



# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних систем

Освітній ступінь бакалавр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»

(назва)

## ЗАТВЕРДЖУЮ

Завідувач

кафедри Інформаційних систем

“ ” \_\_\_\_\_ 2022 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Левонюк Катерина Вікторівна

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення модуля навчання для системи дистанційного навчання для молодших школярів

керівник роботи Костіков Микола Павлович, доцент, кандидат технічних наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “31” березня 2022 року №163-кс

2. Строк подання здобувачем роботи 07.06.2022

3. Вихідні дані до роботи: Навчальні матеріали, інформація про навчальний процес

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1) Системний аналіз

2) Опис комплексу задач автоматизації

3) Формування загального висновку

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Костіков М.П.	04.04.2022	07.06.2022
2.	Костіков М.П.	04.04.2022	07.06.2022

7. Дата видачі завдання 01.04.2022 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми, подання заяви в деканат	01.04.2022	Виконане
2	Підготовка першого розділу роботи та подання його керівнику.	18.04.2022	Виконане
3	Підготовка другого розділу роботи та подання його керівнику.	15.05.2022	Виконане
4	Підготовка вступу та висновків	20.05.2022	Виконане
5	Доопрацювання роботи з урахуванням зауважень керівника.	30.05.2022	Виконане
6	Подача електронного варіанту роботи для перевірки на плагіат.	01.06.2022	Виконане
7	Створення Презентації	07.06.22	Виконане
8			
9			
1			

Здобувач \_\_\_\_\_  
(підпис)

Левонюк К.В.  
(прізвище та ініціали)

Керівник роботи \_\_\_\_\_  
(підпис)

Костіков М.П.  
(прізвище та ініціали)

## АНОТАЦІЯ

Головною метою цієї бакалаврської роботи є розробка та створення платформи дистанційного навчання для учнів 1-4 класів. Основна мета цієї платформи – зробити навчальний процес цінним і захоплюючим для дітей, які переходять до 1-4 класів, і надати кожному учню рівний, необмежений доступ до якісного освітнього контенту.

Робота включає в себе схеми декомпозиції процесу навчання і опис реалізації електронного засобу навчання.

Для реалізації поставлених завдань використано СКБД Firebase 9.6.10 для створення бази даних, IDE WebStorm2021.2, мова програмування JavaScript та фреймворк React (для інтерактивних інтерфейсів).

Бакалаврська робота містить 67 сторінок, 3 таблиці, 38 рисунків, 5 додатків і 18 літературних джерел.

**КЛЮЧОВІ СЛОВА:** ВЕБ ДОДАТОК, НАВЧАЛЬНИЙ МАТЕРІАЛ, ПРЕДМЕТИ, УЧНІ, ВЧИТЕЛІ, БАЗА ДАНИХ.

## **ANNOTATION**

The primary purpose of this bachelor's thesis is to develop and create a platform for distance learning for students in grades 1-4. The main goal of this platform is to make the learning process valuable and exciting for children in grades 1-4, and to provide each child with equal, unlimited access to quality educational content.

The work includes schemes of decomposition of the learning process and a description of the implementation of e-learning.

To implement the tasks, DBMS Firebase 9.6.10 was used to create a database, IDE WebStorm2021.2, JavaScript programming language, and React framework (for interactive interfaces).

The bachelor's thesis contains 67 pages, 3 tables, 38 figures, 5 appendices, and 18 references.

**KEYWORDS: WEB APP, STUDY MATERIAL, SUBJECTS, STUDENTS, TEACHERS, DATABASE.**

## Зміст

ВСТУП .....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ МОДУЛЯ НАВЧАННЯ УЧНІВ 1-4 КЛАСІВ ЗА ДОПОМОГОЮ ТЕСТУВАННЯ ТА ІНШИХ МАТЕРІАЛІВ ТА ПОСТАНОВКА НА ПРОЕКТУВАННЯ.....	9
1.1. Загальна характеристика процесу навчання школярів.....	9
1.2. Організаційна структура типової молодшої школи .....	10
1.2.1. Загальна схема організаційної структури .....	10
1.2.2. Структура школи та навчання в ній .....	11
1.3. Аналіз нинішнього стану комп'ютеризації навчання для системи дистанційного навчання для молодших школярів.....	15
1.4. Розроблення функціональної моделі та аналіз існуючих бізнес- процесів. 16	
1.4.1. Функціональна модель навчання у школі.....	16
1.4.2. Виявлені проблеми і задачі автоматизації.....	17
1.5. Огляд існуючих рішень для розв'язання виявлених проблем.....	18
1.5.1. – Всеукраїнська школа онлайн .....	18
1.5.2. Атмосферна школа .....	19
1.5.3. – На урок .....	20
1.5.3. .... – Проведений аналіз аналогів 22	
1.6. Обґрунтування доцільності проектування й розроблення EasyLearn. 22	
1.7. Концептуальна модель системи. ....	23
1.8. Постановка задачі .....	25
1.8.1. Призначення та цілі створення системи .....	25
1.8.2 Вимоги до створюваної системи .....	26
1.8.3 Функції, які повинна виконувати система.....	27
1.8.4 Вхідні та вихідні дані системи.....	27

РОЗДІЛ 2. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	28
2.1. Технічне та системне забезпечення розробки. ....	28
2.2. Інформаційне забезпечення системи. ....	29
2.3. Алгоритмізація та реалізація комплексу задач автоматизації. ....	32
2.4. Інструкція користувача.....	42
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	52

## ВСТУП

Ми вступаємо на дистанційне навчання. І для дітей складно сприймати інформацію через відео та програми, такі як: «Zoom» і «Google meet». Втома від відеоконференцій може бути справжньою професійною небезпекою в епоху COVID-19, але це не привід відмовлятися від інтенсивних навчальних занять. Веб додатки можуть мати вроджену перевагу — відсутність відволікаючих факторів у середовищі. Та завдяки ретельному дизайну можуть досягти сильного рівня залучення, що сприяє інтенсивному навчанню у віртуальній обстановці.

Творче використання технологій та ретельний дизайн курсу можуть значною мірою сприяти розвитку віддаленого навчання, завдяки багатьох інструментів, які дозволяють обговорювати, виконувати вправи та вирішувати командні проблеми. Насправді дистанційне навчання надає перевагу перед особистим навчанням: людям, яким менш зручно виступати на публіці чи груповій діяльності — або хто може побоюватися розгортання делікатного, але важливого питання — може бути зручніше брати участь на відстані. Таким чином, дистанційне навчання може спровокувати більшу участь і виявити проблеми, які інакше не виникли б.

Інтенсивне, захоплююче навчання важко створити за допомогою екрана комп'ютера, але завдяки ретельному дизайну ми можемо досягти сильного рівня залучення дітей, що сприяє інтенсивному навчанню у віртуальній обстановці.

Основна мета цього додатка – зробити процес навчання зручним та захоплюючим для дітей, які йдуть до 1-4 класів, та надати кожному вчителю рівний, необмежений доступ до якісного навчального контенту.

# **РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ МОДУЛЯ НАВЧАННЯ УЧНІВ 1-4 КЛАСІВ ЗА ДОПОМОГОЮ ТЕСТУВАННЯ ТА ІНШИХ МАТЕРІАЛІВ ТА ПОСТАНОВКА НА ПРОЕКТУВАННЯ**

## **1.1. Загальна характеристика процесу навчання школярів**

Освіта є сферою діяльності держави щодо забезпечення громадянин отримання знань, процес і результат засвоєння особистістю певної системи наукових знань, практичних умінь та навичок і пов'язаного з ними того чи іншого рівня розвитку її розумово-пізнавальної і творчої діяльності.

Наразі у школах реформа навчання ще не наблизилась до нової системи освіти. Вчителі використовують стару та усім звичну нам форму освіти. Сьогодні вся середня і старша школи без винятку працюють за старою та не дуже ефективною в наш час системою, в якій учитель викладає тему, учні відповідають на запитання і роблять вдома завдання.

У цій системі вчителі не звикли, а іноді й просто не вміють співпрацювати з учнями, а якщо точніше – не бачать у цьому сенсу. Їм треба викласти матеріал, поставити завдання і перевірити його. Це чудово можна зробити, навіть користуючись найпростішими технічними засобами.

Наприклад, можна скинути у вайбер-групу класу посилання на рандомно обраний онлайн-урок за необхідною темою. Або навіть на конкретне пояснення нової теореми, частини мови чи формули – без зворотного зв'язку. Хай дитина все це подивиться вдома. Виконання домашнього завдання зараз проходить так: сфотографуйте, будь ласка, сторінки зошиту з виконаними вправами і надішліть мені, я поставлю оцінки.

Тобто реальна проблема впровадження повноцінного дистанційного навчання полягає не в тому, що хтось не має потужного комп'ютера або не вміє зайти в Zoom – усе це за бажання можна вирішити. Проблема в тому, що запропонована сучасна форма навчання не корелюється через принципи, та методи нереформованої системи освіти.

Вочевидь, дистанційне навчання не є калькою очного навчання у класі і потребує нової структури організації уроку – хоча б тому, що увага і втома біля комп'ютера працюють геть інакше.

Тобто, коли дитина перед уроком дійсно самостійно опановує матеріал, як змогла – а час уроку використовується, щоб разом з учителем проговорити, що незрозуміло, виправити помилки, якщо вони є, з'ясувати складні питання. Саме така система закладає в дитині паростки вміння вчитися самостійно. І зараз – найкраща ситуація для того, щоб її починати опановувати.

## **1.2. Організаційна структура типової молодшої школи**

### **1.2.1. Загальна схема організаційної структури**

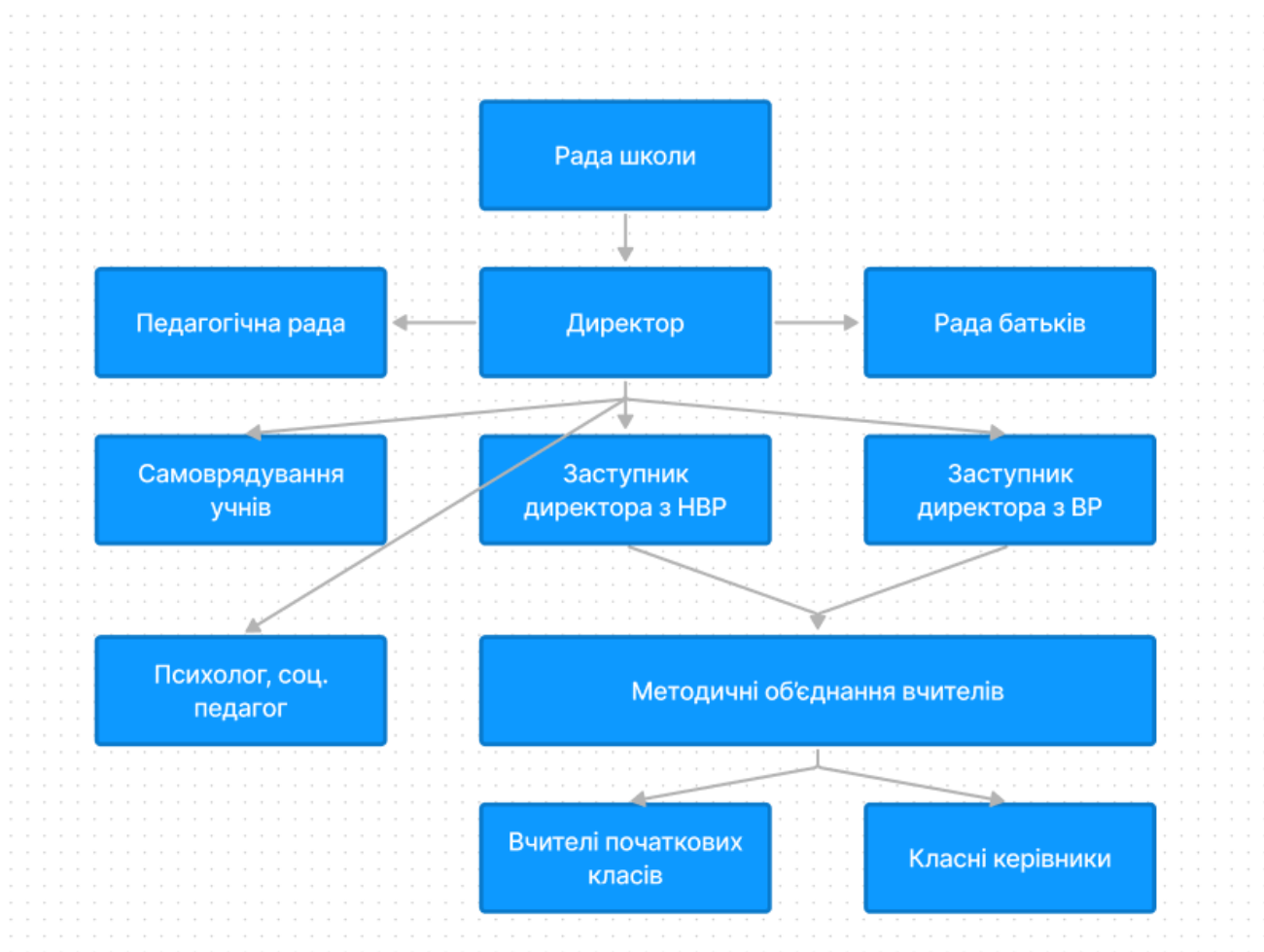
Дистанційне навчання, яке також називають електронним навчанням та онлайн-навчанням, основними елементами якого є фізичне відокремлення викладачів і студентів під час навчання та використання різних технологій для полегшення спілкування між студентом і викладачем, та студентом з студентом.

Школи, як і багато інших організацій, як правило, дотримуються стандартних організаційних структур. Вони мають певну ієрархію та чіткі межі щодо того, які працівники чи відділи відповідають за різні завдання, пов'язані з функціонуванням школи.

У школах, як правило, є директори, які відповідають за роботу всієї школи та нагляд, а також викладачі різних предметів, поділені на основі рівнів класів або навчальних предметних областей. У них також є офісні працівники та допоміжний персонал для виконання різних функцій, необхідних для роботи школи [7].

У початкових і середніх школах зазвичай вчителі поділені на команди на основі рівня учнів, хоча вчителі середньої школи також можуть бути розділені на команди за основними академічними напрямками, такими як мистецтво мови, математика, природознавство та суспільствознавство. Вчителі середньої школи, як правило, працюють на академічних факультетах, заснованих на академічних предметних областях в основних областях, а також на таких факультетах, як бізнес,

іноземна мова та кар'єра. Відділи викладачів часто планують окремо та разом, щоб забезпечити викладання необхідної навчальної програми та обслуговування своїх учнів у міру своїх можливостей.



*Рисунок 1.1 – Організаційна структура школи*

### **1.2.2. Структура школи та навчання в ній**

Школа – це навчальний заклад, який забезпечує початкову, середню, а в деяких випадках вищу чи спеціальну освіту кожному із своїх учнів. В залежності від знань, які можуть запропонувати школи вони діляться на певні типи[13].

Основними з них є загальноосвітні та професійні школи. Рівень навчання визначає поділ шкіл на початкові, неповні середні, середні та вищі.

В певні періоди навчання в школі становило десять або дванадцять років, а по закінченні школи учні мали скласти іспити із затверджених дисциплін за

результатами яких видається Свідоцтво про базову або повну середню освіту. Навчання в школі має три основні ступені – молодші класи, середні класи та старші класи[9].

Таблиця 1.2 – Функції та задачі вчителів

Повинні	Зобов'язані стосовно учнів
<ul style="list-style-type: none"> <li>● постійно підвищувати:               <ul style="list-style-type: none"> <li>○ свій професійний і загальнокультурний рівні</li> <li>○ педагогічну майстерність</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>● сприяти:               <ul style="list-style-type: none"> <li>○ розвитку здібностей учнів</li> <li>○ формуванню навичок здорового способу життя</li> </ul> </li> <li>● дбати про фізичне і психічне здоров'я учнів</li> </ul>
<ul style="list-style-type: none"> <li>● виконувати освітню програму для досягнення учнями передбачених нею результатів навчання</li> </ul>	<ul style="list-style-type: none"> <li>● особистим прикладом утверджувати повагу до суспільної моралі та суспільних цінностей, зокрема:               <ul style="list-style-type: none"> <li>○ правди</li> <li>○ справедливості</li> <li>○ патріотизму тощо</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● дотримувати:               <ul style="list-style-type: none"> <li>○ академічної доброчесності та забезпечувати її дотримання учнями в освітньому процесі</li> </ul> </li> <li>● педагогічної етики</li> </ul>	<ul style="list-style-type: none"> <li>● формувати в учнів:               <ul style="list-style-type: none"> <li>○ усвідомлення необхідності дотримуватися Конституції та законів України, захищати суверенітет і територіальну цілісність України</li> <li>○ прагнення до взаєморозуміння, миру, злагоди між усіма народами, етнічними, національними, релігійними групами</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● поважати гідність, права, свободи і законні інтереси всіх учасників освітнього процесу</li> </ul>	<ul style="list-style-type: none"> <li>● виховувати в учнів повагу до державної мови і державних символів, цінностей України, навколишнього природного середовища тощо</li> </ul>
<ul style="list-style-type: none"> <li>● дотримувати:               <ul style="list-style-type: none"> <li>○ установчих документів закладу освіти</li> <li>○ правил внутрішнього розпорядку</li> </ul> </li> <li>● виконувати посадові обов'язки</li> </ul>	<ul style="list-style-type: none"> <li>● захищати учнів під час освітнього процесу від:               <ul style="list-style-type: none"> <li>○ будь-яких форм фізичного та психічного насильства</li> <li>○ приниження честі та гідності</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ дискримінації за будь-якою ознакою тощо</li> <li>● запобігати: <ul style="list-style-type: none"> <li>○ вживанню учнями та іншими особами на території закладів освіти алкогольних напоїв, наркотичних засобів</li> <li>○ іншим шкідливим звичкам</li> </ul> </li> </ul>
--	---

### **Права та обов'язки учнів[17]:**

#### Обов'язки учнів:

1. Виконувати статут школи.
2. Оволодівати знаннями.
3. Не пропускати уроки, не запізнюватися на заняття.
4. Приходити до школи охайними, в шкільній формі.
5. Акуратно вести щоденник і подавати його за першою вимогою вчителя.
6. Брати активну участь у житті школи, підтримувати і примножувати традиції школи, брати активну участь у позакласній і гуртковій роботі.
7. Бути ввічливим у спілкуванні з учителями, з повагою ставитись до батьків. Проявляти милосердя, турбуватися про молодших.
8. Гідно поводити себе, дотримуватися культури поведінки.
9. Бережливо ставитись до державного, суспільного і особистого майна, охороняти природу.
10. Діяти завжди на благо колективу, турбуватися про честь і підтримку традицій школи.
11. Підкорятися вказівкам та розпорядженням директора школи, педагогічної ради, а також постановам шкільного самоврядування.

#### Права учнів:

1. На захист прав особистості і ввічливе ставлення до себе.

2. На вибір факультативів, гуртків, екзаменів із окремих предметів і складання екстерном програми НФН.

3. На відкрити оцінку своїх знань, отримання об'єктивної оцінки із предметів.

4. На завчасне повідомлення про строк контрольної роботи, вимагати проведення не більше двох контрольних робіт у день і не більше п'яти контрольних робіт у тиждень.

5. Вносити пропозиції в раду, адміністрацію школи по покращенню програмного навчання і виховання.

6. На забезпечення якісним харчуванням.

7. Висловлювати класному керівнику, директору, вчителям свої проблеми і отримувати від них допомогу, пояснення, відповіді.

8. Відкрито висловлювати свою думку стосовно життя школи, не принижуючи гідності інших.

9. Представляти школу в конкурсах, оглядах, змаганнях та інших заходах відповідно зі своїми можливостями і вміннями.

10. На звільнення від уроків згідно заяви батьків.

У наш час слово «школа» вживається для позначення такого місця або установи, де навчаються, де отримують необхідні для життя знання. Але первісне значення слова «школа» нагадує всім учням і вчителям, що перед дверима школи повинен стихати галас життєвої метушні. У школі немає місця пустошам і життєвим турботам, а головною метою занять у школі повинен бути розвиток духовних якостей учнів.

Учень, як істота розумна та вільна, з надходженням до школи закликається до розумного життя, до самостійного засвоєння пропонованих їй знань, до свідомого й добровільного виконання правил моральної поведінки. Школа не буде відповідати своєму імені, якщо учні перенесуть в неї всі недоліки сімейного життя, якщо вони не залишать за порогом школи звичок до неуважності й неробства, грубих ігор та сварок, або якщо вони виявляють небажання вчитися.

### **1.3. Аналіз нинішнього стану комп'ютеризації навчання для системи дистанційного навчання для молодших школярів**

Програми дистанційного навчання охоплюють мінімальний фізичний контакт між вчителем і школярем, але наголошують на тому, що в основному покладаються на електронні комунікації. Це означає, що учень отримує формалізоване навчання, поки знаходиться вдома [14].

Причина запуску програм дистанційного навчання полягає в тому, що учні можуть навчатися з вибраних місць, які можуть бути за тисячі миль. Тому очікується, що дистанційне навчання вимагатиме широкого використання інформаційно-комунікаційних технологій.

