

Computer Science Department
Senior Project II - Spring 2026
AI-enhanced Educational Platform

Nazarbayev University

Group 46 members:
Akmaral Amanzholova
Nurassyl Mukan
Yernar Kusseibek
Dauirzhan Agaidarov

Project Advisor: Askar Boranbayev

I. Executive Summary

In today's world student workload and cognitive load are continuing to rise. However, the currently available products that tend to highlight one advantage, such as class management or spaced repetition do not cater adequately to the need for personalization of study materials.

This project addresses exactly that problem by designing and implementing a computer-based educational platform that combines adaptive learning, gamification, teacher-facing insight in a single system. To support analytics and personalization, the platform integrates user-created study content with interaction data like accuracy, time-on-task, streaks. The objectives of this project were to:

- Provide secure role-based access for students, teachers and admins
- Support creation, sharing and iterative progress through learning sets (flashcards, quizzes, problem sets) with analytics
- Integrate AI-assisted capabilities such as hints and explanations
- Deliver cross-platform clients with usability and portability goals.

As for the methodology, it followed the identification of how students and teachers learn and interact with educational systems, the design of system architecture which includes backend services, a database for persistent data storage and client applications such as a web and a mobile app.

As a result, the team put together a fully functioning prototype in which FastAPI backend, a React web app, Expo/React Native mobile app, all

runnable locally with straightforward instructions, are brought together. Authentication is fully covered with registration, login, token refresh, Google sign up/sign in, along with profile management and full study workflow: creating sets and questions, being added to classes, receiving assignments, tracking progress, answers submissions with early offline support. In addition, the prototype includes rule-based adaptive recommendations, gamification, optional AI-powered assistance, tailored views for teachers and students and admin tools.

Evaluating the system in this phase meant demonstrating these flows on a local deployment through development testing and a user study.

This project aligns with designing, implementing and evaluating a computing-based solution. Design began with actual users (students, teachers), translating their needs into requirements. Implementation turned into working software. Evaluation involved bringing the system to potential users and having them test it on a local machine followed with a survey asking whether they were satisfied with the system and its different features.

II. Introduction

Looking into the limitations of existing educational platforms revealed that many current systems mostly focus on only one aspect of learning such as classroom management, engagement or content provision without unifying all of those into a single system. As a result, students do not get the structured support specifically tailored for them and teachers may face challenges in monitoring performance and providing feedback.

The motivation behind this project is to address these problems by combining the important components of modern education technology into a single system. To be specific, there is a need for platforms with adaptive learning that also facilitate teacher-student interaction and incorporate intelligent assistance.

To address these challenges, we propose Nova Edu, an AI-enhanced educational platform that brings study set management, class workflows, progress tracking and AI-assisted learning together. The system allows students to create/engage with structured materials, monitor their level of mastery and receive further recommendations, while teachers can create content, assign sets and analyze student performance via dedicated dashboards. On top of that, our system offers gamification features and

optional AI assistance to increase engagement and provide additional learning support.

The report's organization:

- I. Executive Summary
- II. Introduction
- III. Identification of Constraints
 - A. Security and Safety Constraint
 - B. Social and Ethical Constraint
 - C. Economic Constraint
 - D. Time Constraint
- IV. Background and Related Work
 - A. Prior Research
 - B. High-level Platform Comparison
- V. Project Approach
- VI. Project Execution
 - A. Web Application
 - B. Mobile Application
 - C. Division of Responsibilities and Leadership
- VII. Evaluation
- VIII. Conclusion and Possible Future Work
- IX. References

III. Identification of Constraints

A. Security and Safety constraint

Since the platform is intended for use by students of different ages, robust access control and data protection measures are mandatory to implement. The platform had to have a secure authentication system (OAuth), enforced Role-Based Access Control (RBAC) to ensure that the users, who are students, teachers and administrators, access only information that are relevant to each of them, and do not step out of the boundaries and get personal information of other users. Also, the website was protected from common web vulnerabilities like SQL injection.

B. Social and Ethical constraint

Since the platform provides AI-enhanced features, and collects user interaction data for analytics, the project was constrained by social and ethical considerations. The system was developed to protect user data and

prevent any data leakage from happening by providing strong security measures. There is full transparency in regarding generating AI assisted personal recommendations, making the AI assistance fully optional rather than a mandatory process. Other than focusing on protecting user privacy, the platform also considered its accessibility and fairness. There are no bias based information that could be obtained from the website, and it was ensured that the users would not be at a disadvantage if they are accessing the platform using a weak device, creating a dependency on the technology that might hinder deeper learning or critical thinking skills.

C. Economic constraint

The project was constrained by a zero-dollar budget. So, during the development of the project, open-source frameworks were used rather than large scale, enterprise-level computing solutions. This would not be an issue at the beginning, but in the long run, options like cloud computing are necessary to provide the best experience to the users.

