

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота
на здобуття ступеня бакалавра**

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**РОЗРОБКА ВЕБ-СЕРВІСУ «МІКОЛОГІЯ» З ВИКОРИСТАННЯМ
МАШИННОГО НАВЧАННЯ**

Виконав студент 4-го курсу
Шеремет Іван




(підпис)

Науковий керівник:
Доцент, кандидат фіз.-мат. наук
Катеринич Лариса

(підпис)

Засвідчую, що в цій роботі немає запозичень
з праць інших авторів без відповідних
посилань.

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри інтелектуальних
програмних систем
25 травня 2022 р.,
протокол № 10

Завідувач кафедри
Олександр Провотар

(підпис)

РЕФЕРАТ

Обсяг роботи – 26 сторінок, 7 ілюстрацій, 7 джерел посилань.

Тема: Розробка веб-сервісу «Мікологія» з використанням машинного навчання.

Перелік ключових слів: веб-сервіс, сервіс, Python, Flask, Amazon Web Services, машинне навчання, Docker.

Метою роботи є розробка веб-сервісу, який міг би покращити життя користувачів. Предметом роботи є веб-сервіс для розпізнавання отруйних грибів від неотруйних. Для досягнення запланованої мети поставлено такі задачі:

- Аналіз концепції та можливостей для розробки веб-сервісів;
- Аналіз аналогічних веб-сервісів;
- Визначення особливостей для створення веб-сервісу;
- Визначення технологій для реалізації проекту;
- Проектування архітектури веб-сервісу;
- Розробка і проведення тестів веб-сервісу.

Результати роботи: під час виконання кваліфікаційної роботи було розглянуто та досліджено засоби для розробки веб-сервісу, опрацьовано та отримано навички використання бібліотек та технологій для розробки веб-сервісу.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
1. ОСОБЛИВОСТІ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-СЕРВІСУ	7
1.1 Мова програмування Python	7
1.2 Середовище розробки Visual Studio Code	10
1.3 Контейнеризатор додатків Docker	11
1.4 Amazon Web Services	12
1.5 Машинне навчання	15
1.6 Актуальність	16
2. ОПИС ВЕБ-СЕРВІСУ	18
2.1 Концепція веб-сервісу	18
2.2 Основні функції	19
ВИСНОВКИ	25
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	26

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Python – це високорівнева інтерпретована мова програмування загального призначення.

Flask – це мікро-веб-фреймворк, написаний на Python, який прискорює розробку додатків.

Docker – платформа контейнеризації з відкритим кодом.

AWS – хмарна платформа (Amazon Web Services).

МН – машинне навчання (ML).

VS Code – IDE Visual Studio Code.

WSGI – це допоміжна бібліотека для мови програмування Python для додатків Web Server Gateway Interface (WSGI).

Jinja2 – є шаблонним механізмом для мови програмування Python.

ВСТУП

Оцінка сучасного стану об'єкта дослідження або розробки. Інтернет сьогодні – це те, без чого, скоріше за все, не може прожити переважна кількість населення. Ця технологія об'єднує людей з усього світу, з різних частин планети. Мабуть дуже важко знайти людину, яка хоча б раз не скористалася Інтернетом, а ще важче знайти ту, яка взагалі про нього не чула.

У розвинених країнах Інтернет тривалий час домінував у сфері соціальних послуг: наприклад, за допомогою веб-сервісів можна купувати квитки на літак, чи в кіно, оплачувати комунальні послуги, навіть згадуючи український додаток «Дія», в ньому також можна пожертвувати певні кошти до армії, чи отримати певні послуги. Все це позбавляє людей від втрати часу на стояння в чергах, чи пошуку магазинів і тому подібне.

Існує кілька визначень веб-сервісу. Веб-сервісом може бути будь-яка програма, яка має доступ до Інтернету, наприклад веб-сторінка з динамічним вмістом. У більш спеціалізованому розумінні веб-сервіс — це програма, яка представляє відкритий інтерфейс і може використовуватися іншими додатками в Інтернеті. Концепція веб-сервісів реалізована за допомогою набору технологій, стандартизованих World Wide Web Consortium (W3C).

Актуальність роботи та підстави для її виконання. З розвитком технологій люди починають забувати, що за межами їх домівок досі існує небезпека, а саме отруєння грибами. Зараз діти не усвідомлюють, що не кожен гриб, навіть той, що приносять додому батьки насправді є нешкідливим. Не всі батьки також розбираються в грибах. Оскільки існування отруйних грибів на сьогодні ніяк не контролюється людиною, то я пропоную можливість застерегти людей від отруєння. Тільки уявіть, що цілих 20% від отруєних є дітьми, але за допомогою веб-сервісу «Мікологія» можна буде зменшити відсоток отруєних, чи можливо навіть і зовсім знизити його до нуля. У зв'язку зі сказаним, можна вважати, що розробка та застосування веб-сервісів для застереження людей є дуже актуальною темою у суспільстві.

Мета і завдання роботи. З розвитком технологій починає набувати розробка веб-сервісу, який міг би покращити життя користувачів. Предметом роботи є веб-сервіс для розпізнавання отруйних грибів від неотруйних. Для досягнення запланованої мети поставлено такі задачі:

- Аналіз концепції та можливостей для розробки веб-сервісів;
- Аналіз аналогічних веб-сервісів;
- Визначення особливостей для створення веб-сервісу;
- Визначення технологій для реалізації проекту;
- Проектування архітектури веб-сервісу;
- Розробка і проведення тестів веб-сервісу.

