Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (мігістерський)

Дослідження та використання технологій ШІ при розробці веб додатків інформаційних систем
(тема)

Виконав:
студент _2_ курсу, групи АПСм-22-1
Потапов М. І.
(прізвище, ініціали)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма
Архітектурне проектування інформаційних систем
(повна назва освітньої програми)

Керівник професор Цопа О.І.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав кафедри _____ Зарудний О. А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет_Інформаційних радіотехнологій і технічного захисту інформації_

Кафедра_____Радіотехнологій інформаційно-комунікаційних систем_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____126 Інформаційні системи та технології _____

Тип програми _____освітньо-професійна_____

Освітня програма Архітектурне проектування інформаційних систем_____

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____2023__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____Потапову Микиті Ігоровичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _Дослідження та використання технологій ШІ при розробці веб додатків інформаційних систем_____
_____

затверджена наказом університету від 3 жовтня 2023 р. № 1295Ст_

2. Термін подання студентом роботи до екзаменаційної комісії 24 січня 2024 р.

3. Вихідні дані до роботи _____
_Провести аналіз що таке штучний інтелект, його переваги та недоліки, провести аналіз використання штучного інтелекту в вебі, його переваги та недоліки в контексті вебу, визначити технології реалізації веб додатку с використанням ШІ та переваги цих технологій,

4. Перелік питань, що потрібно опрацювати в роботі Вступ_____
_1. Проблеми та переваги штучного інтелекту, історія його розвитку, алгоритм роботи_____ нейроної мережі_____
_2. Досліжєення використання ШІ у веб технологіях_____
_3. Функціонал веб додатку із штучним інтелектом, опис моделі ші та технології для його__ розробки_____
_4. Програмна реалізація функціоналу кастомізатора футболок з використанням ші_____
Висновки_____
_____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
____Слайди у форматі Power Point(назва роботи, вступ, необхідний функціонал, _проектування веб додатку, технології реалізації, реалізований функціонал, висновки)_____
_____
_____
_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	професор Цопа О.І.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів ро-	Примітка
1	Ознайомлення із завданням.Уточнення тз	07.10.2023	Виконано
2	Підбір літератури за темою роботи	12.10.2023 – 15.10.2023	Виконано
3	Проблеми та переваги ШІ історія його розвитку	16.10.2023 – 20.10.2023	Виконано
	алгоритм роботи нейроної мережі		
4	Дослідженя використання ШІ у веб технологіях	17.10.2023 – 30.10.2023	Виконано
5	Функціонал веб додатку із штучним інтелектом,	18.10.2023 – 06.11.2022	Виконано
	опис моделі ШІ та технології його розробки		
6	Програмна реалізація функціоналу кастомізатора	07.11.2023 – 23.12.2023	Виконано
	футболок з використанням ші		
7	Створення презентації, підготовка до захисту	23.12.2023 – 24.01.2024	Виконано

Дата видачі завдання 6 жовтня 2023р.

Студент _____
(підпис)

Керівник роботи _____ _____професор Цопа О.І.___
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 73 с., 24 рис., 15 джерел, 2 додатка.

ЗГЕНЕРУВАТИ ЗОБРАЖЕННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, ЛОГОТИП, ФУТБОЛКА, JAVASCRIPT, REACT, JSX, TYPESCRIPT, NODE.JS

Актуальність теми: багато людей зараз почали активно використовувати штучний інтелект для навчання, роботи чи бізнессу. Різноманітність моделей ШІ дозволяє вирішувати велику кількість питань та проблем, люди використовують ШІ щоб пришвидшити чи навість автоматизувати процессі для різних сфер.

Мета та завдання дипломної роботи: дослідити вплив штучного інтелекту на сучасний веб, проаналізувати розвиток штучного інтелекту в контексі веб розробки, розробити веб додаток кастомізатор футболок із ШІ.

THE ABSTRACT

Explanatory note: 73 p., 24 fig, 15 sources, 2 app.

GENERATE IMAGE, ARTIFICIAL INTELLIGENCE, LOGO, T-SHIRT, JAVASCRIPT, REACT, JSX, TYPESCRIPT, NODE.JS.

Relevance of the topic: many people have now started to actively use artificial intelligence for education, work or business. The variety of AI models allows solving a large number of questions and problems, people use AI to speed up or even automate processes for various fields.

The purpose and objectives of the thesis: to investigate the impact of artificial intelligence on the modern web, to analyze the development of artificial intelligence in the context of web development, to develop a web application for customizing t-shirts with AI.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАК ТА СКОРОЧЕНЬ

API – Application Programming Interface – інтерфейс прикладного програмування;

BE – Back End;

FE – Front End;

JS– Javascript;

SD – Stable Diffusion;

TS – Typescript;

ІС – Інформаційна система;

ШІ – Штучний інтелект.

ВСТУП

У наш час штучний інтелект став невід’ємною частиною нашого повсякденного життя, проникаючи в різні сфери суспільства, починаючи з автоматизації виробництва та медичних діагнозів і закінчуючи створенням персональних віртуальних асистентів. Використання штучного інтелекту стає все більш актуальною та обговорюваною темою, як у світі науки та технологій, так і в громадському дискурсі.

Штучний інтелект представляє собою галузь інформатики, яка прагне створити програми та системи, здатні аналізувати дані, приймати рішення та навіть "вчитися" на основі досвіду. Цей технологічний прорив відкриває перед нами безмежні перспективи і викликає безліч питань щодо його впливу на суспільство, економіку, етику та навіть майбутнє людства.

Штучний інтелект відіграє ключову роль у розвитку та вдосконаленні веб-технологій, що призводить до значних змін в Інтернеті та способів взаємодії користувачів з онлайн-сервісами. Ось кілька способів використання штучного інтелекту в веб-технологіях:

Персоналізація контенту: Штучний інтелект аналізує дані про поведінку користувачів на сайті та, на основі цієї інформації, створює персоналізований контент. Це може включати персональні рекомендації, які підібрані на основі інтересів і попередньої взаємодії користувача з сайтом.

Чат-боти: Інтелектуальні чат-боти, які працюють на основі машинного навчання, допомагають вирішувати запити користувачів, надаючи швидку та ефективну підтримку. Вони можуть відповідати на запитання, вирішувати проблеми та взаємодіяти з користувачами на веб-сайтах.

Аналіз великих даних: Інтелектуальні алгоритми дозволяють веб-сайтам аналізувати великі обсяги даних для виявлення патернів, трендів та прогнозування користувацьких потреб. Це може допомогти покращити якість послуг та

оптимізувати веб-сайти.

Розпізнавання образів і голосу: Інтелектуальні системи здатні до розпізнавання образів та голосових команд користувачів. Це використовується, наприклад, в системах розпізнавання тексту на зображеннях та в голосових пошуках.

Безпека: Існує велика кількість інтелектуальних рішень для виявлення та запобігання кіберзлочинності, включаючи веб-захист та ідентифікацію користувачів на основі біометричних даних.

Використання штучного інтелекту в веб-технологіях дозволяє створювати більш інтелектуальні та відповідальні веб-сервіси, покращуючи користувацький досвід і забезпечуючи більшу ефективність у взаємодії з онлайн-ресурсами.

Генерація контенту: Штучний інтелект може бути використаний для створення тексту, зображень, відео та іншого контенту для веб-сайтів. Наприклад, системи генерації тексту на базі штучного інтелекту можуть автоматично створювати статті, огляди товарів або навіть літературні твори. Системи обробки природної мови дозволяють аналізувати текст та структуру контенту для покращення якості та релевантності інформації.

1. ПРОБЛЕМИ ТА ПЕРЕВАГИ ШТУЧНОГО ІНТЕЛЕКТУ, ІСТОРІЯ ЙОГО РОЗВИТКУ, АЛГОРИТМ РОБОТИ НЕЙРОНОЇ МЕРЕЖІ

1.1 Проблеми та переваги штучного інтелекту

Штучний інтелект (ШІ) - це галузь комп'ютерних наук та інженерії, яка зосереджена на створенні систем та програм, які здатні виконувати завдання, що зазвичай потребували би інтелекту людини. ШІ спроектований для імітації когнітивних функцій, таких як розум, сприйняття, розуміння, вирішення проблем, навчання та прийняття рішень. Ця технологія включає в себе використання алгоритмів, нейронних мереж, статистичних моделей та інших методів обробки інформації для аналізу даних і прийняття відповідних дій. ШІ використовується в різних сферах, включаючи медицину, автомобільну промисловість, фінанси, маркетинг, веб-технології та багато інших [1].

Основною метою розробки штучного інтелекту є створення систем, які можуть розв'язувати складні завдання та адаптуватися до змінних умов, навчаючись на основі досвіду та даних. Це може включати в себе розпізнавання образів, обробку природної мови, прийняття рішень на основі даних, аналіз великих обсягів інформації та багато інших завдань, які вимагають інтелектуальних здібностей.

Переваги штучного інтелекту:

- автоматизація та продуктивність: Штучний інтелект дозволяє автоматизувати багато рутинних завдань, що підвищує продуктивність та звільняє людей для більш складних та творчих завдань;
- точність та швидкість: ІС в змозі аналізувати великі обсяги даних швидко та без помилок, що особливо корисно в областях, де потрібна висока точність, наприклад, в медицині чи фінансах;
- здатність до навчання: Системи штучного інтелекту можуть "вчитися" на

основі даних і вдосконалювати свої навички з часом, що робить їх дедалі ефективнішими;

– постійна доступність: ІС можуть працювати неколижну, не вимагаючи відпочинку, що робить їх ідеальними для надзвичайно обчислювальних завдань.

Проблеми штучного інтелекту:

– етичні питання: Використання ІС може породжувати питання етики, зокрема в областях, де вони приймають рішення, які впливають на життя та свободи людей;

– безпека даних: Збільшена залежність від ІС може створювати загрози для безпеки даних, зокрема у вигляді злому або витоку конфіденційної інформації.

– втрата робочих місць: Автоматизація завдань, що раніше виконували люди, може призвести до зменшення потреби в ручній праці та, як наслідок, до втрати робочих місць;

– питання прозорості: Деякі моделі машинного навчання можуть бути недостатньо зрозумілими для пояснення своїх рішень, що ускладнює їх прийняття та контроль.