Досягнення електронних інформаційно-комунікаційних технологій (ІКТ), які швидко змінюють соціальні та економічні умови в усьому світі, принесли значне покращення в секторі освіти. Це відбувається шляхом надання нових інструментів для забезпечення доступу до інформації та управління знаннями, а також обміну. Комп'ютери справили значний вплив на наше суспільство, особливо в галузі освіти. Комп'ютери є поширеними інструментами в більшості шкіл, і вони все частіше використовуються в усіх предметних областях.

Використання комп'ютерів полегшує навчання, оскільки учні використовують програми для обробки текстів, щоб набирати завдання, виконувати класну роботу та переглядати Інтернет, щоб отримати доступ до поточної інформації.

Наразі, половина загальної кількості учнів не мають доступу до класів через пандемію COVID-19, не мають доступу до домашнього комп'ютера, а 43% (706 мільйонів) не мають Інтернету вдома в той час, коли цифрове дистанційне навчання використовується для забезпечення безперервності освіти в переважній більшості країн.

Навіть для вчителів у країнах із надійною інфраструктурою інформаційно-комунікаційних технологій (ІКТ) швидкий перехід на онлайн-навчання був складним завданням. Для вчителів у регіонах, де ІКТ та інші дистанційні методики менш доступні, перехід був ще складнішим або неможливим.[16]

Вчителям також потрібна підготовка, щоб ефективно проводити дистанційну та онлайн-освіту, але такої підтримки особливо мало в країнах з низьким рівнем доходу.

## **1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів.**

### **1.4.1. Функціональна модель навчання у школі**

У першу чергу, необхідно проаналізувати, як працює створювана система. Для цього будується функціональна модель AS-IS, яка описує процес навчання для системи дистанційного навчання для молодших школярів, в тому вигляді, як він здійснюється зараз.

Аналіз функціональної моделі AS-IS дозволяє зрозуміти, де знаходиться проблемна ситуація, в чому полягатимуть переваги нових процесів і яким змінам піддається існуюча структура організації процесу.

Контекстна діаграма має два рівні декомпозиції, та наведена на рис.1 у Додатку А.

Входи моделі:

- Навчальний матеріал – це сукупність матеріалу про навчання. Він може складатися з основної літератури або посилатися на неї, ураховуючи рекомендації МОН.
- Рекомендована література – це додаткова література, для самостійної роботи школяра за темою занять.

Виходи моделі:

- Закріплені знання – засвоєний та опрацьований школяром навчальний матеріал.

Управління моделі:

- Рекомендації МОН – це рекомендації, надані міністерством освіти та науки України. Саме за цим стандартом складаються теми та домашні завдання.

Механізми моделі:

- Школярі 1-4 класів;
- Вчитель.

На першому рівні декомпозиції (блоку “Вивчення теми школярем“) діаграма складається з чотирьох блоків:

- Викладання матеріалу школяреві;
- Вивчення інформації школярем;
- Перевірка отриманих знань;
- Виконання домашнього завдання для закріплення матеріалу.

Модель AS-IS показує процес навчання для системи дистанційного навчання для молодших школярів.

#### **1.4.2. Виявлені проблеми і задачі автоматизації**

В результаті проведеного функціонального аналізу було виявлено основну складність використання систем дистанційного навчання – це те, що у процесі навчання вони мають власну програму навчання, яка не пов’язана з 9 конкретними підручниками або навчальною літературою.

Саме тому такі електронні засоби, на жаль, не можливо використовувати при навчанні в школах, адже між їхнім змістом і змістом навчальної програми НЗ є суттєві відмінності. Таке поєднання програм навчання мало того, що не принесе

ніякого покращення або полегшення засвоєння матеріалу, а навпаки — ускладнить процес розуміння та заплутає учня.

Задачі автоматизації, які дозволять усунути наявні проблеми:

- подання учням навчального матеріалу без допоміжних речей;
- можливість дистанційно навчатись та проходити теми по предметах;
- тестування знань для самоперевірки та контролю знань без оцінки.

### **1.5. Огляд існуючих рішень для розв'язання виявлених проблем.**

Для навчання за допомогою систем дистанційних навчань є досить велика кількість програм, але дуже мало є саме для школярів 1-4 класів. За для досконалості у розробці даного веб-додатку було проаналізовано декілька аналогів схожого типу. Переглянемо їх:

#### **1.5.1. – Всеукраїнська школа онлайн**

Всеукраїнська школа онлайн — це платформа [10] для дистанційного та змішаного навчання учнів 5-11 класів та методичної підтримки вчителів.

Мета Всеукраїнської школи онлайн — забезпечити кожному українському учневі та вчителю рівний, вільний і безоплатний доступ до якісного навчального контенту.

Платформа містить відеоуроки, тести та матеріали для самостійної роботи з 18 основних предметів: українська література, українська мова, біологія, біологія та екологія, географія, всесвітня історія, історія України, математика, алгебра, алгебра і початки аналізу, геометрія, мистецтво, основи правознавства, природознавство, фізика, хімія, англійська мова та зарубіжна література.

Увесь навчальний контент відповідає чинним державним освітнім програмам, а його якість перевірена Українським інститутом розвитку освіти. Перелік та тематику курсів запропоновано Міністерством освіти і науки України.

Платформу створено Громадською спілкою «Освіторія» на замовлення Міністерства цифрової трансформації України, Міністерства освіти і науки України та державної установи «Український інститут розвитку освіти», за

підтримки Швейцарії в межах Швейцарсько-українського проекту DECIDE, який виконується Консорціумом ГО DOCCU та PH Zurich.

Переваги програми:

- Можна скачати мобільний додаток;
- Є відео з поясненням як користуватися платформою;
- декілька спроб для виконання тесту.

Недоліки програми:

- Важка навігація по курсу;
- Завантажування конспекту для уроку.

### **1.5.2. Атмосферна школа**

Атмосферна Школа [12] – інноваційний проект у галузі дистанційної освіти в Україні від компанії ENSOF – найбільшого українського розробника та виробника шкільного обладнання для STEAM та інклюзивної освіти з 10-річним досвідом роботи на українському та міжнародному ринках. Ми розуміємо важливість якісної освіти та необхідність її доступності для кожного. Працюючи над Атмосферною Школою, ми, перш за все, зробили формат максимально цікавим та мотивуючим до навчання. Для цього ми використовуємо виключно авторські навчальні матеріали, що є інформативними та зрозумілими для сприйняття. Безліч відеоматеріалів, а саме відеолaboratorні роботи з природничих наук, наочно та лаконічно демонструють вивчений матеріал, а нестандартна форма викладання викликає інтерес до подальшого навчання. Для тих, хто не впевнений, що варто вивчати не усі теми, ми створили мотиваційні ролики, які допоможуть дитині налаштуватися на ефективне навчання онлайн. Ми цінуємо індивідуальність, тому розкладом онлайн занять в Атмосферній Школі учень керує сам. Дистанційне навчання у власному ритмі дає можливість поєднувати школу з хобі та особливостями стилю життя саме Вас та Вашої дитини. Ми завжди на зв'язку, а наші вчителі готові допомогти учням із засвоєнням матеріалу уроків, поясненням тестів для самоперевірки та обговоренням результатів контрольних робіт. Наша ліцензована дистанційна школа створює умови отримання найкращої освіти

онлайн саме для Вашої дитини! Атмосферна Школа Створи свою атмосферу навчання.

Переваги програми:

- Різні форми навчання: «Учень», «Слухач», «Екстерн», «Для закладів освіти»;

- Відео з прикладами уроків, як користуватися платформою, та інше;

- Окремий акаунт для батьків;

Недоліки програми:

- Тусклі кольори;

- Потрібно використовувати книжки для навчання;

### 1.5.3. – На урок

Всеукраїнські онлайн-конкурси, олімпіади, курси та журнал для школярів. Онлайн-школа «На Урок» [11] створена для того, щоб кожен школяр, незалежно від місця проживання, зміг отримувати якісні знання в цікавому форматі.

За допомогою цієї Онлайн-школи ви зможете: покращити знання, підвищити шкільні оцінки, підготуватися до ДПА та ЗНО. Щотижневі уроки в режимі онлайн з можливістю повторного перегляду у запису, домашні завдання з перевіркою, можливість спілкуватися з викладачем.

Основні функції програми:

- Забезпечення тестуванням для учнів;

- можливість створення тестів вчителями;

- створення текстової частини уроку;

- можливість зареєструватися на курси.

Недоліки програми:

- Реклама на пів сторінки;

- дуже багато функцій на сторінці через це розбігаються очі.

Таблиця 1.5.1. Результати порівняння систем-аналогів

Функції	Системи			
	Всеукраїнська	Атмосферна	На урок	EasyLearn

	школа онлайн	школа		
Забезпечення хорошою навігацією	-	-	+	+
Забезпечення хорошого настрою для дітей за допомогою дизайну	+	-	-	+
Забезпечення уроків для учнів 1-4 класів	-	-	-	+
Тестування для учнів	+	+	+	+
Використання додаткової інформації	+	+	-	+
Нагороди за пройдені тести	-	-	-	-

**Висновки:** Отже, проаналізувавши три програмні продукти, для навчання на дистанційній основі можливо зрозуміти, що на всіх платформах є тестування, для закріплення матеріалу. Лише у двох Всеукраїнська школа онлайн та EasyLearn є забезпечення хорошого настрою для дітей за допомогою дизайну, Тільки 3 платформи забезпечені хорошою навігацією – це Всеукраїнська школа онлайн, На урок та EasyLearn. Використання додаткової інформації наділено всім, окрім На урок.

Наразі тільки у одного програмного продукту є забезпечення уроків для учнів 1-4 класів, що дає велику перевагу враховуючі всі попередні плюси.

Також з переваг розглянутих програм можна виділити і нагороди за вивчення теми завдяки проходження тесту, як у EasyLearn, адже саме вони надають школяреві мотивації заходити у програму та навчатись.

### **1.5.3. – Проведений аналіз аналогів**

На даний момент для нас уже привичне дистанційне навчання. У просторах інтернету уже є дуже велика кількість різних платформ для навчання учнів [15], головний їхній недолік націлений на те, що це навчання для 5-11 класів, які уже прекрасно уміють користуватися Інтернетом та іншими гаджетами.

Розглянувши аналоги було зрозуміло, що є три дуже великі проблеми у створенні систем дистанційного навчання, а це саме:

- Дизайн, наявний в цих системах, часто є однотипними та одноманітними. Це призводить до зниження мотивації студента працювати з цими засобами або й навчатись загалом. Нудний та незрозумілий дизайн стає для дітей просто сірою платформою, в якій вони мало що розуміють.
- Наступною проблемою є те, що для дітей потрібно щоб були залучені усі органи чуття, для найкращого навчання.
- Також відсутність нагород за вивчення теми завдяки проходження тесту, адже саме вони надають школяреві мотивації заходити у програму та навчатись.

### **1.6. Обґрунтування доцільності проектування й розроблення EasyLearn.**

З огляду на наявні недоліки було вирішено створити систему, яка б розв'язувала існуючі проблеми у навчанні. Оскільки дистанційне навчання прийшло до нас різко, ми не очікували, що нам потрібно буде так швидко навчатись новому та звикати до онлайн навчання. Особливо важко це було для 1-4 класів, яким важко прийняти нову інформацію і ще й через різні спеціальні додатки для проведення дистанційного навчання. Тому ми вирішили розробити саме цей додаток, який націлений для дітей та вчителів, для кращого сприйняття інформації.

Діти зможуть використовувати систему, щоб навчатися під час карантину та вивчати тему, яку пропустили в школі через хворобу чи інші причини. Для вчителів буде розроблено рекомендації щодо змішаного та дистанційного навчання за допомогою навчальних матеріалів платформи.

### **1.7. Концептуальна модель системи.**

Основною ціллю створення системи дистанційного навчання для молодших класів є забезпечити учнів 1-4 класів навчанням за допомогою розважальних відео, використанням нагород та неодноманітного дизайну з різними ілюстраціями. Це дозволить учням мотиваційно та не нудно вивчати матеріал, використовуючи мінімум два органи чуття. Крім того, вони не будуть обмежені лише тим матеріалом, що надається викладачем під час навчання, а зможуть вивчати наступні теми, де матеріал буде вже зручно структурований [18].

Модель TO-BE використовуємо, аби показати можливість частково автоматизувати процес навчання за допомогою спеціальної програми. Контекстна діаграма має один рівень декомпозиції (див. рис.Б.1 в Додатку Б).

У моделі TO-BE з'являється новий механізм, окрім школярів – платформа EasyLearn – засіб навчання, до вихідних даних закріплених знань додається ще й виконаний тест – це остаточний результат вивчення матеріалу школярем. отримує оцінку за проходження тесту з певного предмету та певної теми. З'являються нові вхідні дані – питання та відповіді до тестів.

На першому рівні декомпозиції (блоку “Вивчення теми школярем“) діаграма складається з чотирьох блоків:

- Викладання матеріалу школяреві за допомогою розважальних відео;
- Вивчення та перегляд додаткової інформації школярем;
- Перевірка отриманих знань за допомогою тесту;
- Виконання домашнього завдання для закріплення матеріалу.

Система буде виконувати наступні задачі та функції:

Таблиця 1.7.1. Основні задачі та функції

<b>Підрозділ</b>	<b>Задачі</b>	<b>Функції</b>
Предмети	Забезпечити дітей усіма навчальними предметами та представити їх з легкою навігацією	<ul style="list-style-type: none"><li>- Вибір предмету;</li><li>- Пересування по предметах за допомогою навігації;</li><li>- Перегляд інформації про предмет;</li><li>- Перехід до наступного розділу – «Теми».</li></ul>
Теми	Забезпечити дітей усіма навчальними темами та представити їх з легкою навігацією	<ul style="list-style-type: none"><li>- Вибір теми;</li><li>- Пересування по темах за допомогою навігації;</li><li>- Перегляд інформації про тему;</li><li>- Перехід до платформи де є навчальні відео, додатковий матеріал та тест.</li></ul>
Навчальні відео	Безпосередньо нам потрібно забезпечити доступними та легкими відео для кращого сприйняття інформації для дітей	<ul style="list-style-type: none"><li>- Перегляд відео;</li><li>- Позначення, що робота виконана (відео переглянуте);</li><li>- Перехід до тем;</li><li>- Перехід до додаткової інформації;</li><li>- Перехід до тесту.</li></ul>

Додаткова інформація	Забезпечити додатковим текстом, який доповнює і повторює інформацію з відео	<ul style="list-style-type: none"> <li>- Перегляд тексту;</li> <li>- Позначення, що робота виконана (інформація прочитана);</li> <li>- Перехід до тем;</li> <li>- Перехід до навчальних відео;</li> <li>- Перехід до тесту.</li> </ul>
Тест	Перевірка вивченого матеріалу, повторення його та закріплення за допомогою тесту	<ul style="list-style-type: none"> <li>- Перегляд тесту та інформації про нього;</li> <li>- Перехід до самого тесту;</li> <li>- Виконання тесту;</li> <li>- Показ результатів.</li> </ul>

І хоча дистанційне навчання не є заміною очного та ніколи не планувалось на довгострокову перспективу, воно може стати ефективним інструментом не тільки під час карантину.

## **1.8. Постановка задачі**

### **1.8.1. Призначення та цілі створення системи**

Головною метою створення системи є зробити навчальний процес цінним і захоплюючим для дітей, які навчаються у 1-4 класах. У цей процес включаються такі засоби:

- Додання різних ілюстрацій, які допомагають створити веселий настрій та високий рівень енергії, а також допомагають підтримувати взаємодію, особливо з предметами, які містять багато вмісту;

- Заохочення учасників якомога більшою кількістю органів чуття. Також важливо запобігти монотонності;

- Створення зручної навігації, щоб користувачі могли легко переміщатися по програмі.

Головними завданнями системи є подання навчальних предметів, додаткової інформації, домашнього завдання, та відео. Також подальша перевірка набутих знань шляхом тестування.

Користувачами створюваної системи будуть школярі 1-4 класів.

Ми плануємо створити систему, яка буде містити відеоуроки, тести, винагороди та матеріали для домашніх завдань з 9 основних предметів: українська література, українська мова, основи здоров'я, математика, мистецтво, читання, природознавство, я і світ, та англійська мова.

Розроблювана система призначена для того, щоб учень, що вивчає різні предмети, мав більше можливостей для вивчення нового матеріалу. Та доступність до системи будь-де.

### **1.8.2 Вимоги до створюваної системи**

Особливістю реалізації функцій, які покладені на систему, є необхідність створення навчального процесу більш захоплюючим. Система повинна мати легку у використанні навігацію, та зручний інтерфейс.

Система має бути розрахована на роботу під ОС Windows. Тому що переглянувши статистику використання операційних систем, цією системою користуються більшість.

Засіб повинен забезпечити:

- зручність перегляду навчального матеріалу;
- скорочене подання матеріалу у програмі з можливістю більш детального його вивчення за допомогою основної навчальної або рекомендованої літератури;
- простий та зрозумілий інтерфейс для дітей;

- підвищення продуктивності навчання учнів, завдяки покращенню інтерфейсу та проходження тестів.

### **1.8.3 Функції, які повинна виконувати система**

Основними функціями системи повинні бути:

- Подання матеріалів у різних форматах (текст, картинки, відео, тести ілюстрації);
- Перевірка отриманих знань шляхом тестування.

### **1.8.4 Вхідні та вихідні дані системи**

Вхідними даними до системи є:

- Навчальний матеріал;
- Питання та відповіді тестів;
- Дані про користувачів;
- Перелік навчальної літератури.

Вихідними даними системи є:

- Закріпленні знання;
- Виконана тема, зданий тест.

## РОЗДІЛ 2. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

### 2.1. Технічне та системне забезпечення розробки.

При розробленні системи було вирішено обрати наступні технології. Для створення функціональних моделей було використано Brwin Data Modeler[18].

Формування інтерфейсу користувача відіграє важливу роль при реалізації функціонування системи. Для реалізації інтерфейсу було використано програмне середовище WebStorm та використання мови програмування JavaScript.

JavaScript [8] було створено для того, щоб робити веб-сторінки. Програми на цій мові називаються скриптами. Їх можна писати прямо на сторінці в коді HTML і вони автоматично виконуються при завантаженні сторінки.

Скрипти надаються та виконуються як простий текст. Для запуску їм не потрібна спеціальна підготовка чи компілятор.

Також використання React [5] спрощує створення інтерактивних інтерфейсів. Нам потрібно лише описати, як різні частини інтерфейсу виглядають у кожному стані системи і React ефективно оновить та відрендерить лише потрібні компоненти, коли ваші дані зміняться.

Декларативні інтерфейси роблять ваш код більш передбачуваним і його набагато легше налагоджувати.

Node.js, який теж використовується для розробки – це середовище виконання JavaScript з відкритим вихідним кодом, яке виконує код поза браузером.

WebStorm – інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, розроблена на основі платформи IntelliJ IDEA. WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей. постачається з перед-установленими плагінами JavaScript (такими як для Node.js), котрі доступні для PhpStorm безкоштовно.

WebStorm забезпечує автодоповнення, аналіз коду на льоту, навігацію по коду, рефакторинг, зневадження та інтеграцію з системами управління версіями. Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами (у тому числі, рефакторинг коду JavaScript, що міститься в різних файлах і теках проекту, а також вкладеного в HTML). Підтримується множинна

вкладеність (коли в документ на HTML вкладений скрипт на Javascript, в який вкладено інший код HTML, всередині якого вкладений Javascript) — в таких конструкціях підтримується коректний рефакторинг.

Material-UI – ReactJS's фреймворк, що надає готові google рішення для швидкої і досить простий web розробки.

Material-UI досить велика бібліотека, де ключовою частиною react компонентів і стилізації є @material-ui/core (метою її використання повинен бути немаленький проект).

## 2.2. Інформаційне забезпечення системи.

Основою майже будь якої програми нам потрібно зберігати якусь інформацію. Для цього у нас є база даних. База даних – це сукупність даних, яку можна сортувати, використовувати, змінювати залежно від використання. роботи декількох користувачів з цими базами даних використовують СУБД.

У даній роботі було використано базу даних Cloud Firestore [6]. Cloud Firestore – це документно-орієнтована база даних NoSQL. На відміну від бази даних SQL тут немає таблиць чи рядків. Натомість ви зберігаєте дані в документах, які організовані в колекції.

Кожен документ містить набір пар ключ-значення. Cloud Firestore оптимізовано для збереження великих колекцій невеликих документів.

Усі документи мають зберігатися у колекціях. Документи можуть містити вкладені колекції та вкладені об'єкти, які можуть включати примітивні поля, такі як рядки, або складні об'єкти, такі як списки.

Колекції та документи створюються неявно у Cloud Firestore. Просто призначте дані документу колекції. Якщо колекція або документ не існує, Cloud Firestore створює їх.

Вона має такі ключові можливості:

- гнучкість;
- виразні запити;
- оновлення у реальному часі;
- онлайн-підтримка;

- створений для масштабування.

