

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ПАНАСЕВИЧ Катерина Олександрівна

**Автоматизована система генерації типографічних пар
шрифтів для дизайнерських задач / Automated system for
generating font pairs for design tasks**

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КН-42
К. О. Панасевич

Науковий керівник:
к.т.н., доцент І. В. Турченко

Кваліфікаційну роботу
допущено до захисту:

«___» _____ 20___ р.

Завідувач кафедри

_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки освітньо-професійна
програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«_____» _____ 20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ПАНАСЕВИЧ Катерині Олександрівні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

**Автоматизована система генерації типографічних пар шрифтів для
дизайнерських задач / Automated system for generating font pairs
for design tasks**

керівник роботи к.т.н., доцент І.В. Турченко

затверджений наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

- проаналізувати предметну область, дослідити основні принципи та критерії сумісності шрифтів, виконати огляд існуючих сервісів, визначити їхні переваги та недоліки, здійснити постановку задачі проєктування;

- спроектувати архітектуру системи та розробити алгоритмічне та інформаційне забезпечення системи;

- реалізувати серверний модуль семантичного аналізу, який використовує алгоритми обробки природної мови (NLP);

- розробити клієнт-серверний застосунок для взаємодії користувача з системою за допомогою REST API;

- використати механізм контейнеризації програмного рішення, для його подальшого розгортання та масштабованості.

- провести тестування функціональності системи та перевірити коректність сформованих рекомендацій.

5. Перелік графічного матеріалу у роботі

- Функціональна модель IDEF0 автоматизованої системи
- Схема алгоритму роботи автоматизованої системи
- UML-діаграма станів графічного інтерфейсу користувача
- UML-діаграма класів серверного рівня

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент _____ К.О. Панасевич
підпис

Керівник роботи _____ к.т.н., доцент І.В. Турченко
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Автоматизована система генерації типографічних пар шрифтів для дизайнерських задач» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 61 сторінку і містить 15 ілюстрацій, 4 додатки та 29 використаних джерел.

Метою кваліфікаційної роботи є проектування і реалізація автоматизованої системи генерації типографічних пар шрифтів для дизайнерських задач, яка надаватиме користувачам персоналізовані рекомендації щодо поєднання шрифтів на основі семантичного аналізу тексту та заданих параметрів оформлення.

Об'єктом дослідження є процеси отримання, зберігання, обробки, передачі та використання інформації автоматизованого формування типографічних пар шрифтів.

Результатом роботи є автоматизована система з клієнт-серверною архітектурою, що включає модуль семантичного аналізу, який використовує алгоритми обробки природної мови (NLP). Функціонал системи поєднує аналіз текстових описів, генерацію гармонійних пар та локальне збереження історії.

Розроблено автоматизовану систему для генерації типографічних пар шрифтів із інтеграцією модуля семантичного аналізу. Це відкриває можливості для його широкого практичного застосування у сфері графічного дизайну та суміжних напрямів, пов'язаних із створенням візуального контенту. Інструмент може бути корисний в дизайнерській діяльності, щоб скоротити час на прийняття рішень щодо вибору гарнітур та забезпечити більш обґрунтований підхід до створення візуально гармонійного контенту.

Ключові слова: АВТОМАТИЗОВАНА СИСТЕМА, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, ТИПОГРАФІЧНІ ПАРИ ШРИФТІВ, ОБРОБКА ПРИРОДНОЇ МОВИ, ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР, ДИЗАЙНЕРСЬКІ ЗАДАЧІ.

ANNOTATION

Qualification work on the topic «Automated system for generating font pairs for design tasks» for obtaining a bachelor's degree in the specialty 122 «Computer Science» of the educational program «Computer Science» is written on 61 pages and contains 15 illustrations, 4 appendices, and 29 references.

The purpose of the qualification work is to design and implementation of an automated system for generating font pairs for design tasks, which will provide users with personalized recommendations for combining fonts based on semantic text analysis and specified design parameters.

The object of research is the processes of obtaining, storing, processing, transmitting, and using information for the automated generation of font pairs.

The result of the work is an automated system with a client-server architecture, which includes a semantic analysis module that uses natural language processing (NLP) algorithms. The functionality of the system combines the analysis of text descriptions, the generation of harmonious pairs and local storage of history.

An automated system has been developed for generating font pairs with an integrated semantic analysis module. This opens up opportunities for its wide practical application in the field of graphic design and areas related to the creation of visual content. The tool can be useful in design activities to reduce the time required for decision-making regarding typeface selection and to provide a more informed approach to creating visually harmonious content.

Keywords: AUTOMATED SYSTEM, PERSONALIZED RECOMMENDATION, FONT PAIRINGS, NATURAL LANGUAGE PROCESSING, FONT PAIR GENERATION, DESIGN TASKS.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка задачі проєктування системи генерації типографічних пар шрифтів	10
1.1 Аналіз предметної області та функціональної структури системи.....	10
1.2 Огляд і аналіз існуючих програмних аналогів	15
1.3 Постановка задачі та вимоги до проєктованої системи	21
2 Проєктування алгоритмічного та інформаційного забезпечення автоматизованої системи генерації типографічних пар шрифтів	23
2.1 Функціональна структура та архітектура системи	23
2.2 Алгоритмічне забезпечення системи генерації типографічних пар шрифтів для дизайнерських задач.....	25
2.3 Інформаційне забезпечення системи генерації типографічних пар шрифтів для дизайнерських задач.....	29
3 Реалізація автоматизованої системи генерації типографічних пар шрифтів для дизайнерських задач.....	34
3.1 Програмна реалізація системи	34
3.2 Інтерфейс користувача системи генерації типографічних пар шрифтів	39
3.3 Контейнеризація, розгортання та тестування системи.....	44
Висновки	46
Список використаних джерел	48
Додаток А Порівняльний аналіз функціональних можливостей існуючих аналогів.....	51
Додаток Б Програмний код семантичного аналізу тексту	52
Додаток В Результати функціонального та модульного тестування системи.....	54
Додаток Г Копія опублікованих результатів.....	56

ВСТУП

Стрімкий розвиток цифрових технологій та зростання обсягів інформаційного контенту зумовлюють потребу у вдосконаленні засобів візуального представлення інформації. Одним із важливих аспектів такого представлення є типографіка, яка відіграє ключову роль у забезпеченні читабельності, структурованості та естетичної привабливості текстового контенту.

Розвиток типографіки є результатом тривалої еволюції письма та технологій відтворення тексту, що сприяло появі великої кількості шрифтових гарнітур і варіантів їх накреслень. Водночас значне різноманіття доступних шрифтів суттєво ускладнює процес вибору та поєднання гарнітур у межах одного дизайнерського проєкту, оскільки кількість можливих комбінацій є надзвичайно великою.

Традиційний підбір типографічних пар здебільшого ґрунтується на досвіді та суб'єктивних оцінках дизайнера, що потребує значних часових витрат і не завжди гарантує отримання оптимального результату. У зв'язку з цим актуальним є застосування автоматизованих інформаційних систем, здатних формалізувати процес підбору шрифтів, систематизувати їхні типографічні характеристики та формувати рекомендації щодо найбільш сумісних поєднань.

Розроблення автоматизованої системи генерації типографічних пар шрифтів дозволить підвищити ефективність дизайнерської діяльності, скоротити час на прийняття рішень щодо вибору гарнітур та забезпечити більш обґрунтований підхід до створення візуально гармонійного контенту. Тому тема кваліфікаційної роботи є актуальною.

Метою кваліфікаційної роботи є проєктування і реалізація автоматизованої системи генерації типографічних пар шрифтів для дизайнерських задач, яка надаватиме користувачам персоналізовані рекомендації щодо поєднання шрифтів на основі семантичного аналізу тексту та заданих параметрів оформлення.

Досягнення поставленої мети дозволить зменшити когнітивне навантаження на дизайнера та автоматизувати підбір гармонійних типографічних комбінацій. Це сприятиме підвищенню якості візуальної комунікації в дизайн-проєктах.

Для досягнення визначеної мети необхідно вирішити такі завдання:

- проаналізувати предметну область, дослідити основні принципи типографіки та критерії сумісності шрифтів у дизайнерських проєктах;
- виконати огляд існуючих сервісів для автоматизованого підбору шрифтових пар, визначити їхні переваги та недоліки, здійснити постановку задачі проєктування;
- спроектувати архітектуру системи з розділенням на клієнтську та серверну частини;
- розробити алгоритмічне та інформаційне забезпечення системи;
- реалізувати серверний модуль аналізу тексту та генерації рекомендацій із використанням мови програмування Python;
- розробити клієнтський вебінтерфейс для взаємодії користувача з системою з використанням бібліотеки React;
- забезпечити інтеграцію клієнтської та серверної частин за допомогою REST API;
- реалізувати механізм збереження та повторного використання рекомендованих шрифтових пар;
- провести тестування функціональності системи та перевірити коректність сформованих рекомендацій.

Об'єктом дослідження є процеси отримання, зберігання, обробки, передачі та використання інформації автоматизованого формування типографічних пар шрифтів.

Предметом дослідження є методи та алгоритми генерації типографічних пар шрифтів на основі семантичного аналізу тексту з використанням методів обробки природної мови для дизайнерських задач.

Практичне значення отриманих результатів: основним призначенням інструменту є допомога в дизайнерській діяльності шляхом скорочення часу на прийняття рішень щодо вибору гарнітур та забезпечення більш обґрунтованого підходу до створення візуально гармонійного контенту.

Також розроблений програмний продукт відкриває можливості його

широкого практичного застосування в різних сферах: вебдизайн та інформаційні технології, цифрові медіа та видавнича справа, маркетинг та брендинг, створення контенту для соціальних мереж, освітня сфера, UI/UX дизайн та автоматизація роботи креативних агенцій.

Результати роботи опубліковано в матеріалах міжнародної науково-практичної конференції студентів і молодих вчених «ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами», м. Тернопіль, 21–22 травня 2026 р. (додаток Г).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ПРОЄКТУВАННЯ СИСТЕМИ ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ

1.1 Аналіз предметної області та функціональної структури системи

Із розвитком Всесвітньої павутини (World Wide Web) типографіка перестала бути виключно складовою друкарської сфери та набула вагомого значення у сучасному інформаційному просторі. Вона є невід'ємною частиною повсякденного життя, оскільки присутня в архітектурному середовищі, друкованих виданнях, медіа, елементах візуальної ідентифікації та інших сферах графічного оформлення інформації.

Розвиток типографіки є результатом багатовікової еволюції письма та технологій відтворення тексту, що суттєво вплинули на формування та різноманіття шрифтових форм [1]. З одного боку, це розширює типографічний інструментарій дизайнера, а з іншого – ускладнює вибір та поєднання шрифтів у межах одного проєкту.

У більшості графічних рішень застосовується декілька гарнітур: для заголовків, основного тексту, підписів або допоміжних елементів. Таке поєднання забезпечує формування візуальної ієрархії, структурованість інформації та композиційну узгодженість.

Разом із тим, підбір шрифтових пар є складним багатофакторним процесом. Він передбачає врахування формальних характеристик шрифтів, зокрема наявності або відсутності зарубок, товщини штрихів, пропорцій символів, контрастності, ритміки та стилістичної узгодженості [2]. Недостатній рівень контрасту може призвести до візуальної монотонності, тоді як надмірна відмінність – до порушення цілісності сприйняття тексту.

У типографіці, при генерації типографічних пар шрифтів, найбільш важливими є такі принципи:

- 1) читабельність – легкість розпізнавання окремих символів і знаків;
- 2) візуальна ієрархія – організація текстових елементів за рівнем важливості за допомогою розміру, накреслення, кольору та розташування;

3) контрастність – створення помітних відмінностей між елементами тексту для покращення сприйняття та акцентування уваги;

4) узгодженість – дотримання єдиного стилю оформлення в межах документа або проєкту;

5) пропорційність – гармонійне співвідношення розмірів, відступів і текстових блоків;

6) типографічна сумісність – гармонійне поєднання різних гарнітур і накреслень у межах одного дизайну;

7) відповідність контексту – вибір шрифтів та оформлення відповідно до тематики, цільової аудиторії та призначення контенту.

З огляду на значну кількість доступних гарнітур і варіацій їх накреслень кількість можливих комбінацій є надзвичайно великою. Це ускладнює вибір та суттєво збільшує часові витрати дизайнера на пошук оптимального рішення.

Підбір типографічних пар доцільно розглядати як окремий процес у межах дизайнерської діяльності, який має визначену послідовність дій та може бути формалізований засобами інформаційних технологій. Цей процес охоплює етапи аналізу текстового контенту, визначення вимог до оформлення та генерацію оптимальної шрифтової пари.

З позиції теорії інформаційних систем у структурі цього процесу доцільно виокремити дві взаємопов'язані складові: операційну (транзакційну) та аналітичну.

Операційна складова забезпечує виконання дій, пов'язаних із введенням текстових даних, вибором параметрів генерації, отриманням інформації про гарнітури та формуванням множини можливих комбінацій.

Аналітична складова передбачає оцінювання типографічних характеристик шрифтів, визначення рівня контрастності та стилістичної узгодженості, а також формування фінальної пари на основі встановлених критеріїв сумісності.

З огляду на багатофакторність оцінювання шрифтів і значну кількість можливих комбінацій, процес підбору типографічних пар потребує значних часових витрат і часто залежить від суб'єктивного досвіду дизайнера. У практичних умовах це ускладнює оперативне прийняття рішень, особливо під час

роботи з великою кількістю макетів, різними стилями оформлення або обмеженими термінами виконання завдань.

У таких умовах доцільним є застосування автоматизованих інформаційних систем, здатних забезпечити структурований підхід до підбору шрифтових пар. Використання засобів автоматизації дозволяє систематизувати типографічні характеристики гарнітур, звужити множину можливих варіантів відповідно до заданих параметрів та сформувати рекомендації щодо сумісних поєднань.

Автоматизація не замінює творчого підходу, проте мінімізує рутинні операції та підвищує обґрунтованість вибору завдяки формалізації критеріїв оцінювання. Сучасні інформаційні системи можуть поєднувати алгоритмічну обробку даних та інтелектуальні методи аналізу для підтримки процесів прийняття рішень [3].

Для формалізації описаного процесу та подальшого проєктування системи доцільно побудувати дерево функцій, яке відображає основні етапи та підетапи автоматизованої генерації типографічних пар шрифтів (рисунок 1.1).

