**Національний університет кораблебудування імені адмірала Макарова**

**Навчально-науковий гуманітарний інститут**

Кафедра прикладної лінгвістики

**ЗВІТ**

**ПРО НАУКОВЕ СТАЖУВАННЯ**

**з комп*’*ютерно-лінгвістичної практики**

Виконав: студент  
групи 6721м

Зродніков І. К.

Керівник практики: канд. техн. наук, доц. Ажищев В.Ф.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(посада, вчене звання, прізвище, ініціали)*

Миколаїв 2025

**ЗМІСТ**

| **ВСТУП……………………………………………………………………** | **3** |
| --- | --- |
| **РОЗДІЛ 1. КОНЦЕПЦІЯ ДИПЛОМНОЇ РОБОТИ…………………………………………………………………….** | **4** |
| **РОЗДІЛ 2. ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ………………………………………………………………….** | **6** |
| **РОЗДІЛ З. АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ТА ІС……………………………………………………………………………** | **7** |
| **РОЗДІЛ 4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ ЗАСОБІВ АВТОМАТИЗАЦІЇ ОБРОБКИ ІНФОРМАЦІЇ……………………………………………………………...** | **11** |
| **ВИСНОВКИ………………………………………………………………** | **14** |
| **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ………………………………** | **15** |

**ВСТУП**

Практика присвячена розробці методичного посібника “Дидактичні ігри у викладання англійської мови”.

Посібник розроблений у вигляді електронного документа та вебсайта.

**Актуальність** комп’ютерно-лінгвістичної практики в тому, що під час її проходження є можливість проаналізувати наявні програмні засоби та визначити найбільш релевантні з них для розробки посібника.

**Мета** комп’ютерно-лінгвістичної практики – підбір інформаційних ресурсів, інформаційних систем та програмних засобів для написання магістерської дипломної роботи.

**Завдання** практики:

1. опис концепції магістерської дипломної роботи;
2. пошук та застосування інформаційних ресурсів, що відповідають темі та меті дипломної роботи;
3. аналіз наявних програмних засобів та ІС і вибір їх для реалізації мети дипломної магістерської роботи;
4. набуття навичок роботи з засобами автоматизації обробки інформації

Звіт з проходження комп’ютерно-лінгвістичної практики складається з таких **структурних частин**: вступ, чотири розділи, висновки та список використаних джерел.

У вступі зазначено актуальність, мету та завдання комп’ютерно-лінгістичної практики.

У першому розділі описано концепцію дипломної роботи.

Другий розділ присвячений аналізу інформаційних ресурсів, а саме посібників та підручників, що стали підґрунтям написання магістерської дипломної роботи.

У третьому розділі описується процес дослідження та вибору програмних засобів та інформаційних систем для реалізації методичного посібника.

У четвертому розділі наведено приклади практичного застосування засобів автоматизації обробки інформації.

У результаті проходження практики було створено паперовий посібник та вебсайт для викладання англійської мови, розрахований на використання в умовах дистанційного та змішаного навчання, який враховує використання дидактичних ігор.

**РОЗДІЛ 1. КОНЦЕПЦІЯ ДИПЛОМНОЇ РОБОТИ**

Незважаючи на відносно стабільний характер основних засад освіти, кожне покоління змушує педагогів підлаштовуватися під все нові й нові потреби. Так, наприклад, покоління 1970-1980-х років було дуже рухливим, а тому традиційний формат навчання, де учні зазвичай сидять на одному місці, був для них вкрай важким.

Сучасне ж покоління, назване “поколінням Z” (Generation Z, Gen Z), багато в чому відрізняється від попередніх, а тому виникає потреба у створенні новітніх підходів до викладання. Одним з таких підходів і стали дидактичні ігри.

Справедливо відмітити, що власне дидактичні ігри як такі не є абсолютно новітнім методом. Однак впродовж довгого періоду часу його вважали таким, що призначений виключно для дітей. Через це його використання в старшій школі не було популярним. Втім покоління Z переконує педагогів, що настав час по-новому поглянути на цю методику.

**Актуальність дослідження** поглядає в аналізі історії дидактичних ігор та подібних до них методик, а також у створенні методичного посібника, що враховує сучасний досвід використання дидактичних ігор, особливо в старшій школі та ВНЗ.

**Об'єкт дослідження** - дидактичні ігри, а також методики, що наближені до дидактичних ігор, хоч такими насправді не є.

**Предметом дослідження** є історія використання дидактичних ігор та подібних методик від давнини до сучасності.

**Мета роботи** - створити посібник з викладання англійської мови учням старших класів та перших курсів ВНЗ, що враховує дидактичні потреби покоління Z, зокрема потребу в геймифікації освіти.

