

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: « **РОЗРОБКА ВЕБ-ЗАСТОСУНКУ ДЛЯ
ПЕРСОНАЛІЗОВАНОГО СТИЛІСТИЧНОГО СУПРОВОДУ З
ВИКОРИСТАННЯМ ШІ**»

Виконала: студентка 4-го року

навчання,

Освітньої програми «Інженерія
програмного забезпечення», 121

Медовщук Валерія Віталіївна

Керівник: Борозенний С. О.

Кваліфікаційна робота

захищена з оцінкою

Рецензент

(прізвище та ініціали)

Секретар ЕК

« ____ » _____ 2026р.

Київ – 2026

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

_____ (підпис)
„_____” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на дипломну роботу

студенту 4-го курсу, факультету інформатики

Медовщук Валерії Віталіївні

ТЕМА: Розробка веб-застосунку для персоналізованого стилістичного супроводу з використанням ШІ

Зміст ТЧ до дипломної роботи:

Анотація

Вступ

Розділ 1. Аналіз існуючих рішень та обґрунтування необхідності розробки системи

Розділ 2. Архітектурне проєктування системи

Розділ 3. Опис розробленої системи

Висновки

Джерела

Дата видачі „_____” _____ 2026 р. Керівник _____ (підпис)

Завдання отримав _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

№	Назва етапу	Дати виконання
1	Обрання теми на кваліфікаційну роботу	09.10.2025
2	Дослідження необхідних джерел, ресурсів для глибшого розуміння теми	15.10.2025 – 15.11.2025
3	Вивчення вже існуючих аналогів веб-платформи	17.11.2025 – 18.12.2025
4	Створення дизайну веб-платформи	19.12.2025 – 03.01.2026
5	Розробка веб-платформи	04.12.2025 – 05.05.2026
6	Створення текстової частини	10.04.2026 – 15.05.2026
7	Передзахист кваліфікаційної роботи	30.03.2026– 01.04.2026
8	Захист кваліфікаційної роботи	25.05.2026 – 27.05.2026

Студентка Медовщук Валерія Віталіївна

Керівник: Борозенний С. О.

«___» _____ 2026р.

ЗМІСТ

АНОТАЦІЯ	5
ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ СИСТЕМИ	9
1.1 Приклади реалізації онлайн-платформ та додатків для стилістичного супроводу та моди	9
1.2 Технології та підходи до створення персоналізованих в галузі fashion-tech	15
1.3 Формування вимог до системи персоналізованого стилістичного супроводу на основі аналізу існуючих сервісів	18
Висновки до розділу 1	21
РОЗДІЛ 2. АРХІТЕКТУРНЕ ПРОЄКТУВАННЯ СИСТЕМИ	22
2.1 Розробка архітектурної моделі веб-застосунку персоналізованого стилістичного супроводу	22
2.2 Вибір технологій та інструментів розробки	25
2.3 Розробка підходу інтеграції штучного інтелекту в систему для аналізу вподобань, генерації образів та рекомендацій	29
2.4 Архітектурні діаграми реалізованої системи	32
2.4.1 Діаграма зв'язків сутностей	32
2.4.2 Use-case diagram	35
Висновки до розділу 2	37
РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ	38
3.1 Особливості реалізації: архітектура клієнт-серверної взаємодії, структура бази даних, інтеграція AI-сервісів/моделей	38
3.2 Користувачський функціонал веб-платформи	42
Висновки до розділу 3	53
ВИСНОВКИ	54
ДЖЕРЕЛА	56

АНОТАЦІЯ

Робота направлена на формування архітектурної моделі для веб-застосунку персоналізованого стилістичного супроводу та розробку програмного продукту, який забезпечує аналіз зовнішності користувача, підбір одягу та віртуальну примірку з використанням ШІ.

У ході роботи були дослідженні існуючі аналоги fashion-платформ, для кращого розуміння недоліків і потреб користувачів та формування основних функціональних вимог для майбутнього застосунку.

Поставлене завдання було виконано з використанням Flask, Vue.js, SQLAlchemy, Google Gemini API, CatVTON та інших бібліотек для досягнення найкращої результативності. Розроблено модель AI-аналізу зовнішності, яка використовує механізми комп'ютерного зору для визначення кольоротипу та фігури.

Виконаний функціонал дозволяє користувачам зручно та ефективно взаємодіяти з платформою, отримувати персоналізовані рекомендації від ШІ, завантажувати фото, генерувати комплекти одягу та візуалізувати примірки.

ВСТУП

Персоналізований стилістичний супровід з використанням штучного інтелекту є як ніколи актуальною темою, оскільки все більше людей переходить на онлайн-шопінг та віртуальні консультації, що заощаджує час, кошти та робить моду доступнішою. Наприклад, згідно з дослідженням McKinsey & Company, впровадження AI в fashion-технологіях підвищує лояльність клієнтів на 25% завдяки точним рекомендаціям та зменшенню повернень товарів [1]. Крім того, у деяких країнах, таких як Україна, де економічні та соціальні фактори обмежують доступ до професійних стилістів, онлайн-платформи стають ключовим інструментом для самовираження та стилізації.

Зараз на ринку існує велика кількість стилістичних веб-платформ, проте користувачі часто стикаються з труднощами у їх використанні через обмеженість комерційними каталогами, недостатній рівень персоналізації з використанням новітніх AI-технологій, надмірну залежність від людського фактора, проблеми з реалістичністю віртуальної примірки, етичністю даних та інші.

Саме тому актуальним є створення простої та інтуїтивно зрозумілої веб-платформи для стилістичного супроводу, яка буде зручною як для жінок, так і для чоловіків і задовольнятиме всі їхні потреби. Дана платформа дозволить користувачам легко завантажувати фото для аналізу кольоротипу та типу фігури, отримувати персоналізовані рекомендації щодо комплектів одягу, а також візуалізувати віртуальну примірку з використанням AI. У той самий час, платформа надасть можливість переглядати каталог, зберігати історію образів, оновлювати профіль та отримувати стилістичні поради, забезпечуючи повний цикл від аналізу до візуалізації.

Метою даного дослідження є проєктування та формування архітектури веб-застосунку для персоналізованого стилістичного супроводу з

використанням ШІ, а також розробка програмного продукту, який забезпечить модульність, масштабованість, впровадження функцій на базі технологій штучного інтелекту, можливість подальшого функціонального розширення, передбачається аналіз сучасних технологічних підходів до побудови подібних систем, аналіз присутніх на ринку аналогів та визначення основних вимог до стилістичних онлайн-платформ, вибір оптимального стеку технологій та розробка програмного забезпечення і створення доступної веб-платформи якісного та ефективного супроводу для широкого кола користувачів, зокрема у регіонах з відсутністю можливості відвідування офлайн-консультацій.

Визначено три основні розділи роботи:

У першому розділі було досліджено платформи для стилістичного супроводу та різноманітні інструменти для розробки, визначено критичні аспекти і сформовано функціональні вимоги до майбутнього рішення.

У другому розділі висвітлюється спроектована архітектурна модель веб-платформи з усіма необхідними компонентами, а також методи та інструменти розробки, що дозволили розробити підхід інтеграції AI-моделей в систему для забезпечення персоналізації рекомендацій та віртуальної примірки, структурно-модульний підхід до проектування системи, сучасні методи фронтенд-розробки, зокрема такі інструменти, як Vue.js та JavaScript, а також Flask для бекенду та SQLAlchemy для даних, що дозволили створити зручну, функціональну та ефективну веб-платформу для стилістичного супроводу, яка задовольняє вимоги користувачів, легко масштабується під потреби зростаючої аудиторії, має інтуїтивно-зрозумілий інтерфейс для користувачів навіть без стилістичної підготовки та забезпечує зручний доступ до рекомендацій, інтерактивну взаємодію з AI та сприяє поширенню цифрової моди для всіх охочих.

У третьому розділі описується технічні можливості та інтерфейс реалізованої конкурентоспроможної веб-платформи для стилістичного супроводу для користувачів, яка забезпечує спрощений механізм аналізу

зовнішності, взаємодію з AI для підбору комплектів, зручний перегляд каталогу та візуалізацію примірки, збереження історії та легку взаємодію.

Під час виконання роботи, було проведено аналіз літературних джерел, статей, досліджень, а також сучасних практик у сфері веб-розробки, fashion-технологій та дизайну інтерфейсу, щоб сформувавши повне бачення веб-платформи. Вивчені матеріали дозволили виділити головні засади розробки доступних, масштабованих онлайн-платформ.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ НЕОБХІДНОСТІ РОЗРОБКИ СИСТЕМИ

1.1 Приклади реалізації онлайн-платформ та додатків для стилістичного супроводу та моди

У сучасному цифровому середовищі онлайн-платформи та мобільні додатки для стилістичного супроводу та моди відіграють ключову роль у трансформації традиційного шопінгу, пропонуючи користувачам персоналізовані рекомендації на основі аналізу їхніх уподобань, фізичних характеристик та стилістичних потреб. Ці рішення інтегрують елементи штучного інтелекту, машинного навчання та комп'ютерного зору, дозволяючи не лише полегшити вибір одягу, але й створити віртуальний досвід, наближений до консультації з професійним стилістом. Розглядаючи приклади таких реалізацій, можна помітити еволюцію від простих каталогів до комплексних систем, які враховують культурні, сезонні та індивідуальні фактори, роблячи моду доступнішою та інклюзивнішою.

Одним з яскравих прикладів є платформа Stitch Fix (рис. 1.1), яка функціонує як сервіс підписки на персоналізовані бокси з одягом. Користувачі заповнюють детальний профіль, вказуючи свої розміри, стильові вподобання, бюджет та спосіб життя, після чого алгоритми платформи, засновані на машинному навчанні, аналізують ці дані для підбору речей. Стилiсти-люди переглядають рекомендації ШІ, додаючи людський дотик, що робить процес більш точним. Платформа використовує дані з попередніх покупок для вдосконалення алгоритмів, дозволяючи користувачам повертати невідповідні речі безкоштовно. Це рішення особливо популярне серед зайнятих професіоналів, оскільки воно економить час на шопінг, пропонуючи готові комплекти, адаптовані до повсякденного чи святкового вбрання. Stitch Fix інтегрує елементи гейміфікації, такі як опитування після отримання боксу,

що допомагає ШІ навчатися на зворотному зв'язку, підвищуючи точність рекомендацій до 80% у повторних відправках [1].



Рисунок 1.1 Інтерфейс платформи Stitch Fix

Іншим помітним прикладом є Amazon Personal Shopper (рис. 1.2), інтегрований у екосистему Amazon Prime, де штучний інтелект аналізує історію покупок, перегляди товарів та відгуки для створення персоналізованих стилістичних пропозицій. Користувачі можуть вибрати теми, такі як "офісний стиль" чи "спортивний casual", а система генерує комплекти одягу, взуття та аксесуарів. Ця платформа вирізняється використанням комп'ютерного зору для аналізу зображень товарів, дозволяючи віртуальну примірку через додаток з доповненою реальністю, де користувач бачить, як одяг виглядатиме на його фігурі. Amazon Personal Shopper також враховує сезонні тренди та геолокацію, пропонуючи одяг, адаптований до клімату регіону. Завдяки інтеграції з голосовим асистентом Alexa, користувачі можуть отримувати рекомендації голосом, що додає зручності для людей з обмеженими можливостями. Дослідження показують, що такі системи підвищують лояльність клієнтів на 25%, оскільки зменшують кількість повернень завдяки точним рекомендаціям [2].

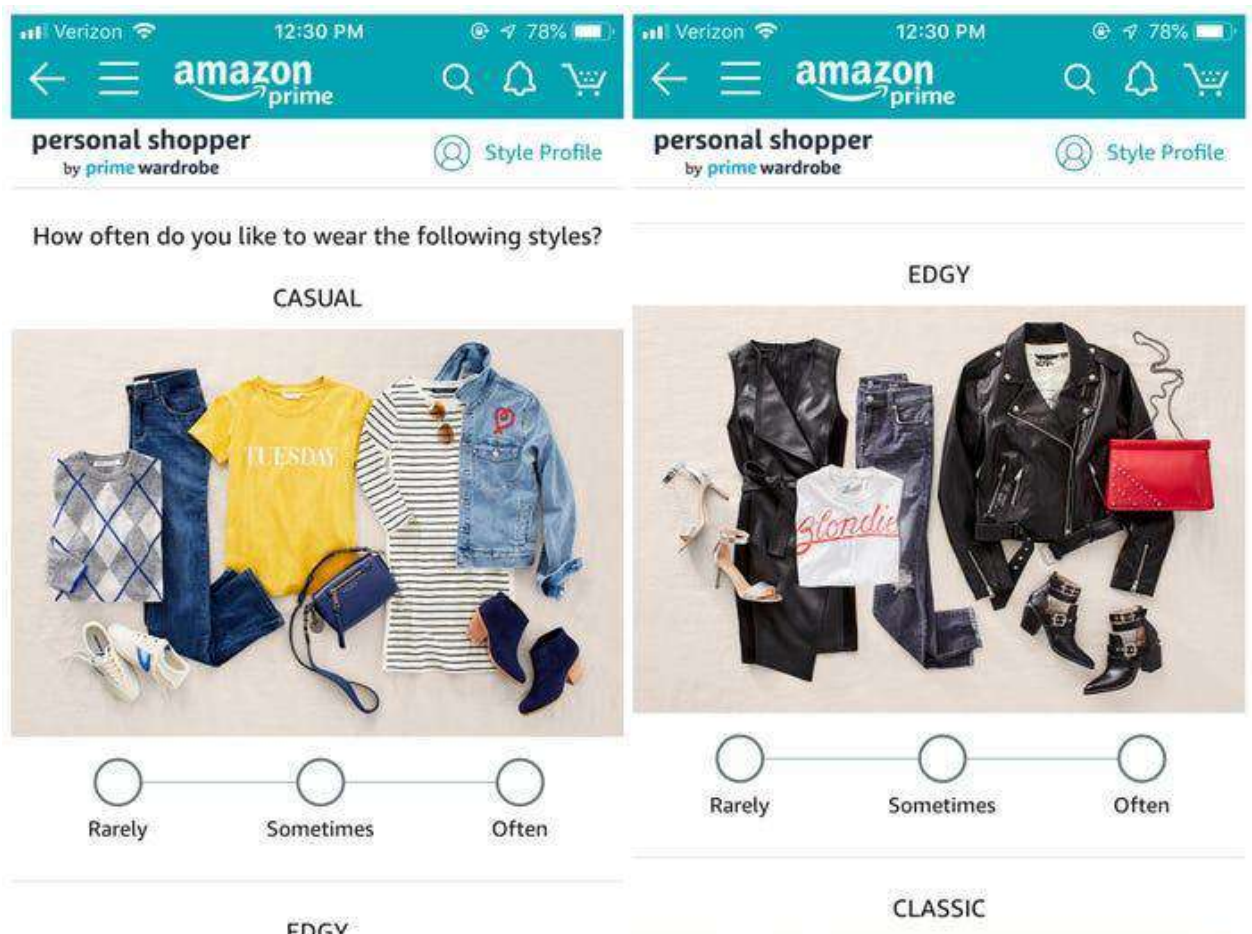


Рисунок 1.2 Інтерфейс Amazon Personal Shopper

Платформа ASOS Style Match (рис. 1.3) представляє собою мобільний додаток, орієнтований на візуальний пошук, де користувачі завантажують фото бажаного вбрання або сканують зображення з журналів, а ШІ знаходить подібні товари в каталозі ASOS. Алгоритми комп'ютерного зору розпізнають кольори, текстури та фасони, пропонуючи альтернативи з урахуванням розміру та бюджету. Ця система інтегрує соціальні елементи, дозволяючи ділитися образами в мережах, а також надає стилістичні поради від експертів. ASOS Style Match особливо ефективний для молодшої аудиторії, яка шукає трендові речі, оскільки платформа оновлює рекомендації в реальному часі на основі глобальних модних тенденцій. Додаток також включає функцію "стилістичний чат-бот", який відповідає на запитання про поєднання речей, роблячи процес супроводу інтерактивним і освітнім [3].