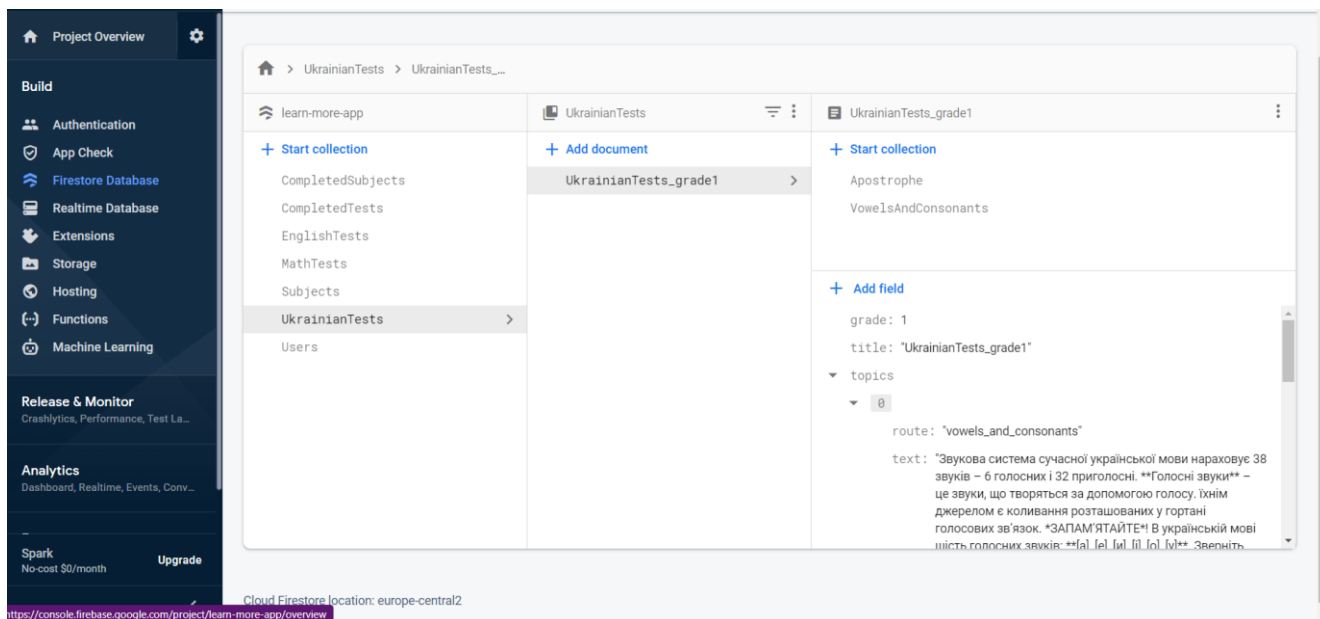


Рисунок 2.1 — Вигляд інтерфейсу та приклад структури Cloud Firestore

Структуру бази даних зображено на рисунку 2.2.

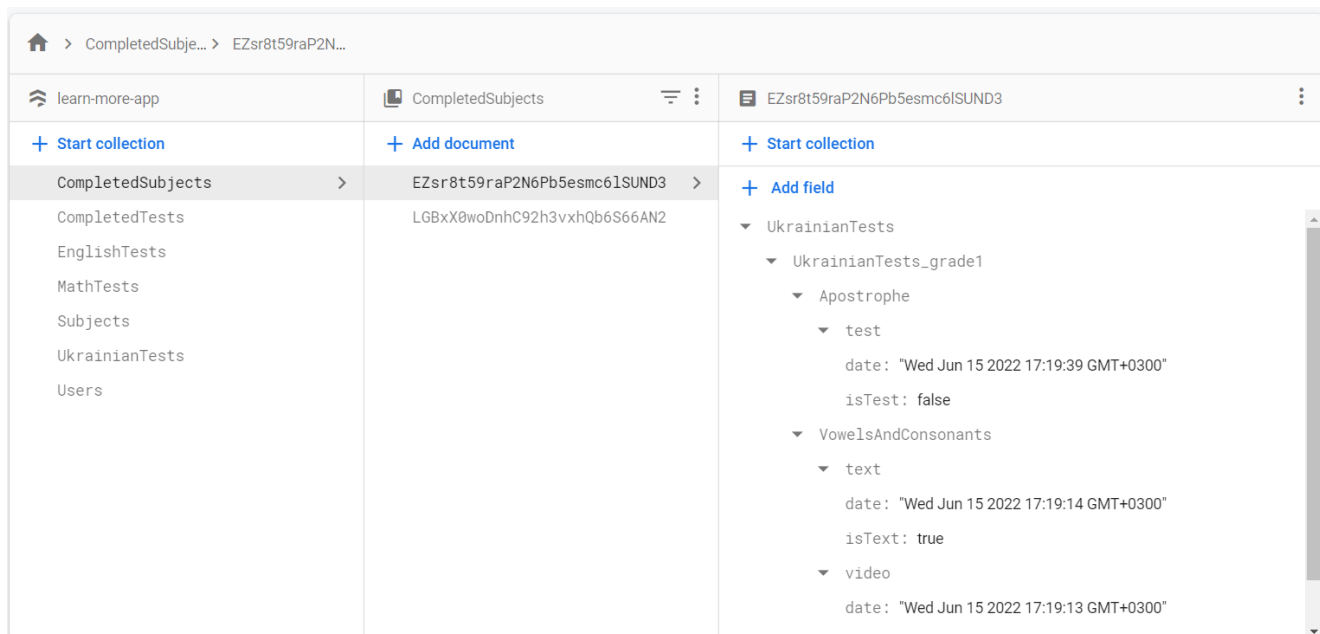


Рисунок 2.2 — Структура бази даних в Cloud Firestore

## Документи

У Cloud Firestore одиницею зберігання є документ. Документ — це спрощений запис, що містить поля, які порівнюються зі значеннями. Кожен документ ідентифікується ім'ям.

Документи, що представляють колекцію «CompletedSubjects», може виглядати так:

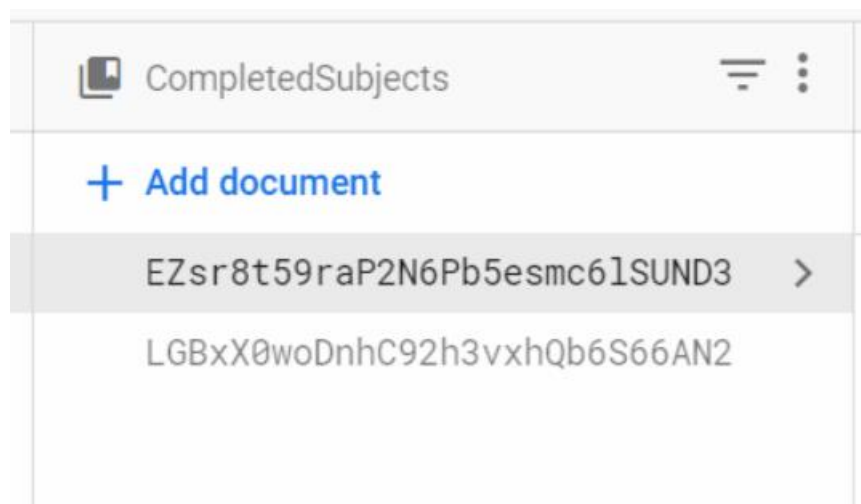


Рисунок 2.3 – Документи для колекції «CompletedSubjects»

Cloud Firestore підтримує різні типи даних для значень: логічні, числові, рядкові, географічні точки, двійкові об'єкти та часові позначки. Також можна використовувати масиви або вкладені об'єкти, які називаються картами, для структурування даних у документі

### **Колекції**

Документи мешкають у колекціях, які є просто контейнерами для документів. У нас є колекція «CompletedSubjects», що містить різні виконані предмети Рисунок 2.3.

Cloud Firestore не має схеми, тому у вас є повна свобода вибору полів, які ми додаємо до кожного документа, і типи даних, які ви зберігаєте у цих полях. Усі документи в одній колекції можуть містити різні поля або зберігати різні типи даних у цих полях.

Колекція містить документи та нічого більше. Вона не може безпосередньо містити необроблені поля зі значеннями та не може містити інші колекції.

Імена документів у колекції є унікальними. Можна надати власні ключі, такі як ідентифікатори користувачів, або дозволити Cloud Firestore автоматично створювати випадкові ідентифікатори.

Нам не потрібно «створювати» або видаляти колекції. Після створення першого документа колекції існує. Якщо видалити всі документи в колекції, вона буде видалена.

### 2.3. Алгоритмізація та реалізація комплексу задач автоматизації.

Для розробки додатків на мовах JavaScript, HTML&CSS добре підходить інтегроване середовище розробки WebStorm від компанії JetBrains.

Після запуску на виконання інтегрованого середовища відкриється вікно запрошення, приблизний вигляд якого зображений на рисунку 2.4.

Для створення нового проекту потрібно вибрати команду «New Project».

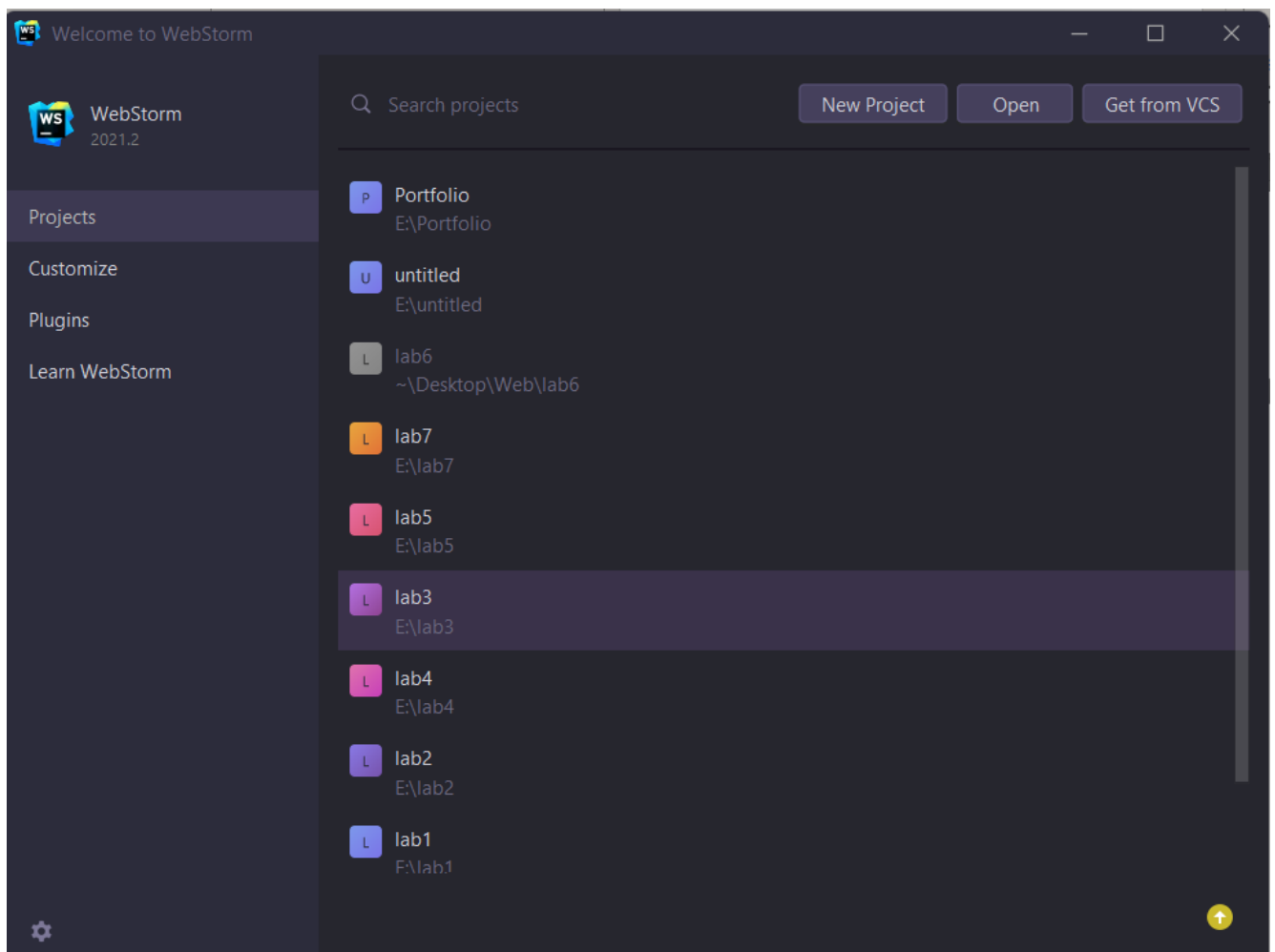


Рисунок 2.4 – Вікно «Welcome to WebStorm»

У результаті відкриється вікно «New Project», зображене на рисунку 3.5. Вибираємо React – проект на основі JavaScript-бібліотеки React з відкритим програмним кодом, яка використовується для створення інтерфейсів користувача у

великих веб-застосунках, які використовують дані, що змінюються з часом без перезавантаження сторінки.

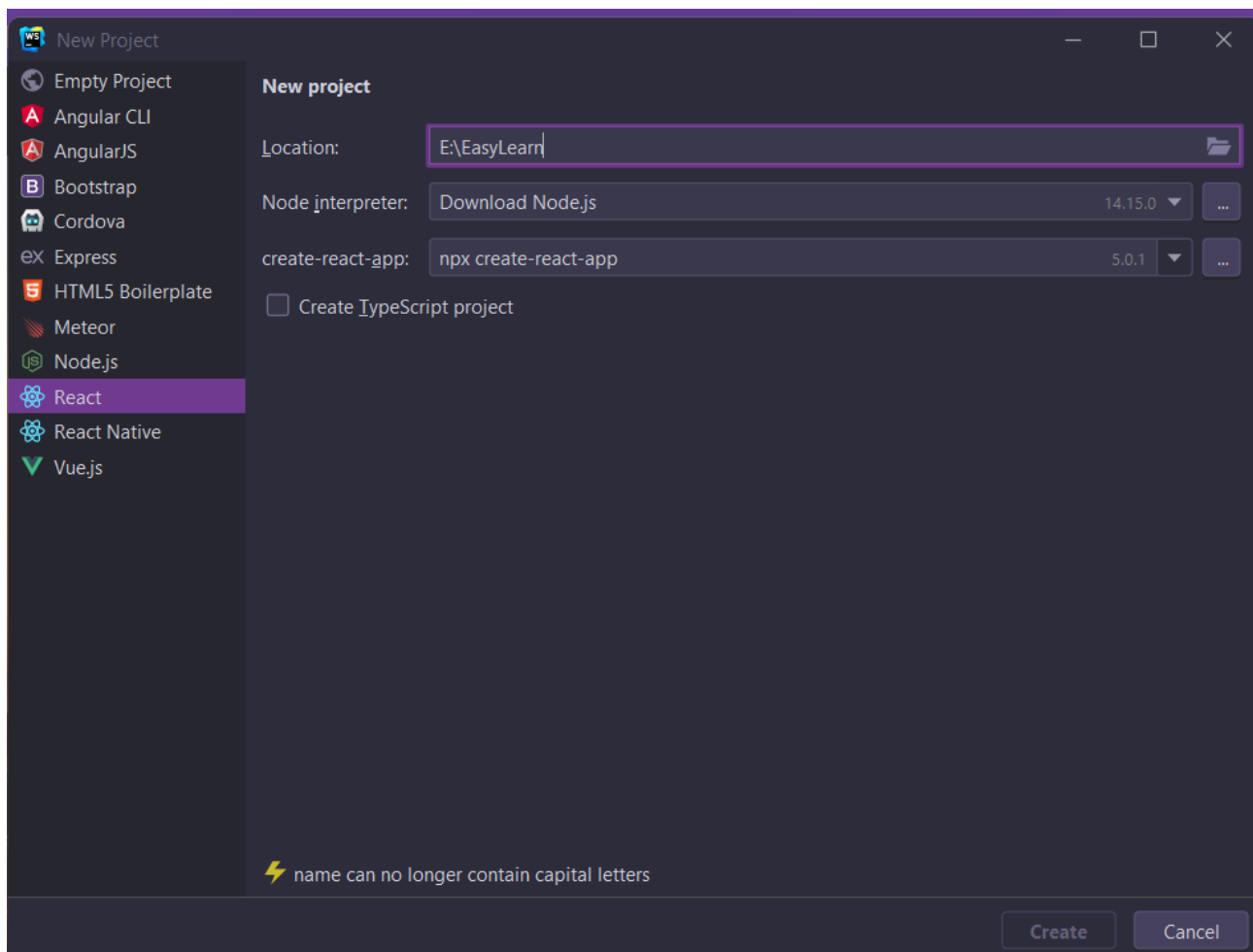


Рисунок 2.5 – Вікно вибору типу нового проекту

Після вибору кнопки «Create», відкриється вікно середовища WebStorm, як показано на рисунку 2.6.

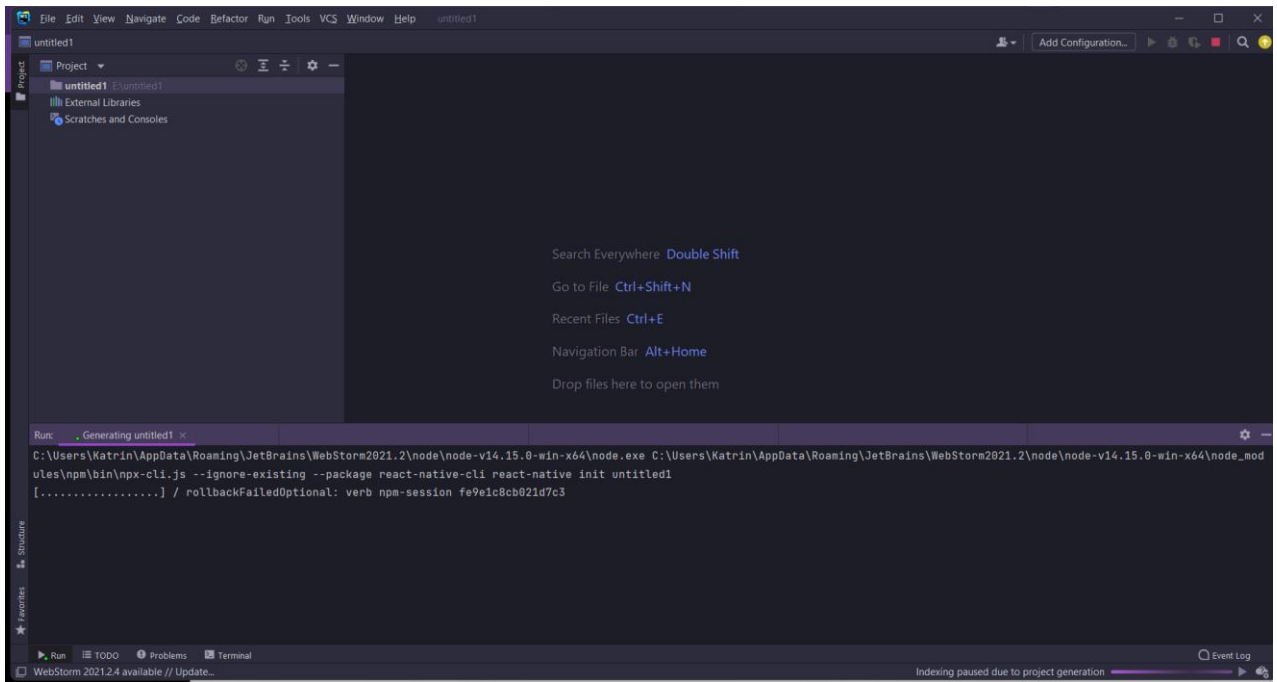


Рисунок 2.6 – Вікно інтегрованого середовища WebStorm

Далі ми підключаємо базу даних. Спочатку ми підключаємо бібліотеку для бази даних Firebase. Далі ми беремо конфігурації з Cloud Firestore та вставляємо їх у файл під назвою App.tsx. Використовуючи функції бібліотеки приєднуємо проект на платформі Firebase з системою.



