Science, Engineering & Education, 11, (1), 2026, 27-35

Platform for Assessment and Supervised Learning

Kristiyana Stankova*, Gabriel Likov

Vocational High School of Electrical Engineering and Electronics - PGEE
26 Peshtersko Shose Blvd., Plovdiv, Bulgaria

Received 05 December 2024, Accepted 17 March 2025
DOI: 10.59957/see.v11.i1.2026.5

ABSTRACT

This paper presents POKO - a platform for assessment and supervised learning designed for private educational institutions and schools. The system integrates essential educational activities into a single, secure, and user-friendly environment. POKO supports multiple user roles, including administrators, teachers, students, and parents, each with clearly defined permissions. The platform provides functionality for course management, assignment distribution, testing, grading, attendance tracking, communication, and access to learning materials. By combining assessment tools and digital learning resources in one place, the platform aims to simplify the educational process, improve transparency, and enhance interaction between teachers and students. The architecture and implementation are based on modern web technologies, enabling real-time updates, scalability, and easy maintenance. The paper describes the system architecture, used technologies, core modules, and development process, as well as the main contributions of the authors.

Keywords: education, learning, security, assessment, materials.

INTRODUCTION

Digital educational platforms have become an essential part of modern education, especially in environments that require remote access to learning materials, assessment, and communication. Existing platforms such as Moodle, Google Classroom, and Microsoft Teams for Education provide a wide range of functionalities, but they are often complex to configure or fragmented across multiple tools. This can make daily use difficult for smaller or private educational institutions.

The software project “POKO - Platform for Assessment and Supervised Learning” was developed to address these issues by providing a centralized and intuitive system that combines assessment, learning materials, and communication in a single platform. The system is intended for use in private educational institutions and schools, where simplicity, security, and role-based access are critical.

**Correspondence to: Kristiyana Stankova, Vocational high school of electrical engineering and electronics, 26 Peshtersko Shose Blvd., Plovdiv, Bulgaria, e-mail: krisi.stankova.07@gmail.com*

Each user has a personal account with rights defined by their role, allowing access to courses, assignments, grades, timetables, and institutional information relevant to them.

USED TECHNOLOGIES

Next.js is a full-stack framework and uses the popular UI library React. It is used because it supports both server-side and client-side rendering, which makes it easier to develop the project, as there is no need to use a backend framework and the entire project is in one folder. It also has advantages such as Link and Image components, through which all links to pages are loaded in the background, as well as images [1, 2].

Firebase is a BAAS (backend as a service) that makes it easy to develop web applications. It provides various cloud services that eliminate the need to create your own backend from scratch. Firebase is a non-relational database for storing website data such as emails, passwords, which class the users are from, their usernames, some personal data, and more. It has a collection-document structure that makes the relationships between documents easy to understand. The project also uses Firebase authentication to register user accounts [3, 4].

TypeScript is a superset of JavaScript that adds types to JavaScript. This makes the code more readable and allows for easier debugging when using objects with more complex types in different project files [5, 6].

Tailwind is a CSS framework that makes it easy to style web applications without having to write your own CSS styles. Instead of creating custom CSS classes, Tailwind provides predefined classes that directly apply styles to HTML elements [7, 8].

Shadcn/ui is a library for UI components that uses Radix UI for accessibility and Tailwind CSS for styling. It is used because it provides a collection of copyable components that can be added directly to code, making it easy to prototype during development [9, 10].

MAIN ACTIVITIES

The study began with planning the database structure and the models it would use. Then the authors discussed how will the user interface be and after ideas were born, the developing of sample page models in Figma started. After that the work on creating the pages began. First, the home page was created, and then the pages it was linked to. The code in the project was organized by functionality: components, helper methods, and contexts. In the final stage of the project development, the authors began to thoroughly review the code for any errors or omissions that could cause problems.

WHY THE CHOSEN TECHNOLOGIES?