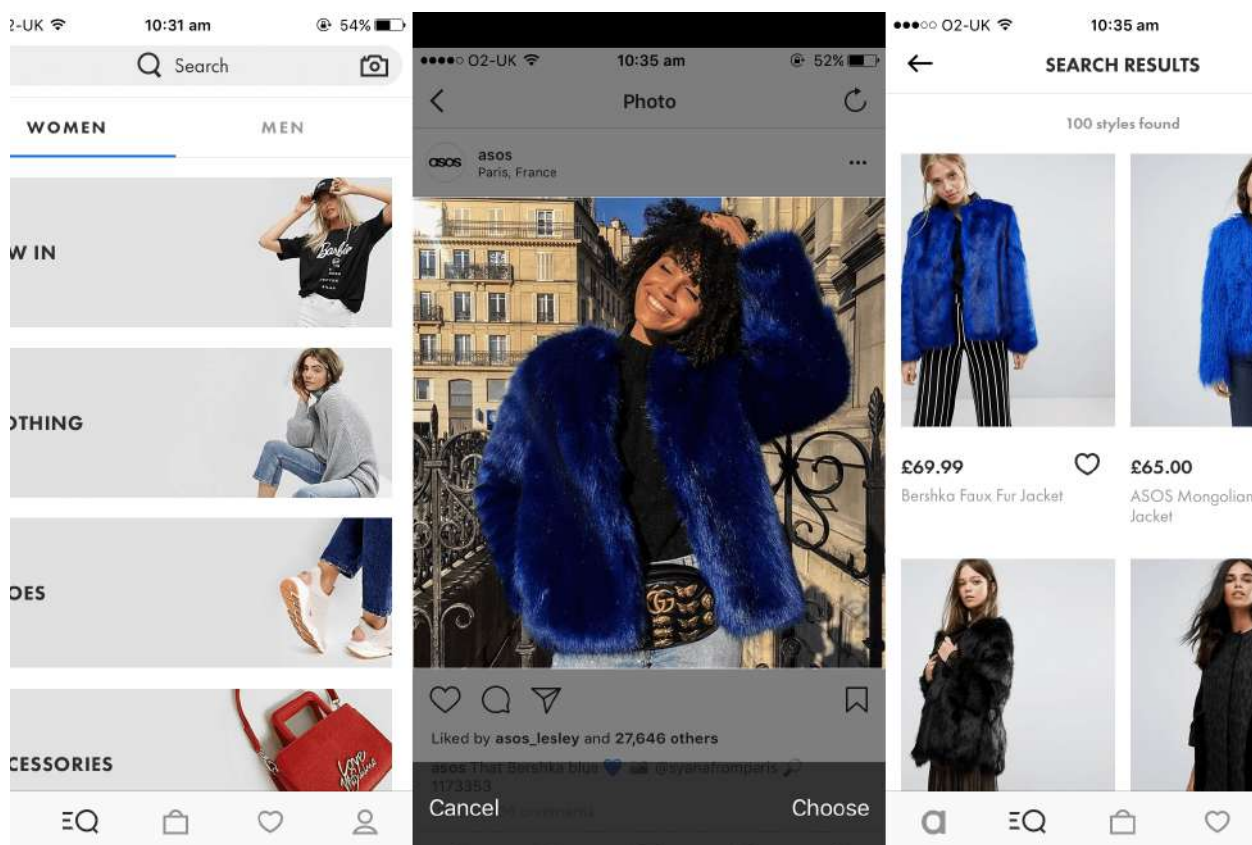


Рисунок 1.3 Інтерфейс ASOS Style Match

Ще одним варіантом є Lookiero (рис. 1.4), європейська платформа, подібна до Stitch Fix, але з акцентом на жіночу аудиторію. Користувачі проходять стилістичний квіз, де вказують свої параметри фігури, колірні вподобання та події, для яких потрібен одяг. ШІ аналізує ці дані, поєднуючи їх з роботою персональних стилістів, які формують бокси з п'ятьма речами. Платформа використовує алгоритми для прогнозування трендів, базуючись на даних з fashion-шоу та соціальних мереж, що дозволяє пропонувати актуальні комплекти. Lookiero вирізняється фокусом на інклюзивності, пропонуючи розміри від XS до 5XL, і включає елементи психологічного профілювання, щоб рекомендації відповідали емоційному стану користувача, наприклад, пропонуючи яскраві кольори для підняття настрою [4].



Рисунок 1.4 Інтерфейс платформи Lookiero

Нарешті, платформа Stylebook (рис. 1.5) є мобільним додатком для самостійного стилістичного супроводу, де користувачі створюють віртуальний гардероб, завантажуючи фото речей, а ШІ генерує комбінації на основі кольорів, сезонів та подій. Додаток використовує алгоритми для календаря образів, нагадуючи про комбінації, і інтегрує погоду для адаптації рекомендацій. Stylebook вирізняється освітнім аспектом, пропонуючи поради щодо догляду за одягом та трендів, роблячи його інструментом для самоосвіти в моді [5].

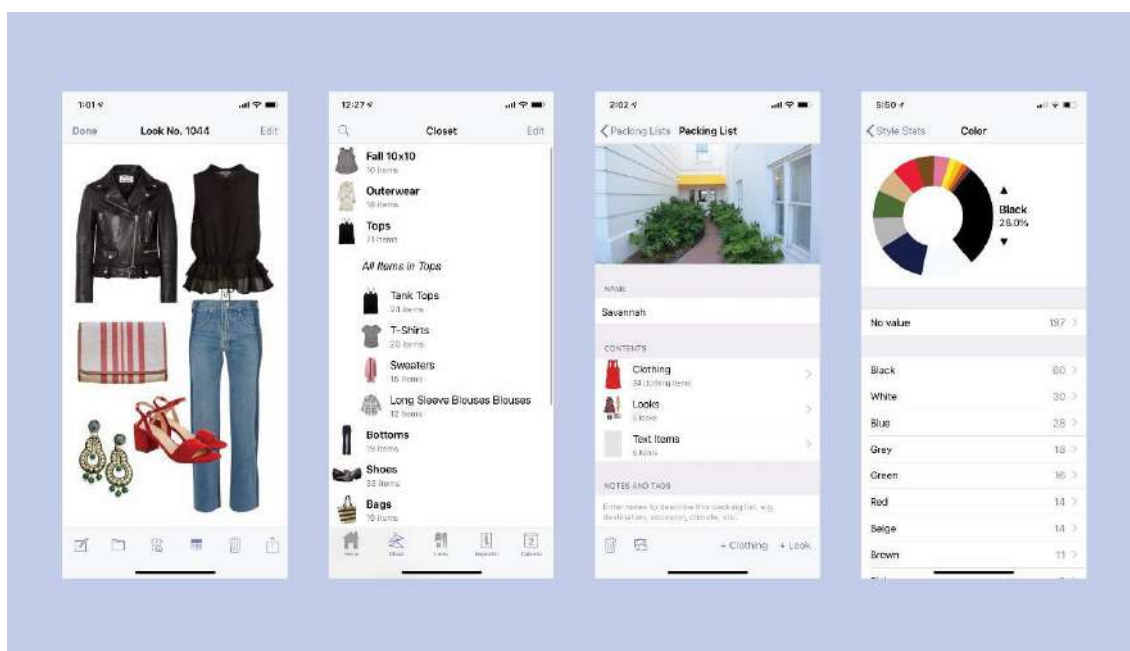


Рисунок 1.5 Інтерфейс Stylebook

Порівнюючи ці платформи, можна побачити спільні тенденції: інтеграцію ІІІ для персоналізації, використання візуального пошуку та фокус на зручності. Вони демонструють, як технології роблять стилістичний супровід масовим, але водночас підкреслюють необхідність подальшого розвитку для глибшої інтеграції з реальним життям користувачів.

Таблиця 1.1

Порівняння функціональності основних платформ стилістичного супроводу

Платформа	Основні функції	Технології ІІІ	Цільова аудиторія
Stitch Fix	Персоналізовані бокси, зворотний зв'язок	Машинне навчання, рекомендаційні системи	Зайняті професіонали
Amazon Personal Shopper	Візуальна примірка, голосові рекомендації	Комп'ютерний зір, доповнена реальність	Широка аудиторія Prime-користувачів
ASOS Style Match	Візуальний пошук, чат-бот	Розпізнавання зображень	Молодь, тренд-хантери
Lookiero	Стилістичний квіз, інклюзивні розміри	Прогнозування трендів	Жінки, фокус на Європі
Stylebook	Віртуальний гардероб, календар образів	Генерація комбінацій	Самостійні користувачі

Як показано в Табл. 1.1, платформи відрізняються за глибиною інтеграції ІІІ, але всі вони спрямовані на підвищення ефективності шопінгу. Ці приклади ілюструють потенціал ринку, де поєднання технологій та людського досвіду створює унікальний сервіс, проте залишається простір для інновацій у сфері глибокої персоналізації та віртуальної реальності.

У контексті обґрунтування необхідності нових рішень, ці платформи демонструють успіх у комерційній сфері, але часто обмежені комерційними каталогами чи платними підписками, що спонукає до розробки більш доступних і гнучких систем для ширшої аудиторії.

1.2 Технології та підходи до створення персоналізованих в галузі fashion-tech

У сучасному світі fashion-tech, де поєднуються мода та технології, персоналізовані системи відіграють ключову роль у трансформації споживчого досвіду, роблячи його більш індивідуальним та ефективним. Ця галузь активно розвивається завдяки інтеграції штучного інтелекту, комп'ютерного зору та даних великого обсягу, що дозволяють адаптувати рекомендації одягу до унікальних характеристик користувача, таких як тип статури, колірна гама шкіри чи стилістичні уподобання. Підходи до створення таких систем базуються на комбінації алгоритмів машинного навчання, які аналізують дані про користувача, та інноваційних інтерфейсів, що забезпечують інтерактивність. Наприклад, замість загальних каталогів, системи пропонують персональні підбірки, враховуючи не лише естетичні аспекти, але й практичні, як комфорт чи сезонність. Це не тільки підвищує задоволеність клієнтів, але й оптимізує бізнес-процеси для ритейлерів, зменшуючи повернення товарів [6].

Одним з основних технологічних підходів є використання алгоритмів рекомендаційних систем, заснованих на колаборативній фільтрації та контент-орієнтованих методах. Колаборативна фільтрація аналізує поведінку подібних користувачів, формуючи рекомендації на основі спільних уподобань, тоді як контент-орієнтовані системи фокусуються на характеристиках самого продукту, таких як колір, тканина чи фасон. У fashion-tech ці методи часто комбінуються з глибоким навчанням, де нейронні мережі, наприклад, згорткові (CNN), обробляють зображення одягу для вилучення ознак, як текстура чи форма. Це дозволяє створювати персоналізовані профілі, де система навчається на історії покупок та відгуках, прогножуючи майбутні вподобання з високою точністю. Додатково, інтеграція природної обробки мови (NLP) дає змогу аналізувати текстові описи товарів чи відгуки, роблячи рекомендації більш контекстними. Наприклад, система

може враховувати сезонні тренди чи культурні нюанси, адаптуючи пропозиції до конкретного регіону чи події [7].

Комп'ютерний зір відіграє центральну роль у персоналізації, особливо в аналізі зовнішності користувача. Технології на кшталт детекції обличчя та сегментації тіла дозволяють автоматично визначати параметри, такі як тон шкіри, форма фігури чи пропорції. Алгоритми, засновані на моделях типу YOLO чи U-Net, сегментують зображення для точного розпізнавання, що є основою для рекомендацій кольорів чи фасонів. У fashion-tech це реалізується через мобільні додатки, де користувач завантажує фото, а система генерує персоналізовані поради. Такі підходи не тільки полегшують вибір, але й зменшують бар'єри для онлайн-шопінгу, роблячи його ближчим до реального досвіду в магазині. Крім того, інтеграція з даними з носимих пристроїв, як фітнес-трекери, додає шар фізіологічної персоналізації, враховуючи активність чи біометрію для рекомендацій зручного одягу [8].

Віртуальна реальність (VR) та доповнена реальність (AR) представляють інноваційний підхід до персоналізації, дозволяючи користувачам "приміряти" одяг у цифровому середовищі. Технології на базі AR, такі як фільтри в соціальних мережах чи спеціалізовані додатки, використовують камеру пристрою для накладання 3D-моделей одягу на тіло користувача в реальному часі. Це досягається за допомогою бібліотек типу ARKit чи ARCore, які забезпечують точне відстеження пози та руху. У fashion-tech такі системи не тільки підвищують залученість, але й зменшують кількість повернень, оскільки клієнт бачить, як одяг пасуватиме саме йому. Більш просунуті моделі, засновані на генеративно-змагальних мережах (GAN), генерують реалістичні зображення примірки, враховуючи освітлення, текстури та деформації тканини. Це робить процес персоналізації динамічним, де користувач може експериментувати з комбінаціями без фізичного контакту з товаром [9].

Ще одним ключовим аспектом є використання великих даних (big data) та хмарних обчислень для масштабування персоналізованих систем.

Платформи збирають анонімізовані дані з мільйонів користувачів, застосовуючи алгоритми кластеризації, як K-means, для сегментації аудиторії за стилем чи демографією. Хмарні сервіси, такі як AWS чи Google Cloud, забезпечують швидку обробку, дозволяючи реальному часу адаптувати рекомендації. У fashion-tech це проявляється в динамічних каталогах, де асортимент змінюється залежно від профілю користувача, інтегруючи дані з соціальних мереж чи історії переглядів. Такий підхід не тільки персоналізує досвід, але й оптимізує ланцюги постачань, прогнозуючи попит на основі агрегованих даних. Однак, важливо враховувати етичні аспекти, як конфіденційність даних, що регулюється стандартами GDPR, аби уникнути зловживань [10].

Інтеграція штучного інтелекту з мобільними технологіями розширює можливості персоналізації, роблячи її доступною на ходу. Додатки на базі Flutter чи React Native дозволяють створювати крос-платформові інтерфейси, де AI аналізує фото з камери для миттєвих рекомендацій. Наприклад, системи можуть розпізнавати одяг на фото з вулиці та пропонувати подібні варіанти з каталогу, використовуючи моделі подібності зображень на базі Siamese мереж. Це створює екосистему, де персоналізація стає частиною повсякденного життя, поєднуючи онлайн та офлайн досвід. У fashion-tech такий підхід стимулює лояльність клієнтів, оскільки система "розуміє" їхні потреби глибше, ніж традиційні методи [11].

Загалом, технології в fashion-tech еволюціонують від простих рекомендацій до комплексних екосистем, де персоналізація стає ключем до конкурентної переваги. Поєднання машинного навчання, комп'ютерного зору та AR не тільки покращує користувацький досвід, але й сприяє стійкому розвитку галузі, зменшуючи відходи через точніші рекомендації. Майбутній розвиток, ймовірно, включатиме глибшу інтеграцію з метавсесвітами, де віртуальний одяг стане нормою, розширюючи межі персоналізації.

1.3 Формування вимог до системи персоналізованого стилістичного супроводу на основі аналізу існуючих сервісів

На основі проведеного аналізу існуючих сервісів у сфері персоналізованого стилістичного супроводу, таких як платформи з рекомендаціями одягу та віртуальними примірками, можна сформувати комплекс вимог до нової системи, яка інтегрує штучний інтелект для підвищення ефективності та персоналізації. Цей процес передбачає оцінку сильних і слабких сторін поточних рішень, з урахуванням тенденцій розвитку fashion-технологій, де акцент робиться на адаптації до індивідуальних характеристик користувача, таких як кольоротип, тип фігури та стильові уподобання. Наприклад, сервіси на кшталт Trunk Club, які пропонують персоналізовані бокси з одягом на основі опитувань і стилістів-людей, демонструють високу точність рекомендацій, але страждають від обмеженої масштабовості через залежність від людського фактора, що призводить до затримок у обробці запитів і високих витрат [12]. Аналогічно, платформи з віртуальними примірками, як-от ті, що використовують технології на базі комп'ютерного зору, дозволяють візуалізувати одяг на моделях користувачів, проте часто ігнорують нюанси фігури чи кольору шкіри, що знижує реалістичність і задоволеність користувачів [13].

