

ДОДАТОК А

Звіт з результатів перевірки на унікальність тексту в базі ХНУРЕ



Рисунок А.1 – Результат перевірки на унікальність тексту

ДОДАТОК Б

Слайди презентації



Програмна система для інтерактивного вивчення іноземних мов

Бондаренко Костянтин Тарасович, ПЗПІ-21-3
Керівник: ст.викл. кафедри Ляпота В.М.



23 червня 2025

Рисунок Б.1 – Слайд презентації

Актуальність та мета роботи

Розробка веб-платформи для інтерактивного вивчення іноземних мов із персоналізованим підходом, підтримкою різноманітних типів завдань та елементів гейміфікації.

Зростає попит на інтерактивні, персоналізовані платформи для вивчення мов, які дозволяють налаштувати навчальний процес відповідно до індивідуальних потреб користувачів.



Рисунок Б.2 – Слайд презентації

Аналіз проблеми (аналіз існуючих рішень)



- Обмежена кастомізація
- Мало типів завдань
- Відсутність можливості створення/імпорту власних матеріалів
- Низька адаптивність до індивідуальних цілей



3

Рисунок Б.3 – Слайд презентації

Постановка задачі та опис системи

Створення інтерактивної платформи для вивчення іноземних мов із різноманітними типами завдань.

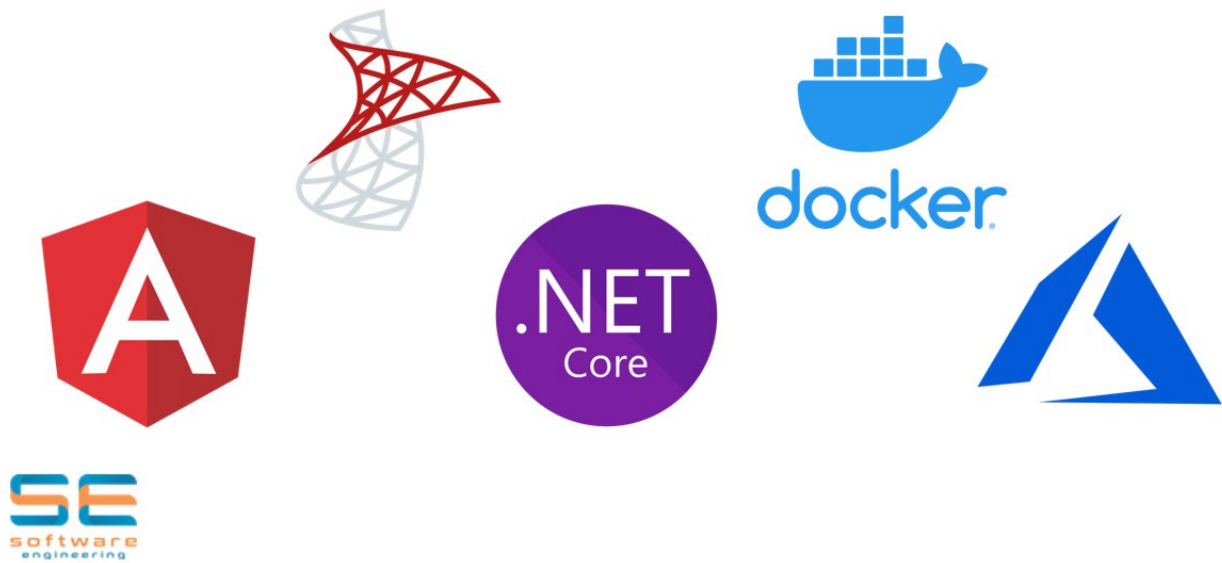
- Гнучка система для створення індивідуальних завдань
- Підтримка різних типів завдань
- Відстеження прогресу користувачів