One of the first things they faced was choosing technologies that would be compatible with the needs of the project. The decision to use Next.js was based mainly on the fact that it has increased performance thanks to React's server-side components, allowing for real-time updates with fast refresh, optimized loading, data management, and of course, the fact that both the frontend and backend can be created in a single framework [1, 2].

TESTING AND OPTIMIZING THE PROJECT

In the final stages of the study development was to give the final product to be tested by friends and family. During the tests, problems were discovered that the authors set about fixing. After they were fixed, the project was again given for testing so that all inconsistencies in the site's operation could be eliminated.

ARCHITECTURE

Each folder except fonts represents a page, which is a path from or to another one. In the components folder we have 2 subfolders - "functional" and "ui". In the "functional" folder the components with logic are saved, while in "ui" the visual components from Shadcn/ui are stored.

The "contexts" folder contains mechanisms that allow data to be passed in the component tree, without the need to pass them manually through props at each level. In the "hooks" folder there are files that contain a configuration file for PWA and toast notifications. In the "lib" folder there are helper methods and the project headers. This structure ensures that individual features can be updated or extended without affecting the overall system integrity, following established software engineering principles [11, 12].

MODULES IN THE PROJECT

The site has fourteen main modules, which are created to present information in a user-friendly manner.

The "Home" module includes options for logging into the platform, registering a parent and an educational institution. Below these options, there is a little more information about the platform and a menu where users can contact the creators in case of problems or questions. Fig .1. Home page.

The "Dashboard" module. After registering on the site, the user is sent to the dashboard, where he can access all the modules related to his role. There is a navigation bar, which has the name of the platform, the name of the educational institution with which the user is associated, a menu for notifications, messages and his profile. There is also a sidebar, which displays the user's name and surname, his role, a menu for the dashboard, courses, assignments and grades. In the module itself, there are different menus according to the user's roles. Fig. 2. Dashboard page.

In the "Courses" module, if the user is a teacher, he can create them, and if he is a student, he can see those he is attending. The teacher has the option to create fully detailed courses, containing chapters, subchapters and topics, allowing for more organized learning material and a more understandable user experience. Students have access to certain courses, depending on their relation to classes or subjects. Fig. 3. Courses page.

In the "Assignments" module, if the user is a teacher, he can create them, and if he is a student, he can see those that he is assigned to complete. Assignments can be assigned deadlines and can be accessed by specific users or classes. They can be initiated in the middle of a class or as homework due next week. Fig. 4. Assignments page.

The "Tests" module, in which students receive

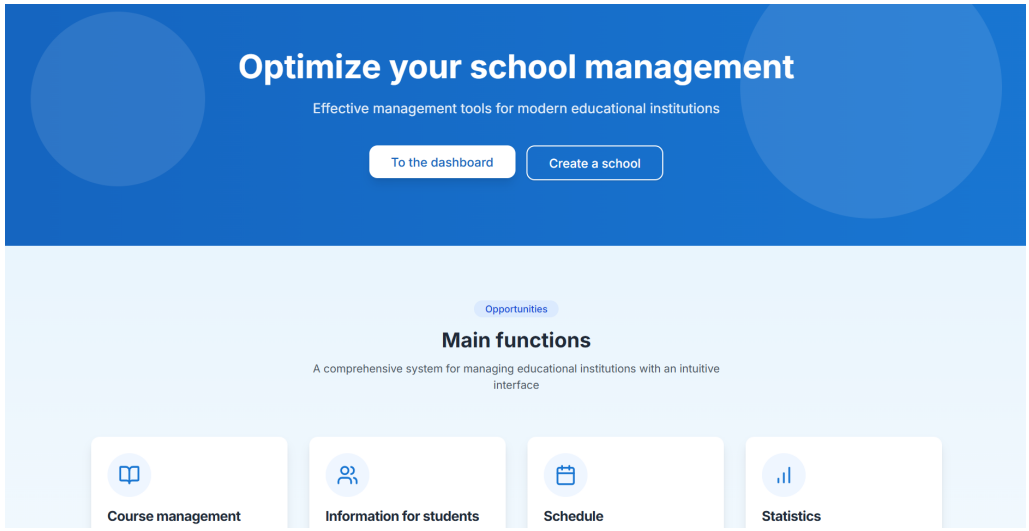

Fig. 1. Home page.

