International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 5 May - 2025, Pages: 209-217

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

Trinh Thanh Do1 and Huynh Minh Tri2

1Software Development Director, Rikai Mind, Email: do.trinh@rikai.technology

2Le Quy Don School, Da Nang, Viet nam

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.

I. Introduction

1.1 Research Objectives

This research aims to address educational challenges and promote equity for ethnic minority students in Vietnam. Key objectives include enhancing access to quality education through a multilingual platform and digital tools, overcoming barriers like geographical isolation and language diversity. It employs adaptive assessments for personalized learning and multilingual resources to support native language engagement and skill-building. Additionally, it provides data-driven insights to help educators refine teaching strategies. These efforts aim to create an inclusive, effective learning environment that fosters understanding, retention, and improved academic outcomes.

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 topic highlights the urgent need to address educational disparities among ethnic minorities, who face unequal access to quality education, perpetuating poverty and limiting opportunities. Technology plays a crucial role in bridging gaps in resources, geography, and language through digital tools and platforms. AI and multilingual systems further enhance accessibility by personalizing learning experiences and breaking language barriers. These innovations create inclusive environments, empowering ethnic minority students with tailored solutions. By leveraging modern technology, this approach offers meaningful progress in addressing educational inequities for underrepresented communities, paving the way for greater engagement and transformative outcomes.

II. Theoretical Overview

This research combines educational technology, linguistics, and adaptive learning to enhance education. Digital tools boost engagement, multilingual content reduces language barriers, and AI-driven adaptive systems personalize learning. Together, these principles create an inclusive application for ethnic minorities, improving access, comprehension, and motivation while promoting equity in education.

III. Methodology

3.1 Application Development

The application was built with React.js for a dynamic frontend, Python for scalable backend logic, and MongoDB for flexible, schema-less data storage. This technology stack ensures functionality, scalability, and user-friendliness, enabling seamless development, efficient data management, and a responsive user experience tailored to modern web application needs.

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.

IV. 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 integrates frontend interaction, backend processing, and AI analytics for a personalized user experience. Users engage through an intuitive interface, with actions triggering backend requests. Python-based backend solutions manage data in MongoDB, ensuring flexibility and consistency. Advanced AI analytics analyze user performance in real time, tailoring content to individual needs. This adaptive approach boosts engagement and learning effectiveness. By combining user-centric design, robust data management, and AI intelligence, the platform delivers a seamless and impactful educational experience.

V. 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.

VII. 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.

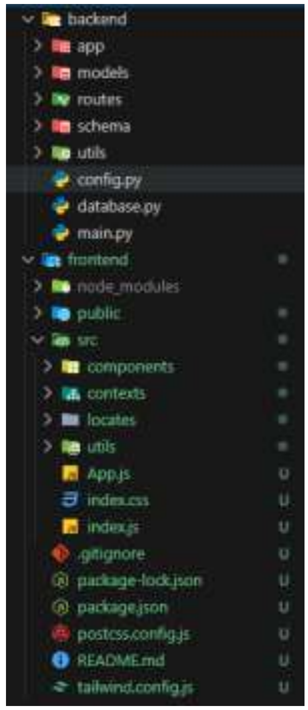
VIII. Conclusion

This research explores multilingual web applications' role in addressing educational challenges for Vietnam's ethnic minority students. Integrating AI and machine learning enables personalized, inclusive learning experiences by breaking language barriers and enhancing accessibility. The platform fosters equitable opportunities, empowering diverse learners. Future enhancements aim to improve functionality, engagement, and scalability for broader educational impact.

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 5 May - 2025, Pages: 209-217

IX. Appendices

9.1 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.

