

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Аліна САВЧЕНКО
«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ»

Тема: «Веб-додаток «Еко-світ» - магазин еко-іграшок»

Виконавець: Олександра ФАТЬКОВА

Керівник: к.т.н., доцент Володимир БОРОВИК

Нормоконтролер: к.т.н., доцент Олена ТОЛСТІКОВА

КИЇВ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:
завідувач кафедри КІТ
Аліна САВЧЕНКО

(підпис)

«_____» _____ 2022 р.

ЗАВДАННЯ

на виконання дипломної роботи

Фатькової Олександри Олександрівни

(ПІБ випусника)

1. Тема роботи: «Веб-додаток «Еко-світ» - магазин еко-іграшок» затверджена наказом ректора № 1774/ст від 28.09.2022р.
2. Термін виконання роботи: з 26 вересня 2022 року по 21 листопада 2022 року.
3. Вихідні дані до роботи: створення демонстративної версії веб-додатку.
4. Зміст пояснювальної записки: 1. Огляд та аналіз предметної області, 2. Аналіз та порівняння обраних технологій для веб-додатку, 3. Опис прототипу веб—додатку магазину дитячих іграшок.
5. Перелік обов'язкового ілюстративного матеріалу: **потрібно перелічити назву основних слайдів.**

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Аналіз предметної області та огляд аналогів. Написання 1 розділу, представлення керівнику	26.09.2022- 16.10.2022	
2.	Вибір та опис використаних технологій. Написання 2 розділу, представлення керівнику	17.10.2022- 30.10.2022	
3.	Розробка додатка за допомогою обраного стеку технологій. Написання 3 розділу, представлення керівнику	31.10.2022- 14.11.2022	
4.	Загальне редагування та друк пояснювальної записки	15.11.2022- 20.11.2022	
5.	Проходження нормоконтролю, перепліт пояснювальної записки.	16.11.2022- 20.10.2022	
6.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації	20.11.2022- 22.11.2022	

7. Дата видачі завдання _____ 26.09.2022р. _____

Керівник кваліфікаційної роботи _____ Володимир БОРОВИК
(підпис керівника)

Завдання прийняв до виконання _____ Олександра ФАТЬКОВА
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Веб-додаток «Еко-світ» - магазин еко-іграшок» містить: 82 сторінок, 55 рисунки, 19 інформаційних джерел.

Об'єкт дослідження – веб-додатки та їх можливості, мова програмування JavaScript та її бібліотека React.

Предмет дослідження – веб-додаток для продажу дитячих еко іграшок.

Мета дипломної роботи – розглянути актуальність розробки веб-додатків, отримати прототип-додатку з продажу дитячих еко-іграшок.

Методи дослідження – мова програмування JavaScript, бібліотека React, інтегроване середовище розробки Visual Studio Code, платформа для розробки макету Figma.

Результати кваліфікаційної роботи рекомендується використовувати для створення власного проекту для продажу еко-іграшок в містах України

Для розробки веб-додатку було використано програму Figma для створення макету веб-додатку, програму VisualStudio Code, а також такі мови як програмування JavaScript, HTML та бібліотеку React.

WEB-ДОДАТОК, ВЕБ-ДОДАТОК, ІНТЕРНЕТ-МАГАЗИН, ІГРАШКИ, ДИТЯЧІ ІГРАШКИ, ГРИЗУНКИ, СЛІНГБУСИ, БІЗБОРД, ПРОДАЖ ІГРАШОК, РОЗРОБКА, CSS, HTML, FIGMA, JAVASCRIPT, REACT, ПРОЕКТУВАННЯ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	9
ВСТУП.....	10
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1. Аналіз предметної області застосування.....	13
1.2. Переваги та недоліки веб-додатків	15
1.2.1 Переваги створення веб-додатків	15
1.2.2 Недоліки створення веб-додатків	16
1.3.Класифікація веб-додатків	17
1.4. Архітектура та принципи роботи веб-додатку.....	21
1.5. Порівняння сайтів аналогів	29
1.5.1. Інтернет-магазин «Антошка».....	29
1.5.2. Інтернет-магазин «Будинок Іграшок».....	30
1.5.3. Інтернет-магазин «NickToys»	31
1.5.4. Інтернет-магазин «MyPlay».....	32
ВИСНОВКИ ДО РОЗДІЛУ 1	34
РОЗДІЛ 2. АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ТЕХНОЛОГІЙ ДЛ Я WEB- ДОДАТКУ	35
2.1. Розробка макету.....	35
2.2. HTML.....	37
2.3. CSS	40
2.4. JavaScript	42
2.5. Порівняння фреймворків	46
2.5.2. Angular	49
2.5.3. Vue	50
2.6. Вибір текстового редактору.....	51
2.6.1 Блокнот.....	51
2.6.2 Sublime.....	55
ВИСНОВКИ ДО РОЗДІЛУ 2.....	58

