

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Інтеграція штучного інтелекту у веб-додаток для покращення
взаємодії користувачів»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Коряк Владислав Вадимович
Керівник: Поночовний Юрій
Леонідович
Рецензент: Муравльов Володимир
Вячеславович

Полтава – 2025 року

ВСТУП

Актуальність теми. У сучасному цифровому світі штучний інтелект перетворюється на ключову технологію, що знаходить застосування у дедалі ширшому спектрі галузей, включно з розробкою веб-додатків, електронною комерцією, маркетингом, освітою та охороною здоров'я. За прогнозами аналітичних агенцій, глобальний ринок рішень на базі ШІ демонструє експоненційне зростання, з очікуваним обсягом у сотні мільярдів доларів США в найближчі роки та щорічним приростом, що перевищує 20-30%. В Україні також спостерігається активний інтерес до впровадження технологій ШІ, зокрема для підвищення ефективності бізнес-процесів та покращення взаємодії з користувачами у веб-середовищі [1]. Особливо актуальним стає завдання інтеграції інтелектуальних функцій у веб-додатки з метою забезпечення персоналізованого, інтуїтивно зрозумілого та ефективного користувацького досвіду.

Однак, незважаючи на значний потенціал ШІ, багато веб-додатків все ще не використовують його можливості повною мірою для оптимізації взаємодії з користувачем. Існує проблема браку комплексних підходів та моделей інтеграції, які б дозволяли розробникам системно впроваджувати елементи ШІ (такі як інтелектуальні чат-боти, системи персоналізації контенту, рекомендаційні системи, голосовий пошук) для досягнення конкретних покращень у UX. Багато існуючих рішень є або занадто спеціалізованими, або не враховують специфіку конкретних веб-додатків та поведінкових патернів їх користувачів. Це створює нагальну потребу в дослідженні та розробці ефективних методів інтеграції ШІ у веб-додатки, спрямованих на підвищення рівня задоволеності, залученості та продуктивності користувачів.

Стан розробки проблеми. Питання застосування штучного інтелекту у веб-розробці та для покращення користувацького досвіду активно досліджуються в науковій літературі. Публікації в авторитетних базах даних, таких як Scopus та Web of Science, підкреслюють важливість персоналізації,

автоматизації підтримки та інтелектуального аналізу даних для створення адаптивних веб-інтерфейсів. Розглядаються різні підгалузі ШІ, зокрема машинне навчання та обробка природної мови, як інструменти для реалізації таких функцій. Проте, більшість досліджень зосереджені або на окремих технологіях ШІ, або на загальних принципах UX-дизайну. Питання розробки комплексних моделей інтеграції ШІ, які б системно покращували ключові показники взаємодії користувачів з веб-додатком, а також практичні аспекти такої інтеграції, залишаються недостатньо висвітленими. Існуючі підходи часто не пропонують чітких методик оцінки впливу впроваджених ШІ-рішень на загальний користувацький досвід [2].

Метою кваліфікаційної роботи є розробка та обґрунтування підходів до інтеграції штучного інтелекту у веб-додаток для покращення ключових аспектів взаємодії користувачів.

Для досягнення мети необхідно вирішити такі завдання:

1. Провести аналіз теоретичних основ інтеграції штучного інтелекту у веб-додатки, дослідити ключові підгалузі ШІ та їх роль у покращенні користувацького досвіду.

2. Здійснити аналіз існуючих веб-додатків та підходів до розробки з елементами ШІ, визначити основні етапи та сфери застосування ШІ для оптимізації інтерфейсу та аналізу поведінки користувачів.

3. Розробити концептуальну модель та прототип веб-додатку з інтегрованими функціями штучного інтелекту, спрямованими на покращення взаємодії користувачів, та оцінити потенційну ефективність запропонованих рішень.

Об'єкт дослідження – процес інтеграції технологій штучного інтелекту у функціонал веб-додатків.

Предмет дослідження – методи, моделі та інструменти інтеграції штучного інтелекту для покращення показників взаємодії користувачів із веб-додатком.

Методи дослідження. Дослідження базується на методах системного аналізу науково-технічної літератури для вивчення предметної області та існуючих рішень; методах порівняльного аналізу для оцінки різних підходів до інтеграції ШІ; методах моделювання для розробки концептуальної моделі веб-додатку; методах прототипування для практичної реалізації елементів ШІ; а також на методах експертних оцінок та аналізу користувацьких даних для оцінки ефективності запропонованих рішень.

Інформаційна база дослідження включає наукові статті та публікації з вітчизняних і зарубіжних джерел (зокрема, з баз Scopus, Web of Science, IEEE Xplore, ACM Digital Library), монографії, технічну документацію до програмних засобів та платформ ШІ, матеріали науково-практичних конференцій, аналітичні звіти та ресурси мережі Інтернет, присвячені тематиці штучного інтелекту та веб-розробки.

Практична значущість одержаних результатів полягає у розробці рекомендацій та моделі інтеграції ШІ, які можуть бути використані розробниками веб-додатків для створення більш інтелектуальних, персоналізованих та ефективних рішень, що сприятимуть підвищенню залученості та задоволеності користувачів. Результати можуть бути застосовані для модернізації існуючих або розробки нових веб-додатків у різних сферах.

Апробація результатів роботи. Основні положення та результати дослідження планується доповісти та обговорити на науковій конференції здобувачів вищої освіти за результатами науково-дослідної роботи у 2024-2025 роках у Полтавському державному аграрному університеті.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ ШТУЧНОГО ІНТЕЛЕКТУ У ВЕБ-ДОДАТКИ

1.1 Сутність штучного інтелекту та його ключові підгалузі

Штучний інтелект – це широка галузь комп'ютерних наук, що займається створенням систем, здатних виконувати завдання, які зазвичай потребують людського інтелекту. До таких завдань належать навчання, розв'язання проблем, сприйняття, розуміння мови, прийняття рішень та розпізнавання образів [3]. Головною метою ШІ є розробка машин, які можуть імітувати когнітивні функції людини, аналізувати складні дані та адаптуватися до нових ситуацій. Штучний інтелект можна класифікувати за різними критеріями, наприклад, за рівнем «інтелектуальності» (слабкий або вузький ШІ, який спеціалізується на конкретних завданнях, та сильний або загальний ШІ, що має здібності, аналогічні людським) або за функціональністю (реактивні машини, машини з обмеженою пам'яттю, теорія розуму, самосвідомість).

Однією з найважливіших та найпоширеніших підгалузей штучного інтелекту є машинне навчання (Machine Learning, ML). Машинне навчання – це підхід, за якого комп'ютерні системи не програмуються явно для виконання певного завдання, а «вчаться» на основі великих обсягів даних (так званих навчальних вибірок) та досвіду (рис. 1.1).

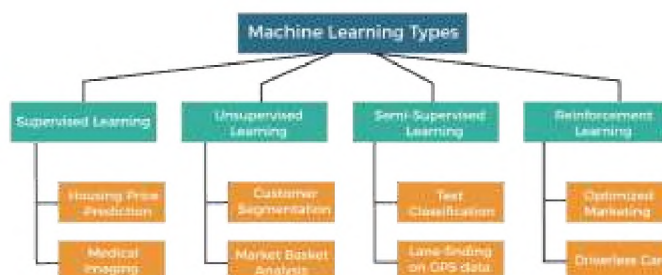


Рисунок 1.1 – Типи машинного навчання

Алгоритми МН виявляють закономірності та структуру в даних, що дозволяє їм робити прогнози, класифікувати об'єкти або приймати рішення без

прямого втручання людини. Основна ідея МН полягає в тому, щоб дозволити машинам навчатися самостійно, покращуючи свою продуктивність із часом на основі нових даних.

В останні роки значного розвитку набуло глибоке навчання (Deep Learning, DL) [4], яке є підмножиною машинного навчання. Глибоке навчання базується на використанні штучних нейронних мереж зі складною багат шаровою архітектурою (звідси й термін «глибине») (рис. 1.2).

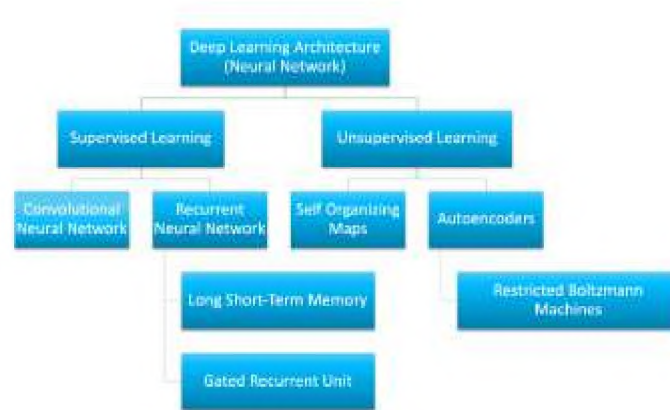


Рисунок 1.2 – Архітектури глибокого навчання

Ці «глибокі» нейронні мережі здатні автоматично виявляти ієрархічні ознаки з сирих даних, що робить їх надзвичайно ефективними для розв'язання складних завдань, таких як розпізнавання образів (зображень, відео), обробка природної мови (Natural Language Processing), генерація контенту та автономне керування. Велика кількість шарів дозволяє моделям DL вивчати ознаки на різних рівнях абстракції, від простих (наприклад, грані на зображенні) до складних.

1.2 Роль ШІ у покращенні користувацького досвіду

Користувацький досвід (User Experience) є фундаментальним аспектом, що визначає успішність сучасного веб-додатку. Він охоплює сукупність вражень, емоцій та результатів, які отримує користувач під час взаємодії з цифровим продуктом – його інтерфейсом, функціоналом та контентом.

Якісний UX передбачає, що користувач може легко, інтуїтивно та ефективно досягати своїх цілей, що не лише підвищує його задоволеність, але й сприяє формуванню лояльності до бренду, збільшенню конверсій та забезпеченню конкурентних переваг на ринку. У цьому контексті штучний інтелект виступає як потужний каталізатор трансформації UX, пропонуючи інструменти для створення глибоко персоналізованих, адаптивних та інтуїтивно зрозумілих веб-додатків [5].

Однією з найбільш вагомих переваг ШІ є його здатність забезпечувати безпрецедентний рівень персоналізації. Сучасні алгоритми машинного навчання можуть обробляти та аналізувати значні масиви даних про користувачів: їхню поведінку на платформі (історію переглядів, тривалість сесій, послідовність кліків), демографічні характеристики, вказані вподобання, історію транзакцій, а іноді, за згодою, й інформацію із зовнішніх джерел, наприклад, соціальних мереж. На основі цього комплексного аналізу ШІ дозволяє динамічно адаптувати контент, демонструючи кожному користувачеві саме ті матеріали, товари чи послуги, які є для нього найбільш релевантними. Це може бути індивідуалізована стрічка новин, добірка рекомендованих товарів у інтернет-магазині, або освітні модулі, що відповідають поточному рівню знань та навчальним цілям студента (рис. 1.3).

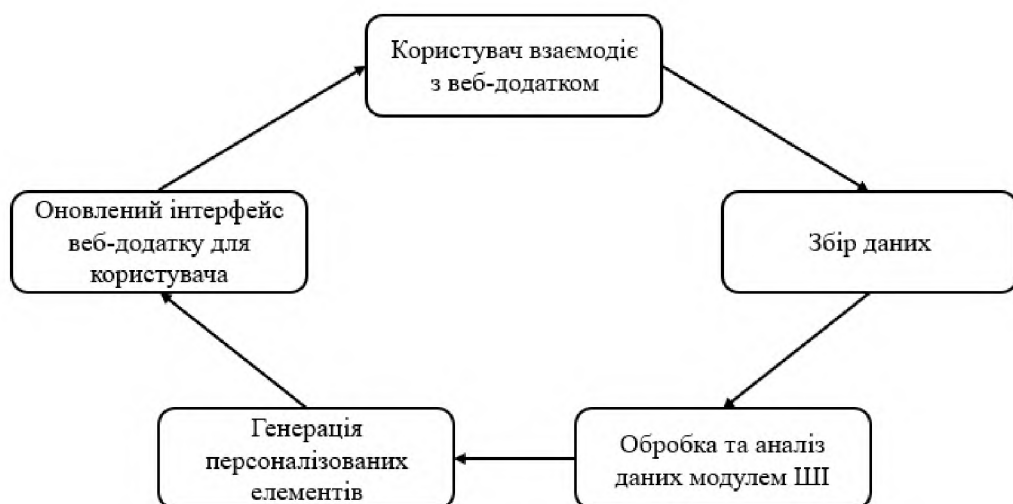


Рисунок 1.3 – Цикл персоналізації користувацького досвіду в веб-додатках на основі ШІ

Штучний інтелект (ШІ) кардинально покращує пошук у веб-додатках, виходячи за межі неефективного пошуку за точними ключовими словами. Завдяки семантичному аналізу ШІ розуміє справжній зміст запиту, навіть якщо він нечіткий [6].