1.2 Проблеми та переваги використання штучного інтелекту в веб-технологіях

Переваги використання штучного інтелекту в веб-технологіях:

– персоналізація та покращений користувацький досвід: Штучний інтелект дозволяє аналізувати дані користувачів і надавати персоналізований контент та рекомендації, що значно підвищує рівень задоволення користувачів та їх залученість;

– автоматизація та оптимізація: ШІ дозволяє автоматизувати процеси, такі як моніторинг і аналіз веб-трафіку, управління рекламою та обслуговування клієнтів. Це зменшує людський труд і підвищує ефективність веб-сервісів;

– аналіз великих даних: Веб-технології генерують великі обсяги даних, які можуть бути проаналізовані з використанням ШІ для виявлення патернів, трендів та розуміння потреб користувачів;

– генерація контенту: Штучний інтелект здатний створювати текст, зображення та навіть відео для веб-сайтів, що дозволяє власникам ресурсів підтримувати актуальний та цікавий контент.

Проблеми використання штучного інтелекту в веб-технологіях:

– проблеми конфіденційності даних: Збір та аналіз даних користувачів може викликати питання про конфіденційність та захист даних. Неправильне використання особистих даних може призвести до порушень приватності;

– алгоритмічна справедливість: ШІ може піддаватися упередженню та нерівності, що може викликати проблеми в алгоритмічній справедливості та дискримінації;

– безпека та вразливості: Веб-застосунки, що працюють з ШІ, можуть стати об'єктом кібератак та зловмисних дій, що потребує додаткових заходів безпеки;

– втрата робочих місць: Автоматизація процесів веб-технологій з використанням ШІ може призвести до зменшення потреби в людській праці, що може мати наслідком втрату робочих місць;

Штучний інтелект в веб-технологіях надає безліч переваг, але також створює ряд викликів, пов'язаних із безпекою, етикою та конфіденційністю даних. Розуміння цих аспектів допомагає максимально ефективно використовувати ШІ в веб-технологіях, забезпечуючи високий рівень сервісу та захист інтересів користувачів.

1.3 Історія розвитку штучного інтелекту

Розвиток історії штучного інтелекту включає безліч важливих моментів та періодів, які відзначаються відкриттями, науковими досягненнями і значущими

подіями. Ось кілька ключових моментів у розвитку штучного інтелекту:

Виникнення поняття "штучний інтелект" (1956 рік): Термін "штучний інтелект" був вперше введений на конференції Дартмутського коледжу в 1956 році, де вчені обговорювали можливість створення машин, здатних моделювати людський розум та робити розумові операції.

Перші успішні інтелектуальні системи (1950-60-ті роки): В цей період були створені перші програми та інтелектуальні системи, зокрема програма "Логічний теорематичний доказатель" Джона Маккарті та "Геометричний теорематичний доказатель" Герберта Гроуера.

Поява експертних систем (1970-80-ті роки): В цей період були розроблені перші експертні системи, які були здатні розв'язувати завдання у спеціалізованих галузях, такі як медицина та фінанси.

Період зростання інтересу (1990-2000-ті роки): Виникло значне інтересу до штучного інтелекту, і було проведено багато досліджень у сфері машинного навчання та нейронних мереж.

Зростання інтересу до глибокого навчання (починаючи з 2010-х років): З'явилися штучні нейронні мережі, які стали використовуватися для розв'язання різних завдань, включаючи розпізнавання образів та природної мови.

Використання ШІ в різних сферах: Сучасний ШІ застосовується в медицині, автоматизації виробництва, фінансах, маркетингу, транспорті та багатьох інших галузях [2].

Видатні досягнення: Великі досягнення включають перемогу штучного інтелекту в грі Го і грі "Jeopardy!", а також створення роботів і асистентів, які взаємодіють з людьми в різних сферах.

Розвиток етичних питань та регулювання: Розвиток штучного інтелекту також супроводжується ростом етичних питань та потребою в регулюванні використання інтелектуальних систем у важливих галузях, таких як медицина та автомобільна промисловість.

Історія розвитку штучного інтелекту свідчить про постійний розвиток цієї галузі і великий потенціал для змін у багатьох аспектах сучасного життя.

1.4 Популярні моделі які вже використовуються людьми

ChatGPT – мовна модель для написання текстів, перекладів, відповідей на запитання. Одним із найвідоміших і найкращих інструментів для створення контенту за допомогою штучного інтелекту є ChatGPT, який з кінця минулого року б'є рекорди популярності. Це рішення, як відомо, розроблене компанією OpenAI. Інструмент генерує текстові відповіді на поставлені запитання та використовується, серед іншого, для обслуговування клієнтів, створення такого контенту, як статті чи описи продуктів, переклад текстів і навіть створення простих програм чи пошук помилок у коді.

Чат-бот Bing AI — це інструмент, що використовує технологію штучного інтелекту на основі моделі GPT, створеної Microsoft. Ним можна користуватися через веббраузер Microsoft Edge. Оновіть його до останньої версії, а потім натисніть значок Bing у верхньому правому куті. Інструмент може, наприклад, підсумовувати статті на вебсайті, який ви переглядаєте, з ним також можна спілкуватись, як у випадку з ChatGPT. Штучний інтелект Bing також може створювати зображення на основі заданого опису.

Midjurney — це інструмент, який використовується для створення зображень, і, як і рішення, запропоноване компанією OpenAI, він дуже популярний. Midjurney може створювати як фотореалістичні, так і абстрактні зображення. Для цього все, що вам потрібно зробити, це ввести відповідну команду, яка є детальним описом очікуваного зображення. Щоб скористатися цим інструментом, потрібно використовувати додаток Discord, тобто популярний месенджер, і приєднатися до сервера Midjourney.

Midjourney має деякі обмеження, пов'язані з популярністю платформи.

Щоб зберігати працездатність серверів, розробники надають безкоштовний доступ, обмежений 25-ма операціями або спробами. До цього ліміту включаються будь-які команди, які вимагають використання нейронних мереж, включаючи повторну генерацію тієї ж картинки з іншим результатом та функцію збільшення зображення. Тому рекомендуємо використовувати платну версію, базова підписка на яку коштує $10 на місяць або $96 на рік.

D-ID пропонує зовсім інші можливості, ніж інструменти, описані вище. Якщо ChatGPT генерує текст, а Midjurney працює з зображеннями, то D-ID — онлайн-програма, що може генерувати обличчя будь-якого персонажа, який при цьому ще й промовлятиме текст, введений користувачем. Інструмент інтегровано з уже згаданим ChatGPT, завдяки чому текст, згенерований інструментом OpenAI, може промовляти віртуальний персонаж, що робить отримання цього вмісту набагато привабливішим.

GitHub Copilot – це інструмент для написання коду на основі штучного інтелекту, який покликаний допомогти розробникам писати код більш ефективно. Він розроблений GitHub у співпраці з OpenAI і позиціонується як “партнер програміста зі штучним інтелектом”. Він використовує OpenAI Codex, щоб надавати підказки та автозавершення розробникам під час написання коду, що пришвидшує виконання програмних завдань. Таким чином, Copilot – це не просто інструмент автозавершення, він пропонує ширший спектр пропозицій, заснованих на контексті коду, що пишеться. Copilot може пропонувати цілі функції або рядки коду, аналізуючи контекст коду, що пишеться, надаючи розробникам швидший і простіший спосіб написання коду і зменшуючи необхідність шукати рішення деінде.

1.5 Алгоритм роботи нейронної мережі

Нейрон - це обчислювальна одиниця, яка отримує інформацію, виконує над нею прості обчислення і передає її далі. Вони поділяються на три основні

типи: вхідний (синій), прихований (червоний) і вихідний (зелений). Також є нейрон зміщення і контекстний нейрон, про які ми поговоримо в наступній статті [3]. У тому випадку, коли нейромережа складається з великої кількості нейронів, вводять термін шару. Відповідно, є вхідний шар, який отримує інформацію, n прихованих шарів (зазвичай їх не більше ніж 3), які її обробляють, і вихідний шар, який виводить результат. У кожного з нейронів є 2 основні параметри: вхідні дані (input data) і вихідні дані (output data). У разі вхідного нейрона: input=output. В інших, в поле input потрапляє сумарна інформація всіх нейронів з попереднього шару, після чого, вона нормалізується, за допомогою функції активації і потрапляє в поле output. Зображення нейроної мережі знаходиться на рисунку 1.1.

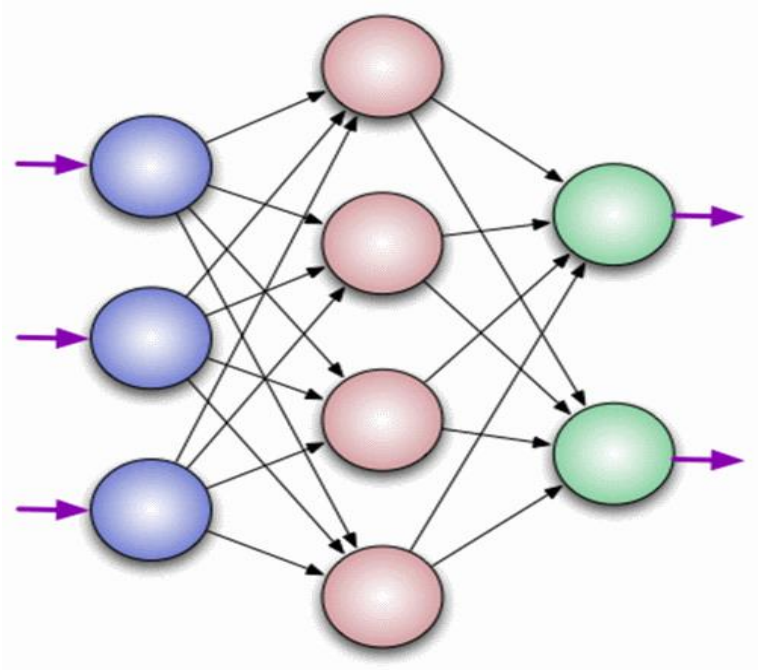


Рисунок 1.1 – Нейрона мережа

Нейрони оперують числами в діапазоні [0,1] або [-1,1]. А як же, ви запитаєте, тоді обробляти числа, які виходять із цього діапазону? На цьому етапі найпростіша відповідь - це розділити 1 на це число. Цей процес називається норма-

лізацією, і він дуже часто використовується в нейронних мережах. Приклад сінапсіса зображений на рисунку 1.2.

