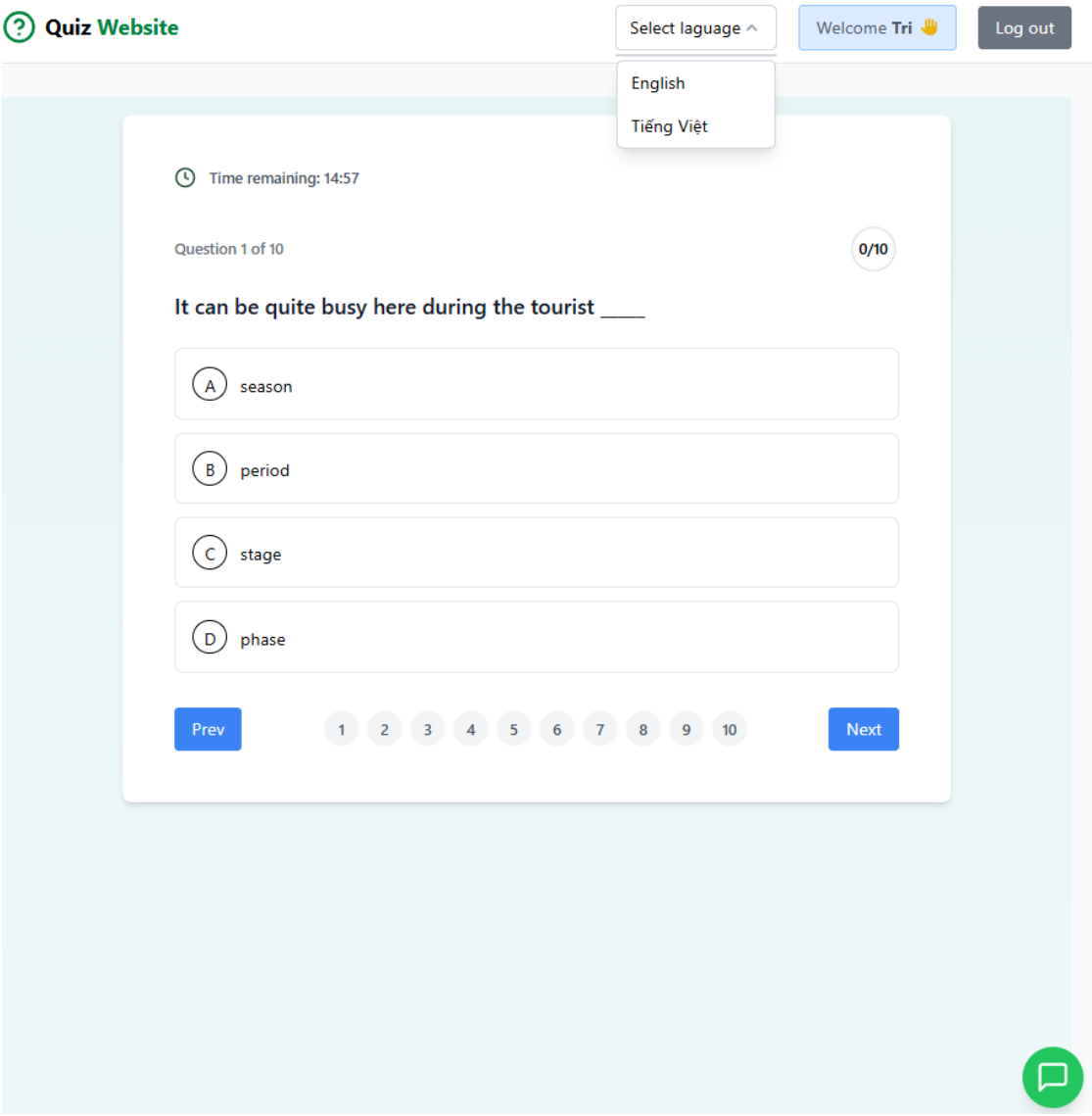
8.2.1 Login page



The login interface serves as the entry point for users to access the website's features. It allows users to authenticate themselves by entering their email and password. Designed with simplicity and security in mind, the interface ensures a smooth user experience while safeguarding sensitive information.

Upon successful authentication, users are seamlessly redirected to the quiz page. This transition marks the beginning of their interactive journey on the platform, enabling them to engage with the quizzes and learning materials provided. The login functionality not only grants secure access but also personalizes the user experience by associating their progress and preferences with their account. This setup ensures a streamlined and user-centric navigation process throughout the platform.