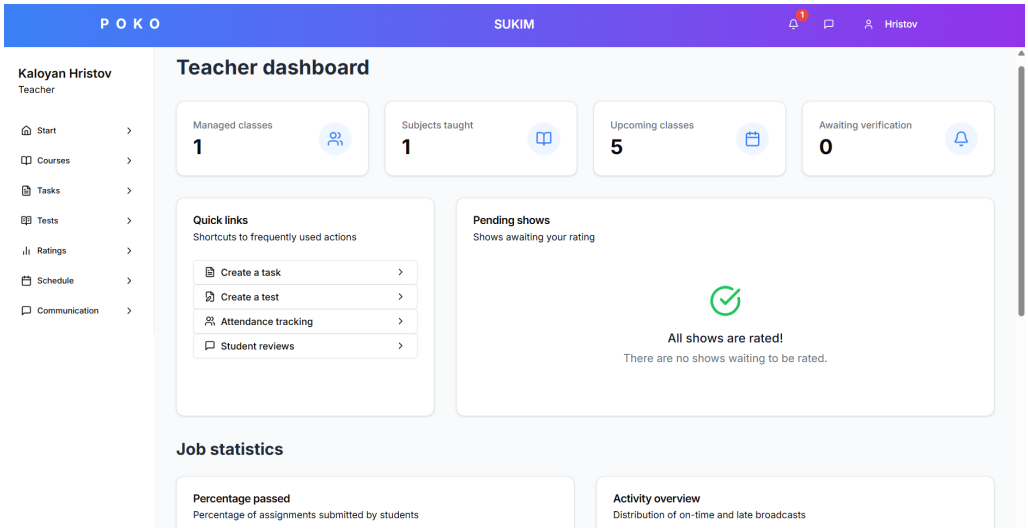

Fig. 2. Dashboard page.

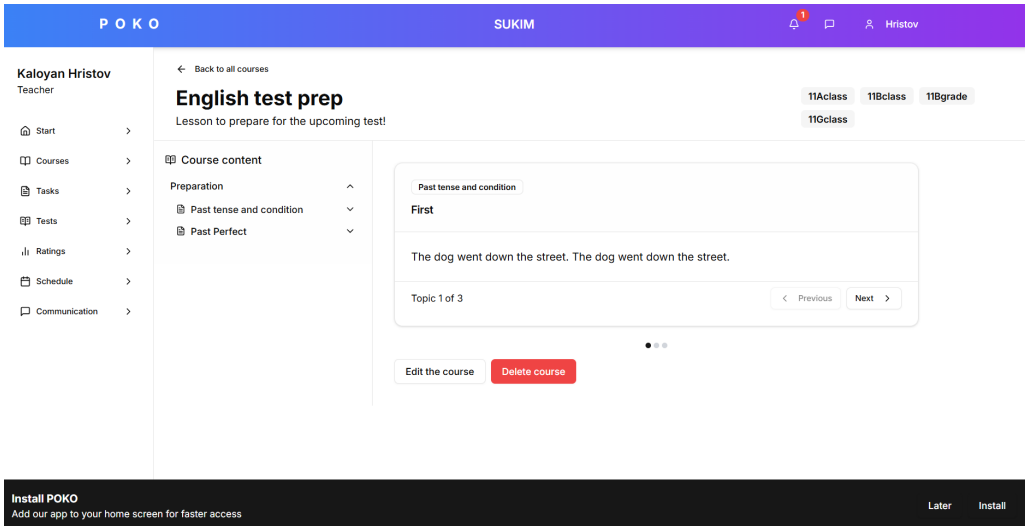

Fig. 3. Courses page.