Рисунок 2.7 – Підключення джерела даних

Далі ми створюємо сторінки для системи. Починаємо з реєстрації Рисунок 2.8. Реєстрація необхідна, щоб в залежності від того, ким є користувач, він отримав доступ до системи. Для цього був прописаний наступний код:

```

const SetUserDataInDB = useCallback( async (user) => {
    const userData = {
        uid: user.uid,
        first_name: user.first_name,
        email: user.email,
    };
    await setDoc(doc(db, "Users", user.uid), userData);
    await setDoc(doc(db, 'CompletedSubjects', user.uid), {uid: user.uid})
    dispatch(setUser(userData))
}, [])

const formik = useFormik<IRegister>({
    initialValues: {
        userName: '',
        email: '',
        password: '',
        confirmPassword: '',
    },
    validationSchema: Yup.object({
        userName: Yup.string().max(20, 'Must be 20 characters or
less').required('Required'),
        // password: Yup.string().uid()
    }),
    onSubmit: (values) => {
        if(values.confirmPassword === values.password) {
            createUserWithEmailAndPassword(auth, values.email,
values.password)
                .then((userCredential) => {
                    const user = userCredential.user;
                    SetUserDataInDB({uid: user.uid, first_name:
values.userName, email: values.email})
                    navigate(pageNames.home)
                })
                .catch((error) => {
                    const errorCode = error.code;
                    const errorMessage = error.message;
                });
        }
    }
});

```

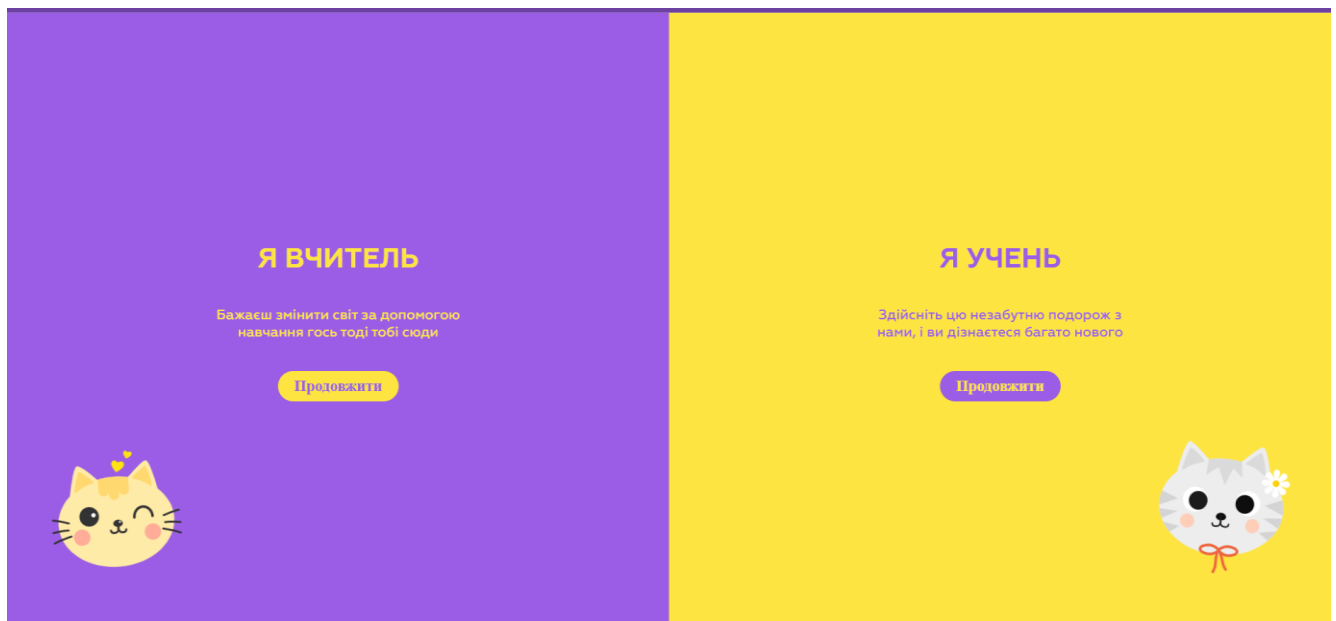


Рисунок 2.8 – Сторінка реєстрації

З пункту меню виберемо пункт предмети.

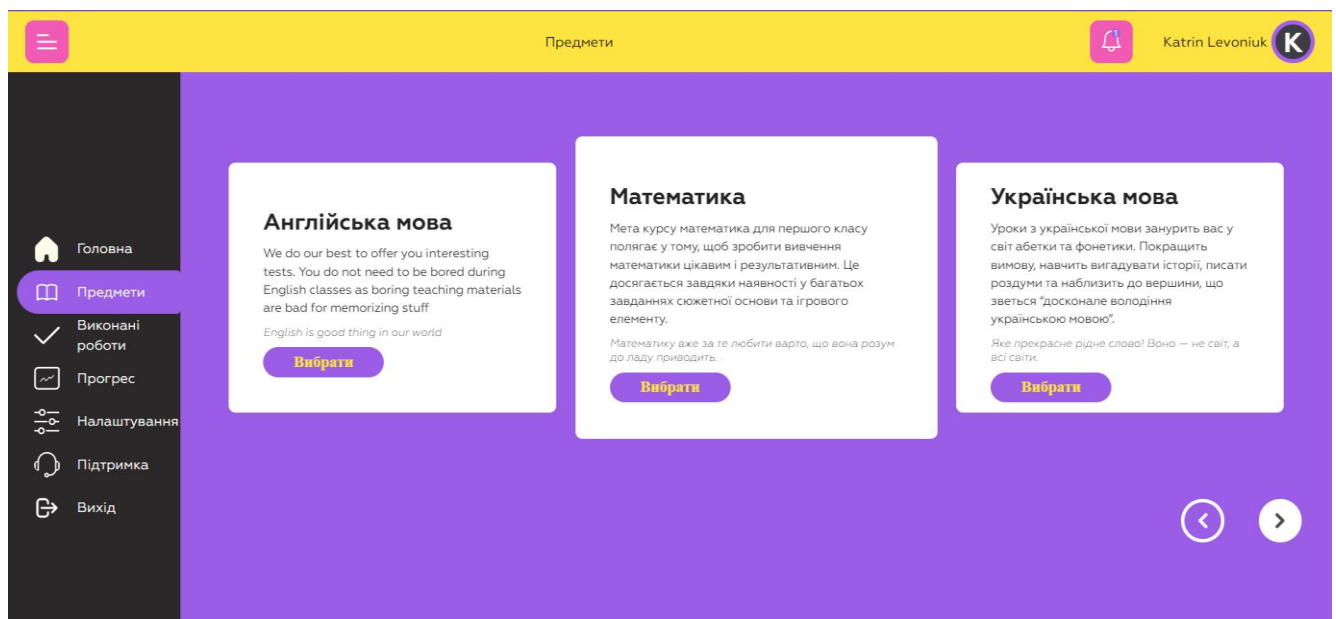


Рисунок 2.9 – Сторінка з предметами

Перехід по предметам представлений наступним кодом:

```
async function fetchData () {
  const querySnapshot = await getDocs(collection(db, 'Subjects'));
  let newData = querySnapshot.docs.map(doc => doc.data())
  setSubjects([...newData])
}

useEffect(() => {
  fetchData()
}, [])

const subjectsObject = useMemo(() => {
  return subjects?.map((subject) => <SubjectComponent
    title={subject.title}
    description={subject.description}
    phrase={subject.phrase}
    handleClick={() => {
      dispatch(setSubject(subject))
      navigate(pathes.topics)
    }}
  />)
}, [subjects])
```

Після вибору предмету нам потрібно вибрати предмети Рисунок 2.11:

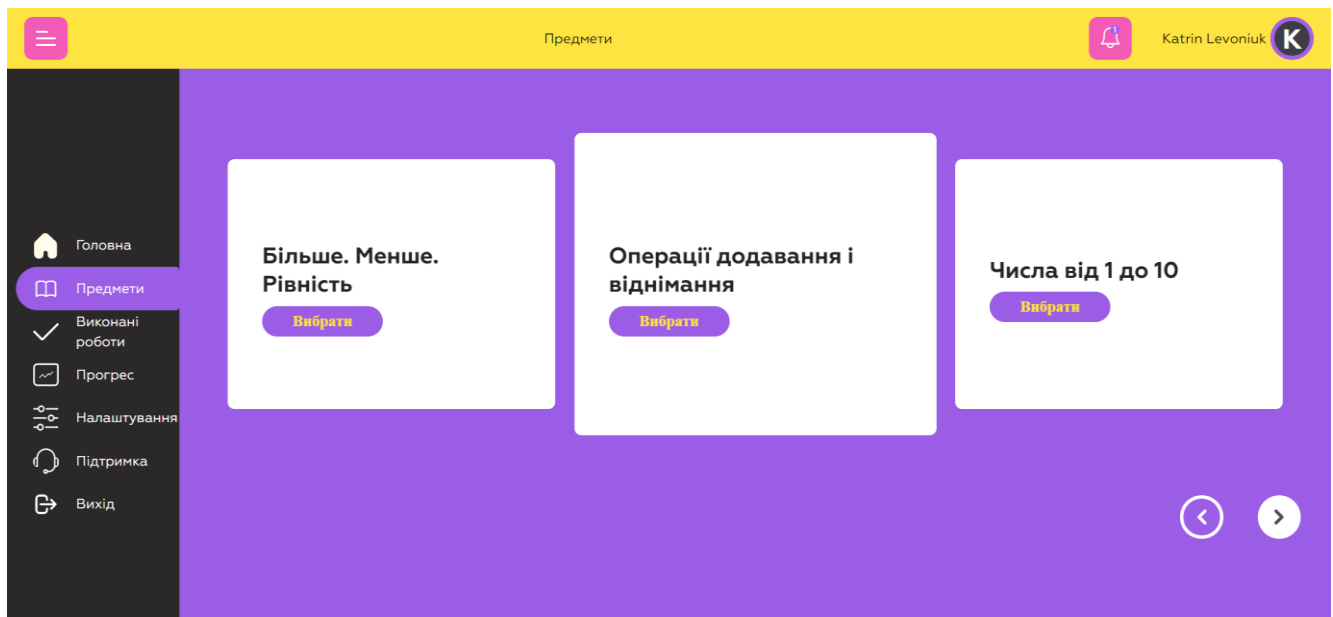


Рисунок 2.10 – Сторінка з темами

Перехід по темам представлений наступним кодом:

```
    async function fetchData () {
      const SubjectTopics = query(collectionGroup(db,
subject.topics_link))
      const querySnapshot = await getDocs(SubjectTopics);
      let newData = querySnapshot.docs.map(doc => doc.data())
      setGradeTopics(...newData)
    }

    useEffect(() => {
      fetchData()
    }, [])

    const topicsObject = useMemo(() => {
      return gradeTopics?.topics?.map((topic) => {
        if(topic.isHidden) {
          return (<SubjectComponent
            isHidden
          />)
        }
        return (<SubjectComponent
          title={topic.title}
          handleClick={() => {
            dispatch(setTopic({grade: gradeTopics.title,
gradeNumber: gradeTopics.grade, ...topic}))
            navigate(pathes.topic.home)
          }}
        />)
      }).reverse()
    }, [gradeTopics])
```

Після цього нас перенаправляє на нове меню, де дитина може вибрати що хоче робити: дивитись відео, читати, здати тест або виконати домашню роботу. Ми будем акцентувати увагу на тестах. Сторінка тесту:

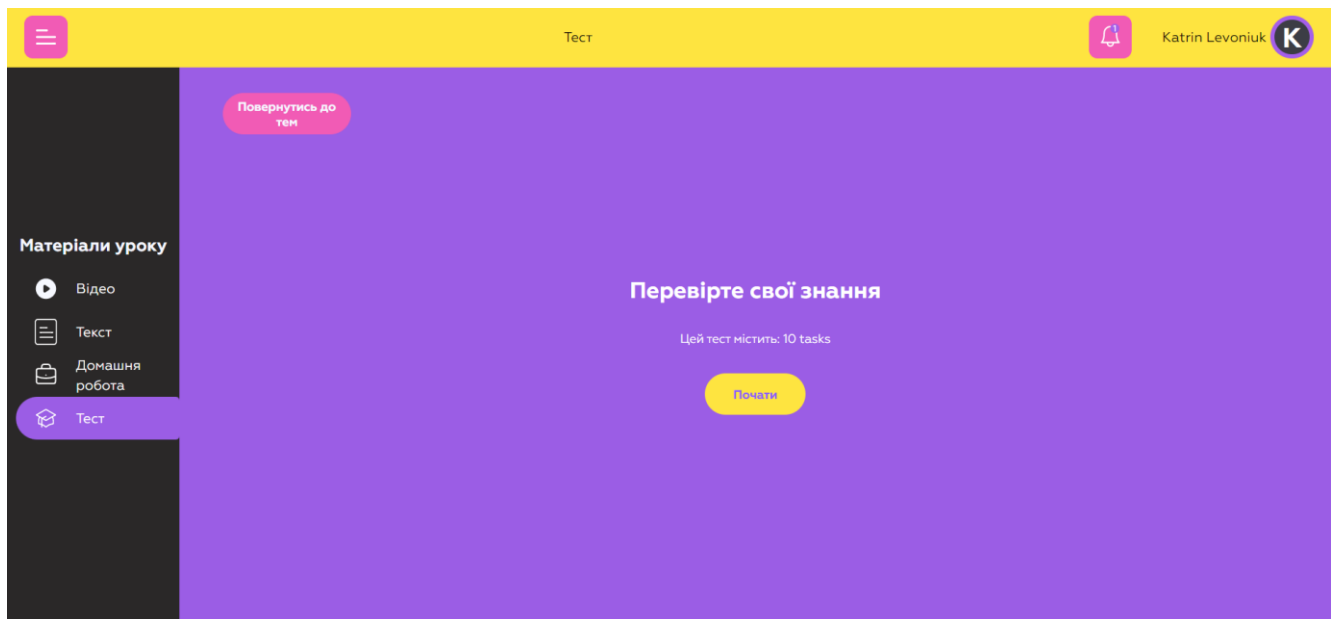


Рисунок 2.11 – Сторінка тесту

Проходження тесту представлений наступним кодом:

```
useEffect(() => {
    fetchData()
}, [])

async function fetchData () {
    const testTopic = query(collectionGroup(db, topic.topic_link))
    const querySnapshot = await getDocs(testTopic);
    let newData = querySnapshot.docs.map(doc => doc.data())
    setTestsFromDB(newData)
}

const onStartTest = useCallback(() => {
    dispatch(startTest(testsFromDB))
}, [testsFromDB])

const matchAnswers = (tests: any, nextIndex: number): any => {
    for (let i=0; i < answers.length; i++) {
        if (answers[i]?.id === tests[nextIndex]?.id &&
            (answers[i]?.question === tests[nextIndex]?.question)) {
            return matchAnswers(tests, nextIndex + 1)
        };
        continue;
    }
}
```

```

const onGiveAnswer = useCallback(() => {
  dispatch(setAnswers([...answers, answer]))
  const userAnswers = [...answers, answer].map((answer: any) => {
    const testRealAnswer = tests.find((test: any) => test.id ===
answer.id);
    return {
      ...answer,
      isCorrectAnswer:          answer?.answer          ===
testRealAnswer?.right_answer
    }
  });
  let countRightAnswers = 0;
  userAnswers.forEach((answer: any) => {
    if(answer?.isCorrectAnswer) {
      countRightAnswers++;
    }
  })
  addCompletedTests(userAnswers, countRightAnswers);
  CompletedWorks(userAnswers, countRightAnswers);
  dispatch(completeTest({results: userAnswers, countRightAnswers}));
}, [answers, tests, answer])

```

```

async function CompletedWorks(userAnswer, countRightAnswers) {
  const myProgressRef = doc(db, 'CompletedSubjects', user.uid);
  const myProgressObject = await getDoc(myProgressRef);
  const myProgressObjectData = myProgressObject.data()
  const progressData = myProgressObjectData ? myProgressObjectData : {};
  const      subjectData      =      myProgressObjectData      &&
Object.keys(myProgressObjectData).includes(subject.topics_link)      ?
myProgressObjectData[subject.topics_link] : {};
  const      gradeData      =      myProgressObjectData      &&
Object.keys(subjectData)?.includes(`${subject.topics_link}_grade${1})      ?
myProgressObjectData[subject.topics_link][`${subject.topics_link}_grade${1}]
: {};
  const      topicData      =      myProgressObjectData      &&
Object.keys(gradeData)?.includes(topic.topic_link)      ?
myProgressObjectData[subject.topics_link][`${subject.topics_link}_grade${1}][
topic.topic_link] : {};
  let isCompletedTest = topicData?.test?.isTest
  if(!isCompletedTest) {
    await setDoc(myProgressRef, {

```

```

        ...progressData,
        [`${subject.topics_link}`]: {
            ...subjectData,
            [`${topic?.grade}`]: {
                ...gradeData,
                [`${topic?.topic_link}`]: {
                    ...topicData,
                    test: {
                        isTest:          countRightAnswers      ===
userAnswer.length ? true : false,
                        date: moment(new Date()).toString()
                    }
                }
            }
        }
    }, {merge: true});
}
toast.success('Записав твої результати')
}

const addCompletedTests = async (answers: any, countRightAnswers: any) =>
{
    await addDoc(collection(db, "CompletedTests"), {
        uid: user?.uid,
        subject: subject?.topics_link,
        subject_route: subject?.route,
        subject_title: subject?.title,
        topic: topic?.topic_link,
        topic_route: topic?.route,
        topic_title: topic?.title,
        answers: answers,
        count_right_answers: countRightAnswers,
        count_tests: answers?.length,
        date: moment(new Date()).toString(),
    });
}

const onNextStep = useCallback(() => {
    if(answer) {
        if (answers?.length && (answers?.find((item: any) => item?.id ===
answer?.id && (item?.question === answer?.question  item?.answers[0] ===
answer?.answers[0]))) {
            let newAnswers = answers?.map((item: any) => {

```

```

        if (item?.id === answer?.id && (item?.question ===
answer?.question || item?.answers[0] === answer?.answers[0])) {
            return answer;
        } else {
            return item;
        }
    })
    dispatch(setAnswers([...newAnswers]))
} else {
    let newAnswers = answers?.length ? [...answers, answer] :
[answer];
    dispatch(setAnswers([...newAnswers]))
}
setAnswer(() => null)
currentTest < tests?.length-1 &&
dispatch(setCurrentTest(currentTest + 1))
}, [currentTest, answer, answers, tests])

const onPreviousStep = useCallback(() => {
    setAnswer(null)
    dispatch(setCurrentTest(currentTest-1))
}, [currentTest])

const onCloseTest = useCallback(() => {
    setAnswer(null)
    dispatch(closeTest())
}, [])

```

## 2.4. Інструкція користувача.

Після запуску програми одразу з'являється вікно авторизації, яка забезпечує доступ до тих або інших функцій, в залежності від того, хто є користувач – студент або викладач.

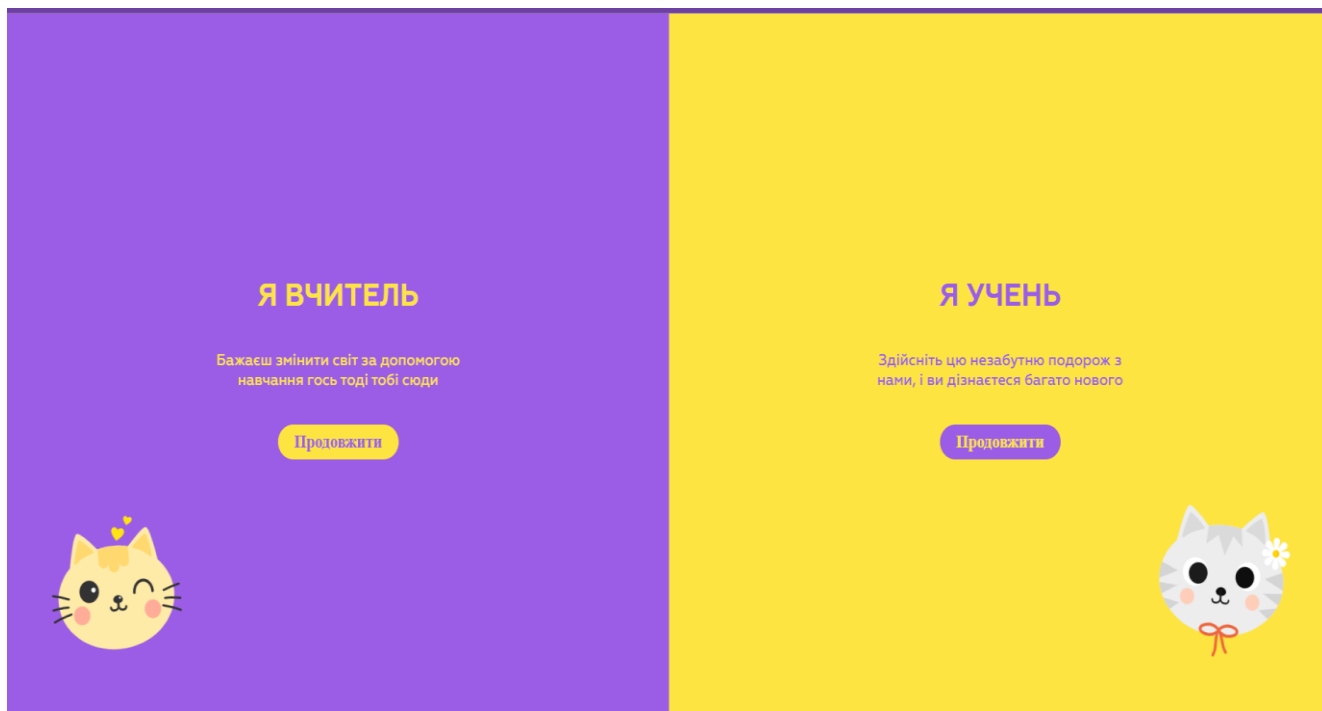


Рисунок 2.12 – Сторінка авторизації

Після авторизації як студент, йому буде відкрите таку сторінку:

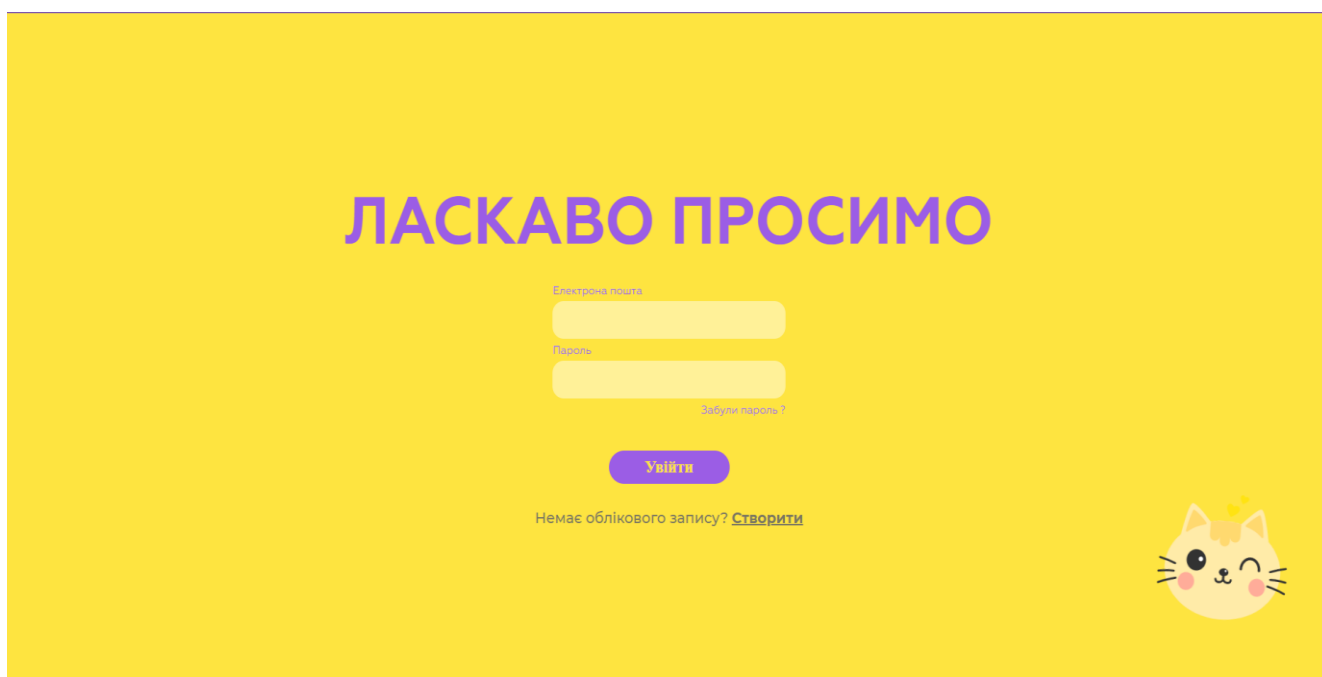


Рисунок 2.13 – Сторінка логізації для учня

Після авторизації як учитель, йому буде відкрите таку сторінку:

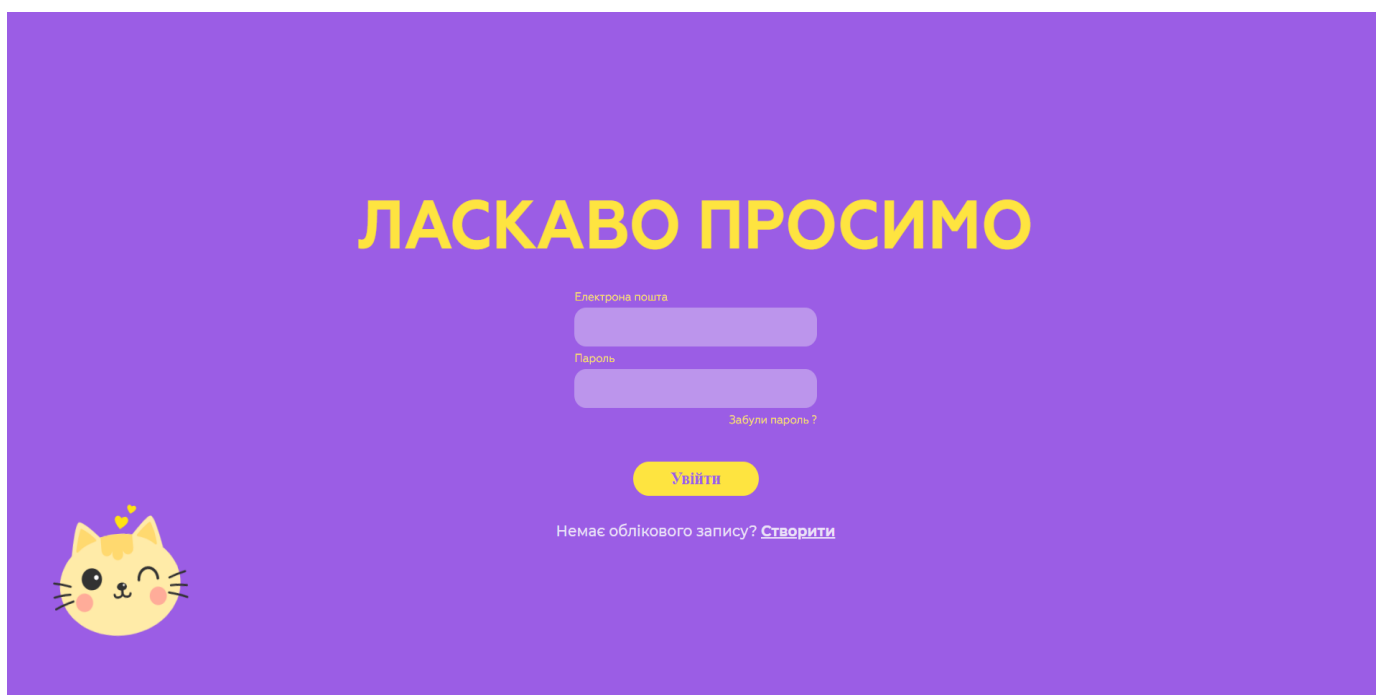


Рисунок 2.14 – Сторінка логінізації для учителя

Ми будемо розглядати проходження системи як студент.

Після реєстрації нам буде показано головне вікно яке зображено на рисунку 2.16.

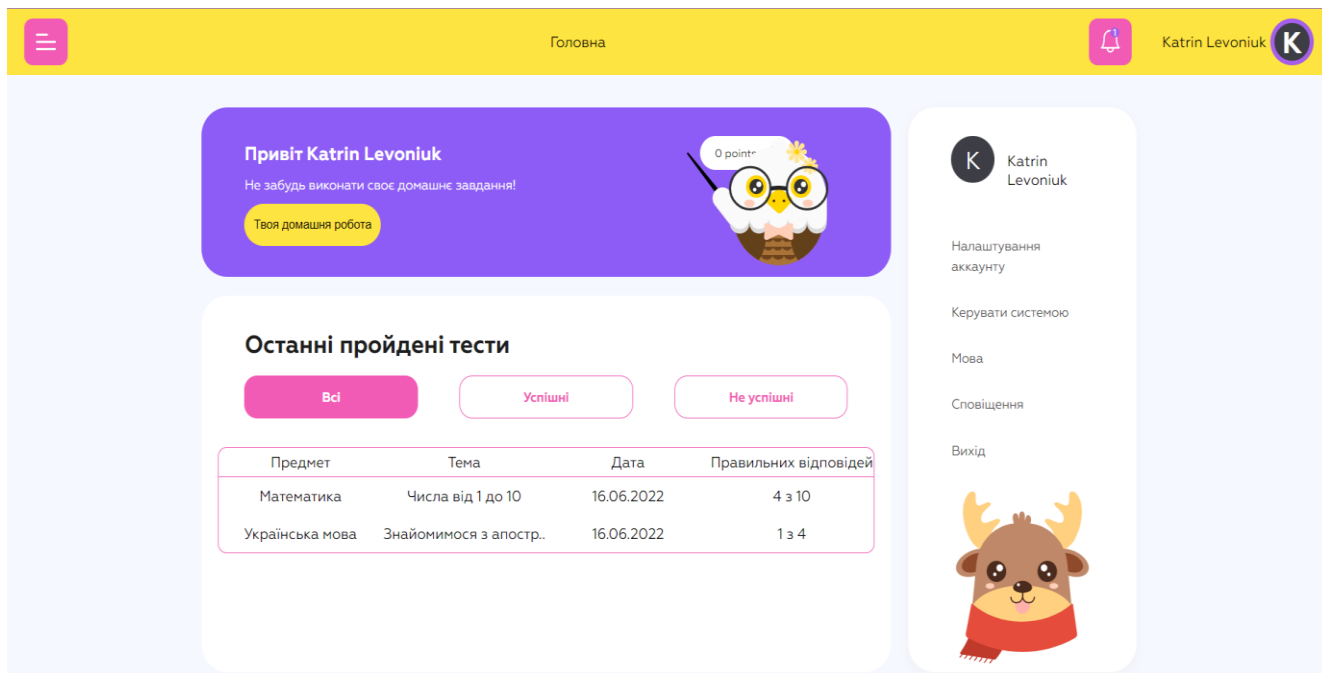


Рисунок 2.15 – Домашня сторінка

З пункту меню вибираємо пункт предмети, щоб вибати що саме сьогодні буде дитина вчити. Для того, щоб вибрати предмет потрібно натиснути на кнопку «Перейти на уроку».

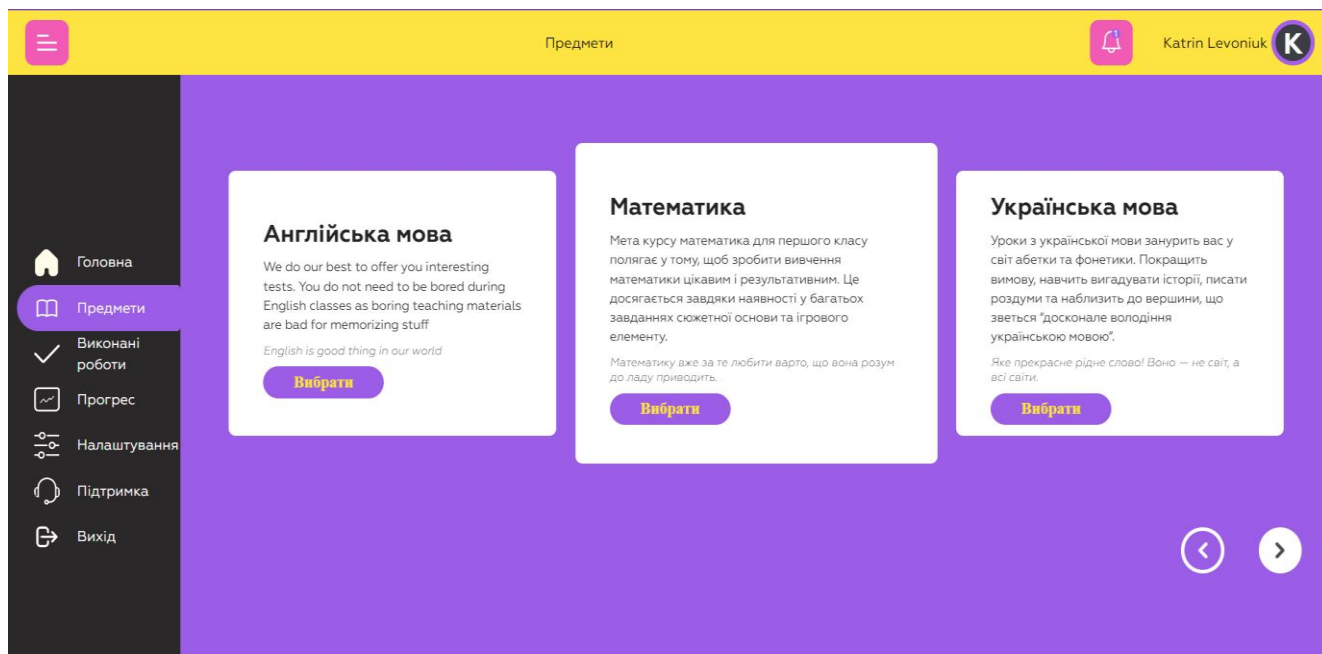


Рисунок 2.16 – Сторінка з предметами

Далі у нас покаже перелік тем, які є у предмета. Для того, щоб перейти на ну чи іншу тему натискаємо на кнопку.

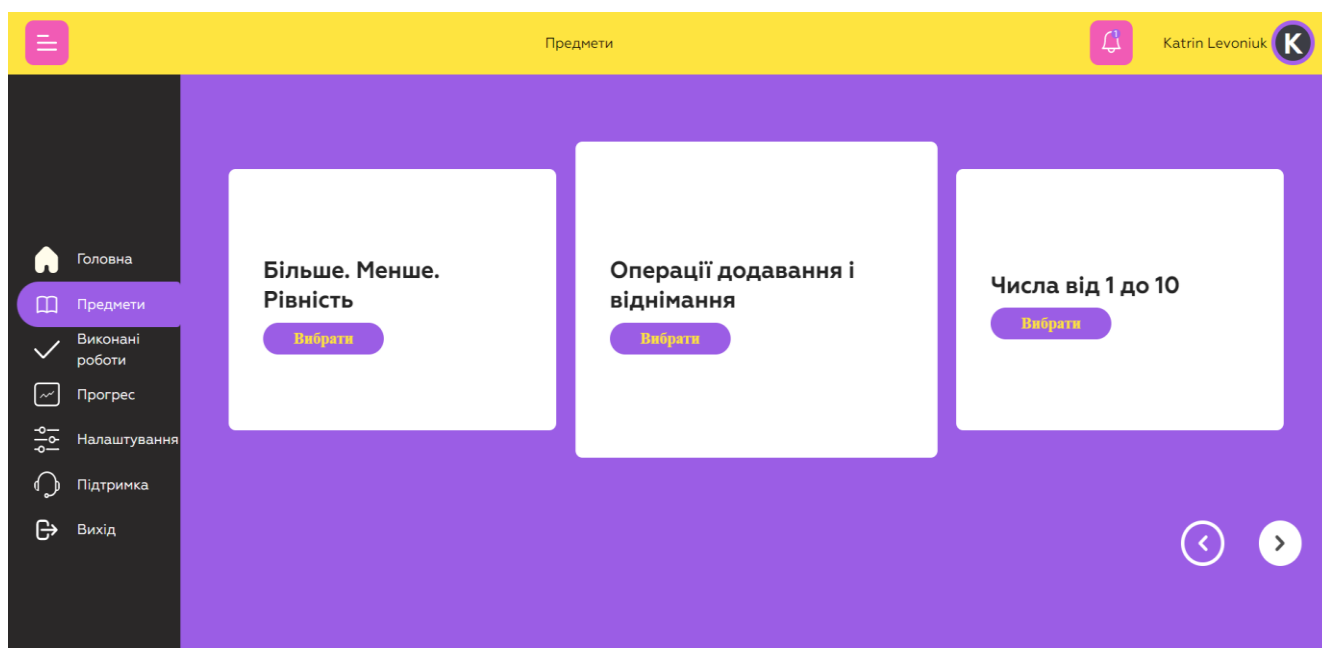


Рисунок 2.17 – Сторінка з темами

Після цього нас перенаправляє на нове меню, де дитина може вибрати що хоче робити: дивитись відео (Рисунок 2.19), читати(Рисунок 2.20), здати тест(Рисунок 2.22) або виконати домашню роботу(Рисунок 2.21). Представлення цих сторінок показано далі:

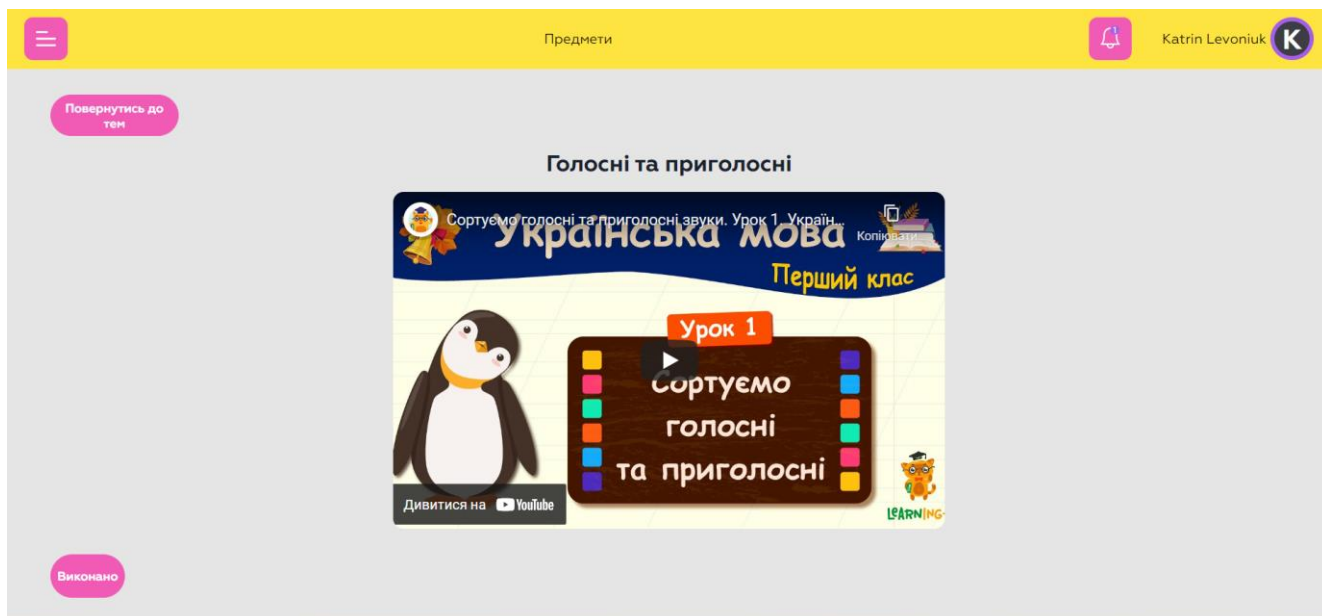


Рисунок 2.18 – Сторінка з відео

На цій сторінці ми можемо переглянути відео по темі. Та перейти на наступну сторінку.

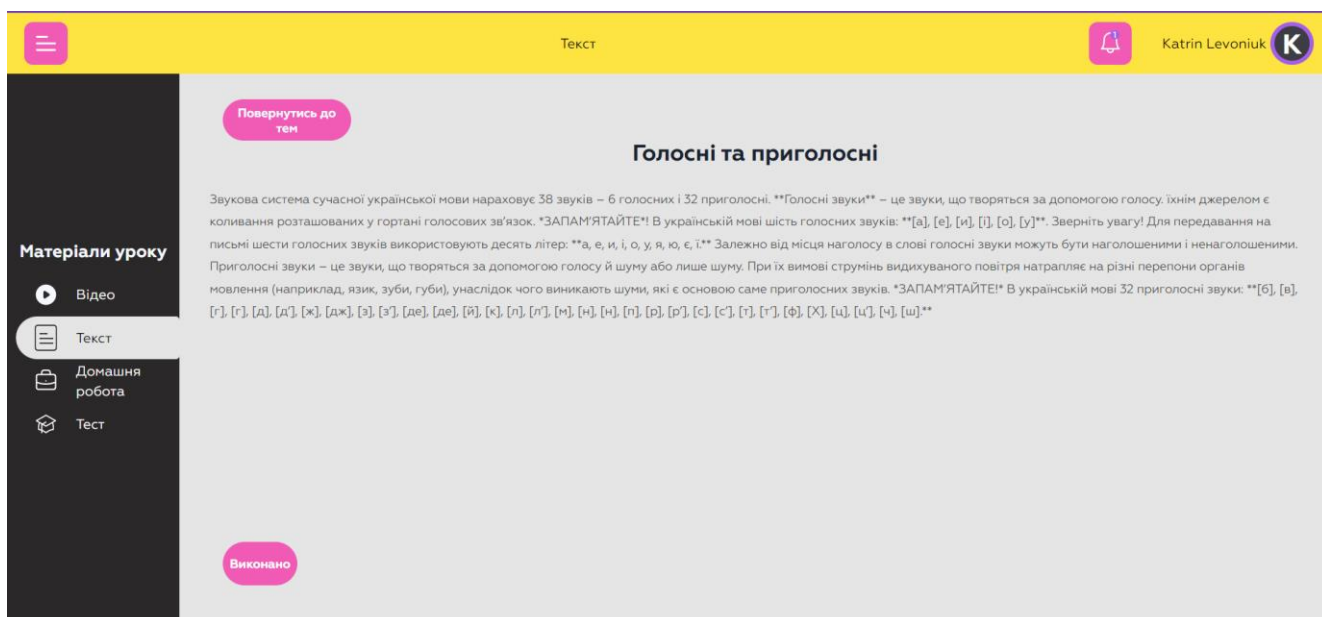


Рисунок 2.19 – Сторінка з додатковою інформацією

На сторінці з додатковою інформацією можна перечитати всю інформацію та відмітити як виконане завдання.