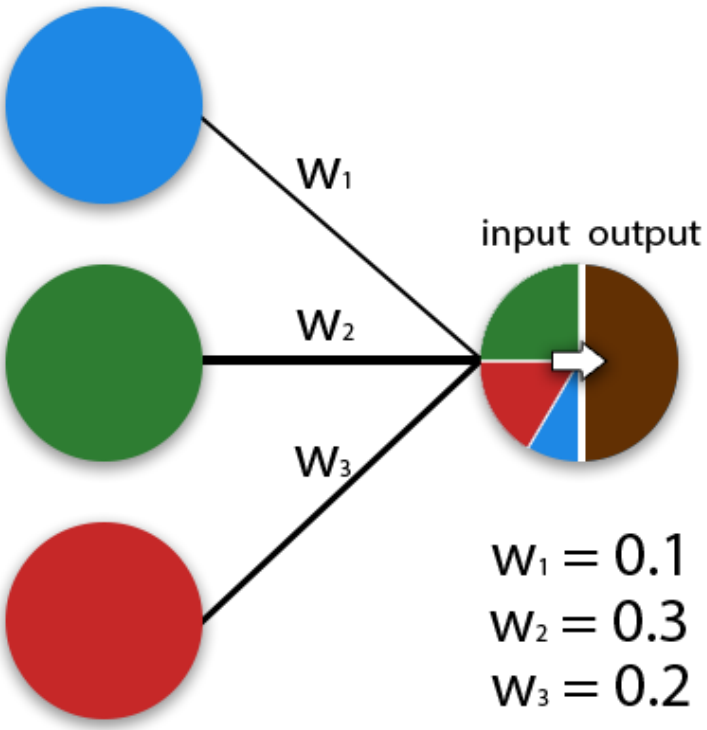
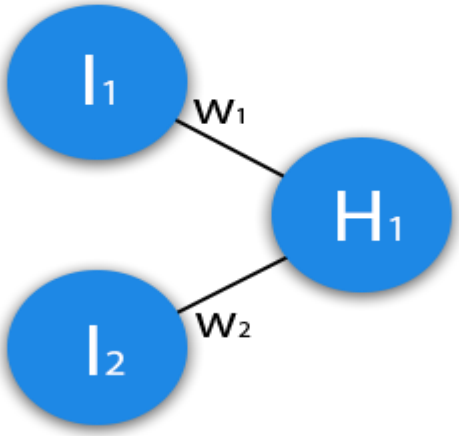


Рисунок 1.2 – Сінапсіс

Синапс це зв'язок між двома нейронами. У синапсів є 1 параметр - вага. Завдяки йому, вхідна інформація змінюється, коли передається від одного нейрона до іншого. Припустимо, є 3 нейрони, які передають інформацію наступному [4]. Тоді у нас є 3 ваги, що відповідають кожному з цих нейронів. У того нейрона, у якого вага буде більшою, та інформація і буде домінуючою в наступному нейроні (приклад - змішання кольорів). Насправді, сукупність ваг нейронної мережі або матриця ваг - це своєрідний мозок усієї системи. Саме завдяки цим вагам, вхідна інформація обробляється і перетворюється на результат. Під час ініціалізації нейронної мережі, ваги розставляються у випадковому порядку. Передача інформації між нейронами зображена на рисунку 1.3.



$$1)H_{1\,input} = (I_1 * w_1) + (I_2 * w_2)$$

$$2)H_{1\,output} = f_{activation}(H_{1\,input})$$

Рисунок 1.3 – Передача інформації між нейронами

У цьому прикладі зображено частину нейронної мережі, де буквами I позначено вхідні нейрони, буквою H - прихований нейрон, а буквою w - ваги. З формули видно, що вхідна інформація - це сума всіх вхідних даних, помножених на відповідні їм ваги. Тоді дамо на вхід 1 і 0. Нехай w1=0.4 і w2 = 0.7 Вхідні дані нейрона H1 будуть такими: 1*0.4+0*0.7=0.4. Тепер, коли в нас є вхідні дані, ми можемо отримати вихідні дані, підставивши вхідне значення у функцію активації (докладніше про неї далі). Тепер, коли у нас є вихідні дані, ми передаємо їх далі. І так, ми повторюємо для всіх шарів, поки не дійдемо до вихідного нейрона. Запустивши таку мережу вперше, ми побачимо, що відповідь далека від правильної, тому що мережа не натренована. Щоб поліпшити результати, ми будемо її тренувати. Але перш ніж дізнатися, як це робити, давайте введемо кілька термінів і властивостей нейронної мережі.

Функція активації - це спосіб нормалізації вхідних даних (ми вже говорили про це раніше). Тобто, якщо на вході у вас буде велике число, пропустивши його

через функцію активації, ви отримаєте вихід у потрібному вам діапазоні [5]. Функцій активації досить багато, тому ми розглянемо найосновніші: Лінійна, Сигмоїд (Логістична) і Гіперболічний тангенс. Головні їхні відмінності - це діапазон значень. Лінійна функція зображена на рисунку 1.4.

$$f(x) = x$$

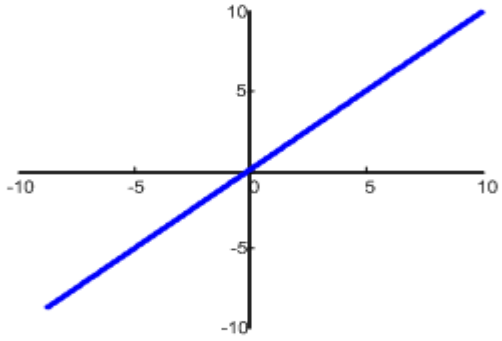


Рисунок 1.4 – Лінійна функція

Ця функція майже ніколи не використовується, за винятком випадків, коли потрібно протестувати нейронну мережу або передати значення без перетворень.

Це найпоширеніша функція активації, її діапазон значень [0,1]. Саме на ній показано більшість прикладів у мережі, також її іноді називають логістичною функцією. Відповідно, якщо у вашому випадку присутні від'ємні значення (наприклад, акції можуть йти не тільки вгору, а й униз), то вам знадобиться функція, яка захоплює і від'ємні значення. Сигмоїд функція зображена на рисунку 1.5.

Має сенс використовувати гіперболічний тангенс, тільки тоді, коли ваші значення можуть бути і від'ємними, і позитивними, оскільки діапазон функції [-1,1]. Використовувати цю функцію тільки з позитивними значеннями недоцільно,

оскільки це значно погіршить результати вашої нейромережі. Гіперболічний тангенс зображенний на рисунку 1.6.

$$f(x) = \frac{1}{1+e^{-x}}$$

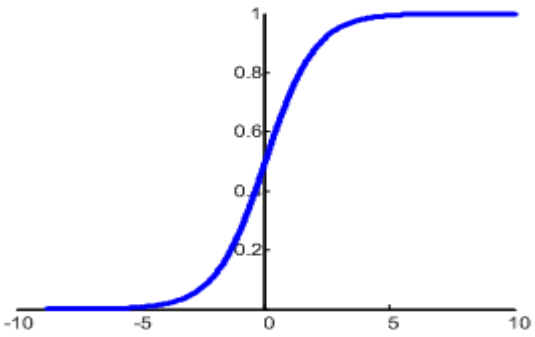


Рисунок 1.5 – Сигмоїд функція

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

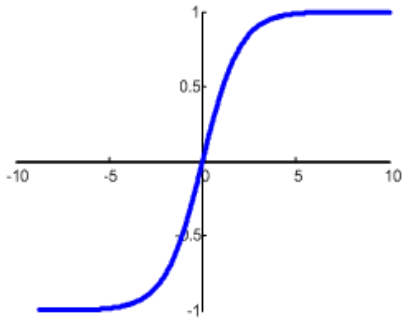


Рисунок 1.6 - Гіперболічний тангенс

Тренувальний сет - це послідовність даних, якими оперує нейронна мережа. У нашому випадку виключного або (xor) у нас всього 4 різних результати, тобто у нас буде 4 тренувальних сети: 0xor0=0, 0xor1=1, 1xor0=1,1xor1=0 [6].

Ітерація - це своєрідний лічильник, який збільшується щоразу, коли нейронна мережа проходить один тренувальний сет. Іншими словами, це загальна кількість тренувальних сетів пройдених нейронною мережею.

Епоха - під час ініціалізації нейронної мережі цю величину встановлюють у 0 і вона має стелю, яку задають вручну. Що більша епоха, то краще натренована мережа і, відповідно, її результат. Епоха збільшується щоразу, коли ми проходимо весь набір тренувальних сетів, у нашому випадку, 4 сетів або 4 ітерацій.

Помилка - це відсоткова величина, що відображає розбіжність між очікуваною та отриманою відповідями. Помилка формується кожну епоху і повинна йти на спад. Якщо цього не відбувається, значить, ви щось робите не так. Помилку можна обчислити різними способами, розглянемо три основні способи: Mean Squared Error (далі MSE), Root MSE і Arctan. Тут немає будь-якого обмеження на використання, як у функції активації. Варто лише враховувати, що кожен метод рахує помилки по різному. У Arctan помилка майже завжди буде більшою, оскільки він працює за принципом: що більша різниця, то більша помилка. У Root MSE буде найменша помилка, тому найчастіше використовують MSE, яка зберігає баланс в обчисленні помилки [7].

2. ДОСЛІДЖЕНЯ ВИКОРИСТАННЯ ШІ У ВЕБ ТЕХНОЛОГІЯХ

2.1 Використання ШІ у веб технологіях

В останні роки штучний інтелект набув величезного значення, активно впроваджуючись у різні сфери нашого повсякденного життя. У контексті веб-розробки він також відіграв значну роль. Завдяки використанню ШІ стало можливим впроваджувати нові стратегії розвитку, покращувати користувальницький досвід та розробляти безліч перспективних програмних рішень. Штучний інтелект надає безліч інструментів, що значно спрощують та покращують різні етапи процесу веб-розробки. Нижче наведені декілька прикладів.

Чатботи і віртуальні помічники є значним прогресом у сфері веб-розробки, можливим завдяки силі штучного інтелекту. Ці інноваційні програми, засновані на технологіях обробки природної мови та машинного навчання, здатні вести діалог з відвідувачами, який не відрізняється від спілкування з реальною людиною. Боти здатні відповідати на запитання, надсилати користувачів на відповідні вебсторінки і навіть допомагати у виконанні різних завдань, таких як заповнення форм та запис на прийом. Такі технологічно просунуті рішення не тільки скорочують тимчасові витрати, а й підвищують задоволення користувачів, покращуючи загальний досвід взаємодії з веб-ресурсами.

Інструменти створення контенту, керовані штучним інтелектом, істотно впливають на галузі, де необхідно забезпечувати високий обсяг виробництва контенту. Ці інноваційні інструменти застосовують алгоритми, здатні генерувати текст, наближений до людського, включаючи статті, опис продуктів і навіть фрагменти коду. І хоча ШІ не може повністю замінити творчий потенціал і редакторські навички людини, вже зараз вона є дуже корисним помічником у швидкому та точному створенні контенту, заощаджуючи час та максимізуючи ефективність виробництва.