8.2.2 Quiz page



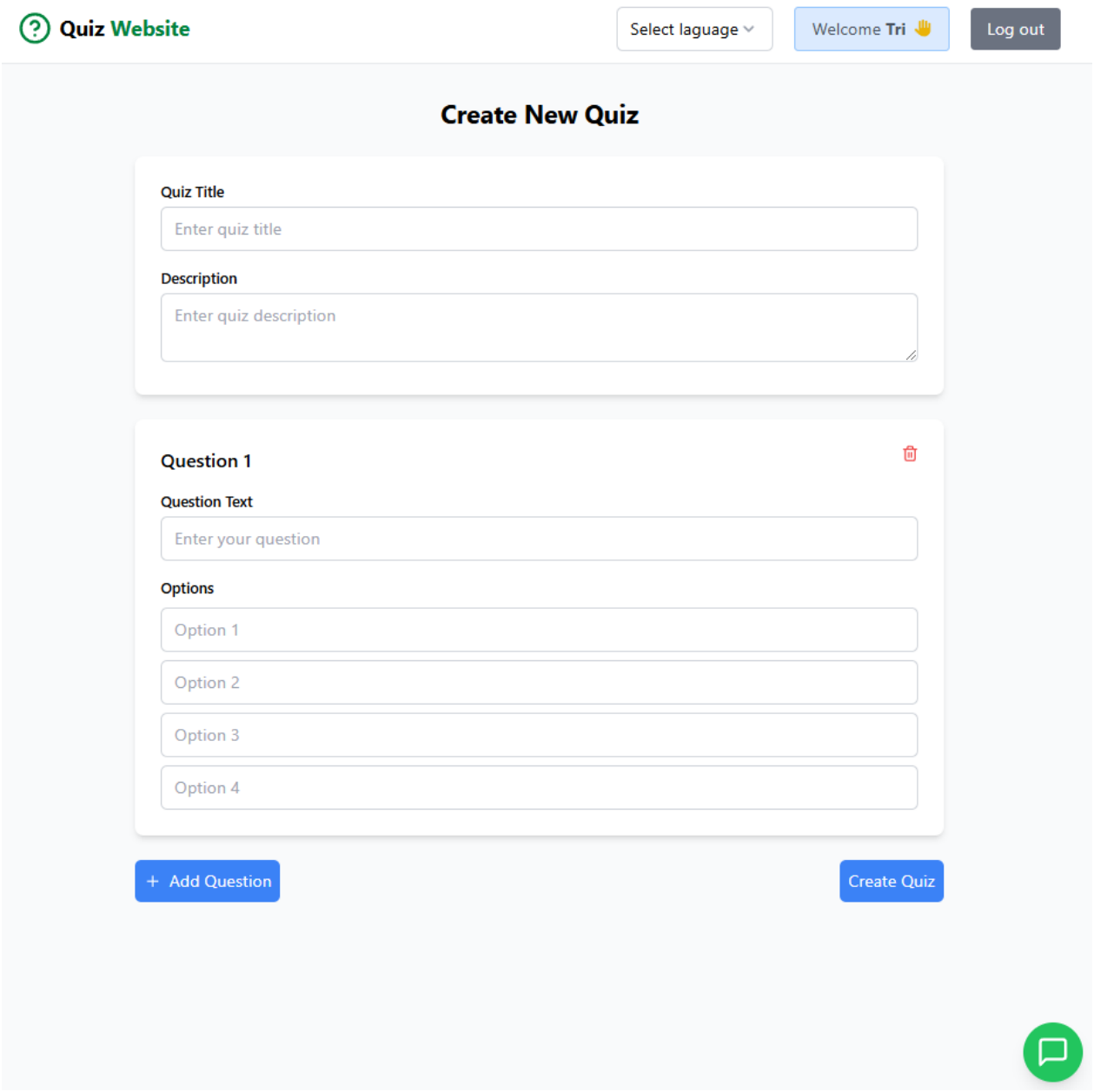
The quiz screen offers students an intuitive interface to complete assessments prepared by their teacher. The screen displays a series of questions tailored to the lesson or subject matter, allowing students to work through them at their own pace.

In addition to its core functionality, the test screen supports multilingual options, enabling students to switch between languages effortlessly. This feature ensures that students from diverse linguistic backgrounds can interact with the content in their preferred language, breaking down language barriers and enhancing comprehension.

Once students have answered the questions, they can press the submit button to finalize their responses. The system then processes the submitted answers, evaluates them, and calculates the results. These results are promptly displayed on the screen, giving students immediate feedback on their performance.

By combining multilingual capabilities with real-time feedback, the platform not only fosters a seamless and inclusive testing experience but also empowers students with the tools they need to succeed, regardless of their language proficiency.

