

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Козловська Тетяна Анатоліївна
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.81:378

КВАЛІФІКАЦІЙНА РОБОТА

Веб-застосунок для вивчення основ ІІІ
(тема роботи)

122 «Комп'ютерні науки»
(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Козловська Т. А.
(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Николук Ольга Миколаївна
д.е.н., професор

Житомир – 2026

Висновок кафедри комп'ютерних технологій і моделювання систем

за результатами попереднього захисту: допущена

Протокол засідання кафедри комп'ютерних технологій і моделювання систем

№ 20 від «05» червня 2026 р.

Завідувач кафедри комп'ютерних технологій і моделювання систем

к. пед. н., доцент

(науковий ступінь, вчене звання)

(підпис)

КОВАЛЬЧУК М. О.

(прізвище, ім'я, по батькові)

«05» червня 2026 р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистила
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

лаборант кафедри

(науковий ступінь, вчене звання)

(підпис)

КОРОЛЬЧУК В.В.

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Козловська Т. А. Веб-застосунок для вивчення основ ШІ. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2026.

Кваліфікаційна робота спрямована на розробку веб-застосунку для вивчення основ штучного інтелекту (ШІ), призначеного для підвищення ефективності навчання користувачів віком від 12 років шляхом поєднання теоретичних матеріалів, практичних завдань та тестування в єдиному навчальному середовищі.

Практична цінність полягає в наданні користувачам можливості отримувати базові знання з технологій штучного інтелекту. Розроблений веб-застосунок може використовуватися в освітньому процесі закладів освіти та для самостійного навчання. Результати роботи сприятимуть формуванню практичних навичок роботи з технологіями штучного інтелекту завдяки поєднанню теоретичних матеріалів, інтерактивних практичних завдань, тестування та системи контролю прогресу в єдиному навчальному середовищі.

Ключові слова: веб-застосунок, штучний інтелект, машинне навчання, класифікація об'єктів, нейронна мережа.

SUMMARY

Kozlovska, T. A. Web application for learning the basics of AI. – Qualification work on the rights of the manuscript.

Qualification work for a bachelor's degree in the specialty 122 – Computer Science. – Polissia National University, Zhytomyr, 2026.

The qualification work is the result of the development of a web application for learning the basics of AI, designed to enhance the learning effectiveness of users aged 12

and over by combining theoretical materials, practical tasks and testing within a single educational environment.

The practical value of the work lies in providing users with the opportunity to acquire basic knowledge of artificial intelligence technologies. The developed web application can be used in educational institutions as well as for self-study purposes. The results of the work contribute to the development of practical skills in the field of artificial intelligence through the integration of theoretical materials, interactive practical tasks, testing, and a progress monitoring system within a unified learning environment.

Keywords: web application, artificial intelligence, machine learning, object classification, neural network.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	9
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження	9
1.2 Моделювання бізнес-процесів предметної області	11
Висновки до розділу 1	14
РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ	15
2.1 Моделювання інформаційної системи веб-застосунку	15
2.2 Проектування структури бази даних інформаційної системи	19
РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИСТЕМИ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ	21
3.1 Реалізація теоретичних матеріалів та системи тестування	21
3.2 Програмна реалізація практичних завдань вивчення основ штучного інтелекту	23
3.3 Інструкція користувачу інформаційної системи	35
Висновки до розділу 3.....	36
ВИСНОВКИ	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІІІ – штучний інтелект

БД – база даних

ІС – інформаційна система

IDEF0 (англ. Icam DEFinition for Function Modeling) – методологія функціонального моделювання і графічного опису процесів

UML (англ. Unified Modeling Language) – уніфікована мова моделювання

HTML (англ. HyperText Markup Language) – мова гіпертекстової розмітки

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів

KNN (англ. k-nearest neighbours algorithm) – метод К-найближчих сусідів

ВСТУП

Штучний інтелект швидко розвивається, спрощуючи процеси та разом з автоматизацією створює кращі підходи до виконання рутинних задач. Внаслідок цього, змінюються усі сфери людської діяльності, що вимагає швидкого пристосування та неперервного опановувати нових навичок. Саме тому для розуміння основ ШІ стає необхідним для кожного, хто прагне адаптуватися до нових технологій. Одним із інструментів вивчення ШІ за допомогою спеціалізованих веб-застосунків, які полегшують доступ до якісних освітніх ресурсів, а процес навчання робить гнучким. Це дозволяє користувачу ефективно засвоїти теоретичні знання та практичні навички.

Мета роботи полягає у розробці веб-застосунку для підвищення ефективності навчання основам штучного інтелекту, який об'єднує тестування, теоретичні й практичні компоненти навчання в єдиному середовищі для аудиторії від 12 років.

Завдання дослідження:

- проаналізувати предметну область та існуючі платформи для вивчення основ ШІ;
- проектування інформаційної системи вивчення основ штучного інтелекту;
- реалізація веб-застосунка.

Предметом дослідження є методи, засоби, інструментів, що використовуються у процесі проектування, моделювання та реалізації інформаційної системи вивчення основ штучного інтелекту.

Об'єкт дослідження – процес застосування методів, засобів та інструментів проектування, моделювання та реалізації інформаційної системи вивчення основ штучного інтелекту.

У роботі використано сучасні методи дослідження, зокрема порівняльний аналіз наявних онлайн-платформ, проектування інформаційної системи, розробка інтерактивних практичних завдань.

За результатами виконання роботи підготовлено публікації:

1. Козловська Т. А. Технологія штучного інтелекту в публічному управлінні. Безпека, технології, інновації: нові горизонти : зб. праць учасників міжфакультетської наук.-практ. інтернет-конф. здобувачів вищої освіти і молодих вчених, 12 листоп. 2024 р. Житомир : Поліський національний університет, 2024. С. 94-95.

2. Козловська Т. А. Проектування веб-застосунку з вивчення основ штучного інтелекту. Інформаційні технології та моделювання систем : зб. праць учасн. Всеукр. наук.-практ. конф. здобувачів вищої освіти і молодих вчених, м. Житомир, 22 трав. 2026. Житомир : Поліський національний університет, 2026. С. 71-74.

Практична цінність полягає в наданні користувачам можливості отримувати базові знання з технологій штучного інтелекту. Розроблений веб-застосунок може використовуватися в освітньому процесі закладів освіти, для самостійного навчання. Результати роботи сприятимуть формуванню практичних навичок роботи з технологіями штучного інтелекту завдяки поєднанню теоретичних матеріалів, інтерактивних практичних завдань, тестування та системи контролю прогресу в єдиному навчальному середовищі.

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

Предметна область дослідження полягає в онлайн-навчанні в сфері штучного інтелекту. Основним завданням цієї галузі є забезпечення зручної платформи для вивчення теоретичного матеріалу, виконання практичних завдань та тестування у єдиному середовищі. Також забезпечити ефективне, доступне та інтерактивне навчання основам ШІ для початківців та усім зацікавлених користувачів.

Інформаційне забезпечення веб-застосунків передбачає організацію та обробку різних типів даних, таких як вхідні, оброблювані та вихідні дані. До вхідних даних належать навчальні матеріали, завдання для практики, а також дані користувачів, зокрема інформацію про реєстрацію, навчальний прогрес, результати оцінювання. В процесі обробки даних здійснюється аналіз успішності користувачів, проведення тестування. Результатом обробки є вихідні дані, до яких належать результати тестів і практичних завдань, звіт про успішність навчання.

Джерела інформації, що використовується у веб-застосунку є внутрішніми. До них належать база даних навчальних матеріалів, історія виконання завдань користувачами, а саме дані про виконання практичних завдань, результати тестування та динаміку навчального прогресу. Ці дані використовуються для організації навчального процесу, оцінювання знань та формування звітів про успішність.

У сучасній освітній практиці для вивчення основ штучного інтелекту використовуються різні інформаційні системи та онлайн-платформи. Для аналізу існуючих аналогів було обрано такі навчальні платформи: Elements of AI, Hour of code, Google Machine Learning. Дослідження зазначених платформ дає визначити їхні переваги та обмеження, а також виявити можливості для вдосконалення

власної ІС. Результати порівняльного аналізу існуючих платформ для вивчення основ штучного інтелекту наведено в табл. 1.1.

Таблиця 1.1 - Порівняння платформ для вивчення основ штучного інтелекту

Параметр	Elements of AI	Hour of code	Google Machine Learning	Власний продукт
Інтерактивність навчання	Середня	Висока	Висока	Висока
Поєднання теорії, практики і тестування в одному курсі	Часткове	Часткове	Часткове	Повне
Орієнтація на початківців	Висока	Висока	Середня	Висока
Орієнтація на аудиторію 12+	Низька	Висока	Низька	Висока
Зручність інтерфейсу	Середня	Низька	Середня	Висока

Платформа Elements of AI орієнтована на ознайомлення користувачів із базовими концепціями штучного інтелекту та характеризується високою доступністю для початківців. Водночас рівень інтерактивності навчання є середнім, а поєднання теоретичних матеріалів, практичних завдань і тестування реалізоване лише частково. Крім того, платформа переважно орієнтована на дорослу аудиторію.

Hour of Code забезпечує високий рівень інтерактивності завдяки використанню практичних вправ та візуальних елементів навчання. Платформа добре адаптована для початківців та користувачів віком від 12 років, проте навчальні матеріали зосереджені переважно на виконанні окремих практичних завдань і не забезпечують повного поєднання теорії, практики та контролю знань у межах єдиного курсу.

Google Machine Learning надає якісні навчальні матеріали та практичні приклади з машинного навчання, однак орієнтована переважно на користувачів, які вже мають базову технічну підготовку. Незважаючи на високий рівень інтерактивності, курс містить складніший матеріал і не підходить для аудиторії віком від 12 років.

Розроблення веб-застосунка відрізняється від розглянутих аналогів комплексним підходом до навчання. Він поєднує теоретичні матеріали, практичні завдання та тестування в межах єдиного освітнього середовища, що забезпечує

цілісний процес навчання. Веб-застосунок орієнтований на початківців та аудиторію віком від 12 років, має високий рівень інтерактивності та зручний інтерфейс користувача. Це дозволяє підвищити доступність навчання основам штучного інтелекту та забезпечити більш ефективне засвоєння матеріалу.

Отже, результати порівняльного аналізу підтверджують доцільність розробки власного веб-застосунку, який усуває виявлені недоліки існуючих платформ та забезпечує комплексне навчання основам штучного інтелекту в єдиному середовищі.

1.2 Моделювання бізнес-процесів предметної області

На етапі проектування було здійснено структурно-функціональне моделювання, яке дало змогу детально відстежити виконання всіх бізнес процесів інформаційної системи та провести аналіз можливих шляхів покращення її функціонування.

