

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-08

Дипломний проект

**здобувачки освіти денної форми навчання
КГ.08.04.000.ДП**

***ГАНЕНКО
ОЛЕНИ ІВАНІВНИ***

**м. Одеса
2025 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн»

Група: 4КГ-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка експертної веб-системи перукарні для оптимізації роботи колористів

Проектний матеріал складається з пояснювальної записки на 77 сторінках та графічного (презентаційного) матеріалу на 14 аркушах (слайдах)

Дипломник _____ (Ганенко О. І.)
Керівник _____ (Скорняков В.С.)

Консультанти:

з економічного розділу _____ (Канський М.Ю.)
з розділу охорони праці та техніки безпеки _____ (Чорновол Н.І.)
з нормоконтролю _____ (Петрашова В.І.)
старший консультант _____ (Кривченко Ю.В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю.В.)
Завідувач відділення _____ (Краснокутська К.Г.)

Захист «27» червня 2025 р. Протокол ЕК № 6
Оцінка ЕК 5/відмінно / 90б.

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітньо-професійна програма ««Комп'ютерна графіка і web-дизайн»»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.
“ 19 ” 08 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачці освіти Ганенко Олені Іванівні
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка експертної веб-системи перукарні для оптимізації роботи колористів.

затверджена наказом по коледжу від “14” листопада 2024р. № 246

2. Термін здачі закінченого проекту _____

3. Вихідні данні до проекту _____

1. Реалізувати інтуїтивно зрозумілий інтерфейс з орієнтацією на колористів;
2. Підібрати сучасні веб-технології, що дозволяють будувати інтерактивну систему;
3. Побудувати математичну модель взаємодії кольорів (змішування, компліментарність);
4. Забезпечити гнучкість дизайну палітр, включаючи підтримку мобільних пристроїв.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

1. Аналіз предметної області;
2. Теоретичні засади кольору та колористики;
3. Аналіз існуючих цифрових рішень у сфері краси;
4. Вибір інструментів та архітектури веб-системи;
5. Реалізація інтерфейсу та логіки обробки кольору;
6. Інтерактивна побудова палітри та системи змішування;
7. Розробка користувацького гайду та сценаріїв взаємодії;
8. Економічний розрахунок;
9. Аспекти охорони праці та техніки безпеки.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Титул; Представлення кольорів (HSL) та відтінків; Репрезентація кольорів у застосунку; Архітектура React і компоненти; Блок-схема шляху користувача; Інтерфейс: головна сторінка; Кольорова модель і колористика; Компліментарність: реалізація; Сторінка «Компліментарність»: аналіз відтінків; Змішування: реалізація; Сторінка «Змішування»: введення і результат; Поради колориста; Візуальні приклади: симуляція та результат; Вивантаження на сервер.

6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Скорняков В.С.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 12.05.2025

Керівник Скорняков В.С.
(підпис)

Завдання прийняв до виконання Ганенко О.І.
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Формулювання теми, мети та актуальності	15.05.25	
2	Аналіз особливостей колористики та потреб	16.05.25	
3	Вивчення технологій із візуалізацією кольору	19.05.25	
4	Підбір фреймворку та архітектурного підходу	20.05.25	
5	Розробка логіки обробки кольору	22.05.25	
6	Інтеграція порад, гіда користувача	23.05.25	
7	Реалізація модуля порад колориста	26.05.25	
8	Побудова механізму генерації відтінків	28.05.25	
9	Тестування створеного веб-застосунку	30.05.25	
10	Оформлення пояснювальної записки	04.06.25	
11	Підготовка графічних матеріалів	06.06.25	
12	Економічний розрахунок	10.06.25	
13	Опис аспектів охорони праці та техніки безпеки	13.06.25	
14	Підведення висновків	16.06.25	

Дипломник
(підпис)

Керівник Скорняков В.С.
(підпис)

ЗМІСТ

Вступ.....	7
1 Основний розділ.....	8
1.1 Огляд тематичного напрямку	8
1.1.1 Визначення ключових викликів	8
1.1.2 Особливості побудови цифрових інтерфейсів у сфері послуг	10
1.1.3 Особливості використання кольору та побудови палітр в інтерфейсах колористичних систем.....	13
1.1.4 Технічні аспекти роботи з кольором в інтерфейсі React-застосунку	16
1.3.5 Моделювання взаємодії кольорів у цифровому середовищі	20
1.2 Проектування додатку колориста.....	23
1.2.1 Огляд архітектури додатку.....	23
1.2.2 Алгоритми аналізу відтінків у колористиці	26
1.2.3 Опис логіки обробки коду відтінку.....	29
1.2.4 Змішування кольорів у застосунку.....	34
1.2.5 Побудова інтерактивної палітри та кольорового кола.....	36
1.2.6 Взаємодія компонентів у повному циклі роботи з кольором.....	38
1.3 Тестування та розробка гайду до застосунку	40
1.3.1 Призначення та загальний огляд інтерфейсу	40
1.3.2 Введення кольору у форматі відтінку	43
1.3.3 Використання блоку змішування кольорів.....	46
1.3.4. Навігація через кольорову палітру та поради колориста	48
1.3.5 Часті запитання та поради	50
2 Економічний розділ	53
2.1 Резюме	53
2.2 Визначення трудомісткості розробки ПЗ.....	53
2.3 Розрахунок ціни програмного продукту	56
3 Розділ охорони праці та техніки безпеки.....	58
3.1 Основні положення.....	58
3.2 Аналіз умов праці	58

					<i>КГ 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

3.3 Гігієнічні вимоги до виробничого середовища	58
3.2.1 Вимоги до приміщення	58
3.2.2 Виробниче освітлення	59
3.2.3 Гігієнічні нормування параметрів повітря робочої зони	59
3.2.4 Організація робочих місць	60
3.2.5 Електробезпека	61
3.4 Пожежна безпека	61
3.5 Підведення підсумків	62
Висновки.....	63
Перелік використаних інформаційних джерел	64
ДОДАТОК А. Фрагмент програмного коду компоненту обчислення кольорів	65
Додаток Б. Слайди мультимедійної презентації	70

ВСТУП

Сфера обслуговування в Україні невпинно розвивається, і зокрема індустрія краси демонструє зростання попиту на сучасні цифрові рішення. Салони перукарських послуг дедалі частіше впроваджують інноваційні підходи до організації своєї роботи, що зумовлено не лише конкуренцією на ринку, а й потребою у підвищенні якості обслуговування клієнтів. У цьому контексті особливої уваги заслуговує напрям розвитку колористики, це галузь, де точність, індивідуальний підхід та швидкість прийняття рішень мають суттєвий вплив на кінцевий результат. Для фахівців у цій сфері наявність спеціалізованих інструментів, що допомагають у підборі кольору, аналізі відтінків та моделюванні результатів, є вирішальною умовою ефективної роботи. Роль візуального сприйняття особливо вагома у сфері краси, де колірна гармонія впливає не лише на зовнішній вигляд, а й на емоційне сприйняття результату. Як зазначено у фаховій праці з візуального дизайну: «Кольорова композиція в традиційному вбранні додавала художньої виразності та забезпечувала композиційну завершеність образу» [1].

Українські фахівці з колористики дедалі активніше цікавляться сучасними інформаційними рішеннями, які дозволяють автоматизувати та покращити процес прийняття рішень. Водночас, попри зростаючу потребу, на ринку відсутні повноцінні онлайн-інструменти, адаптовані під потреби місцевих майстрів і студій краси.

Процеси гармонізації професійних стандартів України з вимогами Європейського Союзу, які стали актуальними в рамках Угоди про асоціацію, стимулюють розвиток цифрових сервісів у сфері освіти, обслуговування та технічної підтримки фахівців. Саме тому створення спеціалізованої веб-системи, яка допоможе колористам проводити аналіз кольорів, підбирати фарби за заданими параметрами, оцінювати компліментарність відтінків і візуалізувати результат, є не лише актуальним завданням, а й внеском у процес цифрової трансформації професійної діяльності в Україні.

					<i>КГ 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ОСНОВНИЙ РОЗДІЛ

1.1 Огляд тематичного напрямку

1.1.1 Визначення ключових викликів

Індустрія краси є однією з най динамічніших сфер обслуговування, що швидко адаптується до цифрових технологій та потребує спеціалізованих інструментів для оптимізації внутрішніх процесів. Колористика, як окремий напрям у роботі перукаря, вимагає глибокого розуміння взаємодії відтінків, особливостей структури волосся, а також індивідуальних запитів клієнтів. У цих умовах зростає інтерес до цифрових рішень, які могли б стати підтримкою у прийнятті складних рішень щодо змішування фарб, вибору тону та прогнозування результату фарбування.

Сучасні тенденції у сфері перукарських послуг демонструють чіткий перехід до використання цифрових додатків, симуляторів кольору та мобільних сервісів для візуалізації результату. Водночас українські фахівці часто стикаються з відсутністю локалізованих інструментів, адаптованих до навчальних програм та практик, прийнятих у багатьох салонах краси. Це ускладнює інтеграцію таких сервісів у щоденну роботу та знижує їхню ефективність у реальному середовищі.

Попри існування окремих зарубіжних рішень для візуалізації кольору волосся, переважна більшість з них орієнтовані на масовий сегмент або на великі мережі салонів. Складні системи не враховують потреб дрібних студій чи індивідуальних майстрів, які складають більшість професійного ринку в Україні. Це створює розрив між потенціалом технологій та реальними можливостями їхнього використання фахівцями.

Одним з ключових викликів у цій сфері є відсутність інструменту, що дозволяв би поєднувати навчальні, практичні й консультаційні функції в одному середовищі. Колорист працює не лише з барвником, але й з клієнтом, тож потребує системи, яка допомагає зберігати історію фарбувань, аналізувати попередні змішування та пропонувати оптимальні рішення на основі досвіду.

Особливої актуальності набуває питання інтеграції цифрових систем у

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

навчальний процес у закладах професійної освіти. Майбутні фахівці мають опанувати інструменти, які будуть корисні у практичній діяльності вже з перших днів роботи. Відсутність відповідного ПЗ у навчальному процесі знижує конкурентоспроможність випускників на ринку праці.

Ще одним викликом є складність розрахунку результату змішування кольорів. В умовах, коли навіть досвідчені колористи можуть припускатися помилок, система, що візуалізує прогнозований результат, дозволяє знизити ризик незадоволення клієнта та зменшити кількість повторних процедур. Такий підхід відповідає прагненню до підвищення якості обслуговування, зменшення витрат часу та матеріалів.

Важливо також враховувати психологічний аспект взаємодії між клієнтом і майстром. Можливість візуалізувати майбутній результат ще до початку процедури підвищує довіру клієнта та полегшує комунікацію. Це особливо актуально в умовах високої конкуренції, де кожна перевага може відігравати вирішальну роль у виборі салону.

У рамках євроінтеграційних процесів Україна прагне гармонізувати освітні й професійні стандарти зі стандартами Європейського Союзу. Це стосується і сфери послуг, де все більше значення надається цифровим навичкам та вмінню працювати з інтерактивними системами. Наявність власного веб-застосунку, орієнтованого на потреби українських колористів, підвищує відповідність ринку праці сучасним європейським вимогам.

Проблематика посилюється ще й тим, що існуючі програми часто мають комерційне спрямування, закриті вихідні коди та не дозволяють розширювати функціонал відповідно до потреб користувача. У таких умовах створення відкритого, гнучкого і доступного інструменту стає не просто бажаним, а вкрай необхідним рішенням.

Кажучи у загальному принципі, ключовими викликами у тематиці проекту є відсутність зручних цифрових інструментів для підтримки колористів, невідповідність наявних рішень локальному контексту, потреба в інтеграції таких інструментів у професійне навчання, а також бажання підвищити ефективність і

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

якість перукарських послуг. Розв'язання цих питань можливе лише за умови створення спеціалізованої веб-системи, орієнтованої на практичне застосування у роботі майстра. У зв'язку з цим особливу увагу було приділено вибору технологічної бази до вибору сучасної кодової бази та бібліотеки яка може допомогти у розв'язуванні питання – це бібліотека React. У сучасній українській літературі зазначається: «React зарекомендувала себе як одна з найпопулярніших та найпоширеніших бібліотек JavaScript завдяки своїй простоті та можливості використання для створення масштабованих додатків» [2].

У відповідь на ці виклики дипломна робота має на меті створення експертного інструменту, що не лише полегшить повсякденну діяльність фахівців, а й сприятиме формуванню нових стандартів у сфері колористики, орієнтованих на якість, технологічність і зручність для користувача.

1.1.2 Особливості побудови цифрових інтерфейсів у сфері послуг

У сучасній сфері послуг, особливо в індустрії краси, взаємодія між фахівцем і клієнтом усе частіше відбувається з використанням цифрових інструментів. У цьому контексті інтерфейс стає не лише технічним засобом, а й частиною емоційного досвіду користувача. Візуальна мова, швидкість реагування системи, зручність доступу до функцій усе це безпосередньо впливає на сприйняття якості обслуговування. Для перукаря-колориста інтерфейс має бути не просто зрозумілим, а й підтримувати його професійне бачення, дозволяти точно візуалізувати результат і сприяти швидкому ухваленню рішень.

Особливу роль відіграє естетична привабливість та інтуїтивна навігація. Система, яка працює у середовищі краси, повинна нести в собі елементи стилю та візуальної гармонії. Колорист, як і його клієнт, очікує побачити інтерфейс, що відображає чітку логіку, продуману композицію та візуальну відповідність тематиці. Грамотний інтерфейс стосується не лише кольорової палітри, а й розміщення елементів, анімацій, реакції на дії користувача. Подібні інтерфейси мають підкреслювати професійність сервісу, одночасно залишаючись легкими у сприйнятті. Реалізація повних вимог до системи потребує врахування не лише функціональності, а й психологічних аспектів дизайну. Користувач повинен

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

відчувати, що система «розуміє» його наміри, реагує на потреби без зайвих зусиль, та сприяє створенню спокійної, комфортної робочої атмосфери.

У середовищі, де результат є візуальним і часто емоційним, гарний та повний інтерфейс повинен мати здатність комунікації зміст не словами, а образами, кольорами, простотою вибору. Саме ці параметри формують вимоги до подальшого архітектурного проектування та вибору бібліотек, компонентів і стилістичних підходів у веб-розробці.

Цифрові інтерфейси у сфері обслуговування мають враховувати сценарії користування, які відрізняються від стандартних бізнес-систем або технічних інструментів. У випадку з колористикою користувач взаємодіє з системою в динамічному середовищі під час консультацій, підбору кольору, аналізу результатів змішування. Це вимагає миттєвого зворотного зв'язку, мінімуму переходів між екранами та візуалізації змін у режимі реального часу. Тому під час проектування інтерфейсу важливо враховувати частоту певних дій, логіку переміщення погляду, очікувану поведінку елементів після кліку чи зміни значення.

Окрему увагу слід звернути на адаптивність цифрового середовища до різних форматів пристроїв. У перукарнях і студіях майстри часто користуються планшетами або смартфонами, а не повноцінними комп'ютерами. Відповідно, всі функції системи повинні бути зручними для взаємодії через сенсорний екран, із урахуванням обмеженого простору та поведінки мобільного інтерфейсу.

Mobile First



Desktop First



Рисунок 1.1. Схематичне порівняння відображення інтерфейсу системи

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Кнопки, меню, поля введення все повинно відповідати принципам mobile-first-дизайну, не втрачаючи при цьому цілісного стилю та візуальної привабливості.

Крім технічної адаптації, важливим аспектом є персоналізація та можливість підлаштовувати інтерфейс під стиль роботи конкретного користувача. Для прикладу, система має враховувати вподобання щодо палітри кольорів, типових комбінацій, частоти використання певних тонів. У проекті це може реалізовуватися через збереження останніх дій, надання рекомендацій на основі попереднього досвіду або застосування механізмів машинного навчання. Використання цих можливостей дозволяє не лише пришвидшити роботу фахівця, а й підвищити його впевненість у результаті.

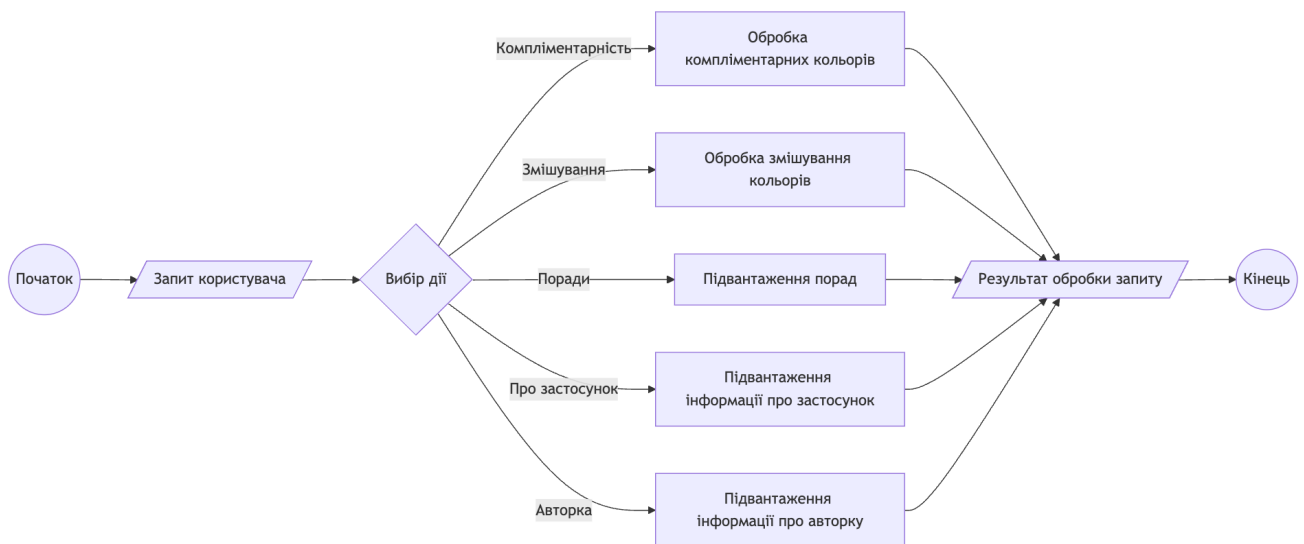


Рисунок 1.2. Блок-схема шляху користувача по системі

Користувацький інтерфейс у сфері послуг є також каналом комунікації з клієнтом, особливо якщо система дозволяє демонструвати очікуваний результат фарбування або порівнювати варіанти. У цьому випадку візуальна складова інтерфейсу має бути не просто елементом зручності, а й засобом емоційної взаємодії. Від того, як виглядає кольорова карта, наскільки реалістичною виглядає тон на зразку, залежить сприйняття як самої процедури, так і професіоналізму майстра. Ці особливості диктують потребу у використанні якісної графіки, анімацій, мікро взаємодій та ефектів підсвічування.

Слід вважати, що побудова інтерфейсу в цифровій системі для перукарської

справи не зводиться лише до розміщення кнопок і полів введення. Це комплексний процес, у якому поєднуються дизайнерські принципи, поведінкова аналітика, адаптивні технології та розуміння галузевої специфіки. Від якості реалізації цих аспектів значною мірою залежить успіх проекту, його здатність бути не лише інструментом, а частиною робочого середовища фахівця.

1.1.3 Особливості використання кольору та побудови палітр в інтерфейсах колористичних систем

Кольорова система є однією з центральних складових у проектуванні інтерфейсу, орієнтованого на роботу з колористикою. У таких випадках колір не лише виконує декоративну або стилістичну функцію, а стає основним інформаційним ресурсом, з яким безпосередньо працює користувач. Від правильності відображення кольорів, логіки їх організації у палітрах, зручності вибору та гнучкості взаємодії з відтінками залежить якість використання системи, її функціональність і професійна цінність для фахівця.

Цифрова палітра має відповідати як технічним, так і психологічним очікуванням користувача. З одного боку, вона повинна відображати точні відтінки відповідно до визначеної кольорової моделі (наприклад, RGB, HSL чи HEX). З іншого боку коректно враховувати професійний досвід колористів, які звикли оперувати кодами типу 8.1, 9.34, де перша цифра відповідає глибині тону, а друга відповідає нюансу відтінку. Саме така система кодування є зрозумілою для цільової аудиторії, і тому повинна бути впроваджена у цифровий продукт.