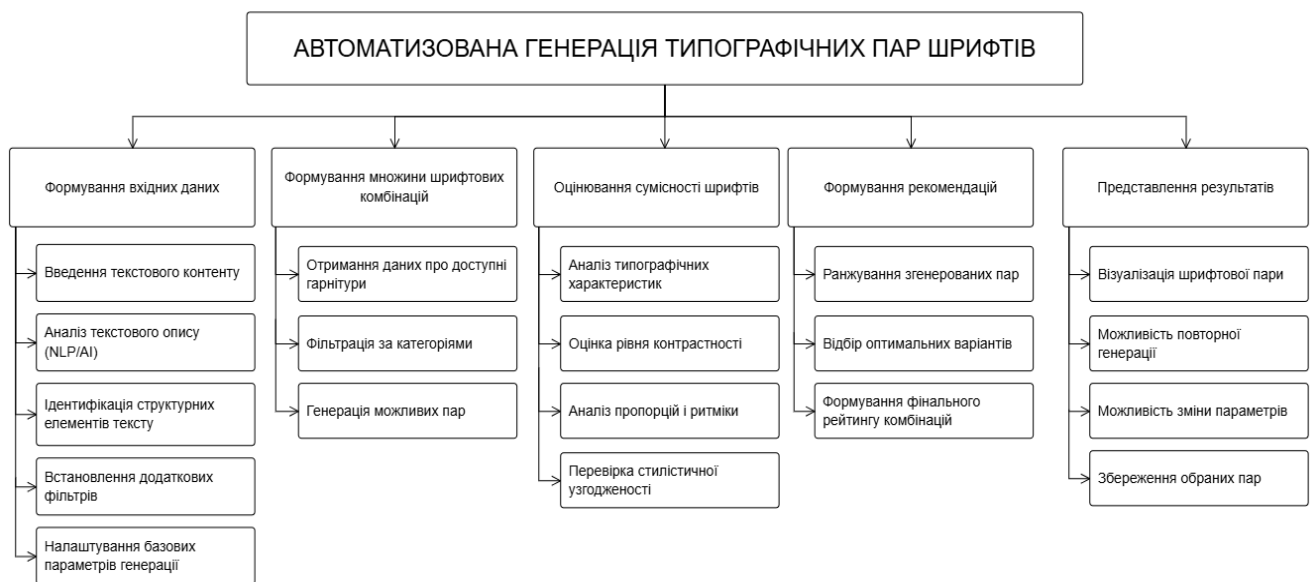


Рисунок 1.1 – Дерево функцій автоматизованої системи генерації типографічних пар шрифтів

Наведена структура дерева функцій автоматизованої системи генерації типографічних пар шрифтів (див. рисунок 1.1) включає низку взаємопов'язаних

функціональних блоків:

а) формування вхідних даних:

1) введення текстового контенту (користувач вводить текст або опис проєкту, який використовується для подальшого аналізу);

2) аналіз текстового опису (NLP/AI) (виконується семантична обробка тексту для визначення стилістичних та емоційних характеристик);

3) ідентифікація структурних елементів тексту (розподіл проаналізованого масиву інформації на логічні складові, зокрема заголовки та основний текст);

4) встановлення додаткових фільтрів (задання специфічних вимог до гарнітур, цільової аудиторії чи стильового спрямування);

5) налаштування базових параметрів генерації (визначення типу проєкту, наприклад: вебсайт, застосунок, постер тощо, та загального режиму роботи системи);

б) формування множини шрифтових комбінацій:

1) отримання даних про доступні гарнітури (завантаження та обробка інформації про шрифти з підключеної бібліотеки);

2) фільтрація за категоріями (відбір шрифтів відповідно до встановлених користувачем параметрів);

3) генерація можливих пар (формування первинних комбінацій шрифтів для ролей заголовка та основного тексту);

в) оцінювання сумісності шрифтів:

1) аналіз типографічних характеристик (оцінювання категорії шрифту, варіантів накреслення, популярності та інших параметрів);

2) оцінка рівня контрастності (перевірка відповідності ваги та насиченості шрифтів для забезпечення візуальної ієрархії);

3) аналіз пропорцій і ритміки (визначення композиційної узгодженості ширини символів та загальної візуальної структури);

4) перевірка стилістичної узгодженості (оцінювання відповідності сформованої пари типу проєкту та заданому емоційному контексту).

г) формування рекомендацій:

1) ранжування згенерованих пар (обчислення інтегральної оцінки кожної комбінації на основі алгоритму скорингу);

2) відбір оптимальних варіантів (виокремлення найбільш релевантних шрифтових пар, що набрали найвищий бал);

3) формування фінального рейтингу комбінацій (упорядкування результатів для підготовки їх до видачі користувачу);

д) представлення результатів:

1) візуалізація шрифтової пари (відображення результатів у вигляді інтерактивних макетів);

2) можливість повторної генерації (створення нових варіантів із частково або повністю зміненими параметрами);

3) можливість зміни параметрів (редагування вхідного тексту та налаштувань безпосередньо на етапі перегляду);

4) збереження обраних пар (додавання вдалих комбінацій до персональної колекції користувача).

Таким чином, підбір типографічних пар шрифтів можна розглядати як структурований технологічний процес у межах дизайнерської діяльності, що охоплює послідовність взаємопов'язаних дій – від аналізу текстового контенту та визначення вимог до оформлення до генерації фінальної шрифтової пари. Поділ цього процесу на операційну (транзакційну) та аналітичну складові дозволяє чітко розмежувати етапи оброблення вхідних даних і етапи оцінювання та прийняття рішень. Такий підхід забезпечує системність, керованість та можливість подальшої формалізації процесу засобами інформаційних технологій.

Запропонована функціональна структура відображає логіку роботи майбутньої системи та створює основу для подальшого проєктування її архітектури, алгоритмічного забезпечення та моделі даних, що розглядатимуться у наступних розділах роботи.

1.2 Огляд і аналіз існуючих програмних аналогів

У зв'язку з необхідністю автоматизації процесу підбору типографічних пар шрифтів доцільно проаналізувати наявні програмні рішення, що частково або повністю покривають необхідну функціональність. Дослідження таких сервісів дає змогу оцінити їхні функціональні можливості, виявити переваги та обмеження, а також сформулювати вимоги до системи, що проєктується.

Серед сучасних програмних рішень, які пропонують автоматизований або напівавтоматизований підбір шрифтових пар, доцільно виокремити такі аналоги: Fontjoy, Fontpair та Monotype AI Font Pairing. Для визначення сильних і слабких сторін цих рішень варто проаналізувати кожен із зазначених сервісів з урахуванням низки визначених критеріїв.

Fontjoy [4] – це онлайн-сервіс для автоматизованого підбору шрифтових комбінацій на основі методів машинного навчання. Основне завдання інструмента полягає у швидкому формуванні сумісних гарнітур для трьох рівнів ієрархії: заголовка, підзаголовка та основного тексту. Алгоритм системи налаштований на пошук візуального балансу між подібністю та контрастом шрифтів, а результати генерації миттєво відображаються у вебінтерфейсі на прикладі реального текстового макета.

Для глибшого розуміння роботи сервісу доцільно розглянути його ключові особливості:

- джерело шрифтів – використовується бібліотека Google Fonts, що забезпечує відкритий доступ до гарнітур та можливість їхнього подальшого застосування без додаткового ліцензування;
- мовна адаптація – відсутня; сервіс не передбачає фільтрацію шрифтів за мовою;
- механізм формування пари на базі AI – підбір здійснюється за допомогою алгоритмів глибокого навчання, які автоматично виділяють візуальні ознаки шрифтів та обчислюють оптимальний рівень контрасту між ними;
- взаємодія з текстом – введення власного тексту не є обов'язковим завдяки

наявності стандартного зразка, але користувач може змінити його для оцінки вигляду пари в реальному контексті;

- гнучкість налаштувань – результат оновлюється миттєво; передбачено регулювання контрасту та можливість фіксації одного зі шрифтів. Проте розширені фільтри (категорії, стиль, настрій) відсутні;

- контекстний аналіз тексту – відсутній; інструмент не розуміє емоційного настрою введеного тексту і підбирає шрифти лише, орієнтуючись на геометричні форми та стиль гарнітур;

- представлення результату – поєднання відображається у вигляді текстового макета з поділом на заголовок, підзаголовок та основний текст;

- збереження даних – відсутнє; оскільки сервіс працює без авторизації користувачів, можливість збереження історії підборів або створення власної бібліотеки відсутня.

- доступність – платформа є безкоштовною та доступною без реєстрації;

На рисунку 1.2 наведено інтерфейс сервісу Fontjoy, який демонструє область попереднього перегляду, елементи керування генерацією шрифтової комбінації та регулювання рівня контрасту.

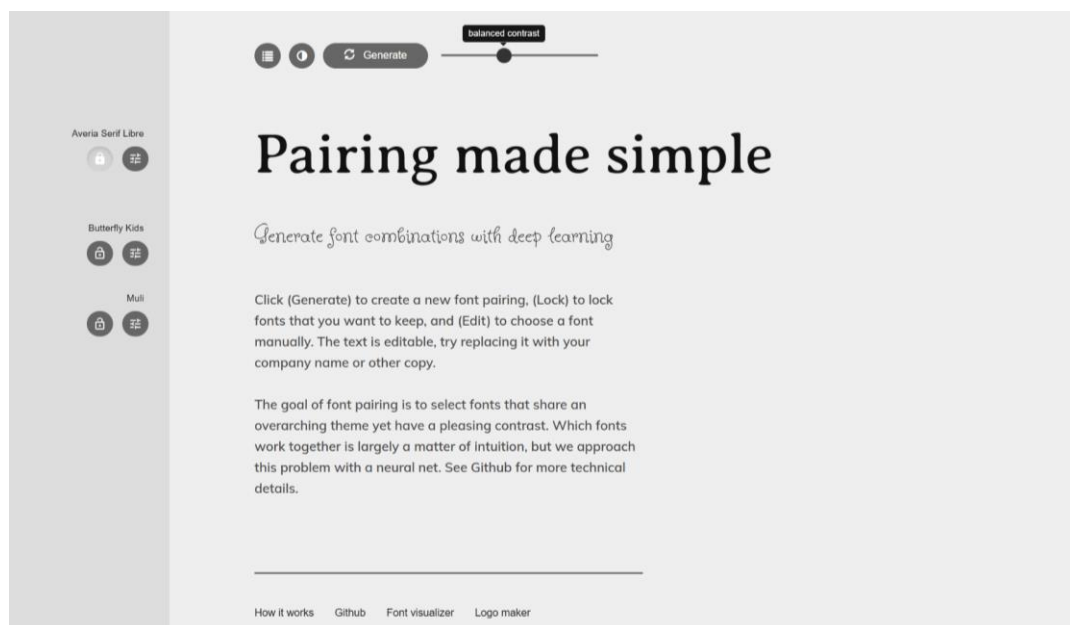


Рисунок 1.2 – Інтерфейс сервісу Fontjoy для автоматичного підбору шрифтових пар

Представлений інтерфейс підтверджує орієнтацію сервісу на простоту використання та швидке отримання результату без складних налаштувань.

Fontpair [5] – це комплексний онлайн-сервіс для підбору шрифтових комбінацій, що забезпечує їх візуалізацію у реальних макетах на основі готових дизайнерських рішень. Платформа пропонує екосистемний підхід, поєднуючи підбір шрифтових пар із кольоровими палітрами та іконками для швидкого прототипування інтерфейсів. Оскільки основний акцент зроблено на демонстрації того, як обрані гарнітури виглядають у типових елементах інтерфейсу, сервіс використанням готових дизайнерських пресетів, що дозволяє базово оцінити читабельність та візуальну сумісність шрифтів у різних функціональних блоках.

Для глибшого розуміння роботи сервісу доцільно розглянути його ключові особливості:

- джерело шрифтів – аналогічно до сервісу Fontjoy, платформа базується на відкритих шрифтових бібліотеках (зокрема Google Fonts), які інтегруються у сервіс для демонстрації широкого спектра шрифтових комбінацій;
- мовна адаптація – відсутня; фільтрації шрифтів за мовною підтримкою (кирилиця/латиниця) не реалізовані, що обмежує врахування мовного контексту при підборі гарнітур;
- механізм формування комбінацій на базі AI – відсутній; поєднання шрифтів представлені у вигляді заздалегідь підібраних дизайнерських рішень без застосування алгоритмічної генерації;
- взаємодія з текстом – надаються стандартні зразки; підтримується редагування тексту в режимі попереднього перегляду, з можливістю змінювати текст безпосередньо в макетах, що дозволяє частково оцінити читабельність та візуальний баланс шрифтової пари;
- гнучкість налаштувань – зміна рішень відбувається через вибір макета відповідно до області застосування, функцію випадкового підбору та ручний вибір гарнітур для різних текстових ролей; підтримується можливість підбору шрифтів за категоріями та фіксація одного із них для пошуку сумісної пари; водночас інструменти для детального коригування технічних параметрів відсутні;

- контекстний аналіз тексту – відсутній; підбір гарнітур здійснюється на основі випадкового або послідовного перегляду варіантів без урахування змісту контенту та його функціонального призначення;
- представлення результату – шрифтові пари відображаються у вигляді статичних макетів, що дозволяє візуально оцінити їх вигляд у контексті реального дизайну;
- збереження даних – для авторизованих користувачів підтримується хмарне створення власних бібліотек і колекцій шрифтових пар, що забезпечує організацію та повторне використання обраних рішень у межах дизайн-проектів; для гостьового доступу передбачено можливість збереження результатів шляхом копіювання коду або завантаження файлів з повним набором відповідних макетів.
- доступність – сервіс має безкоштовну версію з базовим функціоналом; розширені можливості доступні лише у платній версії;

На рисунку 1.3 наведено інтерфейс сервісу Fontpair, у якому відображається обрана шрифтова пара та її візуальне представлення в контексті.

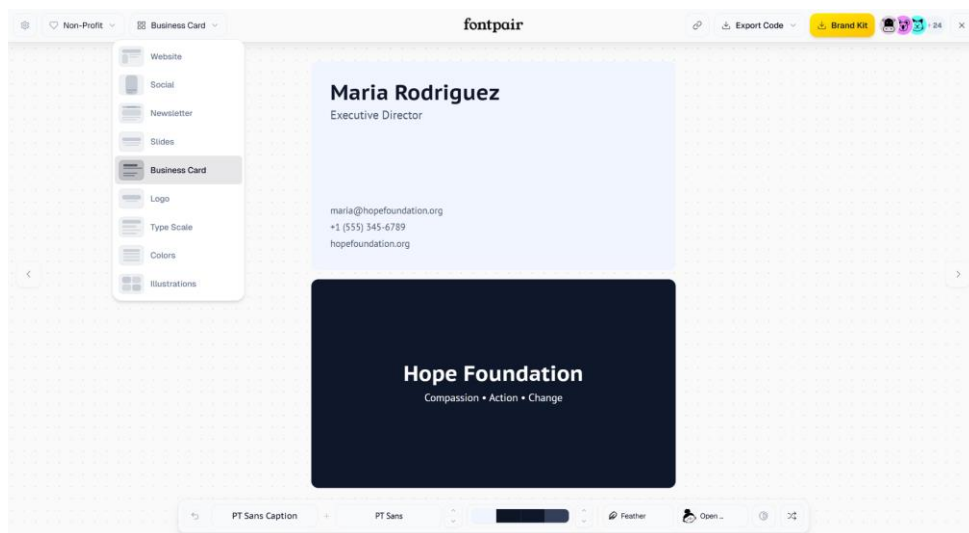


Рисунок 1.3 – Інтерфейс сервісу Fontpair з підбором шрифтових пар

Візуалізований інтерфейс демонструє акцент на відображенні результату в реальному дизайнерському наповненні, що сприяє оцінці практичного застосування та естетичної сумісності гарнітур у структурі готового продукту.