D. Time constraint

The project was developed within a year's development timeline. This limited the scope and complexity of features that are present on the platform. Consequently, the main focus was on the fundamental, core functionalities that are necessary for the main purpose of the project. Non critical, over extensive advanced features had to be excluded from the schedule until the main functionalities are ready and operating very well.

IV. Background and Related Work

A. Prior Research

When it comes to modern learning systems, there are three main components: learner's model (what they know and how they learn), teaching strategies (what to show next and when to help), UI that keeps users engaged and motivated [1]. Our research on Intelligent Tutoring Systems (ITS) revealed that they have a structure based on two levels with outer lever representing the tasks assigned based on the learner's progress and inner level involving feedback, hints and step-by-step guidance. Furthermore, a well-designed ITS is capable of reproducing results obtained with human tutoring in controlled settings [1]. From this it can be inferred that digital learning platforms should include structured feedback and progression rather than just static content.

Research into AI-based education highlighted challenges such as privacy, explainability and fairness [2, 3]. These concerns influenced our design decisions like the use of secure APIs, role-based access control (RBAC), transparent recommendation logic and optional AI-assistance.

As for gamification, existing platforms proved that badges, streaks, leaderboards and quick feedback can keep the users engaged. For example, Duolingo (a language learning platform) is used worldwide as a gamified mobile learning system and its positive impact on learning outcomes, design and motivation has been highlighted [4]. So, this supports our decision to include gamification features such as badges, streaks and progress dashboards.

B. High-level Platform Comparison

Four well-known platforms that relate to our project:

	Quizlet [5]	Kahoot! [6]	Google Classroom [7, 8]	Duolingo [9]	Nova Edu (our platform)
Primary role	Study sets: flashcards, practice modes, sharing	Live competitive quizzes, engagement-first	Class workflow: assignments, materials, grading	Structured language courses	Integrated study, classes, analytics, recommendations, optional AI-assistance features
Adaptive/personalized learning	Learning modes, algorithms to decide what to study next (depends on subscription version)	Pacing modes (focus is on pace + engagement per session)	Depends on the teacher	Adapts learning experience to each user based on performance and behavior	Rule-based recommendations, AI routes for hints/explanations
Gamification	Engagement with points/streaks	Strong: timing, competition, scoring	Typically low	Very strong: streaks, leagues, rewards	Strong: badges, points, streaks, backend-supported leaderboards

Teacher/institution features	Sharing classes (product dependent)	Kahhot subscription designed for organizations	Strong: assignment, institutional integration	Classroom dashboards are not classroom equivalent	Classes, assignments, progress endpoints
Multilingual UI	Broad UI/content language options for learners	Many interface languages	Very broad classroom UI language support	UI localization as global product	Web UI i18n: English, Russian, Kazakh
Multilingual content	User-created sets can be in many languages	User-generated questions, interface language separate from question language	Assignments and posts in a teacher's chosen language	Each course for a combination of two languages	Study sets and questions are author-defined (any language), AI calls pass response_language where implemented
Deployment/openness	Commercial SaaS	Commercial SaaS	Commercial SaaS (institutional workspace)	Commercial SaaS	Self-hosted prototype; open architecture (FastAPI, React, Expo) for expansion
Closest similarity to us	User-generated study sets	Quizzes	Class + assignment semantics	Motivation and adaptation	Combines study tools, class semantics, analytics, gamification, AI assistance

Table 1. High-level comparison of related platforms and Nova Edu

V. Project Approach

a. System Architecture

Our project follows client-server architecture (Figure 1).

1. Client Layer:

- **Frontend (Web application):** React + TypeScript + Vite
 - **Frontend (Mobile):** Expo + React Native
2. Application Layer:
 - **Backend:** FastAPI
 3. Data Layer:
 - **Database:** PostgreSQL
 4. External Services:
 - AI:** Gemini AI API

Interaction: User sends REST API requests to backend → backend calls the database → database responses back → client gets the result.