Важливою вимогою до інтерфейсу є наявність візуального інструмента для швидкого підбору кольору. Такі кольори можуть бути реалізовані у вигляді кола відтінків, де відображається взаємозв'язок між тонами: комплементарність, аналогія, контраст. Ці розрахунки дозволяють майстру не просто вибрати колір на основі номера, а побачити, як він буде взаємодіяти з іншими відтінками. Важливо, щоби ці інструменти були інтерактивними, змінювались у реальному часі та дозволяли легко скасовувати або змінювати вибір. Розглянемо наступну таблицю кольорів:

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

Таблиця 1.1 Порівняння кольорових моделей в інтерфейсах

Назва моделі	Тип представлення	Особливості	Переваги для колористик	Обмеження	Назва моделі
RGB	Червоний, Зелений, Синій (0–255)	Використовується в екранах і браузерях	Пряма підтримка на всіх пристроях, яскраві кольори	Незручність для інтуїтивного змішування кольору	RGB
HEX	Шістнадцятковий код (#RRGGBB)	Стислий запис RGB-моделі	Зручно використовувати в CSS, компактний вигляд	Потребує додаткового інструменту для редагування	HEX
HSL	Відтінок, Насиченість, Світлість	Інтуїтивно зрозумілий опис кольору	Легкість у створенні відтінків і модифікацій	Не підтримується як основна модель у Canvas API	HSL
Цифровий код фарби	Наприклад, 6.34 (глибина тону + нюанс)	Використовується в колористиці (L'Oréal, Wella тощо)	Зрозумілий для фахівців, легко пов'язується з реальними барвниками	Потребує карт перетворення у цифровий формат	Цифровий код фарби

Окрема увага приділяється ієрархії кольору у візуальному оформленні системи. Елементи, пов'язані з вибором або зміною кольору, повинні мати візуальний акцент. Це стосується не лише кнопок чи полів, але й стану елементів (hover, focus, active), які мають забезпечити користувача зворотним зв'язком про поточну дію. Наприклад, вибраний колір має бути підсвічений, а змішування відображене через динамічну анімацію або візуальний результат.

У контексті дизайну важливо дотримуватись якогось конкретного балансу між функціональністю та візуальною привабливістю. Кольори не повинні конфліктувати між собою, створювати надмірний візуальний шум або відволікати

увагу від основної дії. Крім того, палітра повинна враховувати сприйняття користувачами з порушеннями кольорового зору. Для цього рекомендується використовувати схеми з високим контрастом, а також додаткові індикатори (наприклад, текстові підписи або символи).

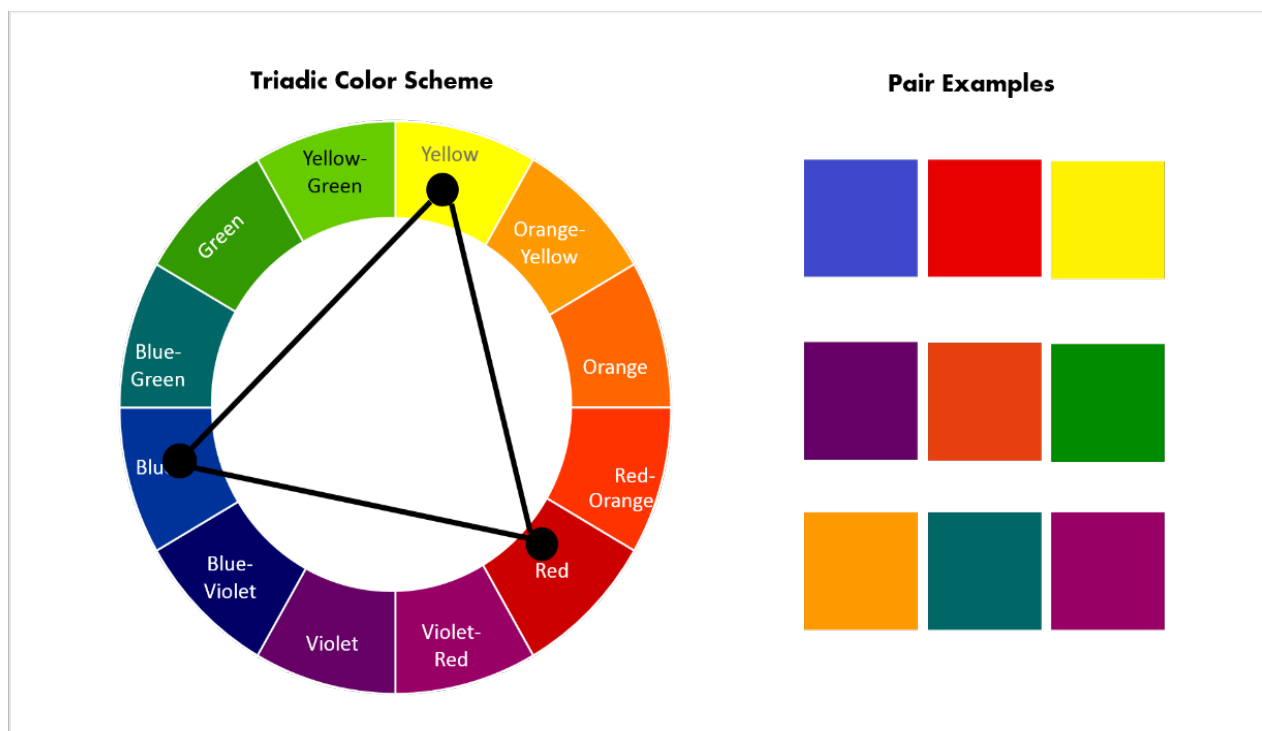


Рисунок 1.2. Приклад схеми з високим контрастом

Особливої складності набуває реалізація змішування кольорів. Для системи, орієнтованої на колористику, важливо не лише дати можливість поєднувати два або більше відтінків, а й наочно показати результат. Процес має враховувати співвідношення компонентів, тональну глибину, а також можливі похибки, пов'язані з фактичним кольором волосся клієнта. Навіть спрощена візуалізація результату змішування значно підвищує довіру до системи та зменшує ризики у реальній роботі. Кольори у таких системах - це не просто пікселі на екрані. Колір несе значення. Він може підсилювати інформацію, передавати емоції і навіть направляти поведінку користувача [3]. Результат складної взаємодії знань, досвіду, очікувань і креативності. Від того, як ці елементи реалізовані технічно, залежить не лише успіх інтерфейсу, а й здатність системи стати повноцінним інструментом у роботі колориста. Саме тому опрацювання кольору в проєкті, не доповнення до функціоналу, а його ядро.

1.1.4 Технічні аспекти роботи з кольором в інтерфейсі React-застосунку

Реалізація кольорової логіки у веб-застосунку на основі React вимагає врахування як дизайнерських принципів, так і практичних аспектів роботи з кодом. Оскільки колористика є центральним об'єктом функціональності, основний акцент робиться на компонентах, які безпосередньо взаємодіють із кольорами: палітри, селектори відтінків, динамічні блоки результатів змішування та інструменти попереднього перегляду.

У рамках даного проєкту було використано React як базову бібліотеку для створення інтерфейсу. Основна перевага полягає в можливості компонентного структурування логіки: кожен елемент, що відображає або обробляє колір, реалізований у вигляді окремого компонента такого як `ShadeInput`, `ColorResult`, `PaletteCircle`, `MixingPreview`. Даний підхід дозволяє ефективно керувати станом, забезпечити повторне використання коду та швидко масштабувати функціонал.

Для роботи з кольорами обрано формат кодування відтінків, який використовується у професійній перукарській практиці, як числове позначення на кшталт 6.34 або 9.1. Ці коди інтерпретуються функцією `parseShade`, яка розкладає їх на окремі компоненти, це глибину тону та нюанс відтінку. Далі ці значення перетворюються у HSL-модель, що дозволяє обчислити відтінок та побудувати відповідне візуальне представлення.

На базовому рівні усі обчислення кольору виконуються за допомогою кастомних утиліт, зокрема `toneToHue`, `tryParseShade` та `hsl()`. Окремі методи дають змогу не лише відображати колір у вигляді фону або маркера, а й створювати логіку для змішування, модифікації, компліментарності та попереднього перегляду. Усі ці обчислення виконуються в режимі реального часу, що дозволяє миттєво оновлювати інтерфейс при зміні параметрів.

Компоненти React використовують локальний стан (`useState`) для зберігання поточного значення кольору, а також `useEffect` для побудови залежностей між параметрами. Наприклад, при зміні базового коду (`value`) система автоматично обчислює відповідний колір і відображає його у формі. Також використовується `useMemo` для оптимізації повторних обчислень, особливо в частині роботи з

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

оберненими або змішаними відтінками.

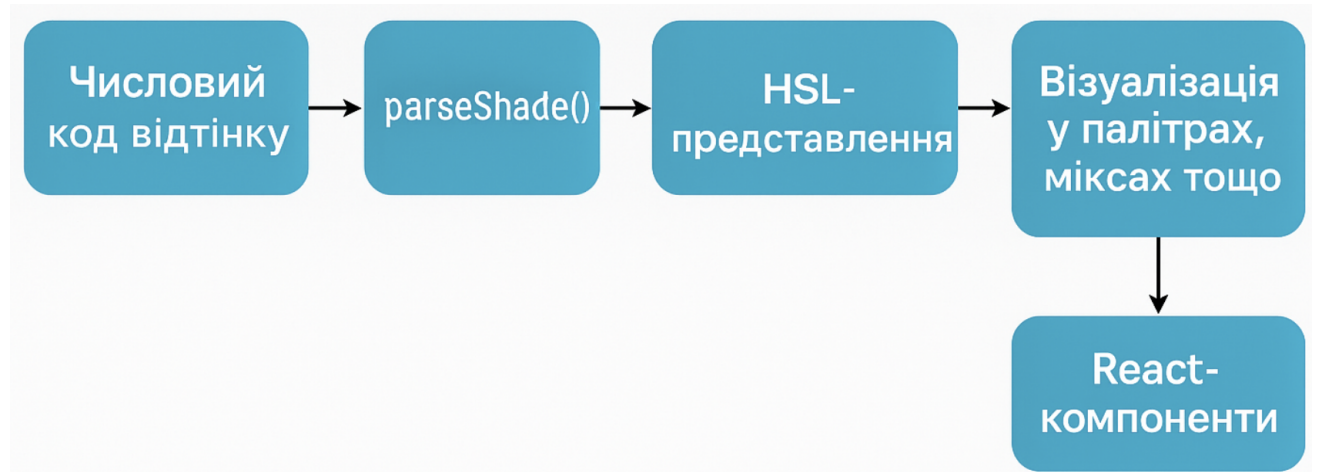


Рисунок 1.3. Обробка кольору в системі: від коду до відображення

Розглянемо приклад функції відображення:

```
export function tryParseShade(input: string): { tone: number; nuance: number | null } | null {
  | null {
    const match = input.trim().match(/^(\\d)(?:\\.\\d{1,2})?$/);
    if (!match) return null;

    const tone = parseInt(match[1], 10);
    const nuance = match[2] ? parseInt(match[2], 10) : null;

    if (tone < 1 || tone > 10) return null;
    if (nuance !== null && (nuance < 0 || nuance > 99)) return null;

    return { tone, nuance };
  }
}
```

Функція `tryParseShade` є початковим етапом обробки введеного користувачем значення відтінку. Вона отримує на вхід рядкове значення, яке може містити лише тон (наприклад, "7") або поєднання тону та нюансу (наприклад, "7.34", "9.1"). Основне її завдання, це виділити ці числові частини з рядка та перевірити їхню відповідність очікуваним межах. У випадку, якщо формат запису не відповідає шаблону або значення виходить за допустимі межі, функція повертає `null`, що сигналізує про помилку введення.

У разі правильного введення функція успішно розділяє тон і нюанс, перетворює їх на числові значення та повертає у вигляді об'єкта. Таким чином, вона виконує роль фільтра та парсера одночасно: забезпечує валідацію формату й готує структуровані дані для подальшої обробки. Реалізація цієї функції дозволяє централізовано керувати правильністю введених значень і гарантувати

стабільність наступних етапів обчислень.

Також розглянемо функцію конвертації у колір (повний текст коду див. у додатку А):

```
export function hsl({ tone, nuance }: { tone: number; nuance: number | null }): string {
  const hue = toneToHue(tone);
  const saturation = nuance !== null ? 60 + nuance % 40 : 50;
  const lightness = 100 - tone * 7;

  return `hsl(${hue}, ${saturation}%, ${lightness}%)`;
}
```

Функція `hsl()` відповідає за перетворення розібраного коду відтінку на формат кольору, який може бути відображено у веб-браузері. На основі значень тону та нюансу вона будує колір у просторі HSL, який є зручним для контролю за відтінком, насиченістю та світлістю. Параметр `hue` визначається залежно від тону він вираховується за допомогою окремої функції, яка пов'язує глибину тону з відповідним сектором кольорового кола.

Насиченість (`saturation`) та світлість (`lightness`) розраховуються динамічно: насиченість збільшується або зменшується залежно від наявності нюансу, а світлість змінюється обернено пропорційно глибині тону. Формула дозволяє відтворити реалістичне візуальне уявлення кольору, наближене до професійного бачення майстра-колориста. Отримане значення у форматі `hsl(...)` безпосередньо використовується у CSS або стилізації React-компонентів.

Важливою складовою інтерфейсу є візуалізація кольорів у вигляді кола або сітки. Для цього використовуються SVG-компоненти, які генеруються динамічно залежно від введених параметрів. Кожен сектор відтінку має свій кут, колірну заливку та подію кліку, що дозволяє реалізувати повноцінну палітру з можливістю вибору, порівняння та комбінування кольорів.

Окрема логіка відведена функціоналу змішування кольорів. У проєкті реалізовано функцію, що приймає два коди відтінків, конвертує їх у HSL і обчислює середнє значення тону та насиченості. Функціонал бібліотеки дозволяє побудувати результат змішування, який виводиться в окремому полі попереднього перегляду. Також функціонал є ключовим для симуляції професійного підходу до створення рецептури фарбування.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

У проєкті активно застосовується контекст (React Context) для передачі глобальних кольорових налаштувань між компонентами.

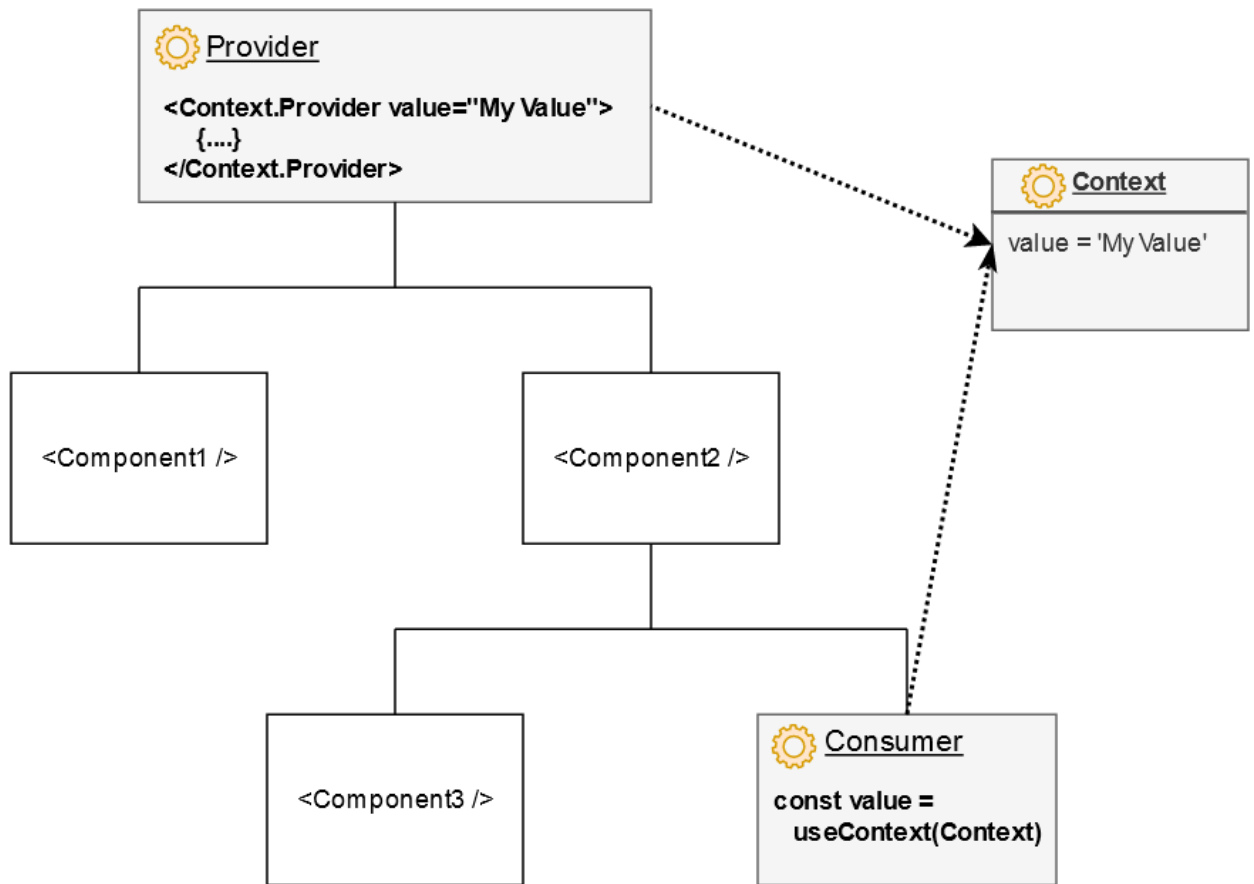


Рисунок 1.3. Блок схема використання React Context

Наприклад, вибраний тон або група відтінків зберігається на рівні провайдера і доступна у будь-якій частині інтерфейсу. Бібліотека синхронізувати відображення кольору у палітрі, формі введення та результатах одночасно без потреби у дублюванні логіки.

Для стилізації використовуються змінні кольору в SCSS, що динамічно підставляються з JavaScript. Вона змогу підтримувати гнучку тему інтерфейсу (наприклад, зміну основного кольору системи залежно від вибраного відтінку або настрою) і створювати повну кольорову ідентичність продукту. Крім того, адаптація до темної/світлої теми реалізована через CSS media queries і параметри контексту.

Особливу увагу приділено доступності кольорових елементів таких як

система яка враховує достатній контраст між кольорами тексту та фону, а також використовує альтернативні підписи й пояснення для користувачів із порушеннями сприйняття кольору. Як зазначає Річард Рутер у своїй книзі: «Інтерфейс, який не враховує контрасту, масштабу, шрифтів і кольору, ніколи не буде однаково доступним для всіх користувачів» [4]. Вбудування кодової бази допомагає налагодити відповідність сучасним вимогам доступності інтерфейсів і робить продукт інклюзивним.

Загалом, архітектура кольорової системи в React-додатку будується на поєднанні точних обчислень (HSL, перетворення кодів), компонентного підходу та гнучкої візуалізації. Підхід дозволяє створити потужний інструмент для колористів, що відповідає професійним потребам і підтримує візуальну ясність, динаміку та гнучкість взаємодії з кольором.

1.3.5 Моделювання взаємодії кольорів у цифровому середовищі

Одним із ключових викликів при створенні експертної системи для колористів є моделювання взаємодії кольорів, що у фізичному середовищі базується на досвіді майстра, а в цифровому повинно бути описане через математичні правила. У реальних умовах фарби змішуються в певних пропорціях, що дає конкретний результат. Аналогічно в інтерфейсі веб-застосунку необхідно реалізувати модель, яка дозволяє наочно побачити результат комбінації двох або більше відтінків у віртуальній палітрі.

В основі цифрової симуляції лежить розрахунок усереднених параметрів кольору. Найчастіше використовується кольорова модель HSL, яка дозволяє окремо працювати з трьома параметрами відтінком (hue), насиченістю (saturation) та світлістю (lightness). Це робить можливим точне математичне поєднання кольорів, наближення до візуально зрозумілих результатів, а також зручну візуалізацію у браузері.

Для базового змішування двох відтінків застосовується проста формула середнього арифметичного. Наприклад, при змішуванні кольорів $A(h_1, s_1, l_1)$ і $B(h_2, s_2, l_2)$, система обчислює результат як:

$$H = (h_1 + h_2) / 2, S = (s_1 + s_2) / 2, L = (l_1 + l_2) / 2 \quad (1.1)$$

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Така формула дозволяє швидко згенерувати проміжний відтінок, який відображає результат комбінування. Конкретно до цього обчислення можуть застосовуватись коригувальні коефіцієнти для імітації реального хімічного змішування.