Monotype AI Font Pairing [6] – це професійний інструмент для генерації

шрифтових пар, інтегрований у глобальну екосистему Monotype. Платформа орієнтована на дизайнерів і бренди, які працюють із комерційними гарнітурами та потребують обґрунтованих типографічних рішень. На відміну від простих онлайн-генераторів, сервіс поєднує алгоритми штучного інтелекту з експертним досвідом, що дозволяє автоматизувати пошук візуально сумісних пар у межах великих бібліотек.

Для глибшого розуміння роботи потужного інструменту, доцільно розглянути його ключові особливості:

- джерело шрифтів – на відміну від попередніх аналогів, сервіс використовує власну ліцензовану бібліотеку, що містить масштабну базу професійних та комерційних гарнітур;

- мовна адаптація – обмежена; сервіс орієнтований переважно на комерційні гарнітури з розширеною мовною підтримкою, однак інструменти для фільтрації шрифтів за мовними наборами відсутні;

- механізм формування комбінацій на базі AI – базується на аналізі топографічних характеристик: пропорцій, контрасту штрихів, структури та висоти малих літер, що дозволяє створювати пари з чітко визначеним рівнем візуальної гармонії або заданим ступенем контрасту;

- взаємодія з текстом – надаються стандартні зразки, проте підтримується можливість повного редагування контенту в реальному часі. Користувач може вводити власний текст безпосередньо в інтерфейсі попереднього перегляду, що є важливим для перевірки читабельності та візуального балансу пари в конкретних заголовках чи текстових блоках;

- гнучкість налаштувань – сервіс дозволяє змінювати розмір шрифтів, закріплювати гарнітури для подальшого підбору гармонійної пари, а також вручну знімати конкретний шрифт на альтернативи з переліку запропонованих системою. Всі вдалі комбінації можна зберегти до списку обраних для подальшого використання;

- контекстний аналіз тексту – реалізований як внутрішня логіка системи, що базується на метаданих шрифтової бази. Алгоритм автоматично розпізнає

стилістичну приналежність та візуальний характер обраної гарнітури, пропонуючи комбінації, які зберігають єдиний контекст і цілісність візуального повідомлення без необхідності додаткового налаштування з боку користувача;

- представлення результату – аналогічно до платформи Fontjoy, поєднання відображається у вигляді текстового блоку з поділом на заголовок та основний текст. Це дозволяє оцінити вигляд пари та за потреби проєкспериментувати гармонійність підбору;

- збереження даних – можливість додавати обрані комбінації до списку «Pairs you liked». Це дозволяє зберігати вдалі знахідки протягом поточної сесії без авторизації, що забезпечує швидкий доступ до сформованої колекції для подальшого порівняння чи вибору остаточного варіанту.

- доступність – сервіс інтегрований у екосистему Monotype є безкоштовним, проте доступ до повного функціоналу шрифтів може бути обмежений умовами ліцензування.

На рисунку 1.4 наведено інтерфейс сервісу Monotype AI Font Pairing із відображенням рекомендованої шрифтової пари та елементів керування.

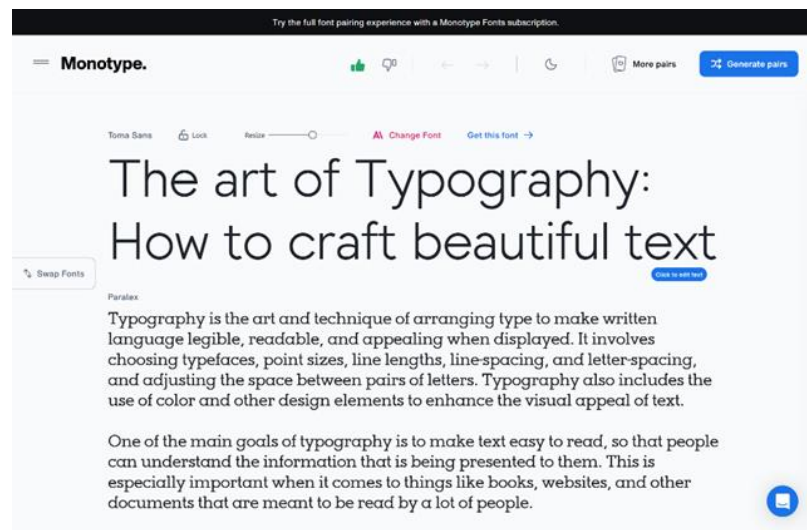


Рисунок 1.4 – Інтерфейс сервісу Monotype AI Font Pairing для автоматизованого підбору пар шрифтів

Наведений інтерфейс свідчить про професійну спрямованість платформи та зручне зосередження інструментів керування гарнітурами (зміна кегля, фіксація

стилів, ручна заміна елементів пари) безпосередньо в робочому вікні інструменту.

Провівши огляд існуючих рішень, варто систематизувати функціональні можливості розглянутих сервісів у порівняльну таблицю (додаток А).

Аналіз розглянутих сервісів (Fontjoy, Fontpair та Monotype AI Font Pairing) свідчить, що існуючі рішення автоматизують технічну сторону підбору шрифтів, проте мають суттєве обмеження – вони не враховують зміст тексту. Алгоритми цих платформ фокусуються на візуальному контрасті або популярності пар, що змушує користувача самостійно шукати відповідність між дизайном та емоційним посилом контенту. Це підтверджує необхідність розробки системи, яка би поєднувала досвід автоматизації візуальної ієрархії із семантичним аналізом. Завдяки такому підходу шрифти будуть краще відповідати тематиці контенту та специфіці конкретного типу проєкту, що значно спростить роботу дизайнера.

1.3 Постановка задачі та вимоги до проєктованої системи

Аналіз існуючих платформ підбору шрифтових пар показав, що сучасні рішення частково автоматизують процес формування типографічних комбінацій, проте мають низку обмежень. Більшість із них працюють на основі чітко визначених правил або статистичних співвідношень між категоріями шрифтів та не враховують зміст самого тексту [7]. Крім того, відсутність інструментів для налаштування параметрів генерації обмежує можливості користувача, що знижує гнучкість використання таких інструментів у творчих проєктах [28].

У зв'язку з цим виникає потреба у створенні автоматизованої системи, яка поєднуватиме алгоритмічний аналіз типографічних характеристик із семантичним аналізом введеного тексту та врахуванням контекстних параметрів.

Метою роботи є проєктування та реалізація системи для автоматизованої генерації шрифтових пар.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати предметну область, дослідити основні принципи типографіки та критерії сумісності шрифтів у дизайнерських проєктах;

- виконати огляд існуючих сервісів для автоматизованого підбору шрифтових пар, визначити їхні переваги та недоліки, здійснити постановку задачі проєктування;
- спроектувати архітектуру системи;
- розробити алгоритмічне та інформаційне забезпечення системи;
- реалізувати серверний модуль (із використанням мови програмування Python) семантичного аналізу тексту із застосуванням методів обробки природної мови (англ. Natural Language Processing, NLP) для визначення стилістичних характеристик та формування рекомендацій щодо вибору категорій шрифтів для заголовка і основного тексту;
- розробити клієнтський вебінтерфейс для взаємодії користувача з системою з використанням бібліотеки React;
- забезпечити інтеграцію клієнтської та серверної частин за допомогою REST API;
- реалізувати механізм збереження та повторного використання рекомендованих шрифтових пар;
- передбачити декілька режимів роботи системи: генерацію на основі текстового аналізу, підбір за параметрами користувача та комбінований режим, що інтегрує обидва підходи.
- провести тестування функціональності системи та перевірити коректність сформованих рекомендацій.

У результаті виконання поставлених завдань має бути створена система, що забезпечує адаптивний, логічно структурований та гнучкий підбір типографічних пар. Це дозволить зменшити час на прийняття дизайнерських рішень щодо візуального оформлення тексту та підвищить ефективність роботи з контентом у творчих проєктах. Автоматизована система генерації типографічних пар шрифтів називатиметься GLYPHA.

2 ПРОЄКТУВАННЯ АЛГОРИТМІЧНОГО ТА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ

2.1 Функціональна структура та архітектура системи

З огляду на поставлену задачу та визначені вимоги до системи генерації типографічних пар шрифтів для дизайнерських задач, у межах її реалізації було обрано клієнт-серверний підхід [8].

Загальна концептуальна схема взаємодії архітектурних рівнів представлена на рисунку 2.1.

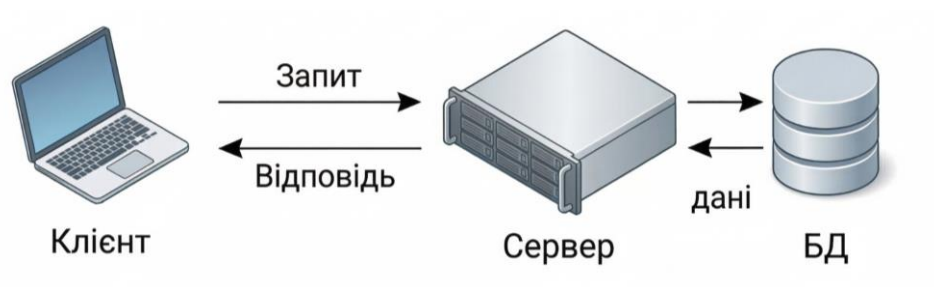


Рисунок 2.1 – Загальна схема тривірневої клієнт-серверної архітектури

З точки зору автоматизації цей підхід є найбільш релевантним, оскільки дозволяє ефективно реалізувати всі поставлені функціональні завдання.

Структурна організація системи базується на розподілі обов'язків між двома ключовими компонентами, що забезпечують усі етапи обробки інформації.

Першим таким є користувацький інтерфейс, іншими словами клієнтська частина. Ця частина виступає зовнішньою оболонкою системи, відповідає за взаємодію з користувачем, збір вхідних даних і відображення результатів. Його функція полягає у зборі вхідних даних: від вільного текстового опису проєкту до вибору конкретних технічних параметрів, таких як мова чи рівень контрасту тощо. Після перевірки коректності введеної інформації, клієнтська частина формує запит і передає його далі для обчислень.

Іншим ключовим компонентом виступає аналітична складова або ж серверна частина. Виступає вона ядром розроблюваної системи і зосереджена тут вся логіка

функціонування. А саме, лінгвістичний розбір тексту, пошук семантичних зв'язків та математичне ранжування шрифтових пар. Сервер обробляє отриманий запит, звертається до бази даних і повертає готовий результат користувачу.

Для відображення повної загальної структури та логіки взаємодії компонентів використано модель у стандарті IDEF0 (рисунок 2.2) [28], яка демонструє як користувацький інтерфейс та серверна частина взаємодіють для обробки запитів користувачів, виконання лінгвістичного аналізу, семантичного пошуку та формування персоналізованих рекомендацій шрифтових пар.

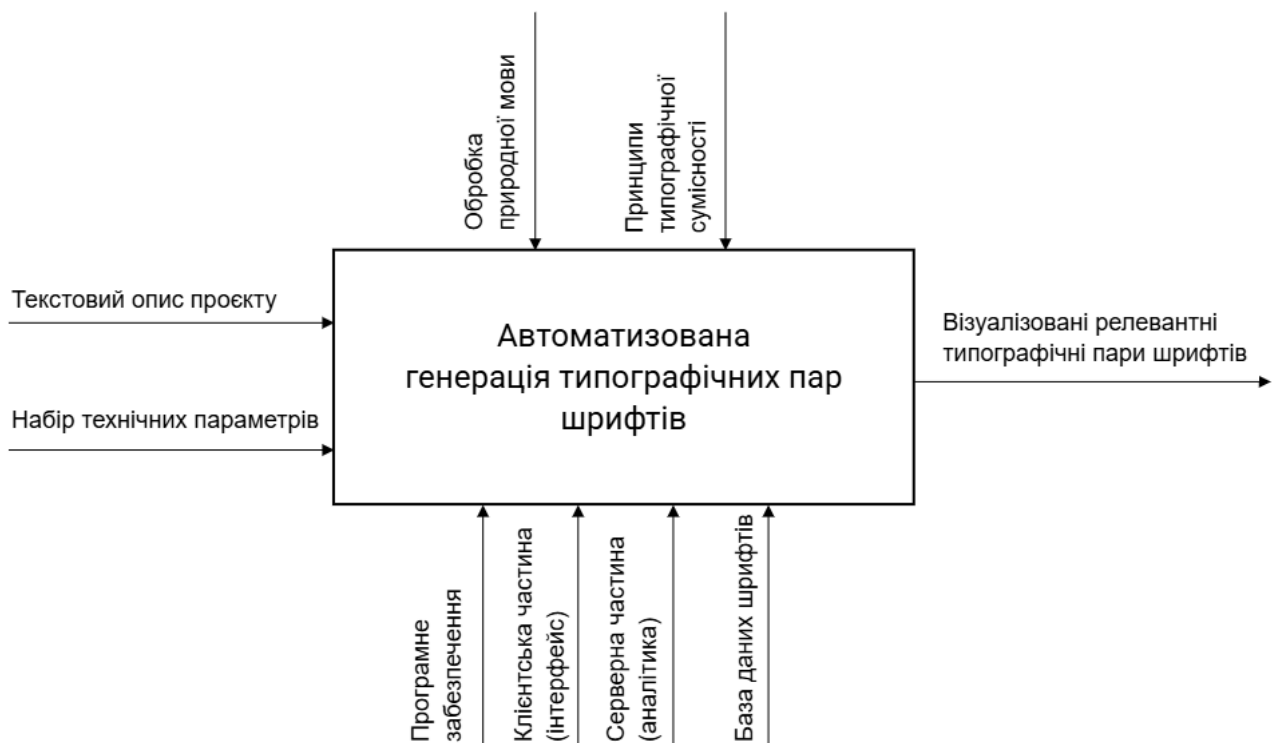


Рисунок 2.2 – Функціональна модель автоматизованої системи
(IDEF0 діаграма)

Відповідно до представленої діаграми, центральним процесом системи є «Автоматизована генерація типографічних пар шрифтів». Цей блок інтегрує в собі всі логічні процеси системи та забезпечує перетворення необробленого запиту користувача у готове візуальне рішення. Його функціонування базується на взаємодії вхідних даних, засобів управління та технічних механізмів.

