

АНОТАЦІЯ

Дипломна робота є складовою частиною проекту по розробці платформи для дистанційного навчання вищої математики, орієнтованої на студентів не математичних спеціальностей.

Метою роботи є проектування і створення інтерфейсу користувача для даної платформи.

Користувачами платформи є студенти і викладачі університетів, при цьому в ролі студента може виступати будь-який користувач.

Проведено порівняльний аналіз і сформульовані вимоги до проектування інтерфейсу дистанційній платформі для навчання вищої математики, описані сценарії взаємодії користувача з системою, які лягли в основу розробки бази даних, спроектовані і побудовані основні шаблони системи, виконана інтеграція інтерфейсу в архітектуру системи.

Для проектування шаблонів використаний онлайн сервіс Figma. Для реалізації шаблонів застосований фреймворк Vue.js.

АННОТАЦИЯ

Дипломная работа является составной частью проекта по разработке платформы для дистанционного обучения высшей математике, ориентированной на студентов не математических специальностей.

Целью работы является проектирование и создание интерфейса пользователя для данной платформы.

Пользователями платформы являются студенты и преподаватели университетов, при этом в роли студента может выступать любой пользователь.

Проведён сравнительный анализ и сформулированы требования к проектированию интерфейса дистанционной платформы для обучения высшей математике, описаны сценарии взаимодействия пользователя с системой, которые легли в основу разработки базы данных, спроектированы и построены основные шаблоны системы, выполнена интеграция интерфейса в архитектуру системы.

Для проектирования шаблонов использован онлайн сервис Figma. Для реализации шаблонов применен фреймворк Vue.js.

ABSTRACT

The graduation work is an integral part of a project to develop a platform for distance learning in higher mathematics, aimed at students of non-mathematical specialties.

The aim of the work is to design and create a user interface for this platform.

The users of the platform are students and university professors, and any user can act as a student.

A comparative analysis is carried out and requirements for the design of the interface of a distance platform for teaching higher mathematics are formulated, scenarios of user interaction with the system are described, which formed the basis for the development of a database, the main system templates are designed and built, the interface is integrated into the system architecture.

The online service Figma was used to design templates. The Vue.js framework was used to implement the templates.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І

ТЕРМІНІВ

6ВСТ

УП 71 ОГЛЯД ІСНУЮЧИХ СИСТЕМ

92 РЕАЛІЗАЦІЯ КУРСУ НА

ОСВІТНІЙ ПЛАТФОРМІ STERIK

223 ВИМОГИ ДО ПРОЕКТУВАННЯ І

РОЗРОБКИ ІНТЕРФЕЙСУ

293.1. Основні вимоги до призначеного для

користувача інтерфейсу

293.2. Вимоги до проектування призначеного для

користувача інтерфейсу освітньої платформи

313.3. Сценарії використання

інтерфейсу платформи

344 ПРОГРАМНА РЕАЛІЗАЦІЯ

ІНТЕРФЕЙСУ

384.1. Засоби реалізації інтерфейсу

користувача

384.2. Взаємодія модулю інтерфейсу з серверною

частиною

404.3. Побудова

шаблонів

424.4. Реалізація

інтерфейсу

45ВИСНОВ

КИ 52ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

53ДОДАТОК А СЦЕНАРІЇ

ВИКОРИСТАННЯ

55ДОДАТОК Б ШАБЛОНИ ІНТЕРФЕЙСУ

КОРИСТУВАЧА МАТЕМАТИЧНОЇ ПЛАТФОРМИ

60

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення

UI – User Interface (користувальницький інтерфейс)

LMS – Learning Management System (система управління навчанням)

SPA – Single Page Application (односторінковий додаток)

MPA – Multi Page Application (багатосторінковий додаток)

API – Application Programming Interface (програмний інтерфейс програми)

REST – Representational State Transfer (передача стану уявлення)

DOM – Document Object Model (об'єктна модель документа)

CLI – Command Line Interface (інтерфейс командного рядка)

ВСТУП

Велика кількість студентів університетів, які займаються крім математики своєю професією, не дуже добре сприймають свій математичний досвід і не усвідомлюють важливість математики. Ця проблема поширена для таких областей, як освіта, інженерія, бізнес і науки про життя.

Курс вищої математики для студентів не математичних спеціальностей повинен будуватися з оглядом на завдання, які вирішуються в рамках конкретної спеціальності, але традиційні методи викладання математики це, як правило, не враховують.

Цікавим досвідом просування в даному напрямку є підхід до викладу математики, запропонований Володимиром Семеновичем Ітенбергом [1]. Він описав проблему поганого сприйняття математики, в такий спосіб. Математика – це не зовсім навчальна дисципліна, а швидше метод системного мислення. В існуючому навчанні математики відсутня емоційний компонент, як результат – нудьга і відсутність інтересу серед студентів. Основним недоліком викладання математики в університетах є відсутність програм і методів, які були б призначені спеціально для не математиків. Тому розробка правильної навчальної програми є найважливішим завданням. В даний час ознайомитися з цим підходом можна по відео-уроку в інтернеті [2], де є приклади математичних фактів, які викликають інтерес і, як результат, засвоюються набагато успішніше, ніж при стандартному викладі. Більшого розповсюдження даного досвіду сприяла би наявність відповідного онлайн-курсу. Традиційні лекції в силу масовості аудиторії позбавлені можливостей прямої зворотного зв'язку викладача з кожним із студентів. Сучасні комп'ютерні засоби дозволяють істотно нівелювати цей недолік і інтенсифікувати процес викладання. За допомогою комп'ютерних технологій набагато об'ємніше можна перевіряти рівень знань і розуміння будь-якого навчального курсу у вигляді тестів. Така перевірка не тільки можлива, а й має низку істотних переваг в порівнянні з традиційними формами контролю.

Метою даного проекту є проектування і створення інтерфейсу освітньої платформи для вивчення математики математиками. Завдання побудови відповідного інтерфейсу вирішується в рамках створення освітньої платформи для вивчення математики [3].

Платформа будується на базі мікросервісної архітектури [4], де клієнт через загальний користувальницький інтерфейс (UI) отримує доступ до окремих слабо пов'язаних між собою мікросервісах.

В інформаційній системі передбачено три типи користувачів: гість, викладач і студент. При цьому:

- гість зможе проходити вхідний тестування і пробний модуль курсу для особистої оцінки матеріалу і власних можливостей;
- викладач отримає можливість створювати свій курс з урахуванням рекомендацій системи і керувати класами студентів;
- студент зможе проходити курс, який адаптується під його успішність.

Для досягнення зазначеної мети в роботі необхідно вирішити наступні завдання:

- 1) виконати аналіз існуючих навчальних рішень;
- 2) перевірити існуючу навчальну платформу на практиці;
- 3) сформулювати вимоги до створюваного інтерфейсу користувача;
- 4) описати сценарії взаємодії;
- 5) вибрати засоби реалізації інтерфейсу;
- 6) спроектувати шаблони майбутнього інтерфейсу;
- 7) реалізувати необхідні шаблони.

1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ

В ході проведення аналізу існуючих навчальних курсів були розглянуті різні онлайн платформи з відмінним призначенням для користувача інтерфейсом в програмному забезпеченні для освіти, що надають можливість вивчати математику. Серед них EdX [3], Moodle [6], Khan Academy [4], Stepik [5], EdEra [6], які мають свої відмінні риси серед інших існуючих навчальних платформ.

EdX – надійна платформа для освіти і навчання, глобальна некомерційна організація. Заснована і управляється коледжами та університетами. Курси досить складні і вимагають регулярного виділення часу для вивчення уроків.

Методика навчання EdX полягає в наданні лекцій, поділених на модулі тривалістю до 10 хвилин, які чергуються з вправами для кращого засвоєння матеріалу. При отриманні неправильної відповіді програма аналізує помилку і видає рекомендації для її виправлення. Платформа має зручний веб-інтерфейс і мобільний інтерфейс для учнів, а також індивідуальний зовнішній вигляд, відповідний бренду.

Для приватних курсів вартість видачі цифрового сертифікату вже включена у вартість курсу. Однак за безкоштовні курси EdX стягується плата за видачу цифрового сертифікату. Варто пам'ятати, що видача цифрових сертифікатів є додатковою послугою, так що учень може прийняти рішення пройти курс безкоштовно і не оплачувати цю додаткову послугу, якщо він не хоче отримати сертифікат, що підтверджує його навчання.

Найчастіше курси англійською мовою, але всі матеріали надані в такому вигляді, що з легкістю можна перевести. Відео йде з субтитрами, які можна як включити, так і відключити на самому відео. Також наявність списку субтитрів праворуч від відео, що дозволяє переключитися на певний момент, який може необхідно прослухати ще раз. Приклад такого інтерфейсу можна побачити на рис. 1.1.

< Previous ✓ 📄 🎥 📄 Next >

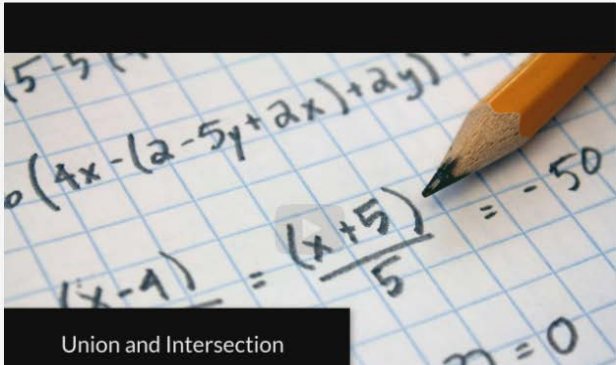
Union and Intersection

🔖 Bookmark this page

Let's spend a little time talking about Union and intersection. For Union we use the letter U to represent it. A key word when we are talking about union is the word or. This is where we're combining or joining two different sets.

Intersection is like an upside-down U (\cap) and the key word that you'll hear when we're talking about intersection is the word and. These are things that are part of both sets. Here is an example to help us think through the difference.

Video



Начало субтитров. Пропустить до конца

Here's 12 pictures of some people. The question is who has blonde hair and glasses?

Take a moment and think about how you would respond to that question.

Hopefully you would identify Joe. If I were to think of the sets of people that

Рисунок 1.1 – Приклад інтерфейсу платформи EdX

Структурована панель навігації по курсу допомагає швидко перемикатися між модулями і уроками. Навігація між уроками в процесі навчання інтуїтивно зрозуміла, і учень завжди може бачити, де вони знаходяться і скільки уроків залишилося для поточного модуля або глави. І легко зрозуміти, чи є якісь тести, які потрібно виконати.

Open edX – це навчальна платформа з відкритим вихідним кодом. У програмній системі Open edX викладачі можуть публікувати курси, розміщувати дошки обговорень, управляти групами і командами, редагувати оцінки і спілкуватися з учнями. Учні в свою чергу отримують доступ до свого профілю, можуть перевірити статус реєстрації та переглянути курси через панель моніторингу.

Коли справа доходить до змісту онлайн-курсу, важлива різноманітність. Open edX за замовчуванням дозволяє створювати онлайн-навчальні матеріали,

використовуючи широкий спектр типів контенту: текст, зображення, відео, різні види завдань (рис. 1.2). Список можна розширити і налаштувати відповідно до потреб конкретного проекту (моделювання, віртуальна лабораторія і т.д.).

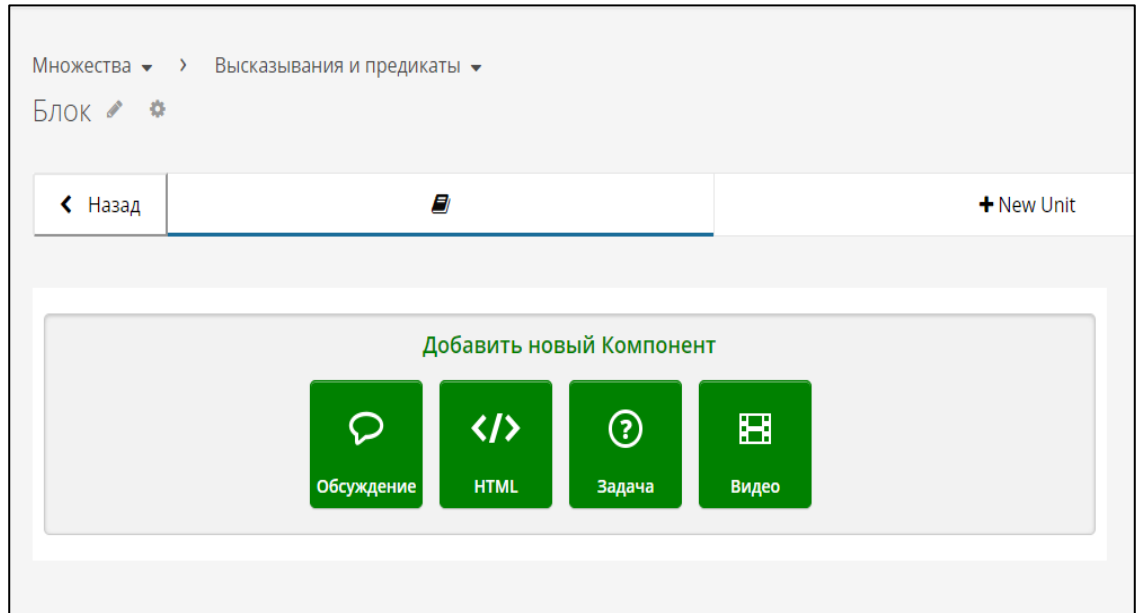


Рисунок 1.2 – Приклад інтерфейсу системи Open edX

Moodle – це платформа управління навчанням (LMS) з відкритим вихідним кодом, призначена для допомоги школам в навчанні своїх учнів. Moodle заснований на модульній конструкції, яка дозволяє вчителям та адміністраторам створювати власні навчальні програми, використовуючи плагіни для різних робочих процесів, контенту і занять.

У користувачів є вибір: встановити свій профіль Moodle на своїх серверах або в хмарі. Хмарна платформа під назвою MoodleCloud надає кілька переваг, таких як доступність в будь-який час, масштабованість і досить короткий процес впровадження. Тарифи MoodleCloud призначені тільки для 1000 користувачів. Організації, яким потрібно план на 1000 користувачів, повинні будуть працювати з Партнером Moodle.

З точки зору налаштування і всього, що з цим пов'язано, Moodle вважається найпотужнішою LMS. Існують сотні плагінів і графічних тем,

створених спільнотою Moodle, які здатні кардинально змінити зовнішній вигляд платформи і її можливості. Однак всі ці можливості доступні тільки в автономній версії Moodle.

Щоб мати можливість додати навчальне завдання і призначити його студентам, необхідно створити свій перший курс. Це робиться через відповідну вкладку в меню «Адміністрування сайту» (рис. 1.3). Вбудований редактор вікторин має безліч різних типів питань на вибір, включаючи перетягування, але, як і з іншою частиною Moodle, може втомлювати працювати через надмірну кількість елементів управління, багато з яких не будуть корисними для більшості користувачів.

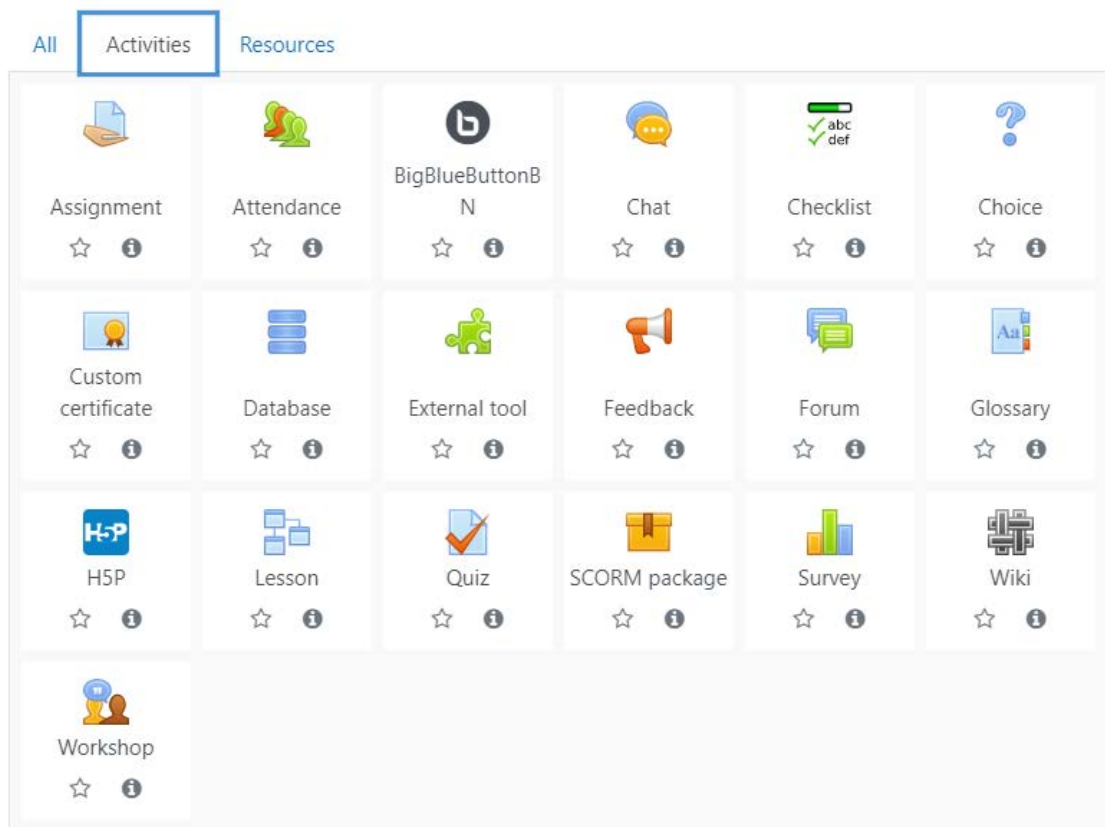


Рисунок 1.3 – Приклад інтерфейсу платформи Moodle

Деякі користувачі також повідомляють, що Moodle не завжди зручний для користувача або інтуїтивно зрозумілий. Необхідно багато терпіння, щоб звикнути до інтерфейсу Moodle з кнопками і опціями.