Окрім змішування, система також передбачає можливість аналізу комплементарних кольорів. У контексті HSL це реалізується як поворот відтінку на 180 градусів на кольоровому колі. Наприклад, якщо базовий колір має значення $h = 30$, то його комплементарний відтінок буде $h = 210$. Подібна логіка дозволяє реалізувати візуальні підказки щодо контрасту, поєднань та протилежностей між відтінками.

Ще одним інструментом візуального моделювання є аналогові схеми система визначає відтінки, які знаходяться поруч на колі, наприклад, $h \pm 30^\circ$. Використовуючи даний функціонал можливо швидко сформувати палітру, яка буде гармонійно виглядати візуально, і яку фахівець може використати для складних багатотональних фарбувань.

Окрему роль у системі відіграє можливість побудови контрастних поєднань. Формування палітри допомагає при складанні візуально цікаве рішення, але й корисний інструмент у тих випадках, коли колорист прагне створити виразний ефект або відтінити ключову зону. Для побудови контрасту в системі враховується не лише відмінність відтінків (hue), а й насиченість та світлість. Наприклад, два кольори з близьким hue, але різкими відмінностями у lightness, все одно можуть виглядати контрастно.

Інтерфейс підтримує візуальну побудову градієнтів, що базуються на поступовому переході між двома або більше HSL-кольорами. Це корисно для імітації реального фарбування або побудови ефектів розтяжки кольору. Технічно градієнт формується через лінійне інтерполювання значень h , s та l між двома кодами відтінків, розрахованими з допомогою `tryParseShade()` і `hsl()`.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

HSL

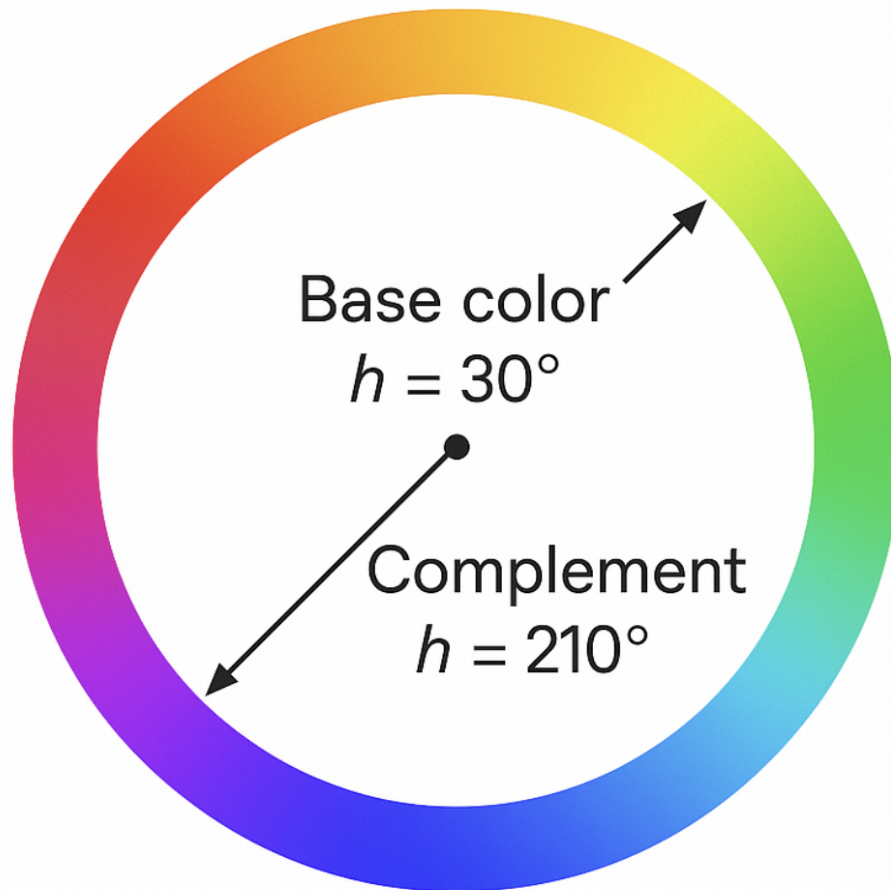


Рисунок 1.3. Коло HSL з компліментарними парами

У блоках, де система відображає результат змішування або підказує комплементарність, колір не є лише декоративним, де він є джерелом інформації. Кольори впливають на сприйняття бренду, створюють емоційний фон і можуть навіть змінити рішення користувача ще до того, як він прочитає текст [5]. Саме тому такі блоки завжди супроводжуються не лише кольоровим фоном, але й текстовим поясненням або кодом відтінку. За допомогою цього алгоритму зробити інтерфейс більш універсальним і зручним для різних категорій користувачів.

Моделювання взаємодії кольорів відбувається також на рівні попереднього перегляду користувач бачить, як обрані відтінки поєднуються між собою ще до того, як зберегти або затвердити змішування. Це реалізується через візуальну панель Preview, що реагує на зміну параметрів у режимі реального часу. Такий підхід формує зручне і гнучке середовище для експериментів із кольором.

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Загалом, моделювання взаємодії кольорів є центральною функцією системи, яка поєднує професійні знання з математичними алгоритмами. Користувач має змогу не лише бачити кінцевий результат, а й розуміти, як він сформувався. Такий прозорий підхід до візуалізації кольору підвищує довіру до цифрового інструменту та сприяє його ефективному застосуванню у практиці колориста.

1.2 Проектування додатку колориста

1.2.1 Огляд архітектури додатку

Архітектура застосунку побудована на основі бібліотеки React, яка забезпечує гнучкий підхід до розробки інтерфейсів завдяки компонентному принципу. У межах даного проєкту структура реалізована таким чином, що кожен логічний блок або окрема функціональність оформлена у вигляді незалежного компонента з чітко визначеною зоною відповідальності. Використання цієї архітектури корисно для масштабування системи, повторно використовувати код і забезпечити легкість в обслуговуванні.

Центральним елементом програми є файл App.tsx, який виступає у ролі вхідної точки додатку та організовує взаємодію між іншими компонентами. У ньому розташовані основні контейнерні блоки, а також реалізоване керування станом сторінки. Звідси здійснюється виклик ключових модулів, відповідальних за введення відтінків, побудову палітри, змішування кольорів та візуалізацію результатів.

Усі візуальні елементи згруповані у директорії components, де кожен файл відповідає за окрему частину інтерфейсу. Наприклад, ShadeInput.tsx відповідає за введення кольору користувачем у форматі цифрового коду (на кшталт “8.1” або “6.34”). У цьому компоненті реалізовано валідацію введення та інтерактивну зміну стану, що миттєво впливає на загальну палітру. Компонент можливо миттєво додавати до відображення кольору.

Компонент ColorResult.tsx обробляє виведення кольору, отриманого внаслідок парсингу або змішування. Він відповідає за візуальне представлення

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

результату, як у вигляді кольорового блоку, так і через текстовий опис.

Окрему функціональну роль виконує `MixingBlock.tsx`, який забезпечує логіку змішування двох кодів кольорів. У цьому компоненті реалізовано обчислення середнього значення параметрів HSL та побудова результату у візуальній формі. Цей блок особливо важливий для відображення професійного підходу до роботи з рецептурами фарбування, де точність і передбачуваність результату є критичними.

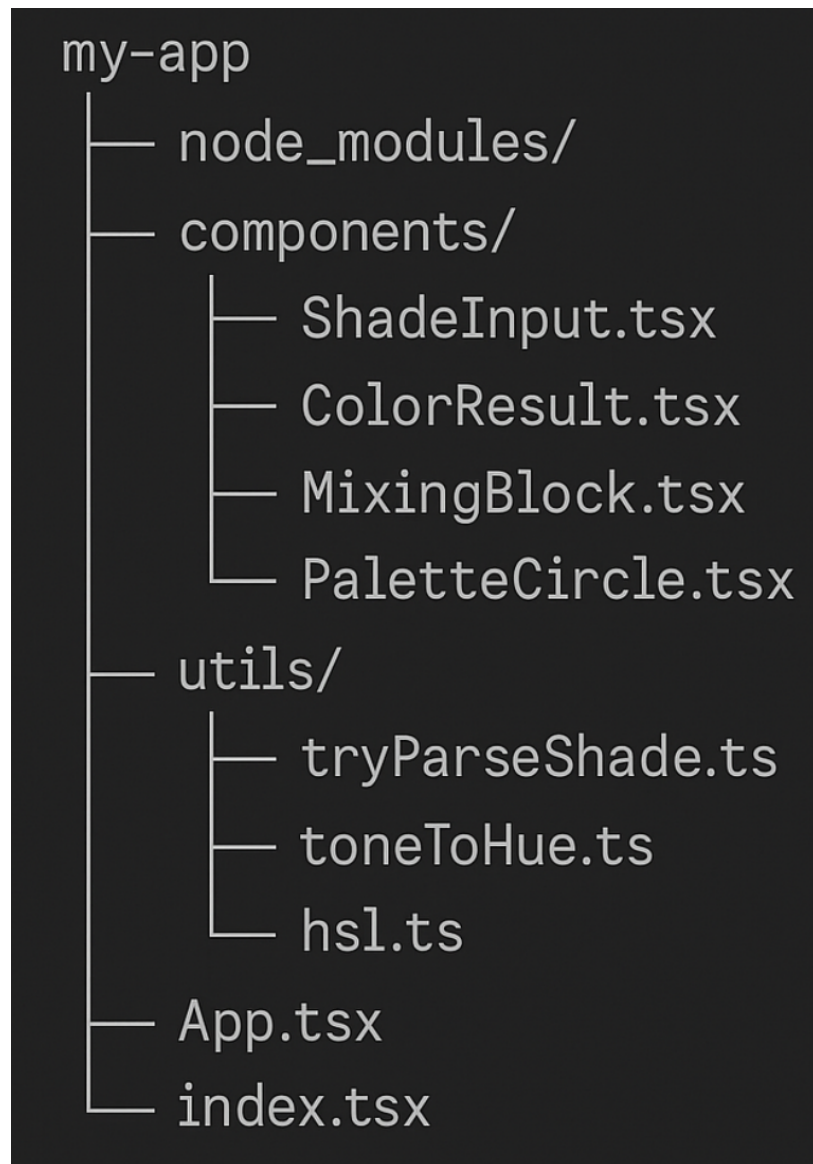


Рисунок 1.4. Огляд архітектури та базових компонентів

Для візуального представлення всіх доступних кольорів застосовується компонент `PaletteCircle.tsx`, який створює кольорове коло. Він динамічно генерує набір відтінків, розташованих за принципом HSL, і дозволяє користувачеві

обирати будь-який відтінок для аналізу або змішування. Компонент працює на базі SVG-елементів, що забезпечує високу якість візуалізації.

На рівні логіки вся обробка кольору зосереджена в папці `utils`. Саме тут розташовані основні функції: `tryParseShade.ts`, `toneToHue.ts` та `hsl.ts`. Перша з них відповідає за розбір введеного цифрового коду та перевірку його коректності. Друга за трансляцію глибини тону у значення кута на HSL-колі. А третя за побудову фінального кольору для відображення в браузері.

Розділення логіки та інтерфейсної частини дозволяє дотримуватись принципів чистої архітектури. Компоненти відповідають лише за візуальне відображення, тоді як логіка обчислень повністю відокремлена та може бути повторно використана або протестована незалежно від UI. Сама бібліотека корисна при високій модульності застосунку та його легкому тестуванню.

Загальна структура побудована з урахуванням масштабування кожен компонент написаний у вигляді ізольованого модуля, що дозволяє без змін у структурі додавати нові функції, наприклад, розширення палітри, додавання шаблонів кольорових рішень або інтеграцію з базою даних кольорів. Використання модульності є особливо важливим у проектах, що орієнтовані на подальший розвиток.

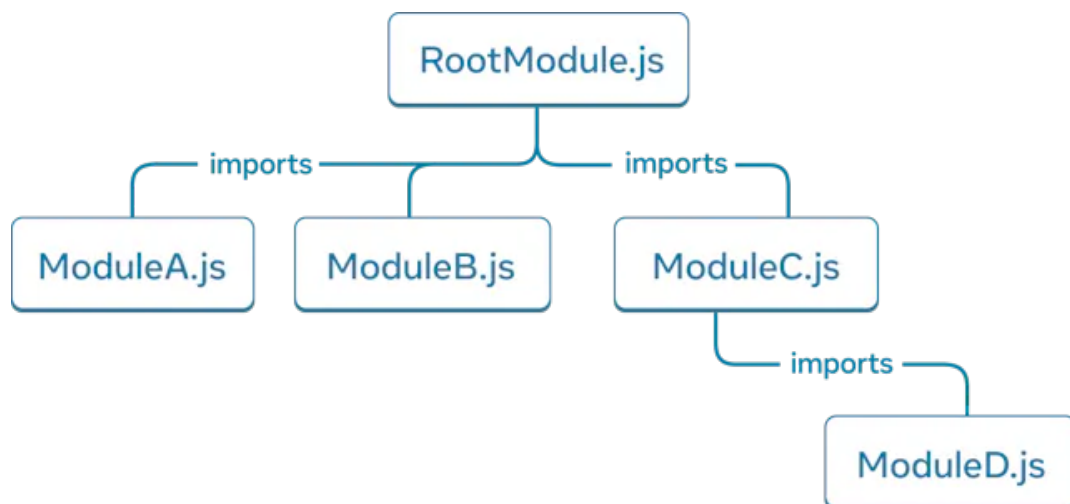


Рисунок 1.5. Приклад архітектури ізольованого модуля

Можемо казати, що архітектура застосунку є прозорою, логічною та адаптованою до потреб як фронтенд-розробки, так і професійного застосування у сфері колористики. До розподілу відповідальностей, компонентній організації та

чіткій структурі каталогів, система демонструє високу гнучкість і готовність до подальшого функціонального розширення, також цю імплементацію можливо зареалізувати у багато інших організацій. Розглянемо наступну таблицю опису файлів та їх призначення у проекті:

Таблиця 1.2 Структура каталогів проекту

Файл / Компонент	Призначення
App.tsx	Головна точка входу, що підключає всі компоненти та управляє станом
index.tsx	Ініціалізація React-додатку, рендеринг у DOM
ShadeInput.tsx	Компонент для введення цифрового коду відтінку та його валідації
ColorResult.tsx	Відображення обчисленого кольору у вигляді блоку з кодом
MixingBlock.tsx	Компонент для поєднання двох кольорів і виведення результату
PaletteCircle.tsx	Побудова візуального кола кольорів, інтерактивний вибір відтінку
tryParseShade.ts	Функція розбору й перевірки коректності введеного коду відтінку
hsl.ts	Обчислення фінального кольору у форматі hsl(...) для стилізації компонентів
toneToHue.ts	Перетворення глибини тону у кут hue для HSL-представлення

1.2.2 Алгоритми аналізу відтінків у колористиці

Система, розроблена в межах даного проекту, має не лише інструменти для побудови палітр, але й здатна виконувати первинний аналіз введених відтінків. Умовно її можна порівняти з порадином-колористом, який орієнтується у взаємозв'язках кольорів і вміє розрахувати, наскільки два відтінки гармонують між собою, чи утворюють контраст, або чи є потенційно комплементарними. Ця логіка є частиною алгоритмічного ядра програми, яке базується на аналітичному обчисленні кольорової відстані та структури палітри.

Користувач вводить відтінок у вигляді цифрового коду, наприклад, "6.34". Система перетворює його у числові параметри тону та нюансу, а далі у формат HSL. На цьому етапі вступає в дію алгоритм обчислення колористичної відстані,

що дозволяє визначити, наскільки два кольори подібні або відмінні. Базова формула, що використовується, є варіацією на тему евклідової відстані, адаптованої до HSL-простору:

$$D = \sqrt{[(h_1 - h_2)^2 + (s_1 - s_2)^2 + (l_1 - l_2)^2]} \quad (1.2)$$

Цей показник D використовується як числовий еквівалент візуальної відмінності між кольорами. Чим менше значення, тим ближчі відтінки між собою. На практиці це дозволяє не лише побудувати градієнт, а й рекомендувати схожі відтінки, що зручно для сценаріїв, де необхідно зберігати певну гармонію.

Далі до функції додамо визначення вектору напрямку зміни між кольорами. Система не лише бачить різницю, але й обчислює, в якому напрямку слід зсунути тон або насиченість, щоб наблизити один відтінок до іншого. Це реалізовано за допомогою формули:

$$\Delta v = (h_2 - h_1, s_2 - s_1, l_2 - l_1) \quad (1.3)$$

Вектор використовуються в системі для анімацій переходів, побудови динамічних градієнтів, а також у підказках для користувача, наприклад: "Підвищте насиченість на 10%, щоб наблизити відтінок до референсу."

Покращенням алгоритму є модуль розпізнавання конфлікту кольорів. Система здатна виявити, коли два відтінки мають надто малий контраст у світлості ($lightness < 15\%$) або у насиченості ($saturation < 10\%$) і повідомити користувача, що такий вибір може призвести до «візуального злиття». Даний алгоритм виступає особливо актуально при побудові двотонових рішень або підборі базового і додаткового кольорів.

У разі виявлення конфлікту система пропонує альтернативні варіанти з кращим контрастом або гармонійністю. Це підвищує якість кінцевого дизайну та зменшує потребу у ручному коригуванні. Алгоритм працює в реальному часі, забезпечуючи інтерактивну взаємодію з користувачем. Таким чином, користувач отримує не лише інструмент для візуального конструювання палітр, а й інтелектуального асистента з колористики.

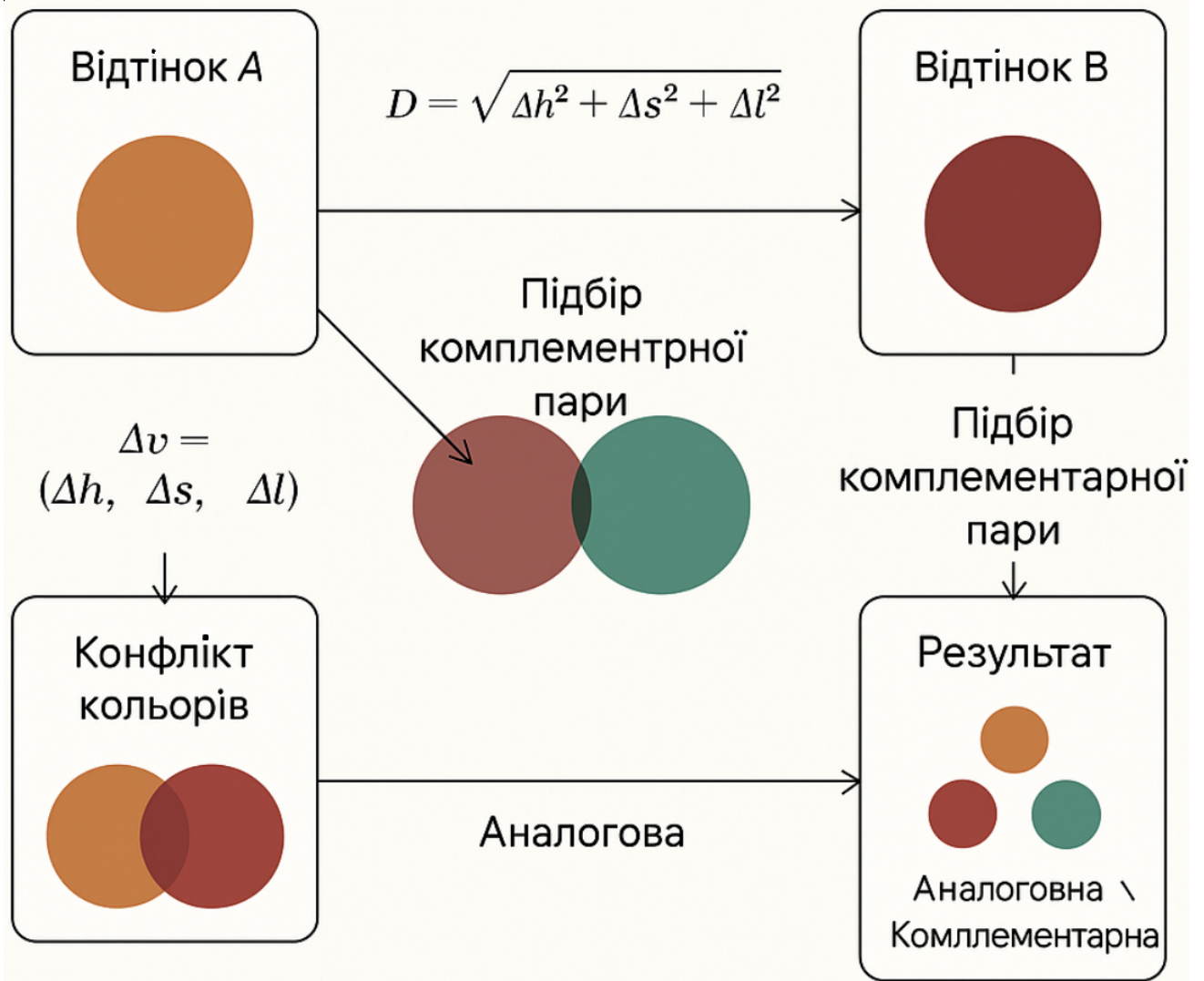


Рисунок 1.6. Алгоритм аналізу відтінків у колористиці

Далі у проєкті, реалізована функція автоматичної побудови комплементарної пари. Вона не просто обчислює $h + 180^\circ$, а враховує контекст тобто, чи не перетинається новий відтінок з уже присутнім у палітрі, чи не суперечить гармонії. У разі виявлення потенційного конфлікту система пропонує альтернативу в межах $\pm 10^\circ$ навколо ідеального комплементарного кута. Довання компліментарності підтримати дизайнерську логіку і водночас адаптувати систему до практичних потреб.