**Завдання дослідження**:

1. Дослідити історію використання дидактичних ігор та ігрових методик впродовж історичного розвитку методики;
2. Проаналізувати методичні потреби покоління Z, приділяючи особливу увагу потребам, що пов'язані з геймифікацією освіти;
3. Дослідити досвід використання ігор вітчизняних та зарубіжних колег;
4. Запропонувати механізм використання дидактичних ігор та ігрових методик для викладання англійської мови учням старших класів та перших курсів ВНЗ з урахуванням необхідного балансу між грою та навчанням;
5. Створити методичний посібник для викладання англійської мови за описаним механізмом;
6. Розробити сайт для інтерактивного супроводу методичного посібника.

Звіт з проходження комп’ютерно-лінгвістичної практики складається з таких **структурних частин**: вступ, чотири розділи, висновки та список використаних джерел.

У вступі зазначено актуальність, мету та завдання комп’ютерно-лінгістичної практики.

У першому розділі описано концепцію дипломної роботи.

Другий розділ присвячений аналізу інформаційних ресурсів, а саме посібників та підручників, що стали підґрунтям написання магістерської дипломної роботи.

У третьому розділі описується процес дослідження та вибору програмних засобів та інформаційних систем для реалізації методичного посібника.

У четвертому розділі наведено приклади практичного застосування засобів автоматизації обробки інформації.

У результаті проходження практики було створено паперовий посібник та вебсайт для навчання англійської мови старших школярів та студентів перших курсів з використанням ігрових методик.

Для виконання дослідження використовувалися такі **методи**: критичний аналіз літератури, феноменологічний, системний методи, а також прийоми порівняння та синтезу.

**Теоретична значущість** роботи полягає в систематизації та описі наявної інформації щодо дидактичних ігор та їх використання в освіти, зокрема при роботі з Gen Z.

**Практична значущість** полягає в розробці методичного посібника для роботи з представниками нового покоління, який включатиме дидактичні ігри як один з головних прийомів.

**РОЗДІЛ 2. ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНИХ РЕСУРСІВ**

В процесі виконання роботи, був проведений критичний аналіз літератури. Загально, усі досліджені джерела можна розділити на дві категорії:

* Джерела, що стосуються дидактичних ігор як таких
  + Сюди входять роботи таких авторів: Бровкіна, Вейландє, Касьянов, Ковальова, Онищенко, Цісковська, Govindarajan, Liu, Titone та ін.
* Джерела, які описують дидактичні потреби Gen Z
  + Це дослідження таких авторів: Цісковська, Alruthaya, Demir, Schieger, Williams.

Також були проаналізовані роботи авторських колективів, наприклад, статті вчених Чиказького університету.

Загалом було виявлено, що тематика, яку описує дипломна робота, майже не висвітлена в українському науковому середовищі. Зазвичай усі роботи зводяться до опису досвіду західних колег, а в сфері використання дидактичних ігор навіть постанови МОН суперечать самі собі.

**РОЗДІЛ 3. АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ТА ІС**

Для створення власне електронного посібника, були використані засоби Microsoft Word та Google Docs. Кожен із цих текстових редакторів має свої переваги та недоліки. Розгляньмо їх детальніше.

**Microsoft Word.** Стандартний текстовий процесор, яким користується більшість людей. Однією з основних його переваги є те, що він, як і інші елементи сім’ї Microsoft Office є попередньо встановленим на всіх комп’ютерах під керуванням ОС Windows. Цей текстовий редактор інтуїтивно зрозумілий для користувачів, а отже не потребує великої кількості часу на ознайомлення. Редактор дозволяє не просто редагувати текст, а й налаштовувати його стилі та зовнішній вигляд. Також є можливість вставити в текст таблиці, зображення або повноцінні фотографії. За допомогою використання макросів (вбудована функція текстового редактора) можна створювати певні програми, що в свою чергу означає можливість створення інтерактивних посібників.

**Google Docs.** Одним із найголовніших недоліків цього текстового редактора є його природа. Оскільки це фактично сайт, він залежить від підключення до мережі Інтернет. Однак, на відміну від описаного вище MS Word, Google Docs дозволяє одночасно кільком користувачам працювати над одним документом, що значно пришвидшує виконання масштабних проєктів. Крім того, цей текстовий редактор дозволяє ділитися створеними файлами з іншими користувачами з або без надання їм права на редагування вихідного документа. Незважаючи на це, згаданий редактор має більше обмежень в питаннях форматування тексту порівняно з MS Word.