Такі інновації, як голосовий, предиктивний та візуальний пошук (за зображенням), роблять взаємодію з додатком інтуїтивною та швидкою. Це спрощує навігацію, економить час користувача і значно підвищує його задоволеність, особливо на мобільних пристроях та в електронній комерції. Інтелектуальні чат-боти та віртуальні асистенти, керовані ШІ, перетворилися на стандартний інструмент для багатьох веб-додатків, особливо у сферах, що вимагають активної взаємодії з клієнтами, таких як електронна комерція, банківські послуги, технічна підтримка (рис. 1.4) [7].

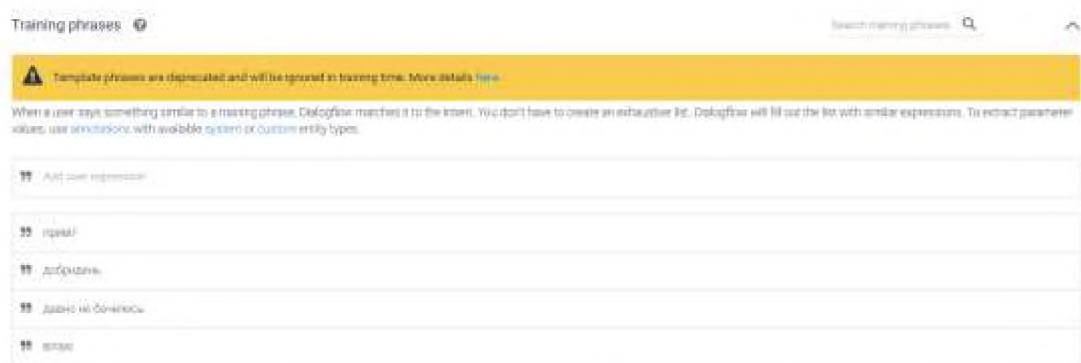


Рисунок 1.4 – Приклад навчання чат-бота взаємодії з користувачами

Цінність чат-ботів для UX полягає у миттєвій цілодобовій підтримці, що звільняє людей-операторів для вирішення складніших завдань. Завдяки прогресу в NLP, сучасні чат-боти ведуть осмислений діалог, розуміють контекст, допомагають з навігацією та можуть навіть проактивно починати розмову, аналізуючи поведінку користувача. Це оптимізує роботу підтримки та значно покращує враження користувачів [8].

Технології штучного інтелекту також відіграють ключову роль у доступності веб-додатків. ШІ може автоматично генерувати альтернативний текст для зображень, що допомагає користувачам із вадами зору. Крім того, системи розпізнавання та синтезу мовлення уможливають голосове

1.3 Огляд підходів та методів машинного навчання для веб-розробки

Ефективне використання машинного навчання у веб-розробці є запорукою створення справді інтелектуальних та адаптивних онлайн-сервісів. Як було окреслено раніше, МН надає інструменти для глибокого аналізу даних, виявлення прихованих закономірностей, побудови прогнозів та автоматизації прийняття рішень, що безпосередньо впливає на функціональність веб-додатків та якість користувацького досвіду. Для успішного впровадження цих технологій необхідно мати чітке уявлення про фундаментальні підходи до навчання, а також про основні типи та методи машинного навчання, що знаходять своє застосування у веб-середовищі [11].

В основі здатності машин «навчатися» лежать різні концептуальні підходи. Два ключові підходи, що визначають спосіб здобуття знань системою, – це індуктивне та дедуктивне навчання.

Індуктивне навчання – це основний підхід у сучасному машинному навчанні, де загальні правила створюються на основі аналізу конкретних прикладів (навчальних даних). Система виявляє закономірності для прогнозування результатів на нових даних. Цей принцип лежить в основі завдань класифікації та регресії. Наприклад, у веб-розробці він дозволяє створювати системи рекомендацій товарів, аналізуючи історію покупок користувачів.

Дедуктивне навчання, навпаки, робить конкретні висновки із загальних правил, закладених у базу знань (аксіоми, факти). Цей підхід характерний для експертних систем. Хоча він рідше використовується для аналізу великих даних через складність створення бази знань, його можна інтегрувати в гібридні системи. Наприклад, бізнес-правила (загальні знання) у веб-додатку можуть доповнювати або коригувати рішення, запропоновані індуктивними моделями.

Залежно від характеру наявних даних та способу, в який модель отримує зворотний зв'язок під час навчання, прийнято виділяти три фундаментальні типи машинного навчання: навчання з учителем, навчання без учителя та навчання з підкріпленням. Кожен із цих типів має свої особливості, методи та

сфери застосування у веб-розробці. Порівняння типів МН наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика основних типів машинного навчання.

Характеристика	Навчання з учителем (Supervised Learning)	Навчання без учителя (Unsupervised Learning)	Навчання з підкріпленням (Reinforcement Learning)
Основний принцип	Навчання на прикладах з відомими «правильними відповідями»	Пошук внутрішньої структури та закономірностей у даних без міток	Навчання шляхом взаємодії з середовищем та отримання винагород/штрафів за дії
Тип даних	Розмічені дані (вхідні ознаки + вихідні мітки/значення)	Нерозмічені дані (лише вхідні ознаки)	Дані про стани, дії та винагороди, що генеруються в процесі взаємодії
Типові задачі	Класифікація, регресія	Кластеризація, зменшення розмірності, пошук асоціативних правил	Прийняття послідовних рішень, оптимізація стратегій, керування
Приклади алгоритмів	Дерева рішень, SVM, Логістична регресія, k-NN, Нейронні мережі	k-Means, Ієрархічна кластеризація, PCA, Apriori	Q-learning, SARSA, DQN, Policy Gradients
Ключові застосування у веб	Фільтрація спаму, аналіз тональності, прогнозування попиту, розпізнавання образів	Сегментація користувачів, виявлення аномалій, рекомендаційні системи (частково)	Динамічна персоналізація UI/UX, адаптивні чат-боти, оптимізація рекламних стратегій

Навчання з учителем є найпоширенішим підходом у машинному навчанні, де модель тренується на розмічених даних для вивчення залежності між входом і виходом, щоб потім генерувати коректні відповіді для нових даних. Дві основні задачі тут – класифікація та регресія. У задачах класифікації модель відносить об'єкт до визначених категорій. Наприклад, вона може фільтрувати спам або небажаний контент, автоматично модерувати коментарі та аналізувати тональність відгуків, що класифікуються як

позитивні, негативні чи нейтральні. Важливим застосуванням також є розпізнавання образів для автоматичного тегування зображень або біометричної автентифікації.

Задачі регресії спрямовані на прогнозування неперервних числових значень. У веб-додатках регресійні моделі можуть прогнозувати попит на товари, оптимізувати складські запаси, оцінювати ймовірність кліку на рекламу або прогнозувати час перебування користувача на сторінці для оптимізації контенту. На відміну від цього, навчання без учителя працює з нерозміченими даними, щоб самостійно виявити приховані структури, закономірності, аномалії або внутрішні зв'язки. Це робить його незамінним для попереднього аналізу даних та отримання неочевидних інсайтів. Ключові задачі тут – кластеризація, зменшення розмірності та пошук асоціативних правил.

Кластеризація об'єднує схожі об'єкти в групи. Популярний алгоритм k-Means часто використовується у веб-розробці для сегментації користувачів на основі поведінкових даних, дозволяючи створювати персоналізовані пропозиції. Кластеризація також застосовується для виявлення аномалій, що можуть свідчити про шахрайство або збої. Пошук асоціативних правил виявляє співзалежності між подіями або об'єктами у великих даних. Класичним прикладом є аналіз ринкового кошика в електронній комерції, де алгоритми знаходять товари, що часто купують разом, що використовується для перехресних продажів та оптимізації пропозицій.

Навчання з підкріпленням – це принципово інший підхід, де агент навчається оптимальній поведінці шляхом взаємодії з середовищем. Агент здійснює дії, отримуючи винагороду чи штраф, прагнучи максимізувати сукупну винагороду. Яскравими прикладами алгоритмів є Q-learning. У веб-розробці навчання з підкріпленням застосовується для динамічної персоналізації інтерфейсу, де ШІ-агент у реальному часі адаптує елементи веб-сторінки для максимізації залученості користувача.

РОЗДІЛ 2

АНАЛІЗ ТА ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ З ІНТЕГРАЦІЄЮ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Відмінності веб-додатків від веб-сайтів та їх ключові компоненти

У сучасному цифровому просторі терміни «веб-сайт» та «веб-додаток» часто вживають як взаємозамінні, проте між ними існують суттєві концептуальні та технічні відмінності. Розуміння цих відмінностей критично важливе, особливо при плануванні розробки складних інтерактивних систем, таких як ті, що передбачають інтеграцію технологій штучного інтелекту. Чітке розмежування дозволяє правильно визначити архітектуру, технологічний стек, обсяг робіт та очікувану функціональність майбутнього продукту.

Традиційний веб-сайт – це набір статичних сторінок під одним доменом, що головним чином надає інформацію, діючи як цифрова брошура. Його інтерактивність обмежена гіперпосиланнями, простими формами чи базовим пошуком. Хоча CMS спрощують оновлення, фокус залишається на презентації контенту, а не складній взаємодії. Технологічна основа таких сайтів зазвичай включає HTML, CSS та JavaScript [16].

Веб-додаток, натомість, є програмним забезпеченням, доступ до якого здійснюється через веб-браузер, але його функціональність виходить далеко за межі простого інформування. Веб-додатки призначені для виконання конкретних завдань, надання сервісів та забезпечення високого рівня інтерактивності з користувачем. Вони дозволяють користувачам не просто споживати контент, а й активно взаємодіяти з даними: створювати, редагувати, видаляти, обробляти інформацію. Прикладами веб-додатків є системи онлайн-банкінгу, платформи електронної комерції, соціальні мережі, поштові клієнти (Gmail), офісні пакети (Google Docs) та системи управління проектами (рис. 2.1). Ключовою характеристикою веб-додатку є його здатність реагувати

на дії користувача, обробляти введені дані, зберігати їх та надавати персоналізований досвід [17].