РОЗДІЛ 3. ОПИС ПРОТОТИПУ WEB-ДОДАТКУ МАГАЗИНУ ДИТЯЧИХ ІГРАШОК.....	59
3.1. Створення макету.....	59
3.2 Верстка макету	72
ВИСНОВОК ДО РОЗДІЛУ 3	80
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

HTML	–	HyperText Markup Language
CSS	–	Cascade Style Sheets
HTTP	–	HyperText Transfer Protocol
JS	–	JavaScript
URL	–	Uniform Resource Locator
W3C	–	World Wide Web Consortium
CDN	–	Content Delivery System
SPDY	–	Speedy
PDF	–	Portable Document Format
DNS	–	Domain Name System
TCP	–	Transmission Control Protocol
UI	–	User Interface
UX	–	User Experience
SQL	–	Structured Query Language
IP	–	Internet Protocol

ВСТУП

Наразі у нашій країні дуже скрутні часи, а саме – війна. Життя багатьох українців змінилось з ранку 24 лютого, коли нас атакувала країна сусід, адже багато, хто зіштовхнувся з такими проблемами, як: акупованість городів, земель, втрата своєї домівки, втрата близьких, втрата роботи, бізнесу, тощо.

Через акупованість великої кількості земель більша кількість мешканців тих районів приймали для себе рішення виїжджати з їх міст, домівок аби бути у безпеці, а не бути під постійними загрозами безпеки, арт обстрілів, прильотів ракет або бомб. В такі хвилини прийняття рішень, коли рахунок йде на години або навіть на хвилини, люди беруть з собою лише найнеобхідніше із думки про те що весь цей жах скоро закінчиться, або ж людям просто немає способу вивезення своїх речей.

Аби перемогти у цій страшній війні обов'язком кожного громадянина нашої країни повинно стати те, щоб ми допомагали один одному у скрутній ситуації. Для себе обрала – допомогти сім'ям з маленькими дітьми, у яких немає коштів купувати іграшки діткам.

Наразі, кількість користувачів Інтернету в Україні зростає, а отже і зростає попит на Інтернет послуги, а саме на онлайн спілкування, онлайн шопінг, читання новин. Цей час стає істотним для переходу більшості ланок бізнесу в Україні на онлайн платформи.

Завдяки наявності Інтернету бізнес в Україні може хоч якось вистояти період війни, та не вийти у великий «мінус» або не закрити зовсім. Адже, Інтернет магазини надають змогу власникам бізнесу зменшити витрати на оренду приміщень, збути свій товар, розширити асортимент, та завдяки цьому всьому ще й мають можливість не втратити своїх працівників та сплачувати їм заробітну платню та підтримати їх у скрутну хвилину.

Розвиток Інтернет-магазинів в Україні розпочався ще на початку 2000хх років і на це вплинуло декілька факторів, а саме:

– розповсюдження комп'ютерів по світу, та зниження цін на комп'ютерну систему загалом;

- зростання кількості онлайн покупок по всьому світу, їх об'єм виріс з 8 млрд. дол. в 1999 р. до більш ніж 100 000 000 000 в 2003 році;
- все більше компаній намагається перевести своїх робітників на роботу вдома, адже це допомагає заощаджувати;
- завдяки аналізу даних на той час було зроблено висновок, що кількість користувачів інтернету до 2020 року зросте до 4млрд.