4

Рисунок Б.4 – Слайд презентації

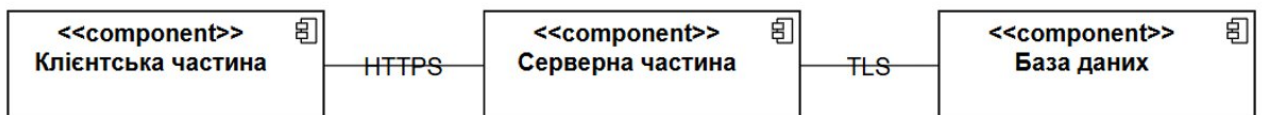
Вибір технологій розробки



5

Рисунок Б.5 – Слайд презентації

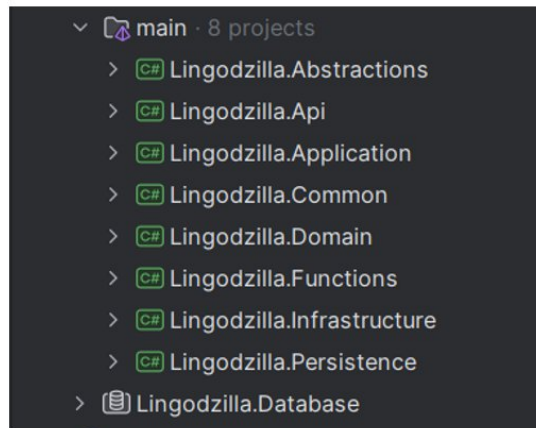
Архітектура створеног програмного забезпечення



6

Рисунок Б.6 – Слайд презентації

Архітектура створеног програмного забезпечення

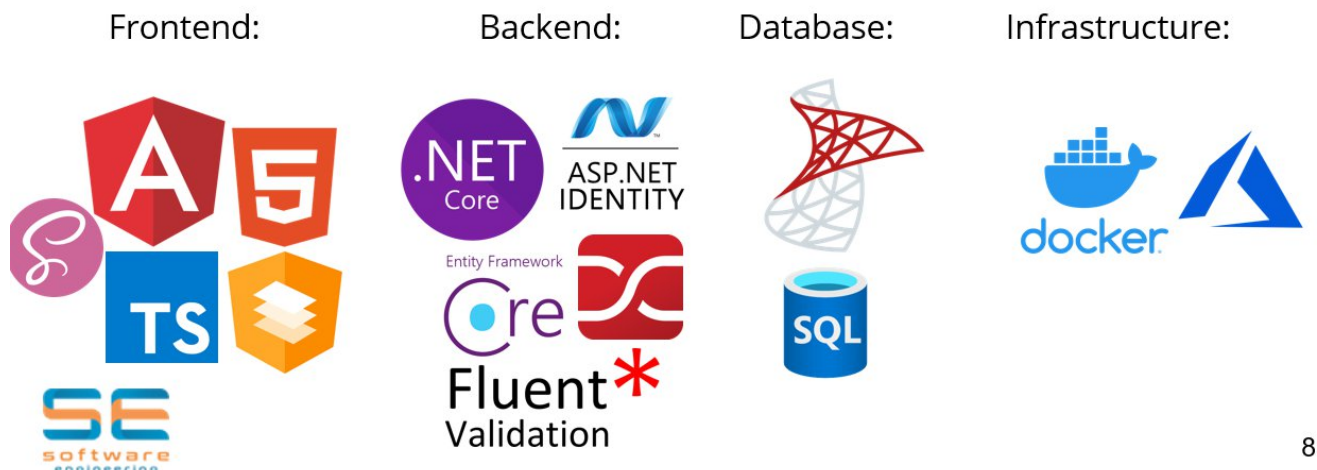


На серверній частині була використана Onion архітектура

7

Рисунок Б.7 – Слайд презентації

Процес розробки та вибрані мови програмування та фреймворки.