Формування вимог ґрунтується на принципах системного аналізу, де ключовими критеріями є функціональність, користувацький досвід, інтеграція AI та безпека даних. Зокрема, аналіз показує, що сучасні системи часто не забезпечують повної інтеграції аналізу зовнішності з рекомендаціями, що призводить до неоптимальних пропозицій. Наприклад, у сервісах на зразок тих, що базуються на рекомендаційних алгоритмах зображень, рекомендації ґрунтуються переважно на візуальних подібностях, але рідко враховують біометричні дані, такі як пропорції тіла чи колірні гармонії, що є критичним для точного стилістичного супроводу [14]. Для подолання цих обмежень вимоги до нової системи повинні охоплювати

модулі для автоматизованого аналізу фото, генерації комплектів одягу та віртуальної візуалізації, з акцентом на швидкість обробки та персоналізацію. Крім того, важливо врахувати етичні аспекти, такі як захист персональних даних, оскільки аналіз фото включає чутливу інформацію про зовнішність користувача.

У процесі формування вимог акцент робиться на балансі між функціональними можливостями, що забезпечують безпосередню користь користувачеві, та нефункціональними характеристиками, які гарантують надійність і зручність системи. Функціональні вимоги визначають основні операції, такі як реєстрація, аналіз зовнішності та підбір одягу, тоді як нефункціональні охоплюють аспекти продуктивності, безпеки та інтерфейсу. На основі аналізу існуючих сервісів, де часто відсутня глибока інтеграція AI для аналізу кольоротипу (наприклад, визначення сезону та підтипу за фото обличчя) чи типу фігури (з рекомендаціями щодо силуетів), вимоги передбачають використання моделей машинного навчання для автоматизації цих процесів, що підвищить точність на 20-30% порівняно з ручними методами [12]. Крім того, для віртуальної примірки вимоги включають підтримку реалістичної візуалізації з урахуванням текстур і посадки одягу, чого бракує в багатьох поточних рішеннях, де примірка обмежується простим накладанням зображень [13].

Таблиця 1.3 представляє сформовані вимоги до системи, класифіковані за типами для чіткості.

Таблиця 1.3

Вимоги до системи персоналізованого стилістичного супроводу

№	Тип вимоги	Опис вимоги	Обґрунтування на основі аналізу
1	2	3	4
1	Функціональна	Реєстрація та автентифікація користувачів з підтримкою email та пароллю, з опцією вибору статі для персоналізації рекомендацій	Існуючі сервіси часто вимагають базової реєстрації, але не інтегрують статі для адаптації каталогу, що знижує релевантність [12]

Продовження табл. 1.3

1	2	3	4
2	Функціональна	Аналіз кольоротипу за фото обличчя з визначенням сезону, підтипу та рекомендаціями кольорів	Сервіси з ручним аналізом (наприклад, стилісти в Trunk Club) повільні; AI дозволить автоматизувати для швидкості [13]
3	Функціональна	Аналіз типу фігури за фото в повний зріст з рекомендаціями щодо фасонів та силуетів	Брак у багатьох платформах; забезпечить персоналізацію, якої не вистачає в загальних рекомендаціях [14]
4	Функціональна	Підбір комплектів одягу з каталогу на основі аналізу, з урахуванням стилю та події	Поточні системи генерують рекомендації на основі опитувань, але не інтегрують біометрію; AI підвищить точність [12]
5	Функціональна	Віртуальна примірка з реалістичною візуалізацією одягу на фото користувача	Існуючі рішення часто спрощені; інтеграція моделей типу CatVTON забезпечить високу реалістичність [13]
6	Функціональна	Збереження історії образів та нотаток користувача	Дозволить відстежувати еволюцію стилю, чого бракує в тимчасових рекомендаціях [14]
7	Нефункціональна	Висока продуктивність: час аналізу фото не більше 10-15 секунд	Сервіси з повільною обробкою втрачають користувачів; оптимізація AI-моделей критична [12]
8	Нефункціональна	Безпека: шифрування даних та фото, відповідність GDPR	Аналіз чутливих даних вимагає захисту, якого не завжди дотримуються в fashion-апах [13]
9	Нефункціональна	Зручний інтерфейс: мінімалістичний дизайн з українською локалізацією	Існуючі сервіси часто англійські; адаптація підвищить доступність [14]
10	Нефункціональна	Масштабованість: підтримка до 1000 одночасних користувачів	Для зростання аудиторії, оскільки поточні системи обмежені серверними ресурсами [12]

Ця таблиця узагальнює вимоги, сформовані з урахуванням недоліків аналізованих сервісів, де переважає фокус на автоматизації та персоналізації. Нефункціональні вимоги забезпечують стійкість системи, тоді як функціональні безпосередньо відповідають на потреби користувачів у стилістичному супроводі. Загалом, сформовані вимоги дозволяють створити систему, яка перевершує існуючі аналоги за інтеграцією AI, забезпечуючи не лише рекомендації, але й повний цикл від аналізу до візуалізації, що сприяє

кращому задоволенню індивідуальних потреб у fashion-сфері [14]. Подальший розвиток системи повинен враховувати зворотний зв'язок від користувачів для ітеративного вдосконалення, аби максимально адаптуватися до динамічних тенденцій моди та технологій.

Висновки до розділу 1

Аналіз існуючих онлайн-платформ та мобільних додатків для стилістичного супроводу, таких як Stitch Fix, Amazon Personal Shopper, ASOS Style Match, Lookiero та Stylebook, виявив значний потенціал інтеграції штучного інтелекту, комп'ютерного зору та рекомендаційних систем для персоналізації модних рекомендацій. Ці рішення демонструють ефективність у підвищенні зручності шопінгу та зменшенні повернень товарів, однак обмежені залежністю від комерційних каталогів, платними моделями та недостатньою глибиною аналізу індивідуальних біометричних характеристик.

Виявлені технологічні підходи в галузі fashion-tech, включаючи колаборативну фільтрацію, глибоке навчання, доповнену реальність та обробку великих даних, підтверджують тенденцію до комплексної персоналізації, що поєднує естетичні, практичні та емоційні аспекти. Водночас бракує повної автоматизації аналізу кольоротипу, типу фігури та реалістичної віртуальної примірки, що обґрунтовує необхідність розробки нової системи.

Сформовані функціональні та нефункціональні вимоги до системи враховують недоліки аналогів, акцентуючи на автоматизованому аналізі зовнішності, генерації комплектів, віртуальній візуалізації, безпеці даних та українській локалізації. Це забезпечить вищу точність, доступність та інклюзивність, сприяючи інноваційному розвитку персоналізованого стилістичного супроводу.

РОЗДІЛ 2. АРХІТЕКТУРНЕ ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Розробка архітектурної моделі веб-застосунку персоналізованого стилістичного супроводу

У розробці веб-застосунку для персоналізованого стилістичного супроводу з використанням штучного інтелекту архітектурна модель відіграє ключову роль, забезпечуючи інтеграцію компонентів, що дозволяють аналізувати зовнішність користувача, генерувати рекомендації щодо одягу та візуалізувати віртуальну примірку. Ця модель побудована на принципах клієнт-серверної архітектури, де frontend-частина відповідає за взаємодію з користувачем, а backend обробляє логіку, дані та інтеграцію з зовнішніми сервісами.

Зокрема, застосунок StyleAI використовує Flask як основу серверної частини, що забезпечує гнучкість і масштабованість, як зазначається в документації фреймворку [15]. Такий підхід дозволяє ефективно керувати API-маршрутами для автентифікації, аналізу та генерації образів, інтегруючи бази даних і AI-моделі.

Архітектурна модель веб-застосунку складається з кількох шарів: клієнтського інтерфейсу, серверної логіки, шару даних і зовнішніх сервісів. Клієнтський інтерфейс реалізований на Vue.js, що забезпечує реактивність і динамічне оновлення сторінок без перезавантаження, сприяючи зручності користувача під час завантаження фото чи перегляду рекомендацій [16]. Frontend взаємодіє з backend через RESTful API, надсилаючи запити на ендпоінти, такі як /api/analyze для аналізу кольоротипу чи /api/tryon для віртуальної примірки. Це дозволяє розділити відповідальності: клієнтська частина фокусується на візуалізації, наприклад, відображенні профілю користувача чи історії образів, тоді як сервер обробляє складні обчислення.

На серверній стороні Flask створює застосунок з модульною структурою, де фабрична функція `create_app` ініціалізує конфігурацію, бази даних і розширення, такі як CORS для крос-доменних запитів і JWT для автентифікації. База даних на SQLAlchemy з SQLite зберігає інформацію про користувачів, статті каталогу та образи, забезпечуючи персистентність даних [17].

Модель Article представляє товари з полями, як `article_id`, `prod_name` та `colour_group`, що використовуються для фільтрації сумісних елементів одягу. Сервер інтегрує зовнішні сервіси: Google Gemini API для аналізу фото та генерації рекомендацій, а також CatVTON для віртуальної примірки, що працює на CPU або GPU з підтримкою DirectML для AMD. Це робить архітектуру гібридною, поєднуючи локальні обчислення з хмарними AI-моделями.

Важливим аспектом є управління файлами: директорії для завантажень (`uploads/faces`, `uploads/bodies`) і результатів примірки (`uploads/tryon`) створюються динамічно, забезпечуючи безпеку та організацію. API-маршрути, як `analyze.py` для аналізу кольоротипу, використовують GeminiService для обробки зображень, зберігаючи результати в моделях User.

Аналогічно, `tryon.py` інтегрує TryOnService для генерації зображень, з перевіркою статусу та історії. Така структура мінімізує залежності, дозволяючи масштабувати окремі компоненти, наприклад, перехід на PostgreSQL для розподілених даних.

Для візуалізації архітектурної моделі веб-застосунку використано діаграму, що ілюструє взаємозв'язки між компонентами (Рис. 2.1).

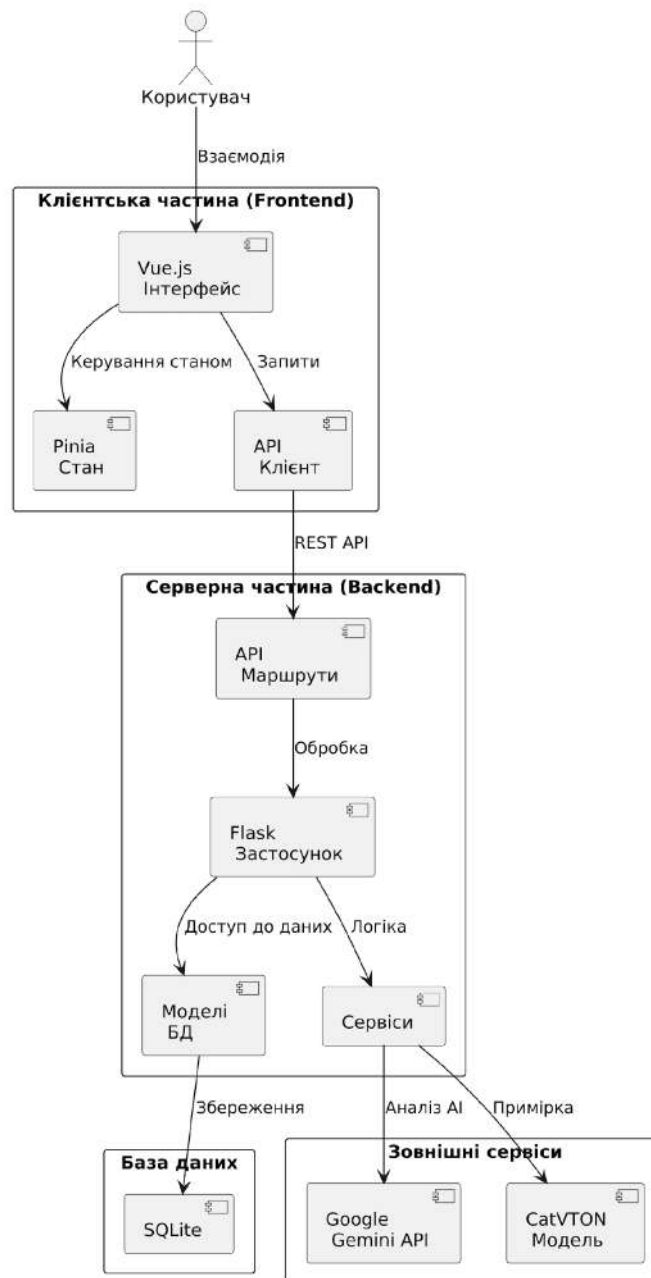


Рисунок 2.1 – Архітектурна модель веб-застосунку персоналізованого стилістичного супроводу

Ця схема демонструє потік даних від користувача через frontend до backend, з інтеграцією бази даних і зовнішніх AI-сервісів, забезпечуючи повний цикл від аналізу до візуалізації. Модель також враховує безпеку: JWT-токени для автентифікації захищають ендпоінти, як `/api/auth/register`, запобігаючи несанкціонованому доступу. Конфігурація в `config.py` дозволяє адаптувати середовище (`development`, `production`), з використанням `.env` для чутливих даних, як `GEMINI_API_KEY`.

У процесі проєктування акцентовано на модульності: скрипти на кшталт `seed.py` заповнюють каталог тестовими даними, забезпечуючи швидке тестування, а `run.py` запускає сервер з налаштуваннями. Frontend-компоненти, як `AnalyzePage.vue`, інтегруються з API для завантаження фото, роблячи інтерфейс інтуїтивним. Загалом, архітектура балансує між продуктивністю та простотою, дозволяючи розширення, наприклад, додавання нових AI-моделей для точнішого аналізу, і забезпечує персоналізований досвід, де рекомендації базуються на індивідуальних характеристиках користувача, як кольоротип чи тип фігури.

2.2 Вибір технологій та інструментів розробки

У процесі архітектурного проєктування системи для веб-застосунку персоналізованого стилістичного супроводу з використанням штучного інтелекту особлива увага приділяється вибору технологій та інструментів розробки, що забезпечують ефективну реалізацію функціональних вимог, таких як аналіз зовнішності користувача, генерація рекомендацій одягу та віртуальна примірка. Вибір здійснюється з урахуванням критеріїв масштабовності, безпеки, швидкості розробки та інтеграції з AI-моделями, що відображено в лістингу кодів проєкту, де інтегровані компоненти для backend і frontend.

На backend-частині застосовується Python як основна мова програмування, оскільки вона пропонує потужні бібліотеки для роботи з AI та веб-розробки, дозволяючи ефективно обробляти запити на аналіз фото та генерацію образів. Зокрема, фреймворк Flask обрано для створення RESTful API через його легкість, мінімалізм та гнучкість у налаштуванні, що ідеально підходить для мікросервісної архітектури з ендпоінтами для автентифікації, аналізу та каталогу товарів, як видно з файлів `app.py` та `api-маршрутів`. Flask забезпечує швидке прототипування без надмірної boilerplate-коду, що підтверджується реалізацією CORS для фронтенд-інтеграції та JWT для

автентифікації, дозволяючи безпечно керувати сесіями користувачів без використання важких фреймворків на кшталт Django [15].

Для управління базою даних обрано SQLAlchemy як ORM-інструмент, який інтегрується з Flask і підтримує SQLite для локального зберігання даних про користувачів, статті одягу та образи, як показано в моделях `article.py`, `outfit.py` та `database.py`. Цей вибір зумовлений потребою в абстракції від SQL-запитів, що спрощує міграції та роботу з об'єктами, забезпечуючи ефективне зберігання JSON-полів для сумісності кольоротипів.