Інтерфейс для учнів не сильно відрізняється від адміністративної версії, в ньому просто відсутня кілька кнопок, які забезпечують доступ до контенту і налаштувань облікового запису (рис. 1.4).

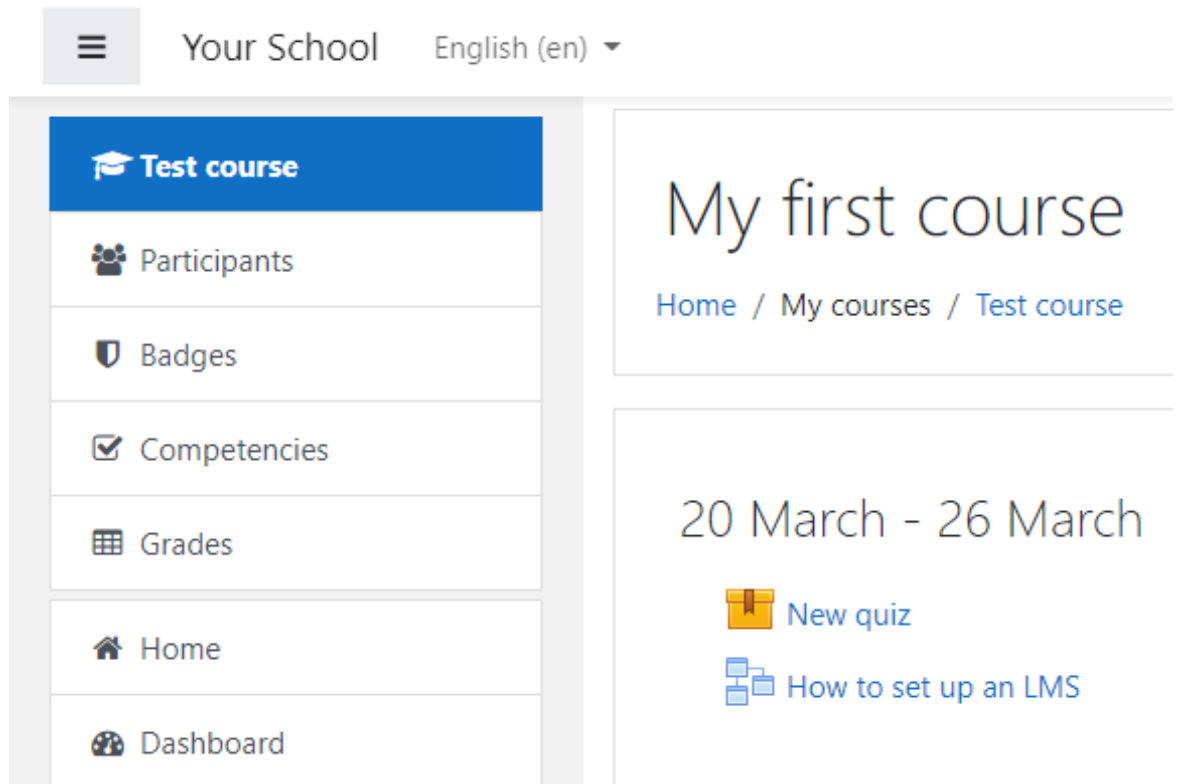


Рисунок 1.4 – Приклад інтерфейсу платформи Moodle

Також Moodle пропонує користувачам власний мобільний додаток. Він надає їм доступ до курсів, оцінками і досягнень, а також дозволяє учням завантажувати призначені навчальні матеріали на свої смартфони.

Khan Academy – некомерційна освітня організація. Пропонує учням практичні вправи, навчальні відео та персоналізовану панель управління. Використовується технологія індивідуальної адаптації, яка визначає, як сильні, так і слабкі сторони кожного учня.

Khan Academy обслуговує студентів американської системи освіти. Багато курси відповідають формальній навчальній програмі для різних рівнів навчання. Крім математики, економіки і фінансів, Khan Academy займається наукою і інженерією, обчисленням, мистецтвом та гуманітарними науками,

підготовкою до тесту, підготовкою до коледжу, підготовкою до кар'єри і мистецтво англійської мови. Для більшості з цих тем є спеціальні курси.

Його призначення – «забезпечити безкоштовну освіту світового рівня для всіх і в будь-якому місці». Таким чином, Khan Academy повністю безкоштовна. Не потрібно навіть створювати обліковий запис і входити в систему, щоб отримати доступ до уроків.

У разі створення навчальної записи і входу в неї, можна відстежувати свої успіхи в різних курсах і отримувати доступ до інших функцій. Одна з цих додаткових функцій – надати вчителю, репетитору або батькам доступ до історії навчання і дати можливість ставити перед учнем задачі або завдання.

Khan Academy побудована на основі відео-лекцій, коротких читань, інтерактивних вікторин (зазвичай з декількома варіантами відповідей) і ланцюжків коментарів. При записі на курс, відкривається доступ до всіх відео-лекцій та інших матеріалів, які є в цьому курсі, і вони представлені в послідовному порядку. Як правило, курс містить блоки, які містять уроки, а всередині уроків – відео, матеріали для читання і тести (рис. 1.3).

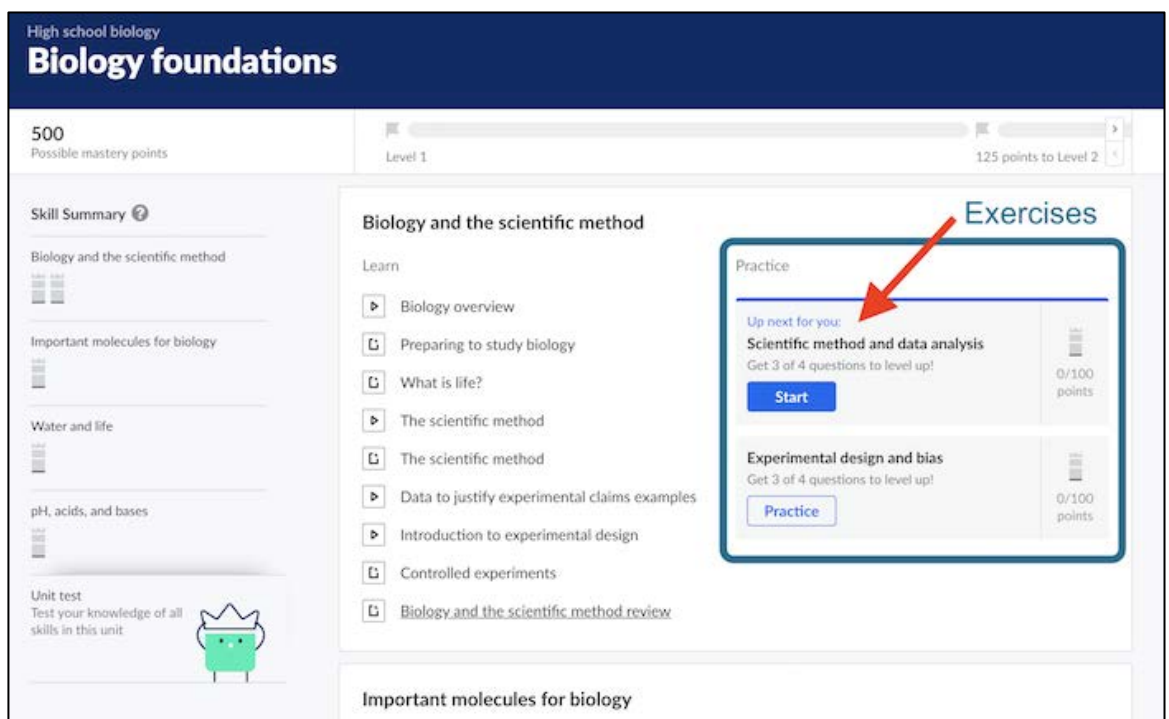


Рисунок 1.5 – Приклад інтерфейсу платформи Khan Academy

У багатьох випадках послідовність уроків і розділів має значення, тому що необхідно придбати базові навички, перш ніж поглибити свої знання. Курси математики – гарний тому приклад. Однак, якщо у студента вже є певний навик, можна пропустити його вперед. На деяких курсах може мати сенс відразу перейти до матеріалу, який необхідно вивчити або переглянути, замість того, щоб вивчати кожен урок по порядку.

Курси в Khan Academy відносяться до групи навчальних матеріалів. На деяких курсах можуть бути інструктори, які відповідають на коментарі, але немає призначеного інструктора, який приділив би індивідуальну увагу при реєстрації на курс. Точно так само немає офіційних однокласників або графіка, в якому потрібно опрацювати матеріал. Платформа повністю самостійна.

Сайт відстежує перегляд матеріалу учня. Він фіксує, скільки витрачається на перегляд відео-уроків, читання текстів, а також бали в тестах. Також накопляються бали майстерності, коли учень проходить матеріал (рис. 1.4), а також значки за досягнення різних етапів.

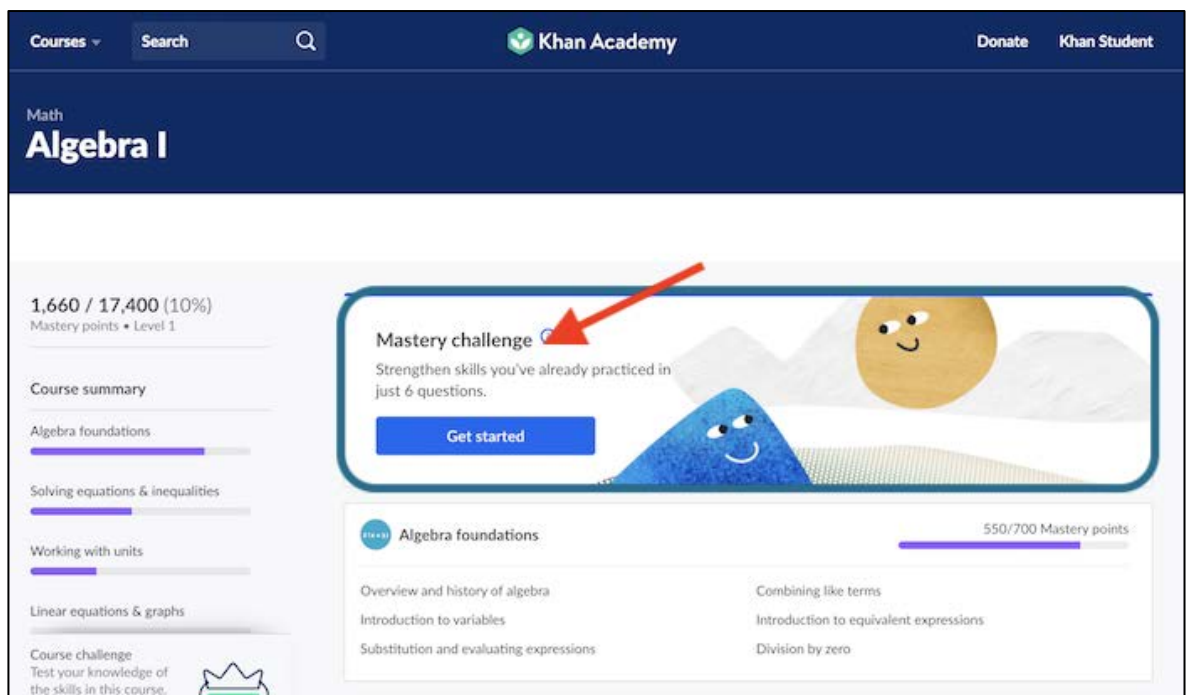


Рисунок 1.6 – Приклад інтерфейсу платформи Khan Academy

При проходженні короткої вікторини, що складається всього з чотирьох питань, можливо майже гарантовано не побачити повторення одних і тих же питань, якщо пройти її ще раз в той же день.

Stepik – освітня платформа і конструктор онлайн-курсів. Stepik активно розвиває напрямок адаптивного навчання, де кожен зможе вивчати матеріал, підібраний індивідуально під свій рівень знань. Також ця платформа надає можливість отримання сертифікату.

Створюваний курс складається з окремих уроків, згрупованих в тематичні модулі. Вони існують окремо або збираються в бібліотеку на платформі. Уроки складаються з «кроків», які можуть бути текстом, відео-лекцією або практичним завданням (рис. 1.5). Кожен «крок» – це картинка, невеликий текст, посилання, питання або тест. Суворої структури немає – можна здійснювати перевірку знань після кожного «кроку». Є можливість використовувати 20 типів завдань, включаючи тести, числові завдання, завдання з математичними формулами і хімічними рівняннями, пазли, завдання на програмування.

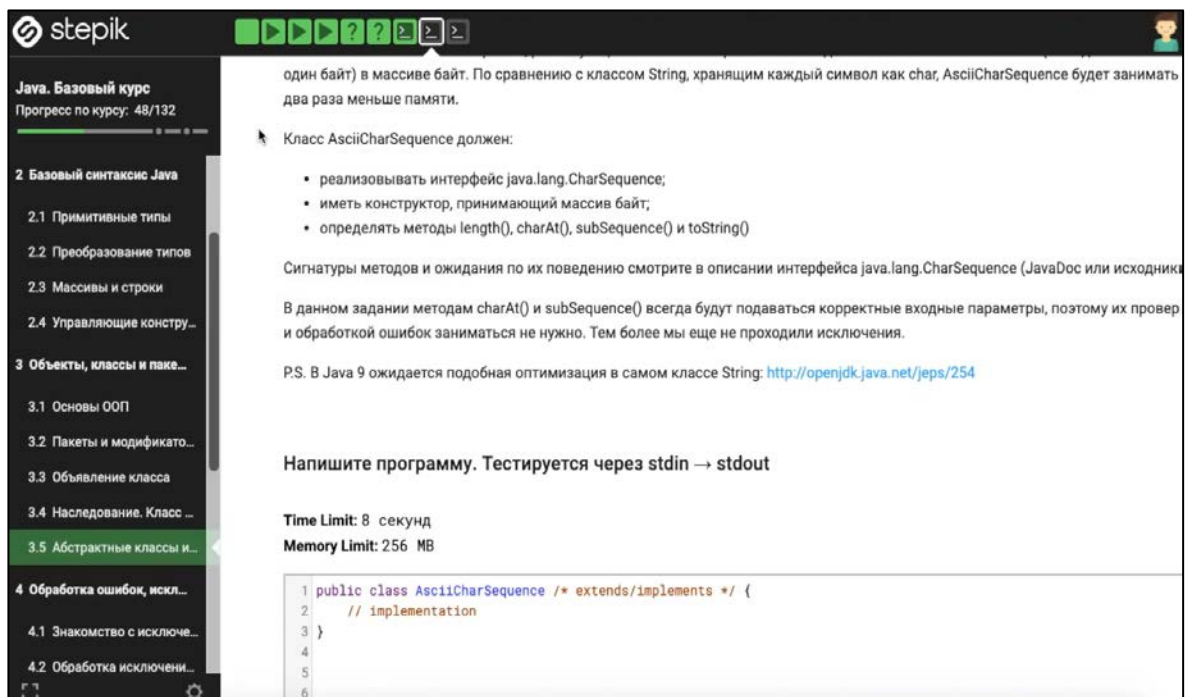


Рисунок 1.7 – Приклад інтерфейсу платформи Stepik

Для викладача має зручний і зрозумілий інтерфейс додавання модулів і «кроків» курсу (рис. 1.6), а також зручний редактор тексту (текст з форматуванням, зображеннями, математичними формулами), особливо, якщо писати з нуля. Різноманітність «кроків», але відсутність можливості об'єднати їх в один. В налаштуваннях до уроків можна змінювати шрифт, колір і розмір тексту, вставляти малюнки, посилання, математичні формули, прикріплювати додаткові матеріали. Аналітика освітнього процесу дозволяє стежити за прогресом учнів.

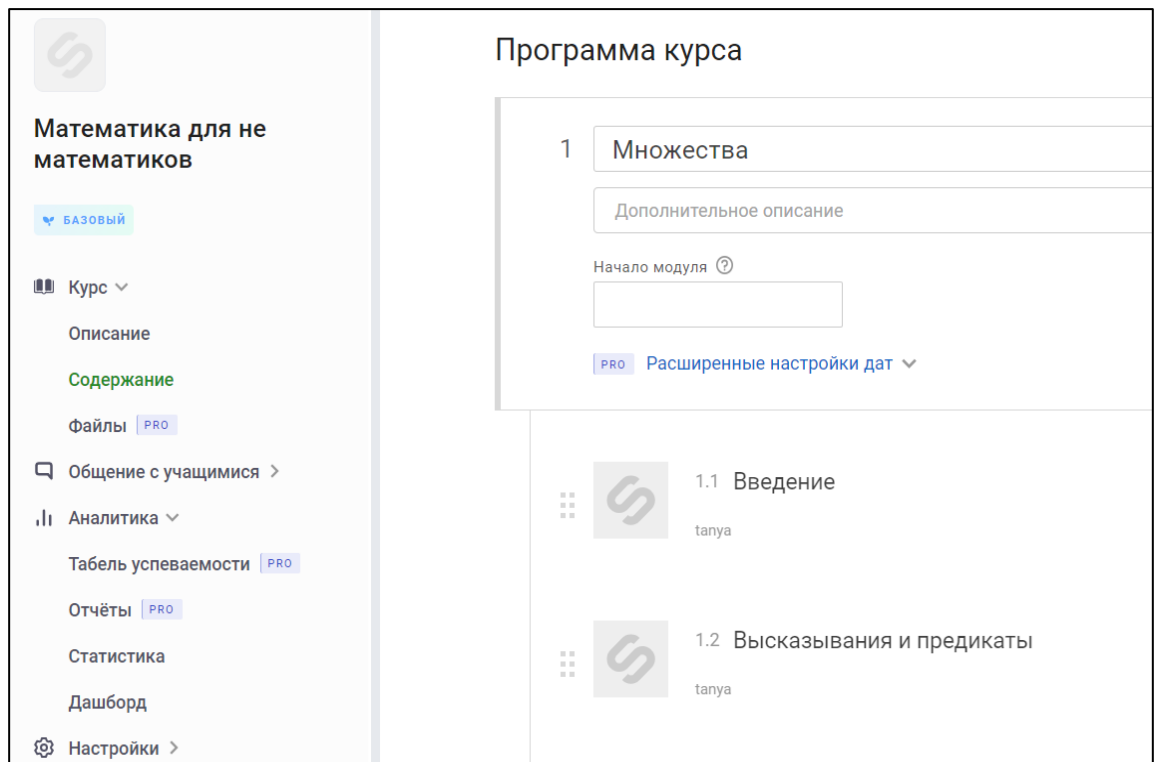


Рисунок 1.8 – Приклад інтерфейсу платформи Stepik

Створені матеріали (курси або окремі уроки), можна зберігати для самостійної підготовки користувачів, вбудовувати на інші сайти і освітні платформи. Можливість вбудовувати інтерактивні уроки, відео-фрагменти, проводити олімпіади та конкурси. Ресурс дозволяє конструювати індивідуальну освітню траєкторію.