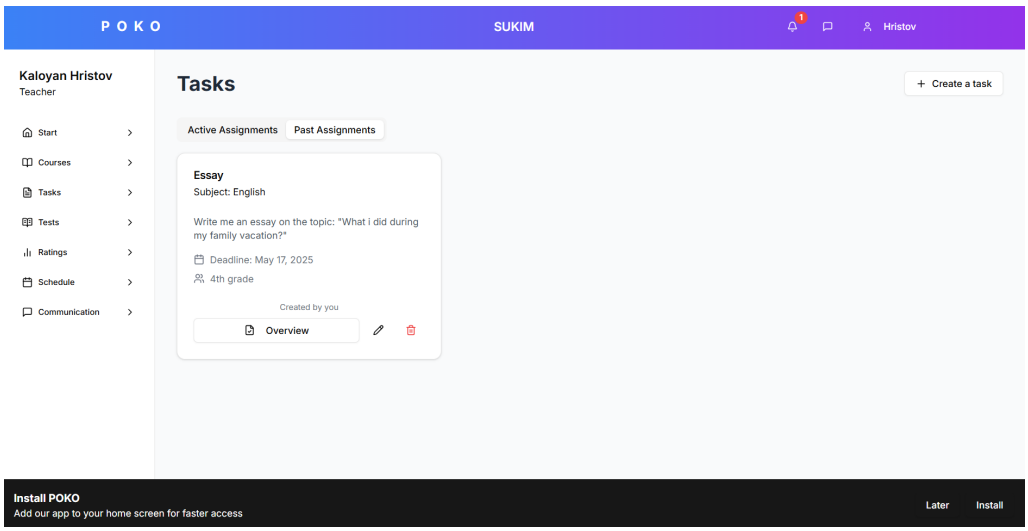

Fig. 4. Assignments page.

tests created by teachers with time to complete. Teachers have the option to make multi choice questions, open-ended questions and true-false questions. He can assign a specific number of points to each question, and a threshold for points required for each grade, allowing him to grade students faster and more accurately. There is also anti-cheat measures built into the test, alerting the teacher in case students attempt to cheat. These include switching browser windows, closing the page, inactivity and leaving the quiz page. The security level of the quiz can be chosen by the teacher, depending on the importance of the quiz or exam. Fig. 5. Tests page.

The "Assessment" module, if the user is a teacher, he can create grades, and if he is a student, he can see those that are assigned to him. Grades can be given based on the Bulgarian scale from 2 - 6 or a custom one if the admin creates one. Each grade has a related subject, assessment type, title and description in case the teacher wants to add one. Grades can be filtered based on classes, students or subject. Fig. 6. Grades page

The "Timetable" module, in which the user sees his school program. Teachers see their assigned classes and at what times they are as

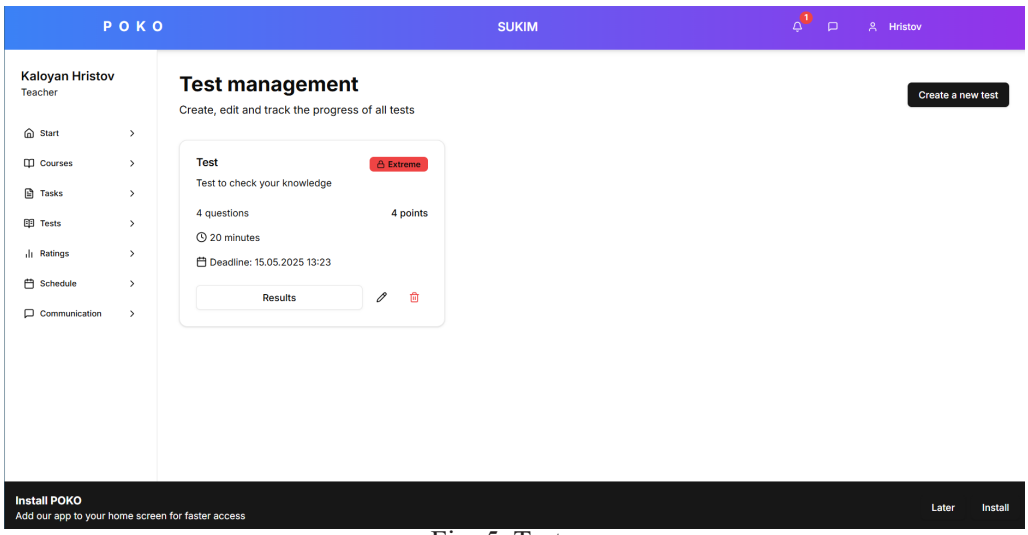

Fig. 5. Tests page.