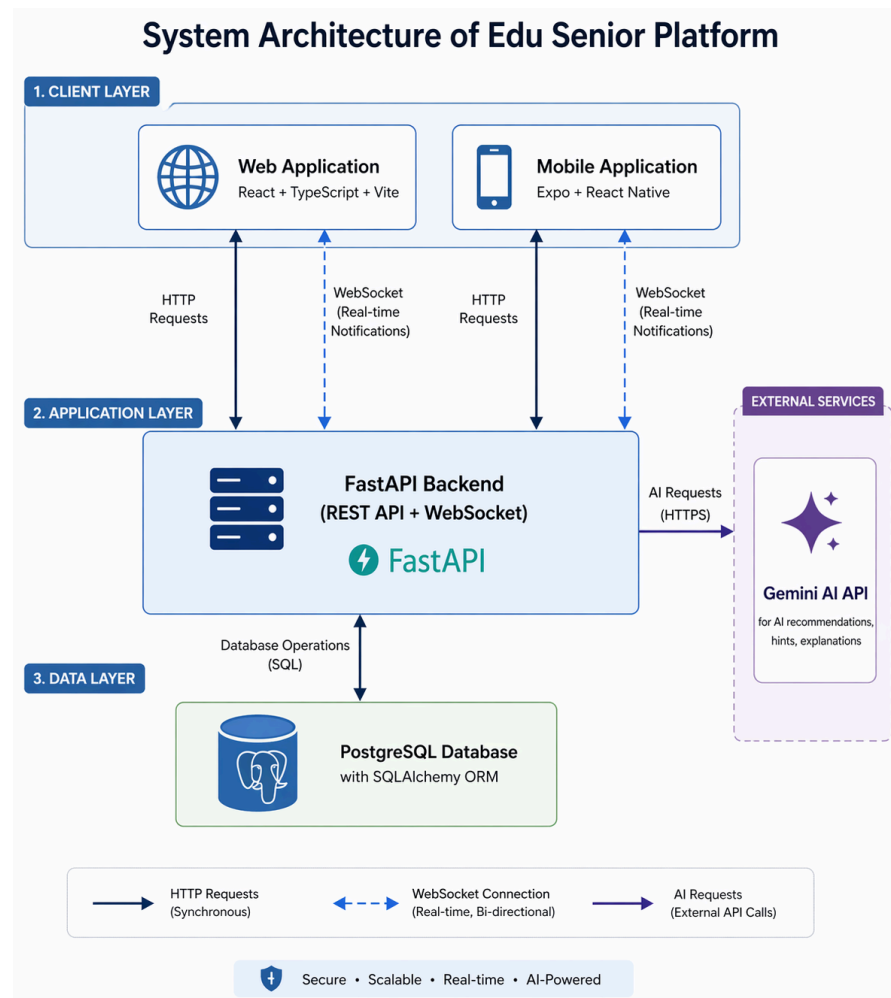


Figure 1. System Architecture of Nova Edu

b. Roles

- **Student:** A regular learner role; can access assigned/own/public study sets
- **Teacher:** An educator role; can create/manage teaching content and classes
- **Admin:** A privileged moderation role; can manage users/roles and delete users or study sets via admin endpoints

c. Key Features

- **Authentication system:** The authentication system is based on JSON Web Tokens (JWT). It supports both traditional email/password login and OAuth authentication via Google and Github. It has Role-Based Access Control (RBAC) to differentiate permissions.
 - **Registration:** During registration, user credentials are stored securely using password hashing (Argon2) and a default role (student) is assigned if not specified by the user.
 - **Roles:** Student, Teacher, Administrator
 - **Login:** When a user logs in, the system generates two tokens: a short-lived access token and a longer-lived refresh token. These tokens are stored in HttpOnly cookies.
- **Study sets:** A study set is a functional unit that contains learning content. Teachers and students can create and delete study sets. There are 2 types of study sets: private and public. In the study set, there are 3 types of educational content: quizzes, flashcards and problem sets. A public study set is a type of study set that is accessible to everybody while a private study set is accessible only for assigned people. Teachers can assign study sets to classes and control the access. Each Study set is stored in a database and accessed via REST API.
- **Class:** A class is a unit that contains several units: study sets and students. Only the teacher can manage the class: adding students and study sets into it. Each class is stored in the database. The classes also have a leaderboard in which the students with highest scores are at the top.
- **Offline download:** A feature unique to the student role. It allows users to store desired study sets in their system to access them without the internet access. It works by saving the set in the browser Indexed Database API. The app fetches the set from API and saves it into the

Indexed DB. When the connection is restored it sends unsynced attempts to backend

- **Progress management:** When a user finishes the set the systems tracks the results and stores in backend and in progress tab is fetched with calculated percentage
- **Gamification:** points, badges, streaks and leaderboards are directly connected to user activity and progress to encourage consistent interaction.
- **Recommendations:** The system analyzes user performance and a mastery level of lower than 60% results in repetition or easier material while higher mastery level results in more advanced study sets.
- **Notifications:** A notification is created when something happens in the system and it is added to a notification object. The total number of notifications is shown in the notification bell UI element

d. Third-party tools:

- Google OAuth: to make it possible to login via Google account
- Gemini API: used to provide AI features
- PostgreSQL: database system
- SQLAlchemy: used to work with the database using Python models
- Expo: mobile application framework
- FastAPI: used to build backend API
- Uvicorn: used to run the FastAPI server when developing
- Alembic: used to manage database schema changes
- React + Typescript: used to build the website frontend
- Vite: used for fast frontend development
- React Router: used for page navigation
- PyJWT: used to create and verify login tokens
- MUI: used for consistent UI components
- i18next : used to support multiple languages
- KaTex: used to render math
- AsyncStorage: used to store small local data on mobile

e. Use Case Diagram

Figure 2 shows the Use Case Diagram of the project.