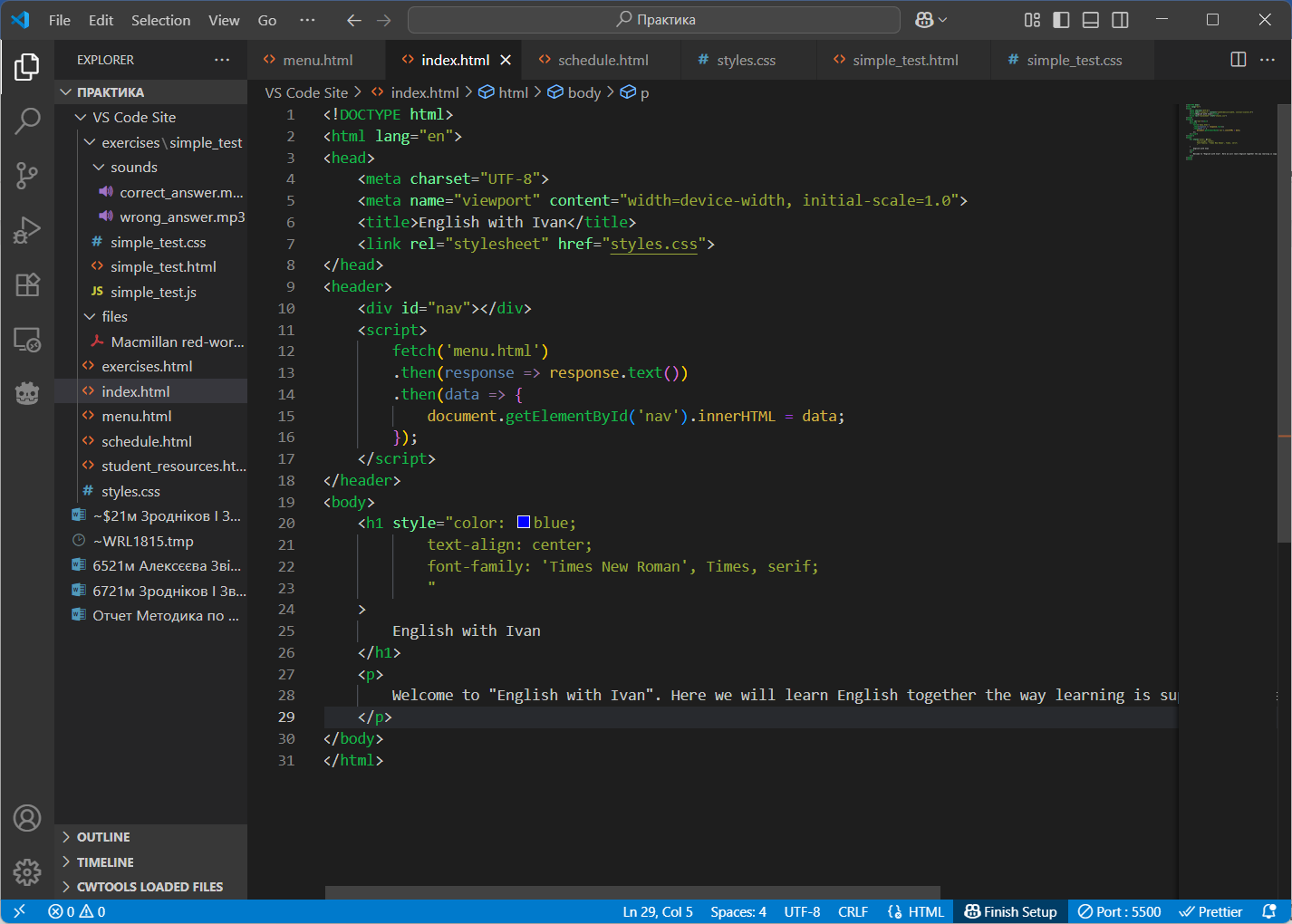
Оскільки процес створення інтерактивного електронного посібника включає в себе створення сайту, розглянемо також інструменти, які дозволяють це зробити.

**Google Sites.** Як і Google Docs, є елементом родини Google, а отже одного облікового запису достатньо для створення як документів, так і сайтів. Редактор візуально зрозумілий, оскільки створення сайту представлено у вигляді конструктора. Є можливість створювати сайти з кількома сторінками, за навігацію між якими відповідає спеціальний елемент, який створюється автоматично. Оскільки цей редактор, як уже було сказано, належить до родини Google, він дозволяє органічно інтегрувати всі інші елементи цієї родини (Документи, Таблиці, Форми тощо) без використання сторонніх ресурсів. Також є можливість інтегрувати файли з Диску, і вони будуть доступні для перегляду безпосередньо з сайту (хоч можливість перейти на Диск все ж залишається). Однак конструктор не пропонує використання анімацій для елементів або звуків, які мають вмикатися, наприклад, при натисканні на певний елемент.

**Canva.** Строго кажучи, цей сайт здебільшого є редактором *дизайнів*. Однак у нього є функціонал створення сайтів, тож цей редактор також є кандидатом на розгляд. Як редактор дизайнів, сайт пропонує велику кількість різноманітних опцій для тих чи інших елементів – як платних, так і безкоштовних. Також є можливість використати ШІ для створення потрібних елементів за текстовим описом – але кількість використань обмежена в безкоштовній версії. На відміну від Google Sites, Canva має спеціальні додатки, які дозволяють додати до дизайнів (у т. ч. й до сайтів) звуки – як стандартні, так і власну озвучку або створену ШІ. Як недолік цього редактора можна винести необхідність реєстрації на ньому.