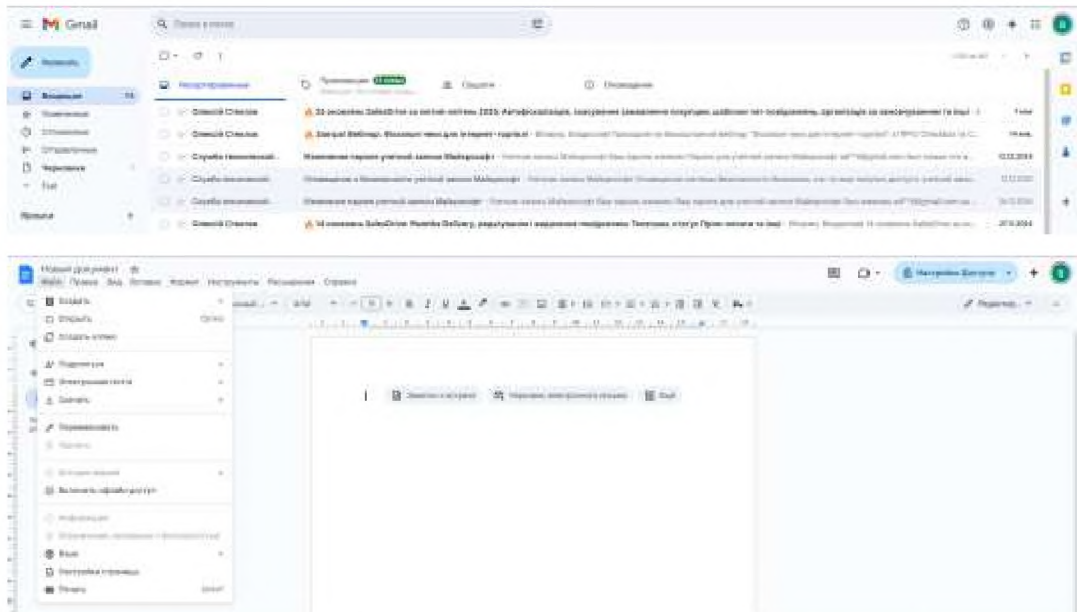


Рисунок 2.1 – Приклади веб-додатків: поштові клієнти, офісні пакети

Контент у веб-додатках є динамічним і часто генерується «на льоту» на основі взаємодії користувача та даних, що зберігаються в базах даних. Технологічний стек веб-додатків значно складніший і включає потужні фронтенд-фреймворки (наприклад, React, Angular, Vue.js) для створення багатих користувацьких інтерфейсів, а також складну серверну логіку, реалізовану за допомогою мов програмування, таких як Python (з фреймворками Django, Flask), Java (Spring), Node.js (Express), Ruby (Rails) та PHP (Laravel, Symfony), а також системи управління базами даних SQL або NoSQL.

Хоча межа між веб-сайтом та веб-додатком іноді може бути розмитою, фундаментальні відмінності зберігаються. Ключова відмінність полягає у призначенні: веб-сайт переважно інформує, тоді як веб-додаток дозволяє виконувати дії та взаємодіяти. Це, в свою чергу, визначає різницю в рівні інтерактивності (низький для сайтів, високий для додатків), складності

архітектури та логіки, а також у вимогах до автентифікації користувачів та управління станом сесії, які є типовими для веб-додатків. Обробка та маніпулювання даними є серцевиною веб-додатку, тоді як для сайту це другорядна функція, якщо взагалі присутня. Розуміння цих відмінностей стає особливо важливим при розгляді інтеграції штучного інтелекту, оскільки ШІ-функціонал за своєю природою вимагає високого рівня інтерактивності, обробки великих обсягів даних та складної логіки, що є характерними рисами саме веб-додатків.

Будь-який сучасний веб-додаток складається з кількох фундаментальних компонентів, що взаємодіють між собою для забезпечення його функціональності: клієнтської та серверної частин.

Клієнтська частина, або фронтенд, – це все, з чим безпосередньо взаємодіє користувач у своєму веб-браузері. Її головне завдання – забезпечити інтуїтивно зрозумілий та ефективний користувацький інтерфейс та позитивний користувацький досвід. Фронтенд відповідає за візуальне представлення даних, обробку введення користувача та надсилання запитів на серверну частину. Основними технологіями, що використовуються для розробки фронтенду, є HTML, що визначає структуру та семантичний зміст веб-сторінки; CSS, що відповідає за візуальне оформлення HTML-елементів; та JavaScript, що «оживляє» веб-сторінку, додаючи інтерактивність, динамічне оновлення контенту без перезавантаження сторінки (AJAX), валідацію форм та обробку подій користувача. Для створення складних та високопродуктивних користувацьких інтерфейсів широко використовуються JavaScript-фреймворки та бібліотеки, такі як React.js, Angular та Vue.js. Вони пропонують компонентний підхід до розробки та інструменти для управління станом додатку, що прискорюють та стандартизують процес. Фронтенд взаємодіє з бекендом за допомогою HTTP-запитів до API, надсилаючи та отримуючи дані для відображення. Браузери також надають можливості для локального

зберігання даних, що може використовуватися для кешування або управління сесіями.

Серверна частина, або бекенд, – це «мозок» веб-додатку, що працює на веб-сервері та невидимий для кінцевого користувача. Бекенд відповідає за обробку бізнес-логіки, взаємодію з базами даних, автентифікацію та авторизацію користувачів, а також за надання даних клієнтській частині через API. Ключові елементи бекенду включають серверні мови програмування та фреймворки, такі як Python (Django, Flask), Java (Spring), JavaScript (Express.js), Ruby (Ruby on Rails), PHP (Laravel, Symfony) та C# (.NET). Веб-додатки потребують зберігання та управління великими обсягами даних, тому використовуються різні бази даних: реляційні (MySQL, PostgreSQL) для структурованих даних з потребою в транзакційності, та нереляційні (MongoDB, Cassandra) для більшої гнучкості схеми даних та масштабованості, часто для неструктурованих даних або кешування. API (зазвичай RESTful API або GraphQL) є контрактом, що дозволяє фронтенду обмінюватися даними з бекендом, а також важливі для інтеграції з іншими сторонніми сервісами, включаючи платформи штучного інтелекту. Веб-сервери, такі як Apache або Nginx, приймають та обробляють HTTP-запити від клієнтів, відповідаючи за балансування навантаження, SSL/TLS шифрування та кешування. Системи автентифікації та авторизації перевіряють особу користувача та його права доступу [18].

У типовому сценарії роботи веб-додатку користувач взаємодіє з фронтендом у браузері. Дії користувача обробляються JavaScript-кодом, який формує HTTP-запит до API на бекенді. Бекенд отримує запит, виконує необхідну бізнес-логіку (включаючи звернення до бази даних або зовнішніх сервісів, зокрема ШІ-сервісів), формує відповідь і надсилає її фронтенду. Фронтенд обробляє дані та динамічно оновлює інтерфейс. Інтеграція штучного інтелекту в таку архітектуру може відбуватися на різних рівнях. ШІ-

моделі для обробки природної мови або рекомендаційні системи можуть бути розгорнуті як частина бекенду або як окремі мікросервіси, з якими бекенд взаємодіє через внутрішні API. Результати роботи ШІ-моделей потім передаються на фронтенд для відображення. Фронтенд також може використовувати ШІ-бібліотеки, що працюють безпосередньо в браузері, хоча більш складні та ресурсомісткі ШІ-операції зазвичай виконуються на серверній стороні.

2.2 Етапи розробки веб-додатку з елементами ШІ

Розробка веб-додатку, особливо з компонентами штучного інтелекту, є складним, багатоетапним процесом, що вимагає ретельного планування та координації. Інтеграція ШІ не просто додає функціонал, а суттєво впливає на всі етапи життєвого циклу продукту. На відміну від традиційної розробки, ШІ-проекти часто мають вищий ступінь невизначеності та експериментальний характер, особливо на етапах дослідження даних та навчання моделей. Сучасні підходи, такі як Agile, Scrum або Kanban, дозволяють ефективно управляти цією складністю завдяки ітеративності, гнучкості та постійному зворотному зв'язку [19].

Ключові етапи розробки веб-додатку з елементами ШІ починаються з Планування та Аналізу. Це фундаментальний етап, на якому закладаються основи майбутнього проєкту. Насамперед відбувається визначення бізнес-цілей та проблем, які має вирішувати веб-додаток, а також цільової аудиторії. Ключовим аспектом для ШІ-проектів є аналіз доцільності застосування ШІ, щоб об'єктивно оцінити, чи є ШІ оптимальним рішенням та які переваги він забезпечить. На цьому ж етапі формулюються вимоги до ШІ-компонентів: що саме має робити інтелектуальна система та якими характеристиками вона повинна володіти. Особлива увага приділяється дослідженню даних: їх наявності, доступності, якості, релевантності та обсягу для навчання та тестування моделей, а також виявленню потенційних проблем та стратегій їх

збору чи очищення. На основі аналізу завдань та даних відбувається попередній вибір підходів та технологій ШІ, а також оцінка ризиків, пов'язаних як з розробкою веб-додатку в цілому, так і специфічних для ШІ. На завершення етапу планування формується команда проєкту, до якої входять спеціалісти з даних (Data Scientists, ML Engineers), складається дорожня карта та попередній бюджет [20].

На етапі Проектування деталізуються вимоги та створюється архітектурний план майбутнього веб-додатку та його інтелектуальних компонентів. UX-дизайн набуває особливого значення при інтеграції ШІ, вимагаючи ретельного продумування взаємодії користувачів з інтелектуальними функціями, прозорості рішень ШІ та можливості контролю. Розробляються детальні сценарії використання та схеми взаємодії. Далі йде UI-дизайн, який втілює UX-концепції у візуально привабливий та зручний інтерфейс, створюючи вайрфрейми, макети та інтерактивні прототипи. Паралельно відбувається архітектурне проектування системи, вибір загальної архітектури, структури бази даних та технологій для фронтенду та бекенду. Надзвичайно важливим є проектування API для взаємодії між клієнтською частиною, серверною логікою та окремими ШІ-сервісами. Окремо здійснюється проектування ШІ-компонентів, деталізація архітектури моделей, методів їх навчання та оцінки, а також проектування конвеєрів обробки даних та визначення ключових метрик якості роботи ШІ.

Етап Розробки – це безпосереднє створення програмного продукту на основі затверджених проєктних рішень, що зазвичай поділяється на розробку ШІ-моделей, бекенду та фронтенду, які можуть відбуватися паралельно. Розробка ШІ-моделей є наукомістким та ітеративним процесом, що починається зі збору та ретельної підготовки даних, їх очищення, трансформації та розмітки. Далі відбувається навчання моделей, експериментування з гіперпараметрами для досягнення найкращої якості, валідація та тестування на відкладених наборах даних, а також оптимізація для забезпечення необхідної продуктивності. Розробка Бекенду включає реалізацію всієї серверної логіки, створення API, налаштування взаємодії з базами даних, механізми автентифікації та авторизації. Критично важливим

завданням є інтеграція розроблених ШІ-моделей. Розробка Фронтенду полягає у перетворенні UI-макетів на інтерактивний інтерфейс за допомогою HTML, CSS та JavaScript-фреймворків. Фронтенд має коректно взаємодіяти з API бекенду та ефективно відображати результати роботи ШІ-функцій користувачеві.

Тестування є невід'ємним етапом для забезпечення якості, надійності та безпеки веб-додатку та його ШІ-компонентів. Особливу увагу приділяють тестуванню ШІ-компонентів, що включає перевірку функціональної коректності, оцінку якості моделей за метриками на тестових даних, тестування на стійкість до нетипових вхідних даних (adversarial testing) та на упередженість (bias testing). Поряд із цим, проводяться традиційні види тестування для всього веб-додатку: функціональне, UI/UX тестування (включаючи юзабіліті-тестування з реальними користувачами), тестування продуктивності та навантаження, тестування безпеки. Важливим є інтеграційне тестування для перевірки коректності взаємодії всіх компонентів системи [21].

Після успішного тестування веб-додаток готовий до Розгортання та Релізу. На цьому етапі відбувається підготовка інфраструктури: налаштування серверів, баз даних, середовищ для виконання ШІ-моделей. Все частіше для автоматизації процесів збірки, тестування та розгортання використовуються практики CI/CD. Відбувається безпосереднє розгортання веб-додатку та ШІ-моделей на підготовлену інфраструктуру. Перед офіційним запуском проводиться фінальне тестування на робочому середовищі. Після цього відбувається реліз – додаток стає доступним для цільової аудиторії.