Actors:

- **Student:** A student can Register/Login, browse and practice Study Sets which includes submitting answer, view their progress, get notifications, receive AI recommendations, download and sync offline study sets and view notifications.
- **Teacher:** A teacher is responsible for content creation and class management. They can log into the system, create and manage study sets and organize classes by adding or removing students. They can assign study sets to classes or specific students and set due dates/time limits per study set. They also have access to analytics, student progress and AI recommendations to guide teaching decisions.
- **Admin:** An Admin can manage system-level operations: includes managing users, deleting them if necessary, moderating study sets and assigning roles.

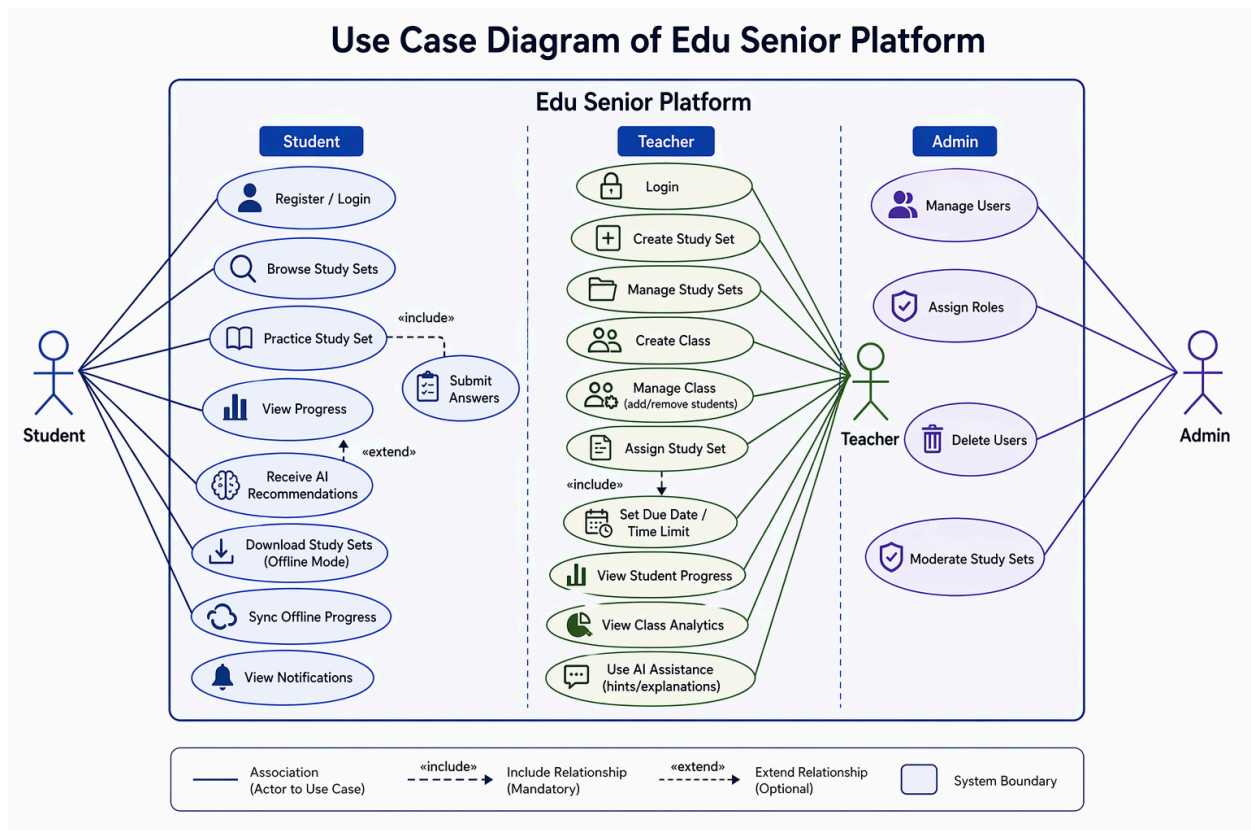


Figure 2. Use Case Diagram For Nova Edu

VI. Project Execution

A. Web application

Fall semester - foundation. The web client was established as the main interface for teachers and students with layouts covering dashboards, navigation and forms. Early work was focused on getting the project structure right: setting up Vite, React, Typescript, wiring routing and authenticated shells and building the core student flow with sign-up, login, session handling and the first integration for study sets and practice. A lot of effort was put into establishing a consistent Material UI based layout early on, so the later pages such as gamification, analytics, classes could fit into the same design structure.

Spring semester - broadening and bringing it all together. This semester the focus shifted to end-to-end functionality: class creation, enrollment assignments tied to study sets, views of progress and statistics, analytics and gamification boards, AI-assisted study sessions with explain, hint, feedback endpoints and admin portal. We expanded the internationalization using react-i18next adding support for English, Russian and Kazakh so the interface matched the user's course language. There also was a significant focus on fixing issues before the demo: resolving API mismatches between backend and frontend, improving how errors are handled and testing the main user flow from start to finish (login -> dashboard -> study set -> practice -> progress).

