

# **Навчальний застосунок “Математика: повторення”**

*Застосунок призначений для теоретичного опрацювання шкільного курсу алгебри з 5 по 11 класи.*

## **1. Вступ**

### **1.1. Мета**

Метою цього документа є визначення вимог до створення вебсервісу “Математика: повторення”, призначеного для повторення шкільного курсу математики (алгебра та геометрія) для учнів 5–11 класів. Сервіс надає інтерактивні можливості для теоретичного вивчення, проходження тестів, а також інтеграцію з Telegram-ботом для зручної взаємодії. “Математика: повторення” стане ефективним інструментом підготовки до державних іспитів, забезпечуючи персоналізований підхід до навчання та контролю знань.

### **1.2. Позиціонування продукту**

“Математика: повторення” створений для задоволення актуальної потреби в сучасних освітніх інструментах, які допомагають учням якісно підготуватися до державних іспитів, таких як ЗНО чи НМТ. Застосунок забезпечує зручний доступ до теоретичного матеріалу з алгебри та геометрії, дозволяючи користувачам працювати у власному темпі. Особливістю сервісу є інтеграція з Telegram-ботом, що спрощує отримання домашніх завдань, нагадувань і аналітики для вчителів. Крім того, “Математика: повторення” пропонує швидкий доступ до архіву тестів ЗНО/НМТ, можливість зберігати нотатки та відстежувати активність учнів. Цей інструмент має зробити процес навчання більш гнучким, структурованим і доступним як для учнів, так і для вчителів.

### **1.3. Цільова аудиторія**

Основною аудиторією продукту є учні 5–11 класів, які прагнуть покращити свої знання з математики. Також вчителі, які шукають інструмент для організації навчального процесу, моніторингу активності та контролю знань учнів і батьки, які хочуть контролювати навчальний прогрес своїх дітей.

## **2. Опис продукту**

### **2.1. Особливості продукту**

“Математика: повторення” — це інтерактивний вебсервіс, створений для допомоги учням 5–11 класів у повторенні та поглибленні знань з математики. Продукт поєднує доступ до теоретичних матеріалів, тестів, персональної статистики та додаткового інструментарію, що робить процес навчання зручним і ефективним.

Основні можливості застосування включають:

1. Вивчення математичних тем: сервіс надає доступ до широкого спектра тем, включаючи алгебру, геометрію, тригонометрію, диференціальне та інтегральне числення, ймовірність і статистику.
2. Теоретичний матеріал: для кожної теми доступні визначення, теореми, формули та приклади, що супроводжуються покроковими поясненнями.
3. Словник математичних термінів: інтегрований словник забезпечує швидкий доступ до визначень основних термінів.
4. Підготовка до тестувань: застосунок пропонує можливість проходження онлайн-тестів, зокрема у форматі ЗНО/НМТ. Сервіс

також допомагає у регулярному закріпленні знань упродовж навчального року.

5. Персоналізована статистика: сервіс веде облік навчального часу, відстежуючи, скільки часу користувачі проводять у програмі. Це дозволяє вчителям отримувати детальні звіти про активність учнів, а самим учням — контролювати свій прогрес.
6. Нотатки: користувачі мають змогу зберігати власні записи, ідеї чи важливу інформацію прямо в застосунку. Всі нотатки зберігаються автоматично та впорядковуються за датою.
7. Доступність матеріалів: теоретичні матеріали з алгебри доступні офлайн, тоді як для доступу до геометричних матеріалів, що містять фото, та проходження тестів необхідне підключення до Інтернету.
8. Інтеграція з Telegram-ботом: бот сповіщає користувачів про нові завдання, нагадує про активність, пропущені завдання та нові теми і дозволяє швидко отримувати зворотний зв'язок.
9. Збір відгуків: продукт передбачає функціонал для збору відгуків через кнопку зворотного зв'язку, що дозволяє розробникам отримувати актуальну інформацію для покращення сервісу.
10. Ролі користувачів: підтримує дві основні ролі користувачів: учня і вчителя. Учні мають можливість проходити тести, переглядати навчальні матеріали, отримувати завдання через Telegram-бот і відстежувати власну активність. Вчителі можуть створювати класи, додавати учнів, контролювати їх активність, призначати домашні завдання і отримувати статистичні звіти.

## **2.2. Функціональні вимоги**

Незареєстровані користувачі можуть:

1. Переглядати теоретичні матеріали за темами.
2. Отримувати зворотний зв'язок через кнопку у застосунку.
3. Мати доступ до матеріалів офлайн (для алгебри та геометрії).