Штучний інтелект широко застосовується для предиктивної аналітики, забезпечуючи веб-розробникам ефективніше розуміння поведінки користувачів. Шляхом аналізу попередньої активності користувача алгоритми ШІ здатні передбачати майбутні кроки відвідувача. У сфері електронної комерції, наприклад, передиктивна аналітика дає веб-сайтам можливість пропонувати персональні рекомендації щодо продуктів, що сприяє збільшенню продажів та підвищенню задоволеності клієнтів.

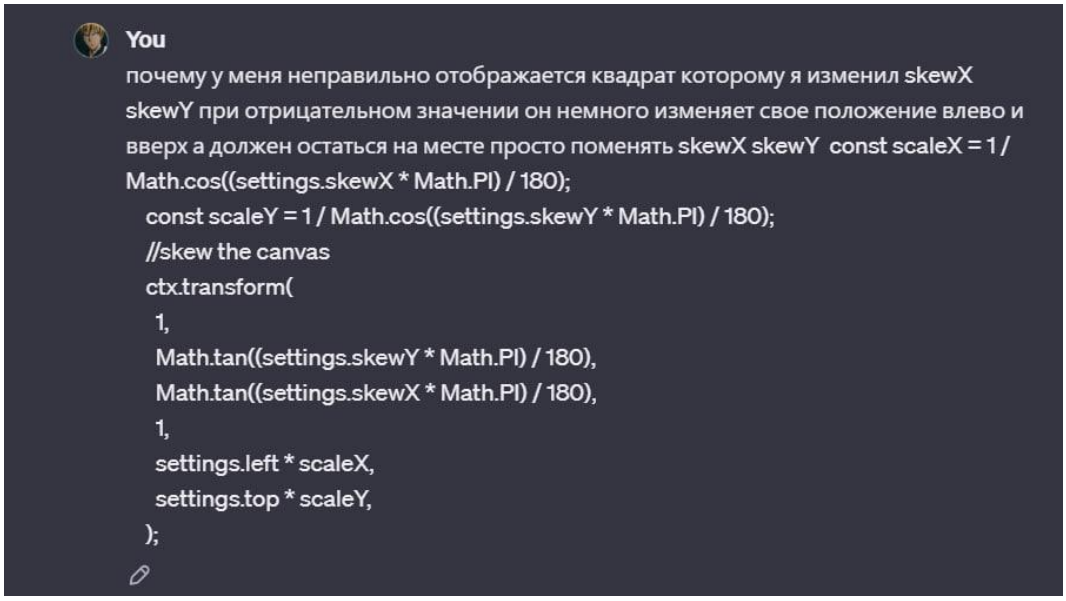
Проекти веб-розробки часто потребують серйозних та тривалих етапів тестування та налагодження. Інструменти, керовані штучним інтелектом, значною мірою оптимізують цей процес, автоматизуючи виявлення та усунення помилок. Ці інноваційні інструменти здатні виявляти різні проблеми, включаючи зламані посилання, елементи, що повільно завантажуються, і вразливості в безпеці, що в свою чергу підвищує надійність веб-сайтів.Їх використання не тільки прискорює процес розробки, але й забезпечує високий рівень якості та функціональності вебдодатків.

Штучний інтелект застосовується у сфері веб-безпеки для виявлення та протидії загрозам у реальному часі. Він здатний аналізувати зразки підозрілої поведінки, виявляти потенційні вразливості та забезпечувати захист веб-сайтів від кібератак, включаючи атаки типу DDoS та витоку даних.

2.2 Переваги веб-розробки з використанням штучного інтелекту

Веб-розробка із застосуванням штучного інтелекту надає ряд переваг, які сприяють більш ефективному, зручному для користувача та економічному процесу.

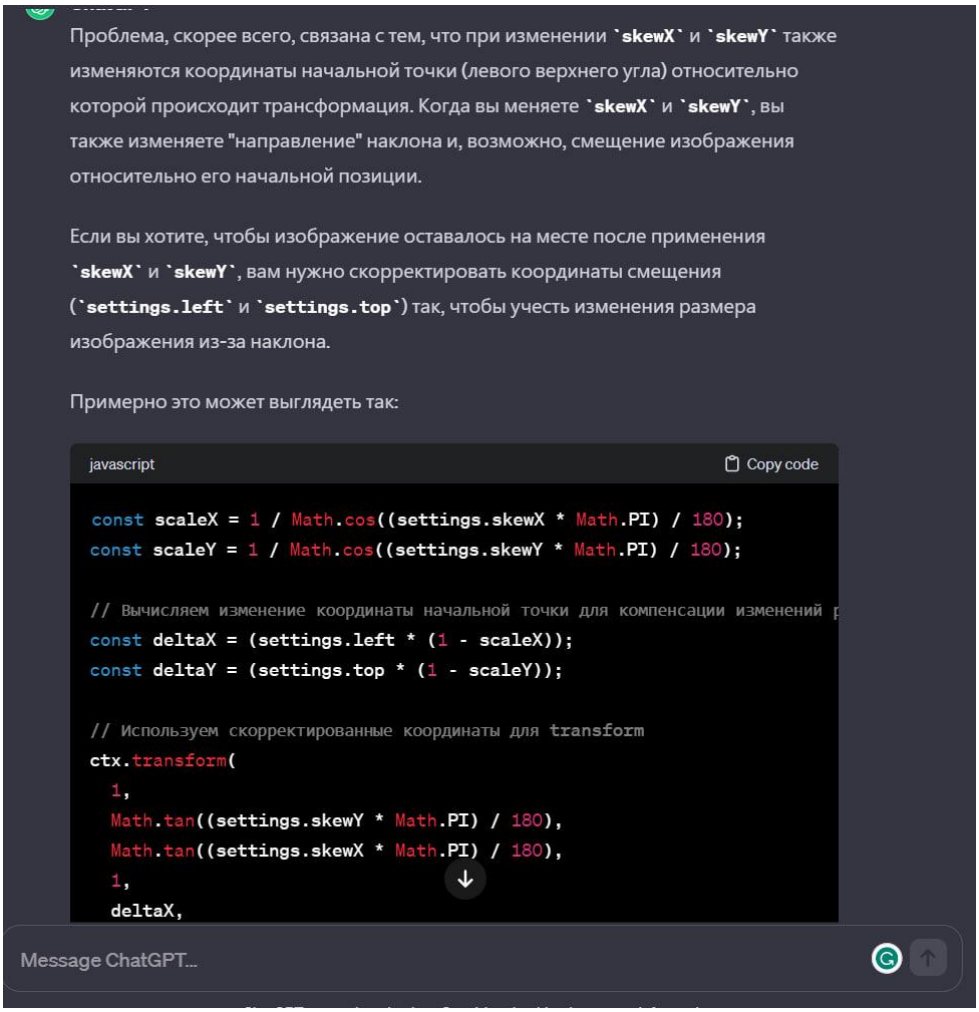
Однією з найбільш істотних переваг використання штучного інтелекту у веб-розробці є автоматизація однотипних та трудомістких завдань. ШІ здатний взяти на себе створення контенту, тестування та аналіз даних, дозволяючи розробникам зосередити увагу на більш креативних та стратегічно важливих аспектах веб-розробки. Приклад використання зображений на рисунках 2.1 та 2.2.



You

почему у меня неправильно отображается квадрат которому я изменил skewX skewY при отрицательном значении он немного изменяет свое положение влево и вверх а должен остаться на месте просто поменять skewX skewY const scaleX = 1 / Math.cos((settings.skewX * Math.PI) / 180);
  const scaleY = 1 / Math.cos((settings.skewY * Math.PI) / 180);
  //skew the canvas
  ctx.transform(
   1,
   Math.tan((settings.skewY * Math.PI) / 180),
   Math.tan((settings.skewX * Math.PI) / 180),
   1,
   settings.left * scaleX,
   settings.top * scaleY,
  );

Рисунок 2.1 – Приклад питання до чату GPT



Проблема, скорее всего, связана с тем, что при изменении **`skewX`** и **`skewY`** также изменяются координаты начальной точки (левого верхнего угла) относительно которой происходит трансформация. Когда вы меняете **`skewX`** и **`skewY`**, вы также изменяете "направление" наклона и, возможно, смещение изображения относительно его начальной позиции.

Если вы хотите, чтобы изображение оставалось на месте после применения **`skewX`** и **`skewY`**, вам нужно скорректировать координаты смещения (**`settings.left`** и **`settings.top`**) так, чтобы учесть изменения размера изображения из-за наклона.

Примерно это может выглядеть так:

javascript Copy code

```
const scaleX = 1 / Math.cos((settings.skewX * Math.PI) / 180);
const scaleY = 1 / Math.cos((settings.skewY * Math.PI) / 180);

// Вычисляем изменение координаты начальной точки для компенсации изменений р
const deltaX = (settings.left * (1 - scaleX));
const deltaY = (settings.top * (1 - scaleY));

// Используем скорректированные координаты для transform
ctx.transform(
  1,
  Math.tan((settings.skewY * Math.PI) / 180),
  Math.tan((settings.skewX * Math.PI) / 180),
  1,
  deltaX,
```

Message ChatGPT...

Рисунок 2.2 – Відповідь на питання чату GPT

Функції персоналізації та чатботи, керовані штучним інтелектом, суттєво покращують показники взаємодії користувачів із веб-сайтами. Персоналізація надає контент та рекомендації відповідно до індивідуальних уподобань кожного користувача, у той час як чати забезпечують миттєву допомогу та підтримку.

Системи штучного інтелекту збирають та аналізують величезні обсяги даних, надаючи цінні інсайти про переваги користувачів, ефективність веб-сайту та інші ключові показники. Зібрана інформація допомагає розробникам приймати обґрунтовані рішення щодо подальшого дизайну веб-сайту, контенту та функціональності, внаслідок чого покращується продуктивність та підвищуються конверсійні показники.

Автоматизація різних аспектів веб-розробки за допомогою штучного інтелекту не лише знижує витрати, а й оптимізує використання часу, скорочує витрати людського ресурсу та мінімізує ймовірність помилок. Така ефективність робить веб-розробку доступнішою для компаній з обмеженими бюджетами, а також надає малим підприємствам можливість конкурувати на цифровому ринку.