Об'єкт, методи й засоби дослідження та розроблення. Об'єктом роботи є процес створення веб-сервісу. В розробці практичної частини передбачалося ознайомлення з концепцією розробки веб-сервісів, дизайном та обґрунтуванням вартості використання певних бібліотек та фреймворків.

При створенні програмного додатку були враховані сучасні тенденції та рекомендації. Окремо, були досліджені статті, в яких оцінювалася популярність того чи іншого додатка, його функціональність, а також розглядалися плюси та мінуси цього ресурсу в практичному використанні.

Можливі сфери застосування. На основі виконаної практичної частини кваліфікаційної роботи можна створювати та модифікувати веб-сервіси з розпізнавання отруйних грибів використовуючи можливості бібліотек та фреймворків. Програмний продукт можна використовувати в майбутньому для розширення та доповнення функціональних можливостей.

1. ОСОБЛИВОСТІ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-СЕРВІСУ

Для розробки додатку було обрано наступні технології:

- Python (Flask) в якості мови програмування
- Visual Studio Code в якості середовища розробки веб-сервісу
- Docker в якості контейнеризатора
- Amazon Web Services в якості хмарної платформи

1.1 Мова програмування Python

Python — це високорівнева інтерпретована мова програмування загального призначення, яка полегшує навчання та не вимагає від вас розуміння деталей комп'ютера, щоб ефективніше розробляти програми. Завдяки його високорівневим вже вбудованим структурам даних разом з динамічною типізацією та динамічним зв'язуванням розробники вважають саме цю мову дуже привабливою для швидкої розробки додатків, а також для використання в якості скриптової мови або мови зв'язків для з'єднання існуючих компонентів. Підкреслює читабельність та знижує витрати на обслуговування саме простий та легкий у засвоєнні синтаксис мови Python. Інтерпретатор Python і величезна стандартна для користувачів бібліотека вільно доступні у вихідному або двійковому форматі для всіх основних платформ і вільно поширюються.

Зазвичай програмістам подобається Python через підвищену продуктивність, яку він забезпечує. Python - це інтерпретована мова. Щоб розробити програму на Python, ви пишете код у файл під назвою «вихідний код». Щоб запустити вихідний код, вам потрібно перетворити його на машинну мову, зрозумілу комп'ютеру. Інтерпретатор Python перетворює вихідний код у машинний код рядок за рядком під час виконання програми написаної на мові Python. Не маючи етапу компіляції цикл редагування, тестування та налагодження є дуже швидким у порівнянні з іншими мовами програмування. Налаштування написаної програми на Python відбувається

дуже легко, бо неправильне введення чи помилка ніколи не спричинить до помилки сегментації. Однак, коли інтерпретатор отримує помилку, то створюється виняток. Якщо додаток не отримує виняток, то інтерпретатор роздруковує трасування стека. Скомпільовані мови, такі як Java та C#, використовують компілятор, який компілює весь вихідний код перед запуском програми.

Python підвищує вашу продуктивність. За допомогою Python ви можете вирішити складні проблеми за менший час і з меншою кількістю рядків коду. Також можна дуже швидко створити прототип на Python. Саме ця високорівнева мова програмування стає рішенням у різних галузях у багатьох сферах, від веб-додатків до науки про дані та машинного навчання.

Python — мова загального призначення. Це означає, що ви можете використовувати Python в різних областях, зокрема:

- Веб-додатки
- Програми для великих даних
- Тестування
- Автоматизація
- Наука про дані, машинне навчання та штучний інтелект
- Настільне програмне забезпечення
- Мобільні програми

Python став невід'ємною частиною науки про дані, дозволяючи усім охочим користуватися цією мовою, наприклад для виконання складних статистичних обчислень, візуалізування даних, розробка алгоритмів машинного навчання та аналізу даних, а також багато іншого, пов'язаного з даними.

Python може створювати різноманітні візуалізації даних, наприклад лінійні графіки та гістограми, кругові діаграми та тривимірні діаграми. За допомогою ряду бібліотек, який програмісти можуть використовувати для

написання писати програм аналізу даних і машинного навчання хутчіше та більш ефективніше, наприклад TensorFlow і Keras.

Keras — це API глибокого навчання, який написаний на мові Python і працює на платформі машинного навчання TensorFlow. Він був розроблений з упором на можливість швидкого експериментування. Якомога швидше перейти від ідеї до результату — це ключ до хорошого дослідження. Keras є простим, але не зовсім. Він зменшує когнітивне навантаження на розробників, щоб вони могли зосередитися на дійсно важливих проблемах. Цей API гнучкий, тобто використовує принцип прогресивного розкриття складності: прості процеси мають бути швидкими і легкими, а розширені процеси мають бути можливими завдяки чіткому шляху, який опирається на тому, що ви вже навчилися. Keras є потужним API, він пропонує провідну в галузі продуктивність та масштабованість.

Flask — це мікро-веб-фреймворк, написаний на Python, який прискорює розробку додатків, надаючи базові внутрішні компоненти для розробки програмістів. Flask є простим і легким – це один з найбільш керованих фреймворків, що містить лише найнеобхідніше для веб-розробки. Однак він також розроблений так, щоб бути дуже розширюваним, що дозволяє розробникам налаштовувати його так, як вони вважають за потрібне.