Тільки в платній версії платформи можна проводити онлайн-іспити, використовувати файли в рішеннях задач з вільним відповіддю і текстової задачі, обмежувати на кількість спроб і знижувати бали, перевіряти рішення тільки викладачем, завантажувати оцінки учнів за завдання з файлу, переглядати розширений табель успішності.

Одна з хороших можливостей – це створення класу на основі курсу, де викладач зможе запрошувати своїх студентів і переглядати тільки їх успішність. Це можна використовувати для того, щоб підтягти всіх учнів до одного рівня або використовувати як додатковий навчальний матеріал.

EdEra – свіжий український проект, який представляє собою сайт з онлайн-курсами в форматі масових відкритих онлайн-курсів. Проект включає в себе інтерактивні лекції, конспекти з ілюстраціями і поясненнями, іспити і домашні завдання, спілкування з іншими учнями і педагогами. Мета проекту – зробити освіту доступною і якісною за прикладом західних кращих освітніх ініціатив. Також проект EdEra здійснює діяльність по запуску в Україні онлайн-курсів з підготовки до ЗНО.

Команда EdEra працює над тим, щоб освітні матеріали викликали емоції та були ефективними. Проект включає в себе лекції від професійних молодих викладачів з використанням різних форматів викладання і проміжними опитуваннями для кращого засвоєння матеріалу. Онлайн-курси відкриті для всіх бажаючих.

Один робочий тиждень складається в середньому з двох-трьох лекцій. Вони являють собою добірку коротких відео (6-10 хвилин), між якими інтегровані питання для контролю знань (рис. 1.7). Особливістю є те, що режим онлайн дає можливість змінювати формати подачі матеріалу в рамках однієї лекції. Усі навчальні матеріали – відео, додаткові матеріали, завдання – зберігаються на платформі та залишаються доступними, тому навчатися можна у будь-який зручний час.

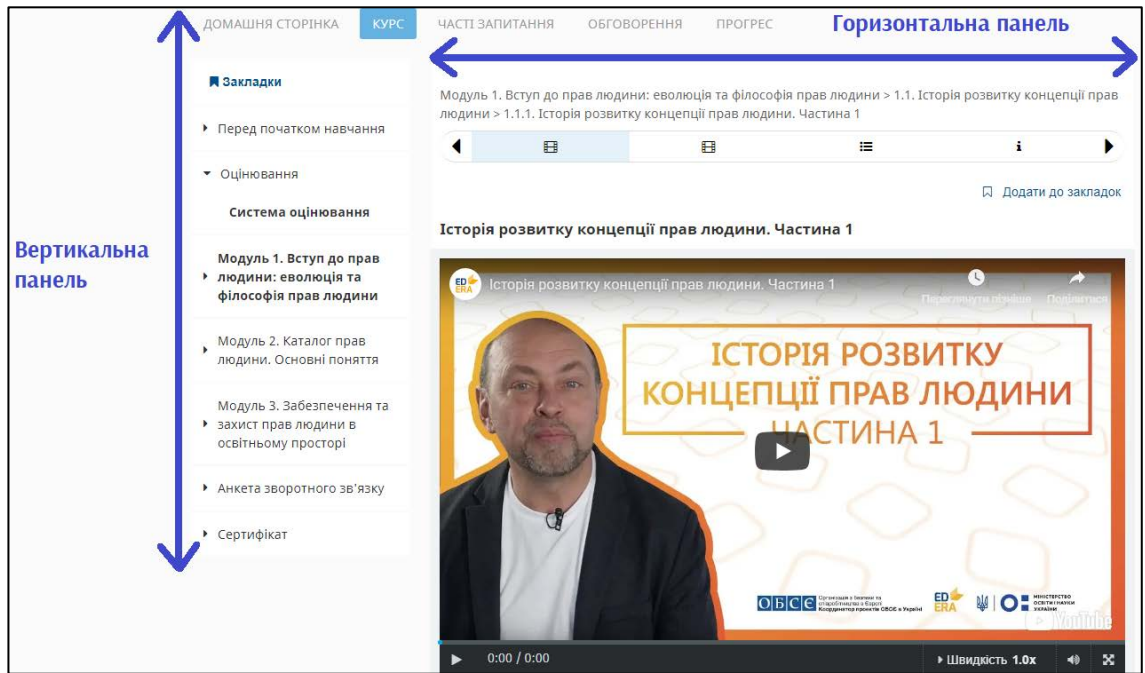


Рисунок 1.9 – Приклад інтерфейсу платформи EdEra

Проблемні завдання можна обговорювати з іншими учнями і педагогами, після чого матеріал може доповнюватися в залежності від ваших потреб (рис. 1.8).

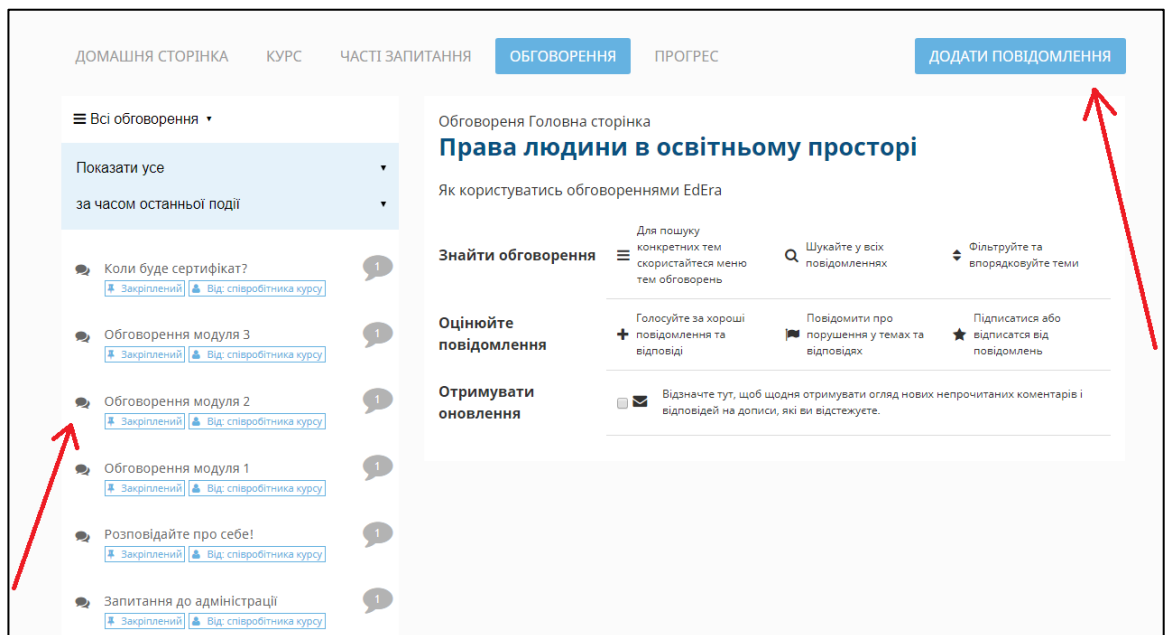


Рисунок 1.10 – Приклад інтерфейсу платформи EdEra

Щотижневі домашні завдання мають термін, до якого потрібно встигнути їх зробити. Також в середині і наприкінці курсу учень повинен скласти іспити. Це може бути тест або проектна робота. Всі результати потрапляють на динамічну сторінку прогресу учня, де формується остаточна оцінка. Щоб отримати сертифікат про успішне проходження курсу, ця оцінка повинна бути вище встановленої межі.

Кожна розглянута платформа має свої відмінні риси і переваги. Для викладача відповідними будуть платформи EdX і Stepik, так як вони мають можливість створення власного курсу і управління прогресом учнів. Команди Khan Academia і EdEra розробляють свої курси самостійно, тому є гарним вибором для самостійного використання учням. Варто так само відзначити складність користування іноземною платформою як для викладача, так і для студента. Замість того щоб відразу приступити до поставлених завдань, спочатку потрібно розбиратися в англomовному інтерфейсі. Більш наочно подивитися відмінності між даними платформами можна в таблиці 1.1.

Таблиця 1.1 – Порівняння освітніх платформ

Можливість	EdX / Open edX	Moodle	Khan Academy	Stepik	EdEra
Створення курсу	+	+	-	+	-
Багатомовний інтерфейс	+-	+-	+-	+	-
Можливість розміщення курсів на різних мовах	+	+	-	+	-
Редактор формул	+	+	+	+-	-
Побудова графіків	-	-	+	-	-
Засоби побудови тривимірних об'єктів	-	-	-	-	-
Побудова курсу з урахуванням вхідного тестування	-	-	+	-	-
Адаптивність курсу по ходу проходження	-	-	+	-	-

Продовження таблиці 1.1

Різноманітність математичних задач	+	+	+	-	-
Рекомендації по створеному курсу	-	-	-	+	-
Інтелектуальний аналіз даних	-	+	+	+	-
Вхідне тестування	-	-	+	-	-
Відкритий код	+	+	+	+-	+

Розглянуті платформи мають різні математичні курси з можливістю перегляду візуально або у вигляді тексту лекції, а також можливість протестувати вивчений матеріал. Хоча вони мають схожу структуру навчання, інтерфейс значно відрізняється. Для великих платформ мінусом є ускладнення пошуку певного курсу, не завжди зрозуміло чи це те, що було необхідно. Також то наскільки надані курси різні і є самою головною відмінністю платформ один від одного. Якщо розглядати дані платформи в рамках математики, то можливостей не так багато, так як це лише невеликий курс серед багатьох інших. Існуючі математичні платформи пропонують можливість навчання шкільної математики або підготовки до відповідних іспитів.

2 РЕАЛІЗАЦІЯ КУРСУ НА ОСВІТНІЙ ПЛАТФОРМІ STEPİK

Для більш детальної перевірки і визначення оптимальних можливостей або недоліків навчальної платформи необхідно випробувати її в реальних умовах.

Серед розглянутих платформ, для реалізації курсу, була обрана Stepik, так як вона передбачає створення і публікацію курсу з боку викладача, а також є російськомовною. Курс був створений з використанням матеріалів В.С. Ітенбергу і розміщений у відкритому доступі [10].

Для роботи з сайтом в першу чергу необхідно виконати авторизацію або реєстрацію. Для реєстрації запитуються тільки ім'я і пошта з паролем, при цьому перевірки на коректність відсутні і можуть здійснюватися тільки з боку користувача.

Користувач в цій системі може виступати як учнем, так і викладачем, тому інтерфейс профілю об'єднує можливості проходити курс і створювати свій.

Курс – набір уроків на загальну тему, організованих в модулі (тижні). Бувають: відкритими (доступні в каталозі і пошуку), платними (потрібно їх оплатити, щоб почати вчитися), приватними (для доступу до них потрібно отримати спеціальне посилання-запрошення від викладача).

Курс має назву і може містити опис, який платформа радить заповнити для готовності курсу. Програма курсу (рис. 2.1) містить модулі (тижні), яких повинно бути не менше двох, а загальна кількість уроків, які входять в модулі, має бути не менше десяти. В про-версії можна встановити дати м'якого і жорсткого останнього терміну здачі, а також кінця модуля, змінити його на екзаменаційний (тривалість іспиту розраховується з моменту запуску іспиту учнем), налаштувати нарахування балів в залежності від кількості спроб, рішень і дедлайнів.

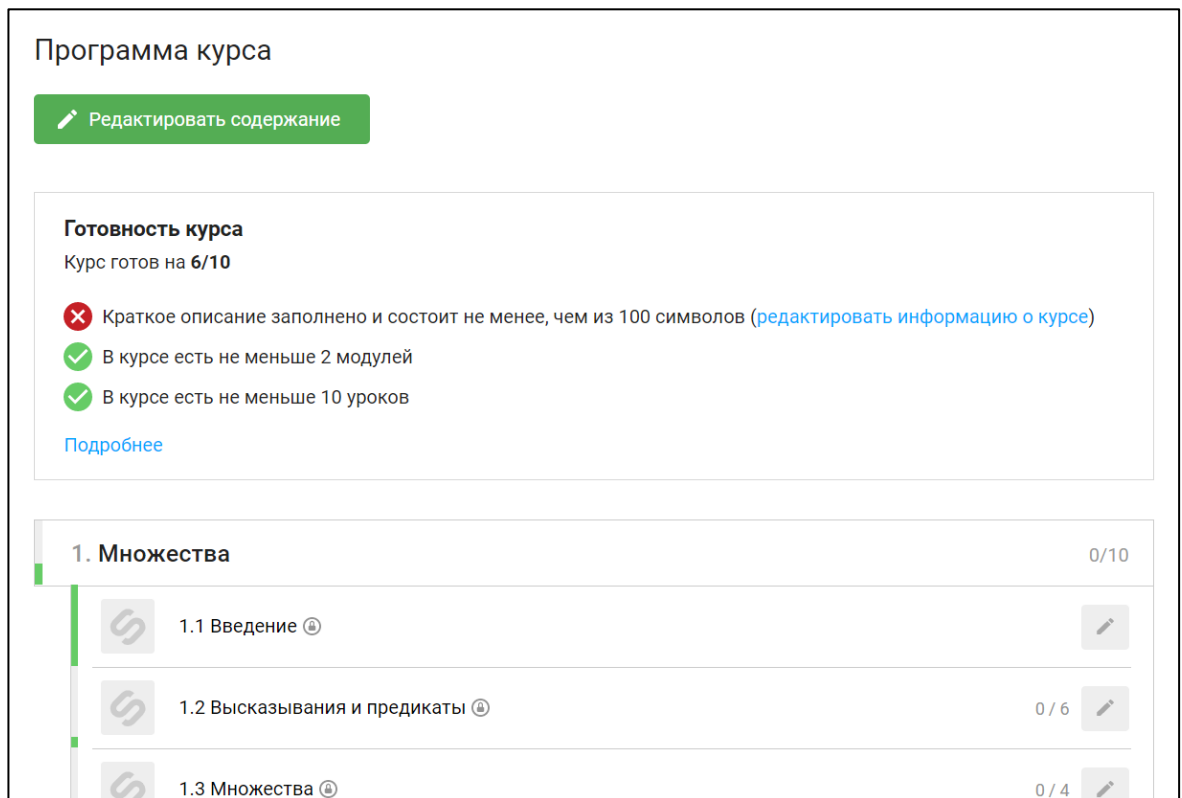


Рисунок 2.1 – Интерфейс програми курсу на Stepik

Урок складається з кроків, яких може бути до 16. Уроки можуть існувати і поза курсами, а також можуть входити в кілька курсів відразу. В курс можна додавати власні нові і вже існуючі уроки. Для створення уроку потрібно ввести тільки його назву. В налаштуваннях до уроку можна завантажити логотип, зробити його публічним або приватним, відключити коментарі. Один урок може складатися з необмеженого кількості «кроків». У режимі редагування можна додавати або видаляти кроки, завантажувати і завантажувати викачані кроки, видалити урок.

«Кроки» (рис. 2.2) діляться на теоретичні (текст, відео) і практичні (різного виду завдання), за які можна виставляти максимально можливу кількість балів. Порядок кроків можна змінювати в рамках одного курсу, або копіювати та вставляти в інших.



Выберите тип шага

Текст Текст с форматированием, изображениями, формулами	Тест Выберите один или несколько вариантов из списка
Численная задача Введите численный ответ	Видео Загружайте видео
Программирование Напишите программу. Тестируется через stdin → stdout	

[+ еще 20 типов шага](#)

Рисунок 2.2 – Типы «кроков» на Stepik

Текст може містити зображення, математичні формули, таблиці. В налаштуваннях до уроків можна змінювати шрифт, колір і розмір тексту, вставляти посилання, прикріплювати додаткові матеріали. Редактор формул складний для об'ємних математичних виразів і спеціальних символів (рис. 2.3) і для звичайного користувача може знадобитися досить багато часу, щоб розібратися в документації. Опублікувати відео можна тільки окремим «кроком», без отікання текстом.