Зареєстровані користувачі можуть:

1. Проходити тести та отримувати результати.
2. Зберігати нотатки і переглядати їх.
3. Переглядати персональну статистику активності та прогресу.
4. Отримувати сповіщення через Telegram-бот.
5. Вчителі можуть створювати та управляти класами.
6. Вчителі можуть призначати домашні завдання учням.
7. Вчителі можуть створювати класи і додавати учнів
8. Вчителі можуть контролювати активність своїх учнів та отримувати звіти.

## **2.3. Нефункціональні вимоги**

### **1. Продуктивність**

- a) Час відгуку сервісу на запит користувача не повинен перевищувати 1 секунди для 95% запитів.
- b) Час завантаження головної сторінки не більше 3 секунд при стандартному навантаженні.
- c) Система повинна обробляти не менше 1000 одночасних активних користувачів без зниження продуктивності.

### **2. Безпека**

- a) Всі дані користувачів повинні зберігатися у зашифрованому вигляді за допомогою алгоритму AES-256.
- b) Паролі зберігаються у вигляді хешів за допомогою алгоритму bcrypt.
- c) Всі зовнішні API-запити мають здійснюватися через HTTPS
- d) Ввести механізм багатофакторної автентифікації (MFA) для вчителів та адміністраторів.

### **3. Надійність**

- a) Доступність сервісу повинна становити не менше 99.9% часу на місяць.
- b) Впровадити автоматичне резервне копіювання бази даних щодня, зберігаючи копії протягом 30 днів.

- c) Визначити процедуру відновлення після збоїв, що не перевищує 2 години.

#### 4. Масштабованість

- a) Система повинна підтримувати горизонтальне масштабування для обробки збільшення кількості користувачів.
- b) Вебсервіс має витримувати збільшення навантаження до 2000 одночасних активних користувачів без критичних збоїв.

#### 5. Кросплатформеність

- a) Сервіс має бути доступний на основних браузерях (Chrome, Safari, Edge).
- b) Забезпечити коректне відображення інтерфейсу на екранах з різною роздільною здатністю та розмірами дисплеїв

#### 6. Юзабіліті

- a) Для нових користувачів повинна бути доступна інтерактивна інструкція користування.
- b) Проводити регулярне тестування інтерфейсу для покращення зручності використання не рідше ніж раз на півроку.

#### 7. Документованість

- a) Всі публічні API повинні бути задокументовані у форматі OpenAPI.
- b) Керівництво користувача повинно бути доступне у вигляді онлайн-документації.