Інтеграція цих алгоритмів із React-компонентами дозволяє реалізувати інтерактивний аналіз кольору. Наприклад, при виборі другого відтінку в блоці змішування система автоматично порівнює його з першим і виводить візуальні підказки: чи буде результат контрастним, аналоговим чи комплементарним.

Розуміння структури пігменту та правильна оцінка бази кольору є фундаментом професійної колористики й гарантією передбачуваного результату [6].

Алгоритм також реалізує просту форму рекомендаційної логіки, яка на основі вже вибраного кольору підказує найпопулярніші поєднання, ґрунтуючись на наборі заздалегідь заданих схем (наприклад, «тепла триада», «прохолодна пара», «нейтральна база»). Це не лише допомагає майстру, а й скорочує час підбору у типових сценаріях.

Розроблений алгоритм не є жорстко закодованим, та він легко масштабується. У перспективі до нього можна інтегрувати машинне навчання, що буде навчатися на реальних палітрах та комбінаціях, обраних користувачами. Але навіть у поточному вигляді логіка аналізу відтінків є вагомим досягненням, що дозволяє вийти за межі базової колірної інтерпретації та наблизити інтерфейс до повноцінного колористичного асистента.

1.2.3 Опис логіки обробки коду відтінку

У системі колористичного аналізу ключовим входом є відтінок у вигляді цифрового коду, наприклад: "7.34", "9.1" або просто "6". Такі записи відображають поширену практику у перукарській справі, де перше число позначає глибину тону, а друге це нюанс або відтінковий нюанс фарби. Щоб працювати з такими значеннями в цифровому середовищі, необхідно не лише їх прийняти, а й перевірити, коректно розбити та перетворити в параметри для подальших обчислень. Виконаний етап є фундаментальним для запуску алгоритмів візуалізації та змішування.

У проєкті логіка розбору такого запису реалізована у функції `tryParseShade`, яка знаходиться у файлі `utils/tryParseShade.ts`. Вона перевіряє формат введеного рядка, використовуючи регулярний вираз:

```
const match = input.trim().match(/^(\d)(?:\.\d{1,2})?$/);
```

Шаблон дозволяє приймати лише ті записи, де передбачено одну цифру до крапки (тон) і максимум дві цифри після неї (нюанс). Якщо запис не відповідає шаблону та повертається `null`, що сигналізує про помилку введення. В іншому

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

випадку, значення тон і нюанс перетворюються в числа та повертаються у вигляді об'єкта.

Наприклад, в результаті виклику:

```
tryParseShade("7.34");
```

отримуємо:

```
{ tone: 7, nuance: 34 }
```

Цей об'єкт вже можна використовувати у наступних обчисленнях. Наступний крок це обчислення кольору у форматі HSL. Для цього в `utils/hsl.ts` реалізовано функцію `hsl()`, яка приймає об'єкт `{ tone, nuance }` і повертає рядок із CSS-форматом `hsl(...)`(повний текст коду див. у додатку А):

```
export function hsl({ tone, nuance }: { tone: number; nuance: number | null }): string {
  const hue = toneToHue(tone);
  const saturation = nuance !== null ? 60 + nuance % 40 : 50;
  const lightness = 100 - tone * 7;

  return `hsl(${hue}, ${saturation}%, ${lightness}%)`;
}
```

Тут `toneToHue()`, це окрема функція, що зберігається у `utils/toneToHue.ts` і визначає базове значення відтінку на кольоровому колі. Наприклад, тон 6 може відповідати помаранчевому спектру, а тон 9 до золотистому. Даний функціонал допомагає закодувати кожен професійний тон у відповідну зону `hue`.

Функція `hsl()` також додає нюансу модифікацію насиченості (`saturation`), щоб візуально відобразити різницю між, скажімо, 7.1 і 7.34. Світлість (`lightness`) визначається пропорційною: чим нижчий тон тим темніший колір. Наприклад, тон 1 має світлість близько 30%, а 10 - до 95%. Оброблений рядок `hsl(...)` використовується у стилях компонентів наприклад, у `ColorResult.tsx`, де результат кольору відображається у стилізованому блоці:

```
<div className="color-box" style={{ backgroundColor: hsl(parsedShade) }} />
```

Цей фрагмент забезпечує інтерактивне відображення кольору безпосередньо після введення. Система реагує в реальному часі, без потреби в натисканні кнопок чи перезавантаженні сторінки. Функціонал особливо важливий для професіоналів, які звикли працювати швидко та інтуїтивно.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30



Рисунок 1.7. Вид введення коду до стилізації

Усі обчислення кольору централізовано реалізовані в `utils`, що дозволяє використовувати логіку не лише в одному компоненті, а й повторно у `MixingBlock.tsx`, `PaletteCircle.tsx`, а також потенційно у функціях збереження історії або побудови рекомендацій. Він відповідає принципам DRY і робить систему більш стійкою до змін, та у подальшому можливо запровадити нові функції до додатку.

Щоб зробити роботу з кодами відтінків стабільною, у проєкті реалізована додаткова обробка невалідних або частково введених значень. Наприклад, якщо користувач вводить лише "8." або "4.", без вказання нюансу, система все одно інтерпретує це як допустимий ввід, при цьому встановлюючи nuance у null. Це дозволяє працювати навіть з неповними значеннями без втрати стабільності й не примушує користувача до суворої форми вводу. Розглянемо наступну таблицю кольорів:

Таблиця 1.3 Відповідність цифрового коду відтінку, значення HSL та візуального результату

Код відтінку	HSL-представлення	Візуальний колір (опис)
6.34	hsl(200, 74%, 58%)	Насичений попелясто-мідний
9.1	hsl(60, 70%, 37%)	Світлий попелясто-бежевий
7.0	hsl(180, 60%, 51%)	Природний середній коричневий
4.65	hsl(270, 85%, 35%)	Насичений фіолетовий відтінок
10.3	hsl(30, 63%, 30%)	Темний золотисто-русявий

Прощення введення сприяє покращенню UX де користувач не отримує помилки одразу, а бачить результат навіть проміжного стану. З технічного боку це реалізується в `ShadeInput.tsx`, де компонент реагує на кожну зміну поля вводу і викликає `tryParseShade(value)` у режимі `debounce` або в залежності від події `onChange`.

До обробки введення, система підтримує візуальне попередження про помилки. Наприклад, якщо `tryParseShade()` повертає `null`, компонент автоматично застосовує до поля клас `input-error`, змінюючи рамку на червону. Логіка реалізується через перевірку `isValidShade(value)` і дає можливість миттєво зорієнтувати користувача без додаткових підказок.

Оскільки результат парсингу далі передається у функцію `hsl()`, обробка `null` у `nuance` має бути коректною. У проєкті це реалізовано обережно: якщо нюанс відсутній, система використовує за замовчуванням значення насиченості

(saturation = 50%) для побудови HSL. Це дозволяє уникнути помилок undefined і зберегти узгодженість у відображенні.

Можна зазначити, що логіка побудови hsl(...) винесена у функцію hsl.ts, що дозволяє централізовано керувати стилістичними параметрами. Наприклад, якщо в майбутньому з'явиться потреба у темнішому або теплішому стилі, достатньо буде змінити правила перетворення у цій функції, і це автоматично позначиться на всіх компонентах, що використовують HSL-колір.

Важливим моментом є тестування обробки коду. Завдяки модульній структурі, функції tryParseShade та hsl можна покрити юніт-тестами без залучення інтерфейсу. За допомогою алгоритму можна перевірити правильність розпізнавання, обробки edge-case'ів (наприклад, "11", "5.999", "8.xx") і впевнитися, що система не дає збоїв при неочікуваному ввході.

Функція toneToHue(), що використовується у зв'язці з hsl(), також є ключовим елементом перетворення. Вона містить відповідність між цифровим тоном (1–10) і hue за кутом на кольоровому колі. Наприклад (повний текст коду див. у додатку А)::

```
const toneMap = {
  1: 240, // темно-синій
  2: 220,
  3: 200,
  ...
  9: 40, // мідний
  10: 20 // золотисто-жовтий
};
```

Функціонал дозволяє візуально кодувати глибину тону на HSL-колі, і робить результат зрозумілим для користувача, що працює з професійними фарбами. Кожне число "6", "9", "10" має свою візуальну асоціацію.

Вся логіка обробки коду кольору працює в реальному часі. Це означає, що зміна значення у полі викликає перерахунок парсингу, HSL і відображення та без ручного підтвердження. У додатку створюється відчуття миттєвого зв'язку між дією і результатом, що надзвичайно важливо в інтерфейсі для роботи з кольором.

Варто зазначити, що поточна реалізація дозволяє не лише відображати один колір, а й використовувати оброблений код для побудови системи змішування, генерації комплементарних відтінків і навіть для побудови рекомендацій. Він

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

робить обробку коду не просто технічною операцією, а ядром логіки, навколо якого обертається весь функціонал системи.

Далі, логіка обробки коду відтінку в проєкті реалізована як послідовність: валідація → парсинг → переведення у HSL → відображення. Кожен крок виконується ізольовано, що полегшує відлагодження та тестування. Результат це зрозуміла, передбачувана поведінка системи, яку легко масштабувати і вдосконалювати у подальшому.

1.2.4 Змішування кольорів у застосунку

Функціональність змішування відтінків реалізована як окремий логічний блок системи і відіграє центральну роль у симуляції реального процесу колористики. На відміну від простого вибору кольору, змішування передбачає обчислення нового результату на основі двох введених кодів фарби. Достатньо відіграє колористу оцінити потенційний результат ще до застосування його на практиці.

Компонент, який відповідає за цю логіку, називається `MixingBlock.tsx`. Він містить два поля для введення кодів відтінків ці кольори `shadeA` і `shadeB`. Кожне поле обробляється через `tryParseShade`, яка повертає об'єкт `{ tone, nuance }`. Усі наступні розрахунки виконуються лише у разі, якщо обидва введення є валідними.

Ключова логіка побудови результату базується на обчисленні середнього значення кожного з параметрів HSL. Як тільки обидва відтінки успішно розпізнані, система викликає наступну обчислювальну функцію (повний текст коду див. у додатку А):

```
function mixHSL(a: HSL, b: HSL): HSL {
  return {
    h: (a.h + b.h) / 2,
    s: (a.s + b.s) / 2,
    l: (a.l + b.l) / 2,
  };
}
```

У загальному формула дозволяє створити проміжний відтінок, що відповідає логіці «1:1» змішування фарб. Варто зазначити, що у реальному світі хімія не завжди підкоряється цим правилам, проте візуально такий підхід дає передбачуваний і зрозумілий результат для користувача.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

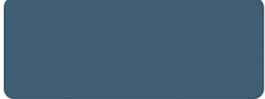
Код відтінків A і B	HSL- репрезентація	Візуальний колір
6.34 + 9.1	hsl(130, 72%, 47%)	
4.65 + 7.0	hsl(225, 73%, 44%)	
5 + 10.3	hsl(195, 56%, 37%)	

Рисунок 1.8. Приклад розрахування кольорів у застосунку

Перед тим, як потрапити до цієї функції, кожен з відтінків кодується у формат HSL через послідовність:

```
const parsedA = tryParseShade(shadeA);
const parsedB = tryParseShade(shadeB);
const hslA = parseToHSL(parsedA);
const hslB = parseToHSL(parsedB);
```

Тут `parseToHSL` це об'єднана функція, яка викликає `toneToHue` для `tone`, обчислює `saturation` і `lightness`, враховуючи `nuance`. Вона зосереджена у файлі `hsl.ts` і забезпечує консистентність побудови кольору по всій системі.

Після обчислення змішаного кольору у форматі `{ h, s, l }`, результат передається у візуальний блок відображення кольору у `MixingBlock.tsx`:

```
<div
  className="mixed-color-preview"
  style={{ backgroundColor: `hsl(${mixed.h}, ${mixed.s}%, ${mixed.l}%)` }}
/>
```

Після додавання цієї логіки користувач миттєво бачить результат змішування. Реактивність реалізована через хук `useEffect`, який відстежує зміни `shadeA` або `shadeB` і автоматично оновлює результат. Це допомагає взаємодіяти з кольором у реальному часі, без потреби ручного оновлення сторінки.

Цікаво, що такий підхід легко масштабувати. Наприклад, можна додати підтримку змішування трьох кольорів або дозволити варіювати пропорції,

застосувавши формули з ваговими коефіцієнтами (повний текст коду див. у додатку А):

```
const mixWeighted = (a: HSL, b: HSL, weight = 0.5) => ({
  h: a.h * (1 - weight) + b.h * weight,
  s: a.s * (1 - weight) + b.s * weight,
  l: a.l * (1 - weight) + b.l * weight,
});
```

У такому варіанті користувач зможе регулювати переважання одного з відтінків. Це значно розширить функціональність і дозволить краще адаптувати систему до реальної практики колориста, де важливо мати контроль над дозуванням фарб.

У проекті, логіка змішування у застосунку - це не лише арифметична операція, а динамічна система, що реагує на введення, генерує передбачуваний результат та створює комфортне середовище для прийняття колористичних рішень. Реалізація через React і модульні функції забезпечують стабільність, масштабованість і гнучкість, необхідні у сучасному веб-інструменті для професіоналів.

1.2.5 Побудова інтерактивної палітри та кольорового кола

Одним із найбільш візуально виразних компонентів застосунку є інструмент побудови інтерактивної палітри: `PaletteCircle.tsx`. Його головна функція полягає у генерації кольорового кола, що представляє доступні відтінки в межах колористичної моделі HSL. Алгоритм побудовано таким чином, щоб дозволити користувачу швидко орієнтуватися в спектрі кольорів, обирати відтінки без потреби вводити їх вручну та досліджувати взаємозв'язки між ними.

Компонент реалізовано на основі SVG-графіки, що дає змогу створювати гнучкі та точні елементи палітри. Кожен сегмент кола генерується циклічно у циклі `for`, де кут повороту визначається значенням `hue` у діапазоні від 0 до 360 градусів. Наприклад, для створення 36 сегментів палітри по 10 градусів система проходить повний колірний круг:

```
for (let h = 0; h < 360; h += 10) {
  const color = `hsl(${h}, 80%, 50%)`;
  // Генеруємо сегмент з відповідним fill
}
```

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Далі будемо вважати, що сегмен - це частина кола, реалізована як `<path>` або `<circle>` у SVG. Стилiзація та позиціонування сегментів здійснюється через обчислення координат на основі радіуса та кута. Кожен сектор є інтерактивним при наведенні або кліку змінює свій стан (наприклад, додається клас `active` або `selected`).

У події `onClick` система фіксує обраний `hue` і зберігає його у локальний стан компонента або через глобальний контекст. Після вибору кольору палітра може передати значення `hue` до інших компонентів, наприклад у форму змішування або `ColorResult.tsx`. Окрема логіка допомагає будувати логічний ланцюг: обрав → переглянув → змішав → зберіг.

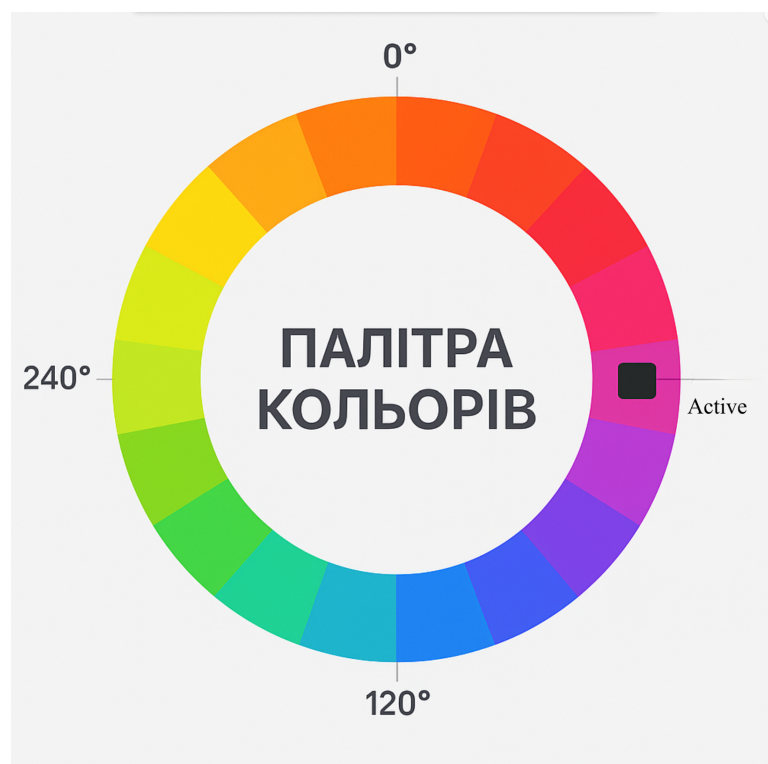


Рисунок 1.9. Схема структури кольорового кола

Окрему увагу приділено UX-реалізації. Колірна палітра повинна не лише точно відображати кольори, але й бути зручною для взаємодії. Тому при наведенні кожен сегмент виділяється через `outline` або тiнь, а обраний сегмент фіксується стилем (наприклад, збільшенням або підсвічуванням рамки). Це дозволяє користувачу легко зрозуміти, який колір наразі активний.

Інтерактивність також підтримується через анімовані переходи. При зміні `hue` система плавно змінює колір попереднього перегляду (`PreviewBlock`), що

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

створює відчуття безперервного і живого процесу. Такий ефект досягається через CSS transitions або бібліотеки типу framer-motion, які можна легко інтегрувати в React-компоненти.

Додатково система підтримує відображення підказок про hue, saturation і lightness для кожного сегмента палітри. При наведенні користувач бачить, наприклад, "H: 210°, S: 80%, L: 50%", що дозволяє краще зрозуміти характеристику відтінку ще до вибору. Така підказка реалізується через title атрибути або окремий tooltip-компонент.

Далі палітру можна розширити до багат шарової: внутрішнє коло, це більш темні відтінки, зовнішнє - світліші або варіації насиченості. Ми можемо надати повноцінну кольорову карту, яка дозволить вивчати взаємозв'язки між групами відтінків без потреби вручну змінювати параметри.

Упровадження кольорового кола значно підвищує зручність роботи в системі, дозволяє користувачу інтуїтивно взаємодіяти з інтерфейсом, а також служить візуальним навчальним інструментом для тих, хто тільки починає працювати з колористикою. Це перетворює застосунок не лише на практичний інструмент, а й на візуальну платформу для експериментів і навчання.

Загалом PaletteCircle.tsx - це більше, ніж просто UI-елемент. Центральна точка взаємодії користувача з системою кольору, яка дозволяє швидко орієнтуватися в колірному просторі, приймати рішення на основі візуальних підказок та взаємодіяти з відтінками на рівні професійної логіки.