На рис. 1.1 представлена контекстна діаграма IDEF0 для роботи веб-застосунку для вивчення основ ШІ.

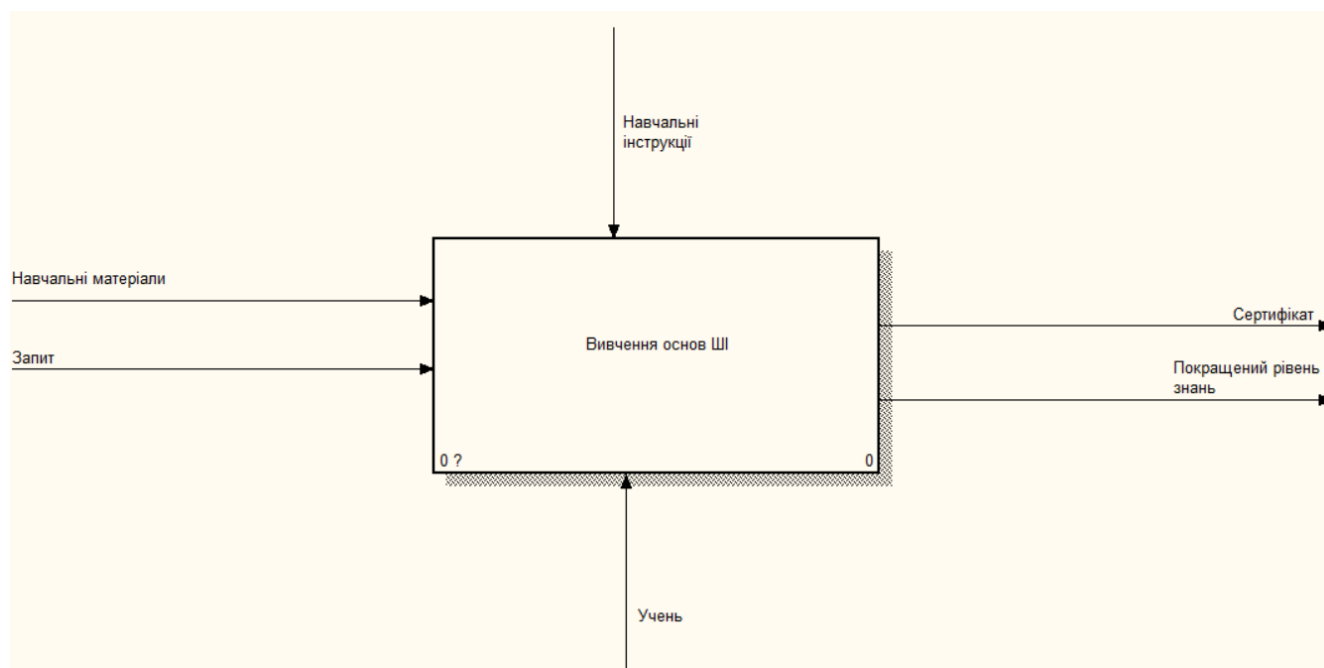


Рисунок 1.1 – Діаграма IDEF0

Для роботи веб-застосунку для вивчення основ ШІ на вході системи навчальні матеріали, запит. Управліннями є навчальні інструкції. Механізмами виконання роботи є учень. На виході ми отримуємо сертифікат та покращений рівень знань.

Діаграма декомпозиції допомагає ширше описати роботи, які протікають у системі. На першому рівні декомпозиції розбивається на 4 блоки: оцінити поточні знання, формування плану навчання, навчання, надання сертифіката. На рис. 1.2 зображений перший рівень декомпозиції.

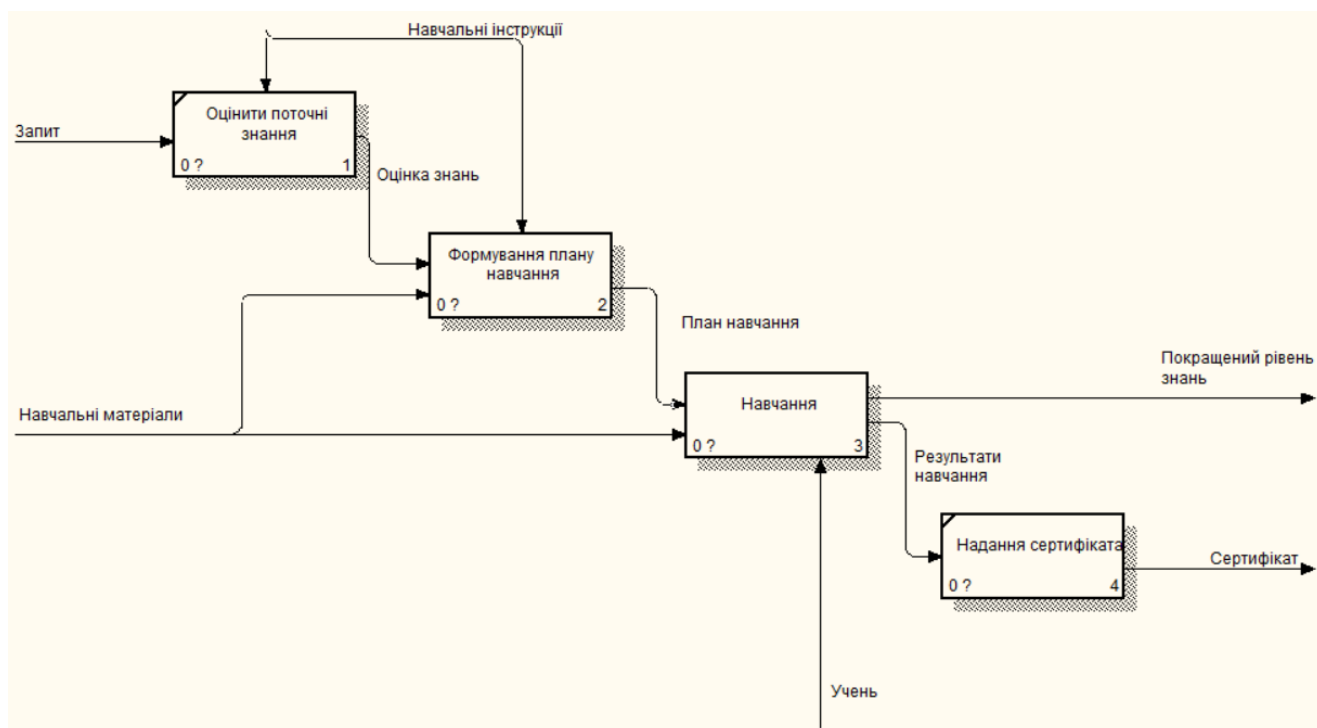


Рисунок 1.2 – Перший рівень декомпозиції діаграми IDEF0

Для блоку «Оцінити поточні знання» необхідні такі входні дані, як запит. У результаті виконання цього етапу маємо оцінку знань. Дугою управління є навчальні інструкції.

Для блоку «Формування плану навчання» необхідні такі входні дані, як оцінка знань та навчальні матеріали. У результаті виконання маємо план навчання. Управліннями виступають – навчальні інструкції.

Для блоку «Навчання» необхідні такі входні дані, як план навчання та навчальні матеріали. У результаті виконання маємо покращений рівень знань та результати навчання. Механізмами виступають – учень.

Для блоку «Надання сертифіката» необхідні такі вхідні дані, як результати навчання. У результаті виконання маємо сертифікат.

Роботу «Навчання» декомпозовано на ще один рівень. Результати декомпозиції зображено на рис. 1.3.

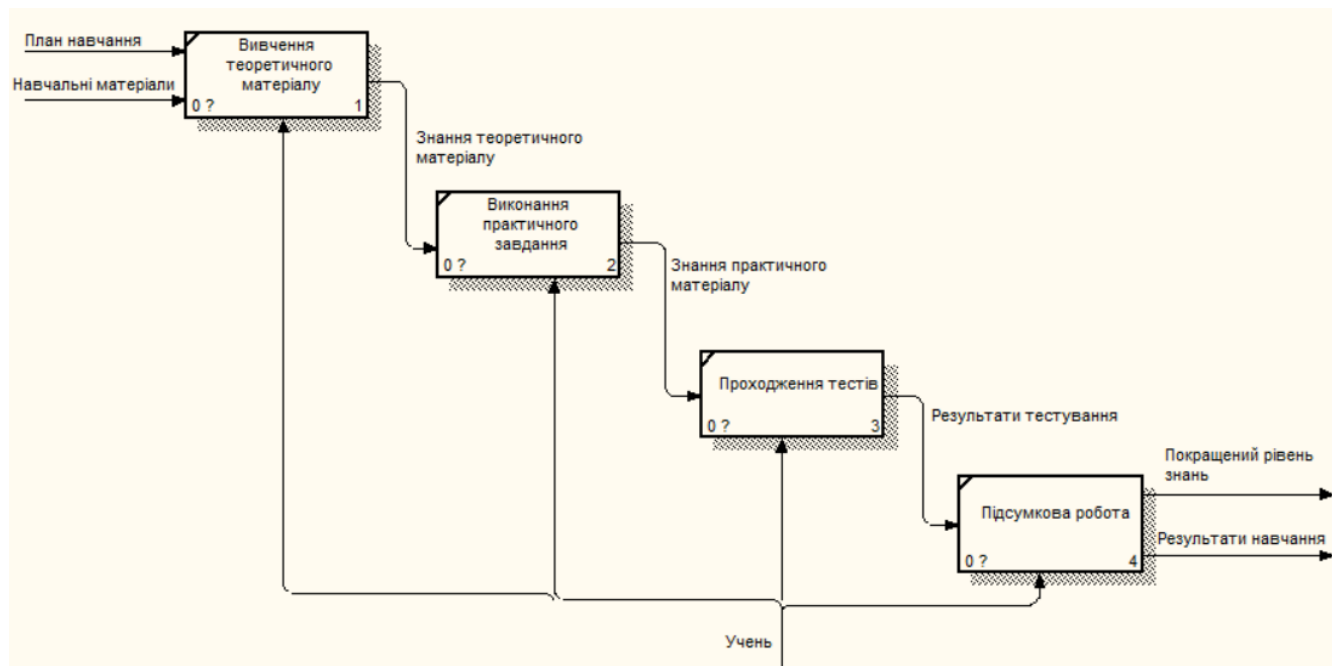


Рисунок 1.3 – Другий рівень декомпозиції діаграми IDEF0

Для блоку «Вивчення теоретичного матеріалу» необхідні такі вхідні дані, як план навчання та навчальні матеріали. У результаті виконання маємо знання теоретичного матеріалу. Механізмом виступає учень.

Для блоку «Виконання практичного завдання» необхідні такі вхідні дані, як знання теоретичного матеріалу. У результаті виконання маємо знання практичного матеріалу. Механізмом виступає учень.