8

Рисунок Б.8 – Слайд презентації

Дизайн системи

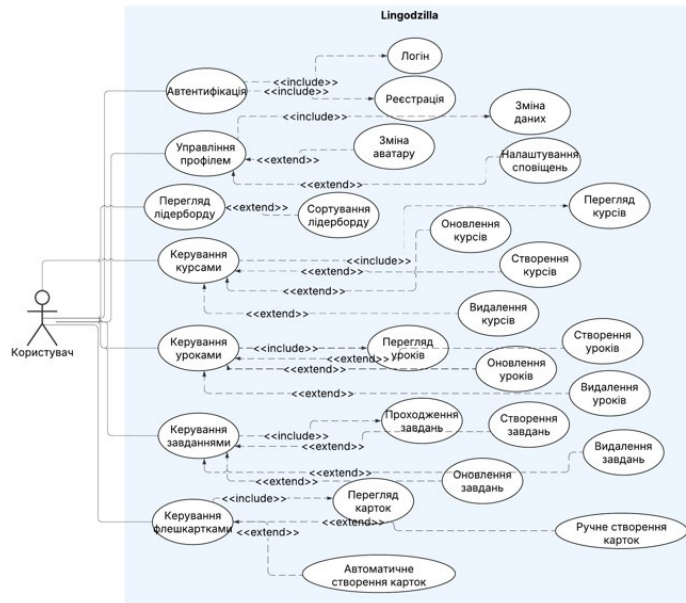


Рисунок Б.9 – Слайд презентації

Дизайн системи

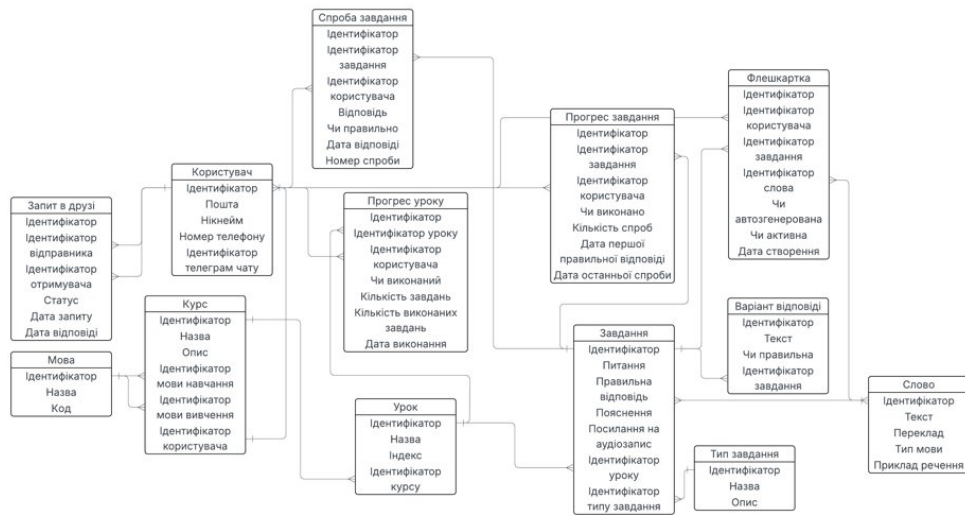


Рисунок Б.10 – Слайд презентації

Приклад реалізації

```

[UpdateMessageContact is not null] && state is BotState.AwaitingPhone)
{
    var phoneNumber = update.Message.Contact.PhoneNumber;
    var user = await _userRepository.SetByPhoneNumberAsync(phone);

    if (user is null)
    {
        await _telegramBot.SendTextMessageAsync(chatId, "User not found. Try again.");
        return req.CreateResponse(HttpStatusCode.OK);
    }

    user.TelegramChatId = chatId;
    _userRepository.Update(user);
    await _unitOfWork.SaveChangesAsync();

    UserState(chatId) = BotState.AwaitingConfirmation;

    await _telegramBot.SendTextMessageAsync(chatId, "Hi {user.UserName}! Would you like to see your flashcards? (Yes/No)");
    return req.CreateResponse(HttpStatusCode.OK);
}

if (state is BotState.AwaitingConfirmation)
{
    var text = update.Message.Text?.Trim().ToLower();
    if (text == "yes")
    {
        FlashcardIndex(chatId) = 0;
        UserState(chatId) = BotState.ShowingFlashcards;
        await ShowNextFlashcardAsync(_telegramBot, chatId);
    }
    else
    {
        await _telegramBot.SendTextMessageAsync(chatId, "Okay. Type 'yes' anytime to view your flashcards.");
    }

    return req.CreateResponse(HttpStatusCode.OK);
}

```

```

[FunctionName("WeeklyReminderFunction")]
[return: typeof(void)]
public async Task Run(
    [TimerTrigger("*/15 * * * * *")] TimerInfo myTimer,
    FunctionContext context)
{
    var logger = context.GetLogger(nameof(WeeklyReminderFunction));
    logger.LogInformation($"Weekly reminder started at {DateTime.UtcNow}");

    var users = _userManager.Users;
    var eligibleUsers = users
        .Where(u => u.SendReminders && !string.IsNullOrWhiteSpace(u.PhoneNumber))
        .ToList();

    logger.LogInformation($"Found {eligibleUsers.Count} users to remind.");

    foreach (var user in eligibleUsers)
    {
        if (user.TelegramChatId is null)
        {
            logger.LogWarning($"User {user.UserName} has no TelegramChatId, skipping.");
            continue;
        }

        try
        {
            await _telegramBot.SendTextMessageAsync(
                chatId: user.TelegramChatId,
                text: $"👋 Hey {user.UserName}, don't forget to learn a language this week!",
                cancellationToken: CancellationToken.None
            );
        }
        catch (Exception ex)
        {
            logger.LogError(ex, "Failed to send reminder to {user.UserName}.");
        }
    }
}

```

Створені Azure Functions як вебхук для Telegram бота. Функції надсилають нагадування та флешкартки користувачам.