1.2.6 Взаємодія компонентів у повному циклі роботи з кольором

Етапом технічного розгляду системи є аналіз того, як її компоненти поєднуються у єдиний робочий механізм. Застосунок, побудований за компонентною архітектурою React, дозволяє кожному модулю відповідати за окрему функцію, при цьому всі вони об'єднуються навколо єдиної логіки взаємодії з кольором. Від моменту введення коду до виведення фінального результату проходить низка узгоджених операцій, що реалізують повний колористичний цикл. Робота системи починається з компонента ShadeInput.tsx, який дозволяє користувачеві ввести цифровий код відтінку. Компонент виконує

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

первинну валідацію за допомогою функції `tryParseShade()`, перевіряючи синтаксис і допустимі межі значень. У разі успішного аналізу тон і нюанс зберігаються в локальному стані або передаються вище через пропси.

Наступний крок, це дати змогу на обчислення HSL-представлення кольору. Це виконується у функції `hsl()` з модуля `utils/hsl.ts`, де на основі тону обирається `hue`, а нюанс впливає на насиченість. Світлість розраховується автоматично з урахуванням глибини тону. Результат використовується для стилізації елементів через CSS-властивість `backgroundColor`. Після введення обох кольорів (у `MixingBlock.tsx`) виконується обчислення нового відтінку за формулою середнього арифметичного параметрів HSL. Компонент містить окрему логіку для обчислення змішаного кольору та виводить його через кольорове поле, стилізоване динамічно. Завдяки React-хукам, перерахунок відбувається автоматично при зміні хоча б одного з вхідних значень.

Компонент `ColorResult.tsx` у свою чергу відображає як окремі кольори, так і результат змішування. Він дозволяє користувачу швидко оцінити візуальний ефект введених даних. Система також може додатково виводити пояснення або попередження, наприклад, при низькому контрасті або відсутності нюансу. Додатковим модулем, який посилює взаємодію, є `PaletteCircle.tsx`. Він працює незалежно від полів введення, однак може бути синхронізований з ними: при виборі сегменту на палітрі відповідний `hue` автоматично підставляється в одне з полів або використовується як орієнтир для змішування. При запиті до двосторонню інтеракцію між візуальним вибором і ручним введенням. Для передачі даних між компонентами використовуються як локальні стани (через `useState`), так і функціональні пропси (`onChange`, `onSelect`). Ми можемо досягти гнучкої зв'язності між частинами застосунку без використання глобального менеджера стану. У разі потреби масштабування, така логіка легко адаптується до `Context API` або `Zustand`.

Увесь колористичний цикл реалізовано реактивно будь-яка дія користувача викликає негайну зміну інтерфейсу. Це особливо важливо у професійній сфері, де зворотний зв'язок має бути миттєвим. Наприклад, при очищенні одного з полів

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

введення система одразу скидає результат змішування та очищає блок перегляду, що запобігає дезінформації. Обробка помилок і невалідного вводу також вбудована у загальну взаємодію. Якщо `tryParseShade()` повертає `null`, компоненти, які залежать від результату, блокують візуалізацію кольору або підставляють дефолтне значення. Це гарантує стабільність навіть при частковому заповненні форми або нетипових ситуаціях.

У загальному, система реалізує повний цикл взаємодії з кольором від введення до результату, від візуального вибору до аналітики. Всі компоненти працюють узгоджено, реагуючи один на одного та забезпечуючи глибоку інтеграцію між логікою, UI та очікуваннями користувача. Завдяки цьому застосунок має завершений вигляд, стабільну архітектуру та високий рівень практичної придатності.

1.3 Тестування та розробка гайду до застосунку

1.3.1 Призначення та загальний огляд інтерфейсу

Розроблений веб-застосунок призначений для підтримки роботи фахівців у сфері колористики, зокрема для візуального аналізу відтінків, симуляції змішування кольорів та підбору комплементарних рішень. Завдяки застосуванню технологій React, інтерфейс реалізовано у вигляді швидкої та інтерактивної системи, що дозволяє оперативно працювати з професійними кодами відтінків, характерними для індустрії краси.

Основне призначення програми надати користувачеві інструмент, який дозволяє вводити цифровий код відтінку (наприклад, "6.34" або "9.1"), отримувати його візуальне представлення та проводити обчислення на основі введених даних. В основі логіки лежить перетворення цифрового запису у формат HSL, що дозволяє точно обробити відтінок у цифровому просторі.

Після запуску веб-застосунку користувач бачить головну сторінку, що умовно поділена на кілька блоків: панель введення відтінку, результат візуалізації кольору, інтерактивну палітру кольорового кола та блок змішування кольорів. Інтерфейс побудований таким чином, щоб забезпечити мінімальну кількість дій

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

для отримання результату усі розрахунки виконуються автоматично в режимі реального часу без необхідності натискати кнопки підтвердження.

Hair Color Expert

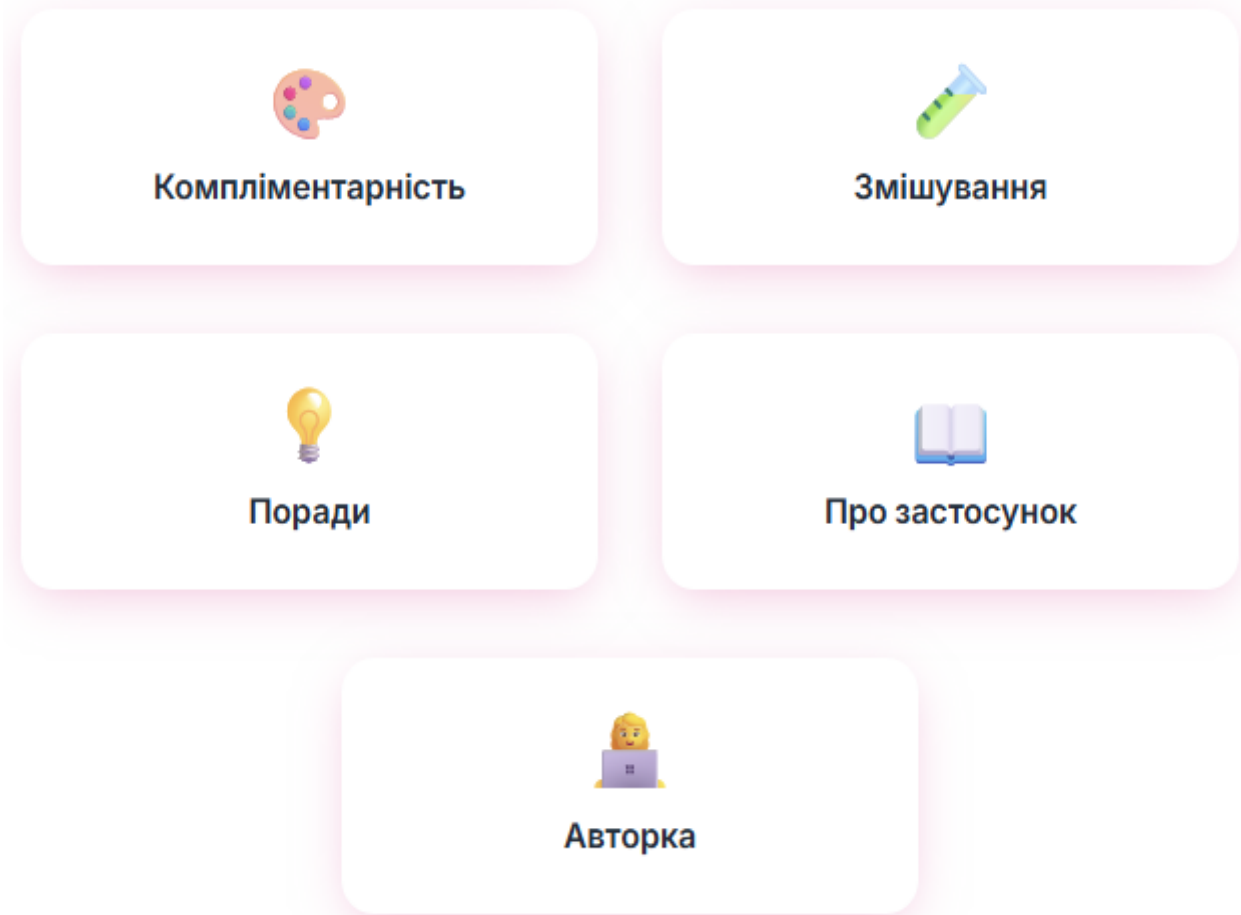


Рисунок 1.10. Головне меню додатку

Кожен блок працює незалежно, однак у взаємозв'язку з іншими. Наприклад, вибраний колір у палітрі може бути автоматично переданий до форми змішування, а змішаний результат виведений у візуальний блок попереднього перегляду. Це створює відчуття єдиної безперервної сесії роботи з кольором.

Інтерфейс побудовано з урахуванням принципів інтуїтивності: усі основні елементи мають зрозумілі позначення, передбачувану поведінку та миттєвий візуальний зворотний зв'язок. Навіть при введенні помилкового значення система або ігнорує його, або пропонує підказки без появи технічних повідомлень або

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

збоїв. Використання такого меню дозволяє працювати з додатком навіть користувачам без досвіду у сфері цифрових технологій.

Унікальною особливістю інтерфейсу є його гнучкість: користувач може взаємодіяти з кольором як через текстове введення, так і через інтерактивну палітру. Це дозволяє адаптувати використання застосунку до різних сценаріїв від швидкого аналізу одного відтінку до глибокого дослідження взаємодії декількох кольорів. Така структура забезпечує комфорт як для досвідчених фахівців, так і для початківців, які тільки знайомляться з професійною колористикою.

Вся взаємодія з системою відбувається без перезавантаження сторінки. Зміни в інтерфейсі відображаються миттєво, що дозволяє не відволікатись від творчого процесу і сконцентруватись на візуальному результаті. Це стало можливим завдяки використанню React-компонентів та ефективного керування станом через хуки. Така технічна реалізація має пряме відображення в зручності для користувача.

Особливо важливо, що застосунок не нав'язує фіксований сценарій використання. Користувач сам вирішує, з чого почати: можна одразу ввести цифровий код і побачити результат, або ж почати з візуального вибору кольору у колі палітри. Така свобода підходів відображає реальні потреби професійної діяльності, де іноді потрібен швидкий результат, а іноді експеримент і візуальний підбір.

Застосунок також враховує принципи візуальної ієрархії: ключові блоки розташовані так, щоб рух погляду був природним зліва направо і зверху вниз. Поля введення кольору знаходяться у верхній частині, що відповідає очікуванням користувача, а результат по центру. Палітра і блок змішування розташовані нижче як додаткові, але логічно пов'язані модулі.

Візуальні елементи реалізовано з урахуванням дизайну для індустрії краси: система використовує м'які тіні, заокруглені кути, комфортні шрифти, що робить інтерфейс естетично привабливим. Це важливо не лише з точки зору користувацького досвіду, а й для відповідності стилістичному контексту салонного середовища, де краса, форма і кольори відіграють ключову роль.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Інтерфейс застосунку є повністю адаптованим до потреб колориста, поєднує точність обробки цифрового коду з гнучкістю візуальної навігації, забезпечує безперервність роботи та сприяє швидкому прийняттю рішень. Використання такого адаптивного дизайну дозволяє ефективно використовувати веб-застосунок як у навчанні, так і у повсякденній професійній діяльності.

1.3.2 Введення кольору у форматі відтінку

Сторінка «Компліментарність кольорів» є однією з найважливіших частин застосунку, оскільки саме тут реалізовано логіку роботи з кодами відтінків і пошуку відповідних компліментарних пар. Основна мета цього блоку дати користувачу можливість ввести відтінок у звичному цифровому форматі, побачити його візуально, а також отримати рекомендацію щодо кольору, який є зовсім протилежним, тобто компліментарним.

Компліментарність кольорів

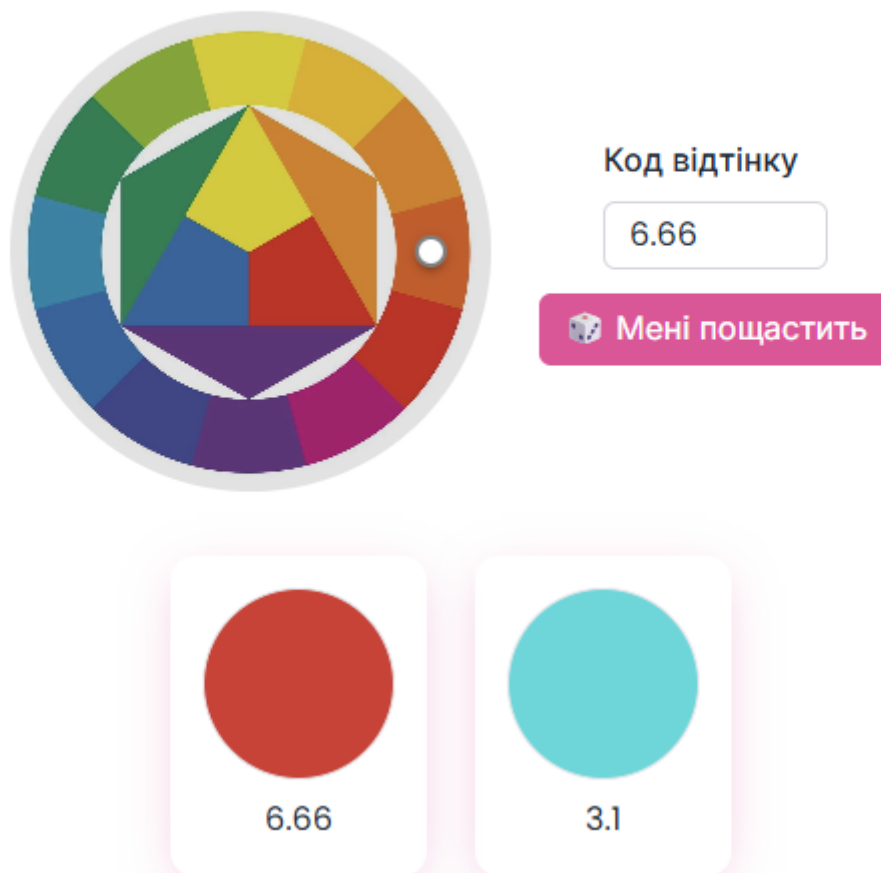


Рисунок 1.11. Сторінка «Компліментарність кольорів»

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

Верхня частина сторінки містить заголовок «Компліментарність кольорів», оформлений у яскравому фіолетово-рожевому відтінку, що добре узгоджується із загальним візуальним стилем сайту. Під ним розташоване інтерактивне кольорове коло з геометричними сегментами, яке демонструє розподіл відтінків по колу HSL. Важливо розуміти що не лише естетичний елемент, а й корисна підказка, що наочно показує, на якому секторі розміщується введений відтінок.

Поруч із кольоровим колом знаходиться поле «Код відтінку», де користувач може вручну ввести значення у форматі професійного запису наприклад, 6.66. Після введення система одразу розпізнає код і обчислює його позицію у HSL-моделі. Важливо, що введення не потребує додаткового підтвердження результат з'являється автоматично, що забезпечує комфортну взаємодію та економію часу.

Крім введення вручну, поруч з полем розміщено кнопку «Мені пощастить», яка виконує функцію випадкової генерації відтінку. При її натисканні система обирає довільне значення відтінку з допустимого діапазону, що може бути корисно для експериментів або навчальних цілей. Ця функція реалізована у легкому стилі, що додає елемент гри й неформального дослідження.

У нижній частині сторінки відображаються два результати: перший це візуальне представлення введеного відтінку, а другий компліментарний до нього. Обидва кольори подано у вигляді великих круглих блоків з плавними тіннями, що добре виділяють їх на фоні. Під кожним колом зазначено цифровий код, який дозволяє чітко ідентифікувати відповідний відтінок.

Наприклад, якщо користувач вводить код 6.66, система виводить цей відтінок зліва, а з правого боку його компліментарний аналог, у цьому випадку 3.1. Це означає, що система автоматично розрахувала відстань у HSL-просторі та знайшла точку, розташовану на 180° навпроти базового hue, з урахуванням відповідної насиченості та світлості. Такий механізм дозволяє користувачу не просто обирати колір наосліп, а керуватися візуальною гармонією.

З точки зору функціональності, компліментарність визначається не лише геометричною симетрією на кольоровому колі, а й адаптацією до колористичного контексту. Наприклад, система автоматично підбирає нюанс таким чином, щоб

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44


результат не виглядав надто агресивним або нерелевантним це робить підказку більш професійною.

Компліментарність кольорів



Код відтінку

2.3

 Мені пощастить



2.3



7.12

Рисунок 1.12. Підбор кольорів

Ще однією важливою деталлю є те, що система дозволяє вводити як повні, так і часткові коди. Наприклад, при введенні лише числа без крапки (6 або 9) програма інтерпретує це як відтінок без нюансу і все одно формує кольорове представлення, використовуючи дефолтне значення насиченості. Це робить взаємодію інклюзивною і допускає гнучкість у форматі запису.

Користувач може повторно ввести інший код або натиснути «Мені пощастить», щоб дослідити інші поєднання. Оновлення кольорів на сторінці відбувається динамічно, без перезавантаження, що забезпечує відчуття безперервної інтерактивної роботи. Усі візуальні ефекти та перерахунки виконуються миттєво, що створює ефект миттєвого зворотного зв'язку.

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Сторінка «Компліментарність кольорів» поєднує технічну точність розрахунків з максимально простим і зрозумілим для користувача інтерфейсом. Вона дозволяє як експериментувати з відтінками, так і приймати обґрунтовані рішення під час підбору кольорових пар, роблячи застосунок зручним інструментом для роботи з гармонією візуального стилю.

1.3.3 Використання блоку змішування кольорів

Розділ «Змішування відтінків» реалізований як окрема функціональна секція на сайті, що дозволяє користувачу створювати новий відтінок шляхом поєднання двох початкових. Функція наслідує логіку професійного змішування барвників у перукарській справі, де майстри комбінують кілька фарб для досягнення бажаного результату. Застосунок дає змогу провести цей процес в інтерактивному цифровому середовищі.

У верхній частині розміщено два поля для введення кодів відтінків. Вони позначені як «Код відтінку» ліве поле відповідає за перший відтінок, праве за другий. Система підтримує формат запису у вигляді десяткового числа, наприклад: 2.12 або 4.43. Введення навіть неповних кодів (лише цифри без дробової частини) також допускається. Як тільки обидва значення розпізнані, система переходить до фази обчислення змішаного результату.

Однією з ключових особливостей цього блоку є наявність повзунка співвідношення, який дозволяє задавати вагу кожного з двох кольорів у результаті змішування. Наприклад, якщо значення встановлено на 45%, це означає, що перший колір буде домінувати на 45%, а другий на 55%. Повзунок реалізовано у стилі Material UI: він кольорово виділений та анімований, що надає інтерфейсу сучасного вигляду.

Розрахунок відбувається шляхом обчислення середньозваженого значення кожного з трьох параметрів HSL hue, saturation, lightness. Приклад формули:

$$\begin{aligned}h &= h1 * p + h2 * (1 - p) \\s &= s1 * p + s2 * (1 - p) \\l &= l1 * p + l2 * (1 - p)\end{aligned}$$

де p - це частка першого кольору в результаті (у відсотках). Завдяки цьому користувач може не лише комбінувати кольори, а й керувати тим, який з них буде

«домінантним» у міксі. Це дуже близько до реального дозування фарб і допомагає зробити підбір більш точним.

Змішування відтінків

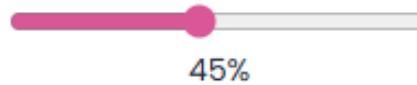
Код відтінку

2.12

Код відтінку

4.43

Співвідношення



45%

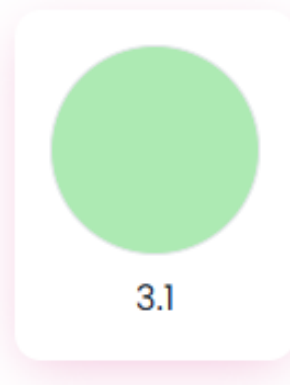


Рисунок 1.13. Змішування відтінків

Нижче повзунка розташовується на сродінки результат змішування - візуально представлений у вигляді кольорового кола на білому фоні з тінню. Колір не тільки допомагає при навігації, а й є частиною комунікації - він задає настрій, підкреслює значущість елементів і підсилює інтерфейс [7]. Цей код може бути використаний повторно наприклад, скопійований у форму підбору компліментарного кольору, або збережений для подальшого порівняння.