Math
✕

Write TeX here

$\sin 2x = 2 \sin x \cdot \cos x;$

Inline
 Display mode

[TeX documentation](#)

$$\sin 2x = 2 \sin x \cdot \cos x;$$

OK
Cancel

Рисунок 2.3 – Редактор формул на Stepik

В рамках створення математичного курсу необхідно розглянути відповідні види завдань на платформі. Це такі, як:

- тест – вибір одного або декількох варіантів відповіді. Доступні функції: будь-яка відповідь правильна, переміщення відповідей, відображення обмеженої кількості відповідей (наприклад, 5 з можливих 10);
- задача сортування – складається список, який необхідно відсортувати в певному порядку;
- задача на зіставлення – створення двох списків, які необхідно зіставити між собою;
- табличне завдання – позначення правильних осередків в таблиці;
- числова завдання – введення числа, яке буде перевірятися на правильність відповіді і з урахуванням зазначеної викладачем помилки;
- математична задача – введення математичної формули, яка перевіряється на еквівалентність правильної формулою.

Хоча можливість відображення формул наявна в текстах і умовах до завдань, в самих тестах (варіантах відповідей) такої можливості немає (рис. 2.4), що ускладнює сприйняття. Крім того, для створення завдання, для якого призначено кілька питань, необхідно дублювати умову в кожному «кроці» (не можна додати кілька питань з відповідями до однієї задачі).

Вы можете решить

Выберите все подходящие ответы из списка

- $(x+1)^2=x^2+2x+1$;
- $\operatorname{tg} x \cdot \operatorname{ctg} x < 1$;
- при некотором x точка на прямой с уравнением $y=2x+3$ лежит в четвертой четверти;
- Неравенство $\sin(x < a)$ относительно переменной x имеет решение при каждом значении параметра a ;
- $x^2=1$;
- функция $f(x)=(x^2-1)/(x-1)$ имеет два корня;
- натуральное число n является четным;
- в равнобедренном треугольнике углы при основании равны;

1 балл за решение.

Отправить

Рисунок 2.4 – Відображення формул в тестах

В про-версії з кожного курсу і уроку викладач може завантажити звіти з даними про користувачів і їх прогрес. Серед них не тільки оцінки і загальний результат кожного слухача, а також: докладна статистика по кількості спроб, коректності, часу відповіді і статусу в розрізі уроків; активність учнів в модулях; плагіат текстових робіт; ймовірність шахрайства на тесті; середній час проходження кроків, уроків, модулів і курсу; оцінка складності завдань курсу; інформація про джерела, які призводять відвідувачів на сторінку курсу.

Публікація курсу у відкритий доступ дозволяється тільки, якщо дотримано умову про кількість модулів і їх уроків, що запобігає можливість запустити курс на початкових етапах і розвивати поступово по ходу проходження. На Stepik можна створювати приватні (закриті) курси. Приватний курс можна знайти в каталозі, пошуку і відкрити по звичайній прямим посиланням, не будучи запрошеним в курс. Функція публікації на запрошення доступна тільки в про-версії. Якщо необхідно стежити за прогресом певних студентів, на основі курсу можна створити клас, доступ до якого здійснюється за посиланням, але сам курс проходити і коментувати зможе будь-який користувач.

Для того, щоб коректніше визначити корисні і негативні сторони платформи був також прийнятий до уваги досвід викладача ОНУ ім. Мечникова Антоненко А. С. з його курсом по програмуванню на мовах C і C ++ [11].

Існуючий на платформі табель успішності дозволяє викладачеві подивитися кількість вирішених завдань і зароблені бали в загальному і по кожному з них. Також для кожного завдання можна побачити історію відповідей учня (рис. 2.5) і дізнатися в чому полягала його помилка, якщо завдання було вирішено не з першого разу. Ця функція сайту є досить корисною, але коментар з боку викладача не передбачає, для цього необхідно відписуватися в коментарях до «кроку», який кожен може переглядати і відповідати на нього.

Решения задач урока Первая программа, переменные, считывание

Класс «Программирование КИ 1 курс, Основы C/C++: Основы C/C++ для спортивного программирования»

id:304446285 Шаг 7: Программирование

Пользователь	Решения	Дата отправки	Оценка
КК Kseniia Kishubaieva	● #308091923	2 месяца назад	2
КК Kseniia Kishubaieva	● #308091541	2 месяца назад	0
КК Kseniia Kishubaieva	● #308089839	2 месяца назад	0
КК Kseniia Kishubaieva	● #308089482	2 месяца назад	0
КК Kseniia Kishubaieva	● #308088996	2 месяца назад	0

Рисунок 2.5 – Історія відповідей

Створені завдання генеруються з прикладом, для наочності, який можна налаштувати під себе або повністю стерти і написати заново. Так само до різних видів завдань є документація.

Для проведення контролю відсутня генерація варіантів завдань для кожного учня, що ускладнює викладачеві роботу з платформою. Всі розміщені завдання доступні кожному в одному і тому ж вигляді.

Розглянуто корисні функції для створення курсу на цій платформі:

- зручний редактор тексту (можливість створювати списки, додавати формули, зображення, таблиці), особливо, якщо матеріал пишеться з нуля;
- різноманітність, так званих, «кроків» (це може бути текст, відео, різні види завдань);
- можливість працювати з вкладеними файлами («про» версія);
- інтерфейс додавання модулів і «кроків» курсу зручний і зрозумілий;
- можливість копіювання, перестановки, додавання вже існуючого «кроку»;
- перегляд історії рішень учня.

Але є деякі проблеми, з якими можна зіткнутися при створенні курсу:

- труднощі копіювання з word (або інших текстових редакторів), доводиться заново редагувати матеріал, на що витрачається багато часу;
- утруднення комунікації викладача з учнем;
- відсутність генерації екзаменаційного модуля з існуючих завдань;
- неможливість проведення вхідного контролю і проходження курсу з урахуванням наявних знань;
- редактор формул складний для звичайного користувача, також відсутня можливість відображення формул в тестах;
- нестача шаблонів математичних задач;
- відсутня можливість відтворення графічних елементів, графіків.

3 ВИМОГИ ДО ПРОЕКТУВАННЯ І РОЗРОБКИ ІНТЕРФЕЙСУ

3.1. Основні вимоги до призначеного для користувача інтерфейсу

Успіху освітньої платформи або програми також сприяють безліч факторів, які включають в себе і ступінь, в якій вони розважають, інформують і викладають відповідну інформацію передбачуваному користувачеві. На додаток до цього, одним з найважливіших факторів, що визначають ефективність застосування програми або платформи, є часто не береться до уваги фактор того, наскільки добре вони спроектовані.

Дизайн додатка або платформи пов'язаний не тільки з його красою, а й з тим, наскільки він функціональний, зручний і приємний для користування. Дизайн диктує почуття, платформа може дати своїм користувачам, і вони особливо не вирішуються використовувати технології, які можуть здатися їм незручними або непривабливими. В основному саме дизайн і ергономічність є вирішальними факторами, що дозволяють користувачеві визначитися в бажанні скористатися платформою або програмою, так як платформи і програми, які не мають ніякої комфортної основи для користувача відбивають будь-яке бажання користуватися ними хоч зрідка, не кажучи вже про використання на постійній основі. Таким чином, можна прийти до висновку про те, що ергономіка і дизайн програми або платформи є мало не головними критеріями для користувача, дозволяють йому з легкістю і естетичним задоволенням проводити час на цій платформі.

Зростаюча складність комп'ютерних систем і, відповідно, програмного забезпечення, що працює на них, не залишає іншого вибору, крім як перемістити інтерфейс між користувачем і комп'ютером в центр розгляду системи, програмного забезпечення та дизайну продукту поряд з постійними оцінками і поліпшеннями [12].

Однак поточна ситуація показує, що ергономічний потенціал системи, програмного забезпечення або дизайну продукту реалізується недостатньо або

навіть використовується непродуктивним чином. Крім того, вимоги до зручності використання часто не виправдовуються: занадто часто користувачам доводиться підкорятися, щоб компенсувати недоліки призначеного для користувача інтерфейсу.

Специфікація зручності використання повинна бути заснована на прийнятому інтерфейсі і принципах проектування взаємодії з користувачем, щоб створювати оптимальне програмне забезпечення або, в більш загальному плані, оптимальні продукти для передбачуваної групи користувачів. Дотримання таких принципів проектування під час написання специфікації також дозволяє в майбутньому оцінювати інтерфейс і взаємодія людини і платформи на основі тих же принципів.

Специфікація є змінною, якщо її структура допускає необхідні модифікації її вимог. Крім того, ці зміни повинні бути легко здійсненні, не суперечачи вимогам повноти і послідовності. Надмірності (включаючи повторення одного і того ж вимоги в документі) слід в значній мірі уникати, оскільки це може легко призвести до помилок, особливо якщо документ переглядається кілька разів.

Взаємодія між користувачем і додатком (робочим інструментом) стає більш складним, якщо користувач не розуміє взаємозв'язку між функціональністю системи, з одного боку, і вимогами його робочого контексту, з одного боку. Отже, прозорість і гнучкість інтерфейсу повинні завжди бути на передньому плані під час розробки програми. Щоб думати і бачити наскрізь систему, користувачеві може знадобитися функція допомоги.

Важливо знайти міру між автоматизацією роботи і повним контролем з боку користувача. Вимога керованості відноситься до принципової можливості впливу з боку користувача.

Користувач повинен мати можливість без особливих зусиль пристосувати її до своїх індивідуальних потреб і вимог. Однак широкий спектр дизайнерських можливостей не може замінити точний ергономічний дизайн.

Отже, обсяг свободи, одержуваної користувачем, повинен бути ретельно збалансований, щоб виключити проблеми, що виникають.

Користувач повинен без особливих зусиль звикнути до додатка. Щоб спростити діалог між користувачем і системою, його можуть підтримувати системи, які включають відповідні стратегії навчання (наприклад, за допомогою прикладів), пропонують прозорі системи і надають допомогу користувачу (наприклад, функції допомоги).

Популяція користувачів може включати кілька рідних мов і рівнів читання. Таким чином, в системі може знадобитися правильна граматика, проста термінологія, діаграми і зображення. Однак термінологія також повинна відображати завдання користувачів, тому в аналізі слід записувати найбільш природні терміни, які слід використовувати, будь то медичні, інженерні, юридичні, фінансові та т. д. Системна мова може також потребувати включення скорочень і кодів, які знайомі користувачеві в його повсякденних завданнях.

3.2. Вимоги до проектування призначеного для користувача інтерфейсу освітньої платформи

Необхідність в розробці освітньої математичної платформи зростає через властивих недоліків вже існуючих платформ, з якими стикається освітнє програмне забезпечення.

Основна вимога – створити не звичайну платформу з розміщенням курсів, а платформу, яка націлена на навчання математикою людей, життя яких не пов'язана безпосередньо з математичними спеціальностями. Підвищити якість засвоєння курсу може адаптивність системи [13], тобто підбір тестів під кожного студента, в залежності від певного рівня учня. Для цього необхідно підібрати зручний метод відображення і надання інформації, а також перевіркою частини.

Крім того, система повинна забезпечити викладача комфортним складанням лекцій і управлінням групами учнів. Відображення рекомендацій по організації навчального курсу [14] і оцінка якості навчальних матеріалів є одними з таких вимог.

Система повинна бути спроектована таким чином, щоб всеохоплююче почуття потреби і мотивації постійно зростала, а почуття примусу зменшувалася. Розглянемо наступні пропозиції щодо підвищення мотивації в освітній платформі.

Голосовий інтерфейс може допомогти подолати навіть дуже стресові ситуації, він дозволяє людям бути більш розслабленими і творчими. Більш того, збіг статі, характеру або акценту викликає почуття довіри та інтересу. Певний тип голосу може вплинути на те, наскільки студентам подобається говорити і наскільки старанно вони намагаються зрозуміти представлений матеріал.

Мова і текст повинні бути представлені в неформальному стилі, щоб заспокоїти учня. Проте, до методу слід підходити обережно, щоб не залишити в учнів враження, що контекст не має значення. Крім того, використання важливих слів, а не владним фраз, покращує навчання.

Анімовані педагогічні агенти – це фігури, які з'являються в навчальних матеріалах і пропонують підказки. Цими фігурами можуть бути персонажі з мультфільмів, тварини або певні віртуальні зображення. Для їх формування можуть бути використані тексти, промови, ілюстрації і анімація.

Кольорове розмаїття викликає в учнів мотивацію і увагу, а також робить навчальні матеріали більш реалістичними. Кольори також можна використовувати для угруповання пов'язаної інформації. Однак слід зазначити, що, якщо задіяно більше 10 кольорів, спотворення співвідношення між інформацією буде неминучим. Правильне використання кольорів також прискорює перегляд і спрощує його. Крім того, на екрані інтерфейсу слід використовувати достатню кількість вільного місця. Ці простори можуть упорядкувати сторінки, щоб не захаращувати інформацію на екрані. Останнє,

що потрібно знати про кольори, – це використовувати темний текст на світлому тлі, що прискорює читання.

Стиль викладання в електронних курсах повинні допомагати учням математикам, використовуючи слова і зображення в курсі, щоб вони поєднувалися зі знаннями в довготривалій пам'яті. Тексти можна фрагментувати і розміщувати в різних частинах зображення. Коли текст описує операцію або режим на зображенні, його можна використовувати як спливаюче повідомлення.

Платформа повинна бути доступною, що означає, що користувачі можуть легко отримати доступ до передбачуваного контенту. Через те, що учні регулярно звертаються до раніше вивчених матеріалів або шукають конкретний зміст, система електронного навчання повинна надавати учням можливість пошуку за змістом і допомагати їм легко досягати бажаного змісту. Більш того, кожного разу, коли слова або фрази, які використовуються в тексті, існують десь ще, вони повинні діяти як посилання для навігації, опису і простого повернення користувача на попередню сторінку.

Електронне навчання – це не просто перетворення традиційних навчальних матеріалів в електронні. В електронному навчанні призначений для користувача інтерфейс грає ключову роль в досягненні освітніх цілей, тим більше, якщо мова йде про викладання математики. Психологічні проблеми в навчанні – це нові відкриття в електронному навчанні, які впливають на дизайн користувальницького інтерфейсу. Як уже згадувалося раніше, для забезпечення сприйняття матеріалу необхідно використовувати мовний інтерфейс, неформальний стиль, агенти і кольори. Робоча пам'ять також відіграє вирішальну роль в навчанні, і вона не повинна бути насичена непотрібною інформацією. При використанні мультимедійної інформації краще використовувати канали слуху і зору, якщо це можливо. Інтерфейс повинен розглядатися не тільки як художній феномен, а й художні інструменти, такі як графіка, музика, анімація і т. д., які повинні бути сумісні з освітніми психологічними проблемами.

Щоб призначений для користувача інтерфейс математичної платформи був ефективним, він повинен відповідати наступним критеріям:

- студентам та викладачам повинно бути легко орієнтуватися;
- інтерфейс повинен викликати позитивні емоції і сприяти навчанню математики.
- він повинен бути орієнтованим на передбачувану групу користувачів, серед яких не математики.
- інтерфейс повинен надавати викладачеві можливість комфортного складання лекцій і тестів.
- інтерфейс повинен підтримувати запис і відображення математичних формул і графіків, а також мати засоби побудови тривимірних об'єктів.

3.3. Сценарії використання інтерфейсу платформи

Для того, щоб знайти ефективне рішення і забезпечити хороший досвід взаємодії користувача з системою і досягти певної мети, можливості користувачів системи описані у вигляді Use Case діаграма. Основний сценарій у вигляді Use Case діаграми можна побачити на рисунку 3.1.

За допомогою Use Case можуть бути описані як призначені для користувача вимоги, так і вимоги до взаємодії систем. Описані сценарії взаємодії лягли в основу розробки бази даних системи.

Не авторизований користувач в системі – гість – має можливість пройти пробний модуль або ж урок курсу, а також деякий набір тестів для власної оцінки курсу і поняття зацікавленості.

Авторизований користувач в системі може виступати як студентом, так і викладачем, що пізніше розмежовується в системі.

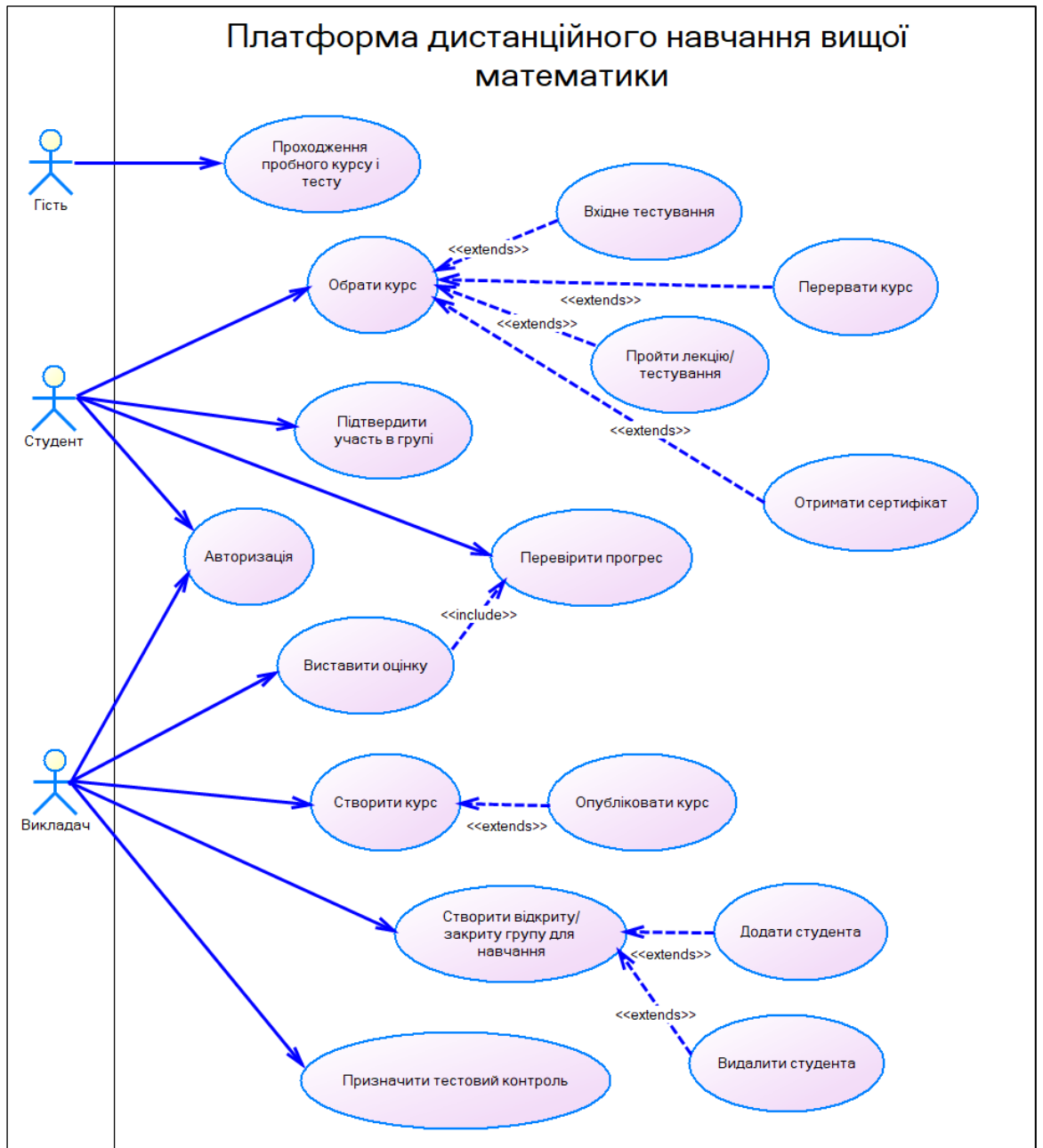


Рисунок 3.1 – Use Case діаграма

Система передбачає виконання основних дій і наявність стандартного набору можливостей освітньої платформи. Серед них, для викладача – створення (табл. 3.1) і публікація курсу, управління студентами в групі, оцінювання та призначення тестових контролів.