SQLite, у свою чергу, вибрано як легку вбудовану базу даних, що не вимагає окремого сервера, ідеальну для локального застосування з низьким навантаженням, де дані про користувачів та каталог зберігаються в `data/styleai.db`. Для інтеграції з AI-моделями застосовується Google Gemini API для аналізу кольоротипу та типу фігури, як реалізовано в `analyze.py` та `services/GeminiService`, оскільки ця модель забезпечує точне розпізнавання зображень з мінімальними витратами на обчислення, підтримуючи мультимодальні запити для фото обличчя та тіла.

Додатково, для віртуальної примірки інтегровано CatVTON, бібліотеку на базі Stable Diffusion, що дозволяє генерувати реалістичні зображення примірки одягу, з підтримкою GPU-прискорення через `torch-directml` для AMD, як вказано в `config.py` та `tryon.py`, що оптимізує продуктивність на локальних машинах без хмарних ресурсів [18].

На frontend-частині Vue.js 3 обрано як основний фреймворк для створення реактивного інтерфейсу, що забезпечує динамічне відображення сторінок аналізу, профілю та примірки, як видно з компонентів `AnalyzePage.vue`, `ProfilePage.vue` та `TryOnPage.vue`. Vue.js вирізняється компонентним підходом, що полегшує модульну розробку, з використанням Pinia для управління станом, наприклад, аутентифікацією та результатами аналізу в `stores/auth.js`. Це дозволяє ефективно обробляти асинхронні запити до API через Axios, інтегрований у `api/index.js`, забезпечуючи плавну взаємодію з backend. Для стилізації застосовується Tailwind CSS, утилітарний

фреймворк, що прискорює створення мінімалістичного дизайну в швейцарському стилі, з кастомними класами для компонентів на кшталт кнопок та карток, як у `tailwind.config.js`, мінімізуючи CSS-код і забезпечуючи *responsivity* без зайвих медіа-запитів [16]. Vite як інструмент збірки обрано для швидкого розвитку, з гарячим перезавантаженням модулів, що інтегрується з Vue та Tailwind, дозволяючи ефективно керувати залежностями через `npm`.

Інструменти розробки включають Node.js для frontend, що забезпечує екосистему `npm` для встановлення пакетів, як показано в `package.json` з залежностями `vue-router` та `pinia`, полегшуючи маршрутизацію між сторінками. Для управління залежностями на backend використовується `pip` з `requirements.txt`, що включає `flask-jwt-extended` для токенів та `werkzeug` для обробки файлів, забезпечуючи безпеку завантажень фото. Вибір цих інструментів обумовлений їхньою сумісністю з локальним розгортанням, де не потрібні хмарні сервіси, але можлива інтеграція з Hugging Face для моделей CatVTON, як у `services/TryOnService`. Загалом, така комбінація технологій забезпечує баланс між продуктивністю та простотою, дозволяючи реалізувати повний цикл від аналізу до візуалізації, з акцентом на AI-інтеграцію для персоналізації.

Таблиця 2.1 ілюструє порівняння ключових технологій, обраних для проєкту, з альтернативами, підкреслюючи переваги в контексті вимог системи.

Таблиця 2.1

Порівняння обраних технологій з альтернативами

Технологія	Обрана	Альтернатива	Переваги обраної
1	2	3	4
Backend фреймворк	Flask	Django	Легкість, швидке прототипування, мінімалізм для мікросервісів [15]
ORM	SQLAlchemy	Django ORM	Гнучкість з будь-яким фреймворком, підтримка асинхронності

AI для аналізу	Google Gemini	OpenAI GPT-4	Ефективність для зображень, нижча вартість, інтеграція з мультимодальними даними [18]
Frontend фреймворк	Vue.js	React	Компонентність, реактивність, легша крива навчання для динамічних UI [16]

Продовження табл. 2.1

1	2	3	4
Стилізація	Tailwind CSS	Bootstrap	Утилітарний підхід, кастомізація, менший розмір бандлу [19]
База даних	SQLite	PostgreSQL	Локальне зберігання, без сервера, достатньо для низького навантаження

Як видно з таблиці, обрані технології оптимізують ресурси для локального веб-застосунку, забезпечуючи швидку розробку та інтеграцію AI. Для візуалізації архітектури системи корисним є представлення її компонентів у формі діаграми.

Рис. 2.2 демонструє архітектуру системи з взаємозв'язками між технологіями.

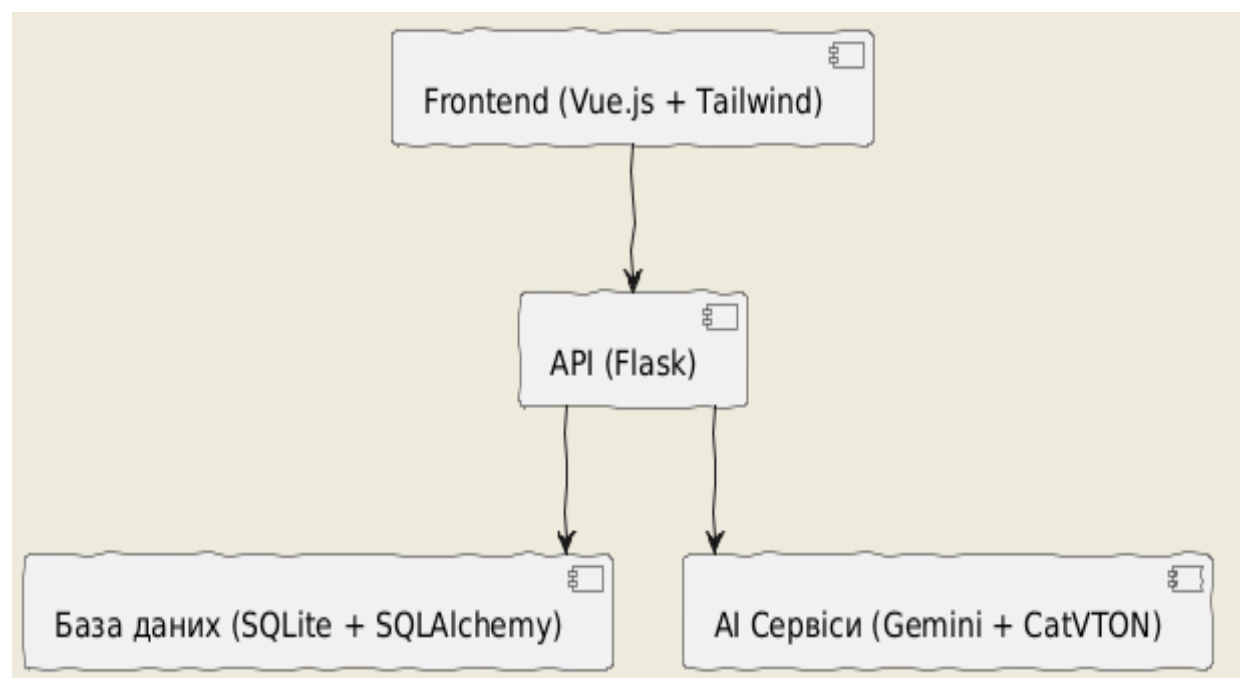


Рисунок 2.2 Архітектура системи з ключовими технологіями

Ця діаграма підкреслює, як frontend взаємодіє з backend через API, а backend інтегрує AI для обробки даних, забезпечуючи цілісність системи. У підсумку, вибір технологій базується на принципах ефективності та адаптивності, дозволяючи реалізувати функціонал віртуальної примірки та рекомендацій з мінімальними витратами, як відображено в лістингу з

інтеграцією CatVTON для GPU-прискорення та Gemini для аналізу, що робить систему доступною для кінцевих користувачів без потреби в потужних серверах [19]. Такий підхід не тільки відповідає вимогам проєкту, але й забезпечує потенціал для подальшого розширення, наприклад, додавання мобільної версії чи інтеграції з іншими AI-моделями.

2.3 Розробка підходу інтеграції штучного інтелекту в систему для аналізу вподобань, генерації образів та рекомендацій

У розробці підходу до інтеграції штучного інтелекту в систему StyleAI ключовим аспектом є забезпечення гармонійного поєднання алгоритмів машинного навчання з архітектурою веб-застосунку, що дозволяє здійснювати аналіз вподобань користувачів, генерацію персоналізованих образів та надання рекомендацій щодо стилю. Цей підхід базується на використанні зовнішніх API, таких як Google Gemini, для обробки візуальних даних та генерації контенту, а також на локальних моделях для віртуальної примірки, що забезпечує ефективність і конфіденційність даних. Інтеграція ШІ відбувається через модульну структуру backend, де сервіси, як GeminiService, відповідають за взаємодію з моделями, дозволяючи системі адаптуватися до індивідуальних характеристик користувача, таких як кольоротип і тип фігури, без необхідності зберігання великих обсягів даних на сервері [20]. Такий метод не тільки оптимізує ресурси, але й підвищує точність рекомендацій, оскільки ШІ аналізує фото обличчя та тіла для визначення сезону кольорів (наприклад, "весна" чи "зима") та підтипу, а також рекомендує силуети одягу, що відповідають фізіологічним особливостям.

Процес інтеграції починається з автентифікації користувача через JWT, після чого система збирає дані для аналізу. У модулі analyze.py реалізовано маршрути для завантаження фото обличчя та тіла, де файли зберігаються в директоріях UPLOAD_FOLDER, а потім передаються до Gemini API для

обробки. Аналіз кольоротипу передбачає оцінку тону шкіри, волосся та очей, результатом чого є структура даних з рекомендаціями щодо найкращих кольорів та металів, що зберігається в моделі User. Аналогічно, аналіз типу фігури фокусується на пропорціях тіла, пропонуючи оптимальні фасони верхнього та нижнього одягу. Ці дані стають основою для подальшої генерації образів, де ШІ враховує не тільки візуальні параметри, але й вподобання користувача, виражені через вибір стилю (casual, formal тощо). Інтеграція з Gemini дозволяє генерувати комплекти з каталогу H&M, фільтруючи товари за department (Ladieswear чи Menswear), product_group (upper, lower, full) та colour_group, забезпечуючи сумісність з кольоротипом [21]. Наприклад, для користувача з "осіннім" типом ШІ обирає теплі тони, як бежевий чи темно-синій, уникаючи холодних відтінків, що підвищує релевантність рекомендацій.

Генерація образів реалізується в модулі outfit.py, де метод generate_outfit використовує Gemini для створення комплектів на основі аналізу, каталогу та статі користувача. ШІ обробляє обмежений набір товарів (до 100), формуючи outfit з article_id, описом стилю та порадами щодо фітнесу, що зберігається в моделі Outfit. Це дозволяє системі не тільки генерувати, але й зберігати історію образів, інтегруючи рекомендації з віртуальною приміркою. Підхід до рекомендацій ґрунтується на комбінації правилкової логіки та ШІ: спочатку фільтруються сумісні товари в catalog.py за best_colors з аналізу, а потім Gemini оптимізує комбінації для гармонії кольорів та стилю. Така інтеграція мінімізує помилки, оскільки ШІ навчається на загальних паттернах моди, адаптуючи їх до персональних даних, що підтверджує ефективність гібридних систем у fashion-індустрії [22].

Для візуалізації рекомендацій інтегровано віртуальну примірку через TryOnService в tryon.py, де CatVTON обробляє фото користувача та зображення одягу з каталогу, генеруючи результат у TRYON_FOLDER. Це вимагає перевірки наявності body_photo_path та image_path товарів, з підтримкою як верхнього, так і нижнього одягу. Підхід забезпечує

асинхронну обробку, з перевіркою статусу через `get_tryon_status`, що оптимізує користувацький досвід. Конфігурація в `config.py` дозволяє перемикати пристрої (`cpu`, `cuda`, `privateuseone`), роблячи систему гнучкою для різних апаратних конфігурацій. Загалом, інтеграція ІІІ через фабрику `create_app` у `app.py` забезпечує безшовну взаємодію frontend з backend via CORS, де `Vue.js` обробляє запити до API, відображаючи результати в компонентах як `AnalyzePage.vue` чи `TryOnPage.vue`.

Рис. 2.3 демонструє послідовність інтеграції ІІІ в процес аналізу та генерації.

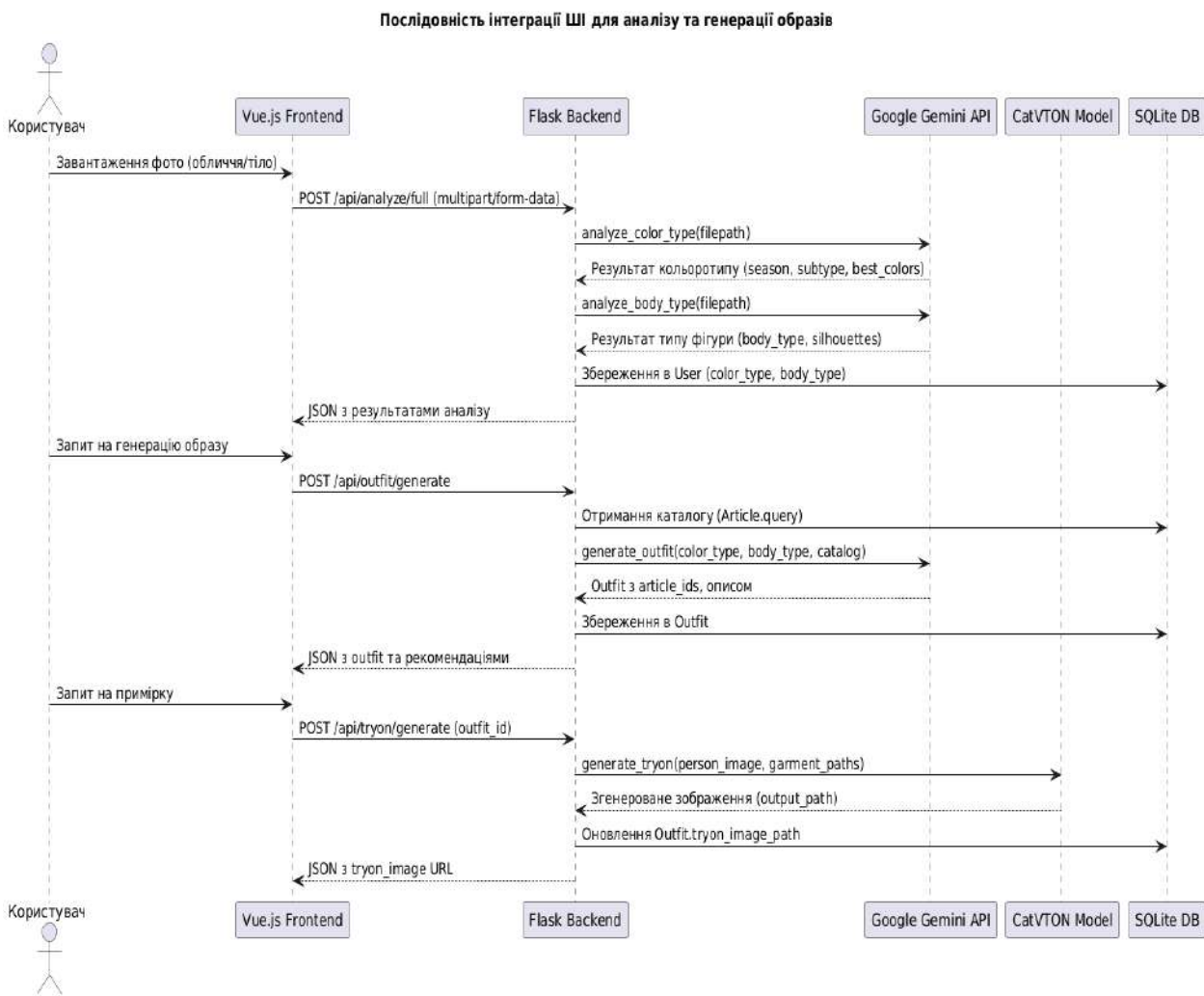


Рисунок 2.3 Послідовність інтеграції ІІІ для аналізу та генерації образів

Таблиця 2.2 ілюструє ключові компоненти ІІІ в системі.