Важливо, що результат оновлюється динамічно при кожній зміні вводу чи переміщенні повзунка. Така реактивна поведінка підсилює враження «живого» інтерфейсу й дозволяє проводити експерименти у реальному часі. Гармонійні кольорові схеми базуються на розташуванні відтінків на колесі аналогові, компліментарні чи триадичні пари формуються геометрично точно [8].

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

З точки зору користувача, блок змішування є максимально інтуїтивним: кожен введений відтінок супроводжується миттєвим результатом, а зміна співвідношення реалізована без зайвих кнопок. Це робить процес схожим на реальну роботу з палітрою плавне змішування, контроль за пропорціями, швидкий перегляд результату.

Блок є повністю автономним: навіть при відсутності одного з відтінків або при невірному введенні, інтерфейс не зникає і не показує технічних помилок лише відсутність результату. Використовуючи цю логіку можливо уникати стресу та створює дружню атмосферу взаємодії, що особливо важливо в індустрії краси.

Сторінка «Змішування відтінків» поєднує технічну точність обчислень з високою зручністю для кінцевого користувача. Вона дозволяє майстрам, студентам і навіть початківцям досліджувати кольорові комбінації з повною свободою, і при цьому з контролем над усіма параметрами.

1.3.4. Навігація через кольорову палітру та поради колориста

Інтерактивна палітра є важливою частиною цифрового інтерфейсу, яка забезпечує користувача швидким доступом до широкого спектру відтінків без необхідності вводити числові коди вручну. У верхній частині сайту кольори організовані у вигляді кільця класичне кольорове коло, розбите на сектори відповідно до значень hue у HSL-моделі. Секція дає змогу візуально орієнтуватися у відтінках і швидко обирати бажаний.

Кожен сектор у палітрі відповідає конкретному діапазону відтінків. При натисканні на певну зону система фіксує вибраний hue і відображає його у формі або передає в блоки компліментарності та змішування. Розроблений механізм дозволяє працювати без потреби знати код кожного кольору напам'ять. Він особливо зручно для новачків або в ситуаціях, коли користувач працює з візуальним прикладом, а не з конкретною рецептурою.

Палітра є також навчальним інструментом. При наведенні на сегмент, користувач бачить характеристику кольору hue, його приблизний тон (наприклад, мідний, попелястий, фіолетовий). Підказки допомагають краще зрозуміти, як цифрові параметри співвідносяться з реальними візуальними враженнями. Для

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

початківців це може бути першим кроком до розуміння принципів гармонії, та дуже допомогати при підборі кольорів.

Окрім інтерактивної роботи з кольором, на сайті також передбачено блок із порадами колориста, який розміщено окремо у вигляді сітки з шести інформаційних карток. Даний розділ не просто довідник, а функціональна частина гіда, що надає практичні рекомендації, які можуть бути використані одночасно з роботою в системі. Він поєднує цифрову логіку із знанням про реальну практику фарбування.

Сторінка «Поради колориста» охоплює важливі теми, що стосуються процесу підбору фарби, її нанесення та догляду після процедури. Наприклад, у першій картці описано «Скільки витримувати фарбу» зокрема наведено час витримки залежно від відсотку оксиду.

Поради колориста

Скільки витримувати фарбу

Стандарт – 35 хв на 6 % оксиді. На 9 % – 30 хв, на 3 % – 40–45 хв. Масляні барвники («gloss» / «toner») тримають 15–20 хв.

Тест-прядь

Завжди офарблюйте 1 см² у зоні потилиці, особливо якщо клієнт вперше використовує освітлювач чи змивку.

Підбір оксиданта

1–2 рівні підйому – 3 %, 2–3 рівні – 6 %, понад 3 рівні – 9–12 %. На пористе волосся додавайте Plex та зменшуйте % оксиду.

Передпігментація

Якщо затемнюєте більш ніж на 2 рівні, додайте ¼ золотого або мідного тону до формули, щоб уникнути «болотного» відтінку.

Догляд після фарбування

Перші 48 годин – тільки прохолодна вода та сульфат-фрі шампунь. Раз на тиждень: маска з кислотним рН для закриття кутикули.

Рисунок 1.14. Сторінка «Поради колориста»

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

Інші картки містять поради щодо тест-пряді (рекомендації завжди робити пробу перед фарбуванням), передпідментації (як уникнути небажаного «болотного» кольору при затемненні), підбору оксиданта (в залежності від рівня освітлення), а також догляду після фарбування (особливо в перші 48 годин після процедури). Усі поради викладені коротко, практично, без зайвої термінології саме так, як звикли працювати майстри в салонах. Вона допомагає користувачам одночасно підбирати колір у застосунку та одразу отримувати підказку щодо його технічного застосування.

Блок візуально оформлений у стилі всієї платформи: картки мають м'які тіні, світлий фон, чіткий заголовок і стиснутий, але зрозумілий текст. Додавання функціоналу не тільки зручно читати поради, а й не втрачати логічний зв'язок із загальним інтерфейсом. Елементи розміщено в сітці 2x3, що робить сторінку симетричною, збалансованою й естетично привабливою.

Узагальнюючи, палітра та поради це два додаткові блоки, що працюють паралельно до основного функціоналу. Вони не тільки підсилюють функціональність платформи, але й додають їй елементи освітньої та професійної допомоги. Виконання такого дизайну перетворює застосунок на повноцінний колористичний інструмент, з яким приємно не лише працювати, а й навчатись.

1.3.5 Часті запитання та поради

Застосунок Hair Color Expert створено як інструмент, що поєднує базову теорію кольору та сучасну веб-технологію. Його цільова аудиторія перукарі-колористи, студенти колористичних шкіл і всі, хто працює з фарбуванням волосся або цікавиться гармонією кольору. Завдяки простому, але гнучкому інтерфейсу, платформа дозволяє не лише підбирати відтінки, а й розуміти, чому саме ці поєднання працюють.

Система базується на колористичній моделі Іттена (Oswald/Itten), де все кольорове коло поділено на 12 основних секторів, які включають як базові (червоний, жовтий, синій), так і вторинні та проміжні відтінки. Саме ця модель візуалізована в інтерфейсі, що дає змогу краще орієнтуватися в колірному просторі під час взаємодії з компліментарними парами.

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

Однією з основних ідей застосунку є принцип 180° , що означає: відтінки, розташовані навпроти один одного на колі, взаємно гасять або нейтралізують небажані рефлекси. Наприклад, щоб нейтралізувати фіолетовий підтекст, використовують жовтий, і навпаки. Вона широко використовується у фарбуванні волосся та відображене у логіці блоку «Компліментарність кольорів».

Система використовує цифрове кодування кольорів наприклад, 6.43. Перше число (до крапки) означає глибину тону (1–10), що у застосунку лінійно трансформується у параметр Lightness (L) простору HSL. Цифри після крапки основний і допоміжний нюанс (наприклад, "6.43" = рівень 6, з відтінком copper/gold). На даному етапі система враховує лише перше після крапки значення, але алгоритм побудований так, що може легко бути розширений для роботи з формулами типу 7.30 або 8.12.

Про застосунок

Hair Color Expert — це міні-лабораторія для перукарів-колористів і всіх, хто працює з фарбуванням волосся. Вона перетворює класичну теорію кольору на інтерактивні інструменти, які допомагають *підбирати компліментарні відтінки, моделювати рецепти змішування і нейтралізувати небажані відблиски.*

1. Колористична основа

- **Коло Освальда / Іттена (360°)** — у застосунку відтворено 12 базових секторів (R, Y, B, проміжні і вторинні).
- **Принцип 180° :** діаметрально протилежні відтінки гасять один одного, прибираючи небажаний рефлекс (напр. фіолет \rightleftharpoons жовтий).
- **Глибина тону (1–10)** — перша цифра коду; у нашій моделі лінійно переводиться в Lightness (L) простору HSL.
- **Цифри після крапки** маркують основний і допоміжний нюанс (Ash = 1, Gold = 3, Copper = 4 тощо). Зараз враховується перша, але алгоритм легко розширити до «70/30» або «50/50».

Рисунок 1.15. Приклад гайду для користувача

Цей блок завершує користувацький гайд, пояснюючи теоретичну основу роботи системи, принципи побудови кольору та відповідаючи на ключові

					КГ 08. 04 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

запитання. Застосунок можна ефективно використовувати навіть без попередньої підготовки у сфері колористики.

Під час створення застосунку особлива увага приділялась доступності інтерфейсу навіть для тих користувачів, які не мають ґрунтовної підготовки в колористиці або не знайомі з цифровими технологіями. Інтуїтивна структура сторінок, прості назви полів, візуальні підказки та миттєве оновлення результатів дозволяють працювати із застосунком у режимі реального часу без потреби у навчанні чи вивченні інструкцій.

Користувач має змогу вільно переміщуватись між блоками системи, обирати зручну послідовність дій і самостійно обирати точку входу до роботи. Це можуть бути як формули для змішування, так і готові приклади відтінків із палітри або випадкові варіанти для натхнення. Така гнучкість дозволяє адаптувати платформу до особистого стилю роботи майстра, що є важливим аспектом у творчих професіях.

Застосунок дозволяє поєднувати аналітичну точність обчислень з практичними порадами. Користувач не лише бачить колір, який він отримує внаслідок змішування чи вибору, але й може зрозуміти, як цей відтінок утворився, які відносини він має з іншими відтінками та чому саме він вважається компліментарним. Використання даного підходу робить роботу не механічною, а усвідомленою, що підвищує рівень професійної впевненості.

Усі частини системи від введення коду до виводу результату пов'язані логікою, яка заснована на реальній колористичній практиці. Сама платформа не лише імітує інструменти роботи з кольором, а й створює новий формат цифрової підтримки для колориста, де поєднуються точність, візуальна зрозумілість і практична користь. Саме це робить застосунок повноцінним інструментом, який можна інтегрувати у професійну діяльність або використовувати в навчанні.

					<i>КГ 08. 04 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У дипломному проєкті розроблено веб-застосунок для оптимізації роботи колористів у перукарні, який дає змогу відстежувати процедури, створювати палітри та оцінювати результати. Такий функціонал сприяє підвищенню ефективності роботи та якості обслуговування. Якість системи оцінюється за критеріями зручності використання, продуктивності та практичної цінності.

2.2 Визначення трудомісткості розробки ПЗ

Тривалість розробки програмного продукту залежить від його обсягу, складності, кваліфікації розробників і встановлених ринком термінів. Метод структурної аналогії дозволяє оцінити обсяг у тисячах умовних машинних команд на основі подібного програмного забезпечення.

Табл. 2.1 містить аналоги ПЗ з подібними функціями; обраний варіант виділено сірим.

Таблиця 2.1. Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , умовних. машинних командах
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП введення інформації	1060 – 5750

Після вибору аналога з обсягом V_0 (умовні машинні команди), трудомісткість визначається за табл. 2.2.

Таблиця 2.2. Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262
4.00	283
5.00	306
6.00	330
7.00	357

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
8.00	385
9.00	414
10.00	445

На основі отриманого значення за довідником визначають укрупнену норму часу, скориговану коефіцієнтом $K_k = 0,7-0,8$ для умов розробки на комп'ютері:

$$T_{ap} = 306 \times 0,8 = 244,8 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість визначається для кожного етапу окремо, з урахуванням складності, новизни та використання стандартних модулів, за відповідними формулами:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.2)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (2.3)$$

$$T_{PP} = T^a p \times L_3 \times K_H \times K_T \quad (2.4)$$

Для розрахунку використовуються такі коефіцієнти:

- L_i – частка i -го етапу (табл. 2.3);
- K_H – коефіцієнт новизни (табл. 2.4);
- K_T – коефіцієнт використання типових програм (табл. 2.5).

Наш варіант виділено сірим.

Таблиця 2.3. Питомі коефіцієнти трудомісткості стадії у загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Таблиця 2.4. Значення коефіцієнта новизни

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово новий ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8

Код ступеня новизни	Ступінь новизни	Значення K_n
В	ПП, що має аналог	0,7

Таблиця 2.5. Значення коефіцієнта використання типових програм

Ступінь охоплення реалізованих функцій розробленого ПП типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Тепер розраховуємо трудомісткість для всіх етапів і зводимо у табл. 2.6:

Трудомісткість технічного завдання:

$$T_{mз} = T_a * L_1 * K_n = 244,8 * 0,12 * 0,8 = 23,50 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки технічного проєкту:

$$T_{mн} = T_a * L_2 * K_n = 244,8 * 0,15 * 0,8 = 29,37 \text{ (люд/годин)} \quad (2.3)$$

Трудомісткість розробки робочого проєкту:

$$T_{pн} = T_a * L_3 * K_n * K_m = 244,8 * 0,58 * 0,8 * 0,7 = 79,51 \text{ (люд/годин)} \quad (2.4)$$

Для розрахунків визначили обсяг документації по етапах:

- технічне завдання $N_{тз}=2$ (стор);
- розробка ТП $N_{тп}=53$ (стор);
- розробка робочого проєкту $N_{рп}=9$ (стор);
- пояснювальна записка відповідно $N_{пз}=15$ (стор).

Таблиця 2.6. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин		
1.ТЗ	$T_{Pтз}=23,50$	$T_{кк}=0,7*N_{тз}=0,7*2=1,4$	$T_{нк}=0,15*N_{тз}=0,15*2=0,30$
2.Розробка ТП	$T_{Pтп}=29,37$	$T_{кк}=0,7*N_{тп}=0,7*53=37,1$	$T_{нк}=0,15*N_{тп}=0,15*53=7,95$
3.Розробка РП	$T_{Pрп}=79,51$	$T_{кк}=0,7*N_{рп}=0,7*9=6,3$	$T_{нк}=0,15*N_{рп}=0,15*9=1,35$
4.Розробка ПЗ	$T_{Pпз}=1,5*N_{пз}=1,5*15=22,5$	$T_{кк}=0,7*N_{пз}=0,7*15=10,5$	$T_{нк}=0,15*N_{пз}=0,15*15=2,25$

Усього, в т.ч.:	213,53		
- на розробку	Тр=154,88		
- контроль керівника		Ткк=48,3	
- нормоконтроль			Тнк=10,35

2.3 Розрахунок ціни програмного продукту

Розраховуємо основну зарплату виконавців, матеріальні та загальні витрати на розробку ПП. Зарплата наведена в табл. 2.7. З 1 квітня 2024 мінімальна місячна зарплата – 8000 грн, погодинна ставка – 46 грн (згідно зі ст. 8 Закону про Держбюджет України).

Таблиця 2.7. Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	154,88	55,35	8573,48
2.Контроль керівника	48,3	120,28	5810,00
3.Нормоконтроль	10,35	120,28	1245,00
Усього	-	-	Зо= 15628,48

Розраховуємо матеріальні витрати на розробку ПП та наведемо їх у табл. 2.8.

Таблиця 2.8. Розрахунок матеріальних витрат на розробку

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	70	4.0	280,0
Разом	-	-	-	$V_{mi} = 280,0$
Транспортно – заготівельні Витрати (10%)				$V_{tr_z} = 0.1 * V_{m1} = 0,1 * 280 = 28,00$
Усього				$V_m = V_{mi} + V_{tr_z} = 308,00$

За отриманими даними складена калькуляція планової собівартості ПП, наведена в табл. 2.9.

					КГ 08. 04 002. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	308,00	V_m (див. табл. 2.8)
2. Основна заробітна плата	15628,48	Z_o (див. табл. 2.7)
3. Додаткова заробітна плата	1562,84	$Z_d = 0.1 * Z_o = 15628,48 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	3864,37	$V_{e.c.v.} = 0.22 * (Z_o + Z_d) = 0,22 * (15628,48 + 1562,84)$
5. Накладні витрати	6251,39	$V_{нак.} = 0.4 * Z_o = 0.4 * 15628,48$
6. Повна собівартість	27681,08	$C_{пов} = V_m + Z_o + Z_d + V_{e.c.v.} + V_{нак.} = 308,00 + 15628,48 + 1562,84 + 3864,37 + 6251,39$

Розмір прибутку розраховується за формулою:

$$P = (C_n * P) / 100 = (27681,08 * 10) / 100 = 2768,11 \text{ грн.} \quad (2.5)$$

Де p – плановий рівень рентабельності (10-15%).

Оптова ціна розраховується за формулою:

$$C_o = C_n + P = 27681,08 + 2768,11 = 30449,19 \text{ грн.} \quad (2.6)$$

За отриманими даними, ціна реалізації ПП за формулою становить:

$$C_p = C_o + ПДВ = 30449,19 + 30449,19 * 0.2 = 36539,03 \text{ грн.} \quad (2.7)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Основні положення

Основним правовим актом України є Конституція, прийнята Верховною Радою 26 червня 1996 року, яка закріплює права та свободи громадян. У трудових відносинах ці права деталізуються в законах, нормативно-правових актах з охорони праці (НПАОП), державних стандартах та постановах Кабінету Міністрів щодо безпеки праці.

Відповідальність за створення безпечних і здорових умов праці лежить на керівництві підприємств і організацій, яке має впроваджувати сучасні технічні засоби безпеки для запобігання травматизму і забезпечувати санітарно-гігієнічні умови, що знижують ризик професійних захворювань.

У цьому розділі дипломного проекту розглядаються питання охорони праці безпосередньо на робочому місці веб-дизайнера.

3.2 Аналіз умов праці

У процесі виконання професійних обов'язків програмісти піддаються впливу низки фізичних небезпечних та шкідливих виробничих факторів. Серед них можна виділити підвищений рівень шуму, надмірну температуру навколишнього середовища, недостатнє або повністю відсутнє освітлення робочої зони, дію електричного струму, а також накопичення статичної електрики та інші подібні чинники.

3.3 Гігієнічні вимоги до виробничого середовища

Робоче місце веб-дизайнера має бути організоване таким чином, щоб забезпечити умови для безпечного виконання обов'язків і досягнення високої продуктивності праці.

3.2.1 Вимоги до приміщення

Розташовувати робочі місця з персональними комп'ютерами у підвальних або цокольних приміщеннях не дозволяється. Об'ємно-планувальні

					<i>КГ 08. 04 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

характеристики будівель і приміщень, призначених для роботи з відеодисплейними терміналами (ВДТ), повинні відповідати нормам, встановленим ДСанПіН 3.3.2.007-98. На кожне робоче місце має припадати площа не менше 6,0 м², а об'єм повітря — щонайменше 20,0 м³. У таких приміщеннях слід щодня проводити вологе прибирання, вони повинні бути забезпечені аптечками для надання першої медичної допомоги. Також передбачено облаштування побутових приміщень для відпочинку працівників.

3.2.2 Виробниче освітлення

Приміщення для роботи з відеодисплейними терміналами повинні мати природне та штучне освітлення, а вікна бажано орієнтувати на північ або північний захід для рівномірного світла. Для захисту від прямого сонячного випромінювання на вікнах слід встановлювати штори або жалюзі.

Кольорове оформлення робочих і допоміжних приміщень має враховувати орієнтацію вікон і поєднувати основні кольори низької насиченості для стін і робочих поверхонь, середню насиченість для рідко видимих зон і яскраві акценти для деталей. Стелі фарбують у білий колір. Щоб уникнути відблисків, обладнання має бути матовим, а стіни — пофарбовані в пастельні тони.

Для штучного освітлення застосовують люмінесцентні лампи типу ЛБ, що мають спектр, близький до природного світла, вищу світловіддачу (у 2–5 разів) і термін служби до 10 тис. годин. У локальних світильниках дозволяється використання ламп розжарювання.

3.2.3 Гігієнічні нормування параметрів повітря робочої зони

Згідно з вимогами ГОСТ 12.1.005-88 та СН 4088-86, у виробничих приміщеннях на робочих місцях повинні бути створені оптимальні умови мікроклімату, які включають температурний режим, відносну вологість повітря та швидкість його руху. Ці параметри мають відповідати встановленим нормативам для забезпечення комфортних та безпечних умов праці.

У таблиці 3.1 наведено значення основних мікрокліматичних показників.

					КГ 08. 04 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

Таблиця 3.1. Параметри мікроклімату

Параметри мікроклімату	Значення параметрі	значення параметрі
	Взимку	влітку
Температура, С ⁰	22-24	23-25
Відносна вологість, %	40-60	40-60
Швидкість руху повітря, м/с	0,1	0,1-0,2

Для забезпечення в приміщеннях повітряного середовища, що відповідає санітарно-гігієнічним нормам, а також для видалення з повітря шкідливих домішок, пилу та газів, застосовується вентиляція. У разі природної вентиляції повітрообмін здійснюється через відкриті вікна завдяки різниці температур і тиску між внутрішнім і зовнішнім середовищем. Механічна вентиляція реалізується за допомогою спеціального обладнання — вентиляторів і кондиціонерів, які примусово забезпечують надходження та видалення повітря.