Flask виник у 2004 році, коли розробник на ім'я Армін Роначер створив його як першоквітневий жарт. Однак він швидко набув популярності у спільноті з відкритим кодом. Таким чином, він перетворився на популярний проект з відкритим кодом і здобув велику кількість прихильників, яких підтримує і сьогодні.

Деякі функції, які роблять Flask ідеальною платформою для розробки веб-додатків:

- Flask надає сервер розробки та налагоджувач.
- Він використовує шаблони Jinja2.
- Він сумісний з WSGI 1.0.
- Він забезпечує інтегровану підтримку модульного тестування.

- Для Flask доступно багато розширень, які можна використовувати для покращення його функціональних можливостей.

Flask вважається мікро-фреймворком, оскільки він легкий і надає лише важливі компоненти для веб-розробки, такі як обробка запитів, маршрутизація, сесії тощо. Flask не потребує спеціальних бібліотек чи інструментів, і він не включає такі компоненти, як перевірка форми, інтеграція бази даних та обробка завантаження, які можуть надати інші фреймворки. Натомість Flask дозволяє вам вибирати, які бібліотеки використовувати, а не змушує вас використовувати попередньо встановлене рішення. Але це не означає, що Flask не є універсальним, а навпаки, він підтримує розширення та користувацькі модулі, які додають функціональні можливості додатків, як ніби вони були вбудовані у Flask.

Мікро-фреймворки є протилежністю фреймворків повного стека, які також пропонують додаткові модулі для таких функцій, як аутентифікація, ORM бази даних, перевірка введення та очищення тощо.

Jinja є шаблонним механізмом для мови програмування Python. Подібно до веб-фреймворку Django, він обробляє шаблони в пісочниці.

Werkzeug (що в перекладі з Німецької – «інструмент») — це допоміжна бібліотека для мови програмування Python для додатків Web Server Gateway Interface (WSGI). Werkzeug може створювати екземпляри об'єктів для запиту, відповіді та допоміжних функцій. Його можна використовувати як основу для спеціального програмного забезпечення та підтримує Python 2.7 і 3.5 і новіших версій.

1.2 Середовище розробки Visual Studio Code

Розширення Visual Studio Code Remote - Containers дозволяє використовувати контейнер Docker як повнофункціональне середовище розробки. Це дозволяє відкривати будь-яку папку всередині (або змонтовано в) контейнері та використовувати повний набір функцій Visual Studio Code.

Файл `devcontainer.json` у вашому проєкті повідомляє VS Code, як отримати доступ (або створити) до контейнера розробки за допомогою чітко визначеного інструменту та стека часу виконання. Цей контейнер можна використовувати для запуску програми або для окремих інструментів, бібліотек або середовищ виконання, необхідних для роботи з кодовою базою.

Файли робочої області монтується з локальної файлової системи або копіюються або клонуються в контейнер. Розширення встановлюються та запускаються всередині контейнера, де вони мають повний доступ до інструментів, платформи та файлової системи. Це означає, що ви можете легко перемикати все середовище розробки, просто підключившись до іншого контейнера. Приклад дивитися на рисунку 1.

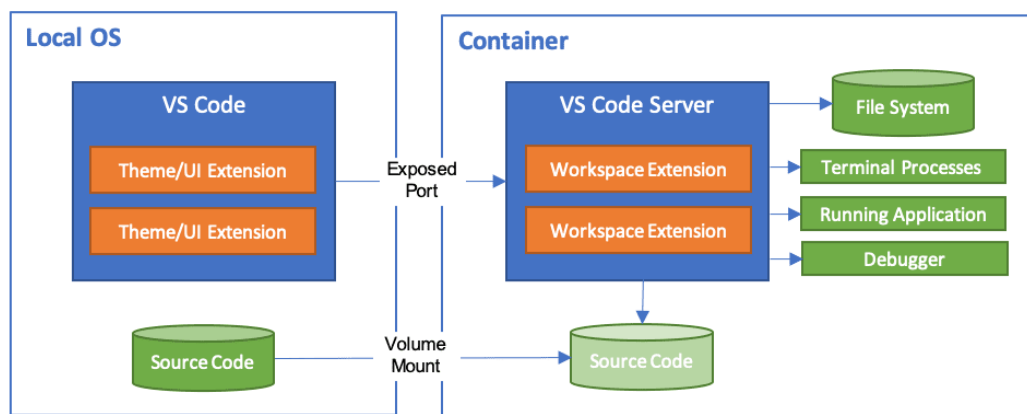


Рис. 1. VS Code Container

Це дозволяє VS Code надавати локальний досвід розробки, включаючи повний IntelliSense (завершення), навігацію по коду та налагодження незалежно від того, де розташовані ваші інструменти (або код).

1.3 Контейнеризатор додатків Docker

Технологія контейнерів Docker була відправлена у світ в 2013 році як Docker Engine з відкритим вихідним кодом.

Контейнери або контейнери Linux – це технологія, яка дозволяє нам ізолювати певні процеси ядра і змусити їх повірити, що вони єдині, які працюють на абсолютно новому комп'ютері.

На відміну від віртуальних машин, контейнер може спільно використовувати ядро операційної системи, завантажуючи лише різні двійкові файли/бібліотеки. Іншими словами, вам не потрібно встановлювати зовсім іншу операційну систему (так звану гостьовою) у вашій операційній системі хоста. Ви можете запускати кілька контейнерів в одній операційній системі без встановлення кількох різних гостьових операційних систем.