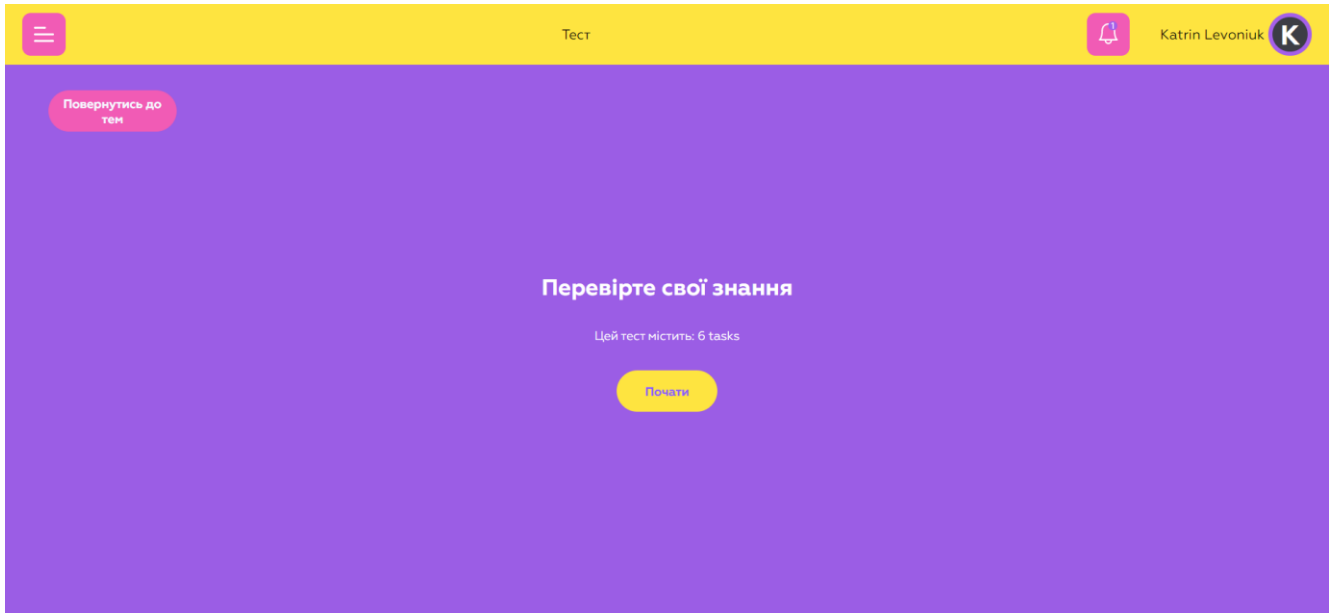


Рисунок 2.20 – Сторінка з інформацією про тест

Сторінка про тест несе загальну інформацію про тест.

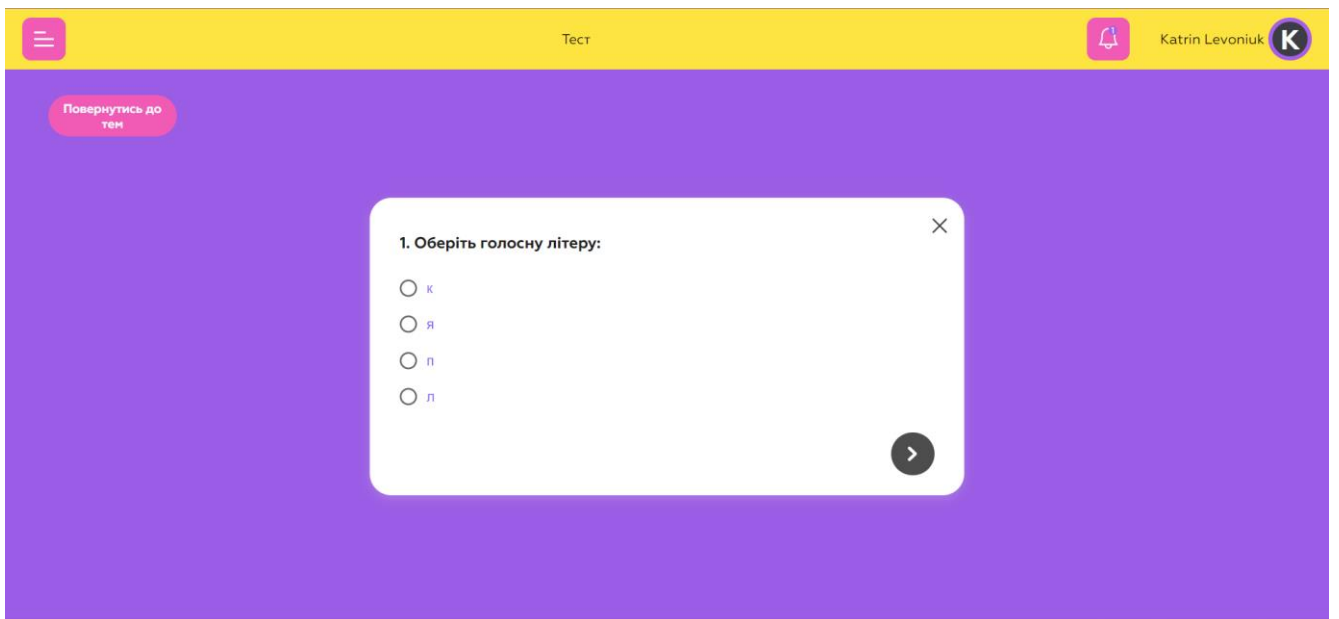


Рисунок 2.21 – Сторінка тесту

Сторінка тесту це сторінка, де можна вибрати відповіді на питання переходити до наступних питань та закінчити тест.

У нас є переглянуте відео, прочитана додаткова інформація та зданий тест.

## ВИСНОВКИ

У дипломній роботі проведено дослідження та детально проаналізовано учбовий процес молодшої школи. Після аналізу були знайдені завдання, які необхідно було вирішити за допомогою автоматизації. Саме для цих завдань було вирішено створити систему.

Після проведеного дослідження було створено функціональні моделі цієї системи (AS-IS та TO-BE), щоб наявно побачити різницю процесу навчання у школі та визначено задачі, які потрібно автоматизувати.

В результаті аналізу прийнято рішення про розроблення системи для навчання, головною метою якої є зробити навчальний процес цінним і захоплюючим для дітей, які навчаються у 1-4 класах за допомогою різних засобів.

При порівнянні аналогічних систем були враховані усі функції та недоліки розглянутих систем, що були внесені до спеціальної таблиці.

Для реалізації інтерфейсу було застосовано програмне середовище WebStorm, мова програмування JavaScript та бібліотека React.

Розроблена інформаційна система дозволяє дітям навчатися та вивчати тему, яку пропустили у школі через хворобу чи інші причини. Та завдяки ретельному дизайну досягати сильного рівня залучення, що сприяє інтенсивному навчанню у віртуальній обстановці.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання дипломних робіт для студентів бакалаврів напрямку 060501 «Комп'ютерні науки» денної та заочної форм навчання / Уклад. В.В. Самсонов, Л.Ю. Маноха, Т.М. Горлова, Л.Г. Загоровська, О.М. М'якшило, О.А Хлобистова – К.: НУХТ, 2010;

3. Управління ІТ проектами [Електронний ресурс]: лабораторний практикум для студентів напрямку підготовки 6.050101 "Комп'ютерні науки" денної та заочної форм навч. / уклад. О. А. Хлобистова, М. В. Гладка. - К.: НУХТ, 2013. – 108 с.

4. Методичні вказівки до виконання кваліфікаційної бакалаврської роботи для студентів за напрямом підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навчання / Уклад.: В.В. Самсонов, Л.Ю. Маноха, Т.М. Горлова, Л.Г. Загоровська, О.М. М'якшило, О.А Хлобистова. – К.: НУХТ, 2011. – 15с.

5. React [Електрон. ресурс]. - Режим доступу : <https://careers.epam.ua/events/react-native-retraining-01022022>

6. Cloud Firestore [Електрон. ресурс]. - Режим доступу : <https://firebase.google.com/docs/firestore/quickstart/>

7. What Is the Typical Organizational Structure of a School? - [Електрон. ресурс]. - Режим доступу : <https://www.smartcapitalmind.com/what-is-the-typical-organizational-structure-of-a-school.htm>

8. JavaScript [Електрон. ресурс]. - Режим доступу : <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>

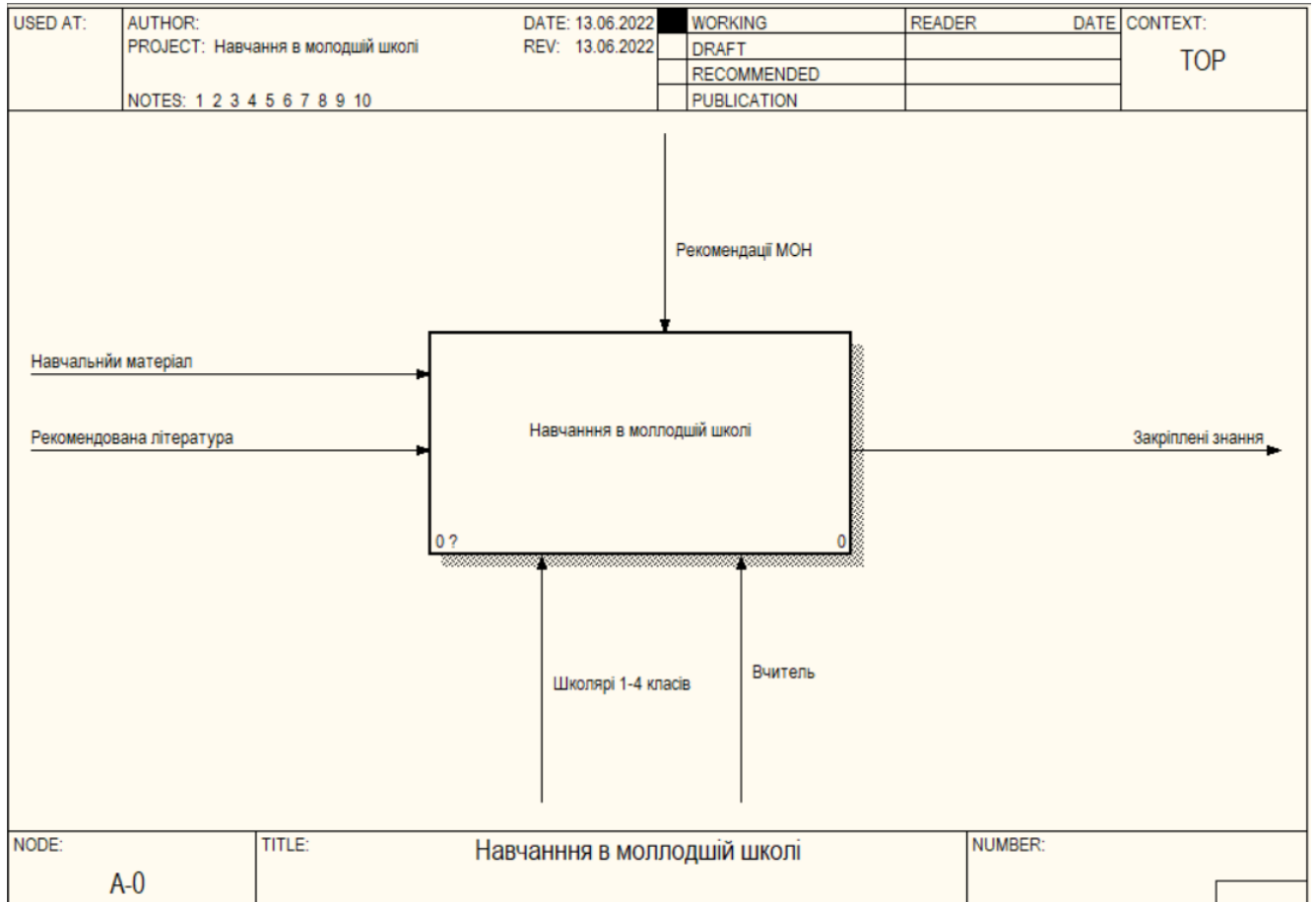
9. Рекомендації студентам щодо активного навчання в умовах онлайн-освіти: навч.-метод. посіб. / Базилен - ко А.К., Іванова І.Б., Кириленко В.Г., Кондукоцо - ва Н.В. — К.: Університет «Україна», 2020. — 85 с.

10. Всеукраїнська школа онлайн [Електрон. ресурс]. - Режим доступу : <https://lms.e-school.net.ua/>

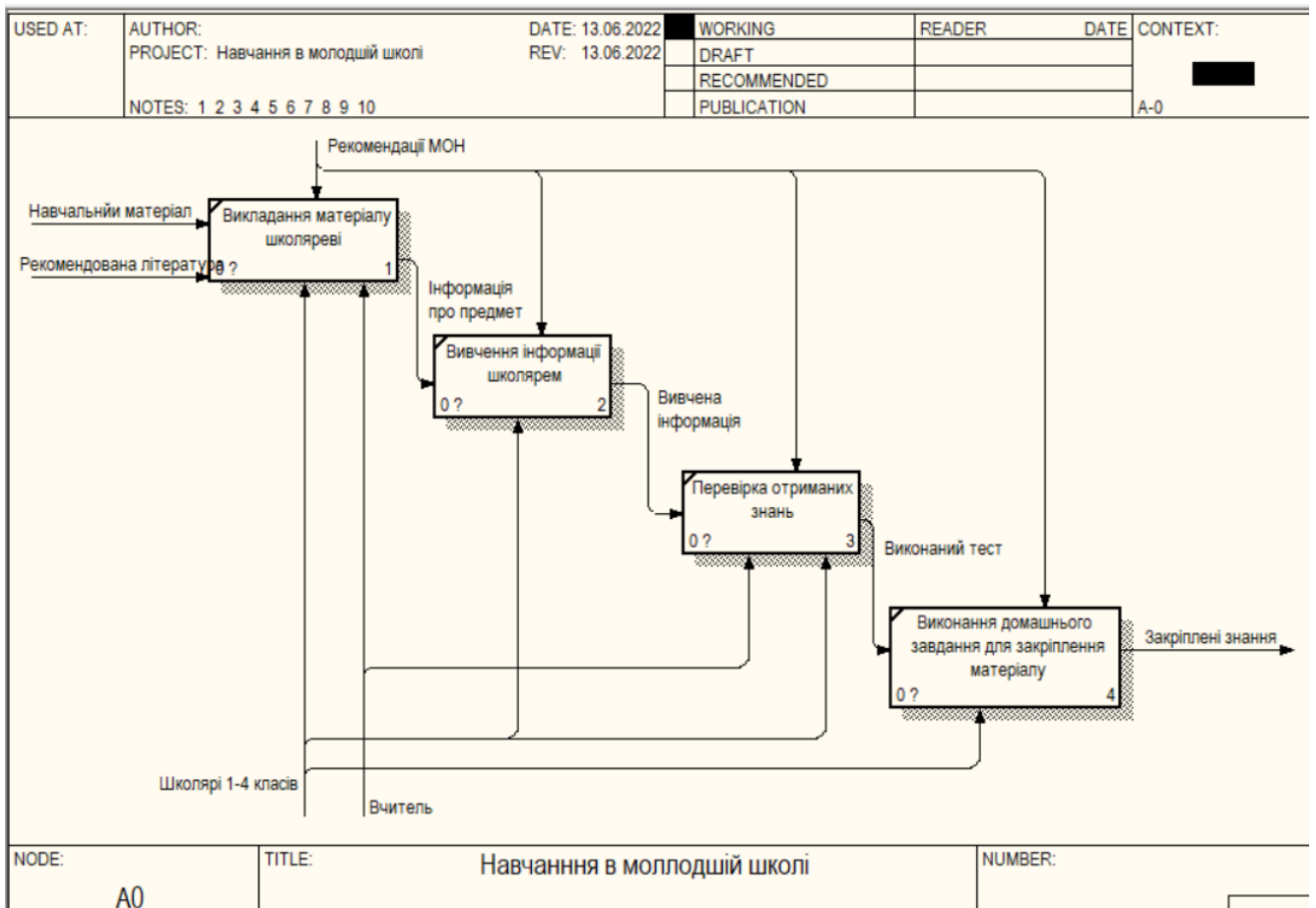
11. На урок [Електрон. ресурс]. - Режим доступу : <https://naurok.com.ua/page/about>
12. Атмосферна школа [Електрон. ресурс]. - Режим доступу : <https://www.atschool.com.ua/>
13. Організаційна структура школи [Електрон. ресурс]. - Режим доступу : <http://prosvitcenter.org/organizaciyna-structura/>
14. Computer and Internet Facilities Use in Distance Education: A Survey of Sandwich Students of University of Ado-Ekiti, Nigeria [Електрон. ресурс]. - Режим доступу: <https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1470&context=libphilprac>
15. Як організувати дистанційне навчання для дітей з ООП [Електрон. ресурс]. - Режим доступу : <https://nus.org.ua/articles/yak-organizuvaty-dystantsijne-navchannya-dlya-ditej-z-oor-dosvid-vchyteliv/>
16. Everyone is within learning distance: Building skills remotely [Електрон. ресурс]. - Режим доступу: <https://www.mckinsey.com/business-functions/operations/our-insights/everyone-is-within-learning-distance-building-skills-remotely>
17. Права та обов'язки учнів [Електрон. ресурс]. - Режим доступу: <http://coh.rv.ua/prava-ta-obov-yazky-uchniv/>
18. Проектування інформаційних систем. [Електронний ресурс]: лабораторний практикум для студ. освітнього ступеню "бакалавр" спец. 122 "Комп'ютерні науки " денної і заочної форм навчання. Частина 2 "Проектування клієнтського додатку" / Уклад.: О.М. М'якшило, О.В. Харкянен – К.: НУХТ, 2017 – 33 с.