Вхідні дані визначаються текстовим описом проєкту та набором технічних параметрів. Це та інформація, яка за ініціації користувача сприяє пошуку і звужує

підбір за мовною підтримкою та візуальною контрастністю майбутньої пари.

Управління основним процесом визначається встановленими правилами. Сюди відносяться алгоритм обробки природної мови та встановлені принципи типографічної сумісності. Алгоритм аналізує сенс запиту з допомогою вилучення ключових значень із тексту, а правила відносно шрифтів обмежують вибір лише тими парами, які гармонійно поєднуються.

Технічні механізми у процесі є ресурси, що забезпечують виконання функцій системи. До таких інструментів належать: програмне забезпечення, клієнтська частина (інтерфейс), серверна частина (аналітика) та база даних шрифтів. У межах клієнт-серверної архітектури клієнтська частина відповідає за первинну обробку введених даних, а серверна частина – за складні математичні розрахунки. Важливу роль тут відіграє база даних, яка надає технічні характеристики шрифтів [28].

Кінцевим результатом роботи основного процесу є сформовані релевантні шрифтові пари, у зручному для візуального оцінювання вигляді. Вони проходять систему фільтрації та ранжування і готові до безпосереднього перегляду користувачем.

Розроблена структура діаграми демонструє, як обрана клієнт-серверна архітектура забезпечує цілісність процесу – від отримання сирих даних до видачі інтелектуальної рекомендації.

Такий розподіл функцій забезпечує стабільну роботу системи, де зовнішня оболонка залишається зручною для користувача, а всі складні розрахунки ізолюються і виконуються окремо на стороні сервера. Це створює надійну основу для подальшого опису алгоритмічного забезпечення та інформаційного наповнення системи.

2.2 Алгоритмічне забезпечення системи генерації типографічних пар шрифтів для дизайнерських задач

Алгоритм розглядається не лише як послідовність дій, а як основа логіки функціонування сучасних інтелектуальних систем. Саме алгоритмічна структура

визначає особливості роботи кожної системи та формує її унікальність.

У розроблюваній системі алгоритм визначає процес обробки запиту користувача та формування рекомендацій типографічних пар шрифтів на основі семантичного аналізу тексту. З математичної точки зору система базується на трансформації неструктурованого текстового запиту у векторне представлення [9] з подальшим розрахунком відповідності шрифтових пар. Фактично даний підхід впроваджує у систему елемент машинного навчання (Machine Learning), адже дозволяє робити рекомендації персоналізованими та семантично релевантними.

В основі роботи програмного комплексу лежить використання метод обробки природної мови NLP із бібліотекою spaCy [10], що дозволяє комп'ютерам розуміти та обробляти людську мову. Наразі використовується модель, орієнтована на україномовну аудиторію, а саме uk_core_news_lg [11]. Це велика попередньо натренована нейронна мережа, навчена на мільйонах українських реченнях та здатна розпізнавати зв'язки між словами та витягувати їх семантичне значення.

Щоб краще зрозуміти логіку функціонування алгоритму підбору типографічних пар шрифтів на основі семантичного аналізу, на рисунку 2.3 відтворено усі етапи обробки запиту користувача – від зчитування тексту та параметрів до формування персоналізованих комбінацій шрифтів.

Відповідно до представленої блок-схеми, алгоритм розпочинається з етапу надходження тексту природною мовою від користувача. Спочатку вхідний текст проходить перевірку мови запиту. Якщо промпт написаний українською мовою, запускається повний цикл NLP-обробки. В іншому випадку система переходить безпосередньо до етапу технічної фільтрації.

Попередня обробка тексту включає наступні кроки: токенизація – поділ речення на окремі слова; лематизація – зведенню слів до їх початкової морфологічної форми для усунення надлишковості мовних форм; а також видалення стоп-слів – ігнорування службових слів, які не несуть семантичного навантаження для дизайну.



Рисунок 2.3 – Схема алгоритму роботи автоматизованої системи генерації типографічних пар шрифтів

На етапі векторизації тексту нормалізовані слова перетворюються на багатовимірні числові вектори. Попередньо навчена модель дозволяє алгоритму оцінювати семантичну відстань між поняттями у векторному просторі, де близькі за змістом слова мають подібні координати.

Далі на аналізі релевантності шрифтів система обчислює косинусну схожість між вектором користувацького запиту та еталонними векторами стилів, закріплених за категоріями шрифтів у базі даних SQLite. Коефіцієнт схожості обчислюється за формулою косинусної відстані [12].

Результатом цього процесу є перетворення неструктурованої текстової інформації на набір кількісних коефіцієнтів (ваг), що визначають пріоритетність тих чи інших категорій шрифтів.

На основі розрахованих коефіцієнтів схожості система формує попередній список відповідних шрифтів. Цей список проходить через технічні фільтри з інтерфейсу, які користувач обрав вручну (наприклад, підтримка кирилиці, рівень контрасту або вікова категорія). Ранжування відбувається шляхом сортування: шрифти з найвищим показником математичної схожості та повною відповідністю фільтрам опиняються на перших позиціях.

Після того, як фінальний список сформовано, система автоматично виконує запис у базу даних SQLite [13]. У таблицю вносяться дані про текст запиту, обрані параметри та ID підібраних шрифтів. Це необхідно для забезпечення персоналізації та можливості перегляду попередніх результатів користувачем.

На заключному етапі сервер відправляє сформований пакет даних (у форматі JSON [14]) назад на клієнтську частину (React [15]). Інтерфейс обробляє ці дані та динамічно генерує картки з прев'ю шрифтових пар, підтягуючи стилі безпосередньо з Google Fonts API для наочного відображення користувачу [16].

Ілюстрована схема алгоритму демонструє принцип роботи системи та взаємозв'язки між основними компонентами.

Таким чином, рівень лінгвістичної обробки забезпечує перетворення якісних характеристик дизайну в кількісні параметри, необхідні для автоматизованого пошуку та ранжування гарнітур у базі даних. Хоч в розроблюваній системі

підтримується лише україномовна модель, проте можливість подальшого масштабування системи і розширення на інші мови доцільна. Цей етап алгоритмізації є ключовим для переходу від суб'єктивного опису проєкту до об'єктивного підбору гарнітур.

2.3 Інформаційне забезпечення системи генерації типографічних пар шрифтів для дизайнерських задач

Інформаційне забезпечення є фундаментальною складовою будь-якої інформаційної системи і визначає структуру, організацію та спосіб зберігання даних, необхідних для безперебійного функціонування компонентів. Воно включає різні типи даних – структуровані, напівструктуровані та неструктуровані – механізми їх обробки, зберігання і захисту [17], а також опис вхідних і вихідних даних, що забезпечує цілісність, актуальність, безпеку та узгодженість інформації.

У межах проєктованої автоматизованої системи кожен тип даних відповідає за окремий функціональний сегмент роботи та виконує визначену роль у процесі обробки інформації.

Структуровані дані є чітко організованими у вигляді реляційних таблиць бази даних і призначені для зберігання інформації про користувачів, їхні налаштування та історію взаємодії із системою. Чітко визначені типи даних для кожного поля у таблицях дозволяє забезпечити цілісність, коректність інформації та легкий доступ.

Напівструктуровані дані характеризуються наявністю часткової структури та відповідають за обмін інформацією між компонентами системи та зовнішніми інформаційними ресурсами через API. У проєктованій системі до цього типу належать JSON-відповіді сервісів, що містять метадані шрифтів, а також конфігураційні файли застосунку, що задають параметри його налаштування. Використання цього типу даних забезпечує стандартизований та зручний формат для обробки, зберігання та інтеграції з іншими компонентами системи.

Неструктуровані типи даних зберігаються у вільній формі і є основою для інтелектуального аналізу. До них належать тексти, введені користувачем у

природній мові, векторні представлення слів із лінгвістичної моделі та графічні файли. Саме для обробки цього типу даних застосовується математичне забезпечення, що витягує семантичний зміст із несистематизованої інформації. Використання цього типу дозволяє системі більш гнучко аналізувати інформацію та забезпечує розширені можливості для персоналізації та адаптації контенту.

Основою інформаційного наповнення системи є інтеграція з відкритим реєстром метаданих Google Fonts API. Використання відкритого вихідного коду дозволяє автоматично отримувати актуальні дані про шрифти: назви, категорії, атрибути (товщина, стиль, підтримка мов) та інші характеристики шрифтів.

Обмін даними з реєстром здійснюється у напівструктурованому форматі JSON (JavaScript Object Notation), що є загальноприйнятим форматом обміну даними. Кожна відповідь від API містить ієрархічну структуру об'єктів, де ключі відповідають технічним характеристикам шрифту, а значення – конкретним параметрам (наприклад, "family": "Roboto", "category": "sans-serif"). Завдяки такій інтеграції система має доступ до структурованої і стандартизованої інформації, що спрощує формування бази даних шрифтів і забезпечує точність та узгодженість даних, а також прискорює їх програмну обробку.

Додатковим джерелом даних виступає користувач, що взаємодіє із системою та ініціює процес підбору пар гарнітур. Вхідна інформація від користувача може надходити у двох форматах: довільного текстового опису проєкту українською/англійською мовами або ж у вигляді фіксованих параметрів, що обираються через графічний інтерфейс. Введений текст не має фіксованого формату і може містити як прямі вказівки на стиль («елегантний ресторан»), так і емоційні характеристики («веселе дитяче свято»). Через елементи інтерфейсу користувач може задати такі параметри проєкту:

- мову (UA/EN) – виступає як фільтр і при виборі автоматично обмежує вибірку лише тими гарнітурами з Google Fonts API, що підтримують кириличний або латинський набори символів;
- тип проєкту (вебсайт, мобільний додаток, книга, глянцеви́й журнал, постер, візитка);

- вік цільової аудиторії (універсальна, діти, підлітки та молодь, дорослі, старше покоління);
- настрої проєкту (нейтральний, офіційний, грайливий, елегантний, технологічний);
- бажаний рівень контрастності між шрифтами (збалансований, різкий, м'який).

Система передбачає гнучкість взаємодії, дозволяючи використовувати ці джерела даних як окремо, так і в комбінації для досягнення необхідного результату.

Важливу роль в ланцюжку структурування і збереження даних відіграє реляційна база даних SQLite. Використання цієї вбудованої СУБД не потребує окремого серверу для роботи та дозволяє зберігати інформацію у вигляді одного файлу, що спрощує інтеграцію, розгортання та забезпечує швидкий доступ до даних.

Для кращого розуміння, на рисунку 2.4 представлено схему реляційної бази даних, що відображає основні сутності бази даних та зв'язки між ними.

ER-модель бази даних формують такі таблиці:

- «Users» – основа персоналізації, забезпечує ідентифікацію та безпеку доступу. Таблиця не просто зберігає email та password_hash для автентифікації, а виступає точкою входу, до якої прив'язаний весь індивідуальний досвід користувача;
- «History» фіксує історію взаємодії користувача із системою, зберігаючи результати інтелектуального підбору. Вона зберігає не лише фінальні назви гарнітур (header_font, body_font), а й повний набір вхідних параметрів: від визначеного емоційного настрою (mood) до технічних фільтрів проєкту (age, contrast, genMode). Це дозволяє користувачу миттєво переглядати історію своїх запитів без необхідності повторного введення параметрів.
- «Favorites» зберігає обрані комбінації шрифтів і пов'язана з користувачем через зовнішній ключ. Така структура забезпечує персоналізований доступ до збережених пар гарнітур, отриманих із Google Fonts;
- «Guest_history» зберігає історію взаємодії неавторизованих користувачів.

Завдяки унікальному ідентифікатору `guest_id` система дозволяє працювати з інструментом без попередньої авторизації, зберігаючи результати підбору шрифтових пар для їх подальшого перенесення до облікового запису після реєстрації;

– «Analytics_events» призначена для логування системних подій. В цій сутності зберігається інформація про активність користувачів, типи обраних проєктів та популярні налаштування мови (language), що є необхідним для подальшого аналізу ефективності алгоритмів та вдосконалення моделей машинного навчання.

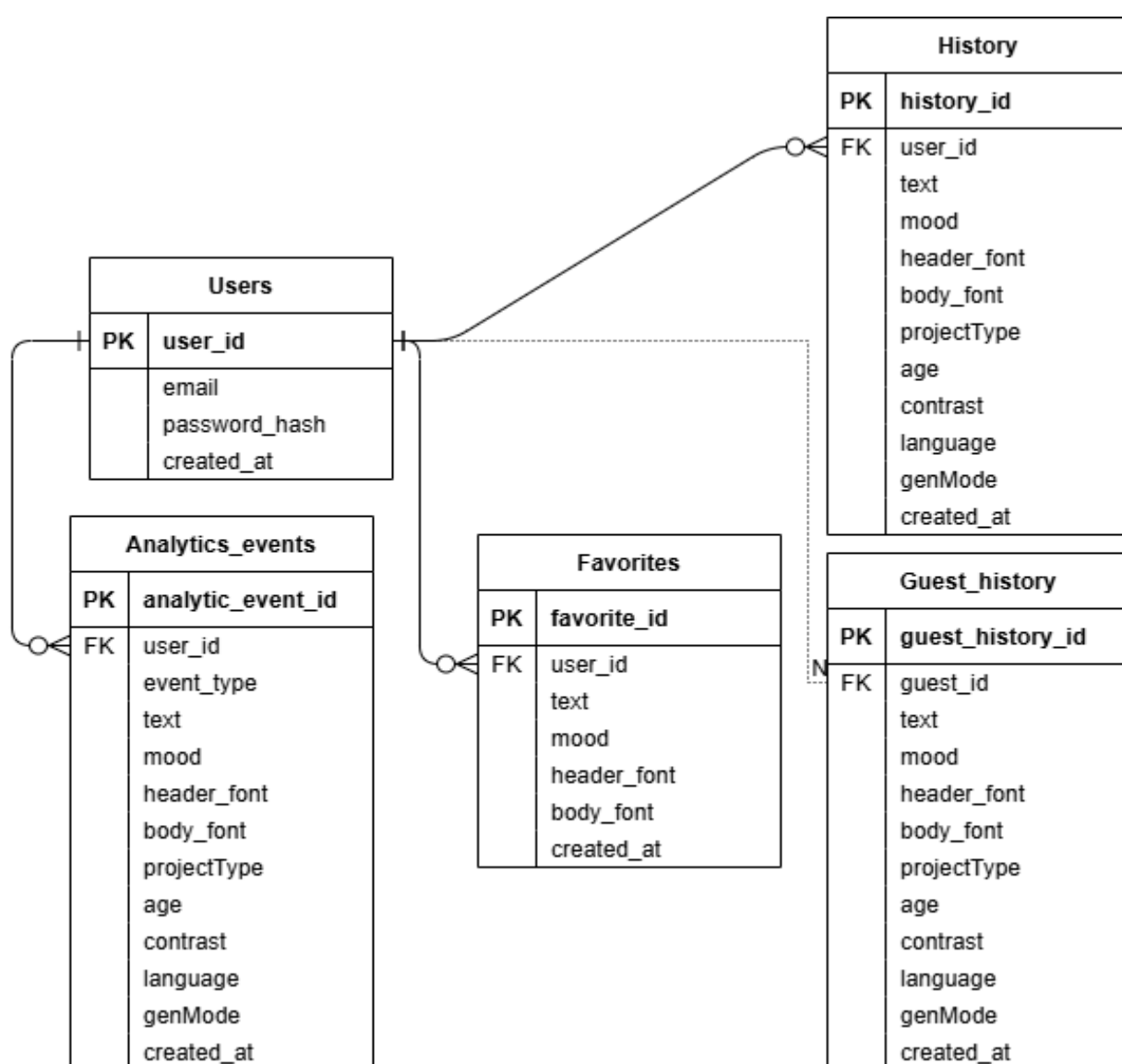


Рисунок 2.4 – ER-модель бази даних системи

Розроблена логічна модель є повністю нормалізованою і демонструє організацію даних користувачів, їхні персональні добірки шрифтів та історію взаємодії з системою, а також як ці об'єкти пов'язані між собою для забезпечення цілісності та узгодженості інформації.