8.2.3 Create new quiz page



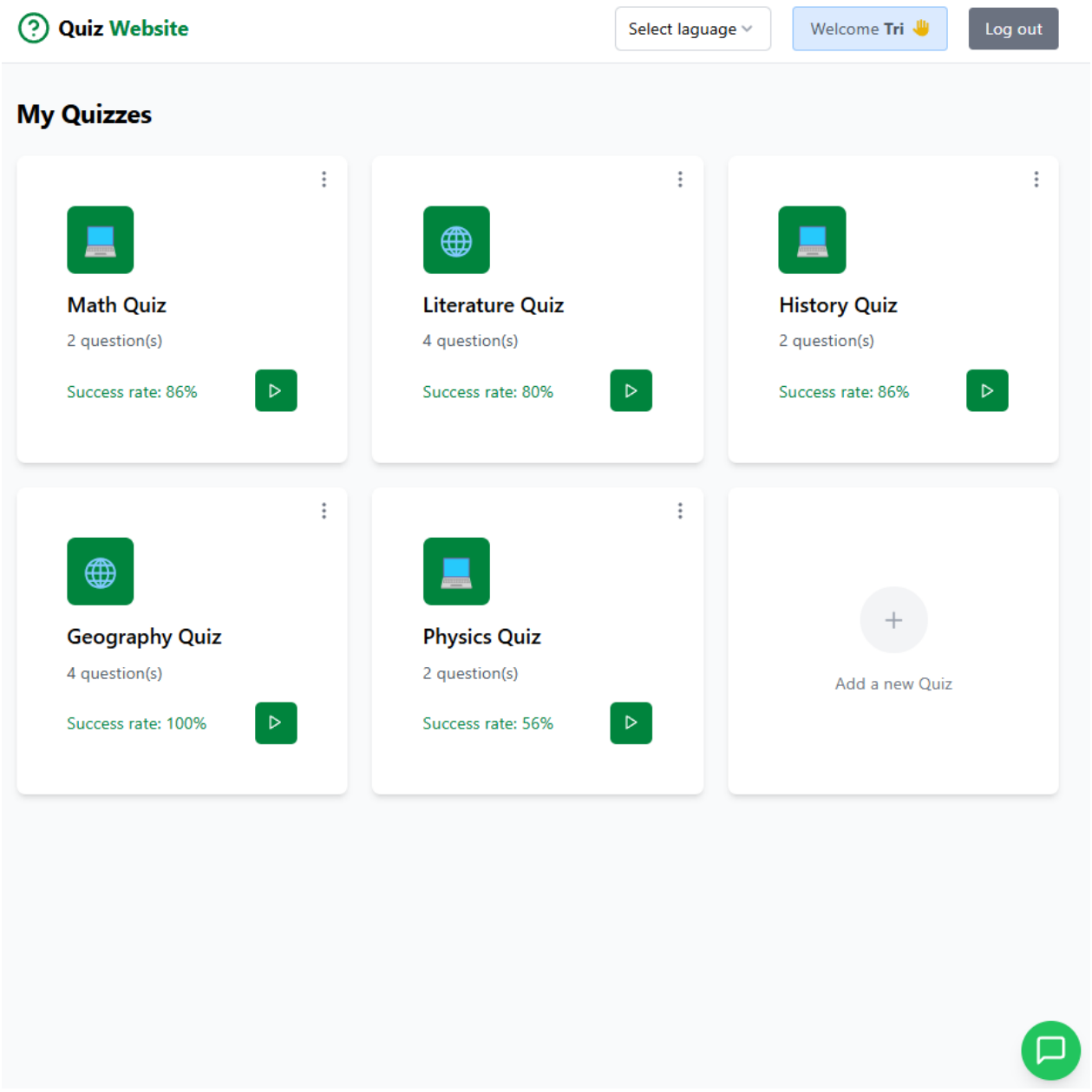
This screen provides teachers with the ability to create and manage question sets for various topics. It offers an intuitive interface where teachers can easily input questions, along with multiple-choice options or other types of answers, depending on the desired format.

Teachers can categorize these question sets by subject, difficulty level, or any other relevant criteria, making it easier to organize and retrieve them when needed. The interface also allows for the addition of explanations or hints for each question, enhancing the learning experience for students.

In addition to creating questions, teachers can customize settings such as time limits, number of attempts allowed, and whether or not to provide immediate feedback after each question. This level of control ensures that the assessments are tailored to the learning objectives and the specific needs of the students.

Overall, this screen equips teachers with a flexible and powerful tool for generating high-quality assessments, enabling them to monitor student progress and adapt their teaching strategies accordingly.