Docker упаковує програму та всі її залежності у віртуальний контейнер, який може працювати на будь-якому сервері Linux. Тому ми називаємо їх контейнерами. Тому що вони містять усі необхідні залежності в одному програмному забезпеченні.

Docker складається з таких елементів:

- демон, який використовується для створення, запуску та керування контейнерами;
- API високого рівня, який дозволяє користувачеві спілкуватися з демоном;
- і CLI, інтерфейс, який ми використовуємо, щоб розкрити все це.

Контейнери Docker, які працюють на Docker Engine:

- **Стандартний:** Docker створив галузевий стандарт для контейнерів, щоб їх можна було переносити куди завгодно;
- **Легкий:** контейнери мають спільне ядро ОС машини і тому не вимагають ОС для кожної програми, що підвищує ефективність сервера та знижує витрати на сервер і ліцензування;
- **Безпечний:** програми безпечніші в контейнерах, а Docker надає найпотужніші можливості ізоляції за замовчуванням у галузі.

1.4 Amazon Web Services

Amazon Web Services (AWS) — це безпечна платформа хмарних сервісів, яка забезпечує обчислювальну потужність, зберігання бази даних, доставку вмісту

та інші функції, які допомагають підприємствам масштабуватися та розвиватися.

Простіше кажучи, AWS дозволяє:

- Запуск веб-серверів і серверів додатків у хмарі для розміщення динамічних веб-сайтів.
- Надійно зберігайте всі свої файли в хмарі, щоб ви могли отримати до них доступ з будь-якого місця.
- Використання керованих баз даних, таких як MySQL, PostgreSQL, Oracle або SQL Server для зберігання інформації.
- Швидко доставляйте статичні та динамічні файли по всьому світу за допомогою мережі доставки вмісту (CDN).

Amazon Web Services надає послуги з десятків центрів обробки даних, розташованих у зонах доступності (AZ) у регіонах по всьому світу. AZ — це сайт, який містить кілька фізичних центрів обробки даних. Регіон – це набір AZ в географічній близькості, які з'єднані мережевими з'єднаннями з низькою затримкою.

Компанія обирає одну або кілька зон доступності з різних причин, напр. В. Відповідність і близькість до кінцевих клієнтів. Наприклад, клієнт AWS може розгорнути віртуальні машини (VM) і реплікувати дані в різних AZ, щоб створити високонадійну інфраструктуру, стійку до збоїв окремого сервера або всього центру обробки даних.

Amazon Elastic Compute Cloud (EC2) — це служба, яка надає віртуальні сервери — так звані екземпляри EC2 — для обчислювальної потужності. Служба EC2 пропонує десятки типів екземплярів різної ємності та розмірів, призначених для конкретних типів робочого навантаження та додатків, таких як: В. обчислювальні задачі, що інтенсивні, і прискорені. AWS також надає інструмент автоматичного масштабування для динамічного масштабування ємності для підтримки працездатності та продуктивності екземпляра.

AWS має найбільшу та найдинамічнішу спільноту з мільйонами активних клієнтів і десятками тисяч партнерів по всьому світу. Клієнти практично в кожній галузі та будь-якого розміру, включаючи стартапи, підприємства та організації державного сектору, використовують усі можливі варіанти використання AWS. Партнерська мережа AWS (APN) включає тисячі системних інтеграторів, які спеціалізуються на послугах AWS, і десятки тисяч незалежних постачальників програмного забезпечення (ISV), які адаптують свою технологію для роботи на AWS.

AWS була розроблена як найгнучкіша та найбезпечніша серед доступних сьогодні хмарних обчислень. Основна інфраструктура розроблена з урахуванням вимог безпеки для військових, глобальних банків та інших високочутливих організацій. Це підтримується широким спектром хмарних інструментів безпеки з 230 сервісами та безпекою, відповідністю, службами та керуванням. AWS підтримує 90 стандартів безпеки та відповідних сертифікатів, а всі 117 служб AWS, які зберігають дані клієнтів, надають можливість шифрувати ці дані.

З AWS ви можете використовувати новітні технології, щоб швидше експериментувати та впроваджувати інновації. Ми постійно прискорюємо наш темп інновацій, щоб винаходити абсолютно нові технології, які можна використовувати для трансформації свого бізнесу.

AWS пропонує ряд хмарних служб безпеки, включаючи AWS Identity and Access Management, що дозволяє адміністраторам визначати доступ користувачів до ресурсів і керувати ними. Адміністратор також може створити каталог користувачів за допомогою Amazon Cloud Directory або підключити хмарні ресурси за допомогою служби AWS Directory до існуючого Microsoft Active Directory. Крім того, служба AWS Organizations дозволяє організації налаштовувати політику й керувати ними для кількох облікових записів AWS.

AWS також включає інструменти та послуги, які забезпечують програмне та апаратне шифрування, захищають від атак DDoS, надають

сертифікати рівня безпечних сокетів (SSL) і безпеки транспортного рівня (TLS), а також фільтрують потенційно шкідливий трафік веб-додатків.

1.5 Машинне навчання

Машинне навчання (ML) — це підмножина штучного інтелекту (AI), яка зосереджена на створенні систем, які навчаються або покращують свою продуктивність на основі даних, які вони споживають. Штучний інтелект – це широкий термін, який відноситься до систем або машин, які імітують інтелект людини. Машинне навчання та штучний інтелект часто обговорюються разом, і терміни іноді використовуються як синоніми, але не означають одне й те саме. Важлива відмінність полягає в тому, що, хоча все ML є AI, не весь штучний інтелект є машинним навчанням.