Запуск веб-додатку – це не завершення, а початок нового етапу його життєвого циклу – Підтримка та Розвиток. Важливою складовою є постійний моніторинг роботи додатку та ШІ-моделей. Необхідно відстежувати технічні показники та якість роботи ШІ-компонентів, оскільки для ШІ-моделей характерне явище деградації якості (model drift або concept drift) з часом через зміни у вхідних даних чи середовищі. Здійснюється збір та аналіз зворотного зв'язку від користувачів для виявлення проблем та можливостей для покращення. На основі результатів моніторингу та зворотного зв'язку

проводиться регулярно оновлення та перенавчання ШІ-моделей на свіжих даних для підтримки їхньої ефективності. Також виконується виправлення помилок, оптимізація продуктивності та, за необхідності, додавання нових функцій, включаючи розширення або вдосконалення існуючих ШІ-рішень. Цей етап також є ітеративним і часто призводить до повернення на етап планування для розробки нових версій продукту [22].

2.3 Сфери застосування ШІ у веб-додатках

Штучний інтелект стрімко трансформує ландшафт сучасних веб-додатків, відкриваючи нові горизонти для функціональності, взаємодії з користувачем та бізнес-ефективності. Від персоналізації контенту до інтелектуальних систем підтримки та оптимізації інтерфейсів – технології ШІ дозволяють створювати більш інтуїтивні, адаптивні та цінні веб-рішення. Сфери застосування ШІ настільки різноманітні, наскільки й завдання, що стоять перед розробниками, і цей перелік постійно розширюється з розвитком самих технологій. Розглянемо ключові напрямки, де інтеграція ШІ вже сьогодні приносить значну користь та визначає майбутнє веб-розробки [23].

Однією з найбільш значущих та затребуваних сфер застосування ШІ у веб-додатках є персоналізація користувацького досвіду. Сучасні користувачі очікують, що веб-ресурси будуть не просто надавати інформацію, а й розуміти їхні індивідуальні потреби та вподобання. ШІ дозволяє реалізувати це на якісно новому рівні. Алгоритми машинного навчання аналізують величезні масиви даних про користувачів – їхню історію переглядів, поведінкові патерни, демографічні характеристики, контекст взаємодії (пристрій, місцезнаходження, час доби) – для створення унікального досвіду для кожного.

Це проявляється, наприклад, у динамічній доставці контенту: новинні портали можуть формувати індивідуальні стрічки новин, освітні платформи – пропонувати навчальні матеріали, що відповідають рівню знань та інтересам

студента, а блоги – показувати статті на теми, які найбільше цікавлять конкретного читача.

Особливо яскраво персоналізація виражена в рекомендаційних системах. Платформи електронної комерції використовують ШІ для пропозиції товарів, які можуть зацікавити покупця на основі його попередніх покупок, переглядів або схожості з іншими користувачами. Стрімінгові сервіси, такі як Netflix чи Spotify, пропонують фільми, серіали та музику, що максимально відповідають смакам користувача. Соціальні мережі рекомендують друзів, групи чи контент, підвищуючи залученість.

Окрім контенту, ШІ може використовуватися для адаптації самого інтерфейсу веб-додатку, динамічно змінюючи розташування елементів, навігаційні шляхи або навіть візуальний стиль для оптимізації взаємодії конкретного користувача [24]. Такий індивідуальний підхід не лише підвищує задоволеність та лояльність, але й суттєво впливає на бізнес-показники, такі як конверсія та середній чек (рис. 2.2).

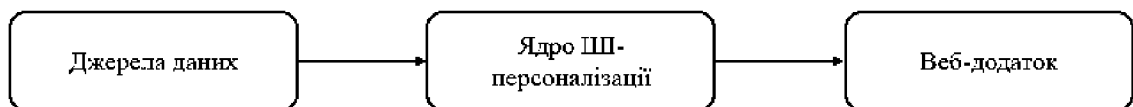


Рисунок 2.2 – Модель персоналізації користувацького досвіду за допомогою ШІ у веб-додатку

ШІ кардинально змінює способи взаємодії користувачів з веб-додатками, роблячи їх більш природними, ефективними та доступними. Центральне місце тут займають інтелектуальні чат-боти та віртуальні асистенти, які стали невід'ємною частиною багатьох сучасних веб-ресурсів, особливо у сферах обслуговування клієнтів, електронної комерції та надання онлайн-послуг.

Такі системи, що базуються на технологіях обробки природної мови та розуміння природної мови, здатні вести діалог з користувачем, розуміти його

запити, сформульовані звичайною мовою, та надавати релевантні відповіді або виконувати певні дії. Вони забезпечують цілодобову підтримку клієнтів, відповідаючи на поширені запитання, допомагаючи з навігацією по сайту, оформленням замовлення, вирішенням простих проблем, тим самим розвантажуючи людських операторів для складніших завдань. Просунуті чат-боти можуть аналізувати контекст діалогу, підтримувати багатоходові розмови, персоналізувати спілкування і навіть проявляти елементи емпатії. Деякі з них здатні до проактивної взаємодії, ініціюючи діалог, якщо система фіксує, що користувач може потребувати допомоги.

Окрім текстових чат-ботів, все більшої популярності набувають голосові інтерфейси у веб-додатках. Можливість керувати функціями сайту або здійснювати пошук за допомогою голосу значно підвищує зручність, особливо для користувачів мобільних пристроїв або людей з обмеженими фізичними можливостями. Інтеграція з голосовими асистентами, такими як Google Assistant чи Amazon Alexa, також розширює канали взаємодії з веб-додатком (рис. 2.3).

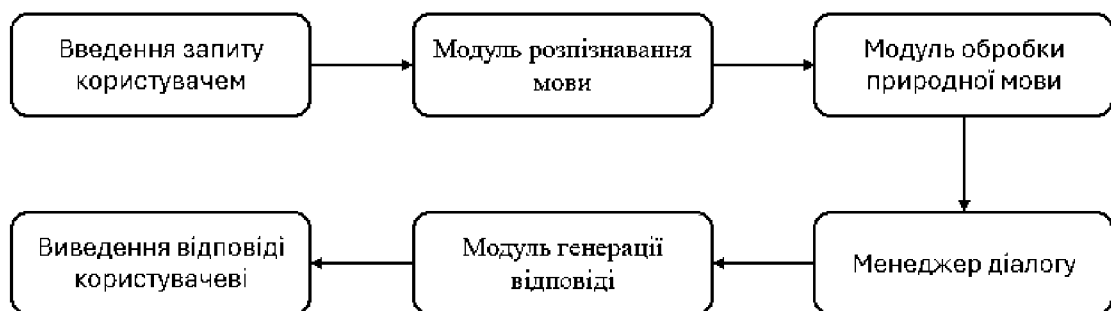


Рисунок 2.3 – Архітектура та принцип роботи інтелектуального чат-бота у веб-додатку

Якість контенту та легкість його знаходження є критичними факторами успіху будь-якого веб-ресурсу. ШІ пропонує потужні інструменти для оптимізації обох цих аспектів [25].

Інтелектуальний пошук у веб-додатках виходить за межі простого зіставлення ключових слів. Завдяки семантичному аналізу та розумінню природної мови, пошукові системи на базі ШІ можуть краще інтерпретувати наміри користувача, враховувати контекст запиту, розпізнавати синоніми та надавати значно релевантніші результати. Функції автодоповнення запитів, виправлення помилок та персоналізації результатів пошуку на основі попередньої активності користувача також реалізуються за допомогою ШІ.

ШІ знаходить застосування і в автоматизованому створенні та куруванні контенту. Хоча повна заміна людської творчості поки що неможлива, алгоритми генерації природної мови (NLG) вже сьогодні можуть створювати короткі новини, описи товарів, звіти за даними, підсумки статей. Це особливо корисно для завдань, що вимагають обробки та структурування великих обсягів інформації. Крім того, ШІ допомагає в автоматичному тегуванні та категоризації контенту, що покращує його організацію, доступність та полегшує пошук [26].

Технології ШІ також використовуються для SEO-оптимізації (Search Engine Optimization). Аналізуючи великі обсяги даних про пошукові тренди, поведінку користувачів на сторінках, ефективність ключових слів та стратегії конкурентів, ШІ-інструменти можуть надавати рекомендації щодо покращення контенту, структури сайту та технічних аспектів для досягнення вищих позицій у результатах пошукових систем.

Забезпечення високого рівня юзабіліті та привабливого інтерфейсу є ключовим завданням веб-розробки. ШІ надає нові можливості для досягнення цієї мети, особливо в контексті різноманітності пристроїв та індивідуальних потреб користувачів.

Одним із важливих напрямків є оптимізація інтерфейсу для мобільних пристроїв. Якщо традиційний респонсивний дизайн забезпечує адаптацію макету до розміру екрана, то ШІ може піти далі. Аналізуючи специфічні патерни використання веб-додатку на мобільних пристроях, ШІ може динамічно коригувати розмір інтерактивних елементів, шрифти,

контрастність, розташування ключових кнопок або навіть спрощувати певні функції для покращення мобільного UX. Це може включати персоналізовані швидкі дії або навігаційні меню, оптимізовані для використання однією рукою.

ШІ також революціонує процес А/В тестування та багатовимірної оптимізації. Замість ручного створення та тестування обмеженої кількості варіантів дизайну, ШІ-системи можуть автоматично генерувати та тестувати сотні й тисячі варіацій UI-елементів (кольорів, текстів кнопок, зображень, їхнього розташування), швидко визначаючи найбільш ефективні комбінації для досягнення цільових показників.

Крім того, машинне навчання використовується для глибокого аналізу поведінки користувачів на сайті. Обробляючи дані з теплових карт, записів сесій, карт кліків та шляхів користувачів, ШІ допомагає виявляти проблемні місця в інтерфейсі, де користувачі відчувають труднощі, плутаються або залишають сайт. Це дозволяє приймати обґрунтовані рішення щодо покращення навігації, структури сторінок та загальної зручності використання.

Безпека є критично важливим аспектом для будь-якого веб-додатку, особливо для тих, що обробляють конфіденційні дані користувачів або фінансові транзакції. ШІ пропонує ефективні інструменти для підвищення рівня безпеки.

Системи на основі машинного навчання здатні виявляти аномальну поведінку користувачів в режимі реального часу. Аналізуючи типові патерни активності, ШІ може ідентифікувати незвичайні спроби входу, підозрілі транзакції, спроби підбору паролів або інші дії, що можуть свідчити про компрометацію облікового запису або шахрайську діяльність.

Біометрична автентифікація, що використовує ШІ для розпізнавання облич, голосу, відбитків пальців або навіть патернів набору тексту, забезпечує додатковий рівень захисту.

ШІ також ефективно використовується для захисту від автоматизованих атак, таких як DDoS, та для фільтрації спаму і блокування шкідливих ботів, які

можуть намагатися отримати несанкціонований доступ, збирати дані або порушувати роботу сервісу.

Веб-додатки генерують величезні обсяги даних про взаємодію користувачів, ефективність контенту, технічну продуктивність тощо. ШІ дозволяє не просто збирати ці дані, а й перетворювати їх на цінні знання для бізнесу.

Прогнозна аналітика на основі МН може передбачати майбутні тенденції та події, такі як відтік користувачів, їхню довічну цінність, ймовірність конверсії для різних сегментів аудиторії, попит на певні товари чи послуги.

Інтелектуальний аналіз даних допомагає виявляти приховані закономірності та кореляції у великих масивах даних, які важко помітити традиційними методами. Це дозволяє краще розуміти потреби клієнтів, оптимізувати маркетингові кампанії, вдосконалювати продукти та приймати більш обґрунтовані стратегічні рішення.

ШІ також може використовуватися для автоматизованого формування звітів та дашбордів, агрегуючи ключові показники та представляючи їх у зрозумілому вигляді для менеджменту [27].

Окрім загальних напрямків, ШІ знаходить унікальні застосування у веб-додатках для конкретних галузей. В електронній комерції це, окрім персоналізації та чат-ботів, динамічне ціноутворення, оптимізація ланцюгів поставок, боротьба з підробленими відгуками. В освітніх веб-платформах ШІ дозволяє створювати адаптивні навчальні траєкторії, автоматично оцінювати завдання, надавати персоналізований зворотний зв'язок учням. У сфері охорони здоров'я веб-додатки з ШІ можуть допомагати в попередній діагностиці на основі симптомів, аналізі медичних зображень, моніторингу стану пацієнтів. У фінансовому секторі це можуть бути веб-платформи для робо-едвайзингу, кредитного скорингу, виявлення шахрайських транзакцій.