Насамкінець, інформаційне забезпечення системи є невід'ємною частиною проєктованої системи, оскільки воно об'єднує неструктуровані користувацькі запити, напівструктуровані дані з зовнішнього API та структуроване реляційне сховище. Такий інформаційний фундамент повністю задовольняє вимоги до цілісності та масштабованості системи, готуючи базу для її подальшої програмної реалізації.

3 РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ ДЛЯ ДИЗАЙНЕРСЬКИХ ЗАДАЧ

3.1 Програмна реалізація системи

На основі спроектованих рішень програмна реалізація системи передбачає створення основної бізнес-логіки. Для цього використано трирівневу клієнт-серверну архітектуру [18], яка включає рівень користувацького інтерфейсу, серверну частину для обробки запитів та базу даних для зберігання інформації.

Рівень користувацького інтерфейсу відповідає за взаємодію користувача з системою та відображення даних. Реалізований він як односторінковий вебзастосунок (Single Page Application) [19] із використанням середовища Node.js для керування роботою менеджера пакетів NPM.

Каркас інтерфейсу користувача формують стандартні мови розмітки та стилізації – HTML і CSS. За допомогою HTML створено структуру сторінки та типографічних макетів, а CSS використовуючи технологію Variables, забезпечив візуальне оформлення цих структур, надаючи сучасний та естетичний вигляд.

Для створення динамічного та компонентного інтерфейсу застосовано бібліотеку React, що працює на базі мови JavaScript.

Використання React дозволило розділити інтерфейс на окремі частини (форми введення, кнопки, блок попереднього перегляду), що спрощує розробку та пришвидшує роботу сайту.

Керування станом і реактивне оновлення реалізовано за допомогою функцій – хуків, таких як useState (для збереження введеного тексту чи обраних параметрів) та useEffect (для виконання дій при зміні стану програми).

JavaScript [20] у свою чергу відповідає за бізнес-логіку на клієнтському рівні, включаючи алгоритм автоматичного підбору та фільтрації шрифтів на основі даних від сервера. Для збереження результату підбору пар гарнітур використано бібліотеку html2canvas [21], яка дозволяє перетворювати створений у браузері макет попереднього перегляду у файл PNG.

Оскільки користувацька частина системи побудована як динамічний

вебзастосунок, для моделювання її поведінки у часі використано діаграму станів автоматизованої системи генерації типографічних пар шрифтів (рисунок 3.1).

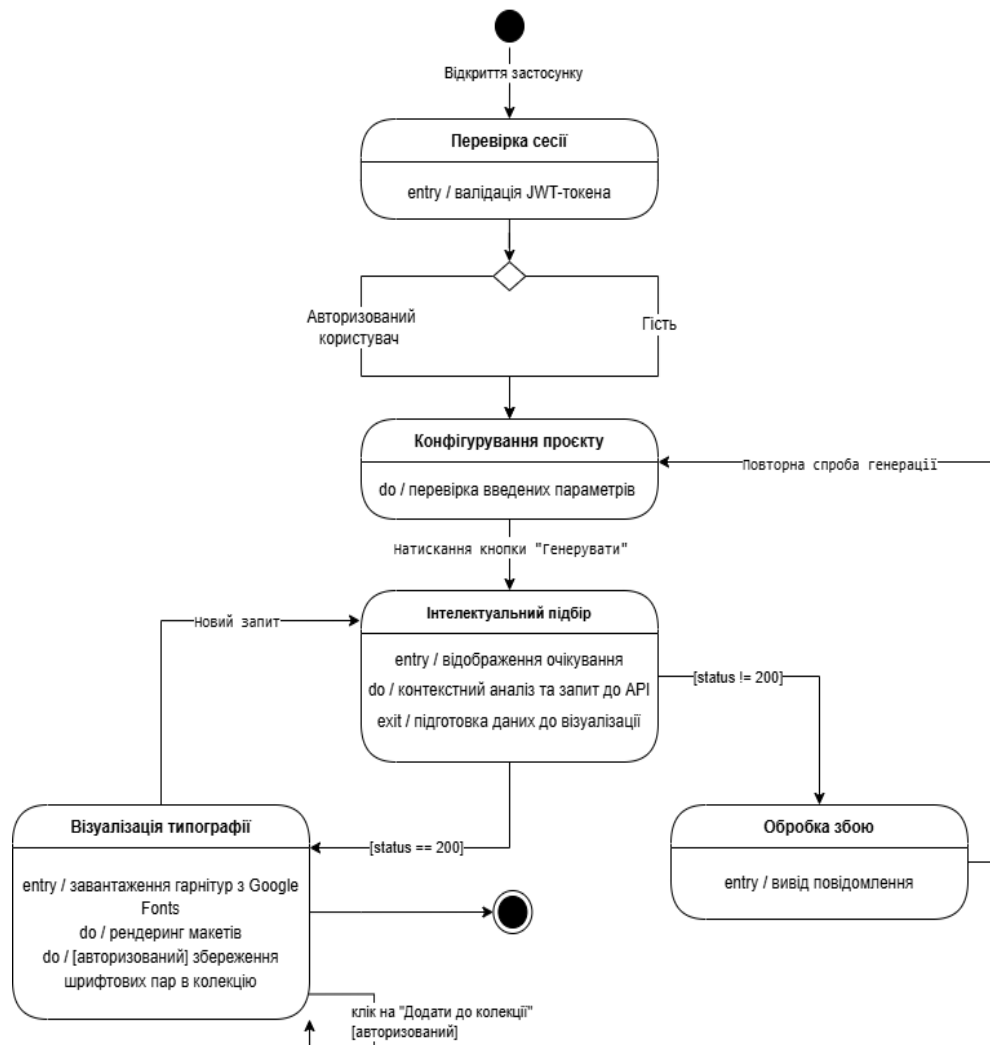


Рисунок 3.1 – UML-діаграма станів графічного інтерфейсу користувача системи

Розроблена діаграма візуалізує переходи між різними етапами роботи інтерфейсу залежно від подій та відповідей сервера. Процес розпочинається з перевірки сесії за допомогою JWT-токена, що дозволяє визначити тип користувача (гість або авторизований) ще до початку взаємодії із системою.

Основний цикл ініціюється натисканням кнопки «Генерувати», після чого система переходить у стан інтелектуального підбору. На цьому етапі виконується аналіз тексту, формується запит до API та здійснюється підготовка до візуалізації. Залежно від статусу відповіді сервера інтерфейс переходить або до відображення підібраних шрифтів, або до стану обробки помилки.

Основні функції, такі як генерація, перегляд шрифтових пар і завантаження макетів, доступні всім відвідувачам сервісу. Натомість розширені можливості, зокрема збереження шрифтів у персональну колекцію, доступні лише після авторизації. Це дозволяє персоналізувати роботу системи для кожного користувача.

Отож, рівень представлення або ж рівень користувацького інтерфейсу формує зовнішній вигляд автоматизованої системи, забезпечує високу швидкість відгуку інтерфейсу та виступає сполучною ланкою між запитами й інтелектуальною бізнес-логікою сервера, забезпечуючи інтерактивну взаємодію користувача з даними та бізнес-процесами.

Серверна частина виступає посередником між інтерфейсом користувача та базою даних, забезпечуючи виконання запитів та аналітичну обробку інформації. Програмна реалізація цього рівня імплементована за допомогою високорівневої мови програмування Python. Вибір цієї технології обумовлений її ефективністю у роботі з обробкою даних, реалізації бізнес-логіки та алгоритмів, зокрема в області обробки природної мови, а також широким набором спеціалізованих бібліотек.

Разом із Python використовується фреймворк Flask [22] для стабільної роботи сервера та обробки REST API запитів. Без їх спільного використання система не могла б функціонувати належним чином, координувати роботу модулів та забезпечувати взаємодію з клієнтською частиною.

Основна інтелектуальна функція системи реалізована за допомогою бібліотеки spaCy із використанням моделі `uk_core_news_lg`. Вона забезпечує семантичний аналіз тексту користувача, обчислює вектори слів і визначає стилістичний настрій промпту через порівняння близькості векторів. Програмний код представлений у додатку Б.

Особлива увага у межах серверної частини надається безпеці передачі даних та ідентифікація користувачів, що реалізована за допомогою стандарту JWT (JSON Web Token) із бібліотекою PyJWT. Даний етап дозволяє генерувати захищені токени для сесій, а саме пропонує надійний захист інформації без необхідності зберігати складні сесії безпосередньо на сервері.

Також на цьому рівні реалізовано крос-доменну взаємодію з використанням бібліотеки Flask-CORS між клієнтською (React) та серверною частинами, запобігаючи помилкам доступу між різними доменами.

Для кращого розуміння статичної структури серверної частини системи генерації типографічних пар шрифтів та взаємозв'язків між основними програмними сутностями на рисунку 3.2 розроблено діаграму класів.

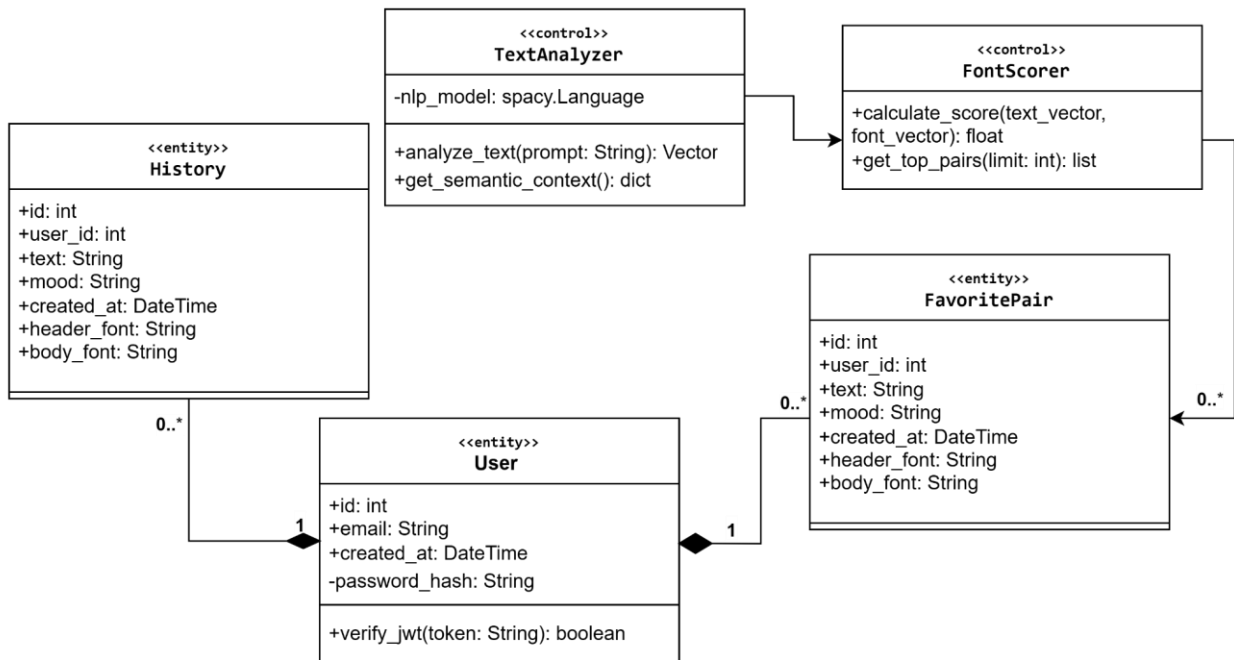


Рисунок 3.2 – UML-діаграма класів серверного рівня

На представленій діаграмі класи розподілені за функціональним призначенням. Керуючі класи TextAnalyzer та FontScorer відповідають за контекстний аналіз тексту та алгоритмічний підбір шрифтових пар. Результат обчислення рейтингу готові шрифтові пари передаються класам-сутностям для запису в базу даних.

Класи-сутності User, History та FavoritePair призначені для збереження даних, що мають зберігатися постійно. Зв'язок між користувачем та його даними реалізовано через відношення композиції із спільним життєвим циклом: історія запитів і збережені пари існують лише в межах профілю та видаляються разом із ним. Кратність цього зв'язку становить 1 до 0..*, що дозволяє зберігати необмежену кількість результатів для кожного авторизованого користувача.

Отож, рівень логіки додатку, або ж серверна архітектура є центральною складовою автоматизованої системи. Тут відбувається обробка даних, реалізація алгоритмів бізнес-логіки та забезпечення безпеки роботи системи.

Рівень збереження даних є фундаментом для накопичення та структурування інформації в системі Glypha. Його реалізовано на основі реляційної системи управління базами даних SQLite, вибір якої зумовлений її портативністю та простою архітектурою. База даних зберігається у вигляді єдиного файлу `database.db`, що забезпечує централізоване зберігання та високу швидкість виконання запитів.

Взаємодія з базою даних на серверному рівні реалізована через мову запитів SQL, що дозволяє структуровано організувати дані у вигляді таблиць та зв'язків між ними, забезпечуючи цілісність інформації та можливість відновлення попередніх результатів. Структуру бази даних спроектовано у вигляді ER-діаграми у підрозділі 2.3, у межах якої визначено основні сутності системи.

Окрім локальної бази, рівень даних включає взаємодію із зовнішнім хмарним ресурсом Google Fonts API. Ця інфраструктурна складова виступає розподіленим сховищем шрифтових файлів у форматах `.woff2` та `.ttf`. Замість зберігання важких файлів безпосередньо в системі, Glypha генерує динамічні посилання на основі результатів роботи алгоритму підбору, що дозволяє миттєво завантажувати необхідні гарнітури в браузер користувача. Це значно знижує навантаження на локальні ресурси, забезпечує ефективну взаємодію із зовнішнім сервісом та доступ до актуальних версій шрифтів.