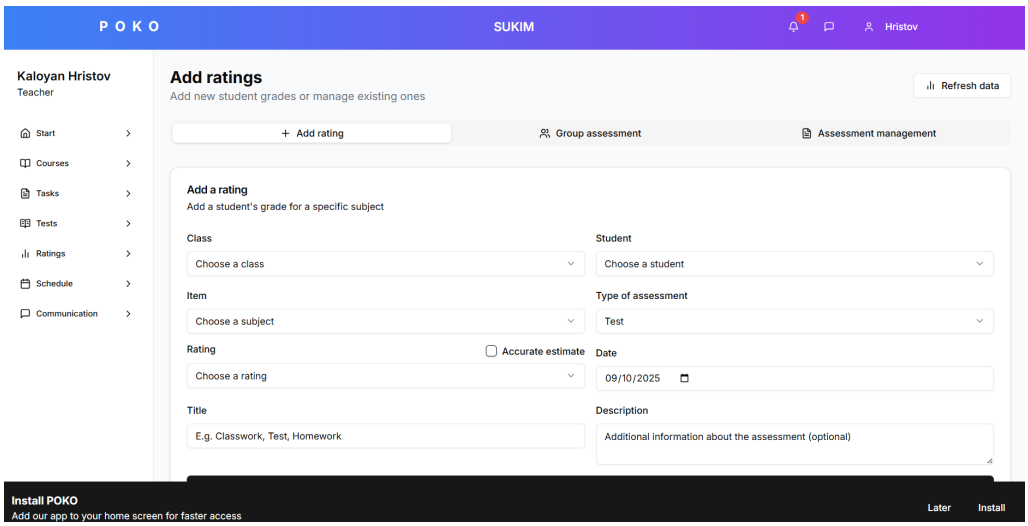

Fig. 6. Grades page.

well as students, including free periods. Specific period times and break times can be adjusted by the administrator in case of changes to schedule. Fig. 7. Timetable page.

The “Attendances” module, in which the student and their parents see the number of absences, tardiest, number of excused absences, and for which periods they apply. From this module, the teacher enters his class as present and enters absent or late students.

The “Feedback” module, where teachers can post a positive or negative review of a student with a description. Students and parents can see their reviews and directly chat through the chat module with the teachers in case of misunderstandings. Fig. 8. Feedback page.

The “Messages” module, in which the users can message each other. When a user messages another user while they are not there, they get a notification in the navigation bar, so they don’t forget to reply. Administrators can make announcements, referring to everybody in the school. Fig. 9. Messages page.

The “Notifications” module, where the user receives notifications about various activities on the platform. For example, teachers are notified