Сьогодні машинне навчання працює навколо нас. Коли ми взаємодіємо з банками, здійснюємо покупки в Інтернеті або користуємося соціальними мережами, алгоритми машинного навчання вступають у дію, щоб зробити наш досвід ефективним, плавним і безпечним. Машинне навчання та пов'язані з ним технології швидко розвиваються, і ми лише починаємо чіпляти поверхню можливого.

Алгоритми є двигунами машинного навчання. Загалом, сьогодні використовуються два основних типи алгоритмів машинного навчання: навчання з наглядом і навчання без нагляду. Різниця між ними визначається тим, як кожен навчається на даних, щоб робити прогнози.

Найбільш часто використовуються алгоритми машинного навчання під керуванням. У цій моделі науковець даних виступає в ролі керівництва, навчаючи алгоритм, які висновки робити. Подібно до того, як дитина вчиться розпізнавати фрукти, запам'ятовуючи їх у книжці з картинками, під час навчання під наглядом алгоритм тренується за допомогою вже позначеного набору даних із заздалегідь визначеними результатами. Прикладами машинного навчання з наглядом є такі алгоритми, як лінійна та логістична регресія, багатокласова класифікація та допоміжні векторні машини.

Машинне навчання без нагляду використовує більш незалежний підхід, коли комп'ютер вчиться розпізнавати складні процеси та закономірності без постійного керівництва людини. Машинне навчання без нагляду передбачає навчання на даних, які не мають міток або певного визначеного результату. Продовжуючи аналогію з навчанням у дитинстві, машинне навчання без нагляду схоже на те, що дитина вчиться розпізнавати фрукти, спостерігаючи за кольорами та візерунками, а не запам'ятовувати назви за допомогою вчителя. Дитина шукатиме схожість між зображеннями та ділить їх на групи, призначаючи кожній групі свій новий ярлик. Прикладами алгоритмів машинного навчання без нагляду є кластеризація k-середніх, аналіз головних і незалежних компонентів, а також правила асоціації.

1.6 Актуальність

Інтернет сьогодні – це те, без чого, скоріше за все, не може прожити переважна кількість населення. Ця технологія об'єднує людей з усього світу, з різних частин планети. Мабуть дуже важко знайти людину, яка хоча б раз не скористалася Інтернетом, а ще важче знайти ту, яка взагалі про нього не чула.

У розвинених країнах Інтернет тривалий час домінував у сфері соціальних послуг: наприклад, за допомогою веб-сервісів можна купувати квитки на літак, чи в кіно, оплачувати комунальні послуги, навіть згадуючи український додаток «Дія», в ньому також можна пожертвувати певні кошти до армії, чи отримати певні послуги. Все це позбавляє людей від втрати часу на стояння в чергах, чи пошуку магазинів і тому подібне.

Не зважаючи на це, з розвитком технологій люди починають забувати, що за межами їх домівок досі існує небезпека, а саме отруєння грибами. Зараз діти не усвідомлюють, що не кожен гриб, навіть той, що приносять додому батьки насправді є нешкідливим. Не всі батьки також розбираються в грибах. Оскільки існування отруйних грибів на сьогодні ніяк не контролюється людиною, то я пропоную можливість застерегти людей від отруєння. Тільки уявіть, що цілих 20% від отруєних є дітьми, але за допомогою веб-сервісу

«Мікологія» можна буде зменшити відсоток отруєних, чи можливо навіть і зовсім знизити його до нуля. У зв'язку зі сказаним, можна вважати, що розробка та застосування веб-сервісів для застереження людей є дуже актуальною темою у суспільстві.

2. ОПИС ВЕБ-СЕРВІСУ

В цьому розділі ми розглянемо основну концепцію нашого веб-сервісу, основні функції та додатковий функціонал.

2.1 Концепція веб-сервісу

Існують різноманітні означення веб-сервісу. Веб-сервісом може бути застосунок, котрий має доступ до Інтернету, наприклад, веб-сторінки з динамічним вмістом. В більш вузькому розумінні, веб-сервіс – це застосунок, що є інтерфейсом та знаходиться у відкритому доступі, і може бути використаний іншими додатками в Інтернеті. Концепція веб-сервісів втілюється за допомогою стеку технологій, котрі стандартизовані World Wide Web Consortium (W3C).

Взаємозв'язок цих технологій можна умовно представити наступним чином. Дивитися рисунок 2.

Технології	Роль технологій
Common Internet Protocols	Середовище обміну повідомленнями
Extensible Markup Language (XML)	Представлення даних
Simple Object Access Protocol (SOAP)	Обмін повідомленнями
Web Services Description Language (WSDL)	Опис можливих сервісів
Universal Description, Discovery, and Integration (UDDI)	Публікація сервісів, пошук сервісів

Рис. 2. Технології веб-сервісу

XML це фундамент для реалізації великої кількості технологій, які пов'язані з веб-сервісами. Для віддаленої взаємодії з веб-сервісами використовується SOAP (Simple Object Access Protocol). SOAP гарантує взаємодію систем, незалежно від їх об'єктної моделі, мови програмування або операційної системи. Інформація передаються за допомогою спеціальних XML документів особливого формату.