8.2.4. Student Dashboard page

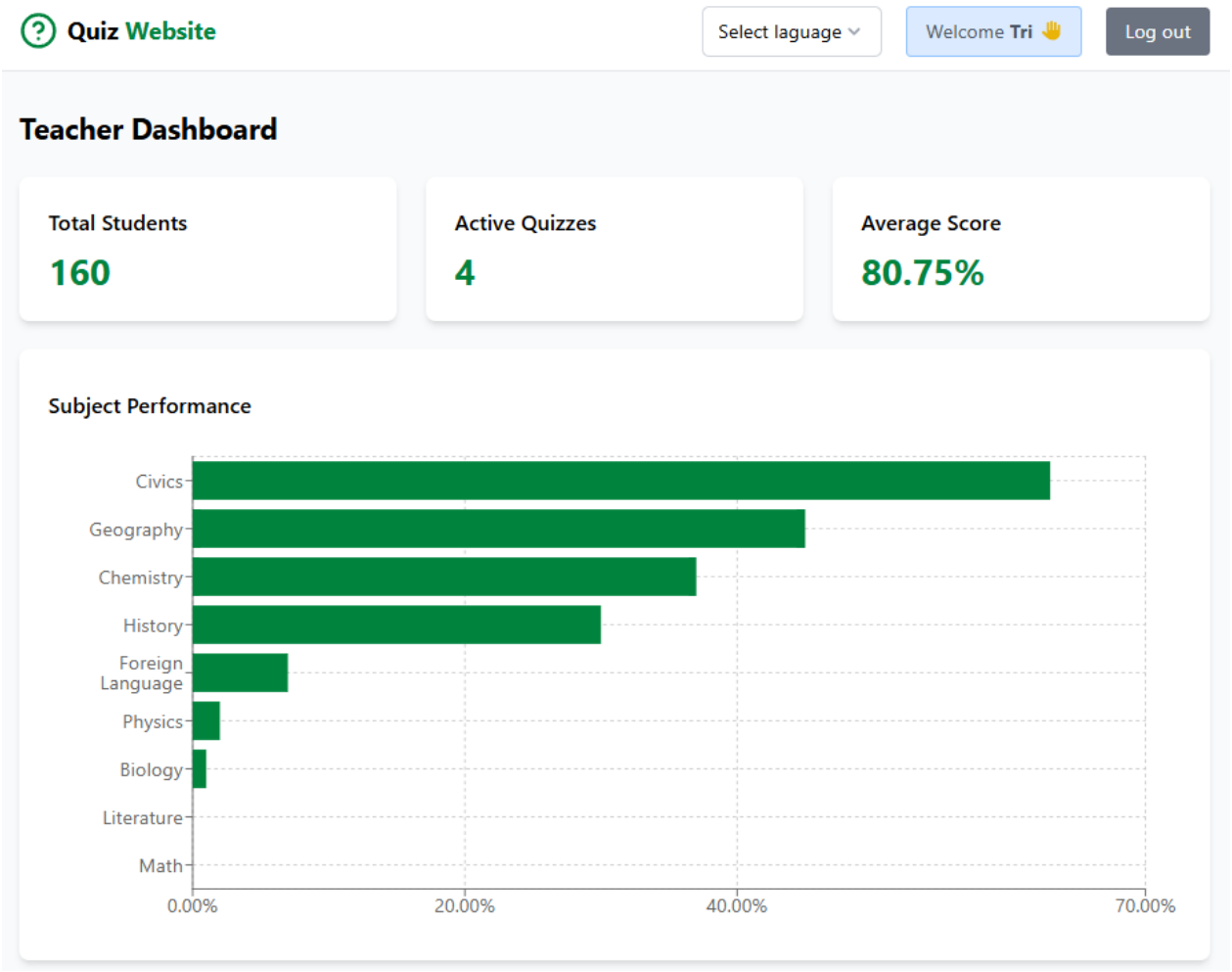


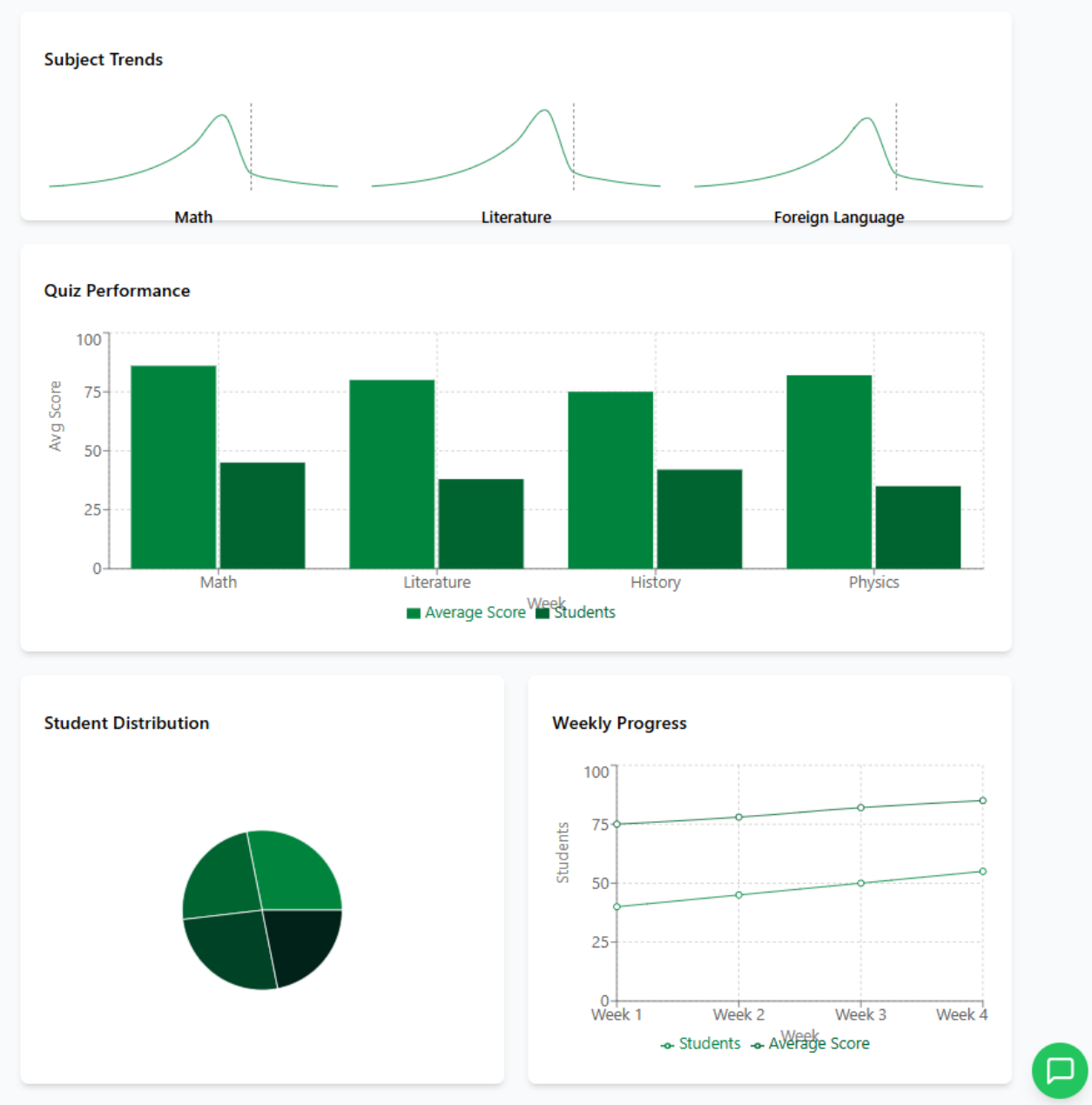
This screen serves as the central hub for students to access question sets, where they can work on assessments prepared by their teachers. Students are presented with a clear, organized list of available question sets, sorted by topic, difficulty, or any other relevant category, making it easy to navigate and select the appropriate quiz or exercise to complete.