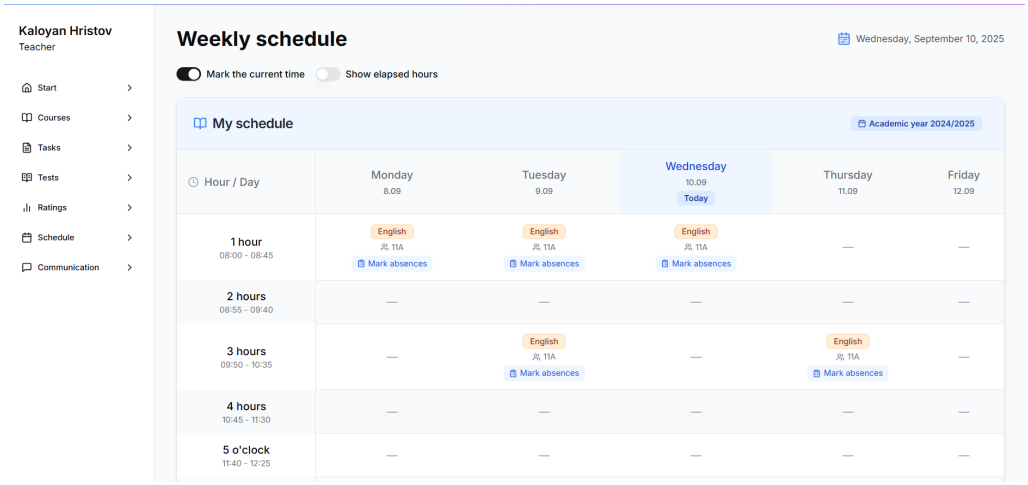

Fig. 7. Timetable page.

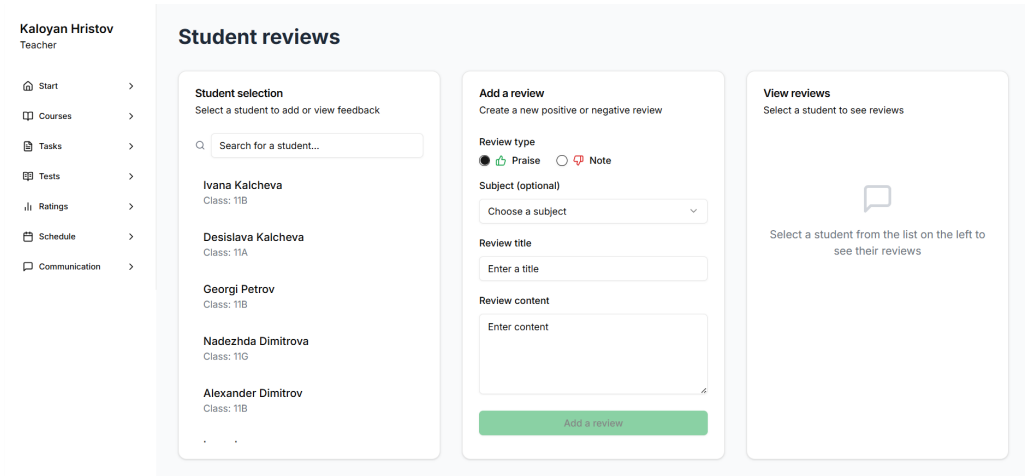

Fig. 8. Feedback page.

when an assignment or test is sent, and students when an assignment, test or grade is posted. Fig. 10. Notifications page.

The “Profile” module, in which the user sees his information. Information cannot be changed by the user. The user can change their password through this module. Fig 11. Account page.

REALISATION

When developing the platform, the necessary functions that it should have were considered and based on this, the following tools were selected for work.

Next.js was chosen because it uses the React UI library, which facilitates the development work, since the code is much more visual, structured and easy to work with. Its performance and refresh rate contribute to the work of the site, given that it works in real time [1, 2].

Considering that the development of a site requires multiple folders with separate components, TypeScript helps by providing the use of types, which greatly helps in writing the program [5, 6].

Firebase as a non-relational database works very well with React, which makes it suitable