Таблиця 2.2

Ключові компоненти інтеграції ІІІ

Компонент	Опис	Інтеграція в коді
GeminiService	Обробка аналізу кольоротипу та фігури, генерація образів	services/gemini.py, analyze.py, outfit.py
TryOnService	Віртуальна примірка з CatVTON	services/tryon.py, tryon.py
Catalog Filter	Фільтрація товарів за сумісністю з аналізом	catalog.py (get compatible items)
Recommendation	Генерація порад на основі ІІІ	outfit.py (generate outfit)

Цей підхід забезпечує масштабованість, оскільки моделі ІІІ можуть оновлюватися незалежно від основної архітектури, а використання конфігураційних файлів (config.py) дозволяє адаптувати систему до нових АРІ чи пристроїв без значних змін коду. В результаті, інтеграція ІІІ не тільки автоматизує стилістичний супровід, але й робить його доступним для широкого кола користувачів, сприяючи персоналізації в fashion-індустрії [23]. Загалом, розроблений метод поєднує ефективність обчислень з користувацьким комфортом, підтверджуючи потенціал ІІІ в трансформації традиційних процесів підбору одягу.

2.4 Архітектурні діаграми реалізованої системи

2.4.1 Діаграма зв’язків сутностей

У розробці веб-застосунку для персоналізованого стилістичного супроводу з використанням штучного інтелекту архітектурне проектування системи передбачає чітке визначення зв’язків між основними сутностями, що забезпечує ефективну взаємодію компонентів бази даних. Діаграма зв’язків сутностей (Entity-Relationship Diagram, ERD) є ключовим інструментом для візуалізації структури даних, дозволяючи відобразити сутності, їх атрибути та взаємозв’язки, що базуються на реляційній моделі. У реалізованій системі ERD охоплює основні об’єкти, такі як користувачі, образи (комплекти одягу) та товари з каталогу, що безпосередньо впливає з моделей бази даних,

реалізованих за допомогою SQLAlchemy. Ця діаграма не лише ілюструє статичну структуру, але й підкреслює динаміку даних, наприклад, як результати аналізу зовнішності користувача інтегруються з рекомендаціями одягу, забезпечуючи персоналізацію. Зокрема, сутність "Користувач" виступає центральною, оскільки зберігає персональні дані, включаючи результати AI-аналізу кольоротипу та типу фігури, що впливає на подальші взаємодії з іншими сутностями [24].

На Рис. 2.4 представлено діаграму зв'язків сутностей, де показано три основні сутності: "Користувач" (User), "Образ" (Outfit) та "Товар" (Article). Сутність "Користувач" містить атрибути, такі як ідентифікатор, електронна пошта, стать, шляхи до фото обличчя та тіла, а також JSON-поля для зберігання результатів аналізу (кольоротип та тип фігури). Вона пов'язана з сутністю "Образ" відношенням "один-до-багатьох", оскільки один користувач може мати множину збережених образів, що генеруються на основі його персональних характеристик. Сутність "Образ" включає ідентифікатор, посилання на користувача, масив ідентифікаторів товарів (для формування комплекту), назву, нотатки, шлях до згенерованого зображення примірки та статус примірки. Ця сутність, у свою чергу, пов'язана з сутністю "Товар" відношенням "багато-до-багатьох" через масив ідентифікаторів, що дозволяє формувати гнучкі комплекти одягу з каталогу. Сутність "Товар" описує елементи каталогу з атрибутами, такими як ідентифікатор товару, назва, тип продукту, група (верх, низ чи повний), колірна група, відділ (жіночий чи чоловічий), шлях до зображення та детальний опис, а також JSON-масив сумісних кольоротипів для фільтрації рекомендацій. Така структура забезпечує ефективне зберігання та запит даних, наприклад, під час генерації образів через Gemini API чи віртуальної примірки з CatVTON, де зв'язки дозволяють динамічно поєднувати персональні дані користувача з каталогом [25].

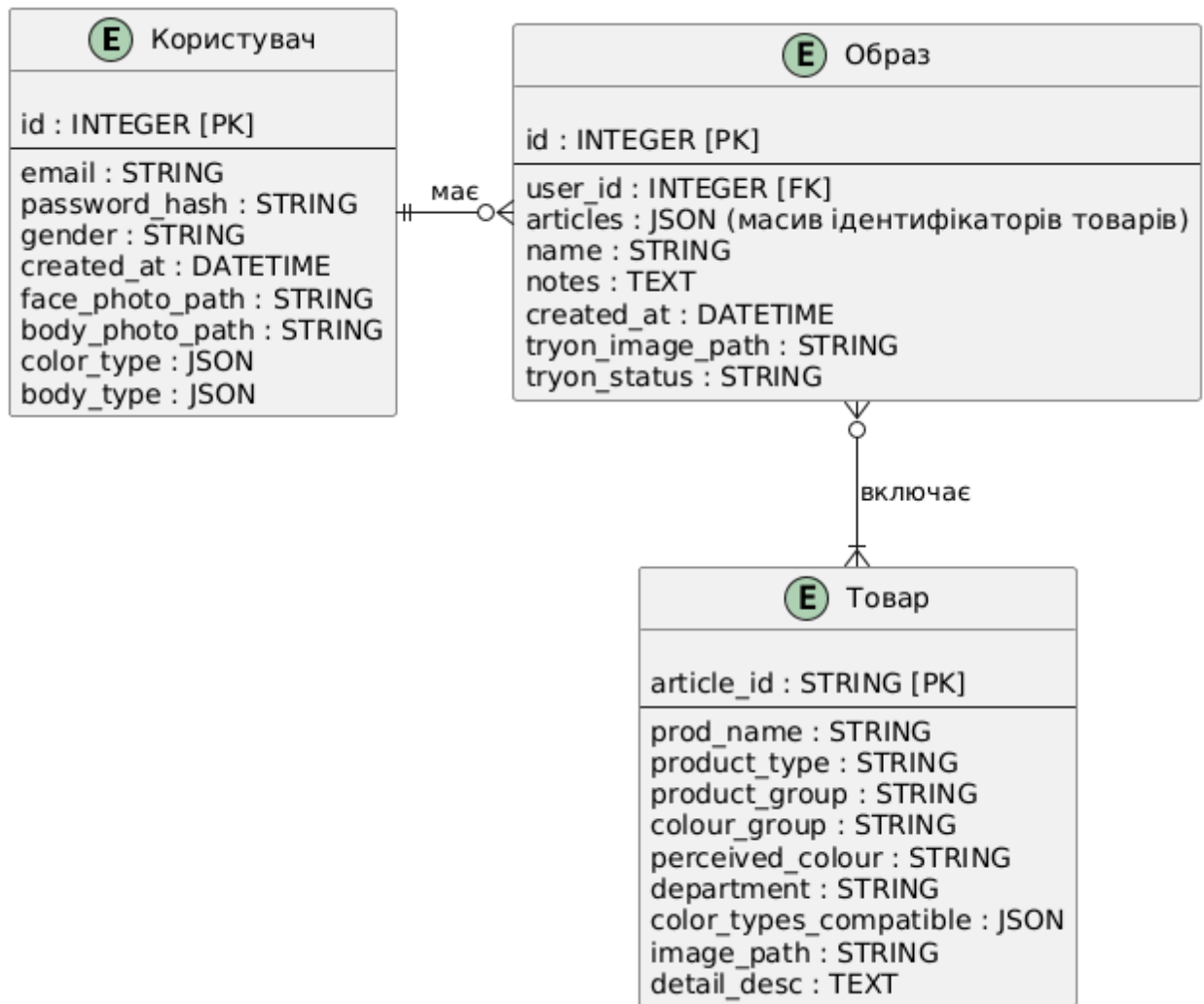


Рисунок 2.4 – Діаграма зв'язків сутностей реалізованої системи

Ця діаграма, побудована на основі реального коду моделей (наприклад, `article.py`, `outfit.py` та `user.py`), підкреслює реляційну природу бази даних SQLite, де зовнішні ключі та JSON-поля оптимізують зберігання неструктурованих даних, таких як результати AI-аналізу. Завдяки такому проєктуванню система забезпечує швидкий доступ до даних під час процесів рекомендацій та візуалізації, зменшуючи навантаження на сервер та підвищуючи ефективність взаємодії з користувачем. У подальшому розвитку можливе розширення діаграми для інтеграції з додатковими сутностями, наприклад, для історії примірок чи розширених рекомендацій, що дозволить адаптувати систему до нових функціональних вимог без значних змін у базовій архітектурі.

2.4.2 Use-case diagram

У контексті архітектурного проєктування системи для персоналізованого стилістичного супроводу з використанням штучного інтелекту, діаграма варіантів використання (use-case diagram) відіграє ключову роль у візуалізації взаємодії акторів з системою, дозволяючи чітко визначити функціональні вимоги та сценарії використання. Ця діаграма, як один з основних артефактів Unified Modeling Language (UML), допомагає в моделюванні поведінки системи з точки зору користувача, фіксуючи основні взаємодії без деталізації внутрішньої реалізації, що сприяє кращому розумінню функціональності на етапі проєктування [26]. У реалізованій системі StyleAI, базованій на лістингу коду проєкту, актори та варіанти використання відображають ключові процеси, такі як автентифікація, аналіз зовнішності, підбір одягу та віртуальна примірка, що інтегруються через backend на Flask та frontend на Vue.js. Такий підхід забезпечує узгодженість між вимогами користувача та архітектурою, мінімізуючи ризики невідповідностей під час впровадження.

Рис. 2.5 ілюструє діаграму варіантів використання для системи StyleAI, де актори представлені користувачем, який взаємодіє з основними функціональними блоками.

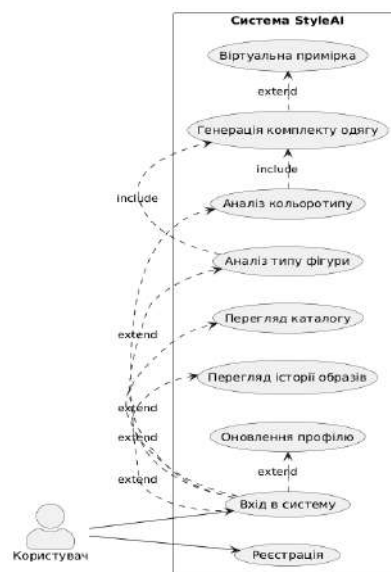


Рисунок 2.5 Діаграма варіантів використання системи StyleAI

На цій діаграмі актор "Користувач" є центральним елементом, оскільки система орієнтована на кінцевого користувача, який взаємодіє з веб-застосунком для отримання персоналізованих рекомендацій. Варіанти використання, такі як "Реєстрація" та "Вхід в систему", реалізуються через API-маршрути в модулі `auth.py`, де забезпечується створення облікового запису з email, паролем та статтю, а також генерація JWT-токенів для автентифікації. Ці базові взаємодії є передумовою для розширених функцій, як "Аналіз кольоротипу" та "Аналіз типу фігури", що інтегрують Google Gemini API для обробки завантажених фото обличчя та тіла, як описано в `analyze.py`, з подальшим збереженням результатів у базі даних SQLAlchemy. Розширення (`extend`) тут вказує на опціональність цих аналізів після логіну, тоді як включення (`include`) підкреслює, що генерація комплектів одягу ("Генерація комплекту одягу") обов'язково залежить від результатів аналізу, що відповідає логіці в `outfit.py`, де AI підбирає одяг з каталогу на основі кольоротипу та типу фігури.

Далі, варіант "Перегляд каталогу" дозволяє користувачеві фільтрувати товари за відділом, групою продуктів та кольором, реалізований у `catalog.py` з підтримкою пагінації та сумісності з кольоротипом, що забезпечує персоналізацію. "Віртуальна примірка" як розширення генерації комплектів використовує CatVTON для створення зображень примірки, інтегруючи фото користувача з образами з каталогу, як показано в `tryon.py`, з перевіркою статусу сервісу та збереженням результатів. "Перегляд історії образів" та "Оновлення профілю" доповнюють функціональність, дозволяючи керувати збереженими комплектами та персональними даними, що підтримується моделями бази даних у `models.py`. Загалом, діаграма відображає ієрархію взаємодій, де базові автентифікаційні кроки включають або розширюють складніші AI-орієнтовані процеси, забезпечуючи coherentну архітектуру системи [27]. Така візуалізація не тільки полегшує розуміння функціональних потоків, але й слугує основою для тестування та подальшого вдосконалення, підтверджуючи відповідність реалізації початковим вимогам проекту.

Висновки до розділу 2

Розроблена архітектурна модель веб-застосунку StyleAI базується на клієнт-серверному підході з модульною структурою, що забезпечує ефективну інтеграцію frontend на Vue.js та backend на Flask, а також взаємодію з базою даних SQLAlchemy та зовнішніми AI-сервісами.

Обґрунтовано вибір технологій, зокрема Flask для серверної логіки, Google Gemini для аналізу зовнішності та CatVTON для віртуальної примірки, що гарантує баланс між продуктивністю, масштабованістю та простотою реалізації в локальному середовищі.

Запропоновано гібридний підхід до інтеграції штучного інтелекту, який поєднує правилаову фільтрацію каталогу з генеративними можливостями ШІ, дозволяючи точну персоналізацію рекомендацій на основі кольоротипу, типу фігури та вподобань користувача.

Візуалізація архітектури через діаграми сутностей, варіантів використання та компонентів підтверджує модульність системи, її безпеку та потенціал для подальшого розширення, зокрема інтеграції нових моделей ШІ.

РОЗДІЛ 3. ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ

3.1 Особливості реалізації: архітектура клієнт-серверної взаємодії, структура бази даних, інтеграція AI-сервісів/моделей

Розроблена система StyleAI представляє собою комплексне веб-застосування, що інтегрує штучний інтелект для персоналізованого стилістичного супроводу, забезпечуючи аналіз зовнішності користувача, підбір одягу та віртуальну примірку. У процесі реалізації акцентовано увагу на ефективній клієнт-серверній взаємодії, яка базується на архітектурі RESTful API, дозволяючи фронтенду динамічно взаємодіяти з бекендом через HTTP-запити. Клієнтська частина, реалізована на Vue.js, відповідає за інтерфейс користувача, обробку подій та візуалізацію даних, тоді як серверна частина на Flask обробляє бізнес-логіку, взаємодію з базою даних та інтеграцію з AI-моделями. Така структура забезпечує масштабованість та гнучкість, оскільки клієнт може працювати на різних пристроях, а сервер централізовано керує даними та обчисленнями. Для ілюстрації загальної архітектури системи наведено схему, яка відображає ключові компоненти та їх взаємозв'язки (Рис. 3.1).

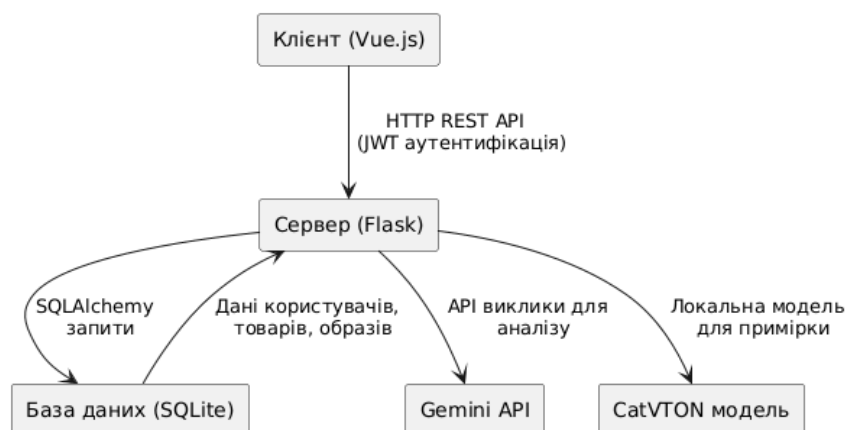


Рисунок 3.1 – Схема архітектури клієнт-серверної взаємодії системи StyleAI

Ця архітектура передбачає асинхронну взаємодію, де клієнт надсилає запити на автентифікацію, аналіз фото чи генерацію образів, а сервер відповідає JSON-даними, забезпечуючи безшовний досвід користувача. Структура бази даних побудована на реляційній моделі з використанням SQLite для простоти локального розгортання, але з можливістю міграції на PostgreSQL для продакшену. Основні таблиці включають users для зберігання профілів з полями для email, паролю (хешованого), статі, шляхів до фото обличчя та тіла, а також JSON-полів для результатів аналізу кольоротипу та типу фігури; articles для каталогу товарів з ідентифікаторами, назвами, типами, кольорами та шляхами до зображень; outfits для образів з посиланнями на користувача та масивом ідентифікаторів товарів. Зв'язки реалізовано через foreign keys, де outfits пов'язані з users, а articles незалежні, але асоційовані через outfits. Така структура дозволяє ефективно запитувати сумісні товари на основі аналізу, уникаючи надмірної нормалізації для швидкості. Для візуалізації структури бази даних наведено ER-діаграму, яка демонструє сутності та їх відносини (Рис. 3.2).