Отож, рівень даних ключовий та вирішальний у ланці трирівневої архітектури. База даних забезпечує надійне зберігання інформації та підтримує ефективну роботу всіх інших рівнів системи. Для невеликого навантаження обрана SQLite цілком відповідає вимогам, проте для масштабування чи одночасного запису багатьох користувачів доцільне використання серверної системи керування базами даних.

Таким чином, реалізація системи за допомогою трирівневої клієнт-серверної архітектури забезпечує чітке розмежування функцій між рівнем представлення,

рівнем логіки та рівнем збереження даних. Такий підхід підвищує гнучкість і ефективність розробки, дозволяючи незалежно вдосконалювати кожен рівень без виникнення конфліктів між компонентами.

3.2 Інтерфейс користувача системи генерації типографічних пар шрифтів

Інтерфейс користувача являє собою засіб взаємодії з програмною системою [23]. Ця частина не менш важлива, ніж програмна реалізація, адже від неї залежить зручність роботи, швидкість сприйняття інформації та ефективність виконання завдань [24].

Інтерфейс розроблюваної автоматизованої системи Glypha представлено у вигляді єдиної інтерактивної сторінки, що забезпечує швидку реакцію на дії користувача, зручну навігацію між функціональними елементами та безперервну роботу з даними. Це дозволяє ефективно виконувати завдання та отримувати необхідну інформацію без зайвих перезавантажень чи складних переходів між секціями.

Проектування візуальних рішень та композиційної структури системи виконувалося засобами онлайн-сервісу Figma. Це дозволило детально опрацювати розташування навігаційних блоків, налаштувати колірну гаму та перевірити читабельність шрифтів на фоновому градієнті до початку етапу програмної реалізації. Результатом цього етапу став високодеталізований прототип, який слугував основою для програмної реалізації інтерфейсу користувача та забезпечив стилістичну цілісність фінального вебзастосунку.

Під час завантаження сторінки користувачу відображається вітальний екран, що містить заклик до дії (рисунк 3.3). Візуальний стиль системи базується на м'яких фонових градієнтах та ефекті скла, що створює глибину інтерфейсу та забезпечує високий контраст тексту. Використання темної колірної гами не лише знижує енергоспоживання, а й забезпечує комфортну тривалу взаємодію з платформою.

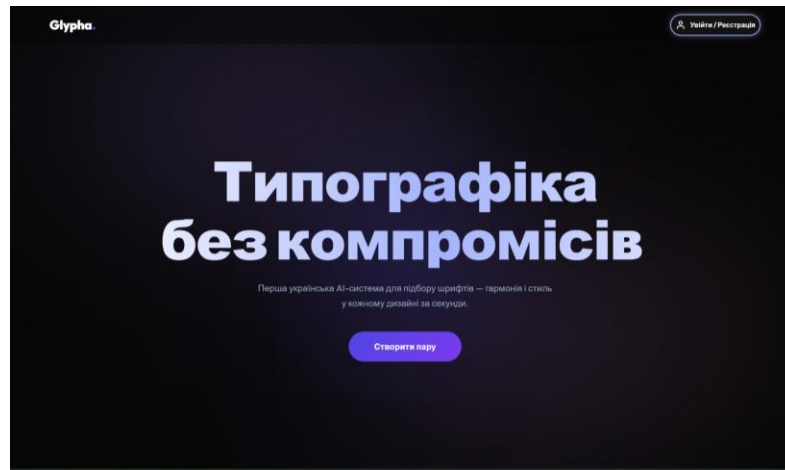


Рисунок 3.3 – Стартовий екран системи Glypha

Далі між вступною частиною та робочою областю інтегровано інформаційний блок, призначений для ознайомлення користувача з принципами роботи системи (рисунок 3.4). Це знижує складність освоєння системи та робить її легкою для початку роботи.

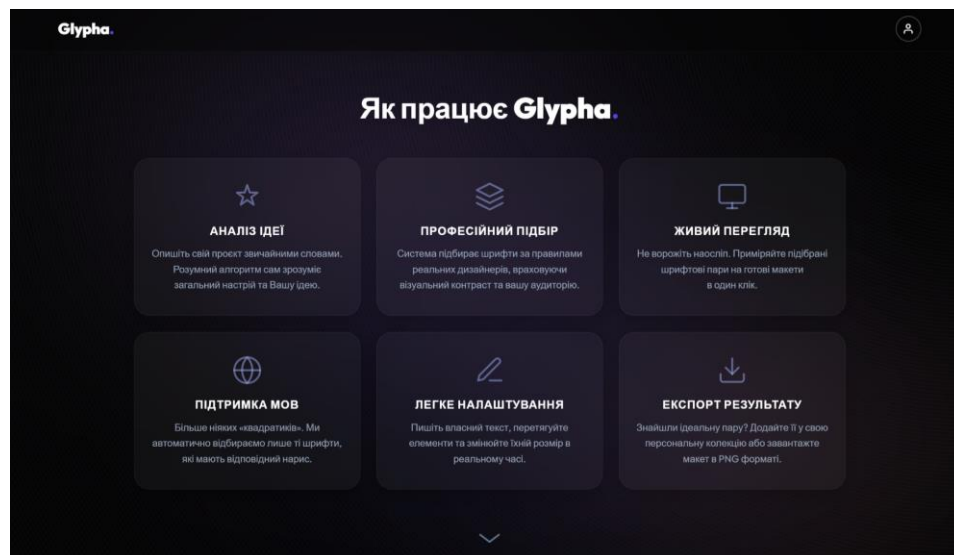


Рисунок 3.4 – Інформаційний блок системи «Як працює Glypha»

Серед ключових переваг системи – контекстуальний підбір шрифтів на основі змісту запиту, миттєвий перегляд результату в реальних макетах та інші можливості, що спрощують роботу користувача.

Основним інструментом взаємодії користувача із системою є дашборд (представлений на рисунку 3.5), який відображає повний цикл генерації

типографічних пар шрифтів – від керування та налаштувань до візуалізації, збереження та редагування результатів. Він реалізований як єдиний функціональний простір, що дозволяє гнучко комбінувати різні джерела вхідних даних, описані на етапі проєктування.

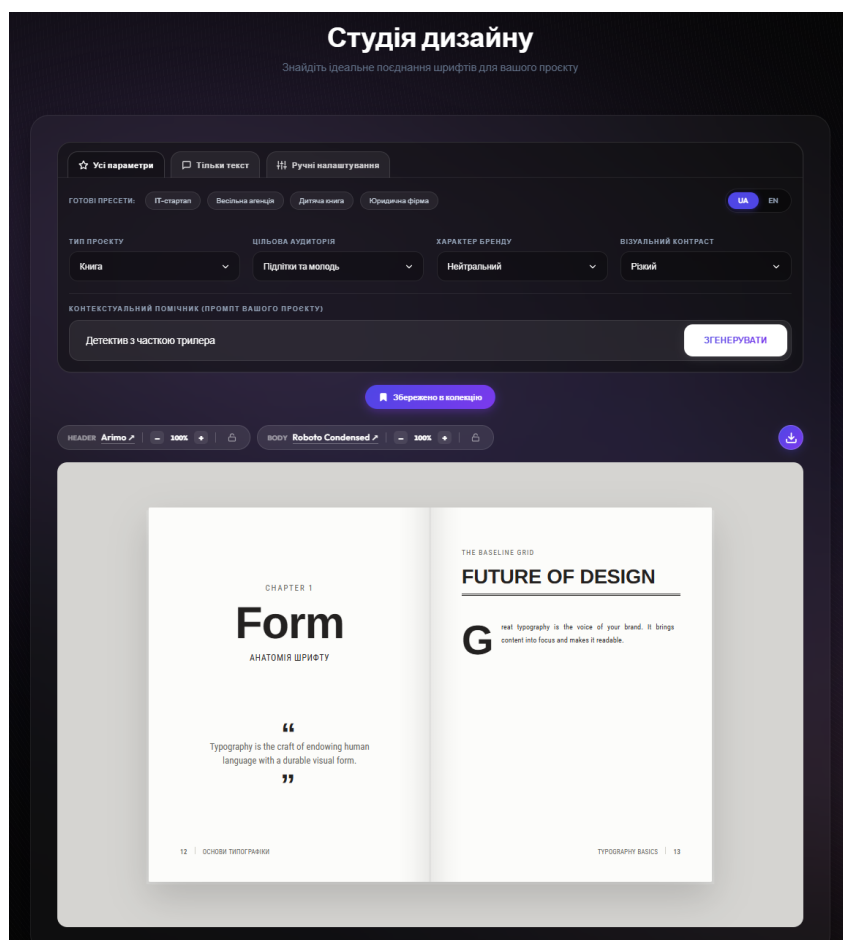


Рисунок 3.5 – Панель керування та блок візуалізації результатів

У межах панелі реалізовано основні функціональні елементи, а саме:

- керування режимами та локалізацією: вибір між комбінованим аналізом, роботою лише з текстом або налаштуванням параметрів, а також адаптація системи під мову проєкту (UA/EN). При зміні мови інтерфейс автоматично перефільтровує доступну базу гарнітур, забезпечуючи коректне відображення символів у макетах;
- панель швидких стилів: набір готових шаблонів для швидкого задання стилістики проєкту;
- інтелектуальне поле запиту: рядок «Контекстний запит» є відправною точкою для генерації. Після введення тексту та натискання кнопки «Згенерувати»

дашборд відображає процес обробки до моменту появи результатів;

– гнучка система фільтрації: випадні списки для уточнення контексту. Користувач може задати тип проєкту, визначити цільову аудиторію, бажаний настрій та рівень контрастності. Це дозволяє регулювати різницю між акцентним та основним шрифтом, а також звужувати пошук для отримання релевантних пар.

Результати підбору інтегруються у контекстні інтерактивні макети: вебсайт, мобільний додаток, глянцева журнал, постер, книга та візитка. Для кожного макета передбачено динамічні посилання на використані гарнітури та кнопку «Завантажити» для експорту візуального результату у форматі PNG.

Розроблена автоматизована система генерації типографічних пар шрифтів Glypha надає розширені функції збереження результатів на двох рівнях. Перший рівень реалізовано для неавторизованих користувачів (рисунок 3.6): історія сесії доступна всім і відображає результати останніх генерацій у межах поточної сесії, що дозволяє швидко повернутися до попередніх варіантів без повторного введення запиту.

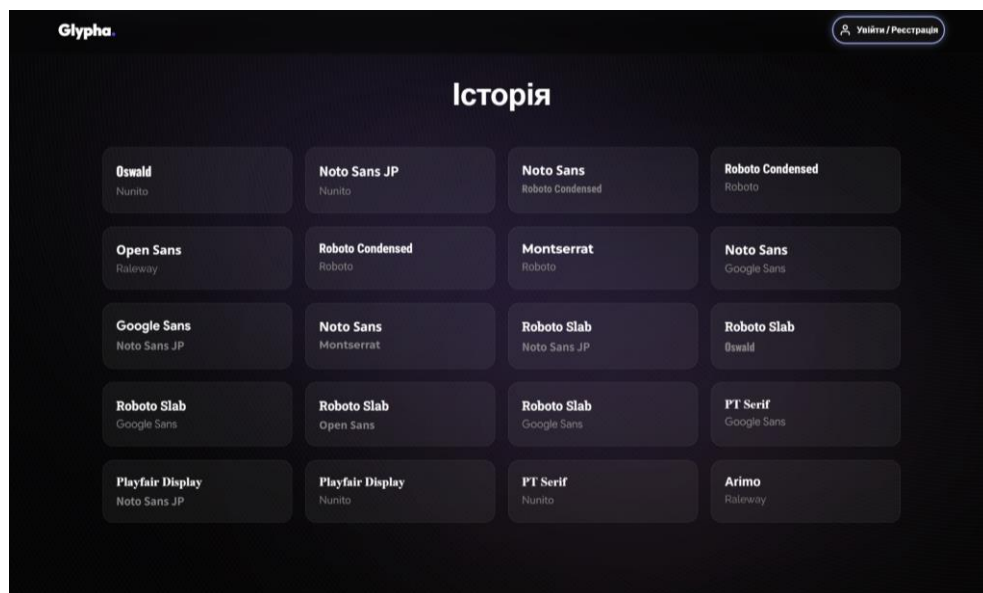


Рисунок 3.6 – Історії сесії для рівня неавторизованого користувача

Другий рівень доступний лише авторизованим користувачам (рисунок 3.7): персональна колекція дозволяє зберігати обрані пари шрифтів у базі даних і використовувати їх при наступних входах у систему.

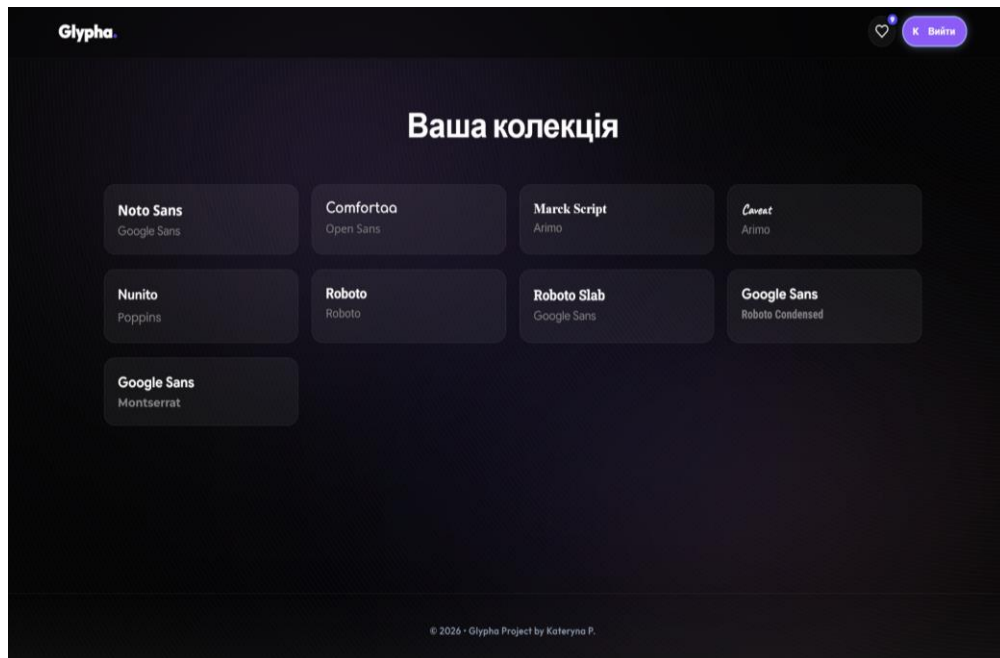


Рисунок 3.7 – Сесія персональної колекції рівня авторизованого користувача

Особливістю реалізованого інтерфейсу є його адаптивність до статусу авторизації, що відкриває доступ до розширеного функціоналу системи. Після входу стандартна кнопка авторизації замінюється на інтерактивну іконку профілю, що підтверджує активну сесію користувача.