Для блоку «Проходження тестів» необхідні такі вхідні дані, як знання практичного матеріалу. У результаті виконання маємо результати тестування. Механізмом виступає учень.

Для блоку «Підсумкова робота» необхідні такі вхідні дані, як результати тестування. У результаті виконання маємо покращений рівень знань та результати навчання. Механізмом виступає учень.

Висновки до розділу 1

У першому розділі було проведено аналіз предметної області та обґрунтовано актуальність створення веб-застосунку для вивчення основ штучного інтелекту. Досліджено інформаційні потреби користувачів, виконано аналіз існуючих навчальних платформ та визначено їх переваги і недоліки. Також змодельовано основні бізнес-процеси системи, що стало основою для подальшого проєктування та розробки веб-застосунку.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ-ЗАСТОСУНКУ ДЛЯ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Моделювання інформаційної системи веб-застосунку

Була розроблена діаграма прецедентів, наведеній на рисунку А.1, що дозволяє уявити типи ролей та їх взаємодію з системою. Діаграма складається з 4 об'єктів: актор, прецедент, система та зв'язок. Акторами виступають користувач та система. Користувач – це людина, що користується веб-застосунком з метою вивчення основ штучного інтелекту, а система – веб-застосунок для вивчення основ штучного інтелекту, що забезпечує обробку запитів користувача, надання навчальних матеріалів та збереження результатів навчання.

До основних прецедентів, ініційованих користувачем належать:

- Введення особистих даних та вхід до кабінету;
- Проходження початкового тестування;
- Вивчення теоретичного матеріалу;
- Виконання практичного матеріалу;
- Виконання тестування;
- Надсилання відповідей;
- Перевірка прогресу.

Зі свого боку система реалізує наступні функції:

- Забезпечення входу користувача до особистого кабінету;
- Надання тестів для початкового тестування;
- Надання теоретичного матеріалу;
- Надання практичних завдань;
- Надання тестів;
- Перевірка результатів;
- Збереження прогресу користувача.

Зв'язки між прецедентами користувача та системи відображають взаємодію сторін у межах навчального процесу. Кожна дія користувача супроводжується відповідною реакцією системи, що забезпечує безперервний та логічно впорядкований процес навчання. Таким чином, діаграма прецедентів наочно демонструє функціональні можливості веб-застосунку та визначає основні сценарії його використання.

Для зображення станів, які може приймати система, й причини переходу з одного стану в інший було розроблено діаграма станів. Діаграма станів відображає послідовність зміни станів інформаційної системи веб-застосунку під час взаємодії користувача з навчальною платформою. Діаграма призначена для опису станів об'єкта й умов переходу між ними. Опис станів дозволяє точно описати модель поведінки об'єкта при одержанні різних повідомлень і взаємодії з іншими об'єктами.

На діаграмі станів, наведений на рисунку А.2, початковим станом системи є активний головний екран, з якого користувач переходить до етапу авторизації. У випадку неуспішної авторизації користувач переходить до стану введення даних для реєстрації, після повторно здійснюється процедура авторизації. У разі активної авторизації користувач отримує доступ до пробного тестування, яке передбачає очікування вибору відповідей, перевірку відповідей та відображення результатів. Після цього система переводить користувача до особистого кабінету.

Подальший навчальний процес включає стан вивчення теоретичного матеріалу., після якого користувач переходить до проходження тестів. Якщо тести не пройдено, система повертає користувача до повторного опрацювання теоретичного матеріалу. У разі успішного проходження тестів користувач переходить до виконання практичних завдань.

Після завершення практичної частини система переводить користувача до підсумкового тестування, яке передбачає очікування вибору відповідей, перевірку відповідей та відображення результату. Після завершення тестування система переходить до кінцевого стану – отримання сертифікату.

Таким чином, діаграма станів наочно демонструє логіку роботи веб-застосунку, можливі варіанти розвитку подій та переходи між станами систем, що залежить від дій користувача і результатів тестування.

Була розроблена діаграма активності, наведена на рисунку А.3, що візуалізує процес використання системи та відображає послідовність виконання дій користувача і системи.

Процес починається з відкриття меню навчального матеріалу, після чого користувач переходить до етапу ознайомлення навчальних модулів. На цьому етапі користувач має можливість обрати або теоретичний або практичний матеріал. Після ознайомлення з навчальними модулями користувач може не продовжувати навчання, система перейде до завершального стану без виконання додаткових дій. Якщо користувач продовжує навчання, то він переходить до перегляду лекцій, що супроводжується проходженням тестування та виконанням практичних завдань.

Завершальними етапами навчального процесу є загальне тестування та збереження прогресу користувача. Після фіксації результатів процес навчання переходить у кінцевий стан.

Отже, діаграма активності наочно відображає логіку взаємодії користувача з веб-застосунком, демонструє послідовність навчальних дій та забезпечує розуміння структури і динаміки навчального процесу.

Розроблено діаграму послідовності, що показує часові особливості передачі і прийому повідомлень об'єктами. Дана діаграма демонструє взаємодію користувача з основними компонентами веб-застосунку під час виконання певних дій, а також взаємозв'язок між окремими елементами системи. Діаграма послідовності використовується для моделювання сценаріїв роботи системи та дозволяє наочно представити процес обміну даними між її компонентами.

На діаграмі послідовності, наведеній на рисунку А.4, учасниками виступають користувач, система та база даних. Користувач робить запит на відкриття меню навчального матеріалу. Система в свою ж чергу звертається до бази даних для отримання списку навчальних модулів. Йде обробка запиту та відображення меню з темами.

Користувач вибирає тему, а система звертається до бази даних для завантаження лекційного матеріалу, потім показуються лекції для користувача. Після набуття знань користувач проходить тестування, для цього він робить запит системі, а система звертається до бази даних для завантаження тестових запитань. Користувач проводить тестування та після завершення відправляє відповіді системі для перевірки тесту, база даних проводить обробку і система повідомляє про результат.

Для виконання практичних завдань користувач знову відправляє запит на отримання переліку практичних завдань, система завантажує їх з бази даних та відображає користувачеві. Після виконання завдань результати передаються до системи для перевірки. База даних зберігає результати практики та відображає результати.

Після проходження всього курсу вивчення основ штучного інтелекту, користувачу потрібно пройти підсумкове тестування. Для цього він робить запит системі та проходить тест, відправляє відповіді системі, система відправляє до бази даних для перевірки результатів. Після обробки користувачу приходить повідомлення про оцінку і в базі даних оновлюється прогрес користувача.

Було розроблено діаграму класів, представлену на рисунку А.5, яка відображає структуру системи шляхом моделювання її класів, властивостей, операцій і зв'язків між об'єктами. Центральним елементом діаграми є клас User, який представляє користувача системи. Він містить атрибути: id, name, email, password, а також методи login() і logout(), що забезпечує вхід і вихід користувача з системи.

Користувач взаємодія з навчальним контентом, представленим класами Theory та Practice. Клас Theory описує теоретичні матеріали, має наступні атрибути: theory_id, title, description, content. Методи open() та markCompleted() відповідають за перегляд матеріалу та позначення його як завершеного. Клас Practice, що описує практичні завдання має такі ж атрибути (practice_id, title, description, content) і метод open() для їх відкриття.

Класи Theory та Practice пов'язані з класом Test, який перевіряє за перевірку знань. Клас Test має наступні атрибути: test_id, title, description та методи start() для початку тестування і checkResult() для перевірки результатів.

Клас Final_Test використовується для підсумкового тестування. Його атрибути: final_test_id, title, description. Методи: start() та checkResult().

Для відстеження навчальних досягнень користувача використовується клас Progress і має наступні атрибути: progress_id, percent, completed та метод updateProgress(), який забезпечує оновлення прогресу. Клас Progress пов'язаний з класом User, що дозволяє зберігати індивідуальний прогрес кожного користувача.

Таким чином, діаграма класів відображає логічну структуру системи, показує взаємозв'язок між користувачем, навчальними матеріалами, тестуванням та контролю прогресу, що забезпечує цілісне представлення предметної області.

2.2 Проектування структури бази даних інформаційної системи

Для створення бази даних було розроблено логічний рівень ER діаграми (рис. 2.1). База даних забезпечує зберігання інформації про користувачів, навчальні курси, модулі, лекційний матеріал, тестування, результати виконання практичних завдань, сертифікати та прогрес навчання користувачів. Зв'язки між таблицями реалізовані за допомогою зовнішніх ключів, що забезпечує цілісність даних та відображає логічні залежності між об'єктами системи. У структурі бази даних використано зв'язки типу «один до багатьох» (one-to-many).

Висновки до розділу 2

Було виконано проектування структури бази даних інформаційної системи. Розроблено ER-діаграму, де визначено основні атрибути та зв'язки, необхідні для зберігання інформації про користувачів, виконання практичних завдань, проходження тестувань та прогрес навчання.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИСТЕМИ ВИВЧЕННЯ ОСНОВ ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Реалізація теоретичних матеріалів та системи тестування

Одним із ключових елементів веб-застосунку є теоретичне навчання та перевірка знань. Вона забезпечує послідовне вивчення основ штучного інтелекту, перевірку засвоєння навчального матеріалу та визначення рівня знань на різних етапах проходження курсу.

Для реалізації теоретичної частини було створено набір навчальних лекцій, які охоплюють основні поняття та технології штучного інтелекту. Навчальний матеріал поділено на окремі розділи, що дозволяє користувачу поступово переходити від базових понять до більш складних тем. У веб-застосунку реалізовано лекції з таких тем: поняття штучного інтелекту, основні принципи штучного інтелекту, машинне навчання, лінійна регресія, нейронні мережі (рис. 3.1).

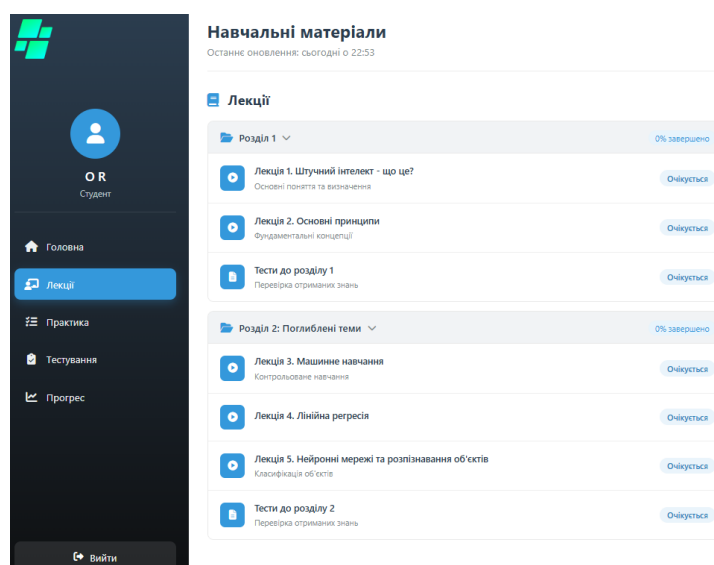


Рисунок 3.1 – Інтерфейс сторінки навчальних матеріалів

Для відображення навчальних матеріалів використовуються веб-сторінки, що формуються засобами HTML та CSS. Дані про лекції зберігаються у базі даних та відображаються після авторизації. Після перегляду лекції система автоматично фіксує її проходження та оновлює інформацію про прогрес навчання.