Decision	Rationale
Single-page application with React Router	Fast navigation between dashboard, classes, practice and settings without reload, keeps users logged in
Vite as the build tool	Fast dev feedback during iterative UI work
TypeScript end-to-end in the client	Safer update for REST API as it became larger and more complex
MUI for layout and components	Accessible defaults, faster delivery of dashboards with lots of data
Centralized API modules	Single place for base URL, headers, error handling
React-i18next	Multilingual UI without component

	duplication for each language
Route guards	Clear separation of public/login routes vs student/teacher vs admin

Table 2. Design Decisions for the Website

What went well: splitting the UI into self-contained components, separate for classes, assignments and study sets, meant that different members could work on different parts at the same time and not get in each other's way. With having shared layout components, we could keep every new page looking consistent without extra design work. Typescript and ESLint also caught a lot of bugs early.

What was difficult: keeping up with a fast-moving API was probably the biggest challenge. With backend changing frequently, things on the frontend would break temporarily and would need quick attention. State management became tricky with screens that showed a lot of information like assignment with due dates and progress tracking (edge case bugs would always creep in). OAuth redirect with URLs (login with Google) on localhost was tricky during local development because redirect links had to be configured exactly right. In addition, we would face token refresh race condition where the app tried to refresh logintoken at the wrong time in the wrong order. Updates to MUI UI library caused parts of the interface to stop working correctly.

Finally, studySetApi module grew extremely large and was hard to manage.

How problems were dealt with: when something broke, the go-to approach was to reproduce the issue step by step using browser dev tools to figure out what went wrong. When the backend changed, the team would sync up quickly to ensure that every member is on the same page. We kept the merges small and frequent to make sure that the codebase stays in a working state.

B. Mobile application

Spring Semester - planning and execution. This semester, a mobile application was created to expand the project. The foundations of web application were used extensively to ensure functionality of the mobile client. It was decided to make it using React Native to ensure cross-platform compatibility and component reuse. authApi and studySetApi modules previously created for the web client were also used for the mobile client. Through that, our team was able to make a functional mobile application that

has most of the features of a web client. Logging in, managing classes and study sets, seeing progress, completing a study set worked well

Decision	Rationale
Creating an application using React Native	Cross-platform compatibility, reuse of web application components
Using Expo framework	Fast and easy testing of the application, decreasing the need for third-party components
Creating multiple tabs	To improve User Experience (UX)
Making spinner refresh	To ensure on-time data retrieval

Table 3. Design Decisions for the Mobile Application

What went well: Firstly, splitting the app into multiple tabs so that the user can access necessary information from the bottom menu. Secondly, reuse of web application API and the same backend and database ensuring full synchronization between both environments.

What was difficult: In the beginning, it was difficult to set up the mobile application such that it can access the backend. The registration and login routes kept crashing.

How problems were dealt with: It was possible to solve the issues by modifying the API module. By adapting it to Expo framework features we were able to make it work properly and in sync with the backend and the database.

C. Division of responsibilities and leadership:

The project was carried out with a structured division of responsibilities with team members contributing in different aspects of design, implementation, testing and evaluation. One team member assumed a coordination role overseeing the overall system and was also responsible for the web application (both backend and frontend). The initial foundation of backend was established by a former team member in the Fall semester but it had to be extended and maintained throughout the whole development process. Another team member focused on mobile application implementing core features. During the first semester this member also contributed to the design and implementation of selected web components. The remaining two team members were responsible for testing the platform functionality and

usability and carrying out the evaluation process with potential users (students and teachers). In addition to those activities, one of these members implemented password reset functionality.

Throughout the project the team followed iterative development process incorporating elements of Scrum to organize work. Development was carried out in short cycles during which specific features and components were implemented.

VII. Evaluation

To see if Nova Edu actually helps students manage their work and get a more personalized learning experience, we tested the platform with two main groups: students and teachers.

How we tested it. We had users try out all the main features, such as setting up accounts, creating study sets, using the AI tools, and testing the rewards system. Afterward, we collected their feedback through two surveys to see how satisfied they were and how well the site actually worked for learning.

What students thought. Overall, students were highly satisfied with the platform, with the majority giving it a perfect **10/10** rating.

- **Gamification:** This was a standout success. Nearly **90%** of students agreed that earning points, badges, and hitting milestones actually motivated them to study more often.
- **AI Features:** Students found the AI very helpful for guiding their studies, giving it a strong **4.4/5** rating. Some did mention they'd like even more accuracy in topic suggestions and more detailed progress reports as the system grows.
- **Ease of Use:** Core tasks like signing up or resetting a password were rated as **"Very Easy"** by every participant, confirming a smooth user experience with no technical frustrations.