3.2.4 Організація робочих місць

Робочі місця, оснащені персональними комп'ютерами, рекомендується розміщувати у внутрішній частині приміщення, подалі від вікон. Недопустимо розташовувати відеодисплейний термінал таким чином, щоб працівник сидів обличчям або спиною до вікна, незалежно від того, який тип загального освітлення використовується — пряме чи відбите світло. Робоче місце слід організувати так, щоб у полі зору працівника не було блискучих поверхонь, вікон або світильників. Освітлювальні прилади повинні розташовуватись під кутом 90–100 градусів відносно вікон, щоб світло потрапляло збоку.

Рекомендовані параметри робочого столу: висота — 725 мм, ширина — в межах 600–1400 мм, глибина — від 800 до 1000 мм. Робоче крісло має бути обладнане підйомно-поворотним механізмом, що дозволяє регулювати висоту сидіння, нахил та висоту спинки. Усі регулювання мають бути зручними, незалежними одне від одного та мати надійну фіксацію в обраному положенні. Висота робочого столу повинна мати діапазон регулювання від 680 до 800 мм, а ширина має забезпечувати виконання операцій у межах зони моторного поля досяжності.

					КГ 08. 04 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

Клавіатуру слід розташовувати на робочій поверхні столу на відстані 100–300 мм від краю, найближчого до працівника.

3.2.5 Електробезпека

Використання електроенергії на виробництві несе ризик ураження електричним струмом через контакт з оголеними проводами, незаземленими корпусами або відкритими рубильниками.

Для запобігання ураженню слід:

1. Чітко дотримуватися правил виконання робіт і технічної експлуатації;
2. Запобігати доступу до частин обладнання під небезпечною напругою, неізольованих елементів і тих, що не підключені до заземлення;
3. Використовувати ізоляційні матеріали для захисту від струму.

У приміщеннях з понад п'ятьма ЕОМ має бути аварійний вимикач у доступному місці для повного відключення електроживлення, крім освітлення.

Комп'ютери з відеодисплейними терміналами підключаються лише через справні заводські штепсельні роз'єми та розетки. Роз'єми для 12 В і 42 В мають відрізнятися конструктивно та кольорово від роз'ємів для 127 В і 220 В.

3.4 Пожежна безпека

Пожежі завдають значних збитків виробничим будівлям, знищують матеріали та готову продукцію, пошкоджують обладнання і на тривалий час зупиняють роботу цехів.

Пожежна профілактика – це сукупність заходів, спрямованих на запобігання виникненню пожежі та створення умов для її швидкого гасіння. Заходи пожежної безпеки поділяють на дві основні категорії: запобігання пожежам і ліквідація вже виниклих пожеж.

Протипожежний захист приміщень забезпечується шляхом застосування автоматичної пожежної сигналізації, наявністю засобів пожежогасіння, використанням будівельних конструкцій з регламентованими межами

					КГ 08. 04 003. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

вогнестійкості та організацією своєчасної евакуації людей. Для гасіння пожеж у початковій стадії використовують первинні засоби пожежогасіння, які повинні бути присутніми у всіх виробничих приміщеннях та цехах.

Оснащення об'єктів первинними засобами пожежогасіння здійснюється відповідно до Правил пожежної безпеки в Україні, що введені в дію наказом МВС України від 22.06.1995 №400.

До первинних засобів належать: вогнегасники, пожежний інвентар (покривала з негорючих теплоізоляційних матеріалів, грубововняної тканини або повсті, ящики з піском, бочки з водою, пожежні відра, совкові лопати) та пожежний інструмент (гаки, ломи, сокири тощо). Бочки для води, що призначені для пожежогасіння, повинні мати об'єм не менше 0,2 м³ і бути укомплектовані пожежним відром місткістю не менше 0,008 м³ згідно з ГОСТ 12.4.009-83.

Пожежні щити (стенди) встановлюються на території об'єкта з розрахунку один щит на кожні 5000 м². Ящики для піску повинні мати об'єм 0,5; 1,0 або 3,0 м³ та комплектуватися совковою лопатою. Вмістилище для піску, що є частиною конструкції пожежного стенду, має місткість не менше 0,1 м³. Конструкція ящика повинна забезпечувати зручний доступ до піску та захищати його від попадання опадів.

3.5 Підведення підсумків

Отже, дотримання законодавчих норм і впровадження сучасних засобів безпеки є ключовими для створення безпечних і здорових умов праці. Відповідальність за це покладається на керівництво підприємств, яке має забезпечувати належний рівень охорони праці. Особлива увага у цьому проєкті приділена забезпеченню безпеки безпосередньо на робочому місці веб-дизайнера.

					<i>КГ 08. 04 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

ВИСНОВКИ

У процесі виконання дипломної роботи було досліджено потребу у створенні спеціалізованого інструменту для підтримки діяльності колористів у сфері перукарських послуг.

У рамках проєкту було спроектовано та реалізовано веб-застосунок Hair Color Expert, який дозволяє користувачам:

- вводити цифрові коди відтінків у форматі професійної колористики;
- здійснювати обчислення компліментарних пар;
- моделювати результати змішування з урахуванням співвідношень;
- користуватись інтерактивною палітрою для вибору та навігації по кольорах;
- отримувати практичні поради з фарбування та підбору формул.

Система побудована з використанням сучасного інструментарію фронтенд-розробки: React, TypeScript, SCSS, з використанням HSL-моделі для візуалізації кольору. Було розроблено набір допоміжних алгоритмів для парсингу коду відтінку, обчислення середнього кольору, пошуку компліментарного тону та виводу результатів у форматі, близькому до професійного салонного середовища. Дизайн інтерфейсу адаптований до потреб реального користувача: проста структура, миттєвий зворотний зв'язок, мінімум кліків.

Під час роботи також було проаналізовано особливості колористичної теорії коло Іттена, глибина тону, відображення нюансів.

Поставлені цілі були досягнуті: розроблений застосунок демонструє стабільну роботу, інтуїтивно зрозумілий інтерфейс і надає функціональність, якої бракує більшості доступних на ринку рішень. Особливою перевагою є орієнтація на локальні потреби, простота взаємодії та поєднання професійного досвіду з цифровими технологіями.

					<i>КГ 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Врочинська А. Українські народні жіночі прикраси XIX – початку XX ст. – Львів: Інститут народознавства НАН України, 2020. – 180 с.
Видання присвячене вивченню традиційних жіночих прикрас в українській культурі XIX – початку XX століття, зокрема їх форм, матеріалів і символіки.
2. Нурі Сан Фелікс М. Розробка веб-додатків з Quarkus та React. – Київ: Ліра-К, 2023. – 294 с.
У книзі розглядаються основи розробки сучасних веб-застосунків з використанням фреймворку Quarkus та бібліотеки React, з прикладами і практичними поясненнями.
3. Norman, D. A. The Design of Everyday Things. – Cambridge, MA: MIT Press, 2013. – 384 p.
Праця аналізує взаємодію людей з об'єктами щоденного вжитку та принципи створення зручного, зрозумілого дизайну.
4. Rutter, R. Web Typography. – New York: O'Reilly Media, 2017. – 224 p.
5. Дизайн-сервіс GRAFX. Психологія кольору у графічному дизайні: як обирати палітру [Електронний ресурс]. – Режим доступу: <https://grafx.com.ua/psykholohiya-koloru-u-dyzaini/> – Назва з екрана. – Дата звернення: 15.05.2025.
6. Академія кольору ESTEL Україна. Основи сучасної колористики: теорія та практика [Електронний ресурс]. – Режим доступу: <https://estel.pro.ua/navchannya/osnovy-kolorystyky/> – Назва з екрана. – Дата звернення: 17.05.2025.
7. Interaction Design Foundation. Color in UI Design: A Practical Framework [Електронний ресурс]. – Режим доступу: <https://www.interaction-design.org/literature/article/color-in-ui-design-a-practical-framework> – Назва з екрана. – Дата звернення: 10.06.2025.

					<i>КГ 08. 04 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

ДОДАТОК А. Фрагмент програмного коду компоненту обчислення кольорів

```
import { shades } from "../data/shades";

export type ShadeCode = `${1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10}.${number}`;

export const shadeRegex = /^(10|[1-9])\.\d+$/;

export function tryParseShade(code: string): HSL | null {
  if (!shadeRegex.test(code)) return null;
  return parseShade(code as ShadeCode);
}

export const typingRegex = /^(\d{0,2})(\.\d*)?$/;

export interface HSL {
  h: number; // 0-360
  s: number; // 0-100
  l: number; // 0-100
}

const toneHueMap: Record<number, number> = {
  0: 35, // natural / beige
  1: 220, // ash
  2: 265, // pearl/violet
  3: 55, // gold
  4: 30, // copper
  5: 300, // mahogany / purple-red ← було 345
  6: 0, // red
  7: 135, // matte / green
  8: 25, // mocha
  9: 200 // cendré
};

export const toneToHue = (tone: number): number =>
  toneHueMap[tone] ?? 0;

const levelToLightness = (level: number) => 100 - level * 8;

export function parseShade(code: ShadeCode | string): HSL {
  const [levelStr, toneStr] = code.split(".");

  const level = Number(levelStr);
  const l = levelToLightness(isNaN(level) ? 6 : level);

  const tone = toneStr ? Number(toneStr.charAt(0)) : 0;

  return { h: toneToHue(tone), s: 70, l };
}

export const hsl = (v: HSL) => v;

export const complementary = ({ h, s, l }: HSL): HSL => ({
  h: (h + 180) % 360,
  s,
  l
});

export function mix(a: ShadeCode, b: ShadeCode, ratio = 0.5): HSL {
  const c1 = hsl(parseShade(a));
  const c2 = hsl(parseShade(b));
```

```

    return {
      h: c1.h + (c2.h - c1.h) * ratio,
      s: c1.s + (c2.s - c1.s) * ratio,
      l: c1.l + (c2.l - c1.l) * ratio
    };
  }
}

export function hslToHex({ h, s, l }: HSL): string {
  const _s = s / 100;
  const _l = l / 100;
  const k = (n: number) => (n + h / 30) % 12;
  const a = _s * Math.min(_l, 1 - _l);
  const f = (n: number) =>
    _l - a * Math.max(-1, Math.min(k(n) - 3, Math.min(9 - k(n), 1)));
  const rgb = [0, 8, 4].map(n => Math.round(255 * f(n)));
  return (
    "#" +
    rgb
      .map(x => x.toString(16).padStart(2, "0"))
      .join("")
      .toUpperCase()
  );
}

export function hueToNearestTone(h: number): number {
  const entries = Object.entries(toneHueMap).map(([t, hue]) => ({
    tone: Number(t),
    diff: Math.abs(((h - hue + 540) % 360) - 180) // «круговая» разница
  }));
  return entries.sort((a, b) => a.diff - b.diff)[0].tone;
}

export function toShadeCode(level: number, h: number): ShadeCode {
  const tone = hueToNearestTone(h);
  return `${level}.${tone}` as ShadeCode;
}

export function randomShade(): ShadeCode {
  const level = Math.floor(Math.random() * 10) + 1;
  const tone = Math.floor(Math.random() * 10);
  return `${level}.${tone}` as ShadeCode;
}

function hexToHsl(hex: string): HSL {
  const r = parseInt(hex.slice(1, 3), 16) / 255;
  const g = parseInt(hex.slice(3, 5), 16) / 255;
  const b = parseInt(hex.slice(5, 7), 16) / 255;

  const max = Math.max(r, g, b);
  const min = Math.min(r, g, b);
  const l = (max + min) / 2;

  if (max === min) return { h: 0, s: 0, l: l * 100 };

  const d = max - min;
  const s = l > 0.5 ? d / (2 - max - min) : d / (max + min);

  let h = 0;
  switch (max) {
    case r: h = (g - b) / d + (g < b ? 6 : 0); break;
    case g: h = (b - r) / d + 2; break;
    case b: h = (r - g) / d + 4; break;
  }
}

```

```

    return { h: h * 60, s: s * 100, l: l * 100 };
  }

export function hexToNearestCode(hex: string): ShadeCode {
  const targetHue = hexToHsl(hex).h;

  let bestCode: ShadeCode = "7.0";
  let bestDiff = Infinity;

  for (const shade of shades) {
    const shadeHue = hexToHsl(shade.hex).h;
    const diff = Math.abs(((targetHue - shadeHue + 540) % 360) - 180);

    if (diff < bestDiff) {
      bestDiff = diff;
      bestCode = shade.code as ShadeCode;
    }
  }
  return bestCode;
}

import { ShadeCode } from "../utils/colorUtils";

export interface ShadeEntry {
  code: ShadeCode;
  hex: string;
  h: number;
}

export const shades: ShadeEntry[] = [
  { code: "1.2", hex: "#181B26", h: 226 },

  { code: "3.0", hex: "#362E29", h: 35 },
  { code: "3.1", hex: "#36404E", h: 220 },
  { code: "3.3", hex: "#5E4B28", h: 55 },

  { code: "4.56", hex: "#5E2E2F", h: 323 },
  { code: "4.65", hex: "#70373C", h: 345 },

  { code: "5.1", hex: "#4F5260", h: 220 },
  { code: "5.5", hex: "#4E374D", h: 300 },

  { code: "6.66", hex: "#7A1B1E", h: 0 },
  { code: "6.46", hex: "#A7442C", h: 12 },
  { code: "6.45", hex: "#835039", h: 30 },
  { code: "6.35", hex: "#8F6A44", h: 38 },
  { code: "6.23", hex: "#806042", h: 35 },

  { code: "7.4", hex: "#C57132", h: 30 },
  { code: "7.43", hex: "#BC4F24", h: 25 },
  { code: "7.12", hex: "#6A6E8A", h: 230 },
  { code: "7.23", hex: "#D7B26C", h: 45 },

  { code: "8.13", hex: "#B1B6D3", h: 230 },
  { code: "8.45", hex: "#C26A35", h: 28 },

  { code: "10.0", hex: "#F2E1B5", h: 35 }
];

import { useState, useRef } from "react";
import {
  ShadeCode,
  parseShade,
  complementary,

```

```

    hslToHex,
    hexToNearestCode,
    toShadeCode,
    tryParseShade,
    randomShade
} from "../utils/colorUtils";
import { ShadeInput } from "../components/ShadeInput";
import { ResultCard } from "../components/ResultCard";
import { ColorWheel } from "../components/ColorWheel";

export default function ComplementaryPage() {
    const [code, setCode] = useState<string>("6.66");

    const lastValid = useRef<ShadeCode>("6.66");

    const parsed = tryParseShade(code);
    if (parsed) lastValid.current = code as ShadeCode;

    const base = parsed ?? parseShade(lastValid.current);
    const comp = complementary(base);

    const handleWheel = (hue: number) => {
        const levelStr = (parsed ? code : lastValid.current).split(".")[0];
        const newCode = toShadeCode(Number(levelStr), hue);
        setCode(newCode);
    };

    const handleLucky = () => {
        setCode(randomShade());
    };

    return (
        <div className="flex flex-col items-center gap-8 p-8">
            <h1 className="text-4xl font-bold bg-gradient-to-r from-pink-400 to-fuchsia-600 bg-clip-text text-transparent">
                Компліментарність кольорів
            </h1>

            <div className="flex flex-col sm:flex-row gap-6 items-center">
                <ColorWheel angle={base.h} onChange={handleWheel} />

                <div className="flex flex-col items-center gap-3">
                    <ShadeInput value={code} onChange={setCode} />
                    <button
                        onClick={handleLucky}
                        className="px-3 py-1.5 rounded-md bg-pink-500 text-white
                        hover:bg-pink-600 active:scale-95 transition"
                    >
                        🍀 Мені пощастить
                    </button>
                </div>
            </div>

            <div className="flex gap-6">
                <ResultCard title={parsed ? code : lastValid.current} color={base} />
                <ResultCard
                    title={hexToNearestCode(hslToHex(comp))}
                    color={comp}
                />
            </div>
        </div>
    );
}

```

```

export default function App() {
  return (
    <BrowserRouter>
      <HomeLink />
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/complementary" element={<ComplementaryPage />} />
        <Route path="/mix" element={<MixPage />} />
        <Route path="/about" element={<AboutApp />} />
        <Route path="/tips" element={<TipsPage />} />
        <Route path="/author" element={<AboutAuthor />} />
      </Routes>
    </BrowserRouter>
  );
}

export default function MixPage() {
  const [codeA, setCodeA] = useState<string>("6.66");
  const [codeB, setCodeB] = useState<string>("7.43");
  const [ratio, setRatio] = useState(50);

  const shadeA = tryParseShade(codeA);
  const shadeB = tryParseShade(codeB);

  const mixed =
    shadeA && shadeB
      ? mix(codeA as ShadeCode, codeB as ShadeCode, ratio / 100)
      : null;

  return (
    <div className="flex flex-col items-center gap-8 p-8">
      <h1 className="text-4xl font-bold bg-gradient-to-r from-pink-400 to-fuchsia-600 bg-clip-text text-transparent">
        Змішування відтінків
      </h1>

      <div className="flex gap-8">
        <ShadeInput value={codeA} onChange={setCodeA} />
        <ShadeInput value={codeB} onChange={setCodeB} />
      </div>

      <MixSlider label="Співвідношення" value={ratio} onChange={setRatio} />

      {mixed && (
        <ResultCard
          title={hexToNearestCode(hslToHex(mixed))}
          color={mixed}
        />
      )}
    </div>
  );
}

```

ДОДАТОК Б. Слайди мультимедійної презентації



РОЗРОБКА ЕКСПЕРТНОЇ ВЕБ-СИСТЕМИ ПЕРУКАРНІ ДЛЯ ОПТИМІЗАЦІЇ РОБОТИ КОЛОРИСТІВ

ВИКОНАЛА ЗДОБУВАЧКА ОСВІТИ: ГАНЕНКО О.І.

КЕРІВНИК: СКОРНЯКОВ В.С.