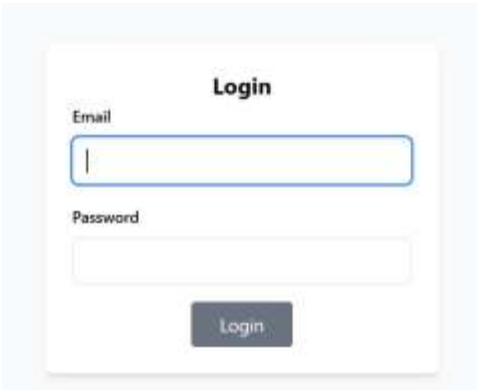
It provides:

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

```
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";

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>
  );
}

function App() {
  return (
    <LanguageProvider>
      <Router>
        <AppContent />
      </Router>
    </LanguageProvider>
  );
}

export default App;
```

9.2 Main interfaces

9.2.1 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.

9.2.2 Quiz page

International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 5 May - 2025, Pages: 209-217

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.

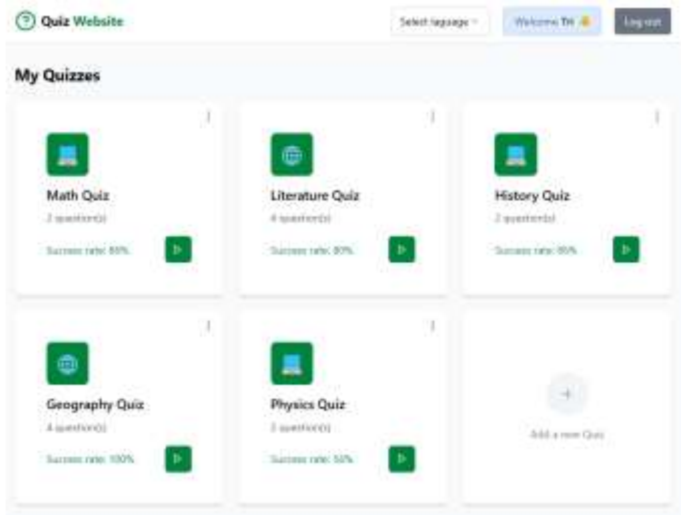
Once students have answered the questions, they can press the submit button to finalize their responses. The system then processes the submitted answers, evaluates them, and calculates the results. These results are promptly displayed on the screen, giving students immediate feedback on their performance.

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.

9.2.3 Create new quiz page

This screen enables teachers to create and manage question sets with an intuitive interface for inputting questions and answer formats. Teachers can categorize questions by subject or difficulty and add explanations or hints to enhance learning. Customizable settings include time limits, attempts, and feedback options, tailoring assessments to learning objectives. This tool supports flexible, high-quality assessments, helping teachers monitor progress and adapt strategies effectively.

9.2.4. Student Dashboard page



This screen serves as a hub for students to access and complete organized question sets by topic or difficulty. Teachers can create, edit, and customize assessments, adjusting settings like time limits, question types, and feedback preferences. It supports student learning while empowering teachers to tailor content to evolving educational needs.

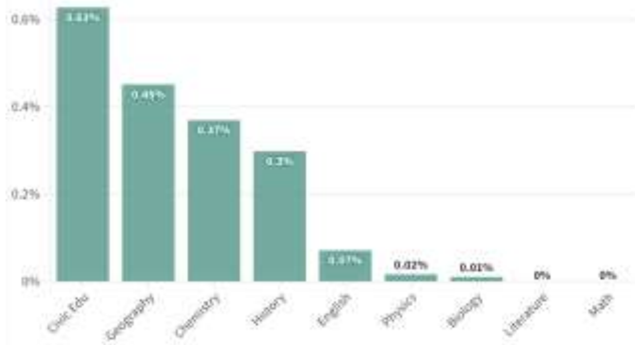
9.2.5 Teacher Dashboard page

This screen provides teachers with real-time analytics on student performance, including scores, test results, completion times, and areas of difficulty. Data visualizations, such as graphs, summarize class trends, while filters enable focused analysis. These insights help identify learning gaps, improve teaching strategies, and offer personalized feedback to enhance student progress.