2.4 Аналіз поведінки користувачів та оптимізація інтерфейсу

Створення успішного веб-додатку не завершується на етапі його запуску. Для забезпечення довгострокової ефективності, високого рівня залученості та задоволеності користувачів необхідно постійно аналізувати, як саме вони взаємодіють з інтерфейсом, та на основі цих даних проводити його оптимізацію. Розуміння поведінкових патернів, виявлення «больових точок» та зон, де користувачі відчувають труднощі, є ключем до прийняття обґрунтованих рішень щодо покращення дизайну та функціональності. Сучасні інструменти аналітики, доповнені методами машинного навчання, надають потужні можливості для глибокого дослідження користувацького досвіду та його цілеспрямованого вдосконалення [28].

Існує низка методів та інструментів, що дозволяють збирати та аналізувати дані про взаємодію користувачів з веб-додатком. Кожен з них надає унікальний погляд на поведінку та допомагає формувати комплексну картину.

Системи веб-аналітики, як-от Google Analytics, є основним інструментом для будь-якого веб-ресурсу. Вони дозволяють збирати великий обсяг кількісних даних про відвідувачів та їхню активність. Google Analytics відстежує загальний трафік, джерела переходу (пошук, соцмережі, прямі заходи), демографію та географію аудиторії (рис. 2.4).



Рисунок 2.4 – Приклад дашборду системи веб-аналітики для моніторингу ключових показників поведінки користувачів

Система також фіксує ключові метрики залученості: кількість переглянутих сторінок за сесію, середню тривалість сесії, показник відмов,

шляхи навігації. Вона дозволяє налаштовувати відстеження досягнення конкретних цілей, наприклад, реєстрацій чи замовлень. Аналіз цих даних допомагає зрозуміти популярність контенту, причини залишення сайту користувачами та ефективність каналів залучення. Ця інформація є відправною точкою для формування гіпотез щодо проблем в інтерфейсі та їх оптимізації.

Теплові карти є інструментом візуалізації, який наочно демонструє, як користувачі взаємодіють з окремими елементами веб-сторінки. Вони агрегують дані про активність багатьох користувачів і представляють їх у вигляді кольорових накладень на сторінку. Існують декілька основних типів теплових карт:

Кarti кліків показують, на які елементи сторінки користувачі найчастіше натискають. «Гарячі» зони (де багато кліків) позначаються теплими кольорами (червоний, жовтий), а «холодні» – холодними (синій, зелений). Це допомагає зрозуміти, які кнопки, посилання чи зображення привертають найбільше уваги, а які залишаються непоміченими, або чи не клікають користувачі на неінтерактивні елементи [29].

Кarti скролінгу візуалізують, наскільки глибоко користувачі прокручують сторінку. Верхня частина сторінки зазвичай «найгарячіша», а колір поступово змінюється до «холодного» вниз. Це дозволяє оцінити, чи бачать користувачі важливий контент, розміщений нижче «лінії згину», і яка частина сторінки отримує найменше уваги.

Кarti руху курсору відстежують рухи миші користувачів по сторінці, що часто корелює з напрямком їхньої уваги. Такі карти можуть допомогти зрозуміти, які області сторінки привертають погляд, навіть якщо користувач не здійснює клік.

Теплові карти надають цінні якісні інсайти, доповнюючи кількісні дані веб-аналітики, та допомагають оптимізувати розташування ключових елементів, закликів до дії (СТА) та контенту на сторінці.

Кластерний аналіз – це метод машинного навчання без учителя, що автоматично групує користувачів у сегменти (кластери) на основі схожості

їхньої поведінки. На відміну від заздалегідь визначених груп, кластеризація виявляє природні сегменти, ґрунтуючись на реальній активності у веб-додатку (рис. 2.5).

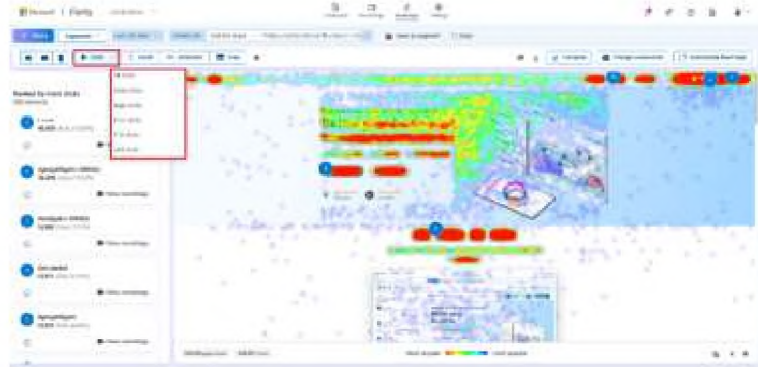


Рисунок 2.5 – Приклад теплової карти кліків для аналізу зон активності користувачів на веб-сторінці

Вхідними даними можуть бути метрики веб-аналітики, серверні логи або специфічні події взаємодії. Популярні алгоритми, такі як k-Means, об'єднують користувачів зі схожими поведінковими характеристиками. Наприклад, можна виділити сегменти на кшталт «активних покупців», «дослідників товарів» або «нових користувачів, що швидко залишають сайт». Розуміння цих сегментів дає змогу точніше адаптувати інтерфейс, контент і маркетингові активності до потреб кожної групи, а також виявляти користувачів із проблемами юзабіліті (рис. 2.6).

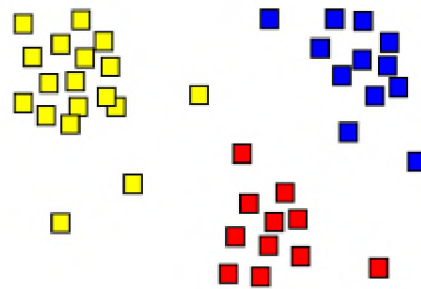


Рисунок 2.6 – Візуалізація результатів кластерного аналізу для сегментації користувачів веб-додатку за поведінковими характеристиками

Окрім зазначених, існують і інші корисні методи. Записи сесій дозволяють переглядати анонімізовані відеозаписи реальних сеансів взаємодії користувачів з сайтом, що дає можливість побачити їхній шлях, моменти вагань, помилки та загальні труднощі [30]. А/В тестування (або багатовимірне тестування) полягає у порівнянні двох або більше варіантів дизайну сторінки чи окремого елемента, щоб визначити, який з них працює краще з точки зору ключових метрик. Штучний інтелект може значно автоматизувати та оптимізувати процеси А/В тестування, аналізуючи результати та пропонуючи нові варіанти.

Зібрані дані та отримані інсайти є основою для процесу оптимізації користувацького інтерфейсу. Цей процес є ітеративним і включає кілька ключових кроків.

По-перше, на основі аналізу формулюються гіпотези щодо можливих проблем та шляхів їх вирішення.

По-друге, розробляються та впроваджуються зміни в дизайні або функціональності веб-додатку. Це можуть бути як незначні коригування (зміна кольору кнопки, тексту заклику до дії, розташування блоку), так і більш суттєві перебудови (зміна навігаційної структури, редизайн ключових сторінок). Важливо, щоб ці рішення базувалися на отриманих даних, а не на суб'єктивних припущеннях.

По-третє, після впровадження змін проводиться повторний аналіз та тестування, щоб оцінити ефективність внесених змін. Якщо гіпотеза підтвердилася і показники покращилися, зміни залишаються. Якщо ні – процес повторюється з новими гіпотезами.

Штучний інтелект відіграє все більшу роль не лише в аналізі, але й в самій оптимізації. Наприклад, веб-додаток може використовувати ШІ для динамічної персоналізації інтерфейсу в реальному часі для різних сегментів користувачів або навіть для окремих індивідів. Це може стосуватися показу різного контенту, пропозиції різних шляхів навігації або адаптації візуальних елементів.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ІНТЕГРАЦІЇ ШІ

3.1 Переваги та обмеження впровадження ШІ у веб-додатки

Інтеграція штучного інтелекту у веб-додатки є одним з найбільш динамічних напрямків у цифровій індустрії. ШІ обіцяє не просто покращення, а якісну трансформацію взаємодії користувачів та методів ведення онлайн-бізнесу, від персоналізації до автоматизації аналітичних завдань. Однак, реалізація цього потенціалу супроводжується значними викликами, обмеженнями та відповідальностями. Тому для будь-якої організації, що розглядає впровадження ШІ, критично важливим є всебічний аналіз як переваг, так і недоліків. Такий аналіз дозволяє сформулювати реалістичні очікування, ефективно розподілити ресурси, мінімізувати ризики та максимізувати ефект від інвестицій у ШІ.

Застосування технологій ШІ відкриває перед розробниками та власниками веб-додатків широкий спектр можливостей для підвищення їхньої цінності, конкурентоспроможності та ефективності [31].

ШІ значно покращує користувацький досвід завдяки глибокій персоналізації. Аналізуючи великі обсяги даних (історію переглядів, поведінкові патерни, демографію, покупки), ШІ динамічно адаптує контент, пропонуючи найрелевантніші статті, товари чи послуги. Це очевидно в рекомендаційних системах для електронної комерції, стрімінгових та освітніх платформ. Персоналізація може також оптимізувати інтерфейс, спрощуючи навігацію. Як результат, зростає задоволеність та лояльність користувачів, а також ключові бізнес-показники, як-от час перебування на сайті, глибина переглядів, коефіцієнт конверсії та середній чек.

Другою ключовою перевагою є автоматизація рутинних завдань та складних бізнес-процесів. Багато операцій у веб-додатках є повторюваними та

трудомісткими при ручному виконанні, але ШІ дозволяє перекласти їх на інтелектуальні системи.

Класичний приклад – інтелектуальні чат-боти та віртуальні асистенти, які завдяки обробці природної мови можуть цілодобово обробляти стандартні запити клієнтів, надавати інформаційну підтримку та допомагати з замовленнями, значно розвантажуючи контакт-центри та прискорюючи реакцію.

Автоматизація за допомогою ШІ також застосовується у модерації користувацького контенту, автоматичному тегуванні та категоризації інформації, інтелектуальній обробці документів та оптимізації логістики. Це не тільки скорочує операційні витрати та мінімізує людські помилки, а й дозволяє співробітникам зосередитися на більш творчих та стратегічних завданнях (рис. 3.1).



Рисунок 3.1 – Компоненти автоматизації бізнес-процесів

Штучний інтелект (ШІ) дозволяє глибоко аналізувати дані та отримувати цінні бізнес-інсайти. Веб-додатки генерують багато інформації про поведінку користувачів, маркетинг, продуктивність системи та ринкові тренди. Технології машинного навчання обробляють ці великі обсяги даних, виявляючи приховані закономірності та тенденції. Це допомагає компаніям краще розуміти аудиторію, сегментувати її, прогнозувати попит, виявляти причини відтоку клієнтів та оптимізувати ціноутворення й маркетингові

стратегії. Наприклад, предиктивна аналітика може не тільки прогнозувати події, а й визначати ключові впливові фактори, сприяючи ефективнішим управлінським рішенням.

У сукупності, ці переваги ШІ підвищують ефективність, продуктивність та інноваційний потенціал веб-додатків. Оптимізований пошук інформації, швидка клієнтська підтримка та автоматизовані процеси скорочують час виконання завдань для всіх [32]. Крім того, ШІ відкриває можливості для створення принципово нових функцій та інноваційних сервісів, які раніше були неможливі. Це може включати системи розпізнавання образів, голосове управління, генерацію контенту, адаптивні навчальні системи або складні прогностичні інструменти. Такі інновації не лише покращують користувацький досвід, а й можуть забезпечити ключову конкурентну перевагу.

Незважаючи на вражаючий потенціал, інтеграція ШІ у веб-додатки є складним завданням, що супроводжується низкою суттєвих обмежень та викликів, які потребують ретельного аналізу та продуманих стратегій їх подолання.