Таблиця 3.1 – Сценарій створення курсу

USE CASE	Система створення курсу на платформі	
Основна діюча особа	Викладач	
Ціль	Створити курс	
Тригер	Викладач вирішує створити курс на платформі	
Результат	Курс створений	
Опис	Крок	Дія
	1	Викладач дає курсу назву і опис
	2	Викладач створює модуль курсу і дає йому назву
	3	Викладач створює урок і наповнює його текстом або відео
	4	Викладач створює тести для контрольного або модульного тестування
	5	Система пропонує викладачеві рекомендації по доданим матеріалами

Головна відмінність – можливість запускати курс для певної групи, для якої був відкритий набір для проходження матеріалу під керівництвом викладача (наприклад, якщо це група студентів в університеті). Крім того, є можливість дозволити користувачам проходити курс незалежно від групи (проте, при цьому зменшується підтримка з боку викладача).

Викладач зможе виставляти час, необхідний для проходження модульного контролю, в такому випадку завдання виконані після його закінчення не будуть зараховуватися. Якщо обмеження по часу немає, платформа все одно фіксує його для учня, з можливістю його приховати.

Вхідне тестування (табл. 3.2) використовується для оцінки здібностей учня. Це зручний спосіб, за допомогою якого учні можуть самостійно проаналізувати наявні навички в рамках курсу і набрати певні бали за окремі модулі курсу і відсоток освоєного курсу. Також це дозволяє адаптувати курс особисто для кожного студента.

Таблиця 3.2 – Сценарій проходження вхідного тестування

USE CASE	Система проходження вхідного тестування	
Основна діюча особа	Студент	
Ціль	Пройти вхідне тестування	
Тригер	Студент вирішує пройти вхідне тестування для адаптивності курсу	
Результат	Вхідне тестування пройдено і система підлаштовує курс під результати	
Опис	Крок	Дія
	1	Студент обирає вхідне тестування по курсу
	2	Система пропонує генеровані тести по кожному модулю
	3	Студент виконує тести і завершує тестування
	4	Система показує відсоток знань по курсу і кожного модуля
	5	Студент обирає проходження курсу з урахуванням набраних навичок
Розширення		Дія розгалуження
	5a	Студент обирає проходження курсу без урахування набраних навичок

Студент може проходити курс самостійно або в групі, при цьому його прогрес, включаючи проходження лекцій і тестів, буде відображатися у викладача. Також участь в групі передбачає спілкування з викладачем і проходження призначених екзаменаційних випробувань.

Крім модульних тестів і завдань курсу, змагання по засвоєнню знань можуть надати учням спосіб як підвищити, так і знизити рівень своїх професійних навичок. По закінченню курсу студент зможе отримати сертифікат з урахуванням складності пройдених їм завдань.

Таблиці з усіма основними сценаріями використання наведені в Додатку А.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРФЕЙСУ

4.1. Засоби реалізації інтерфейсу користувача

Для написання призначеного для користувача інтерфейсу буде використовуватися фреймворк Vue.js [15]. Це інфраструктура призначеного для користувача інтерфейсу на основі компонентів. Він включає в себе більшість основних функцій, які також є в інших фреймворків, і в даний час частіше використовується в стартапах, він легкий і його набагато легше зрозуміти. Оскільки Vue працює тільки на «рівні уявлення» (і не використовується для проміжного програмного забезпечення і серверної частини), він може легко інтегруватися з іншими проектами і бібліотеками.

Висока доступність Vue допомагає швидко задіяти всі свої творчі сили, не витрачаючи багато часу на ознайомлення з додатковими розширеннями синтаксису. Vue також скорочує час, необхідний для розуміння цілому, наприклад, Angular, що особливо зручно при написанні простих веб-додатків.

Модульність бібліотек з використанням фреймворку часто явище у фронтенд-розробці. І React, і Angular мають модуляризацію. Але те, що відрізняє Vue від інших альтернатив, – це те, як легко розширювати функціональні можливості, і наскільки добре всі частини працюють після включення додаткових модулів. Наприклад, якщо необхідно організувати і відобразити невеликі візуальні компоненти, все, що потрібно, – це «основна» бібліотека Vue, немає необхідності включати додаткові бібліотеки. У міру зростання додатку з'являються бібліотеки для керування маршрутами, такі як «vue-router», бібліотеки для управління глобальним станом, такі як «vuex», або бібліотеки для створення адаптивних веб-додатків, таких як «bootstrap-vue». Крім того, якщо додаток вимагає оптимізації, можна включити бібліотеку vue-server-rendering. На рисунку 4.1 видно, як бібліотеки, які були згадані, поступово включаються, від невеликого односторінкового додатку (англ.

Single page application, SPA) до багатосторінкового додатку (англ. Multi page application, MPA).

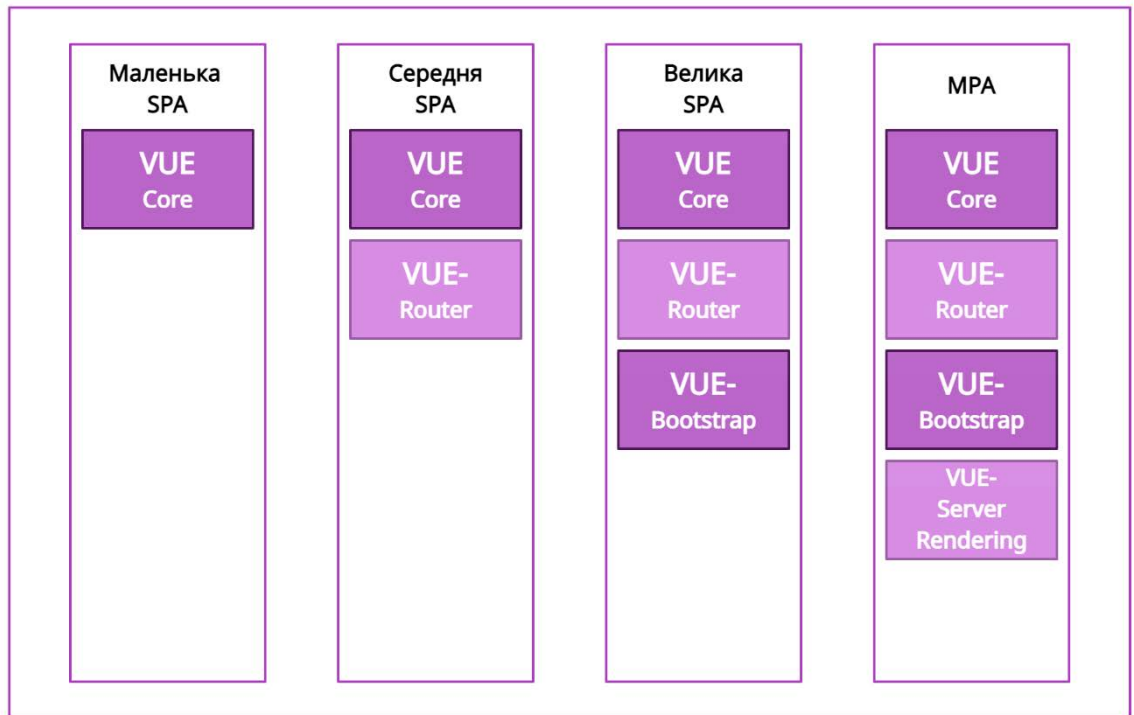


Рисунок 4.1 – Модульність бібліотек Vue.js

Компонентна система Vue є реактивною, що означає, що Vue знає, як взаємодіяти з допомогою асинхронних подій. Директиви призначені тільки для інкапсуляції низькорівневих маніпуляцій з DOM, в той час як компоненти являють собою повноцінні автономні об'єкти, зі своєю власною логікою даних та подання.

З Vue немає проблем з іншими бібліотеками або ресурсами, можна використовувати інструмент, з яким найзручніше. Наприклад, можна писати тільки HTML і JavaScript або, якщо необхідно, додати CSS, JSX або TypeScript.

Vue має спеціальну командний рядок (англ. Command line interface, CLI), створену на Node JS. Цей інструмент дозволяє почати проект, використовуючи шаблон (або базовий шаблон). Це гарантує, що різні інструменти збірки будуть працювати разом з оптимальними настройками за замовчуванням, що

дозволить зосередитися на створенні програми, а не витратити годинник на конфігурацію. У той же час, він як і раніше надає гнучке налаштування конфігурації кожного інструменту у відповідності з конкретними потребами.

У Vue також є власні DevTools, які представлені у вигляді розширення для браузера. DevTools спрощує налагодження додатків і перевірку стану ієрархії компонентів. Вони дозволяють редагувати додаток в реальному часі, відстежувати призначені для користувача події та налагоджувати додаток у часі, щоб побачити попередні версії і зроблені зміни.

Vue.js має абсолютно відмінну офіційну документацію, яка дозволяє розробникам розкривати практично всі аспекти Vue і його екосистеми. В офіційній документації є приклади і докладні описи.

4.2. Взаємодія модулю інтерфейсу з сервісами платформи

В ідеалі UI – це логіка уявлення без будь-якої бізнес-логіки. Якщо розглядати користувальницький інтерфейс як інструмент для даних, то надання єдиного мікросервіса, який діє як інтерфейс для проекту, є хорошим підходом до архітектури мікросервісів. Місце модулю інтерфейсу в загальній архітектурі системи можна побачити на рисунку 4.2.

При роботі з SPA дуже поширене отримання даних з віддалених сервісів. Це дозволяє відображати динамічний контент в залежності від певних застосовуваних критеріїв. Отримання зазначених даних може бути виконано різними способами.

API, або інтерфейс прикладного програмування є програмним посередником, який дозволяє двом додаткам взаємодіяти один з одним. API надає дані, які можна використовувати в системі, не турбуючись про бази даних або відмінності у мовах програмування. Дані витягують з інтерфейсу API, який повертає їх в форматі JSON, потім ці дані можна інтегрувати в інтерфейсні програми. Vue.js відмінно підходить для використання таких API.

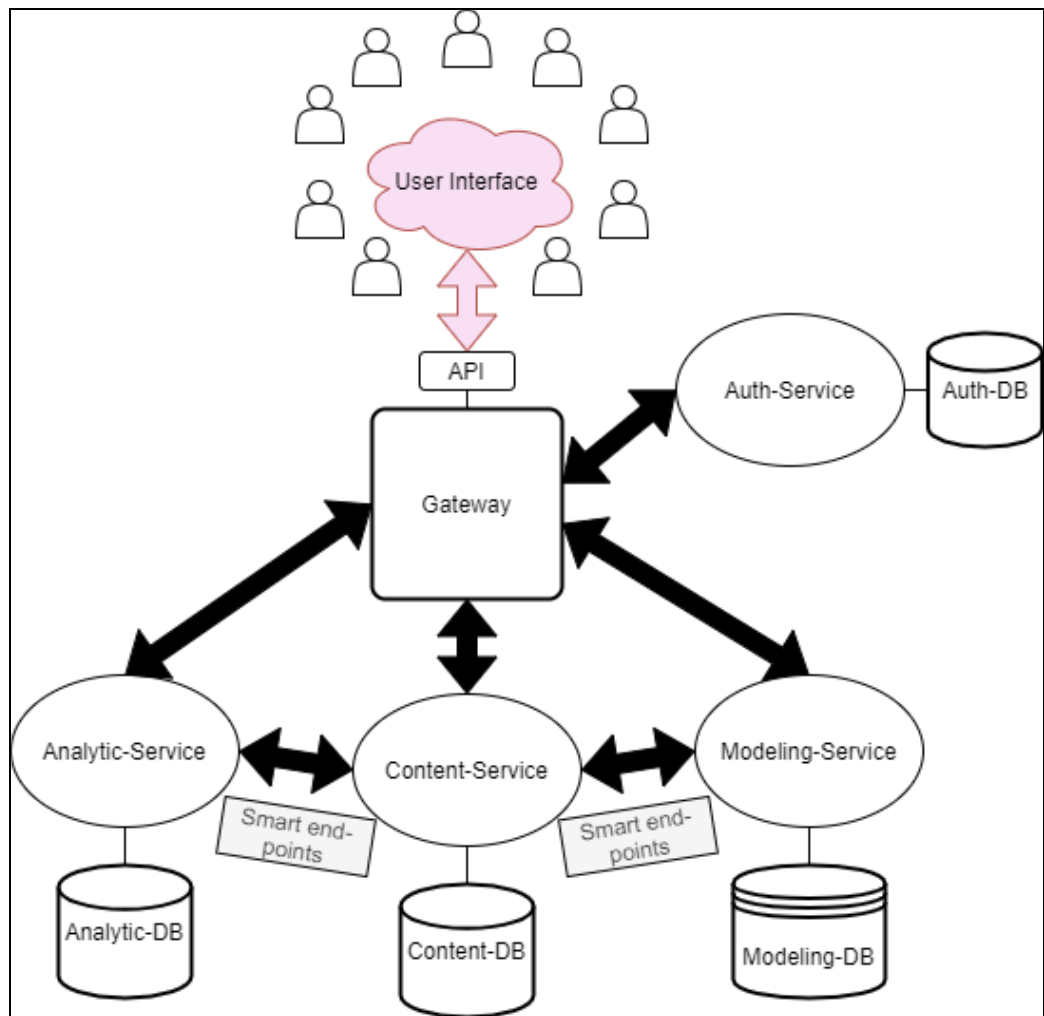


Рисунок 4.2 – Місце модулю інтерфейсу в загальній архітектурі системи

Запити Ajax до REST API – це найбільш поширений спосіб асинхронної вибірки даних з віддалених служб. Запити відправляються на сервер, де обробляються дані. Тим часом, в залежності від реалізації, може бути або підхід, заснований на зворотному виклику, або більш новий підхід, заснований на Promise, де необхідно чекати, поки процес або завершиться успішно, або не видасть помилку.

Axios – відома бібліотека для виконання асинхронних запитів. Він має дуже гарний і простий у використанні API, підтримує обіцянки, більш просунуті функції, такі як перехоплювачі, створення примірників Axios, додавання зовнішніх плагінів і багато іншого.

У написаному клієнті або службі створюється екземпляр Axios і використовується замість того, щоб просто імпортувати Axios всюди. Це не тільки спрощує тестування вашої програми за рахунок використання імітацій або заглушок, але також дозволяє централізувати налаштування HTTP (наприклад, базовий URL-адресу та порт). Крім того, він відокремлює компоненти від внутрішніх функцій HTTP-операцій і REST API.

4.3. Побудова шаблонів

Побудувати і описати шаблони, більш докладно підключення, решта привести як приклад того, що повинно бути.

Для побудови шаблонів призначеного для користувача інтерфейсу був використаний онлайн-сервіс для розробки інтерфейсів і прототипування Figma [16]. Він простий у використанні і має всі інструменти, необхідні для роботи.

Для роботи з сервісом не потрібно завантажувати, встановлювати та постійно оновлювати додаткове забезпечення; немає необхідності зберігати і систематизувати файли, робота автоматично зберігається в загальному просторі в хмарі.