**Ручне створення.** Цей метод найбільш гнучкий, але при цьому і найбільш затратний в плані часу та зусиль. Технічно, писати сайти можна в будь-якому текстовому редакторі, який підтримує використання HTML (HyperText Markup Language), але найзручнішим на думку автора є Visual Studio Code. Згаданий редактор також є продуктом компанії Microsoft, і при цьому є безкоштовною. Власне, ця програма є редактором коду – що включає будь-яку мову програмування. Завдяки чисельним додаткам, часто створеним самими користувачами, згаданий редактор є не тільки ефективним, але й візуально приємним для людського сприйняття.

Першою з корисних функцій цього редактора є його адаптивність. Щоб зрозуміти це, достатньо пройти перших кілька кроків у підготовці до роботи, а саме відкрити редактор та вибрати потрібну папку на диску. Після цього, безпосередньо засобами самого редактора, можна створювати нові файли та папки. Причому при створенні файлу потрібно вказати не тільки його назву, але й розширення, тобто формат. Оскільки сайти створюються мовою HTML, створимо файл розширення .html (зазвичай головна сторінка сайту описується у файлі index.html). Одразу ж після створення, програма автоматично відкриває новий файл, а в правому нижньому куті вікна програми можна побачити мову програмування, яка в ньому використовується – в нашому випадку це HTML. Якщо у користувача вже встановлене відповідне розширення для VS Code, будь-який новий код буде автоматично виділятися різним кольором, залежно від його призначення (мал. 1).



Мал. 1. Автоматичне розфарбовування (color-coding)

Ще однією корисною функцією VS Code є можливість «спростити собі життя» завдяки використанню вбудованих підказок. Зазвичай під цим мається на увазі автоматичне дописування коду завдяки клавіші TAB. Причому якщо є відкриваючий та закриваючий код, редактор автоматично створить закриваючий код за користувача. Достатньо просто вписати решту інформації між двома розділителями. Однак у випадку з HTML є ще одна перевага.

Якщо уважно роздивитися Мал. 1, можна побачити досить велику кількість коду на початку документу в розділі head. Це обов’язковий код, який пояснює браузеру, що це за документ і як його слід читати. Однак цього коду досить багато. Звісно, його можна написати вручну, але редактор пропонує кращий варіант. Якщо в порожньому .html документі написати html:5 та натиснути Enter, редактор автоматично напише такий код:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