In addition to displaying question sets for students, this screen also provides teachers with the functionality to create and edit question sets directly. Teachers can add new questions, modify existing ones, or update the settings for each set, such as time limits, question types (multiple-choice, short answer, etc.), and feedback preferences. This ability to create and edit question sets ensures that teachers have full control over the content and can make adjustments as needed to meet the evolving needs of their students.

With this dual functionality, the screen not only supports students in their learning process but also empowers teachers to customize and fine-tune the assessments, ensuring that the content remains relevant, engaging, and aligned with educational goals.

8.2.5 Teacher Dashboard page





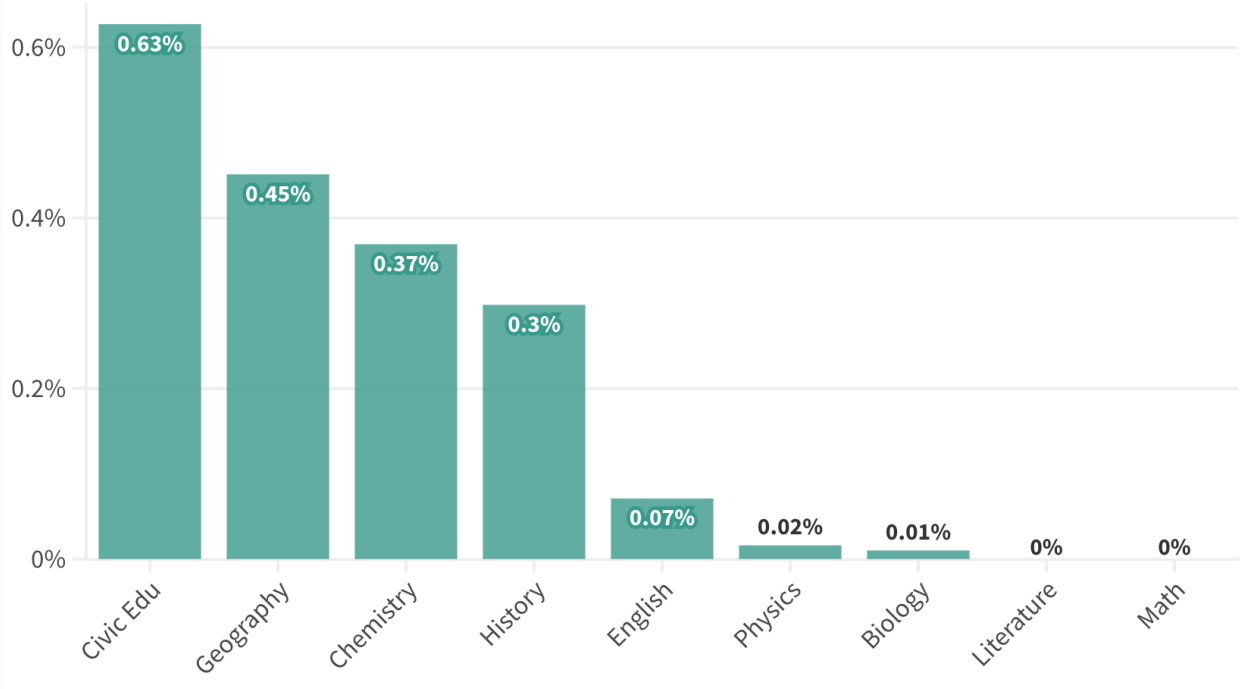
On this screen, teachers have the capability to collect and analyze important information about their students' scores and overall learning quality. The interface presents detailed performance data, including individual question scores, overall test results, time taken to complete each question, and any areas where students may have struggled. This information is gathered in real-time as students complete their assessments, providing teachers with immediate insights into student progress.