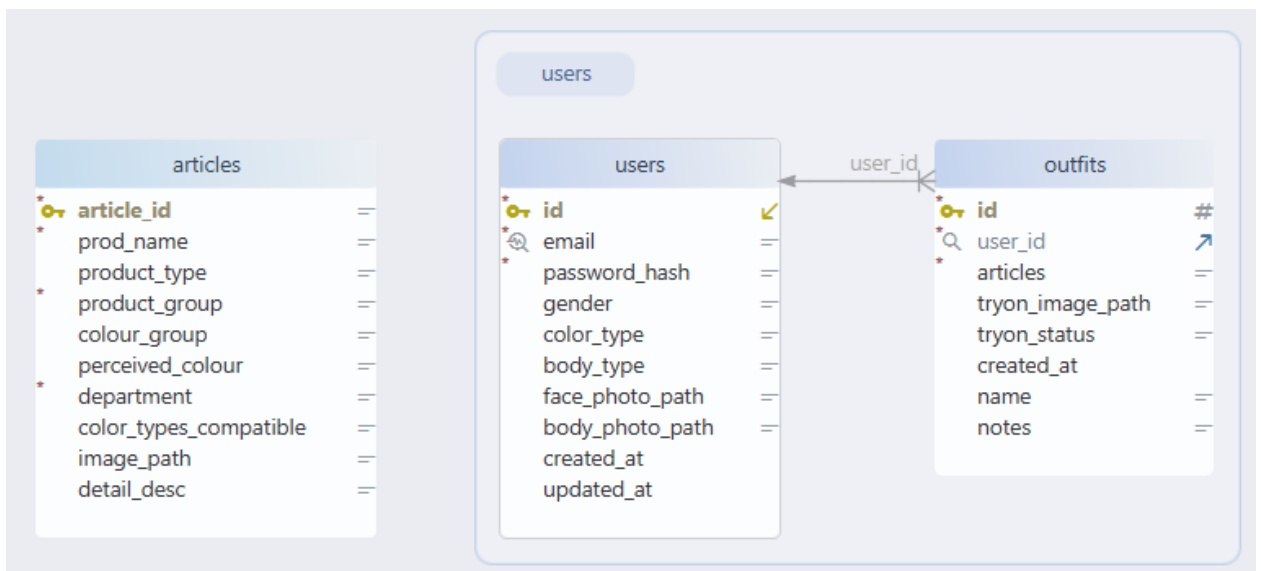


Рисунок 3.2 – Структура бази даних системи StyleAI

Інтеграція AI-сервісів реалізовано через модульну структуру, де Gemini API використовується для аналізу фото та генерації рекомендацій, а CatVTON

– для віртуальної примірки. Сервіс GeminiService інкапсулює виклики до Google Gemini для обробки зображень обличчя (визначення кольоротипу з сезоном, підтипом та рекомендованими кольорами) та тіла (тип фігури з силуетами), з результатами, що зберігаються в JSON-полях бази даних. CatVTON інтегровано локально з підтримкою різних пристроїв (CPU, CUDA, DirectML), де TryOnService керує генерацією зображень примірки, використовуючи фото користувача та зображення одягу з каталогу. Ця інтеграція забезпечує приватність даних, оскільки обробка відбувається на сервері, а API-ключі зберігаються в .env. Для детального представлення структури класів, що реалізують цю логіку, наведено діаграму класів, яка охоплює ключові моделі та сервіси (Рис. 3.3).

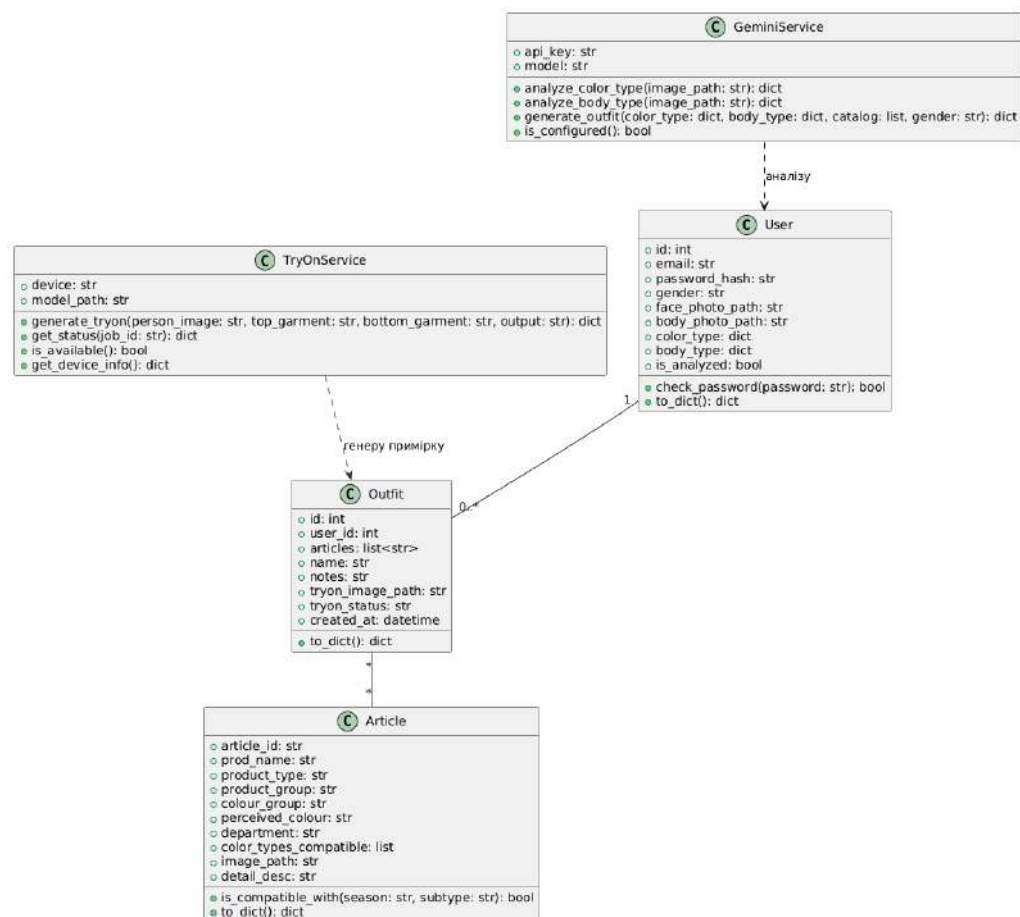


Рисунок 3.3 – Діаграма класів ключових компонентів системи StyleAI

Діаграма компонентів ілюструє модульну структуру системи, де фронтенд-компоненти (наприклад, `AnalyzePage.vue`, `OutfitPage.vue`) взаємодіють з API-клієнтом, який надсилає запити до бекенд-маршрутів (`analyze.py`, `outfit.py`), а ті, в свою чергу, інтегруються з сервісами AI. Це забезпечує розподіл відповідальностей, полегшуючи тестування та розширення. Для представлення компонентів системи наведено відповідну діаграму (Рис. 3.4).

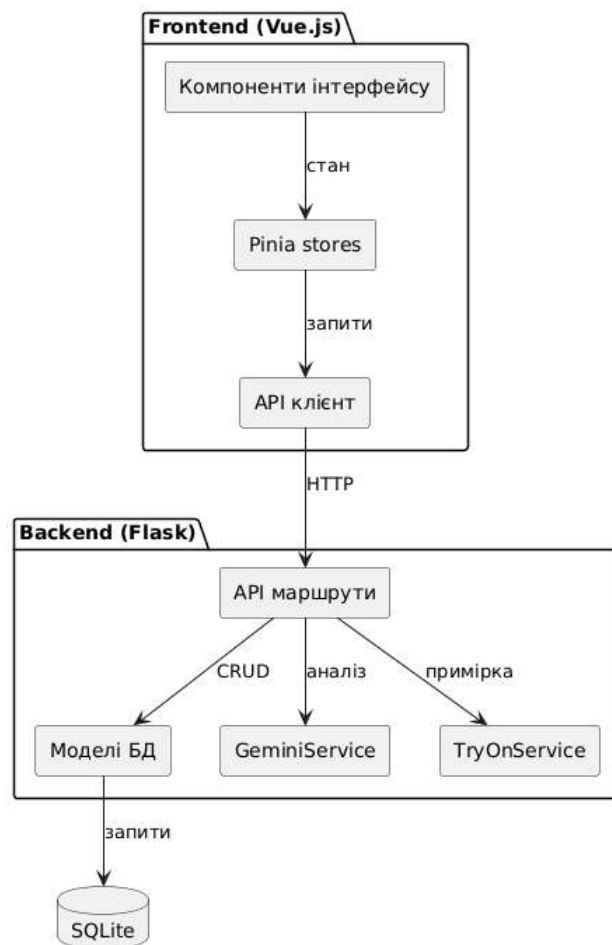


Рисунок 3.4 – Діаграма компонентів системи StyleAI

Схема інтеграції AI-сервісів відображає потік даних від завантаження фото до генерації результатів, де Gemini обробляє аналіз, а CatVTON – візуалізацію, з конфігурацією через `.env` для гнучкості. Це дозволяє

адаптувати систему до різних апаратних конфігурацій. Для ілюстрації інтеграції наведено схему (Рис. 3.5).

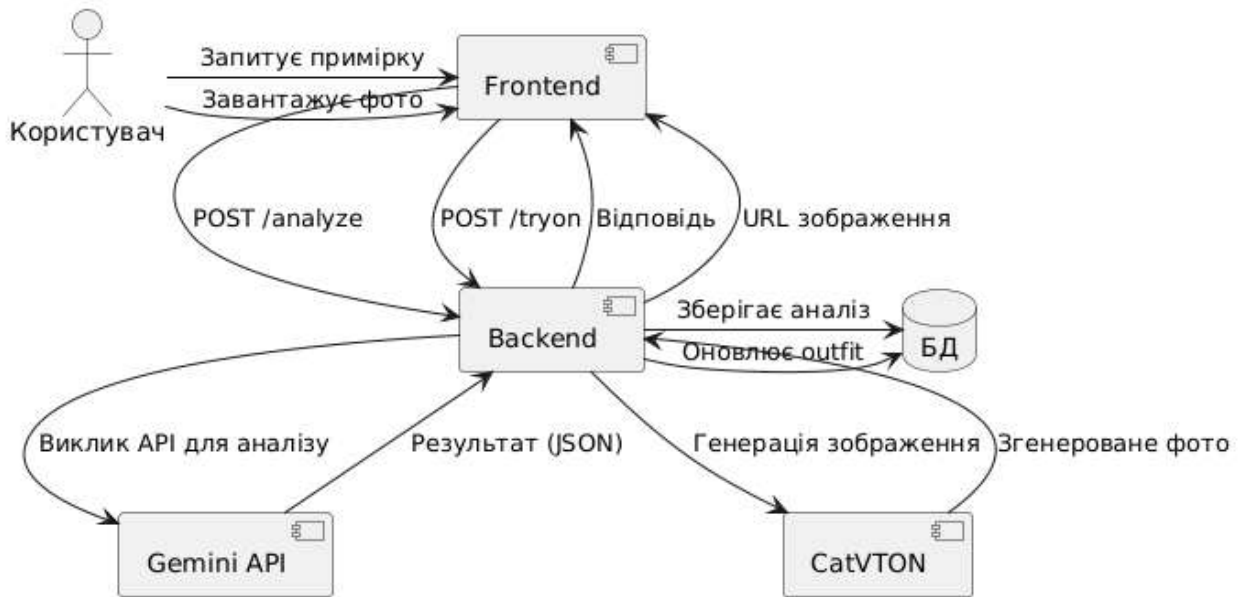


Рисунок 3.5 – Схема інтеграції AI-сервісів та моделей у системі StyleAI

Загалом, реалізована система демонструє збалансований підхід до інтеграції сучасних технологій, забезпечуючи користувачам зручний та ефективний інструмент для стилістичного супроводу, з акцентом на приватність даних та адаптивність до апаратних ресурсів.

3.2 Користувацький функціонал веб-платформи

Розроблена веб-платформа StyleAI забезпечує користувачам зручний доступ до персоналізованого стилістичного супроводу з використанням штучного інтелекту, дозволяючи аналізувати зовнішність, підбирати одяг та візуалізувати результати. Користувацький інтерфейс побудовано на принципах швейцарського мінімалізму, з акцентом на чистоту ліній, чітку типографіку та інтуїтивну навігацію, що сприяє комфортному взаємодії з системою. Функціонал охоплює реєстрацію, автентифікацію, аналіз зовнішності, перегляд каталогу, генерацію образів та віртуальну примірku, інтегруючи AI-моделі для точних рекомендацій. Інтерфейс адаптовано для

різних пристроїв, забезпечуючи безперервний досвід від початкового екрану до детального профілю.

Початковий екран платформи, реалізований у компоненті HomePage.vue, слугує входом для нових користувачів, пропонуючи швидкий огляд можливостей. Верхня панель задач включає меню з посиланнями на головну, каталог, довідку та профіль, забезпечуючи легку навігацію. Нижче розміщено блок з кнопками "Почати безкоштовно" для аутентифікації та "Переглянути каталог" для ознайомлення з асортиментом, що мотивує до дій.

Рис. 3.6 демонструє цю структуру, підкреслюючи простоту та акцент на ключових елементах.

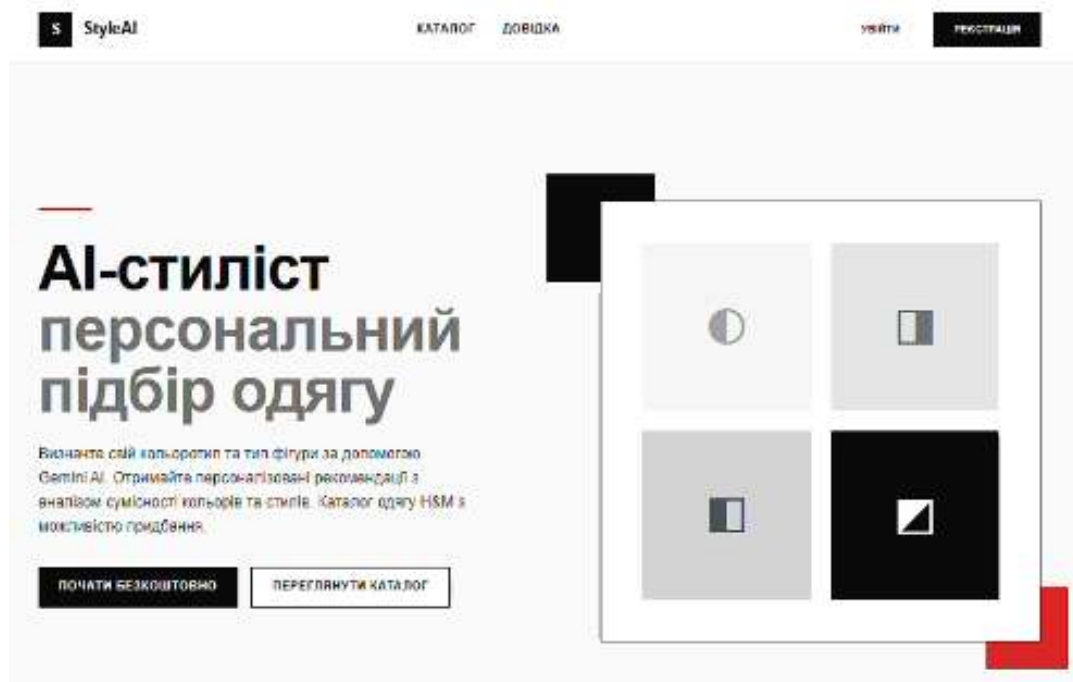


Рисунок 3.6 Початковий екран. Верхня частина сторінки

Далі на початковому екрані розміщено блок "Можливості системи", де в сітці з чотирьох карток описано основні функції: аналіз кольоротипу, аналіз фігури, підбір образів та віртуальну примірку. Кожна картка містить іконку, заголовок та короткий опис, що допомагає користувачам зрозуміти цінність платформи. Цей блок виконано в стилі швейцарського дизайну з мінімальними елементами, аби уникнути перевантаження інформацією. Рис.