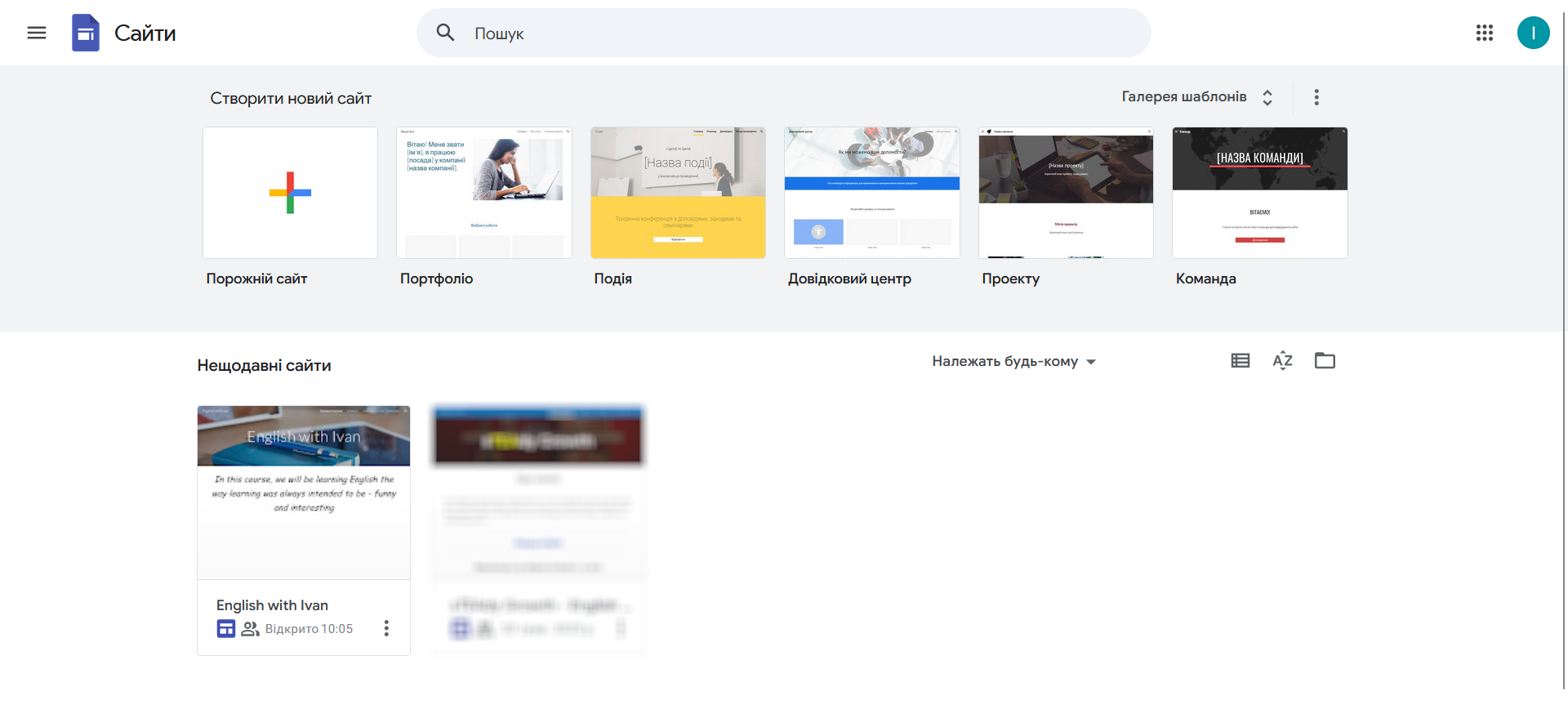
</html>

Технічно, його вже достатньо для створення хоч і порожнього, але все ж робочого сайту. Варто також звернути увагу, що та частина коду, яка належить до розділу head, зсунута вправо порівняно з рештою коду. В термінології програмістів, це називається indentation (абзацний відступ), і, як видно з коду, його дуже зручно використовувати на позначення ієрархії.

**РОЗДІЛ 4. ПРАКТИЧНЕ ЗАСТОСУВАННЯ ЗАСОБІВ АВТОМАТИЗАЦІЇ ОБРОБКИ ІНФОРМАЦІЇ**

Розглянемо детальніше процес створення та публікації сайтів при використанні кожного з описаних підходів.

Для створення сайту за допомогою платформи **Google Sites**, спершу потрібно створити обліковий запис Google. Після цього, потрібно перейти на сайт платформи (<https://sites.google.com/>) та вибрати один із шаблонів або створити порожній сайт (мал. 2).



Мал. 2. Головна сторінка платформи Google Sites

Після вибору дизайну, відкривається сторінка налаштування сайту. Тут можна налаштувати тему сайту, якщо це не було зроблено раніше, а також додати різні елементи майбутнього сайту, такі як текст чи кнопки. Для створення нової сторінки сайту, достатньо перейти на вкладку «Сторінки» та створити нову, зазначивши її назву. У вікні перегляду та редагування сторінки сайту, можна побачити, що в правому верхньому куті з’явилися назви сторінок – це вікно навігації. На кінцевому сайті вони перетворяться на повноцінні посилання.

Як уже було зазначено вище, Google Sites добре інтегрований з іншими сервісами Google. Так, зокрема, платформа дозволяє органічно вставити на сайт документ/таблицю/форму, яка існує на активному акаунті. Для цього достатньо вибрати потрібний елемент та вибрати бажаний документ/таблицю/форму з Диску. При вставленні документа, він буде інтерактивний на сайті, але також буде кнопка переходу на перегляд файлу безпосередньо з Диску (мал. 3).



Мал. 3. Кнопка переходу до перегляду файлу на Диску

Після створення вебсайту, потрібно натиснути на кнопку Опублікувати та виконати інструкції на екрані. Якщо в подальшому потрібно буде зайти на створений сайт, потрібно натиснути на стрілку біля слова Опублікувати та вибрати необхідний пункт випадного меню.

Посилання на вебсайт, створений цим способом: <https://sites.google.com/view/english-with-ivan/%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BD%D0%B0-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0>

Незважаючи на те, що **Canva** в першу чергу є редактором дизайнів, про що вже було зазначено вище, цей сервіс також надає можливість створювати вебсайти. Для того щоб створити сайт у Канві, потрібно спершу зареєструватися на сайті. Після цього потрібно вибрати відповідний елемент на сторінці створення дизайнів.

Після переходу на сторінку створення сайту, Канва запропонує вибрати з набору готових шаблонів. Після цього можна налаштувати сайт за допомогою запропонованих інструментів. Однак важливо враховувати, що оскільки Канва не підтримує інтерактивність на сайтах, для створення інтерактивного тесту, потрібно створювати окрему сторінку не тільки для кожного запитання, але і для кожної відповіді.

Тим не менш, сервіс надає можливість відносної інтерактивності в тому сенсі, що до кожного елемента сторінки можна прикріпити посилання. Це дозволяє, наприклад, зробити кнопку «Домашня сторінка» у вигляді будиночка. Однак оскільки кожна сторінка вважається окремим елементом, цю кнопку потрібно заново налаштовувати для кожної сторінки.