## ДОДАТКИ

### ДОДАТОК А «ФУНКЦІОНАЛЬНА МОДЕЛЬ (AS-IS)»

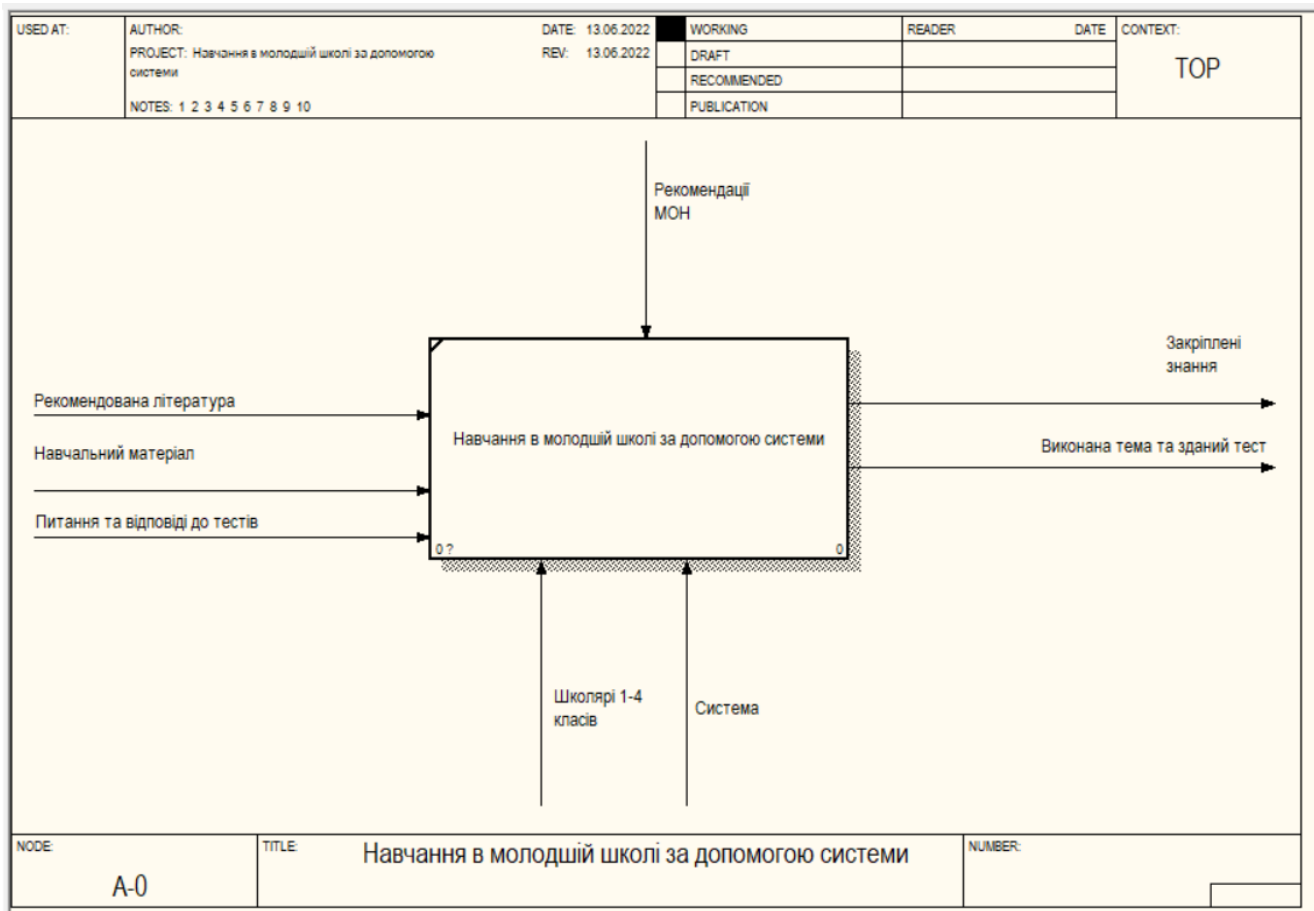


*Рис. А.1. Функціональна схема моделі AS-IS.*



*Рис. А.2. Перший рівень декомпозиції AS-IS.*

## ДОДАТОК Б «ФУНКЦІОНАЛЬНА МОДЕЛЬ (ТО-ВЕ)»



*Рис.Б.1 Функціональна модель ТО-ВЕ.*

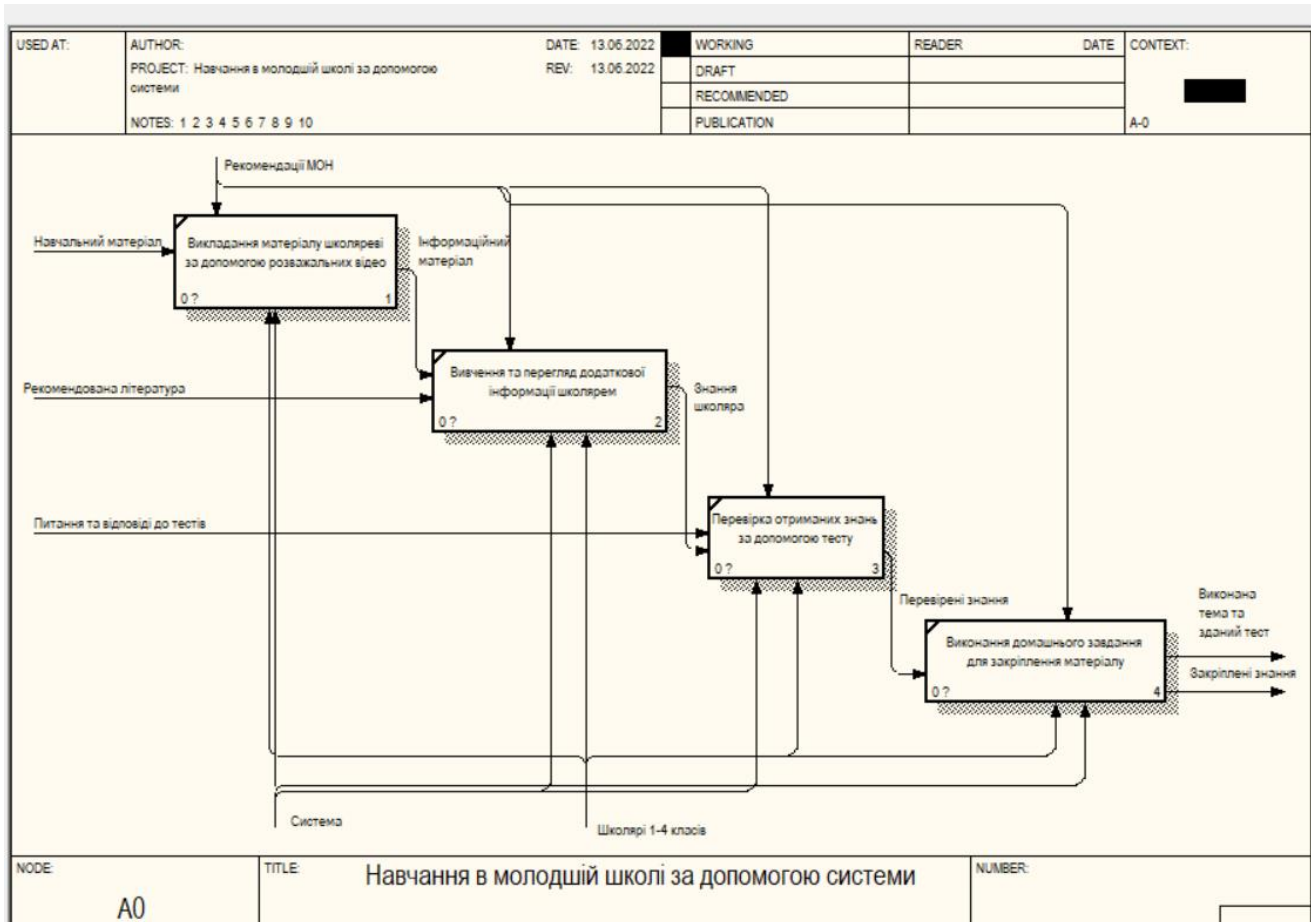


Рис.Б.2 Перший рівень декомпозиції діаграми ТО-ВЕ.