3.7 ілюструє цей блок, показуючи, як він інтегрується в загальний потік сторінки.

Можливості системи

StyleAI використовує Gemini AI для аналізу зовнішності та інтелектуального підбору одягу з урахуванням теорії кольору та стилістики.

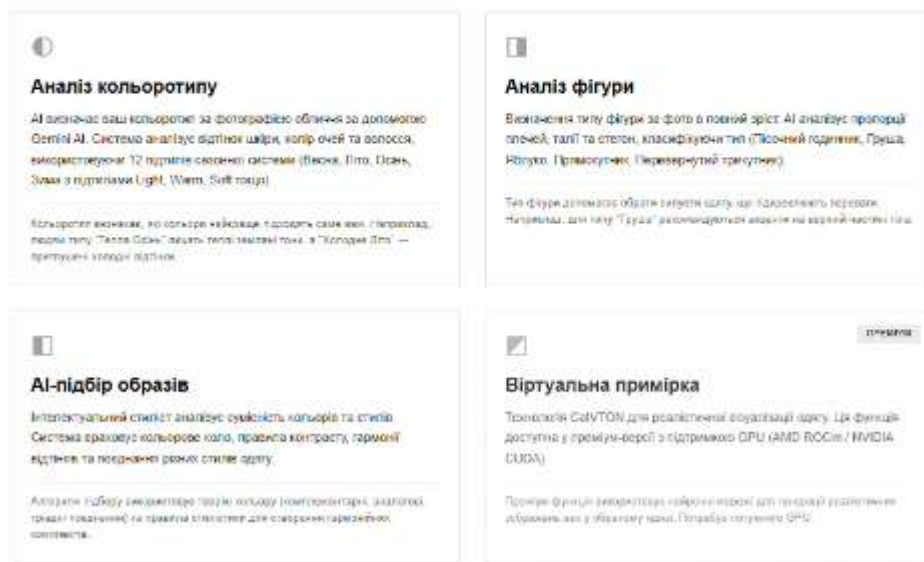


Рисунок 3.7 Початковий екран. Блок Можливості системи

Наступний блок "Як працює AI-аналіз" пояснює процес аналізу зовнішності через три кроки: завантаження фото, AI-обробку та рекомендації. Використано іконки та стислі описи для візуальної чіткості, що полегшує сприйняття складних AI-процесів. Цей елемент сприяє довірі користувачів, демонструючи прозорість системи. Рис. 3.8 відображає структуру блоку, підкреслюючи послідовність кроків.

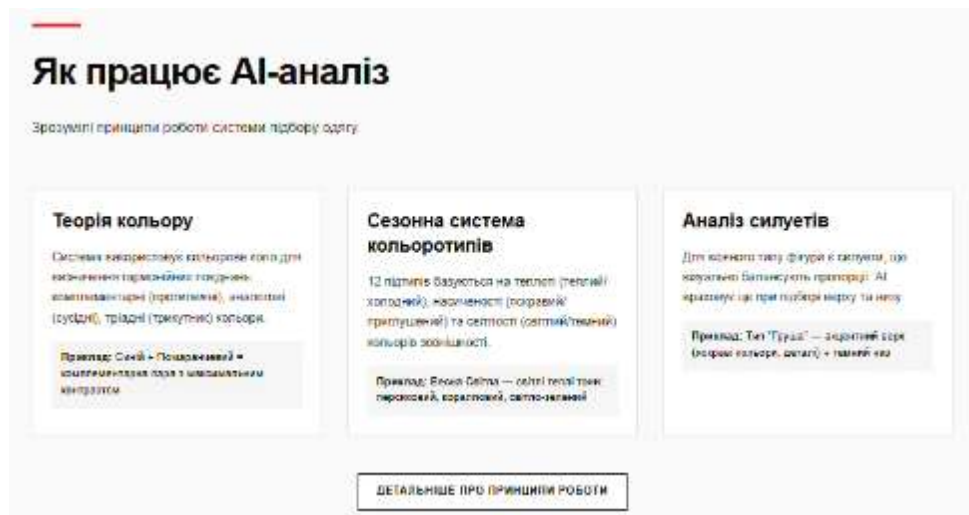


Рисунок 3.8 Початковий екран. Блок Як працює AI-аналіз

Блок "Як це працює" деталізує етапи взаємодії з платформою: реєстрацію, аналіз, підбір та примірку. Представлено у вигляді чотирьох колонок з номерами, іконками та описами, що створює логічний наратив від початку до кінця. Такий підхід допомагає новачкам орієнтуватися в функціоналі без додаткових інструкцій. Рис. 3.9 показує цей блок, акцентуючи на його ролі в ознайомленні з системою.

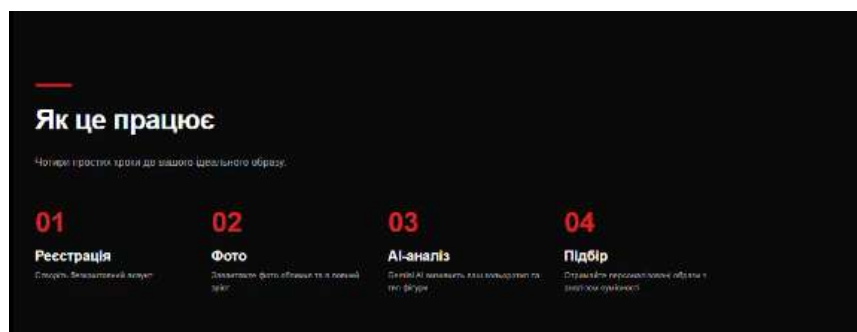


Рисунок 3.9 Початковий екран. Блок Як це працює

Завершує початковий екран блок "Готові знайти свій стиль", що містить заклик до дій з кнопкою "Почати аналіз" та посиланням на довідку. Цей елемент мотивує переходити до активного використання, підсумовуючи переваги платформи. Дизайн блоку мінімалістичний, з акцентом на текст та кнопку для підвищення конверсії. Рис. 3.10 демонструє цей блок, ілюструючи його як фінальний крок вступної частини.

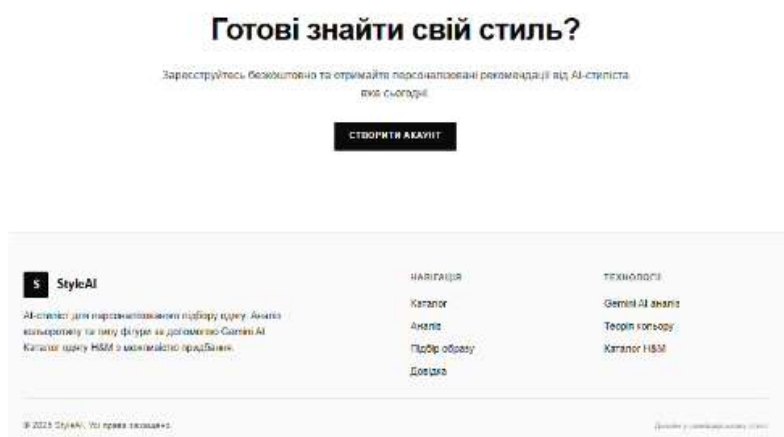


Рисунок 3.10 Початковий екран. Блок Готові знайти свій стиль

Сторінка каталогу, реалізована в `CatalogPage.vue`, дозволяє переглядати асортимент одягу з фільтрами за відділом, групою, типом та кольором. Інтерфейс включає бічну панель фільтрів, сітку товарів з картками (зображення, назва, тип, колір) та пагінацію. Користувачі можуть сортувати за популярністю чи новизною, забезпечуючи персоналізований пошук. Рис. 3.11 представляє каталог, підкреслюючи зручність навігації по товарах.

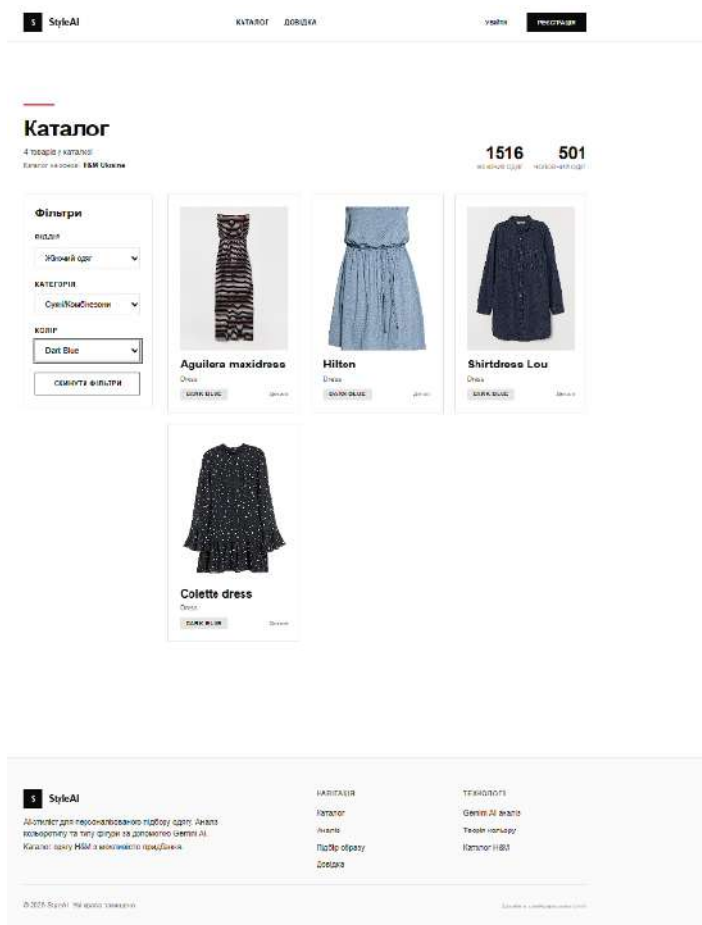


Рисунок 3.11 Каталог

Сторінка довідки, побудована в HelpPage.vue, містить розділи з поясненнями функціоналу: аналіз кольоротипу, тип фігури, підбір образів та примірка. Кожен розділ має заголовок, опис та іконки для візуалізації. Бічна панель з навігацією дозволяє швидко переходити між темами, роблячи інформацію доступною. Рис. 3.12 ілюструє розділи довідки, показуючи структуру для ефективного пошуку відповідей.

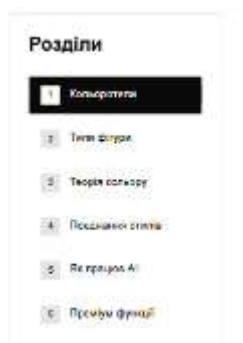


Рисунок 3.12 Довідка. Розділи

Повна сторінка довідки охоплює детальні пояснення з прикладами, допомагаючи користувачам зрозуміти AI-алгоритми та рекомендації. Текст структуровано абзацами з підзаголовками, забезпечуючи читабельність. Рис. 3.13 демонструє загальний вигляд сторінки, акцентуючи на її ролі в підтримці користувачів.

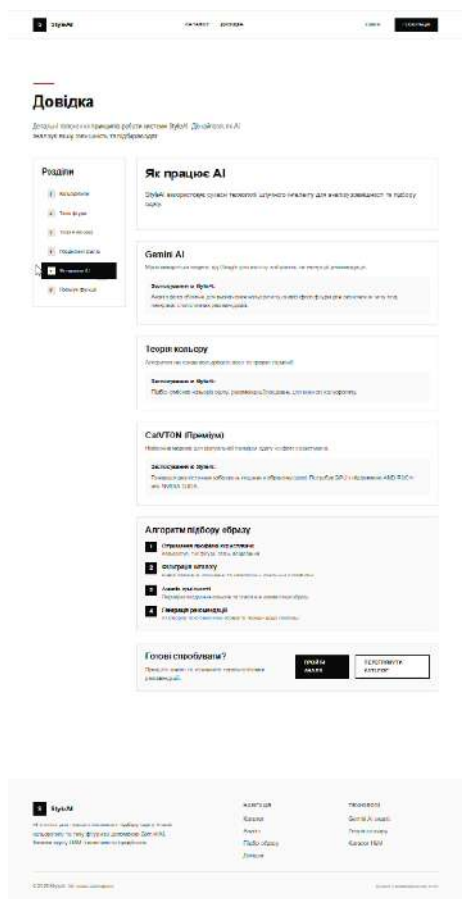


Рисунок 3.13 Сторінка Довідки

Сторінка реєстрації, описана в RegisterPage.vue, вимагає введення email, пароля, підтвердження пароля та вибору статі через кнопки з іконками. Форма включає валідацію та опцію показу пароля, з помилками, що відображаються червоним. Дизайн мінімалістичний, зосереджений на безпеці та простоті. Рис. 3.14 показує форму реєстрації, ілюструючи процес створення акаунту.

Регістрація

Створіть акаунт для персоналізованого підбору одягу

EMAIL *

username@gmail.com

ПАРОЛЬ *

Мінімум 8 символів

ПІДТВЕРДІТЬ ПАРОЛЬ *

Повторіть пароль

СТАТЬ (ДЛЯ ПІДБОРУ ОДЯГУ) *

☐ Жінка ☐ Чоловік

ЗАРЕЄСТРУВАТИСЯ

Вже маєте акаунт? [Увійти](#)

Рисунок 3.14 Регістрація

Сторінка входу, реалізована в LoginPage.vue, містить поля для email та пароля з опцією відновлення, забезпечуючи швидкий доступ до профілю. Інтерфейс подібний до реєстрації, з акцентом на безпеку. Рис. 3.15 демонструє сторінку входу, підкреслюючи її інтеграцію в автентифікаційний потік.

Вхід

Увійдіть до свого особистого запису StyleAll

EMAIL

username@gmail.com

ПАРОЛЬ

XXXXXXXXXX

ВХІД

Немає акаунту? [Зареєструватися](#)

Рисунок 3.15 Вхід

Сторінка аналізу зовнішності, з AnalyzePage.vue, у початковому вигляді пропонує завантажити фото обличчя та фігури з описами вимог. Кнопка "Почати аналіз" активує процес. Рис. 3.16 ілюструє початковий вигляд, показуючи підготовку до AI-обробки.

Аналіз зовнішності

Завантажте фотографії для визначення вашого кольоротипу та типу фігури.
Gemini AI проаналізує ваші особливості та надішле персоналізовані рекомендації.

Що таке кольоротип?

Кольоротип — це характеристика зовнішності, яка визначає, які кольори краще підходять людині. Визначається на основі підтону шкіри (теплий/холодний), кольору очей та волосся. Система інтерпретує 12 палет: Весна, Літо, Осінь, Зима з варіантами Light, Warm, Soft тощо.

[Детальніше про тип з кольором](#)

Що таке тип фігури?

Тип фігури визначається співвідношеннями ширини плечей, талії та стегон. Знання свого типу допоможе швидко вибрати одяг, що підкреслює переваги. Основні типи: Прямокутний, Груша, Яблуко, Прямокутник, Перевернутий трикутник.