До недоліків сервісу можна віднести відсутність можливості інтегрувати документи. Якщо потрібно, наприклад, поділитися файлом Google Диску, потрібно надсилати його як посилання. Це означає, що користувачі сайту будуть змушені переходити на сторонній ресурс (в цьому випадку – Google Диск) аби переглянути документи.

Посилання на вебсайт, створений у Канві: <https://english-with-ivan.my.canva.site/>

Розгляньмо **створення сайту вручну**, зокрема за допомогою Visual Studio Code. Вище вже був описаний найкоротший спосіб створити робочий сайт за допомогою вбудованих у програму засобів. Нижче розглянемо які елементи коду потрібно змінити для налаштування сайту.

У стандартному коді для будь-якої веб-сторінки, нам спершу треба звернути на розділ head («голова» сторінки). Тут ми змінимо тільки назву самої сторінки – для цього потрібно змінити назву у графі title. Після цього звернімося до мал. 1, зокрема до розділу body. Це – «тіло» сторінки. На цьому моменті вже можна побачити, що структурні елементи веб-сторінки досить чітко описують чим саме вони є, а тому знання англійської є вкрай корисним елементом при створенні сайтів. Отож у тілі тексту ми можемо побачити два елементи – заголовок (h1, скорочення від header 1 – чим вища цифра, тим меншим буде заголовок) та абзац (p, скорочення від paragraph). Якщо з абзацом все зрозуміло, то у відкриваючому елементі заголовка можна побачити більше тексту.

Це – стилі. Зазвичай стилі визначають в окремому файлі .css (CSS – Cascading Style Sheets). Однак якщо потрібно одноразово визначити стиль певного одиничного елемента, стилі можна визначити безпосередньо в коді веб-сторінки, що й було зроблено в наведеному прикладі.

Представлений код стилізації тексту веб-браузер розумітиме так (порядок рядків відповідний):

* пофарбувати текст синім кольором;
* розташувати текст в центрі сторінки;
* визначити шрифт як Times New Roman.

Те саме можна було б прописати і в окремому файлі styles.css (власне назва файлу може бути будь-яка, однак styles.css є стандартом для більшості веб-сайтів). Для цього потрібно було б визначити заголовок як специфічний клас елементів (наприклад, “class="header"”), після чого у css документі визначити header = { } і прописати особливості форматування всіх елементів цього класу між фігурними дужками.

Однак використання окремого документу для визначення стилів викликає окремий крок у створенні нових документів. За замовчуванням, браузер не знає, що форматування тієї чи іншої сторінки прописано деінде. Для цього в розділі head потрібно прописати:

<link rel="stylesheet" href="styles.css">

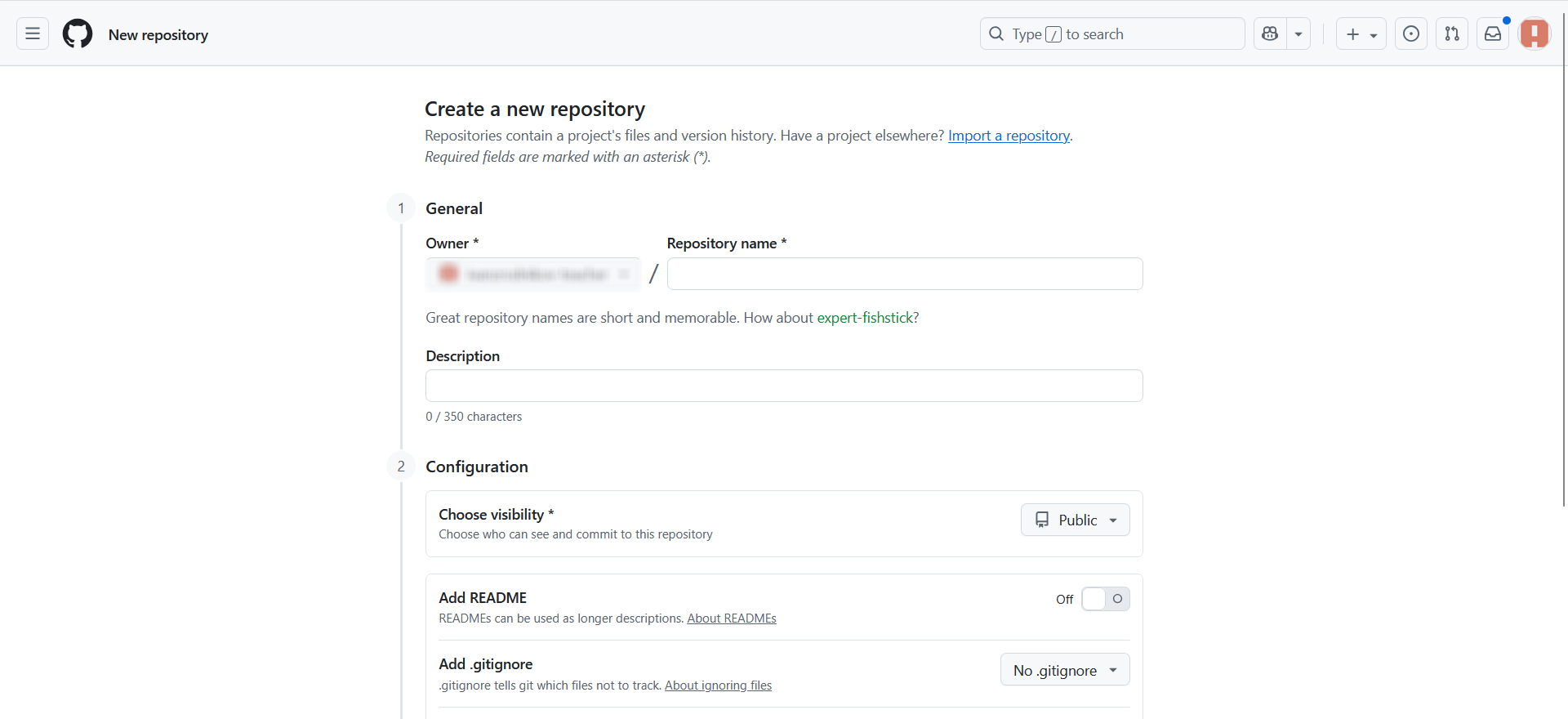
Цей рядок коду фактично говорить браузеру «Стилі для цієї сторінки визначені у файлі styles.css». Власне, значення “href=” є вкрай важливим, оскільки саме завдяки ньому створюють усі посилання (в т. ч. й кнопки). Важливо зазначити, що при створенні нового посилання, шлях рахуватиметься *з того місця, де створене посилання*. Тобто з точки зору ієрархії, посилання можуть рухатися тільки вниз. Це так званий відносний шлях. Але ж часто виникає потреба повернутися на одну, а то й на дві чи більше сходинки назад, тобто вище за ієрархією. Для цього потрібно прописати «../». Якщо поставити одну крапку, то браузер читатиме «Відкрий поточну папку і звідти…». Дві ж крапки означають «Відкрий папку на рівень вище [тобто ту, всередині якої знаходиться поточна] і звідти…». Цей елемент можна використовувати стільки разів, скільки потрібно для повернення у потрібну директорію. Фактично це відповідає натисканню кнопки Назад у вікні файлового провідника комп‘ютера.