ШІ інструменти для використання у веб розробці :

- GPT-3 (OpenAI): Розроблений OpenAI, GPT-3 є передовою мовною моделлю, здатною генерувати текст, навряд чи відмінний від людського. Найчастіше GPT-3 застосовується для створення контенту і розробки чатботів, але також може бути використаний для відповідей на запитання, що часто задаються;
- Wix ADI: Wix ADI (Artificial Design Intelligence) – це інструмент, розроблений для автоматизованого створення веб-сайтів за допомогою ШІ. Wix ADI аналізує переваги користувача та його бізнес-потреби та автоматично створює унікальний дизайн сайту. Цей інструмент ідеально підходить для тих, хто хоче швидко створити стильний та функціональний сайт без глибоких знань веб-розробки;
- Lighthouse: Lighthouse, розроблений Google, є автоматизованим інстру-

ментом аналізу продуктивності веб-сайтів. Використовуючи ШІ, Lighthouse проводить ретельний аналіз веб-сторінок та надає розробникам рекомендації щодо покращення продуктивності, доступності та інших важливих аспектів веб-розробки;

- Dialogflow (Google): Dialogflow — популярна платформа для створення чатів та віртуальних помічників для веб-сайтів. Він використовує обробку природної мови для розуміння та відповідей на запити користувачів, що робить її важливим інструментом для покращення взаємодії з відвідувачами веб-сайту;
- TensorFlow.js: TensorFlow.js — це бібліотека JavaScript, яка дозволяє інтегрувати моделі машинного навчання та штучний інтелект безпосередньо до веб-додатків. Розробники можуть створювати та навчати моделі нейронних мереж, а потім використовувати їх для обробки даних безпосередньо у браузері;
- Amazon Lex (Amazon Web Services): Amazon Lex – це сервіс, який надається Amazon Web Services (AWS). Він надає масштабовану та гнучку платформу для створення розмовних інтерфейсів, які можуть бути легко інтегровані в різні програми, включаючи веб-сайти та мобільні програми.

2.3 Обмеження використання ШІ у веб-розробці

Незважаючи на низку переваг, використання ШІ у веб-розробці також пов'язане з кількома обмеженнями та труднощами. Давайте розглянемо деякі з них:

- відсутність творчої складової: ШІ позбавлений творчих та інноваційних здібностей людини. Хоча він може автоматизувати однотипні завдання та генерувати контент, у нього немає можливості мислити ширше, вигадувати унікальні концепції дизайну або створювати нові рішення;
- проблеми безпеки: Системи, які використовують штучний інтелект, можуть стати легкою метою зловмисників, особливо якщо вони не посилені спеціальними механізмами безпеки. Важливо постійно стежити і вживати заходів щодо

забезпечення безпеки, щоб запобігти можливим загрозам, таким як впровадження шкідливого коду або маніпуляції даними;

– ресурсовитратність: Деякі інструменти ШІ у веб-розробці можуть бути вкрай ресурсомісткими і вимагати значних обчислювальних потужностей та обсягу оперативної пам'яті. Цей факт може створювати додаткові перешкоди для стартапів або тих, хто має обмежений бюджет;

– вирішення складних проблем: Системи штучного інтелекту успішно справляються з вирішенням конкретних, чітко визначених завдань. Однак при зіткненні з комплексними, багатоаспектними проблемами вони можуть зіткнутися з труднощами, які потребують глибокого розуміння контексту, логічного мислення та людської думки. Приклад задач коли ШІ не може вирішити проблему зображений на рисунках 2.3 та 2.4.

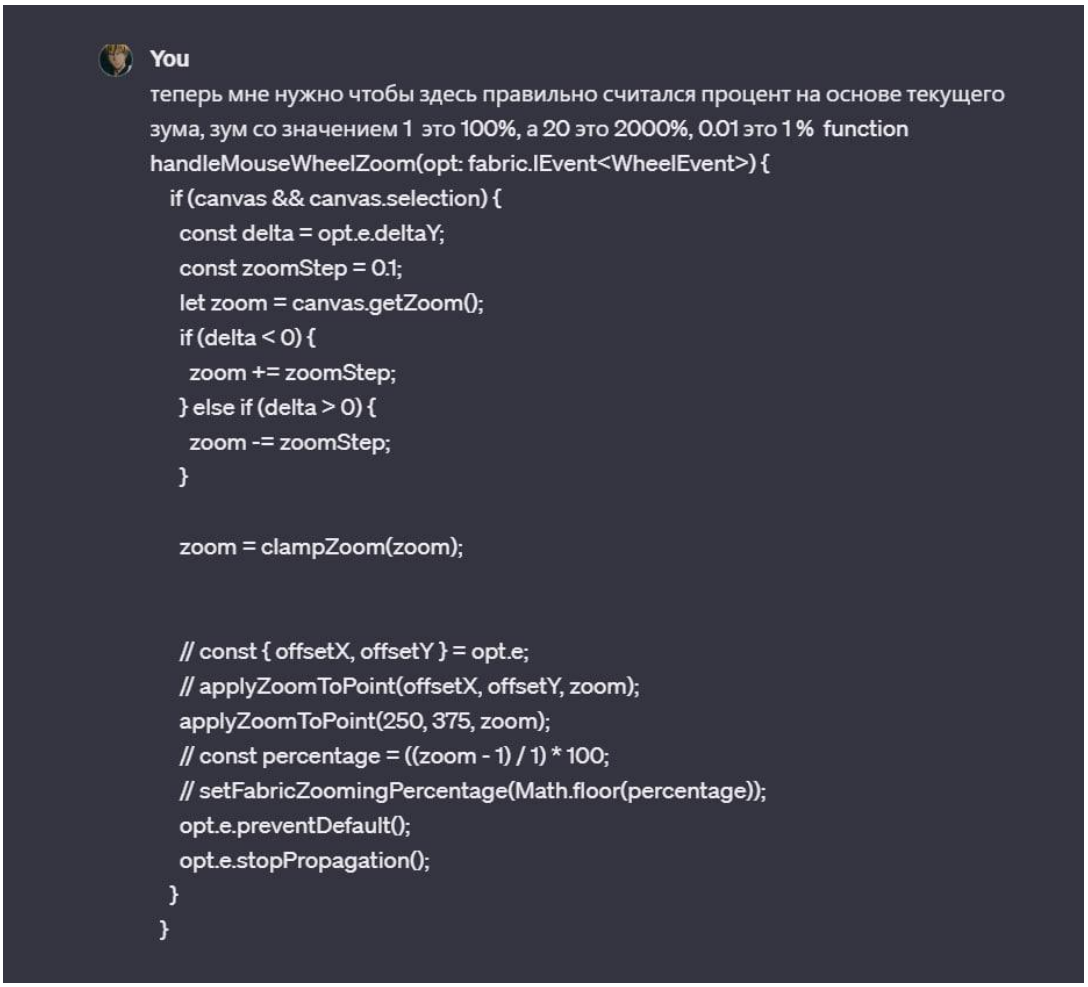


Рисунок 2.3 – Питання до чату де потрібно коректно сформулювати формулу

$$\text{Новый зум} = \text{Текущий зум} + (\text{Текущий зум} - 1) \times 0.1 \times (20 - 1)$$

Вот как это может быть реализовано в вашем коде:

typescript Copy code

```typescript
function handleMouseWheelZoom(opt: fabric.IEvent<WheelEvent>) {
    if (canvas && canvas.selection) {
        const delta = opt.e.deltaY;
        const zoomStep = 0.1;
        let zoom = canvas.getZoom();

        if (delta < 0) {
            // Увеличиваем зум на 10% вверх при прокрутке вверх
            zoom += (zoom - 1) * zoomStep * (20 - 1);
        } else if (delta > 0) {
            // Уменьшаем зум на 10% вниз при прокрутке вниз
            zoom -= (zoom - 1) * zoomStep * (20 - 1);
        }

        zoom = clampZoom(zoom);

        const percentage = ((zoom - 1) / (20 - 1)) * 100;
        setFabricZoomingPercentage(Math.floor(percentage));

        const offsetX = 250; // Значение для offsetX - замените на нужное
        const offsetY = 375; // Значение для offsetY - замените на нужное
        applyZoomToPoint(offsetX, offsetY, zoom);

        opt.e.preventDefault();
```

Рисунок 2.4 – Невірна відповідь через невірну формулу

3. ФУНКЦІОНАЛ ВЕБ ДОДАТКУ ІЗ ШТУЧНИМ ІНТЕЛЕКТОМ, ОПИС МОДЕЛІ ШІ ТА ТЕХНОЛОГІЇ ДЛЯ ЙОГО РОЗРОБКИ

3.1 Функціонал веб додатку

Веб додаток який буде реалізован це кастомізатор футболок з використанням генератору зображень за допомогою моделі Stable diffusion.

Функціонал який потрібно буде реалізувати:

- відображення 3d моделі футболки;
- обертання футболки в просторі;
- зміна кольору футболки;
- завантаження логотипу футболки;
- генерація логотипу через модель Stable diffusion;
- серверний додаток який буде відправляти запит на генерацію зображення та повертати його на клієнт.

3.2 Мова програмування Javascript

Для реалізаціє веб додатку кастомізатору футболок була вибрана мова програмування Javascript. JS використовується у всіх браузерах і є незамінним інструментом на даний час для надання інтерактивності веб-сторінкам [9].

За допомогою нього доступні для виконання наступні функції:

- можливість змінювати сторінки браузерів;
- додавання чи видалення тегів;
- зміна стилів сторінки;
- інформація про дії користувача на сторінці;
- запит на доступ до випадкової частини вихідного коду сторінки;
- внесення змін до цього коду;

- виконання дії з файлами cookie.

Переваги Javascript:

- жоден сучасний браузер не обходиться без підтримки JavaScript;
- легкий у освоєнні;
- корисні функціональні налаштування які полегшують роботу з html;
- мова, що постійно вдосконалюється, з'являються нові рішення та “синтаксичний цукор”;
- перспектива використання мови у процесі навчання програмування та інформатики;
- компілювати JavaScript можна навіть в браузері;
- велика кількість готових бібліотек .

Недоліки JavaScript:

- знижений рівень безпеки через повсюдний і вільний доступ до вихідних кодів популярних скриптів [10];
- динамічна типізація, що виливається у велику кількість помилок через типи данних;
- не підтримує віддалений доступ, тому не застосовується для мережних програм.