Спочатку створювалися звичайні сайти які орієнтувалися як ринок роздрібної торгівлі, або web-портали, де систематизувалася корпоративна інформація та доступ до неї мала лише певна кількість людей.

З розвитком технологій сайти покращувалися, відомі світові компанії та бренди почали роботу з багатьма країнами світу аби краще розповсюджувати свій товар.

Наразі потрібно виводити Україну на світові ринки, аби підтримати економіку і все більше країн дізнавались хто ми, якщо Україна буде активним учасником світової економіки їй також потрібно впроваджувати новітні технології, для цього необхідно: впроваджувати новітнє комп'ютерне спорудження для міст і сільської місцевості, за для розвитку інфраструктури, розповсюдження Інтернету по всій території України, розроблення програм для навчання держслужбовців поглибленому вмінню використовувати комп'ютери та Інтернет, а також розповсюдження програм для звичайних користувачів на початковому рівні.

З усього вище сказаного однозначно зрозуміла актуальність питання дослідження Веб-додатків та використання новітніх технологій. В рамках дипломного проекту перед нами було поставлено завдання розробити Веб-додаток «Есо-світ» — магазин еко-іграшок.

Мета роботи. Проаналізувати специфіку діяльності Веб-додатків та Інтернет-магазинів взагалі, їх відмінності між собою, розглянути новітні засоби в розробці Веб-додатків, дізнатись більше про мови програмування та новітні фреймворки, розробити макет додатку та розробити Веб-додаток магазину еко-іграшок.

Завдання на кваліфікаційну роботу:

1. Провести аналіз предметної області використання веб-додатків.
2. Провести порівняння веб-додатків аналогів, виявити їх недоліки та переваги.
3. Розглянути особливості сучасних програмних засобів по розробці Web-додатків використовуючи мови: HTML, CSS, JavaScript та її бібліотеки React.
4. Розробити дизайн веб-додатку у програмі Figma.
5. Розробити Веб-додаток «Еко-світ» - магазин еко-іграшок.

Об'єктом дослідження є проектування та розробка веб-додатку з продажу еко-іграшок.

Предметом дослідження є процес розробки веб-додатків

Для досягнення поставленої мети й виконання завдань застосовано сучасні технології та засоби розробки веб-застосунків, зокрема стек HTML/CSS/JS, фреймворк React, бібліотеки ReactDOM та Router, редактор коду Microsoft Visual Studio Code, графічний редактор Figma, система контролю версій Git.

Веб-додаток буде складатись з головної сторінки, на якій буде зображатись коротко інформація стосовно магазинчику та чим він займається, логотип магазину та кнопка для відкриття меню даного веб-додатку. В меню можна буде обрати сторінку на яку саме ви хочете перейти, а саме : «О нас» - сторінка з інформацією про магазин, перелік розділів з іграшками, «Оплата» - інформація стосовно способів оплати, «Контакти» - інформація з номерами телефонів нашого магазинчику, а також посилання на сторінки в соціальних мережах.

РОЗДІЛ 1

ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз предметної області застосування

Головним завданням підприємницької діяльності є просування своєї продукції в мережі Internet і одним зі способів рішення є створення сайту або ж веб-додатку.

Веб-додаток – це комплекс веб-сторінок, що виконують необхідну кількість функцій, які потрібні користувачеві, застосовуючи веб-браузер як клієнта, при цьому клієнт може використовувати будь-який пристрій або платформу завдяки якому можна відкрити браузер. Логіка веб-додатків або як кажуть просто сайтів, зосереджена на тому що, всі зроблені маніпуляції клієнтом : збереження даних, зміна даних, часто переглядаємі розділи і тд., все це одразу передається на сервер та зберігається там, а сам обмін цих даних відбувається завдяки мережі. Головною перевагою використання таких веб-сервісів є те, що немає залежності від операційної системи, якою користується клієнт, адже веб-додатки є міжплатформенними.