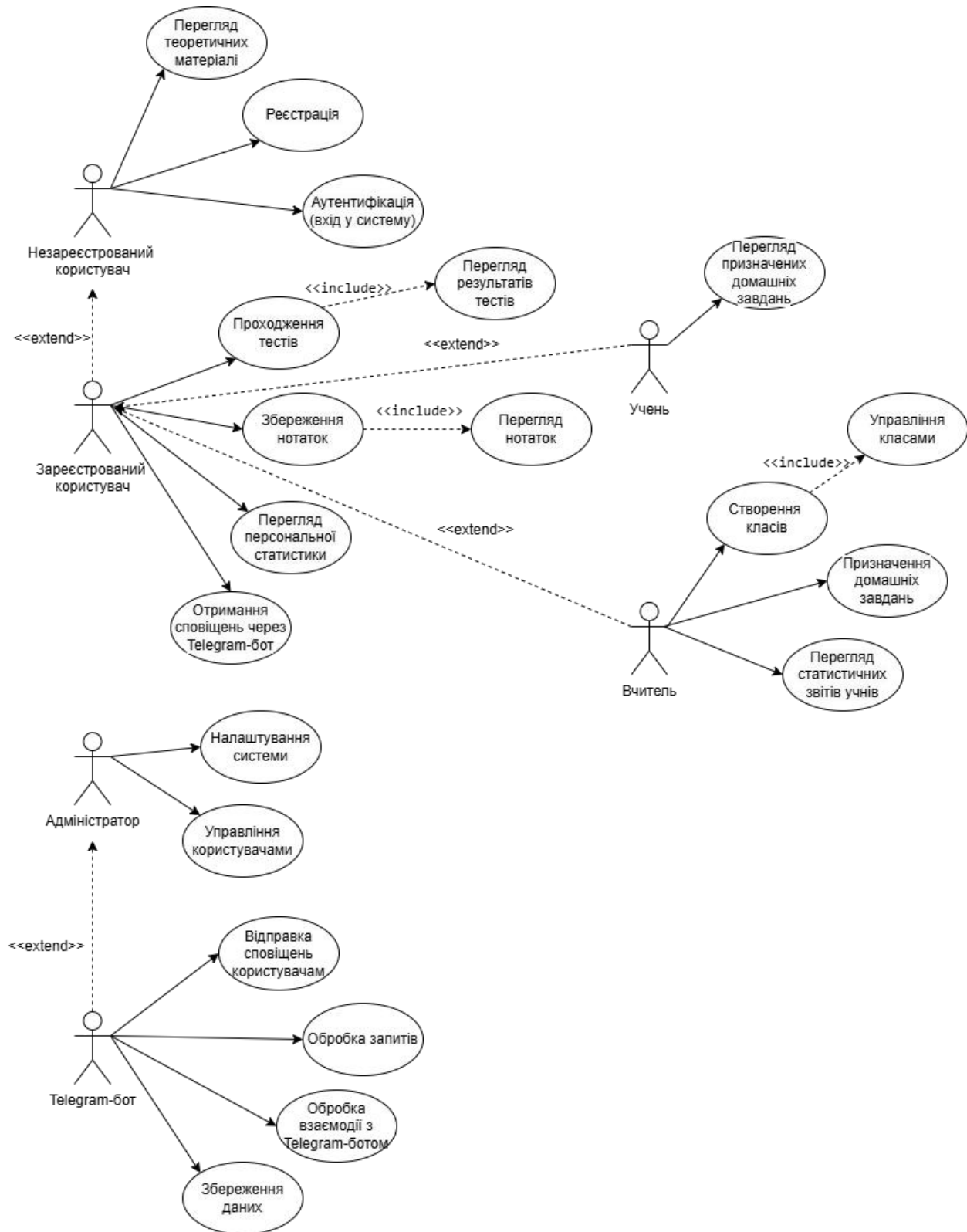
8. Відновлюваність:

- a) У випадку збоїв система повинна забезпечити відновлення останніх збережених даних протягом 15 хвилин після аварії.
- b) Впровадити систему сповіщень про збої для адміністраторів через електронну пошту або Telegram-бота.

9. Вимоги до тестування:

- a) Покриття тестами має становити не менше 80% для основного коду.
- b) Регресійне тестування повинно проводитися перед кожним основним релізом.

## 2.4. Діаграма прецедентів (Use-case diagram)





Назва	Функціональність	Опис
Перегляд теоретичних матеріалів	Надання доступу до навчальних матеріалів.	Користувачі можуть переглядати теоретичні дані з математики для підготовки до тестів.
Реєстрація	Створення нового облікового запису.	Користувач вводить персональні дані для реєстрації у системі.
Аутентифікація (вхід у систему)	Перевірка облікових даних.	Користувач вводить логін і пароль для доступу до персонального акаунту.
Проходження тестів	Виконання тестових завдань.	Користувачі проходять інтерактивні тести для перевірки знань.
Збереження нотаток	Додавання особистих нотаток	Користувач може зберігати свої записи під час навчання.
Перегляд нотаток	Доступ до збережених нотаток	Перегляд, редагування або видалення власних нотаток.
Перегляд результатів тестів	Аналіз успішності	Користувачі можуть переглядати результати виконаних тестів
Перегляд персональної статистики	Моніторинг успішності	Відображення персональних статистичних даних щодо пройдених тестів
Перегляд призначених домашніх завдань	Доступ до завдань від вчителя	Учні можуть переглядати завдання, які їм призначено
Управління класами	Організація навчальних груп	Вчитель може створювати, редагувати та видаляти класи
Створення класів	Додавання нових навчальних груп	Вчитель створює класи для групування учні
Призначення домашніх завдань	Призначення завдань для виконання	Викладачі можуть призначати домашні завдання конкретним класам