3.3 React бібліотека для Frontend розробки

В якості допоміжною технологія для розробки FE для мови програмування JavaScript була вибрана бібліотека React.js. Вона є найкращим вибором через такі переваги:

- повторне застосування компонентів;
- JSX синтаксис який дозволяє об’єднати html розмітку і JS код;
- східний потік даних;
- віртуальна об'єктна модель документа;

3.3.1 Повторне використання компонентів

Принцип роботі з React.js є створювання багаторазових компонентів найчастіше це код який повторюється та який потрібно використовувати у інших частинах сайту, також можна передавати у ці компоненти дані за допомогою циклу приклад на рисунку 3.1 . У цьому прикладі ми пробігаємося по масиву об’єктів відсортованих постів та передаємо данні по ключам у інший компонент де використовуємо ці данні у компоненті.

```
<CatalogFiltersContainer>
  {filters &&
    filters.data.map((filter :CatalogFilterType ) => (
      <CatalogFilter
        isLoading={false}
        newFilterData={undefined}
        filter={filter}
        key={filter.modelId}
        onChange={onChange}
      />
    ))}
</CatalogFiltersContainer>
```

Рисунок 3.1 – Передача параметрів масива об’єктів через цикл у компонен

3.3.2 JSX синтаксис

JSX – це абстракція, яка дозволяє використовувати синтаксис HTML всередині вашого коду JavaScript і за допомогою якої ви можете створювати компоненти React, які виглядають як стандартна розмітка HTML. JSX – це мова шаблонів для елементів React, тому вона є основою для будь-якої розмітки, яку React відображатиме у вашій програмі. Тобто ми маємо змогу передавати в атрибути html тегів значення із Javascript змінних, вставляти у самі html теги значення із Javascript змінних, відображати потрібні компоненти у циклі , переда-

вати у компоненти дані, відображати тільки при певній умові, або навіть повертати JSX розмітку як результат роботи функції [11]. JSX дозволяє React оптимізувати процес рендерингу компонентів, що призводить до більш ефективної роботи і покращує продуктивність додатків. Приклад JSX розмітки наведений на рисунку 3.2.

```
getValue: ({ id :ID  }) => {
  const dispatch :Dispatch<AnyAction>  = useDispatch();
  return (
    <TableControlWrapper>
      <PermissionGate permissions={[UserPermissionsEnum.LocationEdit]}>
        <TableEditAction
          dataTestId={`edit_location_action`}
          onClick={() :void  => {
            dispatch(
              modalSlice.actions.onShowModal( payload: {
                title: 'Edit Locations',
                ContentComponent: AddLocationModal,
                size: ModalSizes.sm,
                onOpen: () :void  => {
                  dispatch(locationsSlice.actions.onOpenEditLocationModal( payload: { locationId: id }));
                },
                onClose: () => dispatch(locationsSlice.actions.onClearForm()),
              }),
            );
          }}
        />
      </PermissionGate>
      <PermissionGateAdminOnly>
        <RowAction icon={<CloseIcon />} onClick={() => onDelete(id)} />
      </PermissionGateAdminOnly>
      <StyledTableDetailAction
        direction={
          useSelector(selectSelectedLocationId) == id
            ? ArrowDirection.Left
            : ArrowDirection.Right
        }
      />
    </TableControlWrapper>
```

Рисунок 3.2 – Приклад JSX розмітки

3.3.3 Східний потік даних

Односторонній потік даних у React – ще одна дуже корисна функція. Такий потік даних також називають "згори донизу" або "від батька до дитини". Пояснення досить просте – у програмі React дані між елементами передаються лише

одним способом. Східний потік даних запобігає помилкам, полегшує налагодження. Передавати данні між компонентами можна лише згори до низу від самого головного компоненту батьківського до компонентів потомків. React не дозволяє передавати компоненти знизу догори [12]. Приклад принципу як передаються дані у React можна подивитися на рисунку 3.3.

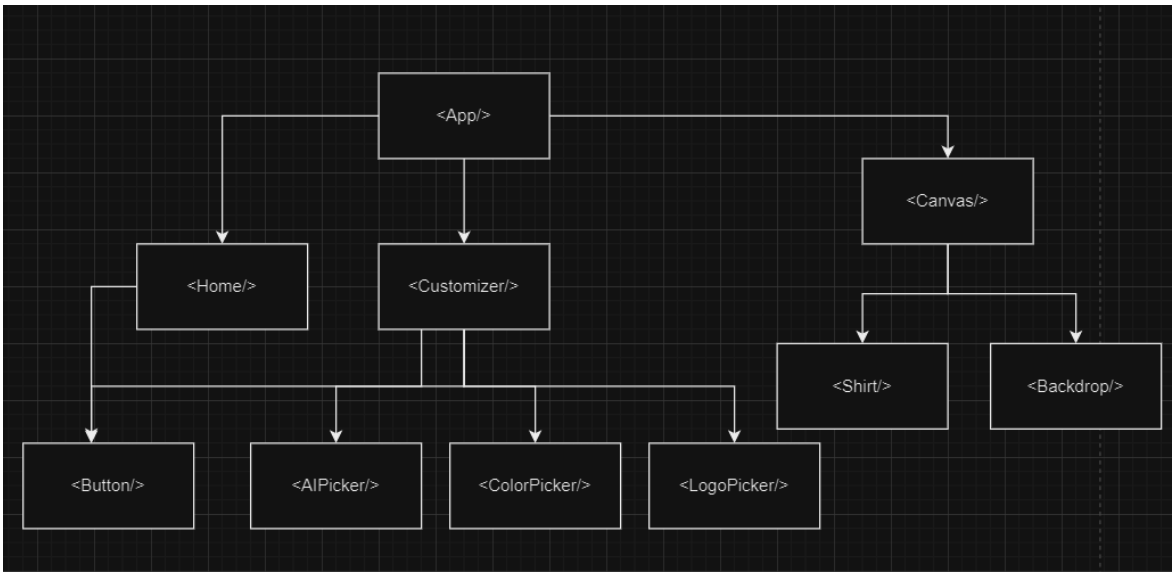


Рисунок 3.3 – Принцип передачі параметрів між компонентами

3.3.4 Віртуальна об'єктна модель документа

Браузер регулярно перевіряє будь-які зміни в DOM, елементи які входять в DOM, оновлюючи її належним чином. Зміна об'єктної моделі документа провокується безліччю факторів, наприклад введенням даних, HTTP-запитом, отриманням даних від API і так далі. Браузер оновлює DOM кожного разу, коли сторінка змінюється. Оскільки DOM сучасних сайтів величезні, оновлення займає багато часу, уповільнюючи загальну продуктивність веб-програми.

Замість повільної та незручної взаємодії безпосередньо з реальною об'єктною моделлю документа, React взаємодіє з її полегшеною копією — віртуальної DOM, тому реальна DOM оновлюється лише після взаємодії з віртуальною DOM.

Оновлення всього DOM, щоб зробити веб-сторінку “реактивною” — вкрай неефективним, оскільки споживає занадто багато ресурсів. Тому, власне, при зміні веб-сторінки (наприклад, в результаті запиту або дії користувача) React оновлює віртуальний спеціальний DOM.

React зберігає дві версії віртуального DOM — оновлений віртуальний DOM та його резервну копію, створену до оновлення. Після оновлення React порівнює обидві версії між собою, щоб знайти змінені елементи, а потім — оновлює частину реального DOM, що виключно змінилася.

Віртуальна DOM, на відміну від реального DOM, займає мало місця та швидко оновлюється, тим самим підвищуючи продуктивність програми.

Віртуальна DOM дозволяє сторінці негайно отримувати відповіді від сервера та відображати оновлення. Наприклад, Facebook застосовує технологію віртуального DOM для оновлення чатів та стрічок користувачів без перезавантаження сторінки. Принцип роботи зображений на рисунку 3.4.

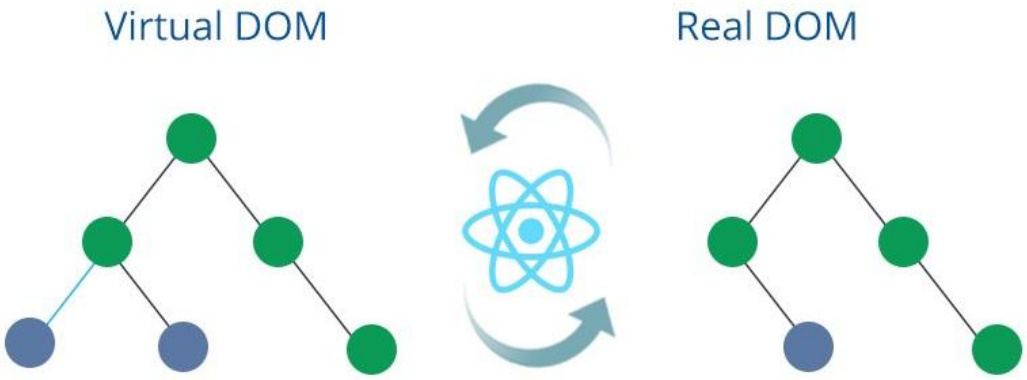


Рисунок 3.4 – Принцип роботи віртуального DOM

3.4 Typescript

TypeScript є надмножиною JavaScript і компілюється в останній. Фактично, після компіляції програму на TS можна виконувати у будь-якому сучасному браузері або використовувати разом із серверною платформою Node.js.

TypeScript відрізняється від JavaScript можливістю явного статичного призначення типів, підтримкою використання повноцінних класів (як у традиційних об'єктно-орієнтованих мовах), а також підтримкою підключення модулів, що покликане підвищити швидкість розробки, полегшити читання, рефакторинг та повторне використання коду, допомогти здійснювати пошук помилок на етапі розробки та компіляції, і, можливо, прискорити виконання програм [13].

TypeScript – це розширення мови ECMAScript 5. Додано наступні опції:

- анотації типові і перевірка їх узгодження на етапі компіляції;
- виведення типів;
- інтерфейси;
- перераховані типи;
- домішка;
- узагальнене програмування;
- модулі;
- скорочений синтаксис "стрілок" для анонімних функцій;
- додаткові параметри та параметри за замовчуванням;
- кортежі.

Задачі які виконує TS для полегшення роботи:

- відображає тип даних, щоб розробник міг висловити намір використати значення певного типу let title: string;
- контролює можливість використання методів значення у контексті нашої програми;
- повідомляє розробнику про помилкове використання операцій щодо змінних у редакторі, до запуску програми;
- дає можливість розробки у методології ООП. Стають доступними ключові слова private, protected, public, abstract, extends, implements;
- відкриває можливість узагальненого програмування. Додає таке поняття як “дженеріки” які дозволяють створювати компоненти з реалізацією алгоритмів

у загальному вигляді. Приклад коду на TS показаний на рисунку 3.5.