Веб-додатки бувають прості, які складаються з однієї або декількох веб-сторінок, а також складні, прикладом яких є мобільні ігри, програми для редагування та створення фото, відео та ін. Прості веб-додатки можна класифікувати за призначенням, а саме (рис.1.1):

Кафедра КІТ				НАУ 22 34 80 000 ПЗ			
	ПІБ			Огляд та аналіз предметної області	Літ.	Аркуш	Аркушів
Виконала	Фатькова О.О.						21
Керівник	Боровик В.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						

- головна сторінка або домашня сторінка – це та сторінка з якої починається перегляд веб-додатку при переході за URL-адресою. На цій сторінці розкривається сутність додатку, щоб користувач зрозумів тематику веб-додатку, ознайомився з матеріалами які можна знайти та зрозумів чи зможе щось корисне знайти на сторінках ресурсу;

- сторінки-контейнери – це веб-додаток який складається з веб-сторінок на яких розміщені посилання на різні ресурси, так найпоживанішим є веб-каталог, в якому розміщуються посилання на різні веб-ресурси, каталоги файлів, які служать посиланням для завантаження необхідних даних;

- комунікативні сторінки – це інтерактивні сторінки, які розроблені для того щоб можна було отримувати зворотній зв'язок розробника веб-додатку та користувача. Зазвичай такі сторінки мають такі форми взаємодії, як : чати, вікно для написання відгуків, коментарів, різні форми анкетування.



Рис.1.1. Схема класифікації веб-сторінок

Що до створення веб-додатків то бувають також різні підходи:

- адаптивний веб-дизайн, цей вид дизайну використовується для створення односторінковий веб-додатків, ще їх називають просто лендінгами;
- прогресивний, тобто об'єднання веб-сторінки з мобільною версією;

- гібридний, завдяки такому підходу ви можете вбудувати будь-який веб-сайт всередину свого веб-додатку;

- мобільний, даний підхід застосовується на мобільних пристроях і не потребує підключення до вай-фаю або ж сотового інтернету. Веб-додатки такого виду зазвичай пишуться на мовах програмування Java та C#, та в сьогоденні дні вже є спеціальні програмні каркаси, які дозволяють писати код одразу для декількох веб-додатків використовуючи лише одну мову програмування - JavaScript.

Ще веб-додатки можна відрізнити за технологією відображення:

- статичні – такі додатки мають однаковий макет на фіксованій кількості сторінок. Такі веб-додатки не змінюються та не реагують на дії користувача, зазвичай такі веб сайти пишуться за допомогою лише однієї мови програмування HTML (англ. Hyper Text Markup Language) та CSS (англ. Cascading Style Sheets) – каскадна таблиця стилів, у простих текстових редакторах.

- динамічні – в таких веб-додатках користувач може взаємодіяти з інформацією яка розташована, вводити свої данні, зберігати та редагувати їх.

1.2. Переваги та недоліки веб-додатків

1.2.1. Переваги створення веб-додатків

Веб-додатки створюються за допомогою HTML і CSS, добре відомий серед ІТ-фахівців.

На відміну від рідних програм, веб-програми можна використовувати для на всіх пристроях. Він запрограмований на роботу в будь-якій операційній система. Він повинен підходити для iOS, Android, Windows телефонні та інші системи.

Веб-додатки не потребують ресурсів, вірно до апаратних платформ.

З підтримкою та сумісністю зі старими версіями програми проблем немає. Адже це не рідкість для користувачів, коли виходить нова версія нативної програми необхідно вирішити проблеми, пов'язані з установленними

оновленнями їх копіювальне обладнання. У випадку з браузерними додатками такої проблеми немає існує тільки одна версія, в якій можуть працювати всі користувачі і в разі виходу нового, на нього переходять усі без винятку, часом навіть не помічаючи цього.

Веб-додаток не потрібно встановлювати на пристрій.

Програми запускаються у власному веб-браузері пристрою простим способом URL-адресу.

Їх не потрібно завантажувати та встановлювати з магазину програм, наприклад, програми Google Play або Apple Store. Це означає економію грошей, оскільки пряме посилання через веб-додаток безкоштовне.

Вони також можуть відкривати веб-сайти. Це означає, що вони цього не роблять і його потрібно оновлювати, як звичайну програму.

Надійність мережевого зв'язку і високий розвиток мережевих технологій.

Розробка веб-додатків є дешевшим типом розробки додатків. Він створюється з посилання або декількох посилань між програмою та URL-адресою. Розробка власних або інтерпретованих додатків вимагає вищого рівня вартості, але шанси на успішне застосування значно вищі.