Одним з найбільш значущих бар'єрів є високі початкові витрати та значна складність процесу розробки. Впровадження ШІ – це не просто встановлення готового програмного модуля. Воно часто вимагає значних інвестицій у залучення висококваліфікованих спеціалістів, таких як Data Scientists, Machine Learning Engineers, спеціалісти з обробки та інженерії даних. Крім фінансових витрат на персонал, необхідно враховувати вартість збору, очищення, розмітки та підготовки даних. Також потрібні інвестиції у створення або оренду потужної обчислювальної інфраструктури для навчання складних моделей та їх подальшого розгортання. Сам процес розробки ШІ-рішень часто має дослідницький характер, вимагає численних ітерацій, експериментів з різними моделями та гіперпараметрами, і не завжди гарантує швидкий та передбачуваний результат, що ускладнює планування та бюджетування [33].

Критичною є залежність від якості, обсягу та доступності даних. Ефективність та надійність будь-якої ШІ-моделі, особливо тієї, що базується на машинному навчанні, прямо пропорційна якості даних, на яких вона була навчена. Принцип «сміття на вході – сміття на виході» тут працює безвідмовно. Якщо навчальні дані є неповними, нерепрезентативними, містять помилки, шуми або, що особливо небезпечно, системні упередження, то навчена модель буде відтворювати ці недоліки, генеруючи неточні, несправедливі або навіть шкідливі результати. Забезпечення достатнього обсягу якісних, релевантних даних, їх належна підготовка, а також дотримання суворих вимог щодо конфіденційності, безпеки та захисту персональних даних) є складним, безперервним та ресурсомістким завданням.

Впровадження ШІ актуалізує складні етичні виклики та питання юридичної відповідальності. Алгоритмічна упередженість може призводити до дискримінаційних практик, що особливо небезпечно у сферах кредитування чи працевлаштування. Проблема «чорної скриньки» означає відсутність прозорості у прийнятті рішень ШІ, що ускладнює виявлення помилок та підриває довіру користувачів. Питання захисту приватності даних, їх нецільового використання та визначення відповідальності за шкоду, завдану ШІ, вимагають розробки чітких нормативно-правових рамок та технічних стандартів [34].

Впровадження ШІ також стикається зі значними технічними обмеженнями та складнощами у підтримці ШІ-систем. Забезпечення необхідної продуктивності, низької затримки та масштабованості ШІ-моделей у реальному часі, особливо для веб-додатків з високим трафіком, є серйозним інженерним завданням. Інтеграція ШІ-компонентів з наявними, часто застарілими, архітектурами веб-додатків також може викликати значні труднощі.

ШІ-моделі не є статичними; вони потребують постійного моніторингу ефективності в робочому середовищі, оскільки з часом їхня якість може погіршуватися через «дрейф моделі» або «дрейф концепції». Це відбувається, коли характеристики вхідних даних чи зовнішнього середовища змінюються

порівняно з тими, на яких модель навчалася. Тому необхідні регулярні процедури оновлення даних, перенавчання моделей та їх повторного розгортання.

Проблема «холодного старту» – нездатність системи надавати якісні рекомендації для нових користувачів або щодо нових товарів/послуг через відсутність достатньої історії взаємодій – також є поширеним технічним викликом для багатьох ШІ-систем, особливо рекомендаційних.

Нарешті, успішна реалізація та підтримка ШІ-проектів вимагає наявності висококваліфікованих спеціалізованих кадрів. Ринок праці відчуває значний дефіцит досвідчених фахівців у галузі штучного інтелекту, машинного навчання, аналізу даних та інженерії даних. Це призводить до високої конкуренції за таланти, значних витрат на їх залучення, навчання та утримання, що може бути серйозним обмеженням, особливо для невеликих компаній та стартапів. Потрібні не просто окремі спеціалісти, а злагоджені міждисциплінарні команди, здатні ефективно взаємодіяти на всіх етапах життєвого циклу ШІ-продукту [35].

3.2 Проектування та розробка прототипу веб-додатку з ШІ

Після теоретичного обґрунтування переваг та обмежень впровадження штучного інтелекту, наступним логічним кроком у життєвому циклі розробки веб-додатку з елементами ШІ є створення прототипу. Прототипування відіграє критичну роль, особливо в інноваційних проектах, де присутні нові технології або неперевірені гіпотези. Воно дозволяє перевірити життєздатність концепції, візуалізувати майбутній продукт, зібрати ранній зворотний зв'язок від зацікавлених сторін (користувачів, інвесторів, команди розробки), виявити потенційні технічні труднощі та суттєво знизити ризики перед початком повномасштабної розробки. У контексті ШІ-інтеграції, прототип допомагає не лише оцінити користувацький досвід взаємодії з інтелектуальними функціями,

але й перевірити технічну реалізованість обраних ШІ-підходів, доступність та якість даних, а також попередньо оцінити продуктивність моделей. Процес створення прототипу є ітеративним і зазвичай включає декілька ключових етапів.

На початковому етапі закладається фундамент для майбутнього прототипу. Головне завдання – чітко визначити, що саме буде реалізовано та з якою метою.

Першим кроком є розробка концептуальної моделі, що формалізує ідею веб-додатку та його ключову функціональність. Важливо чітко визначити проблему користувача, яку вирішує додаток, і як інтегрований ШІ-компонент сприятиме цьому (наприклад, персоналізовані рекомендації для електронної комерції, адаптивний підбір матеріалів для освітньої платформи, інтелектуальний чат-бот для підтримки). Концептуальна модель має описувати очікувані вхідні дані для ШІ, етапи їх обробки та бажані вихідні результати, а також ключові сценарії взаємодії користувача з ШІ-функцією.

Після цього необхідно визначити цілі та обсяг прототипу. Важливо перевірити ключові гіпотези: чи технічно реалізовуваний обраний ШІ-підхід? Чи буде ШІ-функція зрозумілою та корисною для користувачів? Якою є попередня оцінка продуктивності ШІ-моделі? Рівень деталізації прототипу (низько- чи високодеталізований) залежить від цих цілей. Для ШІ-проектів часто доцільно створювати «Minimum Viable AI Prototype» (MVAP) – мінімально життєздатний прототип, що фокусується на реалізації однієї-двох ключових ШІ-функцій та найнеобхідніших підтримуючих елементів для демонстрації та тестування інтелектуальної можливості.

На наступному етапі візуалізується майбутній прототип та продумуються деталі взаємодії користувача з ним, особливо з інтегрованими ШІ-функціями.

Процес починається зі створення вайрфреймів. Вайрфрейми – це низькодеталізовані схематичні зображення екранів веб-додатку, що показують розташування основних функціональних блоків, елементів навігації, форм

введення та зон відображення інформації. Особлива увага приділяється тим екранам, де буде відбуватися взаємодія з ШІ: як користувач активує ШІ-функцію, як вводяться дані для неї, де і в якому форматі будуть представлені результати роботи інтелекту. Наприклад, для системи рекомендацій це може бути блок «Вам також може сподобатися» на сторінці товару; для чат-бота – інтерфейс діалогового вікна (рис. 3.2).

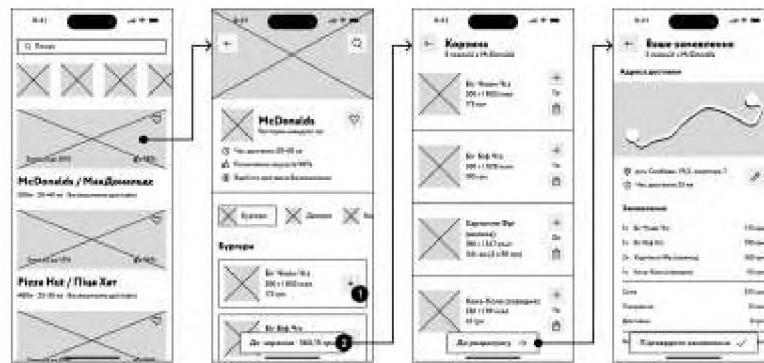


Рисунок 3.2 – Приклади вайрфреймів ключових екранів прототипу веб-додатку, що демонструють інтеграцію ШІ-функціоналу

Після узгодження вайрфреймів розробляються макети – більш деталізовані візуальні представлення інтерфейсу, що включають колірну схему, типографіку, іконки та реалістичне відображення контенту. Макети дають змогу отримати уявлення про кінцевий вигляд продукту. Важливо продумати, як візуально представити інформацію, згенеровану ШІ, щоб вона була зрозумілою, не перевантажувала користувача та викликала довіру. Наприклад, якщо ШІ робить прогнози, можна візуалізувати рівень впевненості моделі. Якщо ШІ надає персоналізовані пропозиції, важливо пояснити чому саме ці пропозиції були зроблені [36].

Паралельно з візуальним дизайном здійснюється проектування потоків взаємодії. Це детальні сценарії, що описують послідовність дій користувача та реакцій системи при виконанні певних завдань, особливо тих, що включають ШІ. Наприклад, як користувач запитує інформацію у чат-бота, як система обробляє запит, які уточнюючі питання може ставити бот, як виглядає

відповідь, і які подальші дії може запропонувати система. Важливо також передбачити обробку помилок та нетипових ситуацій: що відбудеться, якщо ШІ не зможе дати відповідь, надасть нерелевантний результат або якщо користувач введе некоректні дані.

Етап безпосереднього «кодування» прототипу, втілення проектних рішень у працюючий програмний продукт.

Спочатку відбувається вибір технологічного стеку для прототипу. Оскільки швидкість розробки та можливість швидких ітерацій є пріоритетом для прототипування, часто обирають легковагові та гнучкі технології. Для фронтенду це можуть бути сучасні JavaScript-фреймворки або навіть простіші рішення на базі HTML/CSS та нативного JavaScript, якщо не потрібна складна інтерактивність. Для бекенду часто використовуються Python або Node.js через їхню швидкість розробки API. Для зберігання даних у прототипі можуть застосовуватися прості бази даних, такі як SQLite, або хмарні NoSQL-рішення, якщо це відповідає природі даних ШІ.

Далі розробляється базова функціональність веб-додатку – ті мінімально необхідні не-ШІ компоненти, які потрібні для того, щоб продемонструвати роботу інтегрованої інтелектуальної функції. Це може включати реєстрацію/автентифікацію користувачів, можливість введення даних, базову навігацію тощо.

Центральним і найскладнішим завданням на цьому етапі є інтеграція штучного інтелекту. Існує кілька підходів до реалізації ШІ-функціоналу в прототипі:

Використання готових ШІ-сервісів/API: Багато хмарних провайдерів та спеціалізованих компаній пропонують готові до використання API для розв'язання типових завдань ШІ: розпізнавання мови, обробка природної мови, комп'ютерний зір, машинний переклад, надання рекомендацій. Переваги цього підходу – швидкість розробки, доступ до потужних, попередньо навчених моделей, відсутність необхідності у власній глибокій експертизі з машинного навчання на етапі прототипування. Недоліки – обмежені можливості кастомізації, потенційні витрати на використання API, залежність

від стороннього провайдера та можливі питання щодо конфіденційності даних [37].

Розробка спрощеної або імітованої ШІ-моделі: Для швидкої перевірки UX-аспектів взаємодії з ШІ, не вдаючись у складну розробку самої моделі, можна використати імітацію. Це може бути так званий підхід «Чарівник країни Оз», де «інтелектуальні» відповіді системи насправді генеруються людиною-оператором за лаштунками. Або ж це може бути проста система, заснована на наборі правил, яка імітує поведінку ШІ для обмеженого набору сценаріїв. Такий підхід дозволяє швидко отримати зворотний зв'язок щодо інтерфейсу та зручності взаємодії, але не дає змоги оцінити технічну реалізованість чи продуктивність справжньої ШІ-моделі.