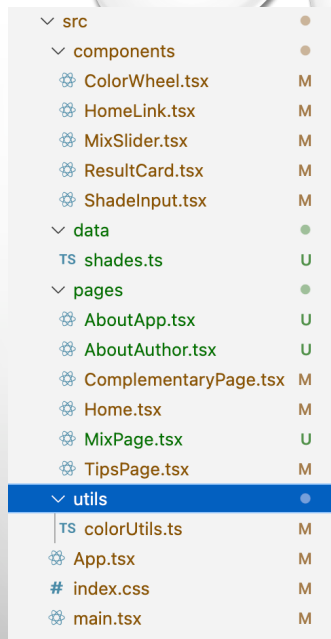
ПРЕДСТАВЛЕННЯ КОЛЬОРІВ У ЦИФРОВОМУ ВИГЛЯДІ (HSL) ТА У ЖИТТІ (ВІДТІНКИ)



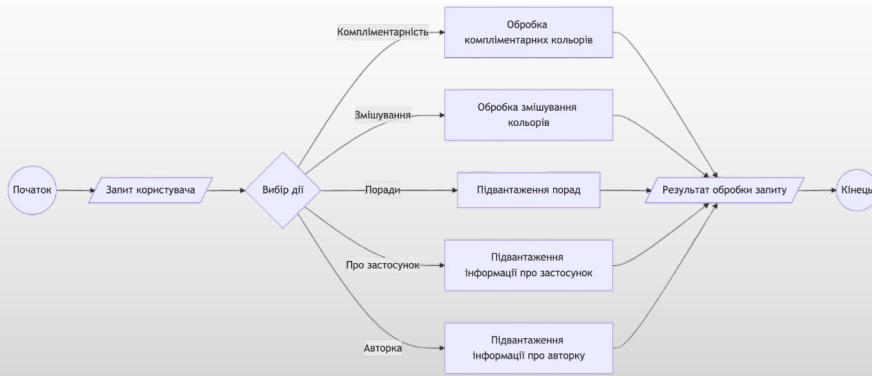
РЕПРЕЗЕНТАЦІЯ КОЛЬОРІВ У ЗАСТОСУНКУ



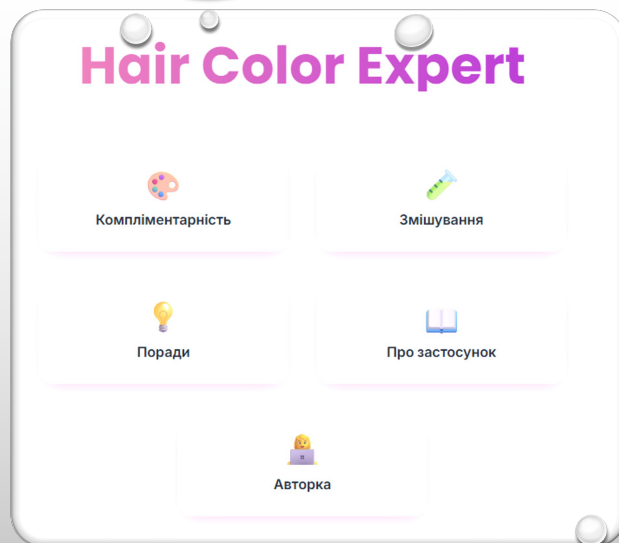
Код відтинку	HSL-представлення	Візуальний колір (опис)
6.34	hsl(200, 74%, 58%)	Насичений попелясто-мідний
9.1	hsl(60, 70%, 37%)	Світлий попелясто-бежевий
7.0	hsl(180, 60%, 51%)	Природний середній коричневий
4.65	hsl(270, 85%, 35%)	Насичений фіолетовий відтінок
10.3	hsl(30, 63%, 30%)	Темний золотисто-русявий



АРХІТЕКТУРА РЕАКТ-ЗАСТОСУНКУ ТА СТРУКТУРА КОМПОНЕНТІВ

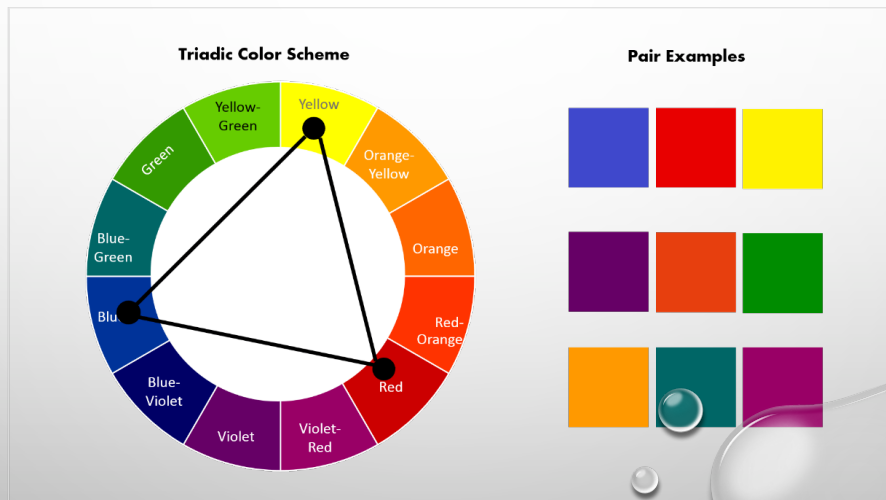


БЛОК СХЕМА ШЛЯХУ КОРИСТУВАЧА ПО СИСТЕМІ

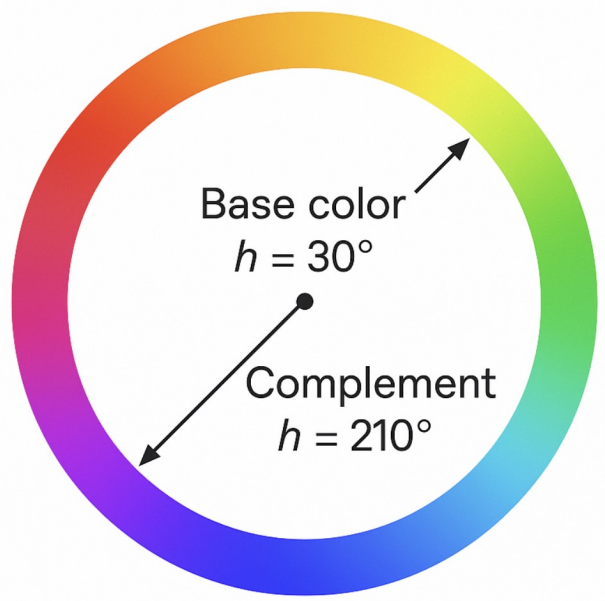


ІНТЕРФЕЙС КОРИСТУВАЧА: ГОЛОВНА СТОРІНКА

ОГЛЯД КОЛЬОРОВОЇ МОДЕЛІ ТА ПРИНЦИПІВ КОЛОРИСТИКИ



HSL



РЕАЛІЗАЦІЯ КОМПЛІМЕНТАРНОСТІ

```
export const complementary = ({ h, s, l }: HSL): HSL => ({  
  h: (h + 180) % 360,  
  s,  
  l  
});
```

Компліментарність кольорів



Код відтінку

Мені пощастить



6.66



3.1

Компліментарність кольорів



Код відтінку

Мені пощастить



1.9

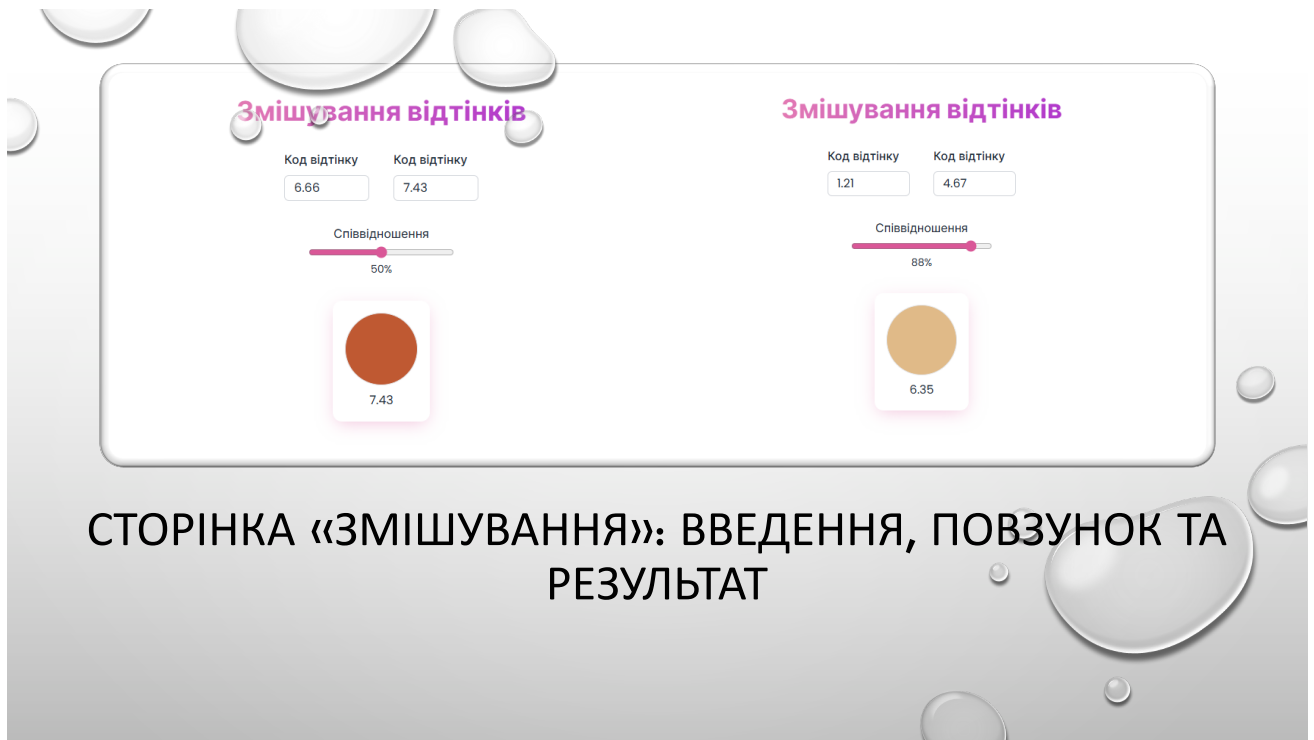


8.45

СТОРІНКА «КОМПЛІМЕНТАРНІСТЬ»: ВИБІР ТА АНАЛІЗ ВІДТІНКІВ

РЕАЛІЗАЦІЯ ЗМІШУВАННЯ

```
6
7 export function mix(a: ShadeCode, b: ShadeCode, ratio = 0.5): HSL {
8   const c1 = hsl(parseShade(a));
9   const c2 = hsl(parseShade(b));
0   return {
1     h: c1.h + (c2.h - c1.h) * ratio,
2     s: c1.s + (c2.s - c1.s) * ratio,
3     l: c1.l + (c2.l - c1.l) * ratio
4   };
5 }
```



СТОРІНКА «ЗМІШУВАННЯ»: ВВЕДЕННЯ, ПОВЗУНОК ТА РЕЗУЛЬТАТ



БЛОК «ПОРАДИ КОЛОРИСТА»: ДОВІДКОВА ДОПОМОГА ДЛЯ МАЙСТРІВ



ВІЗУАЛЬНІ ПРИКЛАДИ РОБОТИ: СИМУЛЯЦІЯ, ОБРОБКА ТА РЕЗУЛЬТАТ

ВИВАНТАЖЕННЯ НА СЕРВЕР

ГОТОВИЙ ВЕБ-ЗАСТОСУНОК БУЛО РОЗГОРНУТО НА ПЛАТФОРМІ FIREBASE HOSTING.

РОЗГОРТАННЯ ВИКОНАНО ЧЕРЕЗ ІНСТРУМЕНТ КОМАНДНОГО РЯДКА FIREBASE CLI, ПІСЛЯ ПОПЕРЕДНЬОГО ЗБИРАННЯ ПРОЄКТУ З ВИКОРИСТАННЯМ NPM RUN BUILD.

САЙТ СТАВ ДОСТУПНИМ ДЛЯ КОРИСТУВАЧІВ ЗА АДРЕСОЮ [HTTPS://HAIRCUTCOLOR-8A505.WEB.APP](https://haircutcolor-8a505.web.app), БЕЗ НЕОБХІДНОСТІ КУПІВЛІ ОКРЕМОГО ДОМЕНУ.



РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Ганенко Олени Іванівни

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і Web-дизайн»

Керівник дипломного проекту (роботи) Скорняков В'ячеслав Сергійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка експертної веб-системи перукарні для оптимізації роботи колористів

Обсяг розрахунково-пояснювальної записки 77 сторінок

Обсяг графічної (презентаційної) частини 14 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений темі розробки експертних систем та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування застосунку, реалізації застосунку, тестування застосунку), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 14 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту Реалізовано математичне моделювання взаємодії кольорів на основі HSL-простору з формулами для змішування, комплементарності, побудови градієнтів. Продуманий UI/UX з адаптивністю під мобільні пристрої, орієнтацією на реальні сценарії використання в салонах.

д) основні недоліки дипломного проекту У моделі змішування кольорів використовуються просто середні значення без врахування складних нелінійностей, що могли б точніше імітувати хімічну реакцію. У роботі майже не згадується про зберігання історії відтінків, роботи з базою даних або інтеграцію з backend-частиною, що є важливою для практичної системи. Деякі недоліки оформлення пояснювальної записки.

Оцінка розрахункової частини Добре
Оцінка графічної частини Відмінно
Загальна оцінка Відмінно

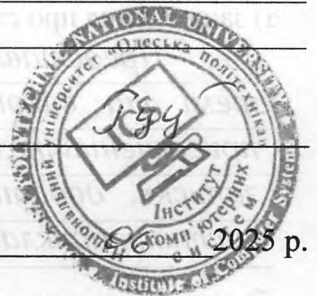
Прізвище, ім'я, по батькові рецензента к.т.н. Рудніченко Микола Дмитрович

Місце роботи і посада рецензента Національний університет «Одеська політехніка»,
доцент кафедри інформаційних технологій

Підпис:

« 23 »

2025 р.



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Ганенко Олени Іванівни

(прізвище, ім'я та по батькові)

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Комп'ютерна графіка і Web-дизайн»

Тема дипломного проекту: Розробка експертної веб-системи перукарні
для оптимізації роботи колористів.

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 77 сторінки. У пояснювальній записці описано етапи розробки експертної веб-системи перукарні для оптимізації роботи колористів засобами HTML, CSS, JavaScript та React. Графічна частина складається з окремих 14 слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної записки та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувачка освіти поступово та послідовно виконувала всі етапи, проявляла ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувачка освіти виконувала самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувачка освіти під час роботи над дипломним проектом вивчила достатньо багато літературних та інтернет-джерел за даною тематикою. Вважаю, що теоретична підготовка дипломника достатня і вона готова до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувачка освіти показала вміння організовано працювати над поставленим завданням, застосовувати знання у галузі програмування та математики, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як MS VS Code, HTML, CSS, JavaScript та Vue.js.

Оцінка розрахункової частини Добре
Оцінка графічної частини Відмінно
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Скорняков Вячеслав Сергійович

Місце роботи і посада керівника дипломного проекту ВСП «Одеський технічний фаховий коледж ОНТУ», викладач спецдисциплін циклової комісії комп'ютерної техніки та програмної інженерії

Підпис Скорняков

«20» 06 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Ганенко Олена Іванівна,
здобувачка освіти гр. 4КГ-08, та


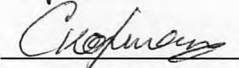
Скорняков Вячеслав Сергійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка експертної веб-системи перукарні для оптимізації роботи колористів» (авторка роботи – Ганенко О.І., керівник роботи – Скорняков В.С.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець		/ Ганенко О.І. /
Керівник		/ Скорняков В.С. /

«16» червня 2025 р.

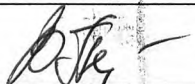
Д О В І Д К А

циклової комісії КТ та ПІ
про допуск до захисту дипломного проекту
здобувача (здобувачки) освіти ІV курсу
відділення комп'ютерних систем групи 4КГ-08

Ганенко Олени Іванівни

на тему *Розробка експертної веб-системи перукарні*
для оптимізації роботи колористів

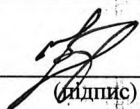
Висновок відповідальної особи за проведення нормоконтролю:
пояснювальна записка до дипломного проекту виконана з несуттєвими
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проектування


(підпис)

20.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагіату *згідно звіту про перевірку від 19.06.2025 р. значення коефіцієнту*
подібності в роботі становить 13,94%, коефіцієнт цитування – 2,72%.


(підпис)

20.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проекту

здобувача (здобувачки) освіти

Ганенко О.І.
(П.І.Б.)

проведена « 20 » червня 2025 р.

Висновки *Пояснювальна записка до дипломного проекту виконана у повному*
обсязі. Випускна кваліфікаційна робота (дипломний проект) відповідає
вимогам Положення про дипломне проектування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка експертної веб-системи перукарні для оптимізації роботи колористів

Автор

Науковий керівник / Експерт

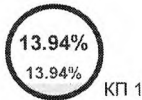
Ганенко Олена ІванівнаСкорняков В'ячеслав Сергійович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

15046

Кількість слів

120320

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв	Ⓛ	16
Інтервали	A→	0
Мікропробіли		1
Білі знаки	Ⓡ	67
Парафрази (SmartMarks)	a	94

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

порядковий номер	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Колір тексту
порядковий номер	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	кількість ідентичних слів (фрагментів)
1	Розробка мобільного застосунку-помічника майстра манікюру 6/18/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	77 0.51 %
2	https://card-file.ontu.edu.ua/bitstreams/9908b7a9-6b3e-46f5-a46e-84d83787cfd4/download	55 0.37 %

3	Розробка мобільного застосунку-помічника майстра манікюру 6/18/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	44 0.29 %
4	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	36 0.24 %
5	Розробка мобільного застосунку-помічника майстра манікюру 6/18/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	36 0.24 %
6	https://card-file.ontu.edu.ua/server/api/core/bitstreams/8da72e29-656f-4ee4-9b22-716dedf53ff5/content	34 0.23 %
7	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	33 0.22 %
8	https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download	31 0.21 %
9	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	30 0.20 %
10	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	30 0.20 %

з домашньої бази даних (1.62 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка мобільного застосунку-помічника майстра манікюру 6/18/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	243 (10) 1.62 %

з програми обміну базами даних (0.11 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Диплом - Дубас 12/9/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	17 (1) 0.11 %

з Інтернету (12.21 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/server/api/core/bitstreams/44c16132-5f53-48e2-b6c0-61e9a2f0fd75/content	549 (46) 3.65 %
2	https://card-file.ontu.edu.ua/bitstreams/1dff552d-7200-49b8-ae1d-ba76a1335685/download	308 (23) 2.05 %
3	https://card-file.ontu.edu.ua/bitstreams/9908b7a9-6b3e-46f5-a46e-84d83787cfd4/download	116 (4) 0.77 %
4	https://card-file.ontu.edu.ua/bitstreams/62baa43e-b968-4993-bb54-8cf8761a89b2/download	83 (7) 0.55 %
5	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	78 (4) 0.52 %
6	https://card-file.ontu.edu.ua/server/api/core/bitstreams/995bdcec-4e4d-4321-8070-4d6badcb8e49/content	75 (7) 0.50 %
7	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	70 (8) 0.47 %
8	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	66 (7) 0.44 %
9	https://card-file.ontu.edu.ua/bitstreams/53ed22ad-8700-4162-b97a-082a1ad472d6/download	54 (4) 0.36 %

10	https://card-file.ontu.edu.ua/bitstreams/bbed74c8-2ea7-44c5-8d00-0fe3fd9790ee/download	46 (4) 0.31 %
11	https://card-file.ontu.edu.ua/bitstreams/82a6d375-2b69-4233-b80f-fbfd149b7747/download	44 (2) 0.29 %
12	https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download	43 (4) 0.29 %
13	https://if.nmc.dsns.gov.ua/files/MetodRozrobky/8-%D0%9F%D0%B5%D1%80%D0%B2%D0%B8%D0%BD%D0%BD%D1%96-%D0%B7%D0%B0%D1%81%D0%BE%D0%B1%D0%B8-%D0%BF%D0%BE%D0%B6%D0%B5%D0%B6%D0%BE%D0%B3%D0%B0%D1%81%D1%96%D0%BD%D0%BD%D1%8F.pdf	41 (3) 0.27 %
14	https://card-file.ontu.edu.ua/bitstreams/11562741-24e6-4201-bc41-a00c8013fca1/download	41 (3) 0.27 %
15	https://mylektsii.su/1-47755.html	41 (3) 0.27 %
16	https://card-file.ontu.edu.ua/server/api/core/bitstreams/8da72e29-656f-4ee4-9b22-716dedf53ff5/content	34 (1) 0.23 %
17	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	30 (1) 0.20 %
18	https://card-file.ontu.edu.ua/bitstreams/9bff5906-114e-422a-9040-1f53839f4e40/download	24 (3) 0.16 %
19	https://www.ifnmu.edu.ua/images/zagalna_informacia/civilnii_zahist/metodichni_materiali/5.pdf	18 (1) 0.12 %
20	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	16 (1) 0.11 %
21	https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download	15 (2) 0.10 %
22	https://cyberleninka.ru/article/n/phenotypic-and-genotypic-characterization-of-lactococci-isolated-from-different-kinds-of-raw-milk-goat-cow-sheep-and-camel	14 (1) 0.09 %
23	http://stp.lingfil.uu.se/~santinim/ml/2014/JurafskyMartinSpeechAndLanguageProcessing2ed_draft%202007.pdf	10 (1) 0.07 %
24	https://card-file.ontu.edu.ua/server/api/core/bitstreams/ead3fa83-2e3d-4cd7-bfd-1d5ed04c1ce4/content	9 (1) 0.06 %
25	https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download	6 (1) 0.04 %
26	https://card-file.ontu.edu.ua/bitstreams/55e2b8f2-7d3c-4235-99fc-2be51199b96d/download	6 (1) 0.04 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ОДНАКОВИХ СЛІВ (ФРАГМЕНТІВ)
------------------	-------	---------------------------------------

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»
 Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн» Група: 4КГ-08

Дипломний проєкт здобувачки освіти денної форми навчання КГ. 08.04.000.ДП

ГАНЕНКО
 ОЛЕНИ ІВАНІВНИ

м. Одеса
 2025 р. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»
 Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн»
 Група: 4 КГ-08