Повернімося до пропущеного раніше елемента header. Це – «заголовок» сторінки. Тут важливо розуміти, що з точки зору кінцевого користувача «голови» майже не видно (окрім хіба заголовку сторінки). Натомість «заголовок» - це найвищий елемент сторінки, який бачить користувач. В цьому випадку в цей елемент винесене меню навігації.

Всередині заголовка ми бачимо два головних елементи – div та код JavaScript. Сам по собі елемент div нічого не робить. Це контейнер, який потрібен для зберігання інших компонентів. Код JavaScript фактично бере інформацію з файлу menu.html та вставляє її у контейнер, який був щойно створений елементом div. Це дозволяє редагувати меню тільки в одному файлі, а всі інші сторінки підтягуватимуть його за допомогою цього коду.

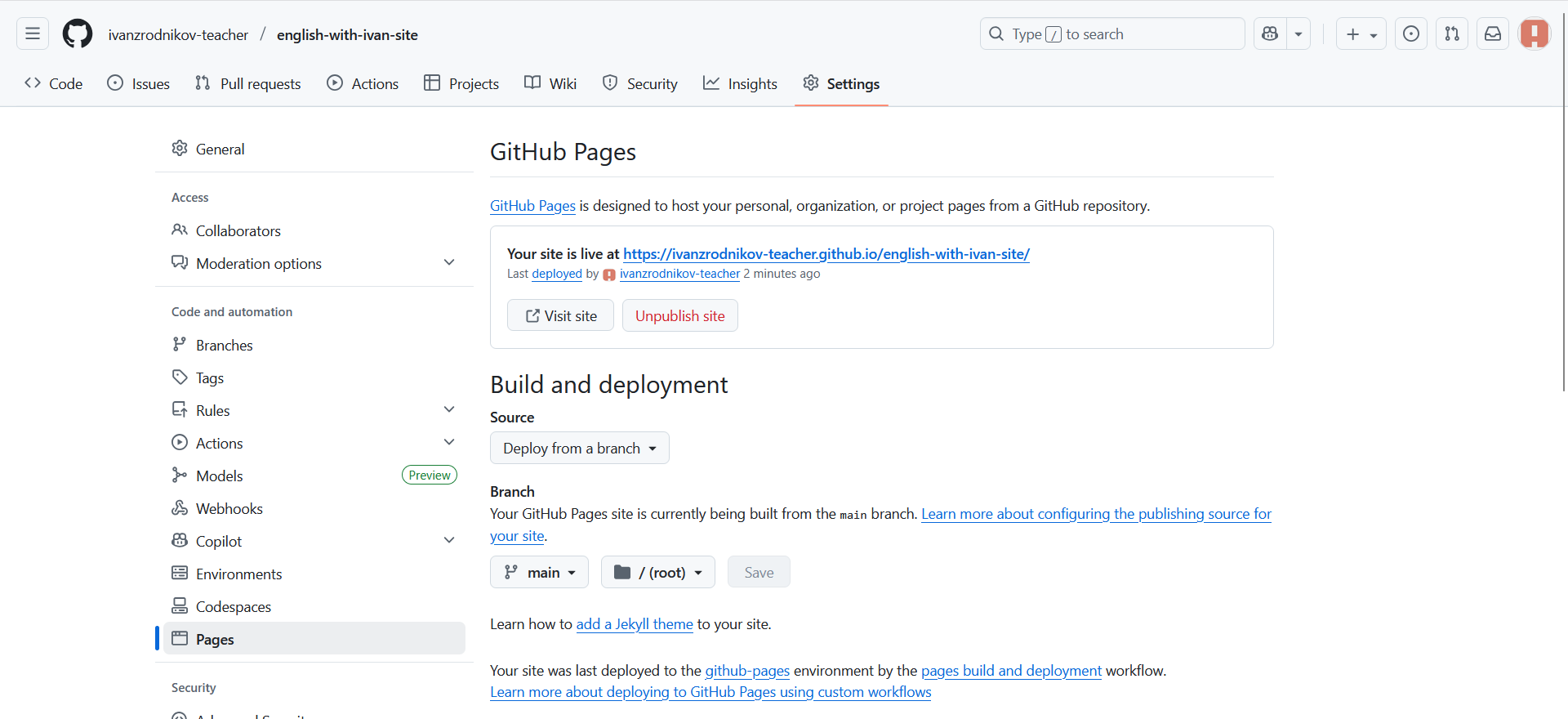
Як можна побачити, перевагою такого методу є його гнучкість. За допомогою інструментів програмного коду, можна реалізувати будь-яку функцію, яку можна знайти в конструкторах сайтів. Крім того, перевагою такого методу є те, що він абсолютно безкоштовний в плані розробки. Єдиним недоліком є його складність в освоєнні, а також час, який потрібно витратити на ручну розробку сайту.

Після створення сайту локально, його потрібно опублікувати. Для цього скористаємося послугами GitHub Pages. Спершу потрібно буде створити акаунт. Після цього натискаємо на кнопку Створити репозиторій та даємо йому назву (мал. 4).



Мал. 4. Інтерфейс створення репозиторію GitHub

Після цього до репозиторію потрібно додати всі файли, які стосуються сайту. Фактично треба повторити файлову структуру відносно головної сторінки сайту. Після цього потрібно зайти в налаштування. **ВАЖЛИВО**: потрібно зайти саме в налаштування *репозиторію*, не облікового запису. Звідти треба вибрати пункт Pages та налаштувати, звідки конкретно буде завантажений сайт. Після очікування на перевірку системою, треба перезавантажити сторінку і з‘явиться повідомлення про успішну публікацію сайту (мал. 5).