Авторизовані користувачі отримують доступ до кнопки «Вподобані пари» з лічильником збережених елементів, розміщеною біля іконки профілю в футері. Крім того, над згенерованим макетом з'являється кнопка «Додати до колекції», яка дозволяє миттєво зберегти обрану пару до персональної бібліотеки в SQLite.

Така динамічна зміна елементів керування забезпечує безперервність робочого процесу та зручний доступ до збережених шрифтових рішень.

Розроблений користувацький інтерфейс поєднує в собі сучасні дизайнерські рішення та високу функціональність. Впровадження системи авторизації дозволило розширити можливості користувачів, надавши їм інструменти для формування персональних колекцій, що робить систему Glypha повноцінним робочим інструментом для дизайнерів.

3.3 Контейнеризація, розгортання та тестування системи

Завершальним етапом програмної реалізації автоматизованої системи Glypha передбачено застосування сучасних підходів для підготовки системи до подальшого використання та перевірку її стабільності і коректності функціонування. Для цього доцільним стало використання технології контейнеризації із застосуванням Docker [25], що забезпечує стабільність роботи системи та спрощує її розгортання.

За допомогою інструментів платформи Docker архітектуру системи було розділено на два функціональні мікросервіси, кожен з яких функціонує у власному ізольованому контейнері [26]:

- клієнтська частина (Frontend) базується на офіційному образі Node. У цьому контейнері виконується збірка React-застосунку, після чого статичні файли обслуговуються за допомогою вебсервера, що гарантує високу швидкість завантаження інтерфейсу;
- серверна частина (Backend) використовує полегшений образ Python 3.10-slim. Ця частина містить середовище виконання Flask, завантажені NLP-моделі для семантичного аналізу та всі необхідні залежності для взаємодії з Google Fonts API.

Для координації роботи обох контейнерів використовується інструмент Docker Compose. Це дозволяє описати конфігурацію всієї системи в одному файлі, автоматизувати створення внутрішньої мережі для обміну даними між сервісами (Python та React), а також налаштувати порти та підключити базу даних SQLite. Такий підхід забезпечує цілісність системи, їх незалежне функціонування та усуває конфлікти залежностей.

Розгортання системи передбачає реалізацію кількох етапів, зокрема застосовується збірка оптимізованих образів, у яких кожен шар містить лише необхідні компоненти для роботи системи. Конфігурація середовища здійснюється за допомогою змінних середовища (Environment Variables) у файлах .env, що дозволяє безпечно зберігати API-ключі та налаштувань підключення до бази даних без зміни вихідного коду.

Для забезпечення безпечного обміну даними між фронтенд-(порт 3000) та бекенд-частинами (порт 5000), що працюють в ізольованих контейнерах, на серверному рівні реалізовано підтримку політики CORS (Cross-Origin Resource Sharing) [27], яка дозволяє безперешкодно обробляти асинхронні запити до інтелектуального ядра системи, гарантуючи стабільну візуалізацію результатів підбору шрифтів. Така архітектура забезпечує гнучкість, масштабованість системи та можливість її розгортання на хмарних платформах без модифікації коду.

З метою оцінки стабільності і коректності функціонування розроблюваної системи здійснено комплексне тестування її основних компонентів. Під час перевірки було протестовано основні можливості системи: алгоритм формування пар, правильність отриманих результатів, роботу інтерактивних елементів інтерфейсу та стабільність функціонування системи.

Тестування системи здійснювалося шляхом виконання набору тестових сценаріїв із застосуванням модульного та функціонального підходів. Результати тестування представлені у таблицях В.1 і В.2.

Повний цикл тестування передбачав перевірку всіх етапів роботи системи – від запуску та взаємодії з інтерфейсом до логіки та аналізу отриманих результатів. У результаті платформа продемонструвала стабільну роботу, відповідність вимогам та коректну взаємодію компонентів. Отримані результати підтвердили готовність системи до практичного використання.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було:

1. Досліджено предметну область з системно-технологічної точки зору, що дозволило структурувати процес підбору шрифтів як цілісний інформаційний бізнес-процес.

2. Проаналізовано існуючі сервіси для автоматизованого підбору шрифтових пар. У результаті виявлено проблему суб'єктивності та великих часових витрат на ручний підбір, що обґрунтувало доцільність створення інтелектуальної системи Glypha.

3. Проведено постановку задачі та сформульовано технічні вимоги до системи, що дало змогу визначити ключові функціональні та нефункціональні характеристики проєктованого рішення.

4. Описано архітектуру автоматизованої системи генерації типографічних пар шрифтів для дизайнерських задач і розроблено її алгоритмічне та інформаційне забезпечення. Це забезпечило чітке розподілення компонентів системи та визначення принципів їхньої взаємодії, що стало основою для подальшої програмної реалізації сервісу.

5. Реалізовано систему з клієнт-серверною архітектурою на основі попередньо спроектованої структури з використанням сучасних технологій розробки програмного забезпечення. Клієнтська частина реалізована на базі бібліотеки React. Серверний бекенд написаний мовою Python із використанням фреймворку Flask. Ключовим компонентом є модуль семантичного аналізу, який використовує алгоритми обробки природної мови (NLP).

6. Використано механізм контейнеризації програмного рішення, що забезпечує ізольоване середовище виконання, спрощує процес розгортання та підвищує масштабованість системи.

7. Проведено тестування основних компонентів системи з метою оцінки стабільності та коректності її функціонування.

8. Отримані результати свідчать про досягнення поставленої мети,

передбаченої темою роботи. Розроблене програмне рішення реалізує автоматизований підбір типографічних пар шрифтів на основі семантичного аналізу тексту та заданих параметрів оформлення.

9. Практичне значення роботи полягає у можливості застосування розробленої системи для автоматизованого підбору типографічних пар шрифтів у сфері графічного дизайну, веброзроблення та створення цифрового контенту. Система сприяє скороченню часу на вибір шрифтових поєднань і забезпечує підтримку прийняття дизайнерських рішень на основі аналізу текстового контенту та параметрів оформлення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ambrose G., Harris P. The Fundamentals of Typography. 2nd ed. Lausanne: AVA Publishing SA, 2011. 192 p.
2. Lupton E. Thinking with type: a critical guide for designers, writers, editors, & students. 2nd ed. New York: Princeton Architectural Press, 2010. 224 p.
3. Бідюк П. І, Тимошук О. Л., Коваленко А. Є., Коршевніук Л. О. Системи і методи підтримки прийняття рішень: підручник. Київ: КПІ ім. Ігоря Сікорського, 2022. 610 с.
4. Fontjoy: інструмент для автоматичного підбору гармонійних шрифтових гарнітур із використанням штучного інтелекту. URL: <https://fontjoy.com/>
5. Fontpair: сервіс шрифтових комбінацій для дизайнерів і розробників з добірками. URL: <https://www.fontpair.co/>
6. Monotype AI Font Pairing: сервіс для автоматичного підбору шрифтових пар із використанням алгоритмів машинного навчання. URL: <https://www.monotype.com/font-pairing>
7. Kulahcioglu T., De Melo G. Semantics-aware typographical choices via affective associations. *Language Resources and Evaluation*. 2021. Vol. 55. № 1. P. 105-126. DOI: 10.1007/s10579-020-09499-0.
8. Client-server overview. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-server_overview
9. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed Representations of Words and Phrases and Their Compositionality. *Advances in Neural Information Processing Systems* 2013. URL: https://www.researchgate.net/publication/257882504_Distributed_Representations_of_Words_and_Phrases_and_their_Compositionality
10. spaCy for Natural Language Processing. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/spacy-for-natural-language-processing/>
11. uk_core_news_lg: Ukrainian news model for spaCy. spaCy Models

Documentation. URL: https://spacy.io/models/uk#uk_core_news_lg

12. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. 3rd ed. draft. 2026. URL: <https://web.stanford.edu/~jurafsky/slp3/>

13. About SQLite. SQLite Documentation. URL: <https://www.sqlite.org/about.html>

14. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format. RFC 8259. Internet Engineering Task Force (IETF). 2017. URL: <https://www.rfc-editor.org/rfc/rfc8259>

15. React Documentation: Managing State and Rendering. Meta Open Source. URL: <https://react.dev/learn>

16. Google Fonts API Documentation. Google Developers. URL: https://developers.google.com/fonts/docs/developer_api

17. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol, CA: O'Reilly Media, 2017. 590 p.

18. N-tier architecture style. Microsoft Learn. 2025. URL: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>

19. Що таке Single Page Application та як працює SPA сайт. Wezom. URL: <https://wezom.com.ua/ua/blog/chto-takoe-spa-prilozheniya>

20. JavaScript. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

21. Hertzen N. html2canvas: Screenshots with JavaScript. URL: <https://html2canvas.hertzen.com/>

22. Flask Documentation. Pallets. URL: <https://flask.palletsprojects.com/>

23. ISO 9241-210:2019. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Second edition. Geneva : ISO, 2019. 34 p.

24. Yablonski J. Laws of UX: Using Psychology to Design Better Products and Services. 2nd ed. Sebastopol, CA: O'Reilly Media, 2024. 164 p.

25. Docker Documentation. Docker overview. URL: <https://docs.docker.com/get-started/overview/>
26. Кудрявцев Д. О., Мичуда Л. З. Огляд мікросервісної архітектури та аналіз типових вразливостей. *Computer Systems and Networks*. 2025. Vol. 7, No. 2. С. 114–120. DOI: <https://doi.org/10.23939/csn2025.02.114>.
27. Cross-Origin Resource Sharing (CORS). MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
28. Панасевич К.О., Турченко І.В. Автоматизована система генерації типографічних пар шрифтів для дизайнерських задач. *ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами: збірник тез доповідей міжнародної науково-практичної конференції студентів і молодих вчених* (Тернопіль, 21-22 травня 2026 р). Тернопіль: ЗУНУ, 2026. С.94-97.
29. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

Додаток А

Порівняльний аналіз функціональних можливостей існуючих аналогів

Критерій порівняння	Fontjoy	Fontpair	Monotype AI Font Pairing
Джерело шрифтів	Google Fonts	Відкриті бібліотеки (зокрема Google Fonts)	Власна ліцензована база шрифтів
Мовна адаптація	Відсутня	Відсутня	Обмежена (шрифти з розширеною мовною підтримкою)
Механізм формування комбінацій на базі AI	Здійснюється алгоритмами глибокого навчання	Відсутній, лише дизайнерські рішення	Здійснюється на аналізі типографічних характеристик
Взаємодія з текстом	Редагування стандартного текстового зразка в реальному часі	Редагування тексту в UI-макетах	Редагування стандартного текстового зразка в реальному часі
Гнучкість налаштувань	Контраст, фіксація шрифту	Тип та область макетів, категорії, фіксація шрифту, ручний підбір	Розмір, фіксація шрифту, ручний підбір
Контекстний аналіз тексту	Відсутній	Відсутній	Здійснюється на базі метаданих базово
Представлення результату	Текстовий блок (3 рівні-заголовки, підзаголовки та основний текст)	Статичні дизайн макети	Текстовий блок (2 рівні -заголовки та основний текст)
Збереження даних	Відсутнє	Хмарне / Локальний експорт	Тимчасове (локальне збереження в колекцію)
Доступність	Безкоштовно	Безкоштовна з базовим функціоналом / Платна з необмеженими можливостями	Безкоштовно (з ліцензійними обмеженнями шрифтів)

Додаток Б

Програмний код семантичного аналізу тексту

```
import spacy

try:
    nlp = spacy.load("uk_core_news_lg")
except OSError:
    print("Помилка: Модель не знайдено.")
    nlp = None

ANCHORS = {
    "cheerful": "свято радість щастя весело сміх подарунок гра вечірка",
    "serious": "бізнес закон звіт фінанси суд робота офіс договір",
    "tech": "код програма сайт технології інтернет робот дані",
    "elegant": "краса мода стиль естетика мистецтво дизайн люкс",
}

# Кешування векторів для швидкодії
if nlp:
    ANCHOR_DOCS = {mood: nlp(text) for mood, text in ANCHORS.items()}
else:
    ANCHOR_DOCS = {}

def get_semantic_mood(user_text: str) -> str:
    """
    Визначає семантичний настрій тексту через косинусну подібність.
    """
    if not nlp or not user_text.strip():
        return "neutral"
```

```

doc = nlp(user_text.lower().strip())

# Перевірка наявності векторів
if not doc.has_vector or len(doc) == 0:
    return "neutral"

scores = {}

# Косинусна подібність до кожної категорії
for mood_key, anchor_doc in ANCHOR_DOCS.items():
    scores[mood_key] = float(doc.similarity(anchor_doc))

best_mood = max(scores, key=scores.get)

# Поріг впевненості
return best_mood if scores.get(best_mood, 0) > 0.2 else "neutral"

```

Додаток В

Результати функціонального та модульного тестування системи

Таблиця В.1 – Функціональне тестування основних компонентів системи

Тестовий сценарій	Очікуваний результат	Фактичний результат	Швидкодія
Відгук головної СТА-кнопки «Створити пару»	Перехід до дашборду – основного інструменту взаємодії	Головна СТА-кнопки переводить користувача до дашборду	INP: 96 мс
Огляд інформаційного блоку «Як працює Glypha»	Візуалізація карток при завантаженні та скролінгу без дефектів	Всі елементи в межах блоку відображаються коректно	CLS: 0
Перемикання між режимами («Усі параметри», «Тільки текст», «Ручні налаштування»)	Зміна конфігурації полів введення відбувається без затримок	Стан інтерфейсу оновлюється коректно, без помітної затримки при перемиканні режимів	INP: 289 мс CLS: 0
Відгук системи при натисканні кнопки «Згенерувати»	Миттєвий запуск алгоритму підбору шрифтових пар	Користувач отримує згенеровану пару відповідно до введених параметрів	INP: 156 мс CLS: 0.01
Навігація при натисканні на назву шрифту (Header/Body)	Перехід на сторінку шрифту у Google Fonts у новій вкладці	Перехід відбувається миттєво	INP: 145 мс CLS: 0
Відгук системи при натисканні кнопки «Завантажити у PNG»	Формування PNG файлу та автоматичне збереження на пристрій	Файл завантажується коректно; рендеринг тексту відповідає макету	INP: 94 мс CLS: 0
Відгук системи при натисканні кнопки «Зберегти до колекції» (для авторизованих користувачів)	Збереження згенерованої пари до колекції та зміна стану кнопки	Кнопка змінює статус і пара шрифтів зберігається в колекцію користувача	INP: 92 мс CLS: 0

Таблиця В.2 – Модульне тестування основних компонентів системи

Тестовий сценарій	Очікуваний результат	Фактичний результат	Швидкодія
Семантичний (NLP) аналіз та обробка текстового промпту користувача	Із запиту «Сучасний бренд екологічної косметики» виділено вагові коефіцієнти ознак	Проаналізовано успішно; зафіксовано коефіцієнти: elegant: 0.569, tech: 0.545. Визначено домінуючу категорію: elegant	Час обробки: 101 мс
	Із запиту «Організація дитячих свят та яскравих шоу з аніматорами» виділено вагові коефіцієнти ознак	Проаналізовано успішно; зафіксовано коефіцієнти: cheerful: 0.376, serious: 0.124. Визначено домінуючу категорію: cheerful	Час обробки: 106 мс
Відповідність семантичних ознак релевантним шрифтовим категоріям	Семантичні характеристики запиту коректно співвідносяться з відповідними категоріями шрифтів (Serif, Sans Serif, Display тощо)	Категорія Cheerful – шрифт Comfortaa (Rounded Sans) Категорія Elegant – Cormorant (High-contrast Serif)	Час обробки: 99 мс
Логіка візуального контрасту	Система формує гармонійну та візуально контрастну пару шрифтів відповідно до обраного режиму	Зміна пари шрифтів при незмінному промпті, але різних режимах контрасту проведено коректно	Час обробки: 115 мс
Обробка некоректних/порожніх запитів користувача	Некоректні або порожні запити не спричиняють збоїв системи;	Некоректні запити ігноруються, порожні запити проходять з оцінкою параметрів, збоїв не спостерігається	Час обробки: 95 мс
Фільтрація шрифтів за заданими параметрами та мовною підтримкою	У вибірці залишаються шрифти, що відповідають заданим критеріям та підтримують необхідну мову	Підбір шрифтів з підтримкою кирилиці відбувається успішно	Час обробки: 160 мс

Додаток Г

Копія опублікованих результатів

**Міністерство освіти і науки України
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління**



Матеріали

I Міжнародної науково-практичної конференції студентів і молодих вчених
**«ICSPM 2026: Intelligent Computing Systems and Project Management /
Інтелектуальні обчислювальні системи та управління проєктами»**
21–22 травня 2026 р.