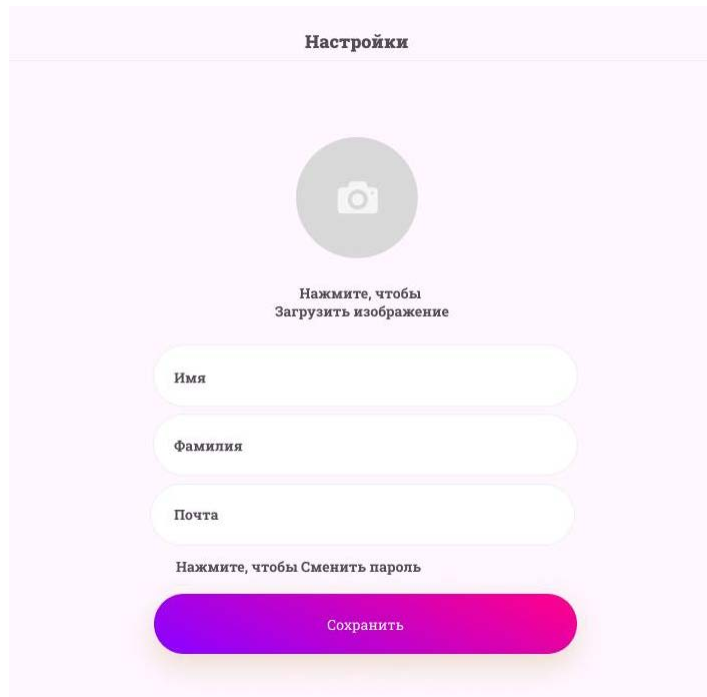
Перехід від дизайну до коду швидкий і послідовний. За допомогою Figma легше структурувати наші проекти таким чином, щоб вони відображали те, як вони будуть кодуватися. Коли розробник перевіряє дизайн, він може побачити елементи призначеного для користувача інтерфейсу, вкладені в відповідні контейнери, а це значить, що буде більш точний план, на який потрібно посилатися при написанні коду.

Спроектовано сторінки: домашня сторінка, сторінка входу, реєстрації, огляд курсу, завантаження файлів, вхідне тестування.

Домашня сторінка має шапку та нижній колонтитул для додаткової інформації та контактних даних. Шапка має логотип платформи, що містить її назву «Математика для не математиків», кнопки переходу на головну сторінку

та сторінку з інформацією про проект. Також на шапці розміщені кнопки входу до облікового запису та реєстрації на платформі. Основна частина сторінки – це привітання та запрошення пройти курс на платформі; кнопки для переходу на сторінку початку курсу і вхідного тестування, які відображаються, якщо користувач виконав вхід в систему, в іншому випадку на сторінку входу.

Сторінки входу в систему та реєстрації мають схожий дизайн і містять поля для заповнення та кнопку підтвердження. Виконав вхід в систему, користувач має можливість змінити дані профілю у налаштуваннях (рис. 4.3), серед яких іконка профілю, ім'я та прізвище. При цьому меню можливостей з'являється поряд з іконкою.



The image shows a mobile application interface for profile settings. At the top, the title 'Настройки' (Settings) is displayed. Below it is a circular profile picture placeholder with a camera icon and the text 'Нажмите, чтобы Загрузить изображение' (Click to upload image). Underneath are three input fields labeled 'Имя' (Name), 'Фамилия' (Surname), and 'Почта' (Email). Below these fields is a link 'Нажмите, чтобы Сменить пароль' (Click to change password). At the bottom is a large, rounded, pink button labeled 'Сохранить' (Save).

Рисунок 4.3 – Налаштування профілю

Головний контекст сторінки вхідного тестування має набір задач та тестів для проходження. Приклад інтерфейсу вхідного тестування можна розглянути на рис. 4.4.

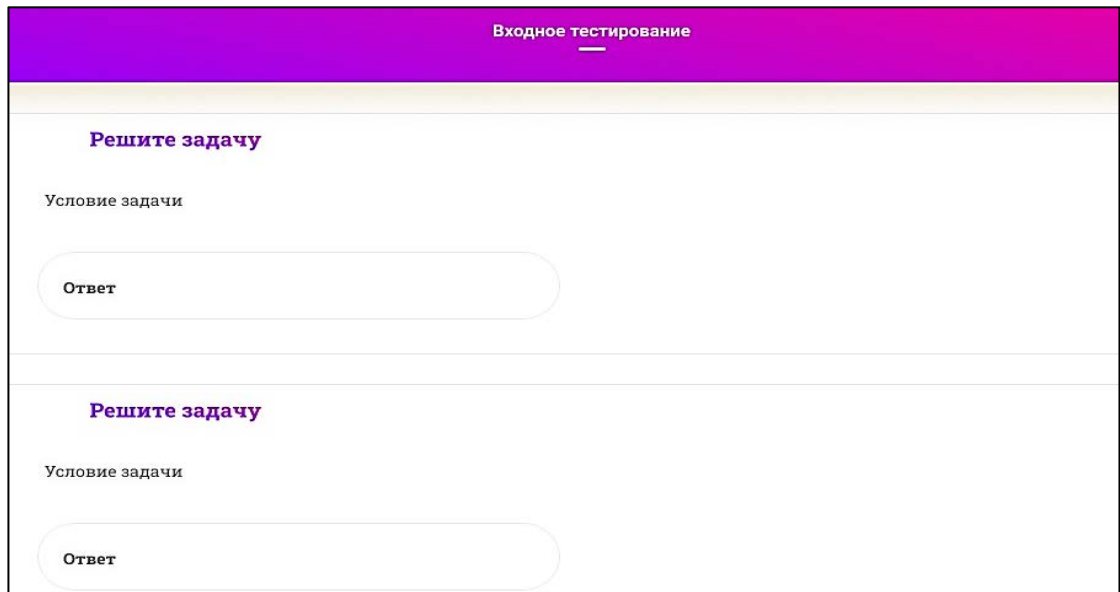


Рисунок 4.4 – Вхідне тестування

Головна сторінка курсу має загальний огляд курсу (рис. 4.5), який необхідно пройти. Він містить інформацію про викладача, загальний опис курсу і меню перемикавання модулів і уроків.

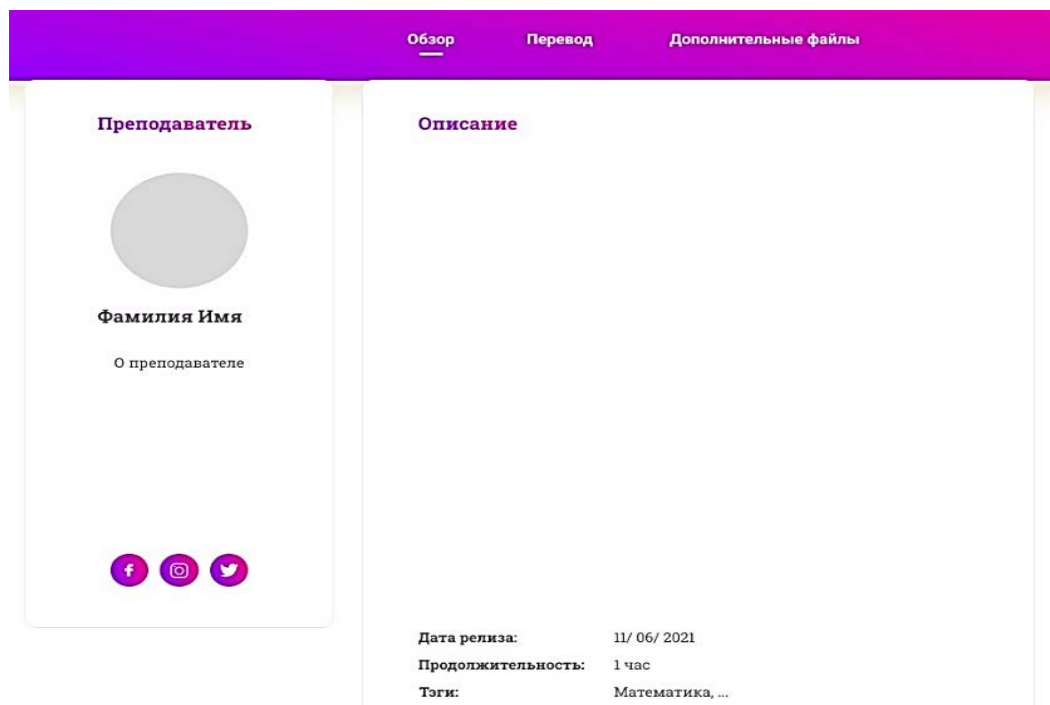


Рисунок 4.5 – Сторінка огляду курсу

Всі спроектовані шаблони наведено у Додатку Б.

4.4. Реалізація інтерфейсу

На основі розроблених шаблонів, був розроблений можливий варіант деяких сторінок прототипу математичної платформи.

Проект створений за допомогою CLI, тому має певну структуру. Створені файли і папки в проекті Vue:

- `public` – папка містить загальнодоступні файли, такі як `html` і `favicon.ico`. Розміщення тут статичні ресурси будуть скопійовані і не упаковані в веб-пакет.

- `src` – папка містить вихідні файли проекту;
- `src / assets` – папка містить ресурси проекту, такі як `png`;
- `src / components` – папка містить компоненти Vue;
- `src / App.vue` – основний компонент Vue-проекту;
- `src / main.js` – основний файл проекту, який завантажує Vue-додаток;
- `config.js` – файл конфігурації;
- `json` – файл містить список залежностей проекту;
- `node_modules` – папка містить встановлені пакети npm.

У проект додана бібліотека `Vue-Router`, це офіційна бібліотека, яка глибоко інтегрується з `Vue.js` і дозволяє легко створювати багатосторінкові програми. Включає в себе можливості вкладених маршрутів, їх модульної конфігурації і контролю навігації.

Також було додано `Vuex` – патерн управління станом і бібліотека. Він служить централізованим сховищем даних для всіх компонентів програми з правилами, що гарантують, що стан може бути змінено тільки передбачуваним чином.

Неодноразово при створенні веб-додатки може знадобитися одержувати і відображати дані з API (програмний інтерфейс програми). Для цього використовується `axios` – підхід до побудови призначених для користувача

інтерфейсів веб-додатків, що полягає в «фоновому» обміні даними браузера з веб-сервером.

Для створення інтерфейсу програми були створені певні компоненти і сторінки Vue. Компонент – це модульний і багаторазово використовуваний блок коду. Він містить всі HTML, JavaScript і CSS, які потрібні для його функціональності. Тепер, коли необхідно вставити де-небудь цей «компонент», все, що потрібно зробити, це імпортувати його на сторінку.

Додаток має свого роду деревоподібну структуру. «Корневим» компонентом в даному випадку є `src/App.vue`, і цей компонент буде імпортувати інші компоненти.

В додатку використовуються такі компоненти як `Header` і `Footer`, використання яких можна побачити на домашній сторінці, сторінці реєстрації та входу в систему.

```

<Header />
<div class="hero-body">
  
  <div class="container">
    <h1 class="title"> обучение уже сейчас h1>
    <h2 class="subtitle"> специалистами в области высшей
математики </h2>
    <div class="button-block">
      <button class="button is-xl is-white is-rounded">
        <p>Начать обучение</p>
      </button>
      <button class="button is-xl is-white is-rounded">
        <router-link to="/diagnostic"><p>Входное
тестирование</p></router-link>
      </button>
    </div>
  </div>
</section>
<Footer />

```

Також компоненти імпортуються і додаються у вигляді компонент в скрипті сторінки. Оператор імпорту дозволяє імпортувати і використовувати інші компоненти всередині цього компонента. Об'єкт експорту дозволяє вам

визначати і експортувати цей іменований компонент для повторного використання в додатку. Об'єкт компонентів всередині експорту то, де можна перелічити всі дочірні компоненти, що використовуються в цьому компоненті.

```
import Header from '@/components/Header.vue';
import Footer from "@/components/Footer";
export default {
  name: 'home-view',
  components: {
    Header,
    Footer }
}
```

Таким чином на рисунку 4.6 можна побачити домашню сторінку платформи з вбудованими компонентами, один з яких шапка платформи з кнопками входу і реєстрації.

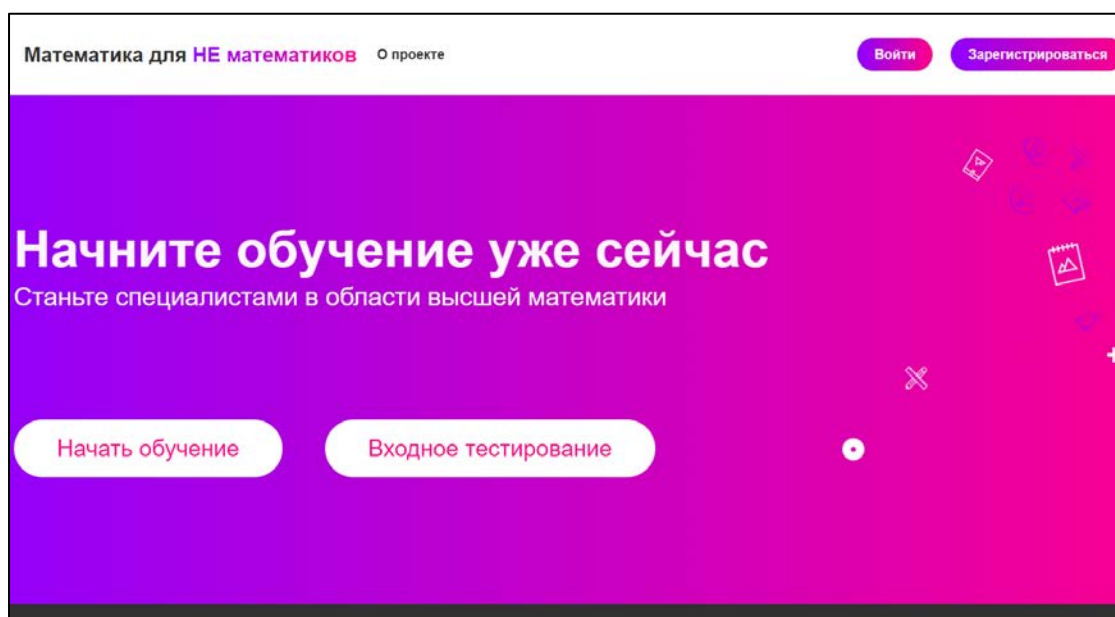


Рисунок 4.6 – Домашня сторінка

Для перевірки зв'язку між модулем призначеного для користувача інтерфейсу і системного модуля використовуємо аутентифікацію. В першу чергу виконана настройка компонентів входу (рис. 4.7) і реєстрації (рис. 4.8).

Математика для НЕ математиков О проекте

Войти Зарегистрироваться

Почта

Пароль

Войти

The screenshot shows a login page with a header containing the site name and a 'О проекте' link. On the right, there are two buttons: 'Войти' and 'Зарегистрироваться'. The main content area features two input fields for 'Почта' and 'Пароль', followed by a 'Войти' button.

Рисунок 4.7 – Сторінка входу

Математика для НЕ математиков О проекте

Войти Зарегистрироваться

Имя

Фамилия

Почта

Пароль

Зарегистрироваться

Математика для не математиков

The screenshot shows a registration page with a header containing the site name and a 'О проекте' link. On the right, there are two buttons: 'Войти' and 'Зарегистрироваться'. The main content area features four input fields for 'Имя', 'Фамилия', 'Почта', and 'Пароль', followed by a 'Зарегистрироваться' button. A footer bar at the bottom contains the text 'Математика для не математиков'.

Рисунок 4.8 – Сторінка реєстрації

В даний час веб-токен популярний для аутентифікації і обміну інформацією. Замість створення сеансу (аутентифікація на основі сеансу) сервер кодує дані в веб-токен і відправляє його клієнту. Клієнт зберігає JWT, потім кожен запит від клієнта до захищених маршрутами або ресурсів повинен бути приєднаний до цього токена (зазвичай в заголовку). Сервер перевірить цей токен і поверне відповідь.

У порівнянні з аутентифікацією на основі сеансу, при якій необхідно зберігати сеанс у файлі cookie, велика перевага токenu полягає в тому, що токен зберігається на стороні клієнта.

Токен складається з трьох важливих частин: заголовка, корисного навантаження і підписи. Разом вони об'єднані в стандартну структуру: `header.payload.signature`.

Клієнт зазвичай прикріплює токен в заголовку авторизації з префіксом `Bearer: Authorization: Bearer [header]. [Payload]. [Signature]`

Служба аутентифікації надає три важливих методи за допомогою axios для HTTP-запитів і відповідей:

- `login ()`: POST {адреса електронної пошти, пароль} і збереження JWT в локальне сховище
- `logout ()`: видалення JWT з локального сховища
- `register ()`: POST {ім'я користувача, прізвище користувача, адресу електронної пошти, пароль, роль}

Також є методи для отримання даних з сервера. У разі доступу до захищених ресурсів HTTP-запиту потрібно заголовок авторизації. Він перевіряє локальне сховище для користувача елемента. Якщо є авторизований користувач з токеном, повернути HTTP-заголовок авторизації. В іншому випадку повернути порожній об'єкт.