З метою визначення початкового рівня знань користувача реалізовано вхідне тестування. Після завершення реєстрації користувач проходить тест, що складається із запитань з основ штучного інтелекту (рис. 3.2). На основі отриманих результатів система визначає рівень підготовки користувача та відображає відповідний результат.

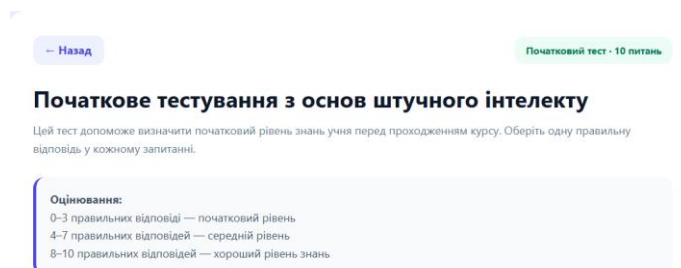


Рисунок 3.2 – Сторінка початкового тестування

Для контролю засвоєння матеріалу після вивчення кожного розділу реалізовано тестування. Кожен тест містить набір запитань із варіантами відповідей, що охоплюють матеріал відповідного розділу. Після завершення тестування система автоматично підраховує кількість правильних відповідей та відображає результат користувачу.

Перевірка правильності відповідей виконується програмно за допомогою JavaScript. Для кожного запитання зберігається правильний варіант відповіді, який порівнюється з вибором користувача (рис. 3.3). Після завершення тесту виконується обчислення загального результату та визначення рівня засвоєння матеріалу.

```
for (let question in correctAnswers) {  
  const selected = document.querySelector(`input[name="${question}"]:checked`);  
  const correctValue = correctAnswers[question];
```

Рисунок 3.3 – Перевірка відповідей

Після проходження всіх лекцій та практичних завдань стає доступним підсумкове тестування. Його метою є комплексна перевірка знань, отриманих під час навчання. Результати підсумкового тесту використовуються для оцінювання рівня підготовки користувача та є однією з умов отримання сертифіката про завершення курсу.

3.2 Програмна реалізація практичних завдань вивчення основ штучного інтелекту

У веб-застосунку для вивчення основ штучного інтелекту реалізовано чотири типи практичних завдань, а саме:

- побудова нейронної мережі та дослідження впливу етапів обробки даних на точність розпізнавання об'єктів;
- навчання класифікатора для розпізнавання об'єктів за допомогою прикладів, наданих користувачем;
- побудова та налаштування моделі лінійної регресії для прогнозування значень на основі набору даних;
- складання інтерактивних пазлів, що демонструють принципи обробки інформації, пошуку закономірностей та прийняття рішень системами штучного інтелекту.

Для реалізації практичних завдань було використано HTML, CSS, JavaScript, TensorFlow.js та попередньо навчену модель MobileNet.

Метою першого практичного завдання є ознайомлення користувача з базовими принципами роботи нейронних мереж та формування розуміння основних етапів обробки даних у системах штучного інтелекту. Практичне завдання реалізовано у вигляді інтерактивного симулятора побудови нейронної мережі. Користувачу пропонується самостійно створити модель штучного інтелекту для розпізнавання об'єктів, використовуючи набір функціональних блоків, що відповідають окремим етапам обробки даних (рис.3.4).

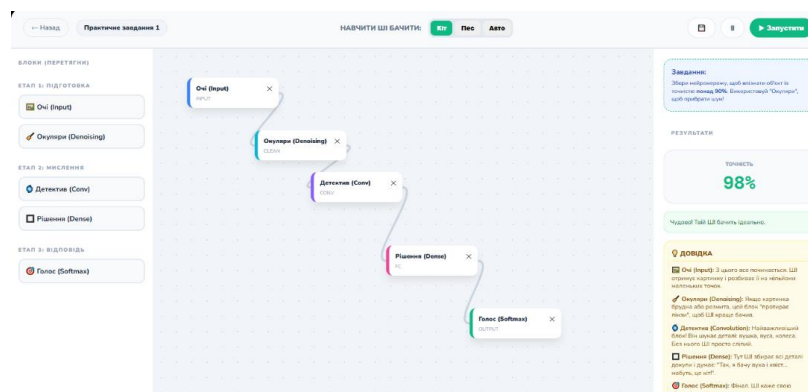


Рисунок 3.4 – Практичне завдання 1

У данному практичному завданні мова програмування JavaScript використовується для реалізації інтерактивної логіки веб-застосунку. Зокрема, вона забезпечує вибір об'єкта для розпізнавання, перетягування функціональних блоків на робочу область, побудову схеми нейронної мережі, перевірку правильності її структури, обчислення точності моделі та збереження результатів роботи користувача.

Вибір об'єкта для розпізнавання реалізовано за допомогою змінної **activeTarget**, яка зберігає поточний тип об'єкта, обраного користувачем для навчання моделі. Для кожного об'єкта визначено назву та рівень складності. Наприклад для *cat* достатньо меншої кількості блоків, а для *car* потрібно більше блоків обробки, оскільки об'єкт має складнішу форму (рис. 3.5).

```
let activeTarget = 'cat';
const canvas = document.getElementById('canvasInner');
const svg = document.getElementById('svg-lines');

const targets = {
  cat: { name: 'Котика', complexity: 1, img: 'https://images.unsplash.com/photo-1514888286974-6c03e2ca1dba?q=80&w=400' },
  dog: { name: 'Песика', complexity: 2, img: 'https://konkurent.ua/media/uploads/prev/2021/08/12/17/50/13/2d211e7b027e21dcba5e27e1aa44dd44.png' },
  car: { name: 'Автомобіль', complexity: 3, img: 'https://images.unsplash.com/photo-1494976388531-d1058494cdd8?q=80&w=400' }
};
```

Рисунок 3.5 - Фрагмент коду вибору об'єкта для розпізнавання

Для забезпечення можливості перетягування елементів нейронної мережі на робочу область використовується механізм Drag-and-Drop, реалізований за допомогою обробників подій **ondragstart**, **ondragover** та **ondrop**. Такий підхід дозволяє користувачу самостійно формувати структуру моделі, розміщуючи блоки у необхідній послідовності (рис. 3.6).

```
document.querySelectorAll('.block-item').forEach(item => {
  item.ondragstart = (e) => e.dataTransfer.setData('node-data', item.dataset.info);
});

canvas.ondragover = e => e.preventDefault();
canvas.ondrop = e => {
  e.preventDefault();
  const data = JSON.parse(e.dataTransfer.getData('node-data'));
  const rect = canvas.getBoundingClientRect();
  createNode(data, e.clientX - rect.left - 90, e.clientY - rect.top - 30);
};
```

Рисунок 3.6 - Фрагмент коду перетягування блоків на робочу область

Функція **createNode()** створює новий елемент моделі на робочому полі. Кожен блок має власний тип, назву та кнопку видалення. Завдяки цьому користувач може не лише додавати елементи, а й змінювати побудовану структуру шляхом видалення або переміщення блоків (рис. 3.7).

```
function createNode(data, x, y) {
  const node = document.createElement('div');
  node.className = 'node';
  node.style.left = x + 'px';
  node.style.top = y + 'px';
  node.style.borderLeftColor = data.color;
  node.dataset.type = data.type;

  node.innerHTML = `
    <div class="node-header">
      <span>${data.name}</span>
      <button class="del-btn" style="border:none; background:none; cursor:pointer;">X</button>
    </div>
    <div class="node-type">${data.type}</div>
  `;

  canvas.appendChild(node);
  node.querySelector('.del-btn').onclick = () => { node.remove(); drawLines(); };
  makeDraggable(node);
  drawLines();
}
```

Рисунок 3.7 - Фрагмент коду створення блоку нейронної мережі

Після побудови моделі виконується перевірка логічної послідовності блоків. Система аналізує, чи починається модель з вхідного шару, чи завершується вихідним шаром, а також чи достатньо блоків для розпізнавання обраного об'єкта (рис. 3.8).

```
if (seq[0] !== 'input') errors.push("Почни з 'Очей' (Input).");
if (seq[seq.length - 1] !== 'output') errors.push("Закінчи 'Голосом' (Softmax).");

const hasCleaner = seq.includes('clean');
const convs = seq.filter(t => t === 'conv').length;

if (convs < targets[activeTarget].complexity) errors.push(`Треба більше 'Детективів' для цього об'єкта!`);
```

Рисунок 3.8 - Фрагмент коду перевірки правильності побудови моделі

Після успішної перевірки структури нейронної мережі система виконує розрахунок показника точності моделі. Обчислення точності моделі залежить від правильно побудованої архітектури. Якщо користувач допустив помилки, система показує низький результат точності (15–25%). Якщо структура моделі правильна, точність зростає. Наявність блоку «Окуляри (Denoising)», який імітує процес очищення даних від шуму дозволяє отримати вищий результат (рис. 3.9).

```
if (errors.length > 0) {
  acc = 15 + Math.random() * 10;
  log.className = 'log-box';
  log.innerHTML = "<b>Помилка:</b><br>• " + errors.join('<br>• ');
} else {
  acc = hasCleaner ? (95 + Math.random() * 4) : (65 + Math.random() * 15);
}
```

Рисунок 3.9 - Фрагмент коду обчислення точності моделі

Отже, програмна реалізація першого практичного завдання поєднує візуальну взаємодію користувача з навчальною логікою. Користувач будує модель, аналізує допущені помилки та спостерігає вплив структури мережі на результат розпізнавання об'єктів. Під час виконання завдання користувач знайомиться з

поняттями вхідного шару, обробки даних, пошуку ознак та класифікації результатів. Однією з типових помилок є неправильна послідовність розташування блоків або недостатня кількість шарів для обробки даних. У такому випадку система повідомляє про помилки та демонструє зниження точності моделі. Правильним рішенням є побудова логічної структури нейронної мережі, що включає вхідний шар, етапи обробки даних та вихідний шар класифікації. Завдяки цьому користувач розуміє, що якість роботи штучного інтелекту залежить від правильної архітектури моделі.

У другому практичному завданні користувач навчає модель класифікувати об'єкти за допомогою прикладів зроблених веб-камерою. Для цього користувач самостійно додає приклади трьох класів: «Людина», «Кіт», «Собака», після чого система в режимі реального часу виконує розпізнавання об'єкта з веб-камери (рис. 10).