## ДОДАТОК В «СТРУКТУРА БАЗИ ДАНИХ NO SQL»

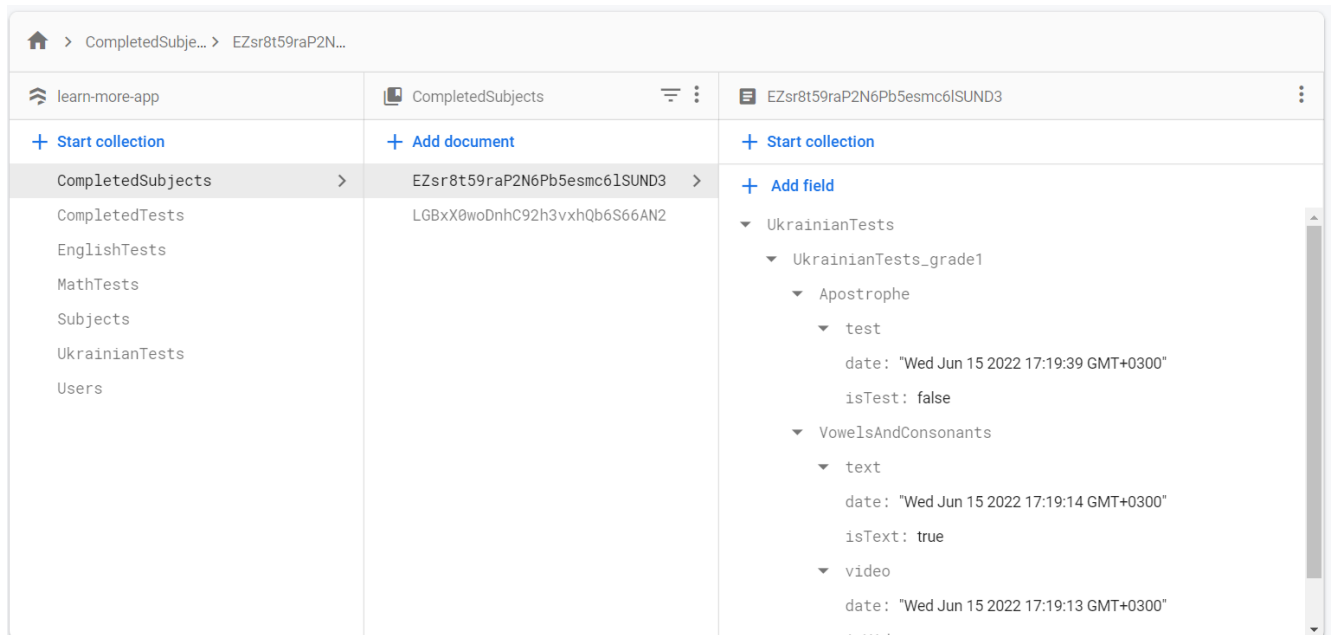


Рисунок В 1 – Структура бази даних

## ДОДАТОК Г «ЗНІМКИ ЕКРАНУ ПРОГРАМИ»

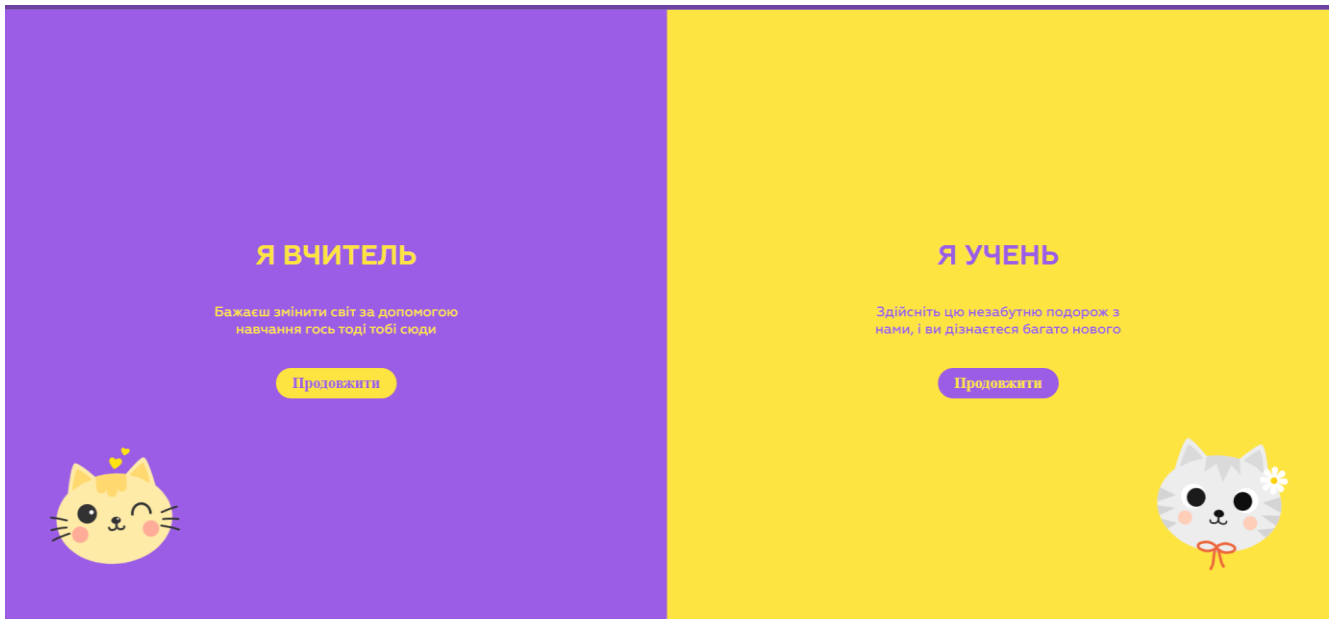


Рисунок Г.1 – Сторінка авторизації

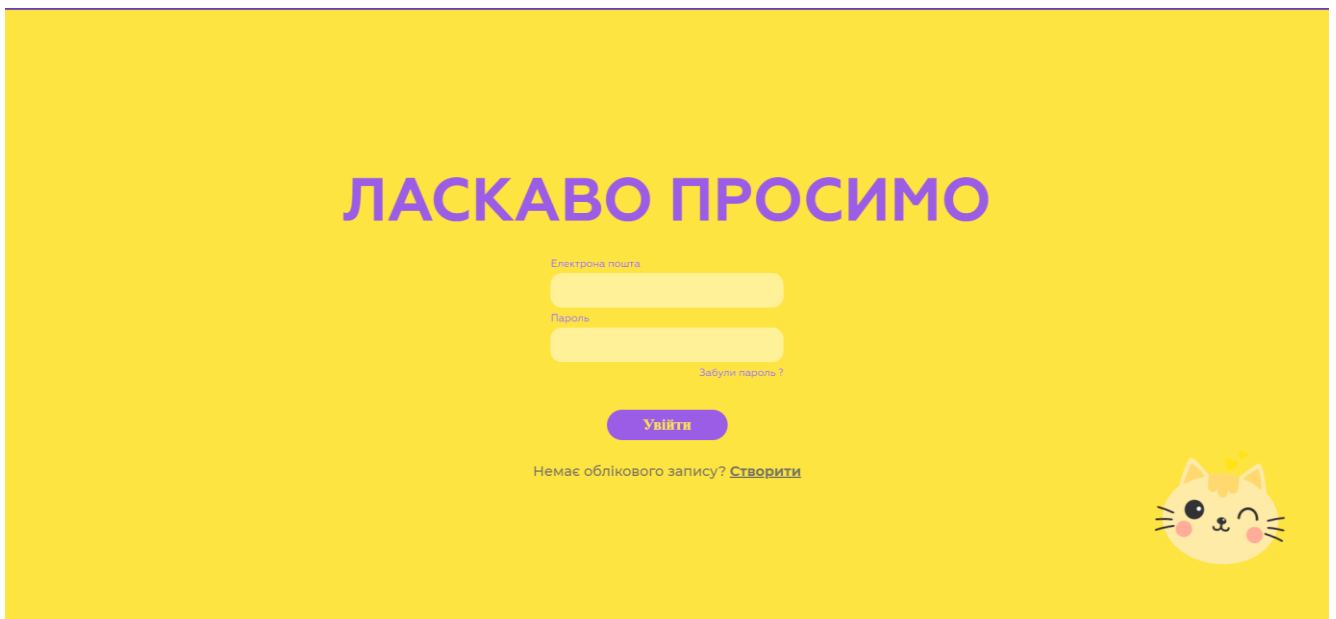


Рисунок Г 2 – Сторінка логінації для учня

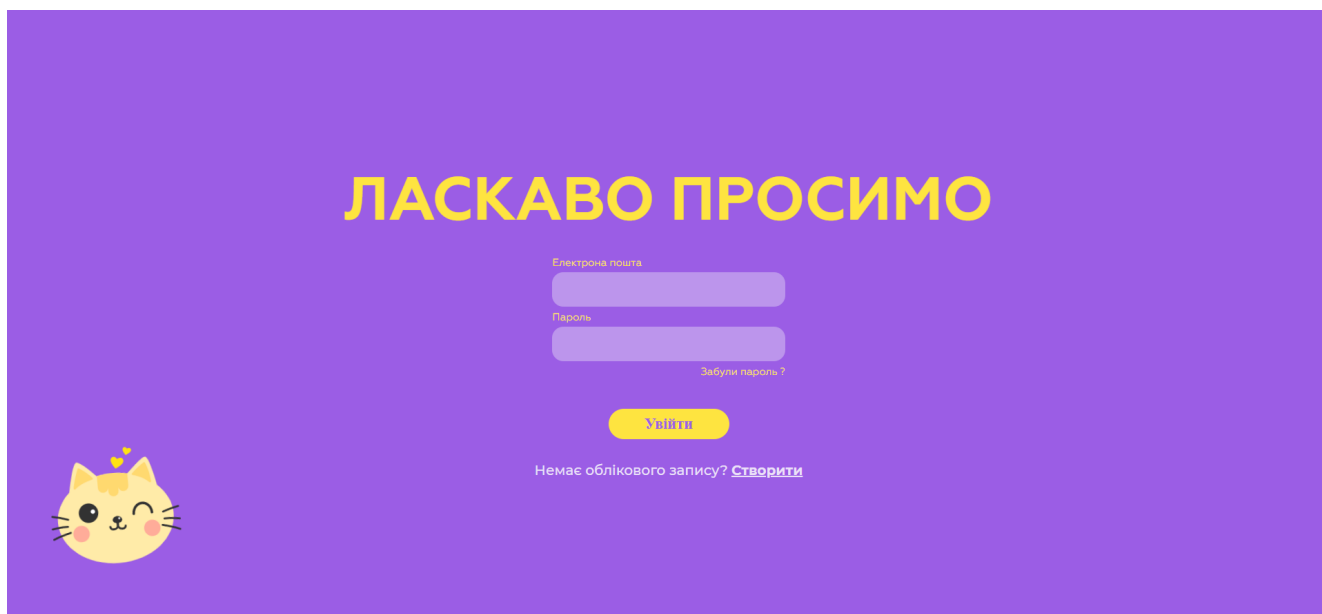


Рисунок Г 3 – Сторінка логізації для учителя

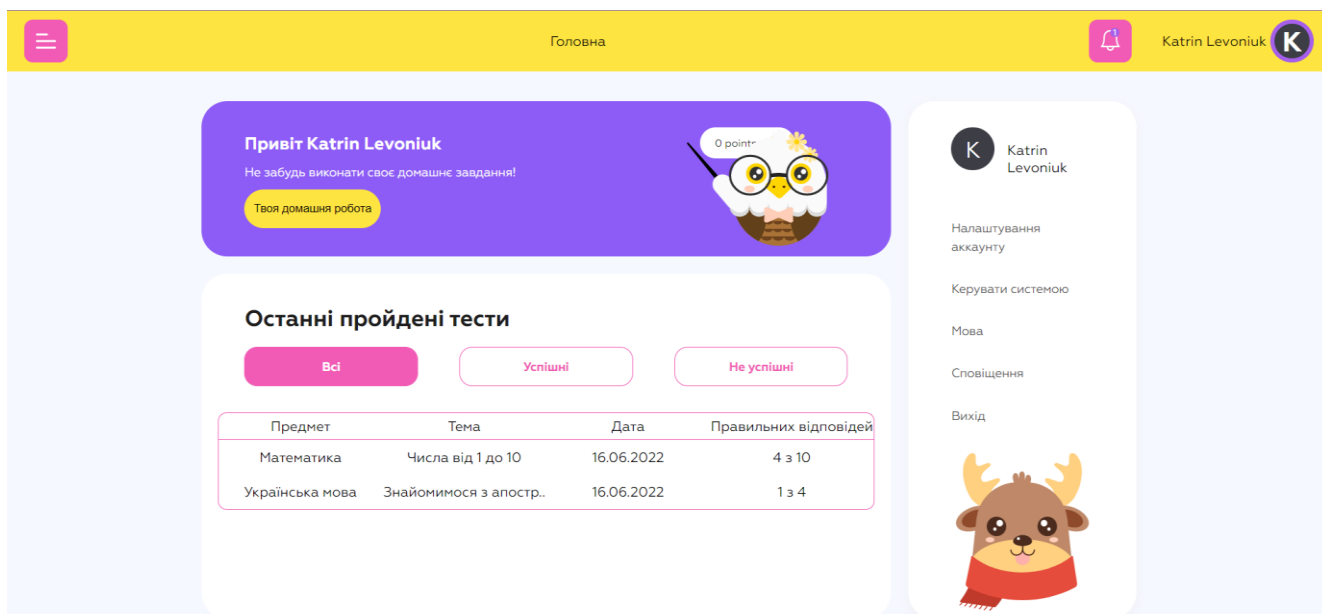


Рисунок Г 4 – Домашня сторінка

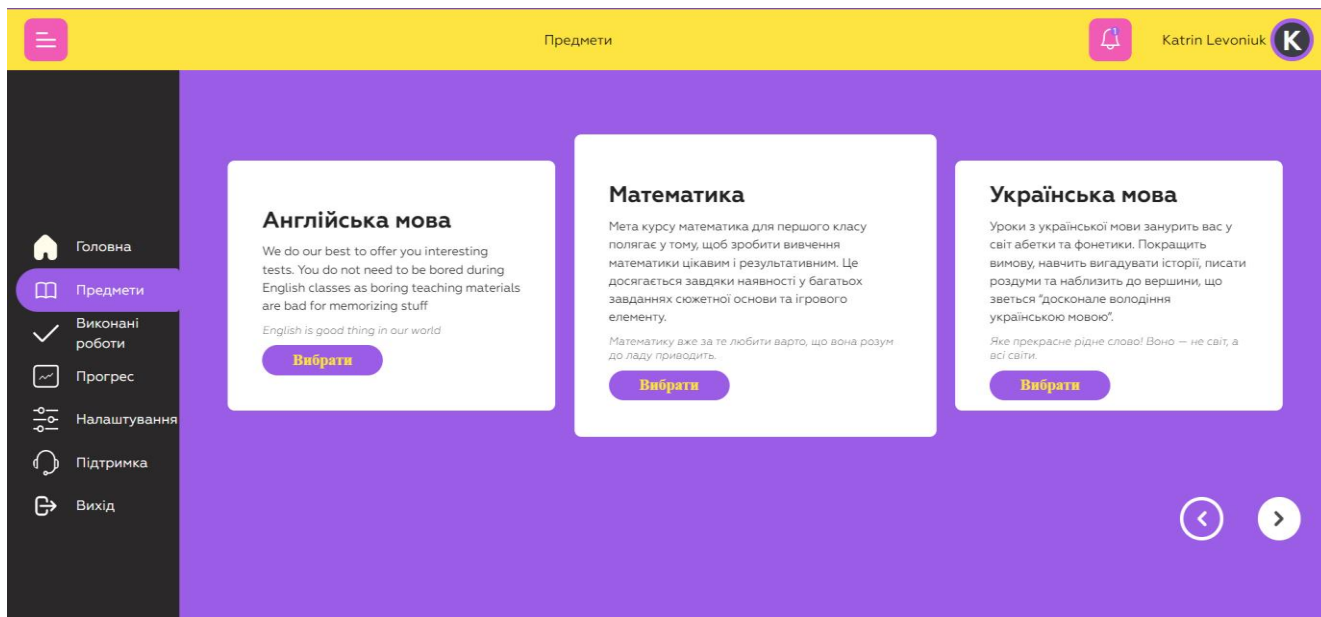


Рисунок Г 5 – Сторінка з предметами

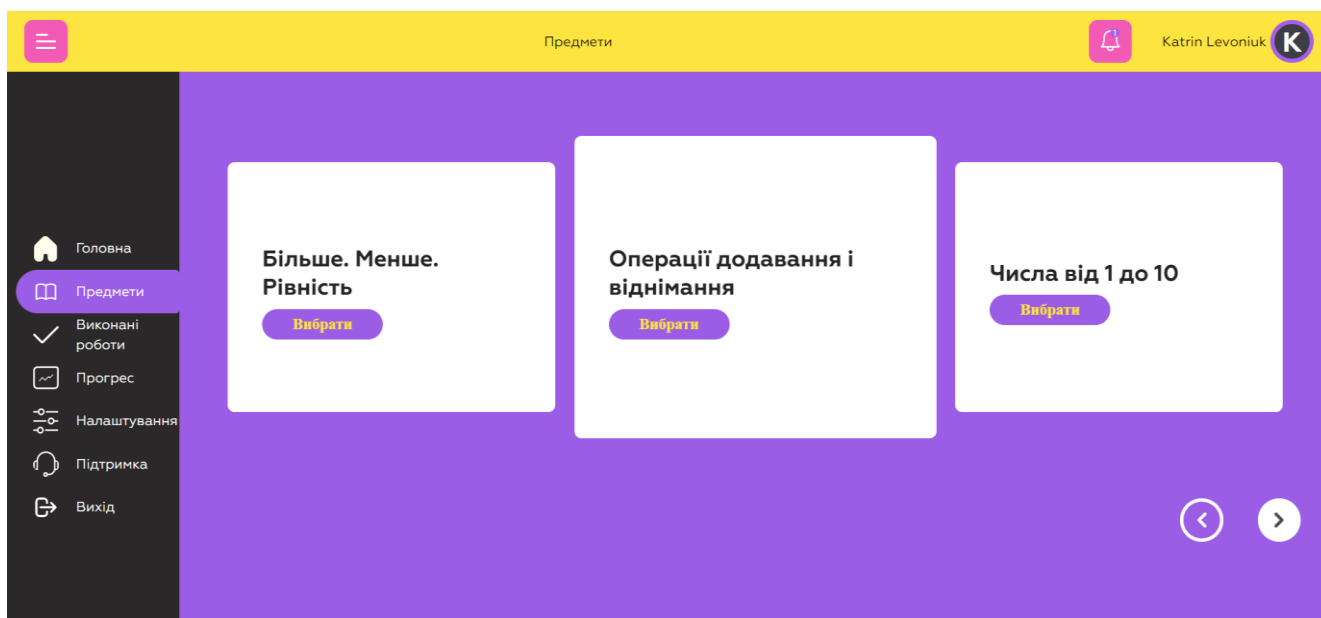


Рисунок Г 6 – Сторінка з темами

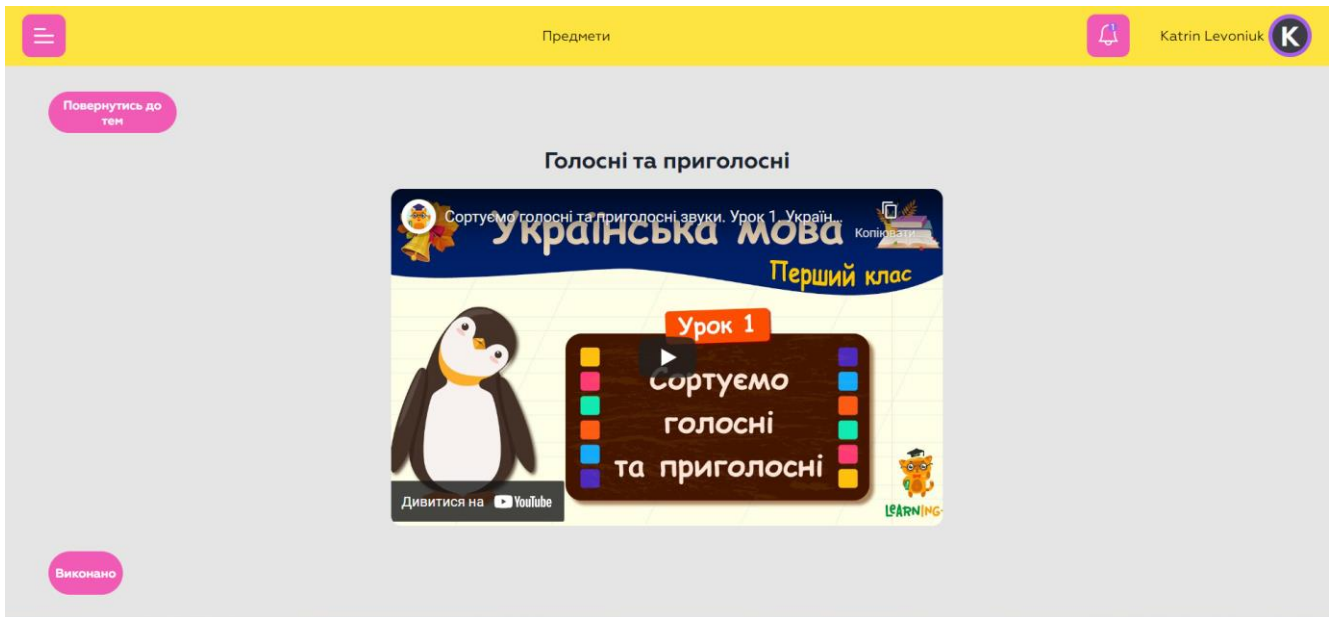


Рисунок Г 7 – Сторінка з відео

На цій сторінці ми можемо переглянути відео по темі. Та перейти на наступну сторінку.

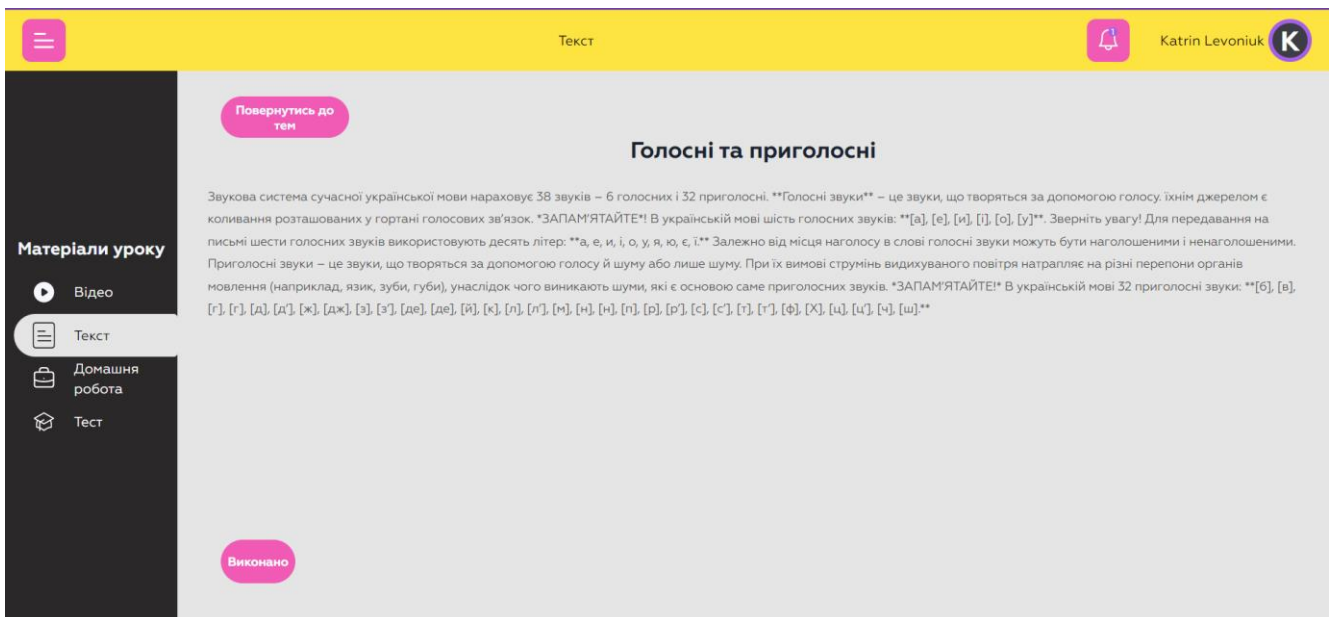


Рисунок Г 8 – Сторінка з додатковою інформацією

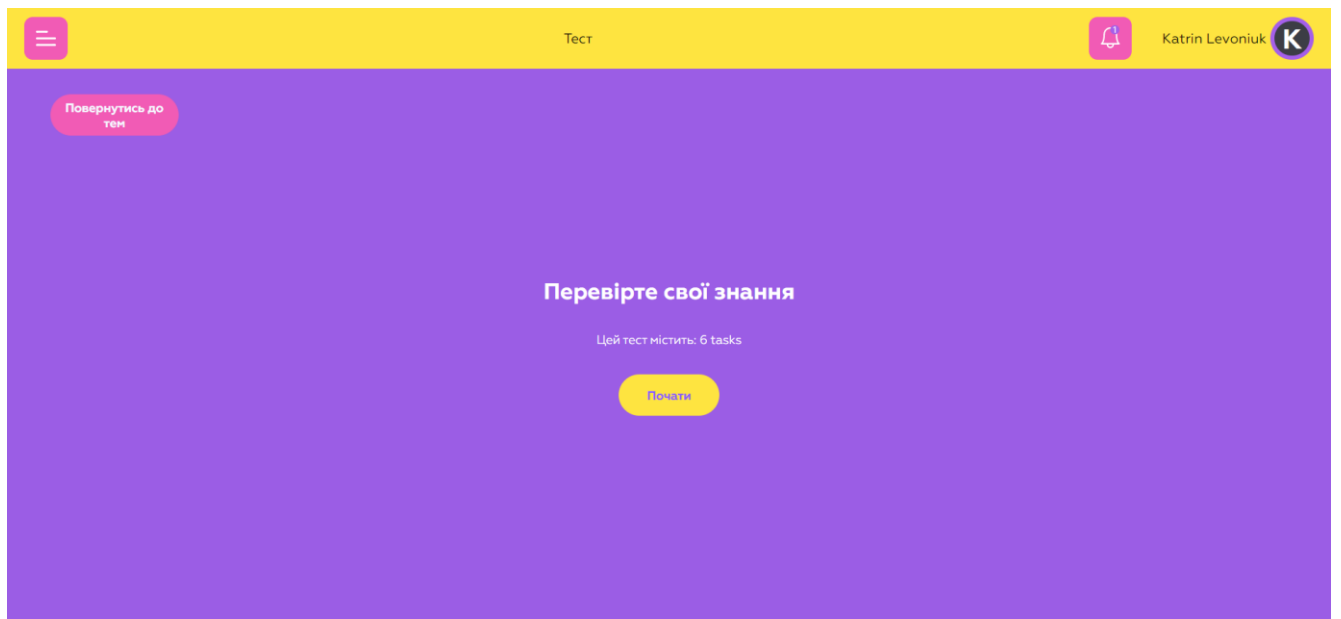


Рисунок Г 9 – Сторінка з інформацією про тест

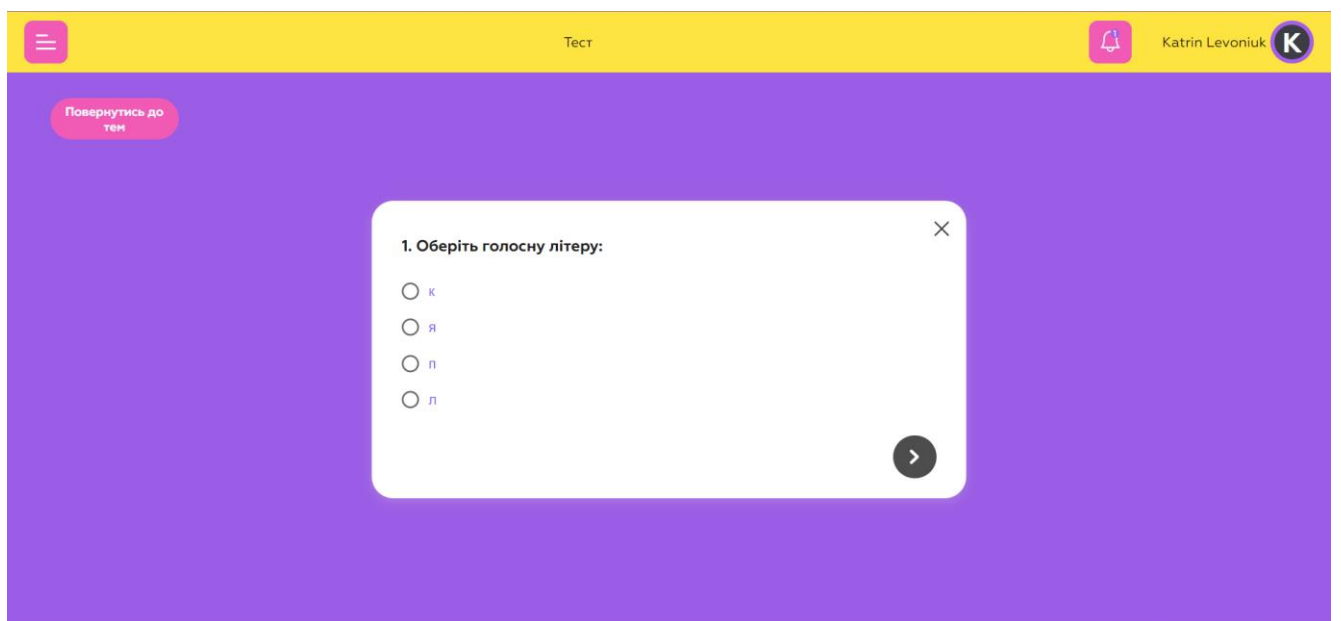


Рисунок Г 10– Сторінка тесту

## ДОДАТОК Ж «ФРАГМЕНТИ КОДУ ПРОГРАМИ

### 1. Код головної сторінки:

```
const SetUserDataInDB = useCallback( async (user) => {
  const userData = {
    uid: user.uid,
    first_name: user.first_name,
    email: user.email,
  };
  await setDoc(doc(db, "Users", user.uid), userData);
  await setDoc(doc(db, 'CompletedSubjects', user.uid), {uid: user.uid})
  dispatch(setUser(userData))
}, [])

const formik = useFormik<IRegister>({
  initialValues: {
    userName: '',
    email: '',
    password: '',
    confirmPassword: '',
  },
  validationSchema: Yup.object({
    userName: Yup.string().max(20, 'Must be 20 characters or
less').required('Required'),
    // password: Yup.string().uid()
  }),
  onSubmit: (values) => {
    if(values.confirmPassword === values.password) {
      createUserWithEmailAndPassword(auth, values.email,
values.password)
        .then((userCredential) => {
          const user = userCredential.user;
          SetUserDataInDB({uid: user.uid, first_name:
values.userName, email: values.email})
          navigate(pageNames.home)
        })
        .catch((error) => {
          const errorCode = error.code;
          const errorMessage = error.message;
        });
    }
  }
});
```

## 2. Код сторінки предметів:

```
async function fetchData () {
    const querySnapshot = await getDocs(collection(db, 'Subjects'));
    let newData = querySnapshot.docs.map(doc => doc.data())
    setSubjects([...newData])
}

useEffect(() => {
    fetchData()
}, [])

const subjectsObject = useMemo(() => {
    return subjects?.map((subject) => <SubjectComponent
        title={subject.title}
        description={subject.description}
        phrase={subject.phrase}
        handleClick={() => {
            dispatch(setSubject(subject))
            navigate(pathes.topics)
        }}
    />)
}, [subjects])
```

## 3. Код сторінки тем:

```
async function fetchData () {
    const SubjectTopics = query(collectionGroup(db, subject.topics_link))
    const querySnapshot = await getDocs(SubjectTopics);
    let newData = querySnapshot.docs.map(doc => doc.data())
    setGradeTopics(...newData)
}

useEffect(() => {
    fetchData()
}, [])

const topicsObject = useMemo(() => {
    return gradeTopics?.topics?.map((topic) => {
        if(topic.isHidden) {
            return (<SubjectComponent
                isHidden
            />)
        }
    })
}, [gradeTopics])
```

```

    }
    return (<SubjectComponent
      title={topic.title}
      handleClick={() => {
        dispatch(setTopic({grade: gradeTopics.title, gradeNumber:
gradeTopics.grade, ...topic}))
        navigate(pathes.topic.home)
      }}
    />)
  }).reverse()
}, [gradeTopics])

```

#### 4. Код сторінки тесту

```

useEffect(() => {
  fetchData()
}, [])

async function fetchData () {
  const testTopic = query(collectionGroup(db, topic.topic_link))
  const querySnapshot = await getDocs(testTopic);
  let newData = querySnapshot.docs.map(doc => doc.data())
  setTestsFromDB(newData)
}

const onStartTest = useCallback(() => {
  dispatch(startTest(testsFromDB))
}, [testsFromDB])

const matchAnswers = (tests: any, nextIndex: number): any => {
  for (let i=0; i < answers.length; i++) {
    if (answers[i]?.id === tests[nextIndex]?.id &&
(answers[i]?.question === tests[nextIndex]?.question)) {
      return matchAnswers(tests, nextIndex + 1)
    };
    continue;
  }
}

const onGiveAnswer = useCallback(() => {
  dispatch(setAnswers([...answers, answer]))
  const userAnswers = [...answers, answer].map((answer: any) => {

```

```

        const testRealAnswer = tests.find((test: any) => test.id ===
answer.id);
        return {
            ...answer,
            isCorrectAnswer:          answer?.answer          ===
testRealAnswer?.right_answer
        }
    });
    let countRightAnswers = 0;
    userAnswers.forEach((answer: any) => {
        if(answer?.isCorrectAnswer) {
            countRightAnswers++;
        }
    })
    addCompletedTests(userAnswers, countRightAnswers);
    CompletedWorks(userAnswers, countRightAnswers);
    dispatch(completeTest({results: userAnswers, countRightAnswers}));
}, [answers, tests, answer])

```

```

async function CompletedWorks(userAnswer, countRightAnswers) {
    const myProgressRef = doc(db, 'CompletedSubjects', user.uid);
    const myProgressObject = await getDoc(myProgressRef);
    const myProgressObjectData = myProgressObject.data()
    const progressData = myProgressObjectData ? myProgressObjectData : {};
    const      subjectData      =      myProgressObjectData      &&
Object.keys(myProgressObjectData).includes(subject.topics_link)      ?
myProgressObjectData[subject.topics_link] : {};
    const      gradeData      =      myProgressObjectData      &&
Object.keys(subjectData)?.includes(`${subject.topics_link}_grade${1})      ?
myProgressObjectData[subject.topics_link][`${subject.topics_link}_grade${1}]
: {};
    const      topicData      =      myProgressObjectData      &&
Object.keys(gradeData)?.includes(topic.topic_link)      ?
myProgressObjectData[subject.topics_link][`${subject.topics_link}_grade${1}][
topic.topic_link] : {};
    let isCompletedTest = topicData?.test?.isTest
    if(!isCompletedTest) {
        await setDoc(myProgressRef, {
            ...progressData,
            [`${subject.topics_link}`]: {
                ...subjectData,
                [`${topic?.grade}`]: {

```

```

        ...gradeData,
        [`${topic?.topic_link}`]: {
            ...topicData,
            test: {
                isTest:        countRightAnswers        ===
userAnswer.length ? true : false,
                date: moment(new Date()).toString()
            }
        }
    },
    }, {merge: true});
}
toast.success('Записав твої результати')
}

```

```

const addCompletedTests = async (answers: any, countRightAnswers: any) =>
{
    await addDoc(collection(db, "CompletedTests"), {
        uid: user?.uid,
        subject: subject?.topics_link,
        subject_route: subject?.route,
        subject_title: subject?.title,
        topic: topic?.topic_link,
        topic_route: topic?.route,
        topic_title: topic?.title,
        answers: answers,
        count_right_answers: countRightAnswers,
        count_tests: answers?.length,
        date: moment(new Date()).toString(),
    });
}

```

```

const onNextStep = useCallback(() => {
    if(answer) {
        if (answers?.length && (answers?.find((item: any) => item?.id ===
answer?.id && (item?.question === answer?.question  item?.answers[0] ===
answer?.answers[0]))) {
            let newAnswers = answers?.map((item: any) => {
                if (item?.id === answer?.id && (item?.question ===
answer?.question  item?.answers[0] === answer?.answers[0])) {
                    return answer;
                } else {

```

```

        return item;
    }
    })
    dispatch(setAnswers([...newAnswers]))
  } else {
    let newAnswers = answers?.length ? [...answers, answer] :
[answer];
    dispatch(setAnswers([...newAnswers]))
  }
  setAnswer(() => null)
  currentTest < tests?.length-1 &&
dispatch(setCurrentTest(currentTest + 1))
}
}, [currentTest, answer, answers, tests])

const onPreviousStep = useCallback(() => {
  setAnswer(null)
  dispatch(setCurrentTest(currentTest-1))
}, [currentTest])

const onCloseTest = useCallback(() => {
  setAnswer(null)
  dispatch(closeTest())
}, [])

```