In addition to individual results, the screen may also offer data visualizations such as graphs or charts, summarizing trends and patterns in the class's performance. Teachers can filter the data based on different criteria, such as specific topics, question types, or even compare results across different student groups.

These analytics enable teachers to identify learning gaps, assess the effectiveness of their teaching strategies, and make data-driven decisions for future lessons. This comprehensive view of students' performance allows for more personalized feedback, tailored interventions, and a deeper understanding of each student's strengths and areas for improvement.

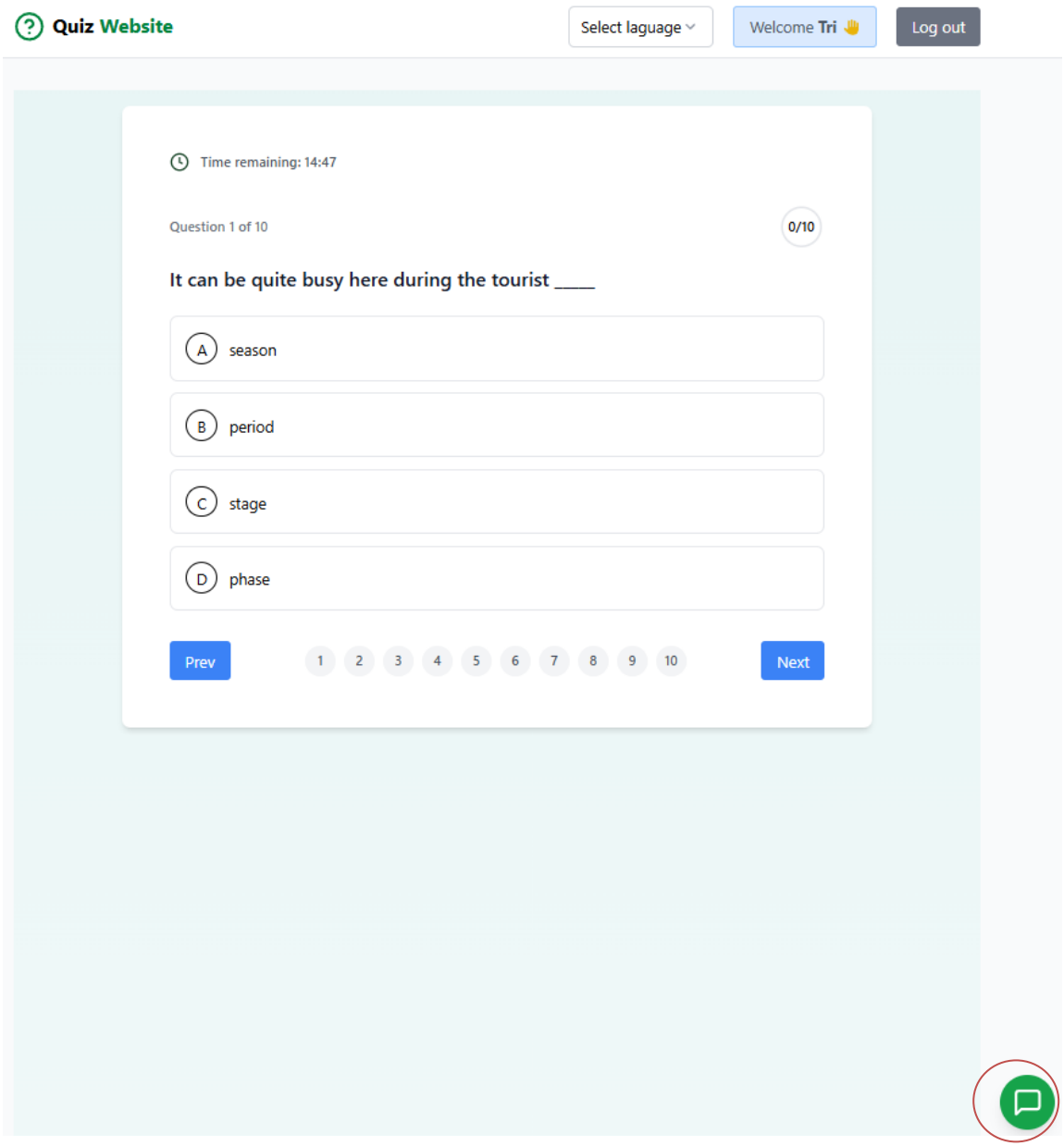
8.2.6 Education Managers

This is an example of how educational managers can rely on the chart to gain specific insights:

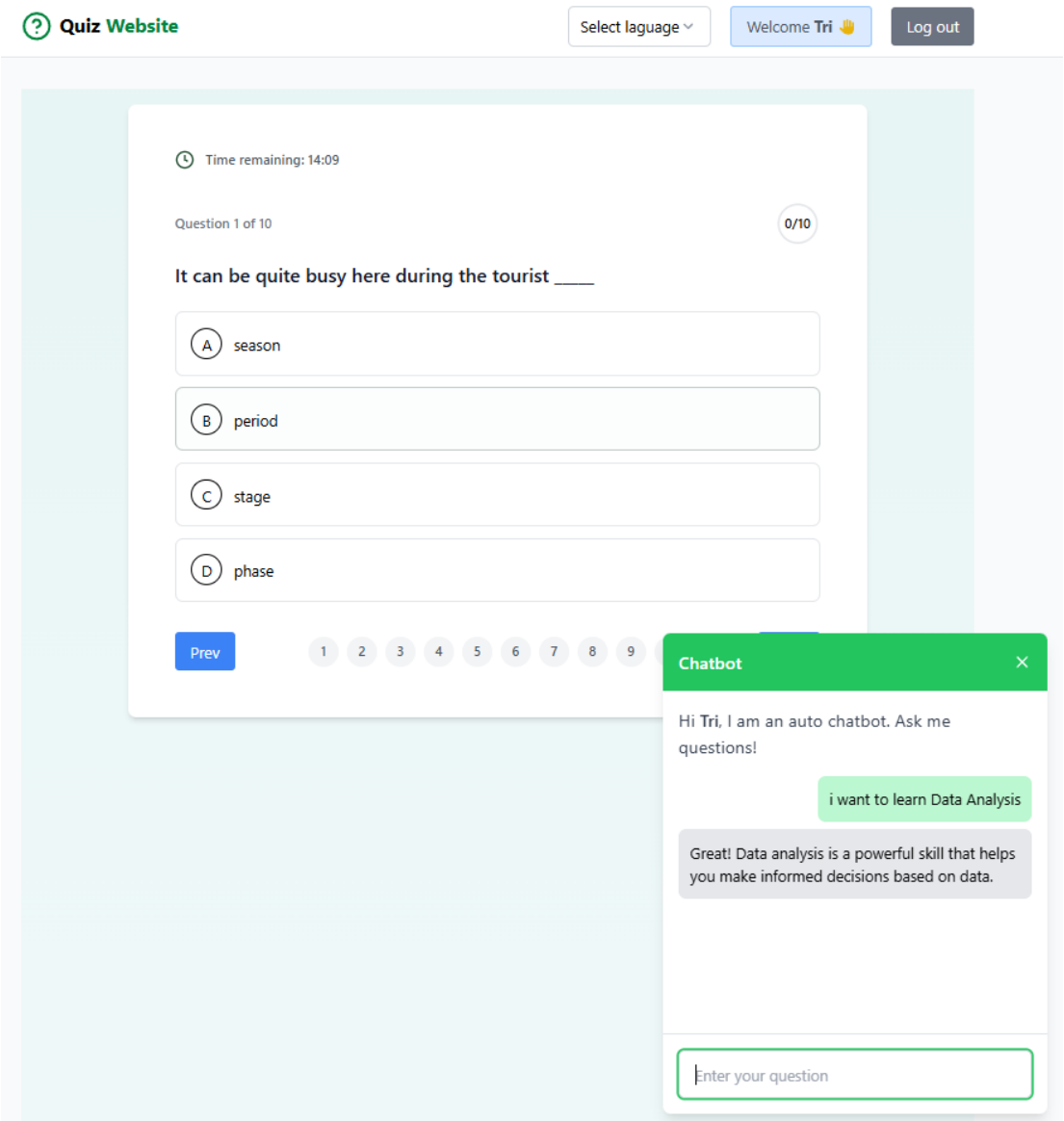


Based on the chart, the percentage of candidates achieving a perfect score of 10 in the graduation exam shows a significant difference across subjects. Civic Education has the highest percentage (0.63%), followed by Geography (0.45%) and Chemistry (0.37%). Meanwhile, subjects such as Mathematics, Literature, and Biology have virtually no candidates achieving a perfect score. This reflects a disparity in the level of difficulty among subjects or the candidates' preparation. Civic Education tends to be easier to score highly in, possibly due to its practical content and less demand for in-depth analytical thinking compared to natural sciences like Mathematics or Physics. However, further analysis is needed to accurately assess the factors influencing these results.

8.2.7 Chatbot



When users click on the chatbot icon, the chat window will immediately appear, providing a direct communication interface with the system. This chat window is designed to be simple and user-friendly, displaying messages from the chatbot as well as questions or requests from the user.



Inside the chat window, users can type in their questions or requests, and the chatbot will respond instantly with useful information, answers, or assistance with their learning process. The chat window may also include additional features such as a send button, support for sending files or images, and suggestions or automated questions from the chatbot to assist the user.