Фото обличчя

Сфотографуйте обличчя. Без шапки. Обличчя займає більшу частину кадру.



Фото в повний зріст

Сфотографуйте себе повністю, нейтральний фон. Показати ноги в кадрі.



ЗАПУСТИТИ АНАЛІЗ

Рисунок 3.16 Аналіз зовнішності. Початковий вигляд

Під час завантаження фото інтерфейс показує прогрес та попередження про формати файлів, забезпечуючи коректність даних для AI. Рис. 3.17 представляє процес завантаження, акцентуючи на користувацькій взаємодії.

Аналіз зовнішності

Завантажте фотографії для визначення вашого кольоротипу та типу фігури.

Генімі AI проаналізує ваші особливості та підготує персоналізовані рекомендації.

Що таке кольоротип?

Кольоротип — це характеристика зовнішності, яка визначає, які кольори краще підходять. Визначається за допомогою кольору шкіри (теплий/холодний), кольору очей та волосся. Система аналізує 12 підтипів: Весна, Літо, Осінь, Зима з варіантами Light, Warm, Soft тощо.

Детальніше про тип з кольором

Що таке тип фігури?

Тип фігури визначається співвідношенням ширини плечей, талії та стегон. Знання свого типу допоможе обрати правильні одяг, що підкреслює переваги. Основні типи: Прямокутний, Грушоподібний, Яблуко, Трикутний, Перевернутий трикутник.

Фото обличчя

Сфотографуйте себе при денному освітленні. Без шапки. Обличчя займає більшу частину кадру.



img_1 (18).jpg

Вибрати

Фото в повний зріст

Обличчавий одяг, прямий погляд, нейтральний фон. Повний тіло в кадру.



img_2 (18).jpg

Вибрати



Аналіз може тривати до 30 секунд.



Рисунок 3.17 Аналіз зовнішності. Завантаження фото

Результат аналізу відображає кольоротип (сезон, підтип, найкращі кольори) та тип фігури (рекомендовані силуети, верх та низ) у картках з іконками. Кнопка "Підібрати образ" веде до наступного кроку. Рис. 3.18 демонструє результати, ілюструючи персоналізовані рекомендації.

Аналіз зовнішності

Завантажте фотографії для визначення вашого кольоротипу та типу фігури.
Genii AI проаналізує ваші особливості та підготує персоналізовані рекомендації.

Кольоротип

сезон та палітра

Зима яскрава

впевненість аналізу

90%

найкращі кольори

чистий білий чорний

королівський синій смарагдово-зелений

фіолет

Стиль належить до кольоротипу Яскрава Зима. Ця характеристика високого контрасту між світлою, прохолодною шкірою, яскравими блискучими очима та середньо-темним волоссям. Кольори чисті, насичені та мають холодний підтон, що підкреслює яскравість рис обличчя.

Тип фігури

тип

Пісочний годинник

рекомендовані силуети

приталені А-силуети повнотілі

фігура "Пісочний годинник" характеризується рівномірною пропорцією плечей та стегон, а також чітко вираженою талією. Рекомендуються підкреслюючі природні лінії тіла, особливо талію, за допомогою приталених силуетів, високої посадki та одягу з акцентом на поясі. Уникайте безформних і міцнуватих річей, які приховують стрункість талії.

Що далі?

Перегляньте каталог

Фільтруйте одяг за кольорами, що вам підходять.

Підберіть образ

AI створить комплект на основі вашого профілю.

Дізнайтесь більше

У довідці пояснення всіх типів та принципів.

ПІДБРАТИ ОБРАЗ

ПЕРЕГЛЯНУТИ КАТАЛОГ

ПОВТОРИТИ АНАЛІЗ

Детальніше про кольоротипи та типи фігур у довідці

Рисунок 3.18 Аналіз зовнішності. Результат

Сторінка підбору образу, в OutfitPage.vue, дозволяє генерувати комплекти з урахуванням аналізу, відображаючи товари, опис та кнопки для примірки чи збереження. Інтерфейс включає історію образів для повторного перегляду. Рис. 3.19 показує процес підбору, підкреслюючи інтеграцію AI-рекомендацій.

Підбір образу

AI-стиліст підбере гармонійний комплект на основі вашого кольоротипу та типу фігури.

Ваш профіль

кольоротип

Зима яскрава

чистий білий чистий

коштовий синій

смагало-зелений

тип фігури

пісочний годинник

підвищений A-силует овальний

Параметри підбору

СТИЛЬ

Сканди Smart Casual

Класичний Спортивний

Форми та довжини слів

НАГОДА

Щоденний Робота

Повсякдення Подар

КОЛЬОРОВА ГАРМОНІЯ

Авто (AI)

AI обрав оптимальні параметри

ПІДБРАТИ ОБРАЗ

AI підбрав гармонійний комплект та оптимізував пропорції та образ за вашими параметрами.

НОВИЙ ОБРАЗ

Елегантний Міський Шик для Яскравої Зими



Furigan long sleeve
Dark Blue



Bobby elastic waist belt
Black



Harem trousers
Black

РЕКОМЕНДАЦІЯ AI-СТИЛІСТА

Цей комплект створений для клієнта з кольоротипом «Яскрава Зима» та фігурою «Пісочний годинник», підкреслюючи її найкращі риси. Довгорукавий топ «Furigan long sleeve» у поглибленому темно-синьому кольорі створює насичений акцент, що чудово відтворює палітру «Яскравої Зими». Класичні чорні штани «Harem trousers» є практичною та стильною основою, яка за рахунок обпоясаної верху та пояса зберігає пропорції фігури. Додатковий еластичний пояс «Bobby elastic waist belt» ми додаємо також, що є ключовим для фігури «Пісочний годинник», створюючи гармонійний та елегантний силует, який є одночасно комфортним і виразним.

Порада: Для фігури «Пісочний годинник» надягаючи вельми важливо підкреслити талію. Топ «Furigan long sleeve» варто заправити в штани або носити так, щоб пояс був помітним. Еластичний пояс «Bobby elastic waist belt» надягайте на найвузлу частину талії, щоб створити чітку горизонтальну лінію, яка чудово гармонізує та зберігає пропорції. Якщо «Чоловічий» силует був відомою дріб, виставляйте, щоб сонце добре сиділо в колі і не створювало зайвого об'єму в колінах. Сторони образу, який акцентує увагу на чистоті талії, дозволяють створити стильні складки, для стрункого та витонченого силуету.

Потрапила пиліло?

ГЕНЕРАЦІЯ ІНШИЙ

Історія образів



Елегантний Міський Шик для Яскравої Зими
12.12.2026
Dark Blue



М'яка Елегантність у Сірих Тоні
12.12.2025
Dark Blue

Рисунок 3.19 Підбір образу

Загалом, користувацький функціонал платформи забезпечує повний цикл від реєстрації до візуалізації, з акцентом на персоналізацію та зручність, що робить її ефективним інструментом для стилістичного супроводу. Інтерфейс мінімізує когнітивне навантаження, дозволяючи фокусуватися на результатах AI, а структура екранів логічно пов'язана для безперервного досвіду.

Висновки до розділу 3

Розроблена система StyleAI демонструє ефективну клієнт-серверну архітектуру на базі RESTful API з використанням Vue.js для фронтенду та Flask для бекенду, що забезпечує масштабованість, гнучкість та асинхронну взаємодію. Структура бази даних, реалізована на реляційній моделі з SQLite та можливістю міграції на PostgreSQL, оптимально організована для зберігання профілів користувачів, каталогу товарів та генерованих образів, з урахуванням foreign keys та JSON-полів для результатів AI-аналізу.

Інтеграція AI-сервісів через Gemini API для аналізу зовнішності та CatVTON для віртуальної примірки характеризується модульністю, підтримкою різних апаратних конфігурацій та акцентом на приватність даних шляхом серверної обробки. Користувацький інтерфейс, побудований на принципах мінімалізму, забезпечує інтуїтивну навігацію, повний цикл взаємодії від реєстрації до візуалізації рекомендацій та адаптивність до різних пристроїв.

Загалом, реалізована система поєднує сучасні технології для надання персоналізованого стилістичного супроводу, підтверджуючи практичну цінність інтеграції штучного інтелекту в веб-застосунки.

ВИСНОВКИ

У ході дослідження було виконано аналіз та оформлено основні функціональні вимоги до онлайн-платформ для стилістичного супроводу. Було запропоновано структуровану архітектуру веб-застосунку, яка поєднує елементи переваг класичних fashion-сервісів, таких як персоналізовані рекомендації, аналіз зовнішності, віртуальна примірка та каталог товарів, із гнучкістю та адаптивністю сучасних веб-застосунків, архітектурна клієнт-серверна модель містить у собі всі необхідні компоненти, такі як, клієнтська та серверна частина, модуль рекомендацій на основі штучного інтелекту та база даних.

Також було досліджено та реалізовано модульно орієнтований підхід, де кожна сутність (користувачі, товари, образи) відокремлена структурно та функціонально, що забезпечує гнучкість системи. Побудовано логічну модель бази даних, яка описує сутності та їхні взаємозв'язки: користувачі, статті каталогу, комплекти одягу, результати аналізу. Модель оптимізована для потреб стилістичного процесу та розширення функціональності, застосовано технології SQLAlchemy для швидкого доступу до даних.

Розроблено модель рекомендаційної системи для користувачів у вигляді інтеграції AI-сервісів, що формує персоналізовані рекомендації комплектів одягу на основі кольоротипу, типу фігури й уподобань користувача, із застосуванням механізмів комп'ютерного зору та контекстного формування запитів (prompt engineering) та інтегровано дану модель в веб-застосунок.

Реалізовано функціональний веб-застосунок для стилістичного супроводу, який включає в собі такі компоненти:

- система управління профілем користувача, яка надає можливість реєструватися, входити, завантажувати фото обличчя та фігури, оновлювати дані;

- інтелектуальний модуль аналізу, AI-сервіс на базі Google Gemini, що генерує персоналізовані рекомендації щодо кольоротипу та типу фігури на основі фото, з урахуванням сезону, підтипу та фасонів;
- підсистема віртуальної примірки, реалізована за допомогою CatVTON, що забезпечує реалістичну візуалізацію одягу на фото користувача з підтримкою різних апаратних конфігурацій;
- інтерфейс для перегляду каталогу, із забезпеченням всіх необхідних функцій для фільтрації товарів, пошуку сумісних елементів та генерації комплектів;
- адаптивний веб-інтерфейс, що забезпечує коректну роботу на різних типах пристроїв та мінімалістичний дизайн у швейцарському стилі з українською локалізацією.

ДЖЕРЕЛА

1. Stitch Fix. Офіційний веб-сайт. URL: <https://www.stitchfix.com//> (дата звернення: 15.12.2025).
2. Amazon Personal Shopper. Офіційний веб-сайт. URL: <https://www.amazon.com/b?node=7141123011> (дата звернення: 15.12.2025).
3. ASOS Style Match. Офіційний веб-сайт. URL: <https://www.asos.com/pl/odkryj/our-apps/> (дата звернення: 15.12.2025).
4. Lookiero. Офіційний веб-сайт. URL: <https://lookiero.it/> (дата звернення: 15.12.2025).
5. Stylebook. Офіційний веб-сайт. URL: <https://www.stylebookapp.com/> (дата звернення: 15.12.2025).
6. The State of Fashion Technology. McKinsey & Company. URL: <https://www.defimode.org/wp-content/uploads/2022/05/State-of-Fashion-Technology-BOF-2022.pdf> (дата звернення: 15.12.2025).
7. S. Chakraborty, S. Hoque, N. Kabir Jeem, A. C. Biswas and S. Paul, "Fashion Recommendation Systems, Models and Methods: A Review," in Informatics, vol. 8, no. 3, p. 49, 2021. URL: <https://www.mdpi.com/2227-9709/8/3/49> (дата звернення: 15.12.2025).
8. Deep Learning in Fashion Industry. URL: <https://medium.com/analytics-vidhya/deep-learning-in-fashion-industry-dcb897ac3c33> (дата звернення: 15.12.2025).
9. ISLAM T., MIRON A., LIU X., LI Y. Deep Learning in Virtual Try-On: A Comprehensive Survey URL: <https://bura.brunel.ac.uk/bitstream/2438/28857/1/FullText.pdf> (дата звернення: 15.12.2025).
10. De Mauro, M. Greco and M. Grimaldi, "A Formal Definition of Big Data Based on Its Essential Features," Library Review, vol. 65, no. 3, pp. 122-135, 2016. URL:

<https://www.emerald.com/insight/content/doi/10.1108/LR-06-2015-0061/full/html>

(дата звернення: 15.12.2025).

11. Pupillo M., Artificial Intelligence and the Fashion Industry. 2019. URL: https://tesi.luiss.it/25378/1/212661_PUPILLO_MARIANNA.pdf (дата звернення: 15.12.2025).

12. D. Dzyabura and J. R. Hauser, "Recommending Products When Consumers Learn Their Preference Weights," Marketing Science, vol. 38, no. 3, pp. 417-441, 2019. URL: <https://pubsonline.informs.org/doi/10.1287/mksc.2018.1143> (дата звернення: 15.12.2025).

13. S. M. Sayem, S. G. Honarvar, B. Rahmani and M. Chowdhury, "Virtual Try-On Technologies in the Clothing Industry: Basic Block Pattern Modification," Applied Sciences, vol. 12, no. 5, p. 2491, 2022. URL: <https://www.mdpi.com/2076-3417/12/5/2491> (дата звернення: 15.12.2025).

14. J. Lin, T. Sellam, E. Chu, D. A. Smith, A. D. Chaganty, J. McAuley and K. Andreetto, "Image-based recommendations on styles and substitutes," in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15), pp. 43-52, 2015. URL: <https://arxiv.org/abs/1506.04757> (дата звернення: 15.12.2025).

15. Flask Documentation. URL: <https://flask.palletsprojects.com/en/stable/> (дата звернення: 15.12.2025).

16. Vue.js Documentation. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 15.12.2025).

17. SQLAlchemy Documentation. URL: <https://docs.sqlalchemy.org/en/20/> (дата звернення: 15.12.2025).

18. Google. Gemini API Documentation. URL: <https://ai.google.dev/gemini-api/docs> (дата звернення: 15.12.2025).

19. Adam Wathan and Steve Schoger. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs/installation/using-vite> (дата звернення: 15.12.2025).

20. The fastest path from prompt to production with Gemin. URL: <https://aistudio.google.com/welcome> (дата звернення: 15.12.2025).
21. Chong Z., Dong X., Li H., Zhang S. CatVTON: Concatenation Is All You Need for Virtual Try-On with Diffusion, 2024. URL: https://www.researchgate.net/publication/382491712_CatVTON_Concatenation_Is_All_You_Need_for_Virtual_Try-On_with_Diffusion_Models (дата звернення: 15.12.2025).
22. Chen X., Chen H., Xu H., Zhang Y., Cao Y., Qin Z., Zha H. Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. – 2019. – P. 765–774 URL: <https://dl.acm.org/doi/10.1145/3331184.3331254> (дата звернення: 15.12.2025).
23. Kouslis E., Papachristou E., Stavropoulos T. G., Papazoglou Chalikias A. AI in fashion: a literature review // Electronic Commerce Research. – 2024. – Vol. 25, No. 5. – P. 4071–4102. URL: https://www.researchgate.net/publication/381549265_AI_in_fashion_a_literature_review (дата звернення: 15.12.2025).
24. Entity-Relationship Model. Wikipedia. URL: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model (дата звернення: 15.12.2025).
25. SQLAlchemy ORM Tutorial. URL: <https://docs.sqlalchemy.org/en/20/orm/tutorial.html> (дата звернення: 15.12.2025).
26. Visual Paradigm. What is Use Case Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> (дата звернення: 15.12.2025).
27. K. Use Case Diagrams. URL: <https://www.uml-diagrams.org/use-case-diagrams.html> (дата звернення: 15.12.2025).