Базуючись на визначенні W3C, веб-сервіси - це застосунки, які доступні за протоколами, які є стандартизованими для Інтернет. Немає особливих

вимог, щоб веб-сервіси користувались якимось конкретним транспортним протоколом.

В нашому веб-сервісі ми будемо реалізовувати RESTful API для зв'язку нашого веб-сервісу з іншими додатками, які можна створити на свій лад.

API – це набір визначень і протоколів для створення та інтеграції прикладного програмного забезпечення. Інколи його називають контрактом між постачальником інформації та користувачем інформації, який встановлює вміст, який вимагається від споживача (дзвінок) і контент, який вимагається виробником (відповідь).

Інакше кажучи, якщо ви маєте бажання працювати з системою або комп'ютером для одержування даних або реалізації функцій, API надасть можливість передавати системі ті данні, що ви хочете, для того щоб вона змогла зрозуміти та виконати запити.

Ви маєте змогу розглядати API як посередника між користувачами чи клієнтами та ресурсами чи веб - сервісами, які вони мають намір отримати. Це також засіб для компанії обмінюватися даними та ресурсами, захищаючи інформацію, контроль та аутентифікацію, визначаючи, хто до чого отримує доступ. Ще одна перевага API в тому, що вам немає необхідності знати особливості кешування — як витягується ваш ресурс або звідки він надходить.

REST — це архітектурний стиль чи набір правил, а не якийсь протокол чи стандарт. Розробники API можуть реалізувати REST різними способами. Коли запит клієнта здійснюється через RESTful API, він передає уявлення про стан ресурсу запитувачу або кінцевій точці. Ці дані подаються в якомусь з декількох форматів за допомогою HTTP: Python, JSON (нотація об'єктів Javascript), HTML, XML, PHP або просто текст. JSON є одним з найпопулярніших форматів файлів, незважаючи на свою назву, він ніяк не залежить від мови програмування, а також він має такий формат, який надає змогу читати інформацію як людям, так і машинам.

2.2 Основні функції

Основною функцією веб-сервісу «Мікологія» є розпізнавання чи є отруйним гриб, який надійшов у форматі фотографії від клієнта. Це реалізовано за допомогою використання машинного навчання, а саме керованого навчання з бінарною класифікацією. Тобто є лише два класи на яких навчається наша машина – це отруйні та не отруйні гриби.

За допомогою бібліотек Tensorflow та Numpy вдалося написати програму для машинного навчання.

Перш за все, потрібно створити функцію, яка буде саме навчатися та перенавчатися при збільшенні датасету. Дивитися рисунок 3.

```
def teaching():
    mushrooms_data = ImageDataGenerator(rescale=1./255)
    test_data = ImageDataGenerator(rescale=1./255)

    mushrooms_data = mushrooms_data.flow_from_directory(
        'mushrooms/nn',
        target_size=(128,128),
        batch_size=40,
        class_mode='binary')

    test_data = test_data.flow_from_directory(
        'mushrooms/test',
        target_size=(128,128),
        batch_size=10,
        class_mode='binary')

    model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(128, 128, 3)),
        tf.keras.layers.MaxPooling2D((2, 2), 2),

        tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D((2, 2), 2),

        tf.keras.layers.Conv2D(128, (3, 3), padding='same', activation='relu'),
        tf.keras.layers.MaxPooling2D((2, 2), 2),

        tf.keras.layers.Flatten(),
        tf.keras.layers.Dense(512, activation='relu'),

        tf.keras.layers.Dense(1, activation='sigmoid')
    ])

    print(model.summary())

    model.compile(loss='binary_crossentropy',
                  optimizer=tf.keras.optimizers.Adam(0.001),
                  metrics=['accuracy'])

    model.fit(
        mushrooms_data,
        epochs=30,
        validation_data = test_data
    )

    model.save('mushrooms_model.h5')
```

Рис. 3. Функція навчання

Функція **ImageDataGenerator()** відповідає за створення пакетів даних тензорного зображення із збільшенням даних у реальному часі. В якості аргументу ми використовуємо зменшення масштабу даних. Функція **flow_from_directory()** переймає шлях до каталогу та генерує пакети розширених даних, а саме заплановані розміри 128x128 до яких усі знайдені зображення будуть змінені, розмір пакетів даних та класифікація даних, де «двійкові» будуть 1D двійковими мітками. Робимо це для двох пакетів даних, де **mashrooms_data** відповідає за дані з датасету, а **test_data** для тестових даних.

Далі створюємо шари з нейронів через які будуть проходити наші дані. Перший шар створює ядро згортки, яке згортається з вхідним шаром для створення тензора вихідних даних. Використовуючи цей шар як перший шар у моделі, вводимо аргумент ключового слова **input_shape** для зображень 128x128 RGB. Далі збільшуємо кількість нейронів для кожного наступного шару.

Потім дописуємо визначення точності у відстоках та використовуємо бібліотеку **fit()** яку можна використовувати для навчання наших моделей машинного навчання. Задаємо кількість епох через які наша програма має навчитися відрізняти гриби і зберігаємо отриману модель, щоб використовувати вже готові дані.

```

model = tf.keras.models.load_model('mashrooms_model.h5')

#вказуємо зображення для перевірки
path = f'mushrooms/test/{file}'
img = image.load_img(path, target_size=(128, 128))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)

images = np.vstack([x])
classes = model.predict(images)
print(float(classes[0]))
if classes[0]<0.5:
    print("Фотографія містить не отруйний гриб")
else:
    print("Фотографія містить отруйний гриб")

```

Рис. 4. Функція отримання результату

Далі клієнт фотографує гриб, який його цікавить та надсилає його через веб-сервіс, де вже машинне навчання на вже отриманому досвіді зменшує файл до бажаних розмірів, досліджує та робить висновок, чи то є отруйний гриб, чи ні. Дивитися рисунок 4.