м. Тернопіль - 2026

Олексишин Є.О., Васильків Н.М. МЕТОД РЕАЛІЗАЦІЇ AI-АГЕНТА ДЛЯ ДОМЕНУ НАСТІЛЬНИХ РОЛЬОВИХ ІГОР НА ОСНОВІ ПОЄДНАННЯ RAG ТА ДОМЕННОГО FINE-TUNING	85
Островецький С.В., Полуктова Н.Р. МОДЕЛЬ ПРОГРАМНОЇ АРХІТЕКТУРИ ДЛЯ ІНТЕГРАЦІЇ REINFORCEMENT LEARNING В АЛГОРИТМ ALNS.....	90
Панасевич К.О., Турченко І.В. АВТОМАТИЗОВАНА СИСТЕМА ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ ДЛЯ ДИЗАЙНЕРСЬКИХ ЗАДАЧ	94
Реденський О.О., Котляр О.В., Духновська К.К. ПОРІВНЯННЯ МОДЕЛЕЙ СЕМАНТИЧНОГО ПРЕДСТАВЛЕННЯ НАУКОВИХ ТЕКСТІВ ДЛЯ ЗАДАЧІ РЕКОМЕНДАЦІЇ ПУБЛІКАЦІЙ.....	98
Руснак А.М., Сегін А.І. СИСТЕМА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ЗАХВАТОМ МАНІПУЛЯТОРА РОБОТА	102
Созанський А.Б., Васильків Н.М. RAG-АГЕНТ ЯК ОСНОВА ПРОГРАМНИХ МОДУЛІВ ПОШУКУ РЕЛЕВАНТНОЇ ІНФОРМАЦІЇ.....	106
Фурда А.І., Турченко І.В. ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ РЕКОМЕНДАЦІЙ ТА ПРОГНОЗУВАННЯ ПОПИТУ ДЛЯ СЕРВІСУ ОНЛАЙН-ОРЕНДИ АВТОМОБІЛІВ.....	109
Цинайко В.П., Биковий П.Є. ПРОГРАМНИЙ МОДУЛЬ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ПРОДАЖІВ АПТЕЧНОЇ МЕРЕЖІ	112
Яремчук В.М., Лендюк Т.В. ПРОГРАМНИЙ МОДУЛЬ ІНТЕЛЕКТУАЛЬНОГО АГЕНТА ДЛЯ САМОСТІЙНОЇ ХОДЬБИ ПЕРСОНАЖА В СЕРЕДОВИЩІ UNITY	116
 <i>СЕКЦІЯ - ПРИКЛАДНІ ТЕХНОЛОГІЇ ТА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ / APPLIED TECHNOLOGIES AND INFORMATION AND COMMUNICATION SYSTEMS</i>	
Боднаровський І.В., Турченко І.В. ВЕБ-ОРІЄНТОВАНА СИСТЕМА ПРОВЕДЕННЯ ЦИФРОВОГО ДОЗВІЛЛЯ	122

УДК 004.92:004.8

АВТОМАТИЗОВАНА СИСТЕМА ГЕНЕРАЦІЇ ТИПОГРАФІЧНИХ ПАР ШРИФТІВ ДЛЯ ДИЗАЙНЕРСЬКИХ ЗАДАЧ

Панасевич Катерина Олександрівна,
здобувач першого (бакалаврського) рівня вищої освіти,
katiapanasevych@gmail.com

Науковий керівник: Турченко Ірина Василівна,
к.т.н., доцент, доцент кафедри інформаційно-
обчислювальних систем і управління,
Західноукраїнський національний університет,
itu@wunu.edu.ua,
ORCID: 0000-0002-9441-6669

Постановка проблеми. Розвиток типографіки є результатом багатовікової еволюції письма та технологій відтворення тексту, що суттєво вплинули на формування та різноманіття шрифтових форм [1]. З огляду на значну кількість доступних гарнітур і варіацій їх накреслень кількість можливих комбінацій є надзвичайно великою. Це ускладнює вибір та суттєво збільшує часові витрати дизайнера на пошук оптимального рішення.

У таких умовах доцільним є застосування автоматизованих інформаційних систем, здатних забезпечити структурований підхід до підбору шрифтових пар. Використання засобів автоматизації дозволяє систематизувати типографічні характеристики гарнітур, звзвити множину можливих варіантів відповідно до заданих параметрів та сформулювати рекомендації щодо сумісних поєднань.

Більшість існуючих систем працюють на основі чітко визначених правил або статистичних співвідношень між категоріями шрифтів і при цьому не враховують семантичний зміст тексту [2]. Крім того, відсутність інструментів для налаштування параметрів генерації обмежує можливості користувача та знижує гнучкість застосування таких рішень у творчих проєктах.

Проблема автоматизованого підбору типографічних пар шрифтів останніми роками активно досліджується у контексті застосування методів машинного навчання та аналізу візуальних характеристик шрифтів. У роботі Sh. Jiang та ін. запропоновано підхід автоматичного формування шрифтових пар на основі навчання візуальних взаємозв'язків між шрифтами із використанням великої бази реальних документів [3]. Автори застосували методи metric learning та k-nearest neighbors для визначення гармонійних шрифтових комбінацій, що дозволило автоматизувати процес рекомендації шрифтів для дизайнерських задач.

У сучасних дослідженнях також значна увага приділяється семантичному аналізу типографіки. Зокрема, у роботі Y. Tatsukawa та ін. представлено модель FontCLIP, яка поєднує семантичні можливості vision-language моделей із

типографічними характеристиками шрифтів [4]. Запропонований підхід забезпечує пошук і підбір шрифтів на основі семантичних атрибутів та текстових описів, що відкриває перспективи створення інтелектуальних систем генерації шрифтових пар для дизайнерських застосувань.

Виклад основного матеріалу. Розроблена автоматизована система, що надає користувачам персоналізовані рекомендації щодо поєднання шрифтів на основі семантичного аналізу тексту та заданих параметрів оформлення, має тривірневу клієнт-серверну архітектуру.

Рівень користувацького інтерфейсу, або клієнтська частина, забезпечує взаємодію користувача із системою та виступає сполучною ланкою між користувацькими запитами й бізнес-логікою сервера.

Серверна частина, яка є ядром розроблюваної системи, зосереджує логіку функціонування, а саме, лінгвістичний розбір тексту, пошук семантичних зв'язків та математичне ранжування шрифтових пар. Сервер обробляє отриманий запит, звертається до бази даних і повертає готовий результат користувачу.

Фундаментом для накопичення та структурування інформації в системі є рівень збереження даних. Його реалізовано на основі реляційної системи управління базами даних SQLite.

Для відображення загальної структури та логіки взаємодії компонентів використано модель у стандарті IDEF0 (рисунок 1), яка демонструє як користувацький інтерфейс та серверна частина взаємодіють для обробки запитів користувачів, виконання лінгвістичного аналізу, семантичного пошуку та формування персоналізованих рекомендацій шрифтових пар.



Рисунок 1 – Функціональна модель автоматизованої системи (IDEF0 діаграма)

Відповідно до представленої діаграми, центральним процесом системи є «Автоматизований підбір шрифтових пар». Цей блок інтегрує в собі всі логічні процеси системи та забезпечує перетворення необробленого запиту користувача

у готове візуальне рішення. Його функціонування базується на взаємодії вхідних даних, засобів управління та технічних механізмів.

Вхідні дані визначаються текстовим описом проекту та набором технічних параметрів. Це та інформація, яка за ініціації користувача сприяє пошуку і звужує підбір за мовною підтримкою та візуальною контрастністю майбутньої пари.

Вхідна інформація від користувача може надходити у двох форматах: довільного текстового опису проекту українською/ англійською мовами або ж у вигляді фіксованих параметрів, що обираються через графічний інтерфейс.

Управління основним процесом реалізується на основі заздалегідь визначених правил. До них належать алгоритм обробки природної мови та формалізовані принципи типографічної сумісності шрифтів.

Алгоритм обробки природної мови забезпечує обробку запиту користувача та формування рекомендацій типографічно узгоджених шрифтових пар на основі семантичного аналізу тексту. Водночас правила сумісності шрифтів обмежують простір вибору, залишаючи лише ті шрифтові пари, що відповідають критеріям візуальної гармонії та стилістичної узгодженості.

З математичної точки зору система базується на трансформації неструктурованого текстового запиту у векторне представлення [5] з подальшим обчисленням ступеня відповідності шрифтових пар. Запропонований підхід фактично впроваджує у систему елементи машинного навчання (Machine Learning), оскільки дає змогу формувати персоналізовані та семантично релевантні рекомендації.

В основі роботи програмного комплексу лежить використання NLP (Natural Language Processing) із бібліотекою spaCy [6], що дозволяє комп'ютерам розуміти та обробляти людську мову. Використовується модель, орієнтована на україномовну аудиторію, а саме uk_core_news_lg [7]. Це велика попередньо натренована нейронна мережа, навчена на мільйонах україномовних реченнях та здатна розпізнавати зв'язки між словами та витягувати їх семантичне значення.

Основою інформаційного наповнення системи є інтеграція із зовнішнім хмарним ресурсом Google Fonts API. Використання відкритого вихідного коду дозволяє миттєво завантажувати необхідні гарнітури та автоматично отримувати актуальні дані про шрифти: назви, категорії, атрибути (товщина, стиль, підтримка мов) та інші характеристики шрифтів. Це значно знижує навантаження на локальні ресурси, забезпечує ефективну взаємодію із зовнішнім сервісом та доступ до актуальних версій шрифтів.

Технічні механізми у процесі є ресурси, що забезпечують виконання функцій системи. До таких інструментів належать: програмне забезпечення, клієнтська частина (інтерфейс), серверна частина та база даних шрифтів.

Кінцевим результатом роботи є сформовані релевантні шрифтові пари, у зручному для візуального оцінювання вигляді. Вони проходять систему фільтрації та ранжування і готові до безпосереднього перегляду користувачем.

Панель керування та блок візуалізації результатів системи представлено на рисунку 2.

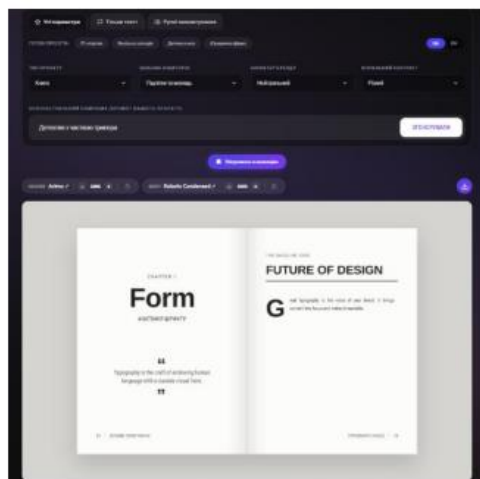


Рисунок 2 – Панель керування та блок візуалізації результатів

Результати підбору інтегруються у контекстні інтерактивні макети: веб-сайт, мобільний додаток, глянцева журнал, постер, книга та візитка.

Висновки. Реалізоване рішення системи забезпечує автоматизацію складного процесу підбору шрифтових гарнітур шляхом використання методів семантичного аналізу тексту. Це створює передумови для широкого практичного застосування системи у сфері графічного дизайну та суміжних галузях, пов'язаних із розробленням і візуальним оформленням цифрового контенту.

Список використаних джерел

1. Ambrose G., Harris P. The Fundamentals of Typography. 2nd ed. Lausanne: AVA Publishing SA, 2011. 192 p.
2. Kulahcioglu T., De Melo G. Semantics-aware typographical choices via affective associations. *Language Resources and Evaluation*. 2021. Vol. 55. № 1. P. 105-126. DOI: 10.1007/s10579-020-09499-0.
3. Jiang S., O'Donovan P., Woodford O., Iqbal K. Graphic design with large-scale font data. *Human Factors in Computing Systems*. Denver: proceedings of the 2017 CHI Conference. USA, 2017. P. 3130-3142. DOI: 10.1145/3025453.3026017.
4. Tatsukawa Y., Shen I.-C., Qi A., Koyama Y., Igarashi T., Shamir A. FontCLIP: A semantic typography visual-language model for multilingual font applications. *Computer Graphics Forum*. 2024. DOI: 10.1111/cgf.15043.
5. Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J. Distributed Representations of Words and Phrases and Their Compositionality. *Advances in Neural Information Processing Systems*. 2013. URL: https://www.researchgate.net/publication/257882504_Distributed_Representations_of_Words_and_Phrases_and_their_Compositionality.
6. spaCy for Natural Language Processing. GeeksforGeeks. 2026. URL: <https://www.geeksforgeeks.org/spacy-for-natural-language-processing/>
7. uk_core_news_lg: Ukrainian news model for spaCy. spaCy Models Documentation. URL: https://spacy.io/models/uk#uk_core_news_lg.