Розробка та інтеграція власної базової ШІ-моделі: Якщо метою прототипу є також перевірка технічної гіпотези щодо можливості створення власного ШІ-рішення, то розробляється спрощена версія цільової моделі. Це включає збір невеликого, але репрезентативного набору даних, їх попередню обробку, вибір простого алгоритму машинного навчання, навчання моделі та її інтеграцію з бекендом прототипу. Переваги – отримання реального досвіду роботи з даними та моделями, можливість оцінити потенційні труднощі. Недоліки – це більш тривалий та ресурсомісткий процес порівняно з першими двома варіантами.

Незалежно від обраного варіанту реалізації ШІ, необхідно розробити API для взаємодії між основною частиною веб-додатку (бекендом) та ШІ-модулем (або зовнішнім сервісом). Цей API визначатиме формат запитів до ШІ, спосіб передачі вхідних даних та формат отримуваних результатів (рис. 3.3).

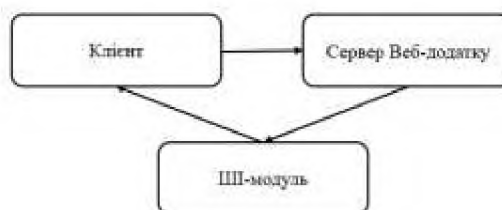


Рисунок 3.3 – Спрощена архітектура взаємодії компонентів у прототипі веб-додатку з ШІ

На завершення цього етапу відбувається реалізація відображення результатів роботи ШІ на фронтенді. Інтерфейс користувача має динамічно оновлюватися, представляючи інформацію, згенеровану або оброблену штучним інтелектом, у зрозумілій та зручній для користувача формі.

Створення першої версії прототипу – це лише початок циклу вдосконалення. Тестування є критично важливим для виявлення недоліків та збору інформації для подальших ітерацій.

Внутрішнє тестування проводиться командою розробки. Перевіряється функціональність всіх реалізованих елементів, коректність інтеграції ШІ-компонента, базова зручність використання та відсутність очевидних помилок [38].

Найважливішим є користувацьке тестування. Для цього залучається невелика група представників цільової аудиторії, яким пропонується виконати певні завдання за допомогою прототипу. Спостерігаючи за їхніми діями, фіксуючи їхні коментарі та труднощі, можна отримати безцінний зворотний зв'язок щодо зрозумілості інтерфейсу, корисності ШІ-функцій, рівня довіри до результатів роботи інтелекту та загального враження від взаємодії. Для тестування ШІ-функцій важливо зрозуміти, чи сприймають користувачі їх як «інтелектуальні», чи не викликають результати здивування або роздратування, чи легко інтерпретувати надану ШІ інформацію.

На основі аналізу результатів тестування приймаються рішення про необхідні зміни. Процес повертається на попередні етапи: коригується концептуальна модель, вдосконалюється UI/UX дизайн, доопрацьовується технічна реалізація. Прототип проходить через кілька таких ітерацій доти, доки не будуть досягнуті поставлені на початку цілі, або поки не стане зрозуміло, що початкова ідея потребує кардинального перегляду.

3.3 Техніко-економічне обґрунтування впровадження ШІ

Рішення про інтеграцію ШІ у веб-додаток – це не лише технічний виклик, а й стратегічний крок, який потребує глибокого економічного аналізу. За обмежених ресурсів та високої конкуренції, інвестиції у складні ШІ-технології мають бути ретельно обґрунтовані.

Техніко-економічне обґрунтування (ТЕО) є інструментом для такої оцінки, що дозволяє системно проаналізувати потенційні вигоди та переваги впровадження ШІ, зіставивши їх з очікуваними витратами, технічними складнощами та ризиками. Це допомагає зробити об'єктивний висновок щодо доцільності та економічної життєздатності проекту.

Особливість ТЕО для проектів у сфері ШІ полягає в необхідності врахування не лише прямих фінансових показників, але й стратегічних, часто нематеріальних, переваг, а також притаманного таким проектам високого ступеня невизначеності на етапах дослідження та розробки. Тому ТЕО для ШІ-проектів має бути гнучким, комплексним та орієнтованим на довгострокову перспективу.

Перш ніж перейти до економічних розрахунків, слід переконатися в технічній реалізованості запланованого ШІ-рішення. Цей аналіз базується на попередніх дослідженнях ринку, внутрішніх експериментах та висновках з тестування прототипу. Критичним аспектом є аналіз даних: їхня доступність, обсяг, якість, репрезентативність та релевантність для поставлених завдань. Необхідно оцінити, чи достатньо наявних даних для навчання ефективних ШІ-моделей, чи немає в них прихованих упереджень, що можуть призвести до небажаних наслідків, та скільки зусиль потребуватиме їх попередня обробка, очищення, трансформація та, можливо, ручна розмітка.

Далі, важливо оцінити придатність обраних алгоритмів та технологій ШІ. Чи відповідають вони специфіці завдання, чи є достатньо зрілими для промислового використання, чи існують надійні програмні бібліотеки, фреймворки або готові платформи для їхньої реалізації? Якою є складність цих

алгоритмів та їхня здатність до інтерпретації результатів, що може бути важливим для деяких застосувань [39].

Таблиця 3.1. Узагальнені компоненти витрат та вигод для ТЕО впровадження ШІ у веб-додаток

Категорія	Складові (Приклади)	Орієнтовна оцінка / Метод вимірювання	Характер впливу / Примітка
1. Витрати			
1.1. Витрати на розробку та впровадження	Персонал: заробітна плата Data Scientists, ML Engineers, веб-розробників, аналітиків, менеджерів.	Сума прямих витрат за проектом, грн.	Одноразові (капітальні) інвестиції на старті проекту.
1.2. Операційні витрати	Інфраструктура: хостинг моделей, оплата обчислювальних ресурсів для роботи ШІ (inference).	Сума періодичних витрат, грн/рік.	Постійні (поточні) витрати протягом життєвого циклу продукту.
2. Вигоди			
2.1. Кількісні (матеріальні) вигоди	Зниження витрат (S _{annual}). За рахунок автоматизації (напр., скорочення штату служби підтримки).	Розрахунок прямої економії коштів, грн/рік.	Прямий позитивний фінансовий ефект.
2.2. Якісні (нематеріальні) вигоди	Покращення користувацького досвіду (UX)..	Експертна оцінка, опитування (NPS, CSAT), аналіз ринкових позицій, аналіз непрямих показників.	Важко виразити в грошах, але мають значний довгостроковий стратегічний вплив на бізнес.

Інфраструктурні вимоги також потребують ретельного вивчення. Це стосується як обчислювальних ресурсів, так і систем зберігання великих обсягів даних. Необхідно визначити, чи буде використовуватися власна інфраструктура чи хмарні сервіси, і які витрати це потягне за собою.

Кадрове забезпечення є одним із ключових факторів успіху. Потрібно об'єктивно оцінити наявність у команді проекту або можливість залучення з ринку фахівців з необхідними компетенціями: Data Scientists, Machine Learning

Engineers, Data Engineers, спеціалістів з MLOps, а також веб-розробників, які мають досвід інтеграції ШІ-рішень.

Не менш важливою є оцінка потенційної складності інтеграції нових ШІ-компонентів з існуючою архітектурою веб-додатку, його базами даних, API та іншими залежними системами. Нарешті, необхідно заздалегідь продумати питання масштабованості та продуктивності розробленого ШІ-рішення, його здатності ефективно працювати при зростанні кількості користувачів, обсягів даних та запитів, а також можливості його подальшого розвитку та адаптації.

Ретельний та всебічний розрахунок усіх категорій витрат, пов'язаних із проектом впровадження ШІ, є фундаментом для подальшої оцінки його економічної привабливості. Витрати доцільно структурувати за двома основними напрямками: початкові інвестиції на розробку та впровадження, та подальші операційні витрати на підтримку та експлуатацію.

Витрати на розробку та впровадження ШІ-рішення — це початкові інвестиції, що охоплюють створення, тестування та запуск. Ця комплексна категорія включає значні витрати на висококваліфікований персонал: розробників ШІ та веб-розробників, аналітиків даних, тестувальників, менеджерів проєктів та консультантів. Обсяг цих витрат залежить від тривалості їх залучення на етапах дослідження, проектування, розробки та тестування. Не менш важливими є витрати, пов'язані з даними: їх придбання, збір, трудомістке очищення, трансформація, а також ручна розмітка, що критично важлива для навчання з учителем. Значних капіталовкладень також потребує формування та підтримка відповідної інфраструктури для розробки, такої як потужні сервери з графічними процесорами, спеціалізовані робочі станції або оренда хмарних обчислювальних ресурсів. До цієї ж категорії належать витрати на програмне забезпечення, включно з ліцензіями на комерційні інструменти аналізу даних, платформи для розробки ШІ, середовища для спільної роботи та системи управління версіями.

Після запуску ШІ-рішення, його безперебійна робота, підтримка та розвиток формують операційні витрати, що є періодичними, зазвичай річними. Вони охоплюють витрати на підтримку операційної інфраструктури, як-от

хостинг ШІ-моделей, оренда обчислювальних ресурсів для їхнього функціонування в реальному часі та зберігання даних. Значною статтею є витрати на моніторинг та технічне обслуговування моделей (MLOps), що включає відстеження їхньої продуктивності, виявлення деградації, регулярне оновлення даних для навчання, перенавчання та безпечно повторне розгортання. Також необхідно враховувати витрати на персонал підтримки, який відповідає за технічний супровід ШІ-системи, аналіз її роботи та вирішення проблем. Якщо ШІ-рішення використовує сторонні API, до операційних витрат додаються періодичні платежі, часто залежні від обсягу оброблених даних чи кількості запитів [41].

Оцінка потенційних вигод від впровадження ШІ є не менш критичним завданням, ніж калькуляція витрат, адже саме зіставлення вигод та витрат визначає доцільність проекту. Вигоди традиційно поділяються на ті, що піддаються прямому кількісному вимірюванню, та якісні, оцінка яких є складнішою, але їхній вплив на бізнес може бути не менш значущим.

Кількісні вигоди від ШІ вимірюються фінансово через зниження операційних витрат та збільшення доходів. Витрати зменшуються завдяки автоматизації рутинних завдань (чат-боти, модерація контенту, аналіз ризиків), оптимізації ресурсів (прогнозування попиту, енергоспоживання, маркетинг) та зменшенню витрат від шахрайства. Збільшення доходів досягається через персоналізовані рекомендації, залучення нових клієнтів унікальними ШІ-функціями, підвищення лояльності та LTV клієнтів, а також створення нових цифрових продуктів.

Окрім прямих фінансових вигод, існують якісні переваги, які, хоч і не вимірюються безпосередньо в грошах, суттєво впливають на довгостроковий успіх бізнесу.

Значне покращення загальної якості користувацького досвіду є однією з ключових якісних вигод. Інтуїтивно зрозумілий веб-додаток, що швидко надає релевантну інформацію, пропонує корисні персоналізовані функції та ефективно вирішує завдання користувача, неминуче підвищує його

задоволеність. Задоволені користувачі стають лояльнішими до бренду, частіше повертаються до веб-додатку та рекомендують його іншим, що в довгостроковій перспективі призводить до фінансових результатів.

Впровадження передових ІІТ-технологій формує позитивний імідж компанії як інноваційної, технологічно просунутої та клієнтоорієнтованої. Це важливий фактор для залучення не лише клієнтів, а й талановитих співробітників. Глибше розуміння ринкових тенденцій, поведінки споживачів та ефективності бізнес-процесів, отримане завдяки інтелектуальному аналізу даних, дозволяє керівництву прискорити та підвищити якість прийняття стратегічних управлінських рішень.

Зрештою, це підвищує загальну гнучкість та адаптивність бізнесу до швидкозмінного ринкового середовища. Також, це може позитивно вплинути на корпоративну культуру та умови праці співробітників, звільняючи їх від монотонної рутинної роботи та дозволяючи зосередитися на творчих та значущих завданнях.

Завершальним етапом ТЕО є розрахунок та аналіз інтегральних показників економічної ефективності, які дозволяють узагальнити інформацію про витрати та вигоди і дати кількісну оцінку інвестиційної привабливості проекту. Ці показники допомагають порівнювати різні інвестиційні альтернативи та приймати обґрунтовані рішення [42].