Сама ж форма для клієнтів виглядає наступним чином. Дивитися рисунок 5.

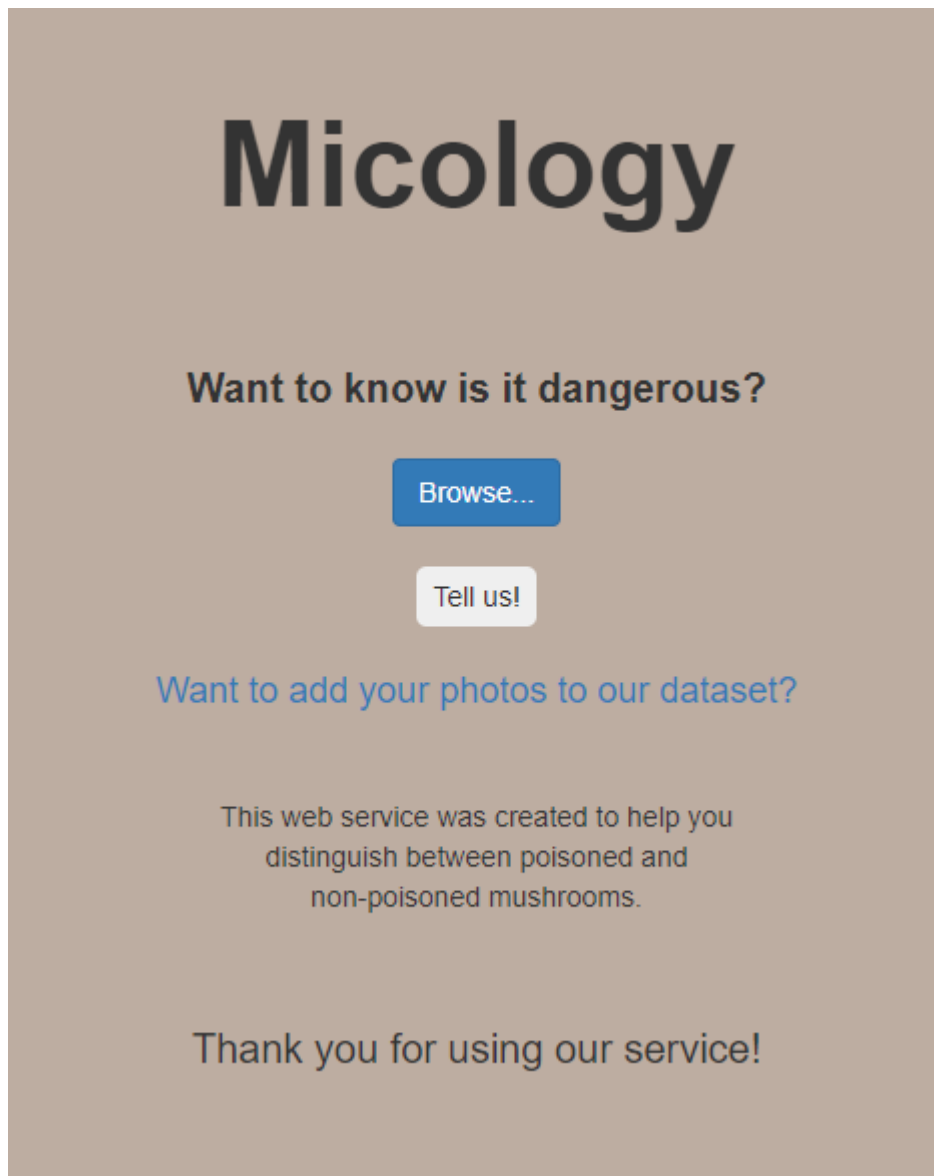


Рис. 5. Головна форма веб-сервісу «Мікологія»

Де нажимаючи на кнопку «Tell us!» клієнту видає результат аналізу. Виглядає це наступним чином. Дивитися рисунок 5.