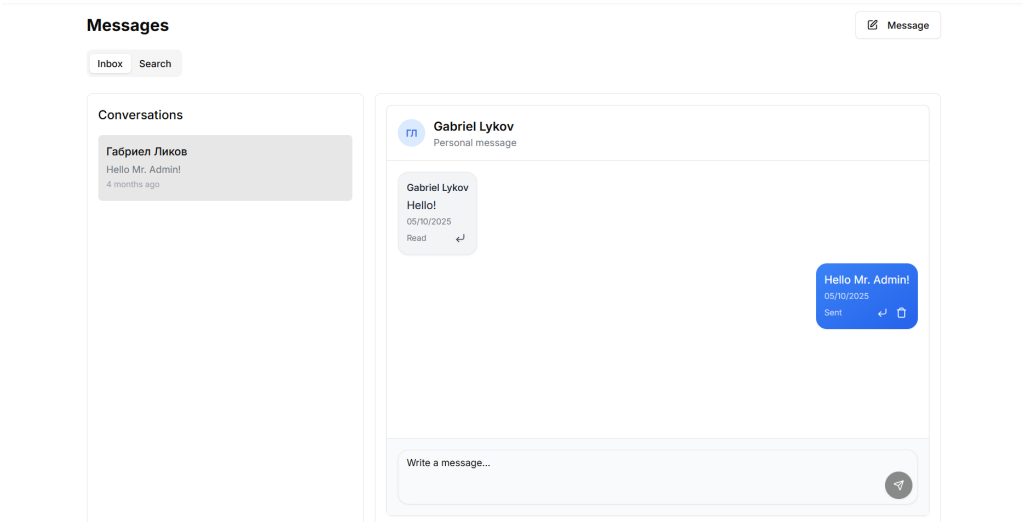

Fig. 9. Messages page.

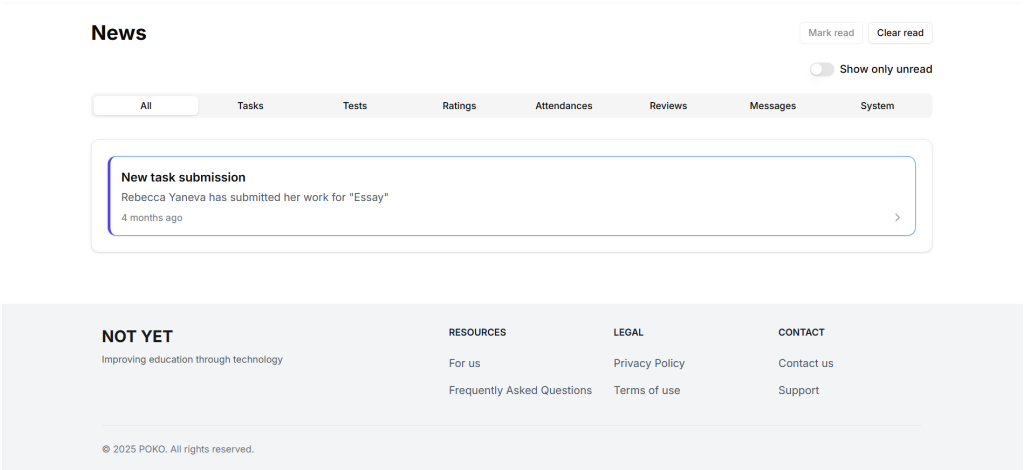

Fig. 10. Notifications page.

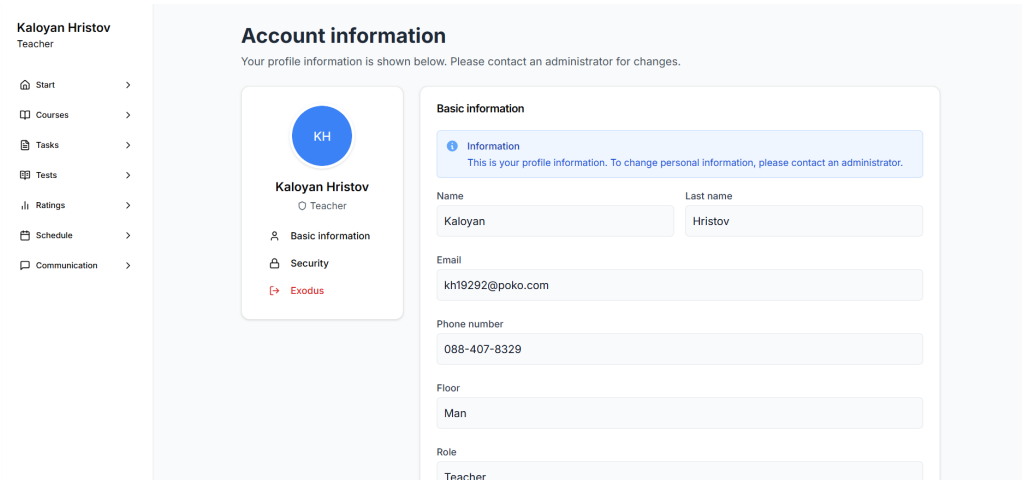

Fig 11. Account page.