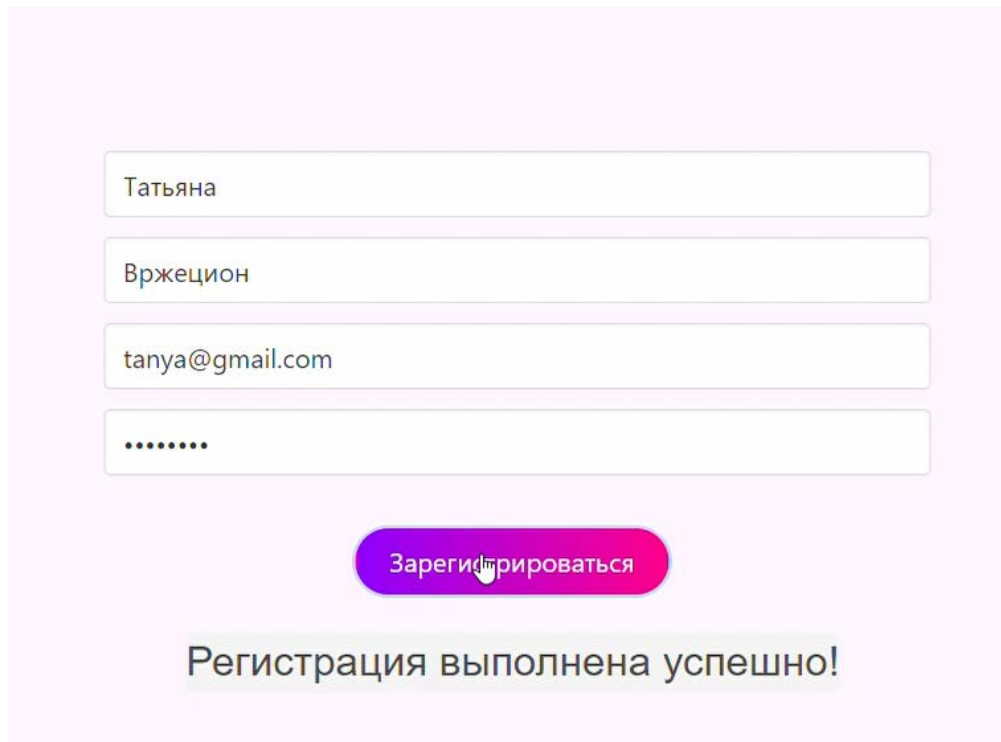
У центрі кожної програми Vuex знаходиться сховище – це в основному контейнер, в якому зберігається стан програми. Модуль аутентифікації Vuex містить:

- стан: {статус, користувач}
- дії: {вхід, вихід, реєстрація}
- мутації: {loginSuccess, loginFailure, logout, registerSuccess, registerFailure}

```
import Vue from 'vue';
import Vuex from 'vuex';
```

```
import { auth } from './auth-module';  
Vue.use(Vuex);  
export default new Vuex.Store({  
  modules: {  
    auth  
  }  
});
```

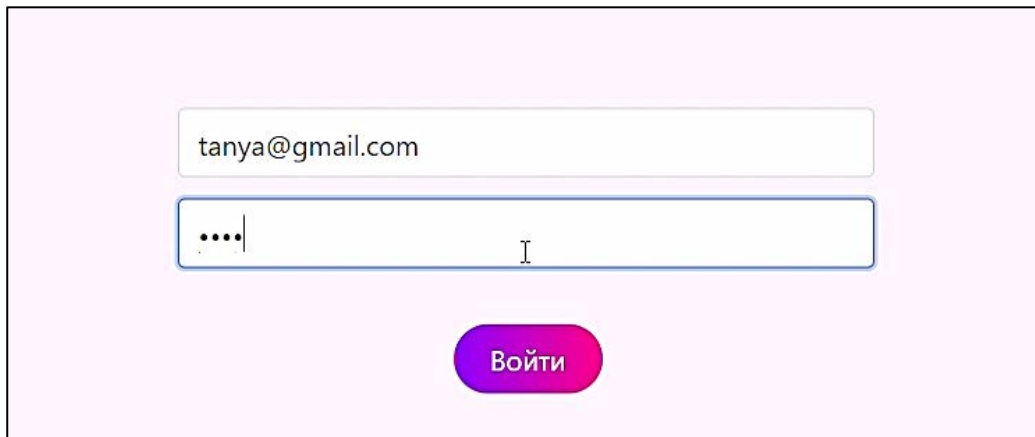
Таким чином, якщо реєстрація пройшла успішно висвічується відповідне повідомлення (рис. 4.9).



The image shows a registration form on a light pink background. It consists of four input fields stacked vertically: the first contains the name 'Татьяна', the second contains the surname 'Вржецион', the third contains the email 'tanya@gmail.com', and the fourth contains a password represented by seven dots. Below the fields is a blue rounded button with the text 'Зарегистрироваться'. At the bottom, a light green message box displays the text 'Регистрация выполнена успешно!'.

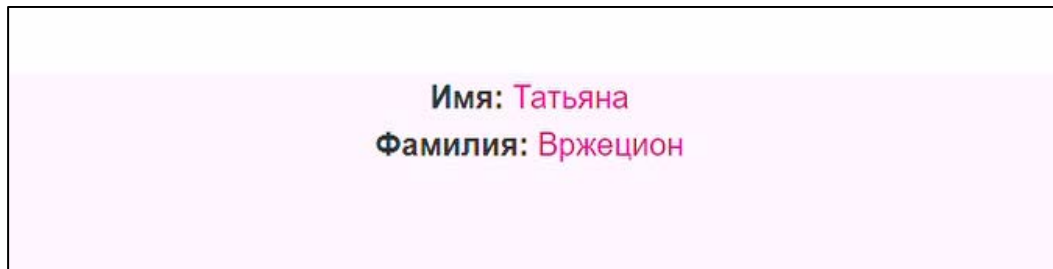
Рисунок 4.9 – Реєстрація в системі

Для того, щоб увійти в систему (рис. 4.10) необхідно ввести вірний електронну адресу і пароль. При успішному вході система перенаправляє на сторінку профілю (рис. 4.11), де відобразяться ім'я та прізвище користувача.



The image shows a login form on a light pink background. It consists of two input fields: the top one contains the email address 'tanya@gmail.com', and the bottom one contains four dots representing a password. Below the fields is a rounded purple button with the text 'Войти' (Login) in white.

Рисунок 4.10 – Вхід в систему



The image shows a user profile section on a light pink background. It displays the text 'Имя: Татьяна' (Name: Tanya) and 'Фамилия: Вржецион' (Surname: Vrzhetsion) in a pink font.

Рисунок 4.11 – Профіль

Для додатка Vue визначаються всі маршрути. Таким чином для роботи з деякими сторінками додатка необхідно виконати вхід в систему, інакше додаток перенаправляє на сторінку входу.

Також необхідно, щоб панель навігації динамічно змінювалася (рис. 4.12) в залежності від ролі поточного користувача, які витягуються з стану Vuex Store.



The image shows a navigation bar on a light pink background. On the left, it displays the user's name 'Имя: Татьяна' and surname 'Фамилия: Вржецион' in pink. On the right, there is a rounded purple button with the text 'Выйти' (Logout) in white. A mouse cursor is visible near the bottom right corner.

Рисунок 4.12 – Змінна панель навігації

ВИСНОВКИ

В результаті аналізу предметної області, проведеного при виконанні дипломної роботи проведено огляд існуючих навчальних рішень і апробація платформи Stepik. Виявлені переваги і недоліки існуючих платформ допомогли сформуванню вимоги до організації призначеного для користувача інтерфейсу призначеного для вивчення математики. Для забезпечення ефективної взаємодії користувача з системою описані Use Case діаграма та сценарії взаємодії, які лягли в основу вимог до управління даними.

Для побудови шаблонів інтерфейсу був обраний онлайн-сервіс Figma, а для реалізації – фреймворк Vue.js.

На основі зазначених в роботі вимог спроектовані та побудовані шаблони інтерфейсу математичної платформи. Доступ до частин інтерфейсу розмежований для не авторизованих користувачів, виконана інтеграція інтерфейсу в архітектуру системи, де зв'язок з системним модулем платформи включає в себе реєстрацію та авторизацію.

Реалізований інтерфейс можна розширювати спроектованими шаблонами, додатковими математичними можливостями, серед яких можливість створення тривимірних об'єктів, і дизайном.

Матеріали роботи доповідались на вісімнадцятій всеукраїнській конференції студентів і молодих науковців «Інформатика, інформаційні системи та технології» (Одеса 23 квітня 2021 р.) та 77-ї звітній студентській науковій конференції Одеського національного університету імені І.І. Мечникова (26 – 27 квітня 2021 року).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В.С. Ітенберг «Математическое образование нематематиков» науковий семінар Міжнародного центру аналізу і вибору рішень 7 червня 2020 р. – [Електронний ресурс] – Режим доступу: <https://www.hse.ru/DeCAn/news/371576279.html>
2. Канал В. С. Ітенберга [Електронний ресурс] – Режим доступу: <https://www.youtube.com/channel/UC87dIoHhtTmDc6hMavtFYUA>
3. Вржеціон Т. О., Трубіна Н. Ф. Специфіка інтерфейсу користувача платформи для навчання вищої математики / Інформатика, інформаційні системи та технології: тези доповідей вісімнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 23 квітня 2021 р. – Одеса, 2021. – с. 162-164
4. Козлов М. С., Трубіна Н. Ф. Розробка архітектури навчальної платформи / Інформатика, інформаційні системи та технології: тези доповідей вісімнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 23 квітня 2021 р. – Одеса, 2021. – с. 164-166
5. Платформа онлайн-курсів EdX [Електронний ресурс] – Режим доступу: <https://www.edx.org/>
6. Система управління курсами Moodle [Електронний ресурс] – Режим доступу: <https://www.edx.org/>
7. Освітня організація Khan Academy [Електронний ресурс] – Режим доступу: <https://www.khanacademy.org/>
8. Освітня платформа Stepik [Електронний ресурс] – Режим доступу: <https://welcome.stepik.org/ru>
9. Студія онлайн-освіти EdEra [Електронний ресурс] – Режим доступу: <https://www.ed-era.com/>
10. Курс «Математика для не математиков» на платформі Stepik [Електронний ресурс] – Режим доступу: <https://stepik.org/course/92929/promo>

11. Курс «Программирование на языках C и C++» [Электронный ресурс] – Режим доступа: <https://stepik.org/course/85190/promo>
12. Vossen, Paul & Maguire, Martin. Guide to mapping requirements to user interface specifications [Электронный ресурс] – Режим доступа: https://www.researchgate.net/publication/243533328_Guide_to_mapping_requirements_to_user_interface_specifications
13. Самбурський В.О., Лісіцина І.М. Використання рекомендаційних систем у навчальних платформах / Інформатика, інформаційні системи та технології: тези доповідей вісімнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 23 квітня 2021 р. – Одеса, 2021. – с. 167-169
14. Петрушина Т. І., Ткаченко А. М. Тематичне моделювання текстів навчальної системи / Інформатика, інформаційні системи та технології: тези доповідей вісімнадцятої всеукраїнської конференції студентів і молодих науковців. Одеса, 23 квітня 2021 р. – Одеса, 2021. – с. 166-167
15. Документація Vue.js [Електронний ресурс] – Режим доступа: <https://ru.vuejs.org/v2/guide/index.html>
16. Figma [Електронний ресурс] – Режим доступа: <https://www.figma.com/>

ДОДАТОК А

СЦЕНАРІЙ ВИКОРИСТАННЯ

Таблиця А.1 – Сценарій проходження пробного курсу

USE CASE	Система проходження пробного курсу	
Основна діюча особа	Гість	
Ціль	Пройти пробний курс	
Тригер	Гість системи вирішує спробувати пройти пробний курс	
Результат	Оцінка за пройдений курс	
Опис	Крок	Дія
	1	Гість переглядає текстовий або відео урок
	2	Гість проходить згенерований набір тестів по пробному модулю
	3	Система показує кількість набраних балів
	4	Гість вирішує зареєструватися в системі
	5	Система зберігає дані про проходження пробного курсу і адаптує до основного
Розширення		Дія розгалуження
	4а	Гість залишає систему
	5а	Система не зберігає дані про проходження пробного курсу

Таблиця А.2 – Сценарій створення курсу

USE CASE	Система створення курсу на платформі	
Основна діюча особа	Викладач	
Ціль	Створити курс	
Тригер	Викладач вирішує створити курс на платформі	
Результат	Курс створений	
Опис	Крок	Дія
	1	Викладач дає курсу назву і опис

Продовження таблиці А.2

Опис	Крок	Дія
	2	Викладач створює модуль курсу і дає йому назву
	3	Викладач створює урок і наповнює його текстом або відео
	4	Викладач створює тести для контрольного або модульного тестування
	5	Система пропонує викладачеві рекомендації по доданим матеріалами

Таблиця А.3 – Сценарій створення групи

USE CASE	Система проходження вхідного тестування	
Основна діюча особа	Викладач	
Ціль	Створити групу на основі курсу	
Тригер	Викладач вирішує створити на основі курсу групу для спрощення управління студентами	
Результат	Група створена і викладач отримує доступ до перегляду успішності і спілкуванню з кожним студентом в групі	
Опис	Крок	Дія
	1	Викладач створює групу і дає їй назву, опис
	2	Викладач робить групу закритою
	3	Система генерує посилання-запрошення в групу
	4	Викладач додає студента в групу
	5	Система надає доступ до історії проходження курсу студентом

Таблиця А.4 – Сценарій призначення тестового контролю

USE CASE	Система проходження вхідного тестування	
Основна діюча особа	Викладач	
Ціль	Провести тестовий контроль	
Тригер	Викладач вирішує провести тестовий контроль для групи студентів	

Продовження таблиці А.4

Результат	Створений тестовий контроль індивідуально для кожного студента і доступний для проходження в зазначений час	
Опис	Крок	Дія
	1	Викладач створює тестовий контроль і дає йому назву
	2	Викладач призначає час та тривалість контролю
	3	Викладач вибирає теми з курсу або модуля для генерації завдань
	4	Викладач додає нові тести і завдання по кожному обраному модулю і виставляє для них максимально-можливу кількість балів

Таблиця А.5 – Сценарій виставлення оцінки

USE CASE	Система виставлення оцінки за завдання	
Основна діюча особа	Викладач	
Ціль	Виставити оцінку за виконане завдання	
Тригер	Викладач вирішує особисто перевірити завдання і виставити за нього оцінку	
Результат	Оцінка виставлена і збережена в таблиці студента	
Опис	Крок	Дія
	1	Викладач відкриває таблиць успішності студентів групи
	2	Система відображає кількість набраних балів за кожне завдання
	3	Викладач вибирає завдання по якому потрібно виставити свою оцінку у певного студента
	4	Система надає можливість подивитися історію відповідей
	5	Викладач змінює оцінку
	6	Система зберігає нову оцінку і перераховує загальну суму балів

Таблиця А.6 – Сценарій проходження вхідного тестування

USE CASE	Система проходження вхідного тестування	
Основна діюча особа	Студент	
Ціль	Пройти вхідне тестування	
Тригер	Студент вирішує пройти вхідне тестування для адаптивності курсу	
Результат	Вхідне тестування пройдено і система підлаштовує курс під результати	
Опис	Крок	Дія
	1	Студент обирає вхідне тестування по курсу
	2	Система пропонує згенеровані тести по кожному модулю
	3	Студент виконує тести і завершує тестування
	4	Система показує відсоток знань по курсу і кожного модуля
	5	Студент обирає проходження курсу з урахуванням набраних навичок
Розширення		Дія розгалуження
	5a	Студент обирає проходження курсу без урахування набраних навичок

Таблиця А.7 – Сценарій проходження модуля курсу

USE CASE	Система проходження модуля курсу	
Основна діюча особа	Студент	
Ціль	Пройти модуль курсу	
Тригер	Студент вирішує пройти один з модулів обраного курсу	
Результат	Модуль курсу пройдено і система відображає відсоток знань студента на основі пройдених тестів	
Опис	Крок	Дія
	1	Студент обирає модуль для проходження
	2	Студент проходить текстові або відео-уроки
	3	Студент виконує тести по модулю

Продовження таблиці А.7

Опис	Крок	Дія
	4	Система показує відсоток знань по модулю
	5	Студент завершує проходження модуля
Розширення		Дія розгалуження
	5а	Система пропонує більш складні тести для того, щоб підвищити відсоток знань
	6а	Студент виконує тести підвищеної складності по модулю
	7а	Система збільшує відсоток знань по успішно виконаних завдань для кожної теми