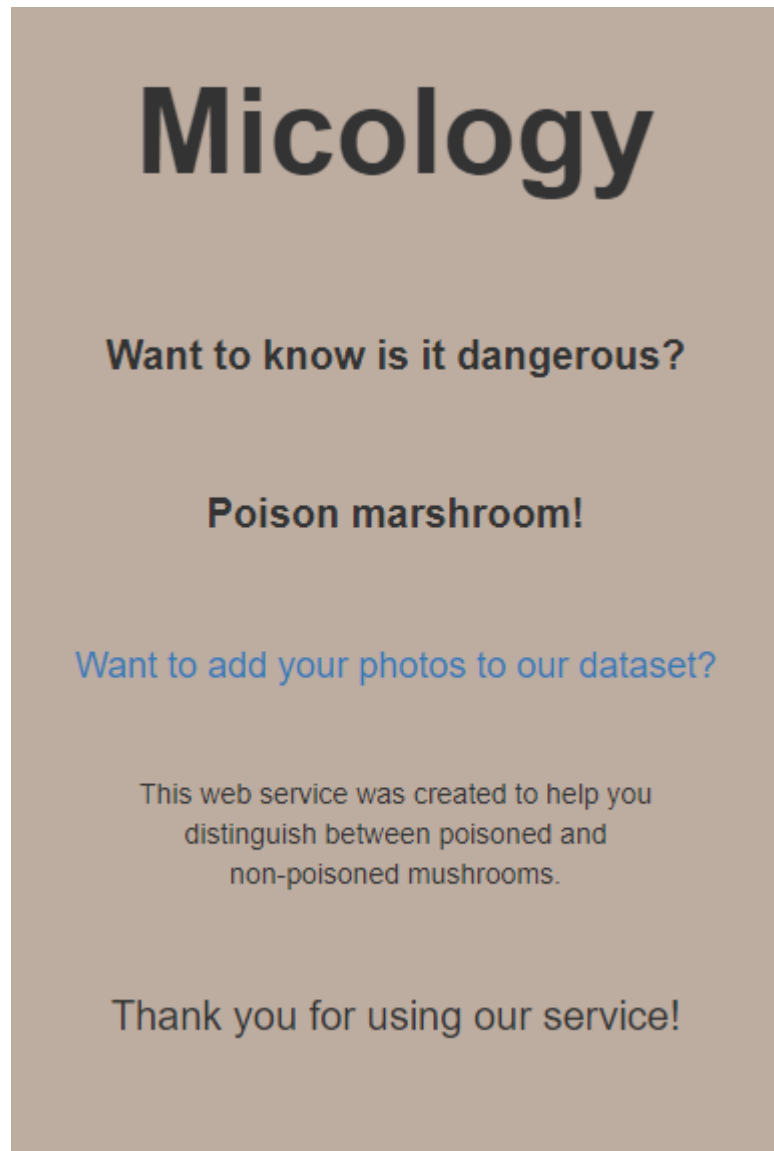


Рис. 5. Форма результату.

Конфігурацію Docker та Docker-compose можна побачити на наступних рисунках 6 та 7.

```
Dockerfile > ...
1 FROM python:3.9-slim
2
3 COPY . /root
4
5 WORKDIR /root
6
7 RUN pip3 install flask gunicorn tensorflow numpy flask_wtf pillow
```

Рис. 6. Конфігурація Docker файлу.

```
Docker-compose.yml
1  version: "3.9"
2  services:
3    flask:
4      build: .
5      environment:
6        - FLASK_APP=hello
7        - FLASK_DEBUG=1
8        - PYTHONUNBUFFERED=True
9      command: gunicorn -b 0.0.0.0:5000 hello:app --reload
10
11     ports:
12       - "5000:5000"
13     volumes:
14       - ./:/root
```

Рис. 7. Конфігурація файлу Docker-compose.

ВИСНОВКИ

Під час написання цієї роботи, було проаналізовано переваги та недоліки інших технологій для створення власного веб-сервісу. Було вирішено розробити веб-сервіс, що допоможе людям розрізнити отруйні від неотруйних грибів тим самим можливо рятуючи чиєсь життя. Використання цього сервісу в наукових цілях, в ігровій індустрії, або ж за власним бажанням може допомогти для досягнення тих, чи інших цілей.

Перш за все, було проаналізовано проблему отруєння грибами та створених певний набір даних з відомих на сьогодні грибів. Наступним кроком, було обрано технології за допомогою яких був написаний веб-сервіс. Python була вибрана, як мова програмування, Flask, як її фреймворк, а середовищем розробки виступав Visual Studio Code, через свою гнучкість та багатофункціональність. В якості контейнеризатора додатків був використаний Docker, додатково можна відзначити те, що він вже інтегрований у VS Code. В якості хмарної платформи був обраний Amazon Web Services через свою простоту, потужність та доступність.

Отже, основні переваги створеного веб-сервіса полягають у своїй простоті використання, зручному та зрозумілому інтерфейсі користувача та швидкості отримання інформації. Авжеж веб-сервіс є відкритим для розширення та модифікування кожним, наприклад можливо збільшити вже існуючий датасет, оптимізувати швидкість навчання програми, створити та додати інший варіант розпізнавання грибів через опис характеристик за допомогою природньої мови, тобто текстом, за допомогою нашого API. В майбутньому веб-сервіс може стати конкурентом для схожих платформ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Офіційна сайт Python [Електронний ресурс] : Документація Python. – Режим доступу: <https://docs.python.org/3/>
2. Офіційна сайт Flask [Електронний ресурс] : Документація Flask. – Режим доступу: <https://flask.palletsprojects.com/en/2.1.x/#>
3. Офіційна сайт Docker [Електронний ресурс] : Документація Docker. – Режим доступу: <https://docs.docker.com/get-started/overview/>
4. Офіційна сайт AWS [Електронний ресурс] : Документація AWS. – Режим доступу:
https://docs.aws.amazon.com/index.html?nc2=h_q1_doc_do_v
5. Wikipedia [Електронний ресурс] : Машинне навчання. – Режим доступу: https://en.wikipedia.org/wiki/Machine_learning
6. Офіційна сайт Visual Studio Code [Електронний ресурс] : Developing inside a Container. – Режим доступу:
<https://code.visualstudio.com/docs/remote/containers#>
7. Machine Learning Mastery [Електронний ресурс] : How Do I Get Started? – Режим доступу: <https://machinelearningmastery.com/start-here/>