for use in development. It is easy to work with, understand and provides the hassle-free saving of new data from users [3, 4].

Tailwind and Shadcn/ui provide easier graphic design, as instead of creating custom CSS classes, we can use predefined ones when creating the elements themselves [7, 9].

STARTING, INSTALLING, USING AND MAINTAINANCE

The POKO platform is accessible through a standard web browser and can also be installed as a progressive web application (PWA) on supported devices. Users can access the system via the provided web address, while installation as a standalone application is offered through the browser's native installation mechanism.

The platform supports four types of user accounts: administrator, teacher, student, and parent. Navigation and available functionality are determined by the user's role, ensuring controlled access to system features and data.

The system is designed for straightforward maintenance and future development. The modular architecture, organized code structure, and clearly defined components allow individual features to be updated or extended without affecting the overall system. In addition, the structured database design enables modifications and scalability while preserving data integrity [14].

CONCLUSIONS

This paper presented the design and implementation of the POKO platform for assessment and supervised learning. The main contribution of the work is the development of a unified web-based system that integrates learning materials, assessment, communication, and role-based access control within a single, user-friendly environment suitable for private educational institutions.

The backend architecture of the platform, including database design, authentication mechanisms, data management, and system logic, was developed by Gabriel Likov. His work ensured secure data handling, real-time functionality, scalability, and reliable interaction between system modules using modern web technologies.

The frontend design and implementation, including user interface development, layout structure, visual consistency, and user experience optimization, were carried out by Kristiyana Stankova. Her contribution focused on creating an intuitive, accessible, and visually coherent

interface that supports efficient navigation and usability for different user roles.

Together, the authors demonstrate how the combination of a well-structured backend and a carefully designed frontend can result in a practical and scalable educational platform. The presented solution shows potential for further development and adaptation to the needs of modern digital education.

Acknowledgment

The authors would like to thank the editor of the journal for providing constructive reviews and valuable comments that helped improve the quality of the manuscript. Special thanks are also extended to their teacher, Mariana Hristozkova, for leading the project and providing continuous guidance throughout its development. The authors additionally thank the teachers and students who participated in testing the platform and provided valuable feedback during its development.

REFERENCES

1. Next.js Official Documentation, Next.js, https://nextjs.org/docs (accessed 2025).
2. Vercel, Introducing the Next.js Framework, Vercel Blog, 2016, https://vercel.com/blog/nextjs (accessed 2025).
3. Google, Firebase Documentation, Google LLC, https://firebase.google.com/docs (accessed 2026).
4. L. Moroney, The Definitive Guide to Firebase, Apress, New York, 2017, ISBN: 978-1-4842-2943-6.
5. Microsoft, TypeScript Documentation, https://www.typescriptlang.org/docs/.
6. G. Bierman, Mart´ın Abadi, Mads Torgersen, Bringing Types to JavaScript with TypeScript, Proceedings of the ACM on Programming Languages, OOPSLA, 2014.
7. Tailwind Labs, Tailwind CSS Documentation, https://tailwindcss.com/docs.
8. A. Wathan, Refactoring UI, Independently published, 2018.
9. Shadcn, Official UI Components Library, https://ui.shadcn.com.
10. Radix UI, Documentation, https://www.radix-ui.com.
11. I. Sommerville, Software Engineering, 10th ed., Pearson, 2015.
12. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
13. M. Poppendieck, T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison-Wesley, 2.
14. Site address: https://poko-2.vercel.app/