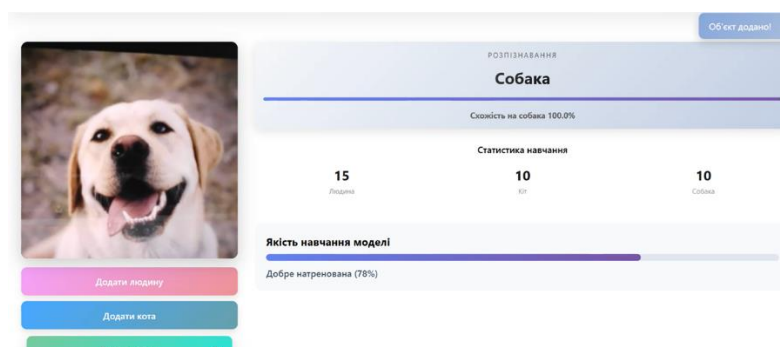


Рисунок 3.10 – Практичне завдання 2

У коді використовується TensorFlow.js, яка дозволяє запускати моделі машинного навчання у браузері. Використовується попередньо навчена модель MobileNet, яка виконує виділення ознак із зображення, та класифікатор KNN-класифікатор, який порівнює нове зображення з прикладами наданими користувачем.

Змінна **classifier** містить KNN-класифікатор за допомогою якого відбувається навчання моделі на прикладах користувача. Елемент **webcamElement** відповідає за отримання відеопотоку з веб-камери. Змінна **net** використовується для збереження моделі MobileNet з обмеженою кількістю в 15 прикладів (рис. 3.11).

```
const classifier = knnClassifier.create()
const webcamElement = document.getElementById("webcam")

let net
const MAX_EXAMPLES = 15
```

Рисунок 3.11 – Фрагмент коду створення класифікатора

Функція **app()** завантажує модель MobileNet та запускає веб-камеру. Якщо в базі даних є раніше збережена модель користувача, то модель відновлюється за допомогою функції **loadModel()** (рис. 3.12).

Функція **addExample()** додає нові приклади до моделі після натискання однієї з кнопок додавання прикладу. Спочатку перевіряється кількість уже доданих прикладів для певного класу. Якщо прикладів ще недостатньо, то здійснюється захоплення зображення з веб-камери і модель MobileNet виконує виділення характерних ознак зображення. Отриманні ознаки передаються до KNN-класифікатора для подальшого навчання моделі (рис. 3.12).

Цикл **while** забезпечує безперервне оновлення результату розпізнавання та передає KNN-класифікатору для визначення найбільш схожого класу.

```
async function app() {
  console.log("Loading mobilnet...")

  net = await mobilenet.load()
  await loadModel()

  console.log("Loaded model")

  const webcam = await tf.data.webcam(webcamElement)

  const addExample = async (classId, className) => {
    const dataset = classifier.getClassifierDataset()

    let count = dataset[classId] ? dataset[classId].shape[0] : 0

    if (count >= MAX_EXAMPLES) {
      showToast("Вже достатньо прикладів!", "warning")
      return
    }

    const img = await webcam.capture()

    const activation = net.infer(img, true)

    classifier.addExample(activation, classId)

    img.dispose()

    updateStatistics()
  }
}
```

Рисунок 3.12 – функції app() та addExample()

Для зменшення кількості помилкових відповідей реалізовано перевірку впевненості моделі. Визначається клас із найбільшою ймовірністю, а також різниця між найкращим та другим результатом. Якщо впевненість моделі менша за 75% або різниця між двома найближчими класами менша за 15%, система виводить повідомлення «Невідомий об'єкт». Це дозволяє уникнути ситуацій помилкового

розпізнавання об'єкта. Результат класифікації відображається в блоку console (рис. 3.13).

```
const bestLabel = result.label
const bestConfidence = confidences[bestLabel]

const values = Object.values(confidences).sort((a, b) => b - a)
const secondBest = values[1] || 0

const diff = bestConfidence - secondBest

let prediction = classes[bestLabel]
let isUnknown = false

if (bestConfidence < 0.75 || diff < 0.15) {
  prediction = "Невідомий об'єкт"
  isUnknown = true
}
```

Рисунок 3.13 – Фрагмент коду перевірки впевненості моделі

Функція **updateModelQuality()** оцінює якість навчання моделі. Якість моделі обчислюється на основі кількості доданих прикладів у кожному класі. Якщо користувач додав достатню кількість прикладів для всіх категорій, рівень навчання моделі зростає. Якщо хоча б один клас не має прикладів, система повідомляє, що потрібно додати дані для всіх класів (рис. 3.14).

```
function updateModelQuality(humanCount, catCount, dogCount) {
  const MAX = 15

  const humanScore = Math.min(humanCount / MAX, 1)
  const catScore = Math.min(catCount / MAX, 1)
  const dogScore = Math.min(dogCount / MAX, 1)

  const score = Math.round(((humanScore + catScore + dogScore) / 3) * 100)
```

Рисунок 3.14 – Функція updateModelQuality()

Отже, у другому практичному завданні реалізовано повний цикл навчання моделі розпізнавання об'єктів: отримання зображення з веб-камери, додавання навчальних прикладів, виділення ознак за допомогою MobileNet, класифікація за допомогою KNN, оцінювання якості навчання, розпізнавання об'єкта в реальному часі. Користувач може припуститися помилок, додаючи недостатню кількість прикладів або використовуючи неякісні зображення, що призводить до зниження точності розпізнавання. Виконання завдання дозволяє зрозуміти залежність якості роботи моделі від обсягу та різноманітності навчальних даних, а також демонструє процес навчання класифікатора на основі прикладів користувача.

У третьому практичному завданні користувач ознайомлюється з принципами роботи алгоритму лінійної регресії та формування розуміння процесу побудови

прогнозних моделей на основі даних. Користувач самостійно налаштовує параметри регресійної моделі та спостерігає вплив цих змін на якість прогнозування (рис. 3.15).



Рисунок 3.15 – Практичне завдання 3

Візуалізація даних здійснюється на елементі **Canvas**, де відображаються точки навчальної вибірки, лінія регресії та відстані між фактичними значеннями і прогнозами моделі.

Для керування моделлю використовуються два повзунки: нахил прямої та зміщення. Значення цих параметрів зчитуються з відповідних елементів інтерфейсу (рис. 3.16). Параметр *slope* визначає кут нахилу прямої регресії, а *intercept* — точку її перетину з віссю ординат. Змінюючи ці параметри, користувач фактично навчає модель прогнозувати значення.

```
const slopeInput = document.getElementById("slope");
const interceptInput = document.getElementById("intercept");
```

Рисунок 3.16 - Фрагмент коду отримання значень коефіцієнтів лінійної регресії

Для обчислення прогнозованого значення використовується функція **predictY** (рис. 3.17). Дана функція реалізує математичну модель лінійної регресії у вигляді рівняння прямої $y = ax + b$. Коефіцієнт a визначає нахил прямої регресії, а коефіцієнт b задає її зміщення відносно початку координат. На основі цих параметрів система обчислює прогнозоване значення y для кожного значення x .

```
function predictY(x, slope, intercept) {
    return slope * x + intercept;
}
```

Рисунок 3.17 – Функція **predictY**

Для оцінювання якості побудованої моделі використовується функція **calculateError** для розрахунку сумарної помилки. Функція проходить по всіх точках набору даних, обчислює прогнозоване значення та знаходить абсолютну різницю між реальним і прогнозованим результатом. Отримані відхилення підсумовуються. Чим менше значення сумарної помилки, тим точніше працює модель.

Для автоматичного визначення оптимальної лінії обчислюються оптимальні значення коефіцієнта нахилу та зміщення. Саме ця лінія забезпечує мінімальну сумарну помилку для поточного набору даних.

Після побудови графіка система виконує оцінювання якості моделі:

```
let quality = Math.round((bestError / error) * 95);
```

Якість моделі визначається шляхом порівняння поточної помилки користувача з мінімально можливою помилкою, отриманою для оптимальної лінії регресії. Максимальне значення якості обмежене на рівні 95%.

Таким чином користувач отримує зворотний зв'язок щодо якості побудованої моделі та може поступово покращувати результат шляхом зміни параметрів прямої.

У практичному завданні реалізовано декілька способів введення даних:

- використання готового прикладу;
- ручне введення точок;
- завантаження CSV-файлу.

Завантаження CSV-файлу реалізовано за допомогою об'єкта **FileReader**:

```
const reader = new FileReader();
reader.readAsText(file, "UTF-8");
```

Після зчитування файлу дані автоматично перетворюються на набір точок та відображаються на графіку.

Отже, третє практичне завдання дозволяє користувачу на практиці зрозуміти принцип роботи лінійної регресії, вплив параметрів моделі на якість прогнозування. Найчастіше помилки виникають у випадках неправильного вибору коефіцієнта нахилу або зміщення прямої, що призводить до збільшення сумарної

помилки моделі. Правильним рішенням є підбір такої лінії, яка проходить максимально близько до більшості точок набору даних та забезпечує мінімальне значення помилки. У результаті користувач засвоює поняття прогнозування, залежності між змінними та впливу параметрів моделі на якість отриманих результатів.

Четверте практичне завдання демонструє принципи комп'ютерного зору та класифікації зображень у системах штучного інтелекту. Завдання реалізовано у формі інтерактивного пазла, під час складання якого користувач взаємодіє із зображенням, а після завершення система автоматично виконує його аналіз та класифікацію (рис. 3.18).

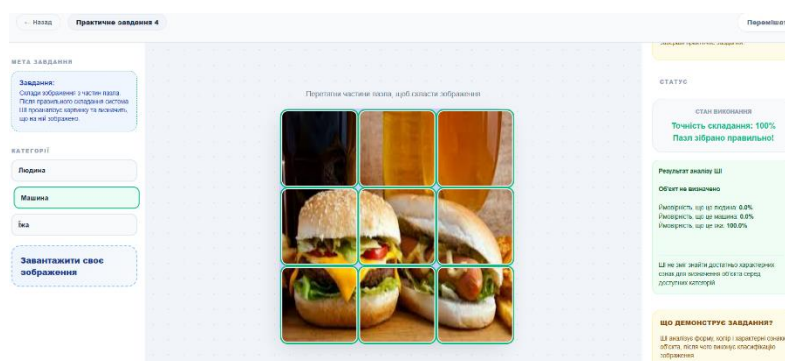


Рисунок 3.18 – Практичне завдання 4

Для аналізу зображень використовується бібліотека TensorFlow.js та попередньо навчена модель MobileNet, яка дозволяє визначати об'єкти на зображенні.