Рисунок Б.11 – Слайд презентації

Інтерфейс користувача

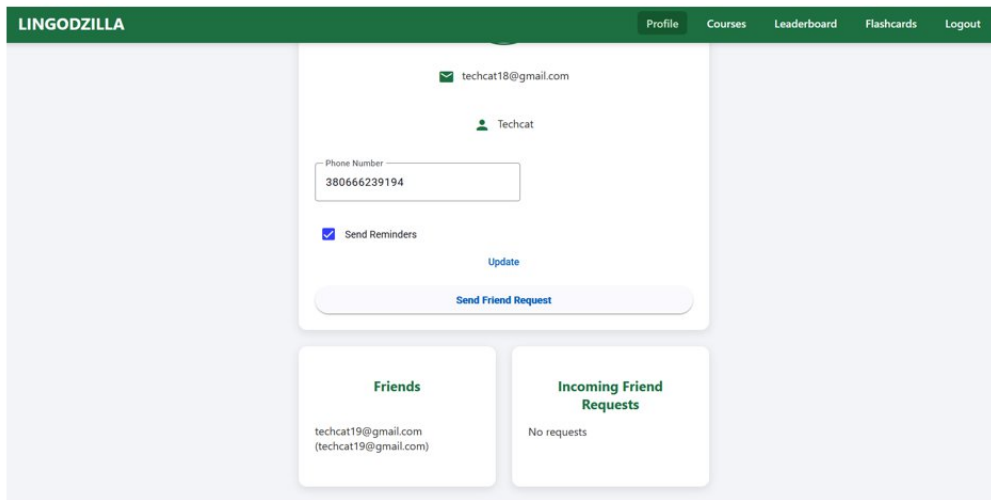
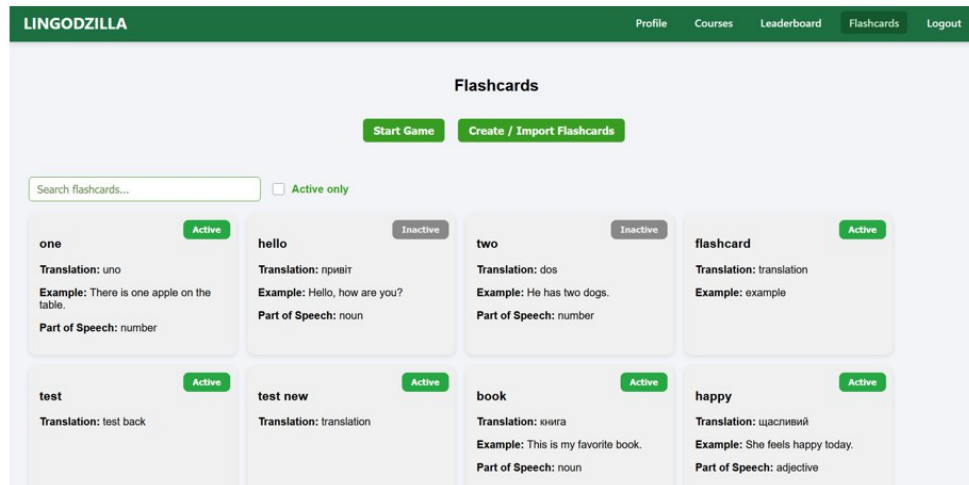


Рисунок Б.12 – Слайд презентації

Інтерфейс користувача



13

Рисунок Б.13 – Слайд презентації

Тестування

Функціональне тестування
Тестування продуктивності
Тестування безпеки

Postman, Swagger, Insomnia



14

Рисунок Б.14 – Слайд презентації

Підсумки

Розроблена система є реалістичною та корисною для широкого кола користувачів, включаючи студентів, працівників і викладачів. Вона дозволяє ефективно вивчати іноземні мови завдяки індивідуалізованим планам навчання та різноманітним типам завдань, що підвищують мотивацію та залученість.

Інтеграція штучного інтелекту для адаптивного навчання, розширення мовної підтримки та інтеграція з іншими освітніми платформами.