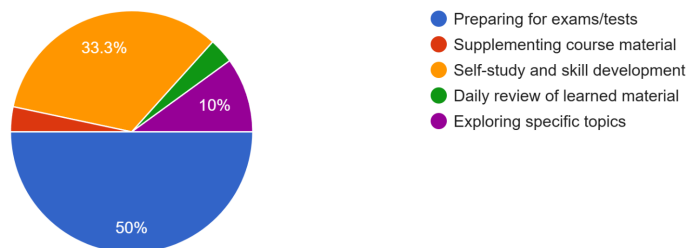


Figure 3. Questionnaire for Students: Question 1

On a scale of 1 to 10, how satisfied are you with the overall website experience?
31 responses

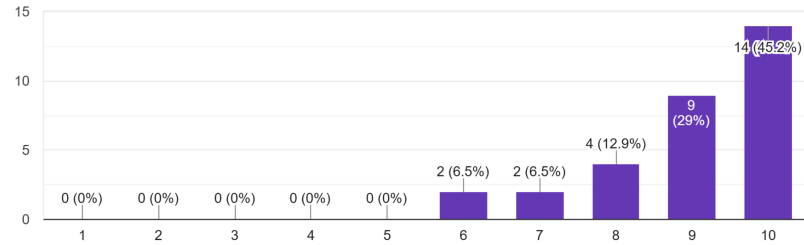


Figure 4. Questionnaire for Students: Question 2

Please rate the following aspects of the 'Study Sets' feature:

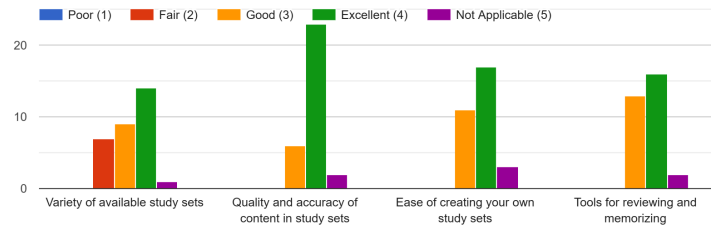


Figure 5. Questionnaire for Students: Question 3

How helpful are the AI features in guiding your study path and identifying areas for improvement?
31 responses

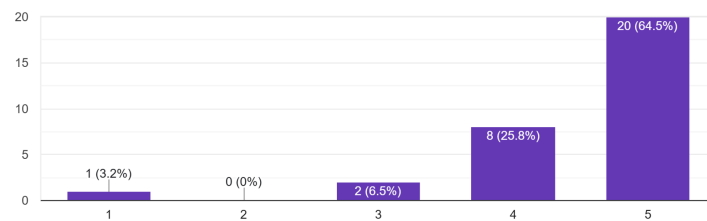


Figure 6. Questionnaire for Students: Question 4

Which parts of the AI recommendation system could be improved?

29 responses

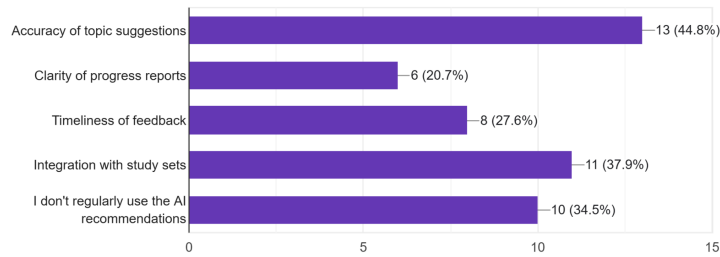


Figure 7. Questionnaire for Students: Question 5

Please rate your agreement with the following statements regarding the Gamification features (milestones, points, badges, etc.):

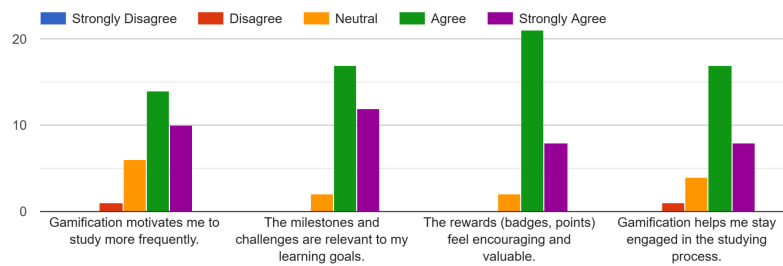


Figure 8. Questionnaire for Students: Question 6

How easy was it to complete the following actions?

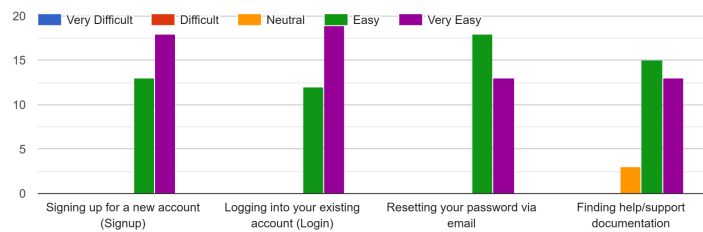


Figure 9. Questionnaire for Students: Question 6

What should be improved:

- **Content & Materials:** The primary request is for **More Study Sets**, specifically needing a wider range, variety, and more public availability of study materials and tasks.
- **AI and Interactivity:** Suggestions include adding **more options for AI assistance and integration** and improving the platform's social and collaborative features, such as integrating ways to **study together with shared materials/threaded discussions** on difficult problems.
- **Feature Enhancements:** Respondents recommended **improving Gamification**, adding a **dark mode theme**, and potentially adding a **chat function to communicate with teachers** directly on the website.
- **General Satisfaction:** A significant number of responses indicated that **everything is fine/good**, or they had **no suggestions** for improvements, with some expressing satisfaction with the current state of the website.