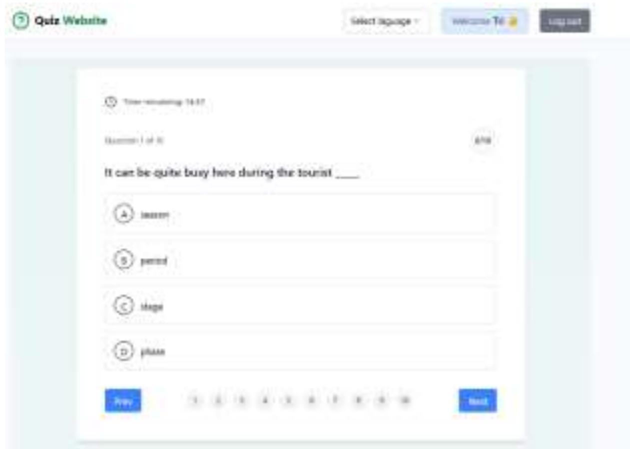
9.2.6 Education Managers

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

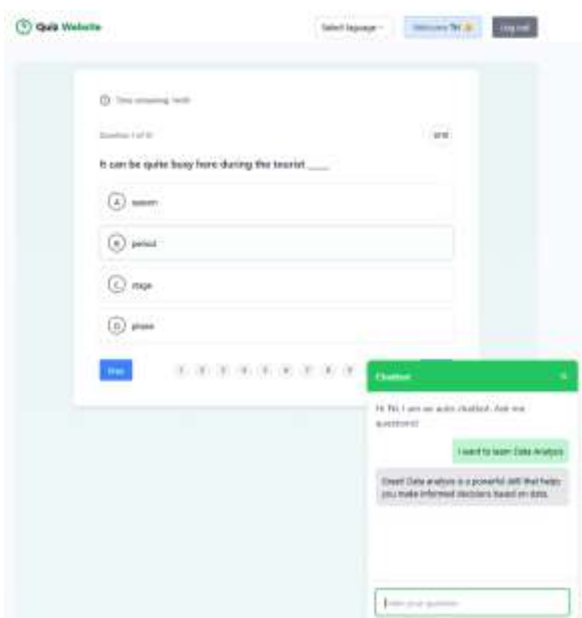
International Journal of Academic and Applied Research (IJAAR)
ISSN: 2643-9603
Vol. 9 Issue 5 May - 2025, Pages: 209-217



9.2.7 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.

X. References

1. Tiangolo, S. (2023), FastAPI Documentation, FastAPI, 1-12.
2. NPM. (2023), react-i18next: Internationalization for React, NPM, 5-15.
3. Smashing Magazine. (2020), Handling Multilingual React Apps, Smashing Magazine, 20-30.
4. Google Cloud. (2023), Cloud Translation API Documentation, Google, 10-25.
5. Khanh, P. (2022), Mobile and Web-Based Learning Systems for Ethnic Minorities in Vietnam, ResearchGate, 15-27.
6. Abdulazeez Adeshina (n.d.), Developing a Single Page App with FastAPI and React, TestDriven.io, 30-42.
7. Adam Wathan(2024), Install Tailwind CSS with Create React App, Tailwind CSS, 5-12.
8. David Fateh (2024), Start a React App with Tailwind CSS in Under 5 Minutes, Contentful, 7-18.
9. Liam Holmes (2024), Build a Chatbot Application Using ReactJS and FastAPI, Real Python, 20-35.
10. Isabella Martinez (2024), Enhance Your ReactJS Chatbot with AI Models Using Python, Hugging Face, 40-55.
11. Emily Tran (2024), Deploying a Python-Powered Chatbot on ReactJS Frontend with Docker, Dev.to, 22-30.
12. Michael Rivera (2023), Python NLP Integration for Chatbots in ReactJS, GitHub, 10-20.