		або учням
Перегляд статистичних звітів учнів	Аналіз успішності учнів	Викладачі можуть переглядати детальні статистичні дані про успішність учнів
Отримання сповіщень через Telegram-бот	Отримання повідомлень	Учні отримують повідомлення про завдання чи результати через Telegram-бот.
Відправка сповіщень користувачам	Надсилення важливої інформації	Адміністратор може надсилати сповіщення всім користувачам
Обробка запитів	Опрацювання запитів користувачів	Система обробляє запити, отримані через вебінтерфейс чи Telegram-бот
Обробка взаємодії з Telegram-ботом	Інтеграція з ботом	Система забезпечує обробку даних і відповідей, отриманих через Telegram-бот
Збереження даних	Постійне збереження інформації	Система забезпечує надійне збереження інформації у базі даних
Налаштування системи	Конфігурація платформи	Адміністратор може змінювати налаштування системи відповідно до потреб
Управління користувачами	Додання та видалення користувачів	Адміністратор може керувати обліковими записами користувачів

## 2.5. User Stories

1. Як учень, я хочу переглядати теоретичні матеріали, щоб підготуватися до тестів.
2. Як учень, я хочу пройти реєстрацію в системі, щоб отримати доступ до функціональності платформи.
3. Як учень, я хочу мати можливість аутентифікуватися, щоб захистити свої дані та результати.
4. Як учень, я хочу проходити тести, щоб перевірити свої знання з математики.
5. Як учень, я хочу зберігати нотатки, щоб мати швидкий доступ до важливої інформації.
6. Як учень, я хочу переглядати свої нотатки, щоб повторювати матеріал.
7. Як учень, я хочу переглядати результати тестів, щоб оцінити свою успішність.
8. Як учень, я хочу бачити свою персональну статистику, щоб відстежувати прогрес.
9. Як учень, я хочу отримувати сповіщення через Telegram-бот, щоб не пропускати важливі новини.
10. Як учень, я хочу переглядати призначені домашні завдання, щоб знати, що потрібно виконати.
11. Як викладач, я хочу створювати класи, щоб організувати учнів у навчальні групи.

12. Як викладач, я хочу призначати домашні завдання, щоб перевірити знання учнів.

13. Як викладач, я хочу переглядати статистичні звіти учнів, щоб оцінювати їхній прогрес.

14. Як викладач, я хочу надсилати сповіщення учням, щоб інформувати їх про важливі події.

15. Як адміністратор, я хочу налаштовувати систему, щоб забезпечити її ефективне функціонування.

16. Як адміністратор, я хочу управляти користувачами, щоб додавати або видаляти облікові записи.

17. Як адміністратор, я хочу обробляти запити користувачів, щоб вирішувати їхні проблеми.

18. Як адміністратор, я хочу забезпечити обробку взаємодії з Telegram-ботом, щоб підтримувати комунікацію з користувачами.

19. Як адміністратор, я хочу зберігати дані, щоб забезпечити їх надійне збереження.

## **2.6. Технологічний стек**

“Математика: повторення” реалізований на основі мікросервісної архітектури з використанням сучасних технологій.

Аутентифікація: JWT-токен з обмеженим часом життя для періодичного перелогінювання користувачів. У майбутньому планується додати підтримку двох токенів для покращення безпеки.

Архітектура: Мікросервісний підхід.

Спілкування: REST API (клієнт-сервер).

База даних: PostgreSQL.

Back-end: .NET Core 8.

Front-end: React + TypeScript.

## **2.7. Зацікавлені сторони**

1. Учні: отримання доступу до теоретичних матеріалів, проходження тестів, перегляд результатів і статистики, отримання сповіщень через Telegram-бот.
2. Вчителі: створення класів, призначення домашніх завдань, перегляд статистичних звітів учнів, контроль за успішністю.
3. Адміністратор системи: управління користувачами, налаштування системи, забезпечення стабільної роботи сервісу, відправка сповіщень.
4. Розробники: розробка, тестування та підтримка функціональності системи.

## 2.8. Обмеження та ризики

*Обмеження:* застосунок використовує мікросервісну архітектуру зі статичним JWT токеном, який має обмежений час життя. Це означає, що користувачі мають періодично перелогінюватися для підтримки активної сесії. Система наразі не підтримує механізм із двома токенами (access і refresh), що обмежує рівень безпеки, але це планується реалізувати в майбутньому. Взаємодія клієнта з сервером відбувається через REST API, що обмежує систему у випадках, коли потрібна асинхронна або подієва комунікація.

*Ризики:* основним ризиком є можливість втрати даних або збоїв у роботі бази даних PostgreSQL у разі високого навантаження або неправильної конфігурації. Також існує ризик безпеки, пов'язаний із використанням лише одного токена для аутентифікації, що може призвести до потенційних атак, якщо токен буде скомпрометовано. Крім того, відсутність системи резервного копіювання на початкових етапах може створити загрозу втрати важливої інформації.