General Functionality: The overwhelming majority of respondents indicated they experienced no technical issues, bugs, errors, or slow loading while using the Estudy website. **Positive Experience:** Responses frequently stated that "everything worked great," "everything was fine," or "The site worked well."

Would you recommend this study website to a friend or classmate?
31 responses

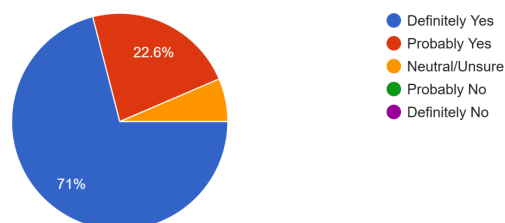


Figure 10. Questionnaire for Students: Question 8

What professors thought. Teachers from various subjects like Math, English, History, and even Cryptography also tested the site.

Reliability: They gave the platform a **9.6/10** for reliability, finding the study sets and AI hints extremely effective for keeping students engaged.

Security: Every professor surveyed felt the authentication system was very secure and safe for student use.

Tech Integration: They gave the site a high **6.6/7** rating for how well it blends modern tech like AI with traditional teaching methods.

Future Requests: Some teachers suggested adding tools to track individual student performance more closely and a way to message students directly within the app.

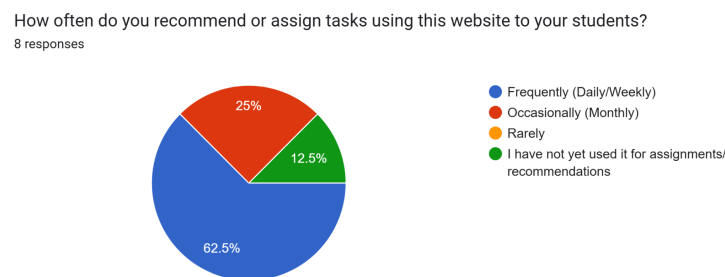


Figure 11. Questionnaire for Teachers: Question 2

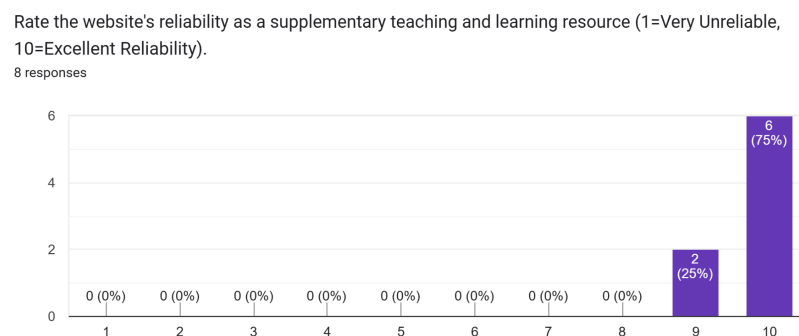


Figure 12. Questionnaire for Teachers: Question 3

Please evaluate the usefulness of the following features for student engagement and learning:

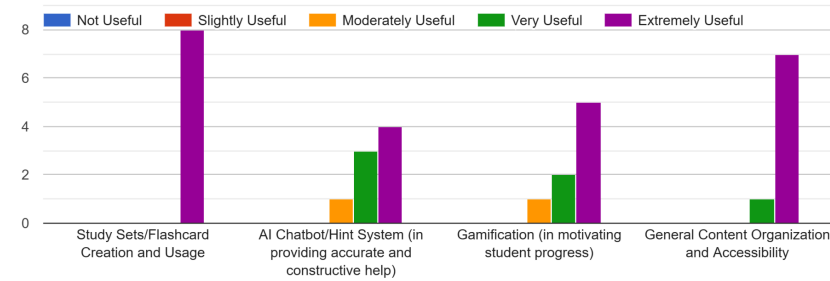


Figure 13. Questionnaire for Teachers: Question 4

Based on your observations, how secure and private does the authentication system (Login/Password Reset) appear for students?