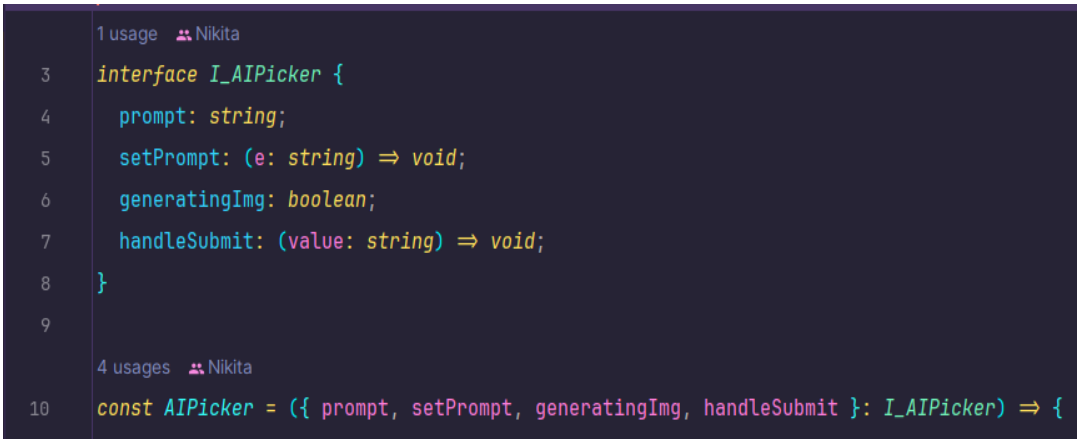


Рисунок 3.5 – Використання інтерфейсів у TypeScript

TS використовувася у цьому проекті оскільки JavaScript немає типів, контролювати і перевіряти всі параметри і змінні, які використовуються, буває досить складно. В той же час дуже легко зробити помилку в коді, наприклад, забути оголосити змінну, робити операції не с підходящим типом даних випадково викликати неіснуючу функцію або передати як параметр змінну, яка зламає весь код [14].

TypeScript спрощує читання коду та допомагає уникнути помилок, які можуть перетворити налагодження на кошмар.

3.5 Node.js на сервері

Node.js - це виконавче середовище (runtime environment) для JavaScript, яке дозволяє виконувати код на стороні сервера. При розробці Node.js за основу було взято двигун виконання JavaScript під назвою V8, який був створений компанією Google і використовувався в браузері Google Chrome. Оскільки після створення Node.js Javascript код можна запустити фактично в будь-якому середовищі, за допомогою цієї бібліотеки можна написати не лише фронтенд, а й серверну частину веб-програми[8].

Важливою особливістю Node.js є асинхронний характер. Термін асинхронний означає, що сервер, створений з використанням Node.js, не повинен чекати, поки дані повернуться, при виконанні різних внутрішніх запитів. При цьому він також має неблокуючий введення-виведення. Це означає, що кілька різних процесів можуть виконуватись паралельно, не блокуючи один одного. Обидві ці властивості роблять Node.js вкрай швидким і забезпечують кращий інтерфейс користувача.

У Node.js всі однопоточні запити і збираються в циклі обробки подій (Event loop). Це означає, що всі програми виконуються в одному потоці, починаючи з отримання запиту і закінчуючи виконанням необхідного завдання та надсиланням відповіді клієнту назад. Ця функція Node.js запобігає повторному завантаженню запитів і скорочує час їх обробки, що робить його більш економічним у використанні. Приклад коду Node.js на сервері зображений на рисунку 3.6

Рисунок 3.6 – Використання node js для запиту до стороннього API

3.6 Tree.js

Three.js - це бібліотека для роботи з тривимірною графікою у веб-розробці. За допомогою Three.js можна легко створювати та візуалізувати 3D-сцени прямо

в браузері[15]. Ось деякі ключові аспекти та можливості цієї бібліотеки:

– легкість використання: Three.js робить роботу з тривимірною графікою доступною і простою. Вона надає високорівневий API, що полегшує створення складних 3D-сцен навіть для розробників без глибоких знань в графіці;

– кросплатформеність: Three.js підтримується на різних платформах і працює в різних браузерах, що дозволяє створювати веб-додатки з тривимірною графікою для широкого кола аудиторії;

– велика спільнота та документація: Three.js має активну спільноту розробників та широку документацію. Це сприяє вирішенню проблем, обміну знаннями та постійному вдосконаленню веб-графіки;

– можливості візуалізації: Three.js дозволяє створювати різноманітні об'єкти та ефекти, такі як світло, тіні, текстури, анімації та багато іншого, що робить візуалізацію тривимірних об'єктів дуже реалістичною;

– підтримка VR та AR: Three.js інтегрована з технологіями віртуальної та доповненої реальності, що відкриває нові можливості для створення інноваційних додатків.

3.7 Модель Stable Diffusion

Stable Diffusion – це модель генеративного штучного інтелекту (генеративного II), за допомогою якої можна перетворити текст та деталізуючі підказки на унікальні фотореалістичні зображення. Спочатку вона була запущена у 2022 році. Stable Diffusion можна використовувати для створення не лише зображень, а й відеороликів та анімацій. Модель заснована на технології дифузії та використовує прихований простір, що значно знижує вимоги до обробки. Крім того, її можна запускати на настільних комп'ютерах чи ноутбуках, оснащених графічними процесорами. За допомогою трансферного навчання модель Stable Diffusion можна налаштувати під свої конкретні потреби, використовуючи для цього всього

п'ять зображень.

Що можна робити за допомогою Stable Diffusion?

Stable Diffusion забезпечує помітно покращену модель перетворення тексту на зображення. Ця модель широко доступна і вимагає значно меншої обчислювальної потужності, ніж багато інших моделей перетворення тексту зображення. Її можливості включають перетворення тексту на зображення та зображення на зображення, генерацію графічних творів, редагування зображень та створення відео.

Перетворення тексту на зображення. Це найпоширеніший спосіб використання Stable Diffusion. Модель Stable Diffusion створює зображення за допомогою текстової підказки. Ви можете створювати різні зображення, змінюючи значення затравки для генератора випадкових чисел або змінюючи режим шумоподавлення для різних ефектів.

Перетворення зображення на зображення. Ви можете створювати нові зображення за допомогою вхідного зображення та текстової підказки. Часто для цього використовується ескіз та підказка.

Створення графіки, ілюстрацій та логотипів. Використовуючи набір підказок, можна створювати ілюстрації, графіку та логотипи у різних стилях. Звичайно, неможливо заздалегідь визначити результат, хоча ви можете керувати створенням логотипу за допомогою ескізу.

Редагування та ретушування зображень. Ви можете використовувати SD для редагування та ретушування фотографій. За допомогою AI Editor завантажте зображення і використовуйте гумку, щоб замаскувати область, яку потрібно відредагувати. Потім, згенерувавши підказку, що визначає, чого ви хочете досягти, відредагуйте зображення або домалюйте деталі. Наприклад, можна відновити старі фотографії, видалити об'єкти зі знімків, змінити особливості зображення та додати нові елементи до нього.

Створення відео. Використовуючи такі функції, як Deforum від GitHub,

можна створювати короткі відеокліпи та анімації за допомогою Stable Diffusion. Інше застосування – додавання різних стилів у відео. Можна також анімувати фотографії, створюючи видимість руху, наприклад, течії води. Приклад однієї з можливостей зображено на рисунку 3.7.

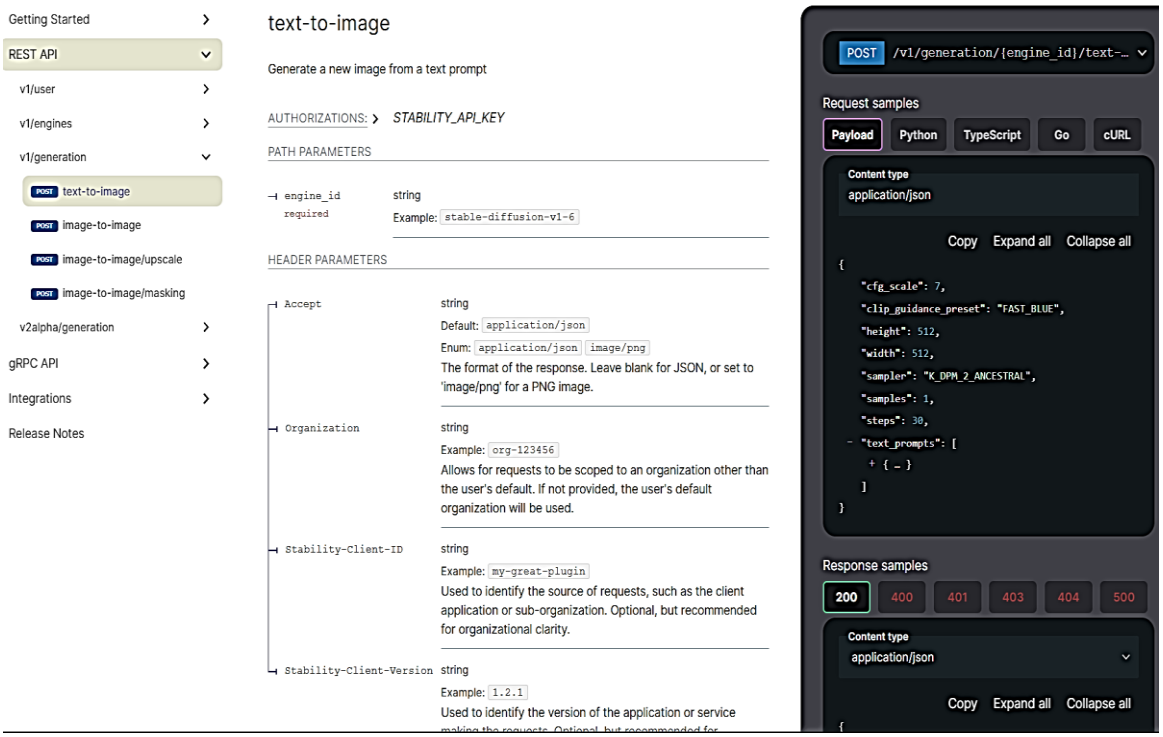


Рисунок 3.7 – Приклад ендпоінту для генерації зображеня на базі текстового вводу у API документації

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ КАСТОМІЗАТОРА ФУТБОЛОК З ВИКОРИСТАННЯМ ШІ

4.1 Реалізований функціонал кастомізатора футболок

У поточному веб додатку був реалізований такий функціонал:

- відображення 3d моделі футболки;
- обертання футболки в просторі;
- зміна кольору футболки;
- завантаження логотипу футболки;
- генерація логотипу через модель Stable diffusion;
- серверний додаток який буде відправляти запит на генерацію зображення та повертати його на клієнт.