Веб-додатки дозволяють своїм користувачам бути справді мобільними. При необхідності ви можете зберегти результати роботи на сервері до яких можна отримати доступ з будь-якої точки планети, де є доступ до Інтернету.

1.2.2 Недоліки створення веб-додатків

Як згадувалося вище, одну веб-програму можна використовувати на всіх девайсах, але, звичайно, сайти мають бути запрограмовані таким чином, щоб він відображався незалежно від ОС пристрою.

Якщо це не адаптивний веб-сайт, можливо, ви не зможете його нормально запустити і відобразити на iOS, Android або Windows Phone.

Для його запуску абсолютно необхідне підключення до Інтернету. Інакше ви не зможете переглядати веб-сайти та використовувати веб-додатки,

на жаль. Наприклад, в нашій країні, немає інтернету у всіх її районах. А саме це ускладнює використання веб-додатків, наприклад, під час подорожі.

Кожне перезавантаження (або оновлення сторінки) викликає помітні затримки, спричинені необхідністю встановлення HTTP-з'єднання для обробки запиту на сервері повідомлення HTTP надсилається через мережу у відповідь. Часті перезавантаження сторінки за допомогою браузера створює переривчастий режим роботи і гальмувати її.

Крім того, існують певні обмеження доступу щодо апаратного забезпечення, функціональність пристрою, на якому він працює.

Існує велика кількість додатків, які не можуть бути відтворені, якщо взяти для прикладу Інтернет, неможливо створити складні 3D-моделі в браузері.

1.3.Класифікація веб-додатків

Мобільний додаток, як просто говорять додаток, є одним з різновидів програмного забезпечення, зазвичай запускаються на таких девайсах : телефонах, смартфонах, планшетах, ноутбуках.

Розробка веб-додатки на мобільні девайси є більш складною, адже ти стикаєшся з тим, що по-перше, мобільні чи планшети працюють від акумуляторів, по-друге, ці девайси мають менш потужні процесори.

З самого початку створення свого бізнесу, підприємці намагаються розробити сайт, адже це допомагає стати більш популярними та отримати більше клієнтів. Однак з розвитком бізнесу стає потрібним розробляти мобільний додаток, адже завантаживши його лише один раз він завжди буде в телефоні клієнту, що буде набагато зручніше, адже не треба більше шукати ваш сайт в інтернеті, будуть приходити повідомлення стосовно якихось новин, акцій, тощо, що буде нагадувати про ваше підприємство, і саме це збільшує процент того, що ваш користувач знов звернеться до ваших послуг.

Наразі найпопулярнішими видами додатків є:

- Контент-додатки – це вид веб-додатків завдяки яким можна прослуховувати музику, дивитись фільми, відео, фотографії, читати книги та завдяки деяким підписками на таких веб-додатках у користувачів в них є можливість завантажити файли.

- Бізнес-додаток – цей вид додатків є одним з найпопулярнішим зараз, адже все більше людей використовують тайм менеджмент у своєму житті, розподіляє свої справи за пріоритетністю. Складність таких веб-додатків полягає в тому що такі додатки повинні відчинятись та синхронізуватись одразу на декількох пристроях.

- Мобільні ігри – це веб додатки які займають значну частину ркнку, адже в наші часи багатьох дітей з дитинства підсаджують на ігри для того щоб заспокоїти, зайняти чимось, розвивати та взагалі завдяки іграм діти можуть вивчати якісь іноземні мови. Але ігри потрібно час від часу оновлювати для забезпечення інтересу то додатку, тож для розробників ігор завжди є робота.

- Соціальні мережі – це веб-додатки які дозволяють користувачам спілкуватись між собою, обмінюватись фотографіями, відео, файлами, аудіо повідомленнями чи відео повідомленнями. В наші дні кількість користувачів зростає з кожним днем, адже все більше країн та міст проводять інтернет і все більше людей використовують смартфони.

Можна виділити 3 типи веб-додатків:

Нативні додатки (англ. Native web-apply) – це вид додатків які розроблені одразу під конкретну платформу, їх дві – Android та IOS, так для першої платформи пишуть на мові – Java, а для другої на – objective C/Swift).