Одним із найбільш фундаментальних та широко використовуваних показників для оцінки довгострокових інвестиційних проектів є чиста приведена вартість. Цей показник враховує ключовий економічний принцип – зміну вартості грошей у часі. Сьогоднішня гривня коштує більше, ніж гривня, яку очікується отримати в майбутньому, через інфляцію, ризики та альтернативну вартість капіталу. NPV розраховується як сума всіх майбутніх чистих грошових потоків від проекту, дисконтованих (приведених) до їхньої сьогоднішньої вартості, мінус сума початкових інвестицій. Проект вважається економічно доцільним та таким, що створює вартість для компанії, якщо його

NPV є позитивним ($NPV > 0$). Формула (3.1) для розрахунку NPV має такий вигляд :

$$NPV = \sum_{t=1}^T \frac{CF_t}{(1+r)^t} - C_0, \quad (3.1)$$

де NPV (Net Present Value – чиста теперішня вартість) – різниця між теперішньою вартістю майбутніх грошових потоків і теперішньою вартістю початкових інвестиційних витрат.

t – індекс часу, який починається з 1 (перший період) і йде до T (останній період).

T – загальна кількість періодів часу (наприклад, років, кварталів, місяців), протягом яких очікуються грошові потоки від інвестиції.

CF_t (Cash Flow at time t – грошовий потік у час t) – очікуваний чистий грошовий потік (прибуток мінус витрати) у період t . Це може бути приплив або відтік коштів.

Окрім NPV, для оцінки економічної ефективності часто використовують й інші показники. Наприклад, рентабельність інвестицій (Return on Investment, ROI) показує загальну прибутковість проекту відносно зроблених інвестицій, виражену у відсотках. Вона допомагає зрозуміти, наскільки ефективно були використані інвестовані кошти. Термін окупності (Payback Period, PP) визначає час, необхідний для того, щоб кумулятивні чисті грошові потоки від проекту повністю покрили початкові інвестиції. Цей показник є важливим для оцінки ліквідності та ризиковості проекту, хоча у своїй простій формі він не враховує вартість грошей у часі. Внутрішня норма прибутковості (Internal Rate of Return, IRR) є ще одним дисконтним методом оцінки; це така ставка дисконтування, за якої NPV проекту дорівнює нулю. Проект вважається привабливим, якщо його IRR перевищує вартість капіталу або мінімально прийнятну норму прибутковості [43].

3.4 Оцінка ефективності та перспективи розвитку

Після проектування, розробки та попереднього техніко-економічного обґрунтування ШІ у веб-додатку, настає критичний етап оцінки ефективності та визначення подальших перспектив. Це служить мостом між поточними результатами та майбутніми стратегічними рішеннями, дозволяючи зробити обґрунтовані висновки щодо успішності інтеграції ШІ, її реального впливу на ключові показники та доцільності подальших інвестицій. Комплексна оцінка та чітке бачення перспектив є запорукою того, що веб-додаток з ШІ буде цінним для користувачів та ефективним для бізнесу.

Оцінка ефективності впровадженого ШІ-рішення має бути багатогранною, базуючись на заздалегідь визначеній методології, що охоплює технічні та бізнесові аспекти, а також безпосередній вплив на користувацький досвід. Методологія ґрунтується на ключових показниках ефективності (KPIs), що відображають цілі проекту та очікувані вигоди. Ці показники включають метрики для оцінки ШІ-моделей (наприклад, точність для класифікації) та для веб-додатку в цілому (зміни у коефіцієнтах конверсії, тривалості сесії, показниках відмов, скороченні звернень до підтримки) [44].

Якісні метрики спрямовані на оцінку суб'єктивного сприйняття користувачами нововведень. Це може включати вимірювання задоволеності, зручності використання, легкості взаємодії з ШІ-функціями, а також збір та аналіз прямих відгуків.

Важливим етапом є аналіз результатів тестування, проведеного під час або після завершення розробки прототипу чи мінімально життєздатного продукту (MVP). Необхідно систематизувати та інтерпретувати дані з функціонального тестування, тестування продуктивності ШІ-моделей, юзабіліті-тестування та A/B тестування. Порівняння фактичних результатів з контролями дозволяє об'єктивно оцінити прогрес та виявити сильні й слабкі сторони рішення.

Особливу увагу приділіть оцінці впливу ШІ на користувацький досвід (UX). Важливо глибоко проаналізувати, як інтеграція інтелектуальних функцій змінила поведінку користувачів у веб-додатку. Чи стало їм легше та швидше досягати своїх цілей? Чи сприймаються ШІ-поради та автоматизовані дії як корисні та релевантні? Необхідно виявити, чи не викликає ШІ у користувачів спантеличення, недовіри до результатів або занепокоєння щодо приватності даних. Відповіді на ці питання допоможе отримати аналіз відгуків, опитувань та юзабіліті-тестування, сфокусованих саме на ШІ-компонентах, що також виявить негативні аспекти UX, які потребують коригування.

Доцільно також провести попередню або вже фактичну оцінку економічної ефективності впровадженого ШІ-рішення. Це передбачає зіставлення реальних витрат на розробку та впровадження з оновленими прогнозами щодо потенційних вигод. Такий аналіз дозволяє перевірити реалістичність початкових припущень у техніко-економічному обґрунтуванні та оцінити, чи перебуває проєкт на шляху до досягнення очікуваної економічної віддачі.

Оцінка ефективності є не лише підбиттям підсумків, але й відправною точкою для планування майбутнього. Успішна інтеграція ШІ – це, як правило, не одноразовий проєкт, а безперервний процес вдосконалення, адаптації та розширення.

Ключовим напрямком розвитку є постійне вдосконалення існуючих ШІ-моделей. Завдяки технологіям машинного навчання, моделі можна регулярно перенавчати на нових даних про взаємодію користувачів з додатком, що підвищує їхню точність та релевантність. Практики MLOps допомагають автоматизувати цей процес. Важливу роль відіграє також збір зворотного зв'язку від користувачів для тонкого налаштування алгоритмів. Для досягнення вищої якості може знадобитися впровадження більш досконалих алгоритмів та архітектур.

Наступним логічним кроком є розширення застосування ШІ в межах веб-додатку. На основі аналізу потреб користувачів та бізнес-цілей можна

ідентифікувати нові сфери або функції, де інтеграція інтелекту принесе додаткову цінність. Це може бути впровадження ШІ для аналізу відгуків, автоматизації нових завдань, розширення можливостей персоналізації або додавання нових інтелектуальних інструментів. Важливо здійснювати таке розширення поступово, ретельно тестуючи кожен нову ШІ-функцію.

Масштабування ШІ-рішення є ще одним важливим аспектом довгострокового розвитку. Зі зростанням кількості користувачів, обсягів даних та інтенсивності запитів до ШІ-сервісів необхідно забезпечити їхню технічну масштабованість. Це може вимагати оптимізації коду моделей, переходу на потужнішу інфраструктуру або застосування мікросервісної архітектури для окремих ШІ-компонентів. Окрім технічного, існує й організаційне масштабування, що передбачає розбудову команди спеціалістів з ШІ та впровадження ефективних процесів управління ШІ-проектами.

Перспективним напрямком є дослідження та впровадження новітніх технологій ШІ. Сфера штучного інтелекту розвивається надзвичайно стрімко. Для підтримки конкурентоспроможності веб-додатку важливо відстежувати ці інновації. Зокрема, перспективи глибокого навчання дуже широкі: у майбутньому можна розглянути застосування більш складних глибоких нейронних мереж, таких як трансформери для обробки природної мови, генеративно-змагальні мережі для генерації контенту, або сучасні архітектури згорткових нейронних мереж для комп'ютерного зору. Також варто звернути увагу на такі напрямки, як пояснюваний ШІ (XAI) для підвищення прозорості та довіри; федеративне навчання, що уможливорює навчання моделей на децентралізованих даних без порушення конфіденційності; або навчання з підкріпленням для створення систем, здатних динамічно оптимізувати свою поведінку [45].

ВИСНОВКИ

У кваліфікаційній роботі досліджено розробку та обґрунтування підходів до інтеграції штучного інтелекту (ШІ) у сучасні веб-додатки для покращення взаємодії користувачів. Підтверджено, що впровадження ШІ є фундаментальним зрушенням у веб-розробці, перетворюючи статичні ресурси на динамічні, персоналізовані та проактивні цифрові середовища.

Теоретичний аналіз показав, що ця трансформація полягає у здатності систем машинного та глибокого навчання аналізувати великі обсяги даних і виявляти приховані закономірності, що відкриває широкі можливості для збагачення користувацького досвіду (UX). Встановлено, що ключовими напрямками застосування ШІ для покращення UX є персоналізація контенту та рекомендацій, впровадження інтелектуальних систем взаємодії (чат-боти), а також оптимізація інтерфейсів. Робота підкреслює, що інструментальним базисом для цього є різноманітні методи машинного навчання – від класифікації та регресії до кластеризації та навчання з підкріпленням.

У ході дослідження розроблено та обґрунтовано комплексний методологічний підхід до створення веб-додатків з елементами ШІ, що охоплює весь життєвий цикл продукту. Наголошено на важливості розмежування веб-сайтів та веб-додатків, оскільки останні є основною платформою для інтеграції ШІ. Запропонована послідовність етапів розробки (від планування до підтримки) враховує специфіку ШІ-проектів, зокрема необхідність ретельного попереднього аналізу даних, оцінки доцільності ШІ, специфічного проектування UX для інтелектуальних функцій, а також ітеративного характеру розробки та валідації ШІ-моделей.

Практичні аспекти реалізації розглянуто через призму аналізу переваг та викликів. Підтверджено, що попри значні переваги (автоматизація, покращена аналітика, інноваційні можливості), впровадження ШІ пов'язане з суттєвими обмеженнями. До них належать високі початкові витрати, критична залежність від якості даних, а також складні технічні та етичні питання (прозорість, упередженість, відповідальність). Робота підкреслює вирішальну роль прототипування для мінімізації ризиків та тестування концепцій.

Важливою складовою дослідження стало обґрунтування необхідності ретельної оцінки ефективності впровадження ШІ, що виходить за межі суто технічних метрик. Показано, що для виважених бізнес-рішень потрібен комплексний підхід, включаючи техніко-економічне обґрунтування. Аналіз проектів за допомогою фінансових інструментів, як-от чиста приведена вартість (NPV), дозволяє оцінити довгострокову інвестиційну привабливість ШІ, зіставляючи витрати з прогнозованими матеріальними та нематеріальними вигодами. Окрім економічної оцінки, наголошено на важливості постійного моніторингу ефективності ШІ-рішення після його запуску, що включає аналіз технічних показників моделей та вимірювання його реального впливу на ключові показники UX та бізнесу.

Таким чином, у кваліфікаційній роботі досягнуто поставленої мети та запропоновано цілісний погляд на процес інтеграції штучного інтелекту у веб-додатки. Практична значущість роботи полягає в тому, що її результати можуть слугувати методологічною базою та практичним посібником для розробників, архітекторів програмного забезпечення, менеджерів продуктів та керівників бізнесу. Систематизований опис етапів, аналіз потенційних переваг та ризиків, а також підходи до оцінки ефективності допоможуть приймати обґрунтованіші рішення, уникати поширених помилок та підвищувати шанси на успішну реалізацію складних, але перспективних ШІ-проектів.

Водночас, визнаються обмеження даного дослідження, яке носить переважно теоретико-методологічний характер. Напрямами для подальших наукових розвідок може стати емпірична перевірка та кількісна оцінка ефективності запропонованих підходів на прикладі розробки конкретного веб-додатку з інноваційними ШІ-функціями. Перспективним є також поглиблене дослідження можливостей застосування новітніх досягнень у сфері глибокого навчання, зокрема трансформерних архітектур та генеративних моделей, для вирішення специфічних завдань у веб-середовищі, а також розробка деталізованих методик для підвищення прозорості та інтерпретованості ШІ-рішень (Explainable AI, XAI) з метою зміцнення довіри користувачів.