Клієнт був зроблений за допомогою Javascipt бібліотеки React для створеня SPA додатку та бібліотеки Tree.js яка допомогає в роботі із 3d графікою. Сервер був зроблений на Node.js разом із Express.js фрейворком який буде звертатися до віддаленого API сервісу Stability.ai щоб отримати доступ до Stable diffusion. Взаємодія клієнта с сервером зображена на рисунку 4.1.

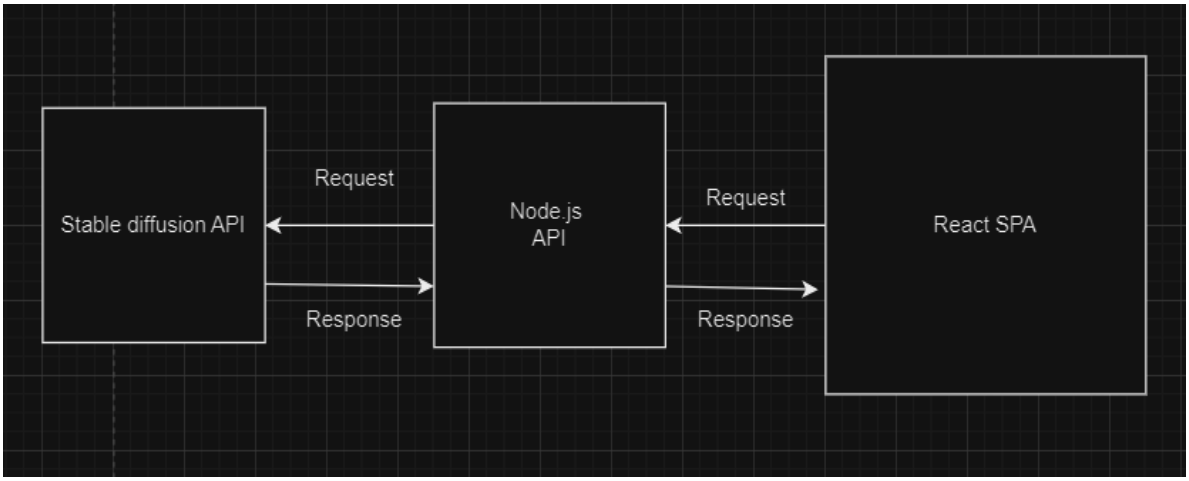


Рисунок 4.1 – Взаємодія клієнту с сервером та api сервісом

4.2 Опис екранів веб додатку та код програми

Стартовий екран веб додатку зображений на рисунку 4.2.

Рисунок 4.2 – Стартовий екран додатку

Екран кастомізатора футболок зображений на рисунку 4.3.

Рисунок 4.3 – Екран кастомізатора футболок

На екрані є три іконки, кожна відповідає за необхідний функціонал. Нижче наведений код який відповідає за зміну кольору футболки. Перша іконка переносить нас до зміни кольору, це можна побачити на рисунку 4.4.

Рисунок 4.4 – Екран зміна кольру футболки

```
const snap = useSnapshot(state);
const { nodes, materials } = useGLTF('/shirt_baked.glb');
const meshRef = useRef<Mesh<BufferGeometry<NormalBufferAttributes>, Material>>(null);
const logoTexture = useTexture(snap.logoDecal);
const fullTexture = useTexture(snap.fullDecal);

useFrame((state, delta) => {
  const lambertMaterial = materials.lambert1 as MeshLambertMaterial;
  easing.dampC(lambertMaterial.color, snap.color, 0.25, delta);
  if (meshRef.current) {
    const meshMaterial = meshRef.current.material as MeshLambertMaterial;
```

```
    // Update the color to the material
    meshMaterial.color = lambertMaterial.color;
    meshMaterial.needsUpdate = true;
    // Set aoMapIntensity to 0
    if ('aoMapIntensity' in lambertMaterial) {
      lambertMaterial.aoMapIntensity = 0;
    }
   }
  });
```

Наступна іконка для того щоб завантажити власний логотип на футболку. Вона зображена на рисунку 4.5

Рисунок 4.5 – Екран завантаження логотипу

Код для завантаження файлу

```
export const reader = (file: File) =>
  new Promise((resolve) => {
    const fileReader = new FileReader();
```

```
  fileReader.onload = () => resolve(fileReader.result);
  fileReader.readAsDataURL(file);
 });
const readFile = (type: string) => {
  if (file) {
    reader(file).then((result) => {
      handleDecals(type, result as string);
      setActiveEditorTab('');
    });
  }
};
```

Остання іконка переносить до сторінки де можна сгенерувати логотип по ключовим словам за допомогою штучного інтелекту. Приклад використання зображений на рисунках 4.6 та 4.7

Рисунок 4.6 – Екран вводу слів опису логотипа

Рисунок 4.7 – Результуюче зображення яке згенерував штучний інтелект

Після того як користувач ввів потрібний опис запит відправився на node.js сервер який вже звертається до API моделі SD.

Код реалізації наведений нижче

Запит від клієнта до серверу

```
import axios from 'axios';

export const generateImage = async (prompt: string): Promise<string | undefined> => {
  try {
    const response = await axios.post(`${import.meta.env.VITE_BACKEND_API_URL}/ai/generateImage`, {
      prompt,
    });
    return response.data.base64;
```

```
    } catch (error) {
      console.log(error);
    }
  };
```

Обробка запиту від сервера, звернення до SD API, повернення зображеня у форматі base64

```
  import fetch from "node-fetch";
  import { StableDiffusionRepository } from "./../../../core/repository/UserRepository/StableDiffusionRepository.js";
  import { GenerateImageResponseDTO } from "../../../core/repository/UserRepository/dto/GenerateImageResponseDTO.js";

  export class StableDiffusionImpl implements StableDiffusionRepository {
    async generateImage(prompt: string): Promise<{base64 : string}> {
      const formatRequest = {
        cfg_scale: 7,
        clip_guidance_preset: "FAST_BLUE",
        height: 1024,
        width: 1024,
        sampler: "K_DPM_2_ANCESTRAL",
        samples: 1,
        steps: 30,
        text_prompts: [
          {
            text: prompt,
            weight: 1,
          },
        ],
```

```
    };
    try {
      const response = await fetch("https://api.stability.ai/v1/generation/stable-diffusion-xl-1024-v1-0/text-to-image", {
        method: "POST",
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${process.env.SD_API_KEY}`,
        },
        body: JSON.stringify(formatRequest),
      });
      if (!response.ok) {
        throw new Error("Network response was not ok");
      }
      const data = await response.json() as GenerateImageResponseDTO;;
      return { base64 : data.artifacts[0].base64 };
    } catch (error) {
      console.log(error);
      throw error;
    }
  }
}
```

ВИСНОВКИ

Штучний інтелект став неодмінною частиною нашого сучасного світу і має широкий спектр застосувань у веб-технологіях. Ось деякі висновки щодо використання іскусного інтелекту в цій сфері:

- збільшення ефективності, використання штучного інтелекту в веб-технологіях дозволяє покращити користувацький досвід, автоматизувати процеси та забезпечити більш ефективну роботу веб-сервісів;
- покращення персоналізації, ШІ дозволяє створювати персоналізовані рекомендації, взаємодію та контент для користувачів веб-сайтів, що робить їхній досвід більш індивідуальним і задовольняючим;
- аналіз та обробка даних, ШІ допомагає веб-сайтам аналізувати великі обсяги даних, щоб визначати патерни, тренди та корисну інформацію для прийняття рішень;
- оптимізація веб-трафіку, за допомогою ШІ можна аналізувати та оптимізувати веб-трафік, щоб покращити швидкість завантаження сторінок та загальний веб-перформанс;
- збільшення безпеки, штучний інтелект допомагає виявляти та запобігати кіберзагрозам та атакам на веб-сайти, що забезпечує безпеку користувачів.

У результаті виконаної роботи був проведенний аналіз штучного інтелекту, як його використовують у вебі, які є моделі та для рішення яких проблем вони підходять.

Як приклад використання ШІ у веб додатках був реалізованний кастомізатор футболок з використанням Stable Diffusion, технології на яких був зробленний веб додаток це React який оптимізує швидкість роботи соціальної мережі завдяки принципу SPA та бібліотека Tree.js яка допомагає працювати із 3d графікою. Для того щоб звертатися до сторонього API був написан невеликий сервер на Node.js, увесь необхідний функціонал був визначений та виконаний а саме :

- відображення 3d моделі футболки;
- обертання футболки в просторі;
- зміна кольору футболки;
- завантаження логотипу футболки;
- генерація логотипу через модель Stable diffusion;
- серверний додаток який буде відправляти запит на генерацію зображення та повертати його на клієнт.

Загалом, штучний інтелект стає ключовим інструментом для оптимізації веб-технологій та надає можливості для створення більш інтелектуальних та інтерактивних веб-сервісів для користувачів. Використання ШІ у веб-технологіях може покращити користувацький досвід, підвищити ефективність та забезпечити більшу безпеку в інтернеті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Рассел Стюарт, Норвіг Пітер. "Штучний інтелект: сучасний підхід."

2. Гудфеллоу Іан, Бенхіо Йошуа, Курвіль Аарон. "Глибоке навчання."

3. Мерфі Кевін П. "Машинне навчання: імовірнісна перспектива."

4. Саттон Річард С., Барто Андрю Г. "Підсилений інтелект: вступ."

5. Лейн, Говард, Хавард Д., Хапке Шон. "Обробка природної мови в дії."

6. Негневіцький Майкл. "Штучний інтелект: посібник з інтелектуальних систем."

7. Бостром Нік. "Суперінтелект: шляхи, загрози, стратегії."

8. Посібник з Node.js. [Електронний ресурс] – URL: https://nodejs.org/docs/latest/api/ (дата звернення: 27.10.2023).

9. Посібник з JavaScript. [Електронний ресурс] – URL: https://metanit.com/web/javascript/ (дата звернення: 30.10.2023).

10. Сучасний підручник JavaScript. [Електронний ресурс] – URL: https://learn.javascript.ru/ (дата звернення: 30.10.2023).

11. Документація щодо використання React. [Електронний ресурс] – URL: https://reactjs.org/docs/ (дата звернення: 10.11.2023).

12. Посібник з React. [Електронний ресурс] – URL: https://metanit.com/web/react/ (дата звернення: 15.11.2023).

13. Документація щодо використання Typescript. [Електронний ресурс] – URL: https://www.typescriptlang.org/ (дата звернення: 08.12.2023). 55

14. Посібник з TypeScript. [Електронний ресурс] – URL: https://metanit.com/web/typescript/ (дата звернення: 08.12.2023).

15. Посібник з Tree.js. [Електронний ресурс] – URL: https://threejs.org// (дата звернення: 08.12.2023).