Функція **loadModel()** завантажує попередньо навчену модель MobileNet. Для виконання практичного завдання користувач працює з трьома категоріями об'єктів, а саме «Людина», «Автомобіль», «Їжа». Кожна категорія містить набір ключових слів та характерних ознак, які використовуються для аналізу та класифікації зображень.

Після вибору категорії формується пазл із дев'яти фрагментів. Для цього використовується функція **createPieces()**. У межах функції створюються дев'ять елементів пазла розміром 3×3. Для кожного фрагмента зберігається його правильна

позиція, що дозволяє надалі перевіряти коректність складання зображення (рис.3.19).

```
function createPieces(imageUrl) {
  board.innerHTML = "";

  for (let i = 0; i < 9; i++) {
    const piece = document.createElement("div");

    piece.className = "piece";
    piece.draggable = true;

    const row = Math.floor(i / 3);
    const col = i % 3;

    piece.dataset.correct = i;

    piece.style.backgroundImage = `url("${imageUrl}")`;
    piece.style.backgroundPosition = `-${col * 150}px -${row * 150}px`;
    piece.style.backgroundSize = "450px 450px";

    piece.addEventListener("dragstart", dragStart);
    piece.addEventListener("dragover", dragOver);
    piece.addEventListener("drop", dropPiece);
    piece.addEventListener("dragend", dragEnd);

    board.appendChild(piece);
  }
}
```

Рисунок 3.19 – Функція **createPieces()**

Функція **shufflePieces()** змінює порядок розташування частин пазла, що створює початковий стан завдання та потребує від користувача самостійного відновлення вихідного зображення (рис. 3.20).

```
function shufflePieces() {
  if (!uploadedImage) {
    statusText.textContent = "Спочатку оберіть категорію або завантажте зображення";
    return;
  }

  const pieces = Array.from(board.children);

  const positions = pieces.map(piece => ({
    bg: piece.style.backgroundPosition,
    correct: piece.dataset.correct
  }));

  positions.sort(() => Math.random() - 0.5);

  pieces.forEach((piece, index) => {
    piece.style.backgroundPosition = positions[index].bg;
    piece.dataset.correct = positions[index].correct;
    piece.classList.remove("correct-piece", "wrong-piece");
  });

  resultBox.classList.remove("show");
  resultBox.innerHTML = "";

  statusText.textContent = "Пазл перемішано. Складіть зображення правильно";
}
```

Рисунок 3.20 - Функція **shufflePieces()**

Переміщення елементів реалізується за допомогою механізму Drag-and-Drop. Під час перетягування користувач може міняти місцями окремі фрагменти

зображення. Після кожного переміщення викликається функція перевірки правильності складання пазла.

Оцінювання результату виконується функцією **checkPuzzle()**:

```
const accuracy = Math.round((correctCount / pieces.length) * 100);
```

Система порівнює поточне розташування кожного фрагмента з його правильною позицією та обчислює відсоток правильності складання. Результат відображається користувачу у вигляді показника точності.

Після успішного складання пазла запускається процес аналізу зображення. Таким чином користувач отримує доступ до наступного етапу лише після повного складання зображення. Аналіз зображення виконується за допомогою моделі MobileNet. Модель отримує вихідне зображення та повертає перелік найбільш ймовірних об'єктів разом із рівнем впевненості для кожного результату.

Для визначення належності до конкретної категорії використовується перевірка ключових слів. Якщо назва знайденого об'єкта містить одне з ключових слів відповідної категорії, до її рейтингу додається значення ймовірності. Аналогічна перевірка виконується для категорій «Людина» та «Їжа».

Після завершення аналізу система відображає результат класифікації. Користувач отримує інформацію про визначену категорію, відсоток впевненості моделі та перелік ознак, за якими система штучного інтелекту виконала класифікацію.

Отже, у четвертому практичному завданні користувач має можливість завантажити власне зображення відповідно до обраної категорії («Автомобіль» або «Їжа»). Система аналізує завантажене зображення, визначає ступінь його відповідності вибраній категорії та відображає ознаки, за якими було прийнято рішення про класифікацію об'єкта. Помилкою вважається завантаження зображення, яке не належить до обраної категорії або не містить достатньо характерних ознак для його розпізнавання. У такому випадку система повідомляє про відсутність достатньої схожості із запропонованими категоріями. У результаті виконання завдання користувач отримує практичне уявлення про те, як системи штучного інтелекту аналізують візуальну інформацію, виділяють характерні

ознаки об'єктів, пояснюють результати класифікації та приймають рішення щодо належності зображення до певної категорії.

Для забезпечення можливості відстеження результатів навчання у веб-застосунку реалізовано систему контролю прогресу користувача. Її основним призначенням є фіксація виконаних навчальних елементів, розрахунок загального рівня проходження курсу та відображення рівня знань користувача.

Система контролю прогресу охоплює три основні складові навчального процесу:

- проходження теоретичних матеріалів;
- виконання практичних завдань;
- проходження тестування.

Після завершення кожної лекції інформація про її проходження записується до таблиці *user_lesson_progress*. Для кожного користувача зберігається статус проходження відповідного навчального матеріалу. На основі цих даних система визначає кількість завершених лекцій та обчислює відсоток їх виконання.

Контроль виконання практичних завдань реалізовано за допомогою таблиці *user_practice_progress*. Після успішного завершення практичної роботи відповідний запис додається до бази даних, а статус завдання змінюється на «Завершено». Це дозволяє контролювати проходження всіх практичних завдань курсу

Для збереження результатів тестування використовуються таблиця *user_test_attempts*, що містить інформацію про кількість набраних балів, дату проходження тесту та рівень підготовки користувача. Отримані результати використовуються для оцінювання рівня засвоєння навчального матеріалу.

Загальний прогрес навчання розраховується на основі кількості виконаних елементів курсу, а саме завершені лекції, практичні завдання та тестування. Результат відображається у відсотках на сторінці прогресу користувача (рис. 3.21).

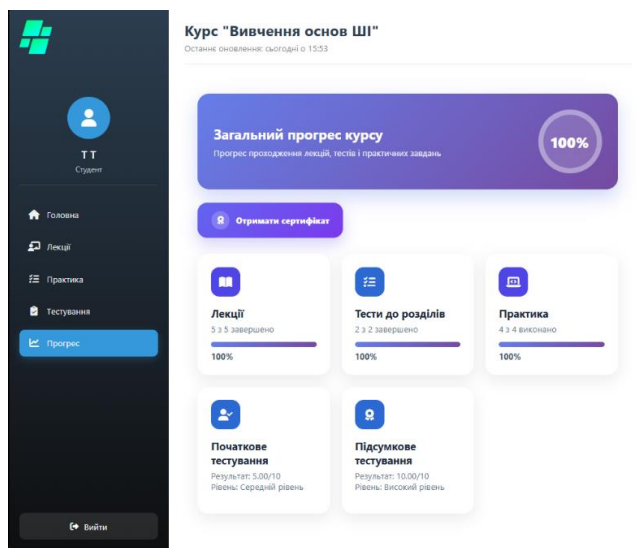


Рисунок 3.21 – Сторінка прогресу користувача

Після успішного завершення всіх навчальних елементів та досягнення 100% прогресу користувачу стає доступною можливість формування сертифіката про завершення курсу. Сертифікат автоматично містить прізвище та ім'я користувача, дату формування документа та назву курсу (рис. 3.22).

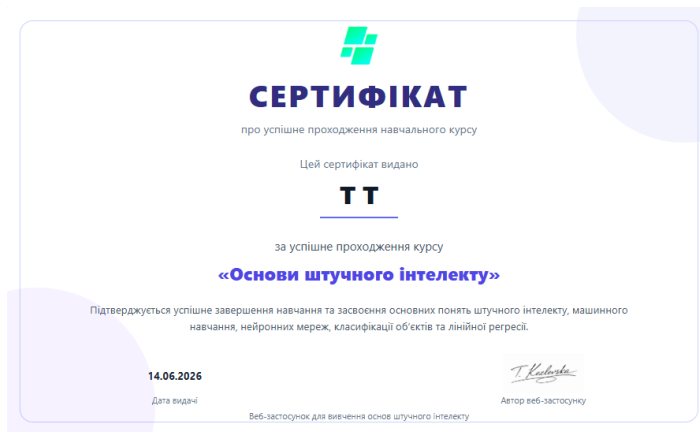


Рисунок 3.22 – Сертифікат

Реалізована система контролю прогресу дозволяє відстежувати результати навчання, автоматично визначати відсоток проходження курсу та надавати сертифіката після успішного завершення всіх етапів навчання.

3.3 Інструкція користувачу інформаційної системи

Веб-застосунок для вивчення основ штучного інтелекту призначений для вивчення теоретичного матеріалу, виконання практичних завдань та тестування,

відстеження прогресу у єдиному інформаційному середовищі. Інструкція користувачу допоможе ефективно використовувати всі можливості системи.

Для початку роботи потрібно авторизуватися в системі, для незареєстрованих користувачів необхідно зареєструватися. Після успішної реєстрації потрібно пройти початкове тестування (рис.3.2). Після завершення тестування здійснюється автоматичний перехід до особистого кабінету.

Бічна панель (ліва частина) містить основні розділи системи:

1. Лекції – перегляд навчальних матеріалів;
2. Практика – виконання практичних завдань;
3. Тестування – проходження підсумкового тестування для перевірки знань;
4. Прогрес – перегляд статистики навчання.

Кнопка «Вийти» використовується для завершення роботи в системі та виходу з облікового запису.

Основна область (права частина) відображає контент вибраного розділу.

Висновки до розділу 3

У третьому розділі було реалізовано систему теоретичного навчання, практичні завдання, тестування, систему контролю прогресу та розроблено керівництво користувачу для роботи із інформаційною системою.

Практичні завдання реалізовані з використанням сучасних веб-технологій та спрямовані на вивчення з основними поняттями штучного інтелекту, принципами роботи нейронних мереж, класифікацією об'єктів, лінійною регресією та розпізнаванням об'єктів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено веб-застосунок для вивчення основ штучного інтелекту, орієнтований на користувачів віком від 12 років. Основною метою розробки було створення єдиного навчального середовища для підвищення ефективності навчання основам штучного інтелекту, яке поєднує теоретичний матеріал, практичні завдання, тестування та контроль результатів навчання.

Було проведено аналіз предметної області та існуючих платформ для вивчення штучного інтелекту. Визначено їх переваги та недоліки, а також обґрунтовано необхідність створення веб-застосунку. Було розроблено структуру веб-застосунку, UML-діаграми, модель бази даних та архітектуру програмного рішення. Спроектowana база даних забезпечує зберігання інформації про користувачів, навчальні матеріали, результати тестування, виконання практичних завдань та прогрес навчання.