8 responses



Figure 14. Questionnaire for Teachers: Question 5

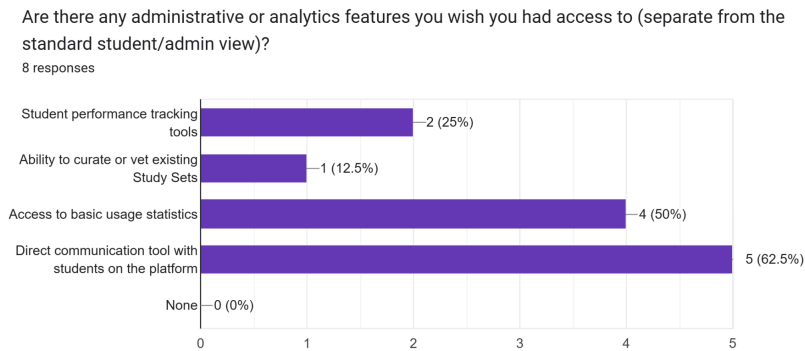


Figure 15. Questionnaire for Teachers: Question 6

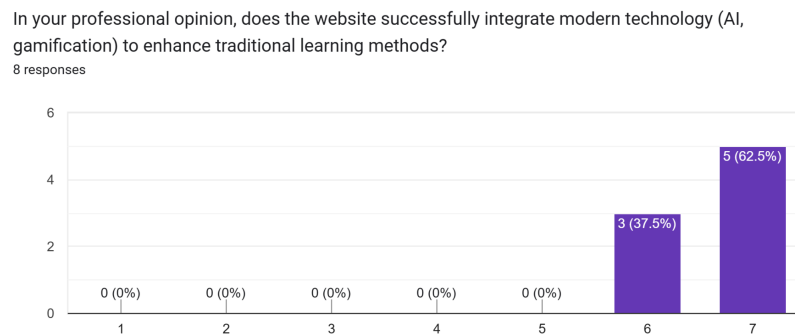


Figure 16. Questionnaire for Teachers: Question 7

Final Verdict. The feedback shows that the project did exactly what it set out to do. With 100% of students saying they would recommend it and professors giving it nearly perfect scores, Nova Edu is clearly an effective tool for the modern classroom.

VIII. Conclusion and Possible Future Work

In conclusion, the project was successful and it provided the basic and necessary functionalities. There were multiple unexpected problems that occurred along the way and the team was able to adjust to sudden changes and focus on providing the highest quality of product that is possible considering the known and unknown factors.

Also, the integration of AI assistance processed as planned. While the current state of the AI-Enhanced Nova Edu study platform creates a strong foundation, there are still improvements that could be made to the project:

- Communication functionality can be added so that the Teachers and the Students can chat with each other through the educational platform. This might be implemented as a separate chatting system or as a support display that connects the Students with available Teachers.
- Improve the mobile client to the point where it would provide the web application's full functionality on mobile devices. This makes the access to the platform comfortable and easy to reach as the users could access it wherever they are and whenever they have just their phones available.
- A lot more study sets could be added to strengthen the core of the platform and provide a wide range of material for users to use from the very beginning. This can help to keep the users' interest in the website and urge them to use the platform's functionalities to the full extent.
- The existing AI features of the platform might be updated and extended to new ranges. One of the examples of integrating AI features into the platform's functionalities is - adding an AI assistance to study sets creation. This might help users create new study sets, which will lead to rapid growth in the number of study sets on the platform. And this growth might attract new users to the platform.

References

- [1] K. VanLEHN, "The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems," *Educational Psychologist*, vol. 46, no. 4, pp. 197–221, Oct. 2011, doi: <https://doi.org/10.1080/00461520.2011.611369>.
- [2] X. Wang *et al.*, "The Efficacy of Artificial Intelligence-Enabled Adaptive Learning Systems From 2010 to 2022 on Learner Outcomes: A Meta-Analysis," *Journal of educational computing research*, vol. 62, no. 6, May 2024, doi: <https://doi.org/10.1177/07356331241240459>.

- [3] Hariyanto, F. X. D. Kristianingsih, and R. Maharani, “Artificial intelligence in adaptive education: a systematic review of techniques for personalized learning,” *Discover Education*, vol. 4, no. 1, Oct. 2025, doi: <https://doi.org/10.1007/s44217-025-00908-6>.
- [4] M. Shortt, S. Tilak, I. Kuznetcova, B. Martens, and B. Akinkuolie, “Gamification in mobile-assisted language learning: a systematic review of Duolingo literature from public release of 2012 to early 2020,” *Computer Assisted Language Learning*, vol. 36, no. 3, pp. 517–554, Jul. 2021, doi: <https://doi.org/10.1080/09588221.2021.1933540>.
- [5] “Creating flashcard sets,” *Quizlet Help Center*, 2025. <https://help.quizlet.com/hc/en-us/articles/360029780752-Creating-flashcard-sets>.
- [6] “Kahoot! Blog,” Kahoot!, Mar. 08, 2019. <https://kahoot.com/blog/>.
- [7] “Classroom Help,” Google.com, 2019. <https://support.google.com/edu/classroom/>.
- [8] “The Keyword,” *www.blog.google*. <https://blog.google/>.
- [9] Duolingo, “Research - Duolingo,” *ai.duolingo.com*. <https://research.duolingo.com/>.