The chat interface is optimized for easy interaction, with messages displayed clearly, and users can scroll up or down to view the chat history. This feature creates an intuitive and convenient learning environment, offering users support in a seamless and engaging manner.

REFERENCES

[1] **Tiangolo, S.** (2023). *FastAPI Documentation*.

Application: FastAPI is an optimal Python framework for quickly and efficiently developing RESTful APIs. It integrates well with language translation services and can support the necessary features for building learning and assessment applications for multilingual use.

Link: https://fastapi.tiangolo.com/

[2] **NPM.** (2023). *react-i18next: Internationalization for React*.

Application: react-i18next is a widely used library for internationalization in React applications. It helps manage language translations and switch between different languages in your application, making it easier to build multilingual learning and testing interfaces for students.

Link: https://react.i18next.com/

[3] **Smashing Magazine.** (2020). *Handling Multilingual React Apps*.

Application: This guide helps you build multilingual React applications, from setting up language files to handling special features like translating meanings or notifications in multiple languages.

Link: https://www.smashingmagazine.com/2020/06/handling-multilingual-react/

[4] **Google Cloud.** (2023). *Cloud Translation API Documentation*.

Application: Google Cloud Translation API supports automatic translation in your application, allowing for quick and accurate text translations between languages. This is particularly useful for translating educational materials and assessments into multiple languages.

Link: https://cloud.google.com/translate/docs

[5] **Khanh, P.** (2022). *Mobile and Web-Based Learning Systems for Ethnic Minorities in Vietnam*.

Application: This document studies the use of mobile and web-based learning systems for ethnic minority students in Vietnam, including tools and methods for effective online learning for these communities.

Link: https://www.researchgate.net/

[6] **Abdulazeez Abdulazeez Adeshina.** *Developing a Single Page App with FastAPI and React*

Application: This document helps to build a CRUD app with FastAPI and React.

Link: https://testdriven.io/blog/fastapi-react/

[7] **Adam Wathan** (2024). *Install Tailwind CSS with Create React App*

Application: This document helps to sett up Tailwind CSS in a Create React App project.

Link: https://tailwindcss.com/docs/guides/create-react-app

[8] **David Fateh** (2024). *Start a React app with Tailwind CSS in under 5 minutes*

Application: Build a simple website using ReactJS and Tailwind CSS.

Link: https://www.contentful.com/blog/react-app-tailwind-css/

[9] **Liam Holmes** (2024). *Build a Chatbot Application using ReactJS and FastAPI.*

Application: Develop an interactive chatbot using ReactJS for the frontend and FastAPI for backend services.

Link: https://realpython.com/react-fastapi-chatbot/

[10] **Isabella Martinez** (2024). *Enhance Your ReactJS Chatbot with AI Models using Python.*

Application: Integrate AI-driven responses in a chatbot by combining ReactJS frontend and Python backend using Hugging Face Transformers.

Link: https://huggingface.co/blog/react-python-chatbot-ai/

[11] **Emily Tran** (2024). *Deploying a Python-Powered Chatbot on ReactJS Frontend with Docker.*

Application: Containerize a Python chatbot backend and integrate it with a ReactJS-based frontend for deployment.

Link: https://dev.to/emilytran/docker-react-python-chatbot/

[12] **Michael Rivera** (2023). Python NLP Integration for Chatbots in ReactJS.

Application: Build a chatbot using ReactJS with Python libraries like spaCy for Natural Language Processing.

Link: https://github.com/michaelrivera/nlp-reactjs-chatbot

[13] **Daniel Morgan** (2024). *ReactJS Chatbot with TailwindCSS and Python API Integration.*

Application: Develop a ReactJS chatbot styled with TailwindCSS and powered by a Python FastAPI backend.

Link: https://danielmorgan.dev/react-tailwind-python-chatbot/

[14] **Ethan Wright** (2024). *Styling a Chatbot Interface with TailwindCSS and ReactJS.*

Application: Design an elegant chatbot UI using TailwindCSS in a ReactJS project while connecting to a Python-based backend.

Link: https://tailwindcss-chatbot-ui-react.com

[15] **Grace Parker** (2024). *Building a Fully Responsive Chatbot Interface with TailwindCSS.*

Application: Use TailwindCSS to create a mobile-friendly chatbot interface integrated with ReactJS and Python Flask for backend processing.

Link: https://graceparker.dev/tailwind-react-chatbot/

[16] **Karl Seguin** *The Little MongoDB Book*

Application: A concise and beginner-friendly guide to understanding MongoDB concepts and usage.

Link: https://github.com/karlseguin/the-little-mongodb-book

[17] **Stack Overflow - MongoDB Tag**

Application: A community-driven Q&A platform where MongoDB users discuss and solve technical challenges.

Link: https://stackoverflow.com/questions/tagged/mongodb

[18] **MongoDB Developer Hub**

Application: A hub with tutorials, articles, and example projects to help developers solve real-world problems with MongoDB

Link: https://www.mongodb.com/developer/

TABLE OF CONTENT