Реалізовано функціональні можливості веб-застосунку. Розроблено систему навчальних лекцій, початкове та підсумкове тестування, механізм контролю прогресу користувача та формування сертифіката. Для закріплення знань реалізовано чотири інтерактивні практичні завдання, які дозволяють користувачам ознайомитися з принципами роботи нейронних мереж, класифікації об'єктів, лінійної регресії та комп'ютерного зору. Також створено зручний користувацький інтерфейс, який забезпечує просту навігацію між усіма розділами системи.

Для реалізації веб-застосунку використано мови програмування PHP, JavaScript, HTML та CSS, а також систему керування базами даних MySQL. Для реалізації окремих функцій штучного інтелекту використано бібліотеки TensorFlow.js, MobileNet та KNN Classifier. У результаті виконання роботи досягнуто поставленої мети та виконано всі визначені завдання. Розроблений веб-застосунок забезпечує комплексний підхід до вивчення основ штучного інтелекту. Веб-застосунок може бути використаний як додатковий навчальний інструмент для учнів, студентів і всіх користувачів, які бажають отримати базові знання у сфері

штучного інтелекту. Надалі проєкт планується розвивати шляхом наповнення навчальної програми, оновлення практичних занять та впровадження інтерактиву для глибшого вивчення ШІ. Перспективним кроком стане перехід до розробки та інтеграції значно складніших задач машинного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штучний інтелект (ШІ) – що це таке, як працює і навіщо потрібен. Termin.in.ua : веб-сайт. URL: <https://termin.in.ua/shtuchnyy-intelekt/> (дата звернення 29.10.2025).
2. Пчелянський Д. П., Воїнова С. А. Штучний інтелект: перспективи та тенденції розвитку. Автоматизація технологічних і бізнес-процесів. 2019. Т. 11, № 3. С. 59–64 (дата звернення: 29.10.2025).
3. Мар'єнко М. В., Коваленко В. В. Штучний інтелект та відкрита наука в освіті. Фізико-математична освіта. 2023. Т. 38, № 1. С. 48–53 (дата звернення: 13.11.2025).
4. Наріжна В. Україна має лише 9,1 % впровадження ШІ, але й великий потенціал — Microsoft. ZN.UA : веб-сайт. URL: <https://zn.ua/ukr/TECHNOLOGIES/ukrajina-maje-lishe-9-1-vprovadzhennja-shi-ale-j-velikij-potentsial-microsoft.html> (дата звернення: 13.11.2025).
5. Майже половина українців користується штучним інтелектом. Укрінформ : веб-сайт. URL: <https://www.ukrinform.ua/rubric-society/4101218-majze-polovina-ukrainciv-koristuetsa-stucnim-intelektom.html> (дата звернення: 13.11.2025).
6. Elements of AI. URL: <https://www.elementsofai.com> (дата звернення: 15.11.2025).
7. Hour of Code. URL: <https://code.org/en-US/hour-of-code> (дата звернення: 15.11.2025).
8. Machine Learning | Google for Developers. URL: <https://developers.google.com/machine-learning/crash-course?hl=uk> (дата звернення: 15.11.2025).
9. Сич Т. А., Заяць А. М. В. Навчальні платформи як освітній інструмент сьогодення. Інтелектуальні комп'ютерні системи та мережі : матеріали VII наук.-практ. конф. молодих вчених і студентів, 23 трав. 2023 р. Тернопіль, 2023. С. 26 (дата звернення: 19.11.2025).

10. Юлія Каграманова. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (дата звернення: 19.11.2025).

11. Діаграма станів. Main. URL: <https://emtd.ztu.edu.ua/view/word-3021/0> (дата звернення: 21.11.2025).

12. Застосування UML (частина 2). Діаграма послідовності - Sequence Diagram. Державний університет інформаційно-комунікаційних технологій. URL: [https://duikt.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy](https://duikt.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy) (дата звернення: 23.11.2025).

13. Джейд Моралес. Що таке діаграма класів UML і найкращий творець діаграм класів UML. URL: <https://www.mindonmap.com/uk/blog/what-is-uml-class-diagram/> (дата звернення: 23.11.2025).

14. Зв'язок у базі даних: основні типи та як встановлюється. FoxmindEd. URL: <https://foxminded.ua/zviazok-u-bazi-danykh/> (дата звернення: 23.11.2025).

15. Звенігородський О.С., Зінченко О.В., Чичкарьов С.А., Кисіль Т.М. Штучний інтелект. Вступний курс: Навчальний посібник. – К.: ДУТ, 2022. – 193 с (дата звернення: 14.01.2026).

16. Методи та системи штучного інтелекту. – Х.: ТОВ «ДІСА ПЛЮС», 2021. -177 с (дата звернення: 19.01.2026).

17. Машинне навчання: Конспект лекцій / Гавриленко С. Ю.,– Харків: НТУ «ХПІ», 2024. – 232 с (дата звернення: 14.01.2026).

18. Сімон К. А., Ніколюк П. К. Нейронні мережі. Комп'ютерні технології обробки даних : матеріали III Всеукраїнської наук.-практ. конф., 8 груд. 2022 р. Вінниця, 2022. С. 170-173 (дата звернення: 19.01.2026).

19. Жеребук О., Фармага І. Використання нейронних мереж для визначення об'єктів на зображенні. Computer Design Systems. Theory and Practice. 2024. Т. 6, № 1. С. 232–240 (дата звернення: 03.02.2026).

20. Гавриш Б. М., Тимченко О. В., Кляп М. П. Алгоритми розпізнавання об'єктів на зображенні. Поліграфія і видавнича справа. 2022. № 1(83). С. 47–58 (дата звернення: 10.03.2026).

21. Мартинюк Т. Б., Круківський Б. І., М'якішев О. А. Особливості моделей нейромережного класифікатора для розпізнавання об'єктів. Вісник Вінницького політехнічного інституту. 2022. № 4. С. 56–63 (дата звернення: 10.03.2026).

22. Сучасний підручник JavaScript. JavaScript.info : веб-сайт. URL: <https://uk.javascript.info/> (дата звернення: 11.03.2026).

23. Що таке алгоритм k-найближчих сусідів (KNN)? Unite.AI : веб-сайт. URL: <https://www.unite.ai/uk/what-is-k-nearest-neighbors/> (дата звернення: 25.03.2026).

24. Калініна І., Гожий О. Дослідження ефективності методів класифікації при прогнозуванні в задачах машинного навчання. Управління розвитком складних систем. 2021. № 46. С. 173–180 (дата звернення: 25.03.2026).

25. TensorFlow.js Documentation. URL: <https://www.tensorflow.org/js> (дата звернення: 26.03.2026).

26. MobileNet Documentation. URL: https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNet (дата звернення: 28.03.2026).

27. WANG, Wei, et al. A novel image classification approach via dense-MobileNet models. Mobile Information Systems, 2020, 2020.1: 7602384 (дата звернення: 29.03.2026).

28. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних. URL: https://web.posibnyky.vntu.edu.ua/fitki/11petuh_bazdanyh_movy_zalitiv/21.htm (дата звернення: 02.04.2026).

29. Донченко, Н. Л.; Константинова, Л. В. Організація баз даних для сайтів. Наукове видання. 514 с (дата звернення: 11.04.2026).

30. Основні переваги субд MySQL. URL:
<https://studfile.net/preview/5607354/page:3/> (дата звернення: 11.04.2026).
31. PHP Підручник. W3Schools українською : веб-сайт. URL:
<https://w3schoolsua.github.io/php/index.html> (дата звернення: 13.04.2026).