Мал. 5. Сайт успішно опублікований

Посилання на створений вебсайт: <https://ivanzrodnikov-teacher.github.io/english-with-ivan-site/>

**ВИСНОВКИ**

Протягом комп’ютерно-лінгвістичної практики були закріплені теоретичні знання та практичні навички з підбору матеріалів, обробки інформації.

Проходження практики дозволило визначити та використати програмні засоби для написання практичної частини дипломної роботи.

Були досягнуті поставлені у вступі завдання:

* описана концепція магістерської дипломної роботи;
* здійснено пошук та застосування інформаційних ресурсів, що відповідають темі та меті дипломної роботи;
* проаналізовані наявні програмні засоби та ІС;
* підібрані відповідні засоби для реалізації мети дипломної магістерської роботи;
* набуті практичні навички роботи з засобами автоматизації обробки інформації.

Отримані знання та навички відіграли важливу роль у професійному розвитку фахівця прикладної лінгвістики.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. ГОСТ 34.601-90 Автоматизированные системы. Стадии создания.- М. ГСП, 1990.
2. Положення про електронні освітні ресурси : Наказ Міністерства освіти і науки України від 29.05.2019 № 749 Верховна Рада України. 1994-2023. С. 3-4.
3. Толочко С.В. Формування екологічної компетентності здобувачів освіти засобами мобільних застосунків // Pedagogical sciences multidisciplinary academic notes. Theory, methodology and practice. – 2022. – С. 684-688.
4. Зубов А.В. Информационные технологии в лингвистике: Учеб. пособие для студ. лингв. фак-тов высш. учеб. заведений / А.В.Зубов, И.И.Зубова — М.: Издательский центр «Академия», 2004. — 208 с.
5. <https://ukraine.ureport.in/story/2152/>
6. <https://naub.oa.edu.ua/analiz-navchalnyh-posibnykiv-dlya-vyv/>