ДОДАТОК Б
ШАБЛОНИ ІНТЕРФЕЙСУ КОРИСТУВАЧА МАТЕМАТИЧНОЇ
ПЛАТФОРМИ



Рисунок Б.1 – Кольорова гамма

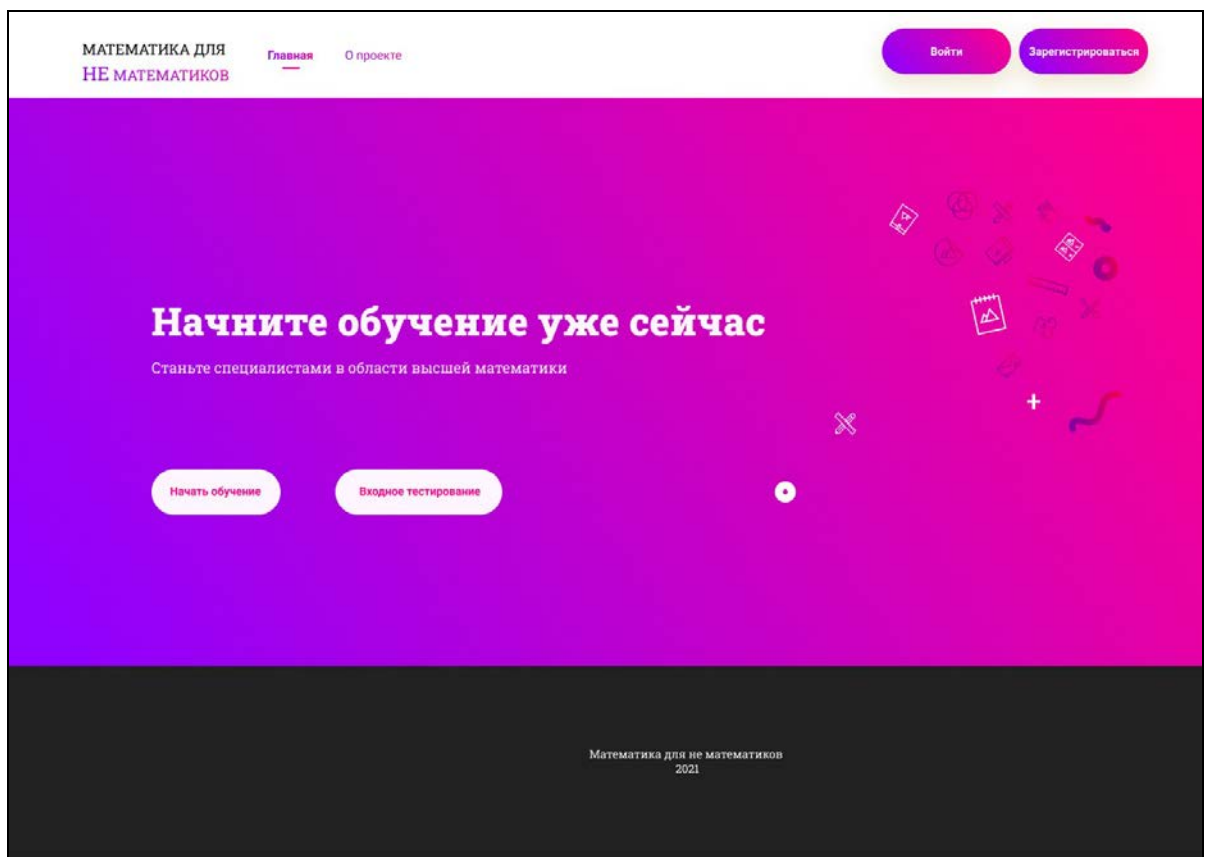


Рисунок Б.2 – Домашня сторінка

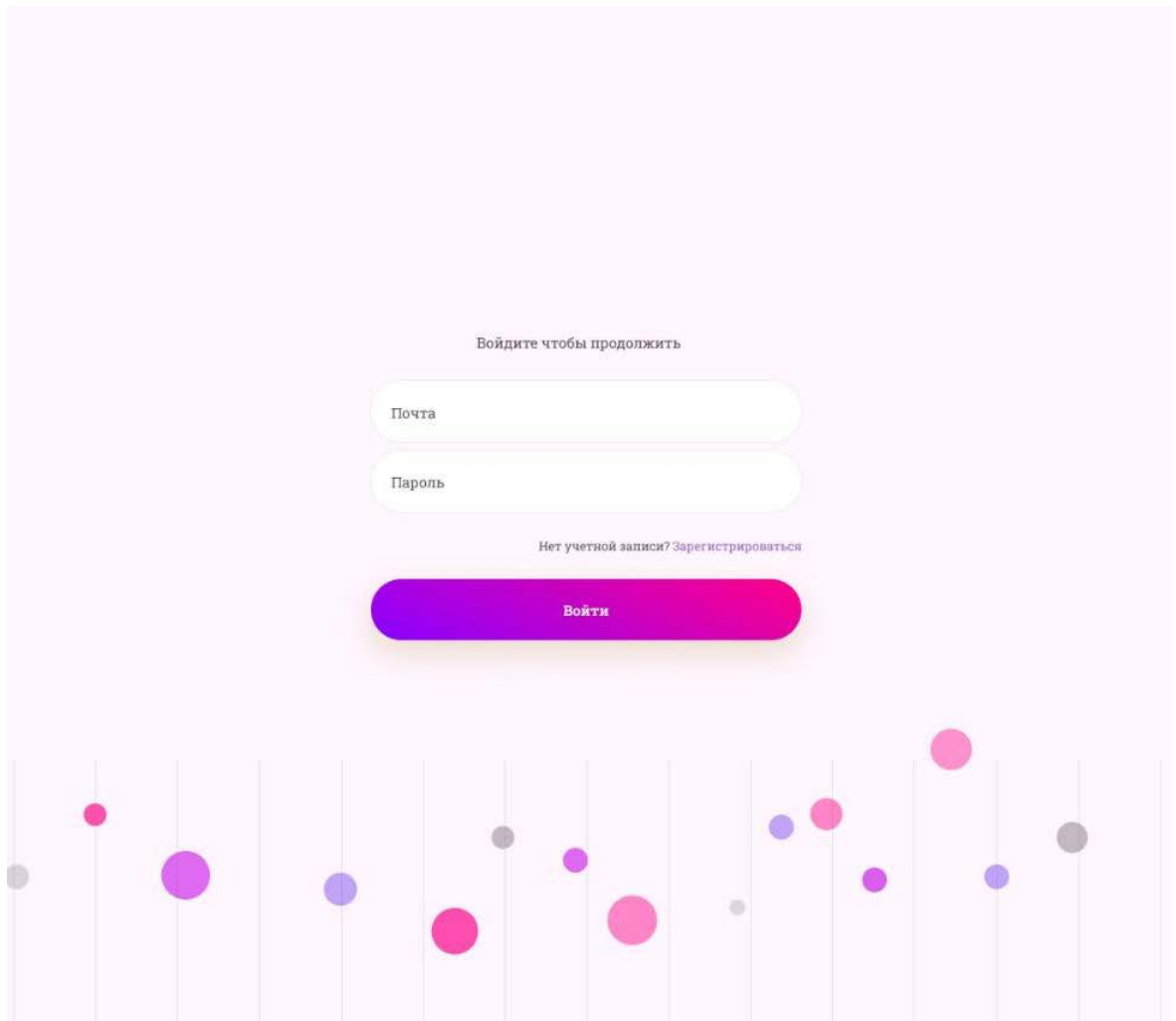


Рисунок Б.3 – Вхід в систему

Пожалуйста, создайте свою учетную запись здесь

Имя

Фамилия

Почта

Пароль

Повторите пароль

Уже есть учетная запись? [Войти](#)

Sign In

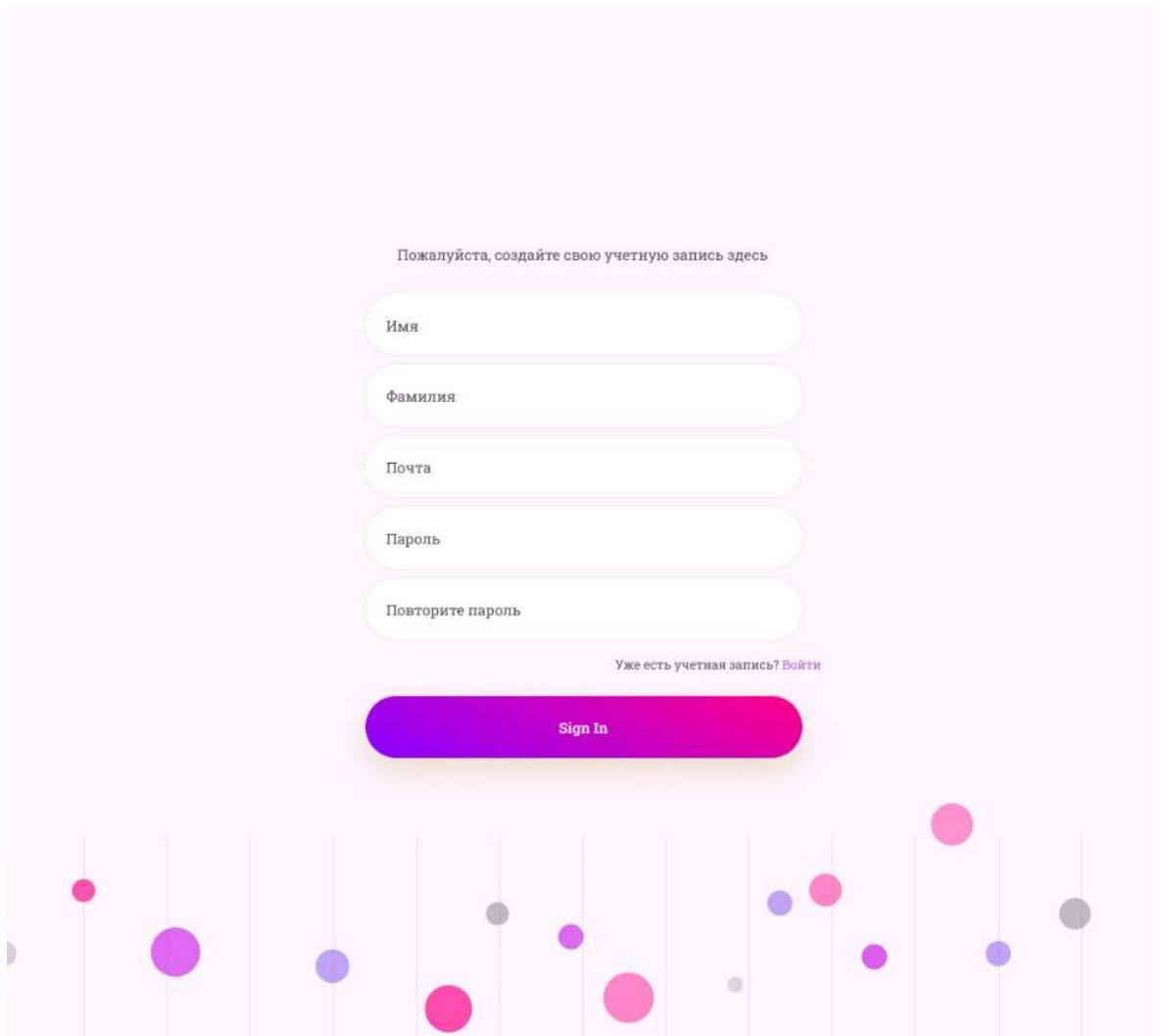


Рисунок Б.4 – Реєстрація

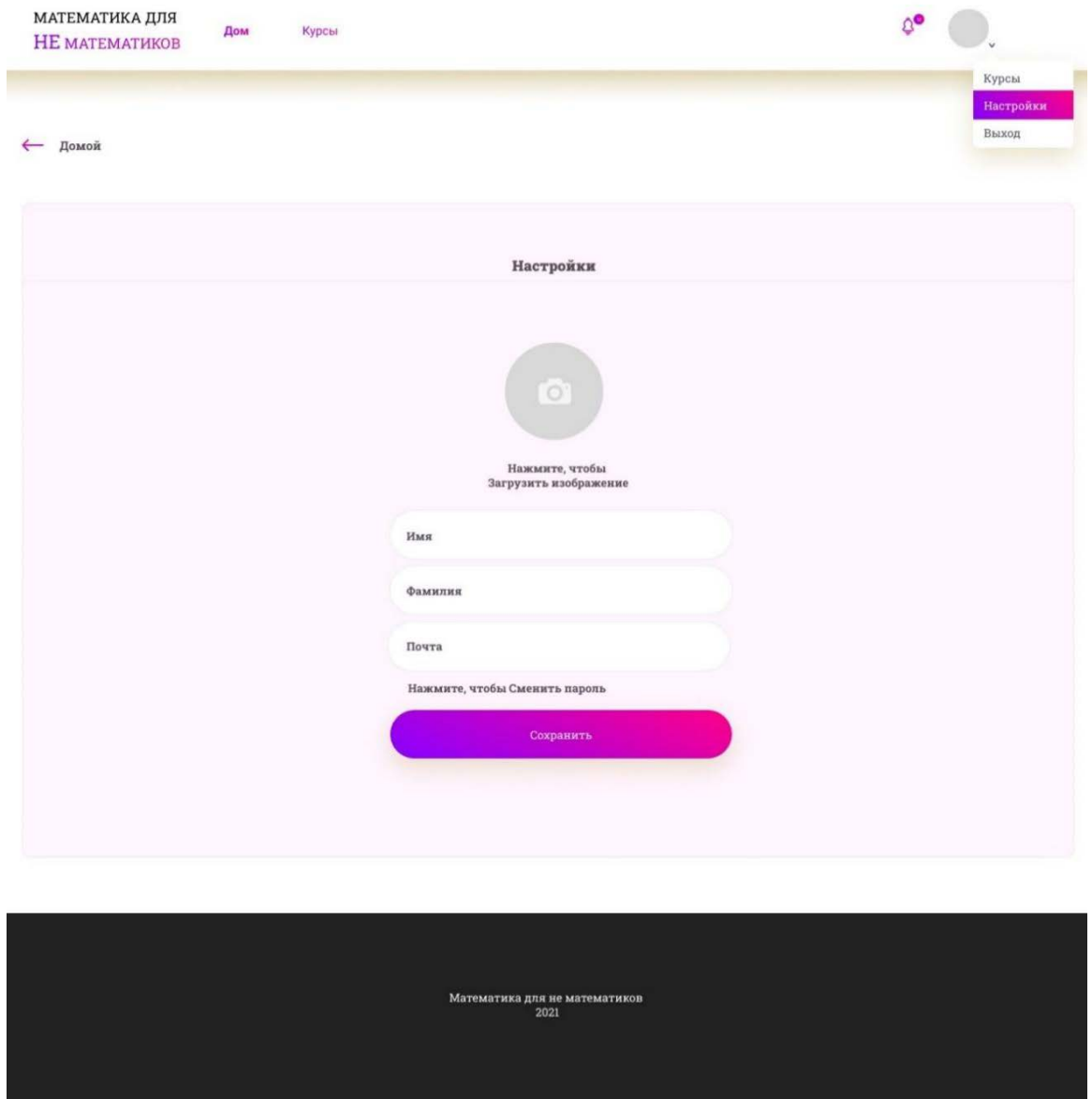


Рисунок Б.4 – Налаштування

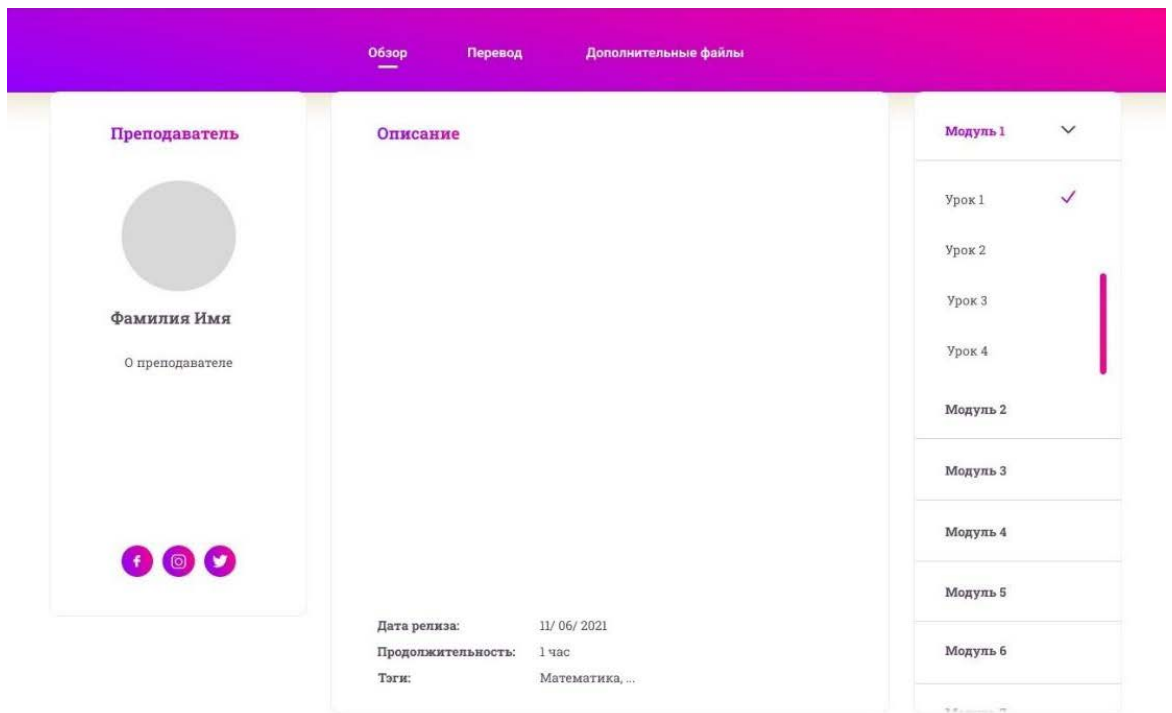


Рисунок Б.6 – Огляд курсу

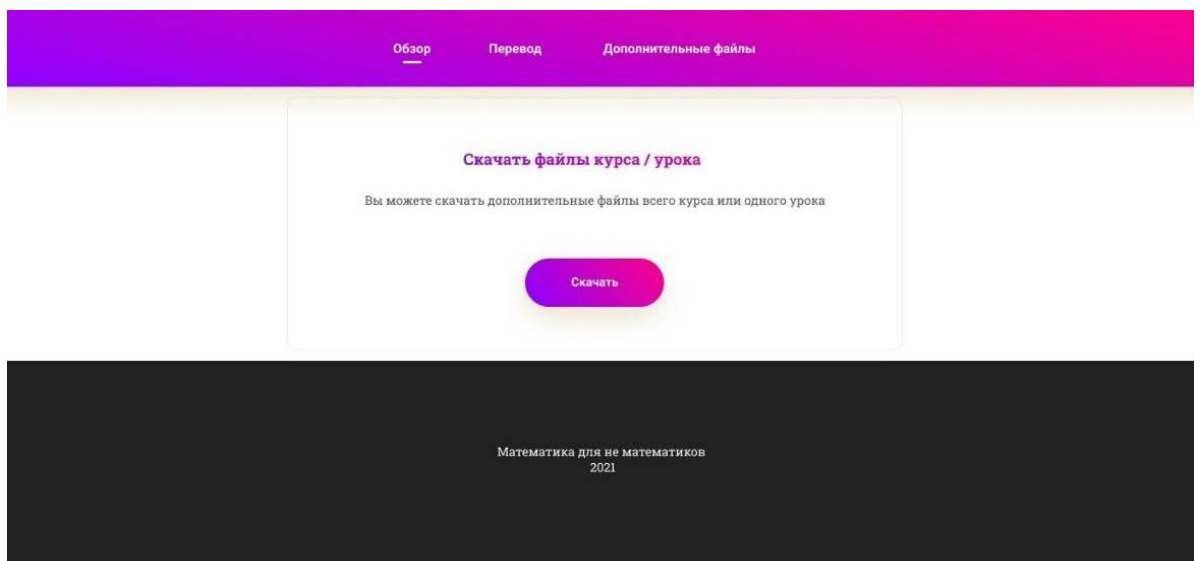


Рисунок Б.7 – Завантаження файлів



Решите задачу

Условие задачи

Ответ

Решите задачу

Условие задачи

Ответ

Решите задачу

Условие задачи

Ответ

Решите задачу

Условие задачи

Ответ

Рисунок Б.8 – Вхідне тестування