ДОДАТКИ

Додаток А



Рисунок А.1 – Діаграма прецедентів

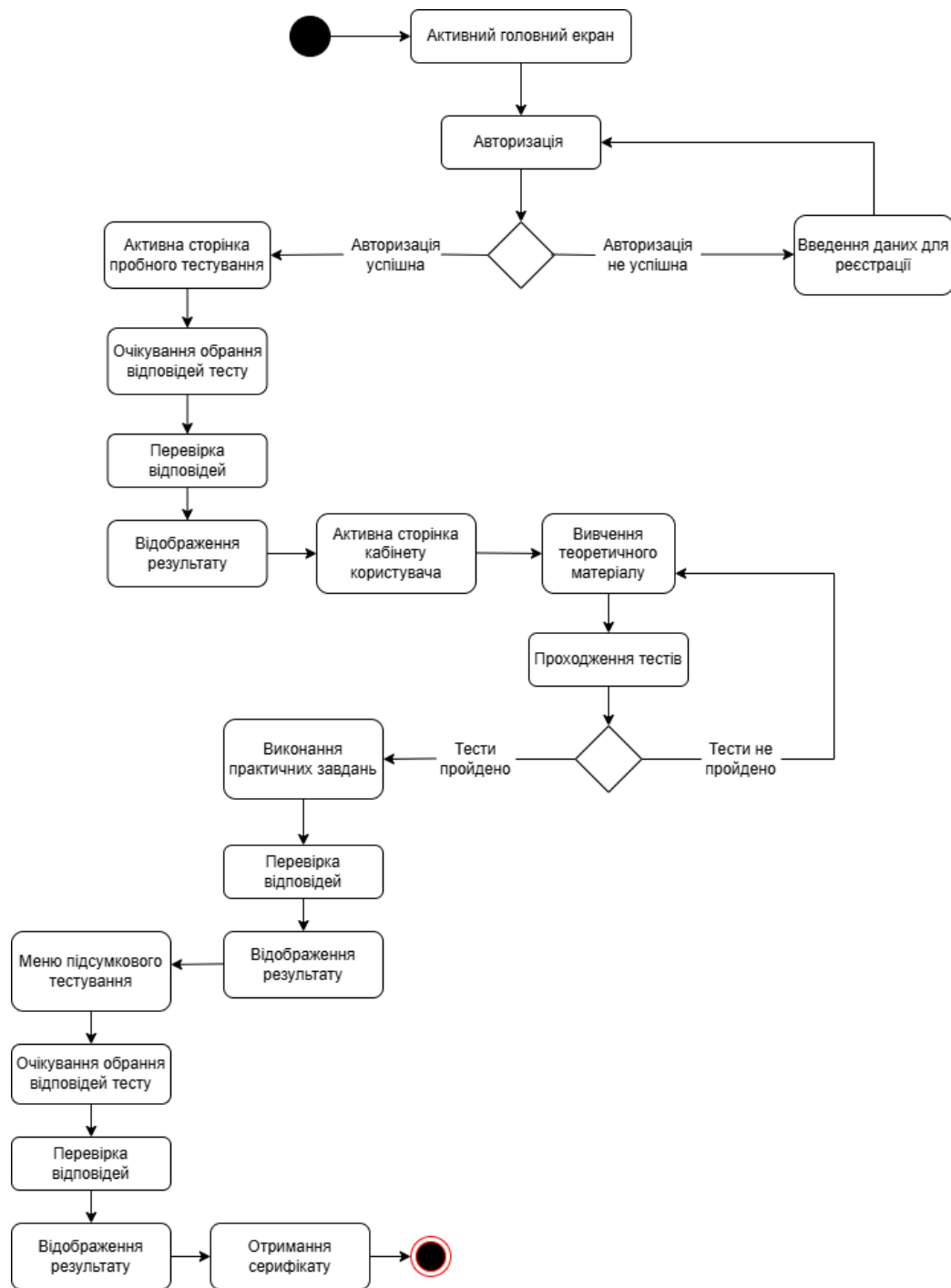


Рисунок А.2 – Діаграма станів

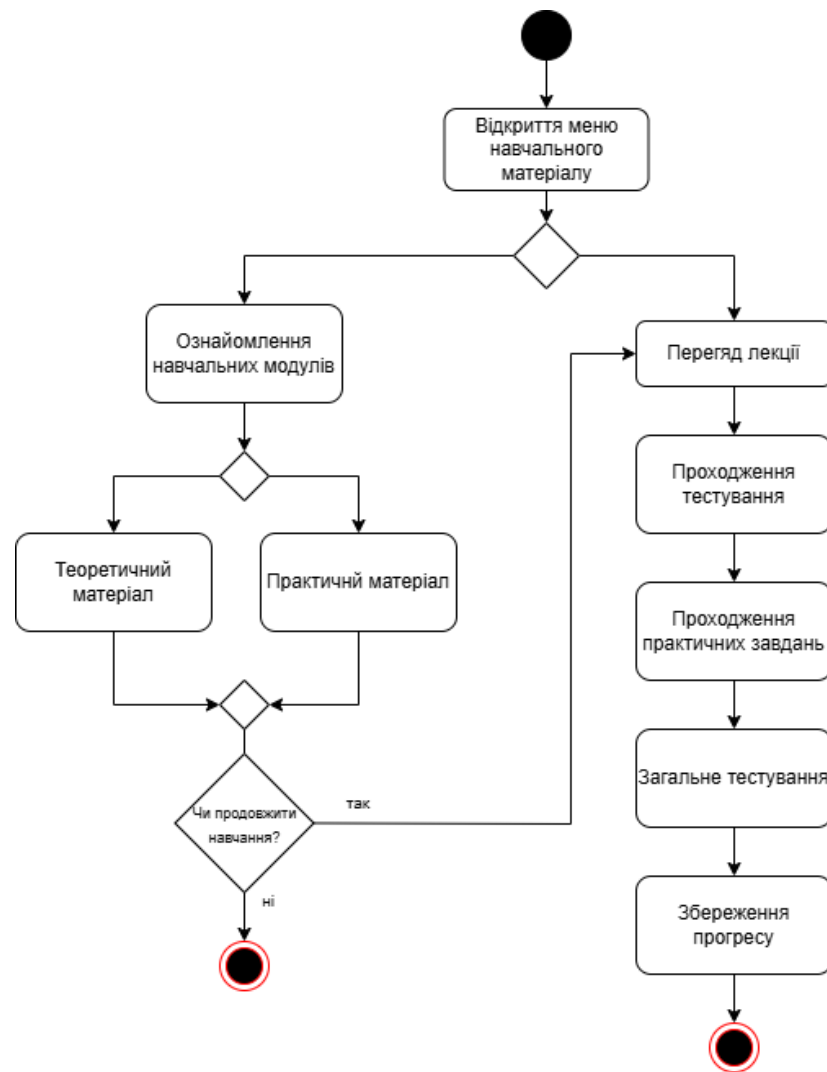


Рисунок А.3 – Діаграма активності

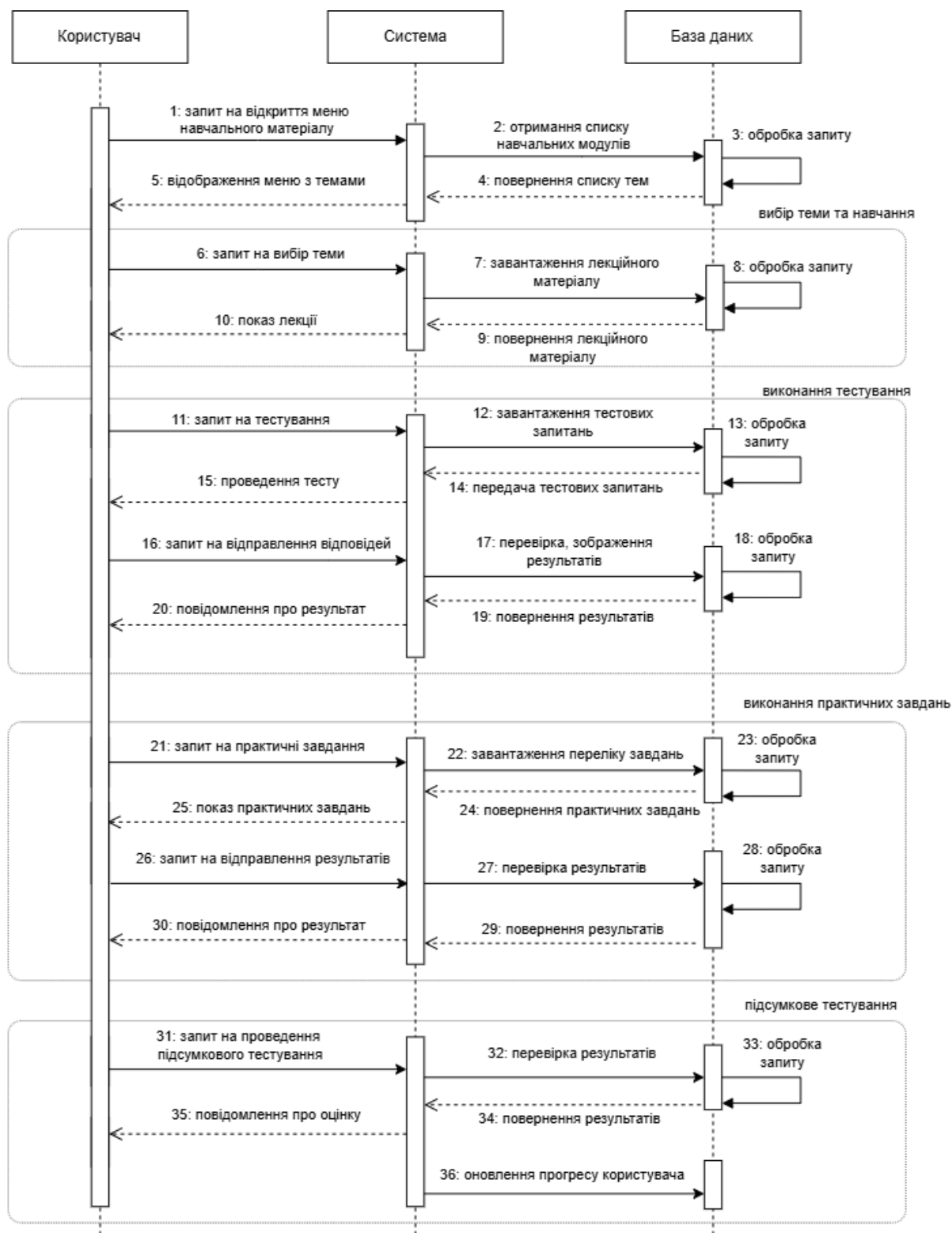


Рисунок А.4 – Діаграма послідовності

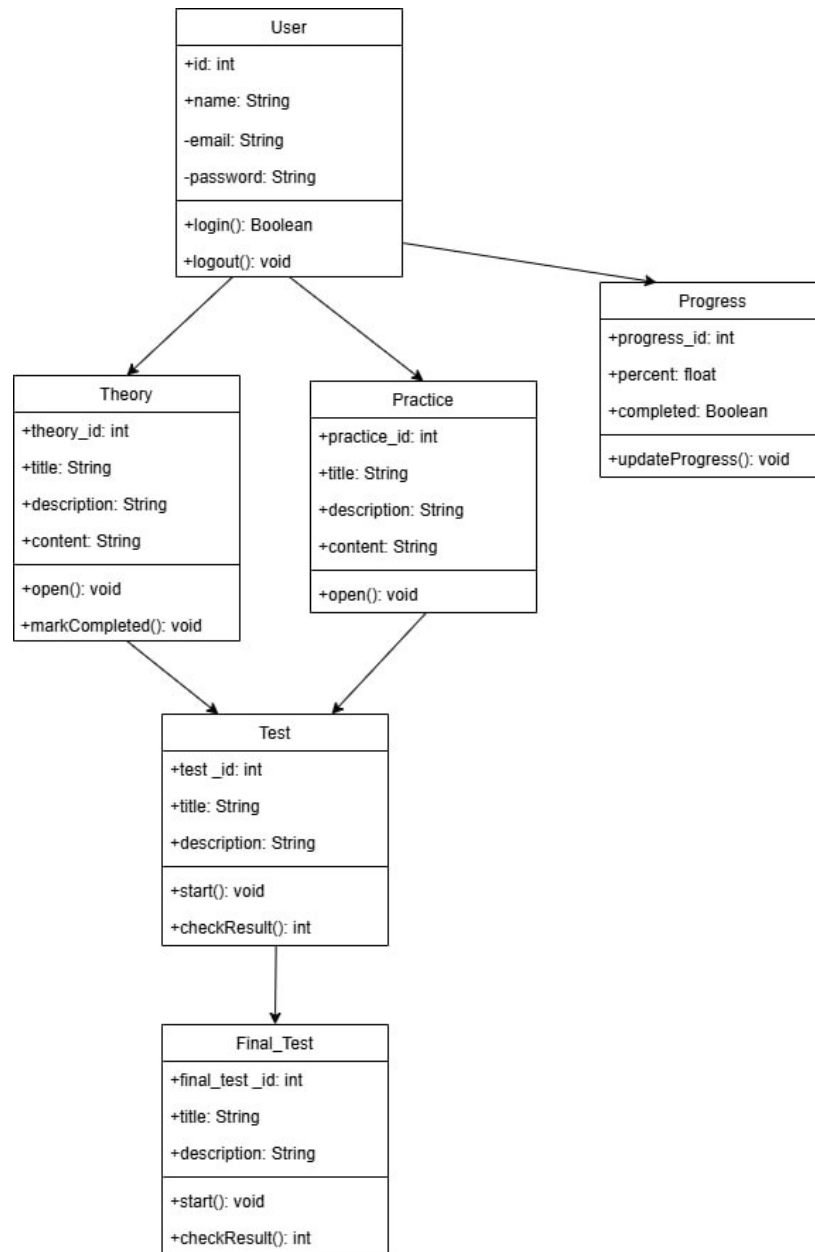


Рисунок А.5 – Діаграма класів