Для повноцінної роботи нативні додатки можуть використовувати всі можливості вашого пристрою – камера, список контактів, примітки, жести, відбитки пальців, розпізнавання обличчя і тд.

Привілею такого виду веб-додатків є те що вони підходять до будь-якого девайсу, мають високу швидкість та продуктивність, також є і мінуси даних додатків (рис. 1.2) :

Переваги	Недоліки
<ul style="list-style-type: none"> • Швидкість роботи • Продуктивність • Високий ступінь безпеки • Здатність працювати без Інтернету • Багата функціональність • Зручність 	<ul style="list-style-type: none"> • Довга розробка • Постійні оновлення • Дорога вартість розробки та оновлень • Довгий час розробки

Рис. 1.2. Переваги та недоліки нативних додатків

Мобільний веб-додаток – даний вид веб-додатку відрізняються тим, що він під лаштований для продуктивної та швидко роботи на мобільних пристроях. Для роботи з такими веб-додатками вам вистачить лише мати доступ до інтернету або ж можна завантажити зі спеціальних платформ і потреба в інтернеті відпаде.

Даний вид веб-додатків є кроссплатформенним, він може функціонувати на будь-якому смартфоні не дивлячись на те яка операційна система на даному пристрою.

Головним недоліком є те, що якщо додаток не завантажений на ваш девайс, то потрібен постійний доступ до інтернету, і саме це впливає на швидкість та продуктивність діяльності веб-додатку. З іншими перевагами та недоліками можна ознайомитись нижче (рис. 1.3):

Переваги	Недоліки
<ul style="list-style-type: none"> • Кроссплатформенність • Проста та дешева розробка • Невелика потреба в кількості розробників 	<ul style="list-style-type: none"> • Підключення до Інтернету для швидкої роботи • Простий, некрасивий інтерфейс • Маленький рівень безпеки • Залежність швидкості і продуктивності від подачі Інтернету • Відсутність push-повідомлень

Рис. 1.3. Переваги та недоліки мобільних веб-додатків

Гібридний веб-додаток – вид мобільного веб-додатку, який об’єднує в собі два попередніх види додатків - Native додатки та мобільні веб-додатки. Даний вид додатків може буде завантажений з офіційних маркет плейсів - Play Market чи App Store та після завантаження деякі з них можуть працювати вже можуть без з’єднання з інтернетом.

Підприємці зазвичай обирають даний вид розробки, адже це дозволяє поєднати одразу переваги, як нативних додатків так і мобільних, наприклад, завдяки нативним рисам веб-додаток можна зробити швидким та при цьому завдяки рисам мобільного веб-додатку ціна за створення даного додатку буде нижчою.

Ще одною перевагою гібридних веб-додатків є те, що можна легко усувати помилки в коді та поновлювати додаток можна постійно, адже це не потребує від засновників пере заливати їх додаток наново на плейс маркет, а достатньо лише випустити поновлення, і все що залишиться користувачеві – це просто поновити додаток і вже користуватись новими фічами.

Розробка даного виду додатку є перспективною адже розробка має на увазі й те що буде розроблено продукт одразу на дві платформи.

Та як і всі види додатків, гібридний має свої переваги та недоліки (рис.1.4):

Переваги	Недоліки
<ul style="list-style-type: none">• Віртість розробки• Швидкість розробки• Мала команда для розробки• Кросплатформність• Можливі оновлення	<ul style="list-style-type: none">• Загальмованість при роботі без інтернету• Середня швидкість роботи через риси нативних додатків• Мінімалістичний дизайн

Рис.1.4. Переваги та недоліки гібридних веб-додатків

1.4. Архітектура та принципи роботи веб-додатку

Логіка багатьох веб-додатків є просто клієнто-серверною, де браузер – це клієнт, а в ролі веб-серверу – сервер. Обмін даними відбувається мережею Інтернет.

Для того щоб працював наш веб-додаток, одночасно працюють два коди, один з них є клієнтський, це саме той код який ми одразу можемо продивитись на сторінці, просто натиснувши правою клавішею миші і обравши «подивитись код», приклад можна подивитись на рис.1.5:

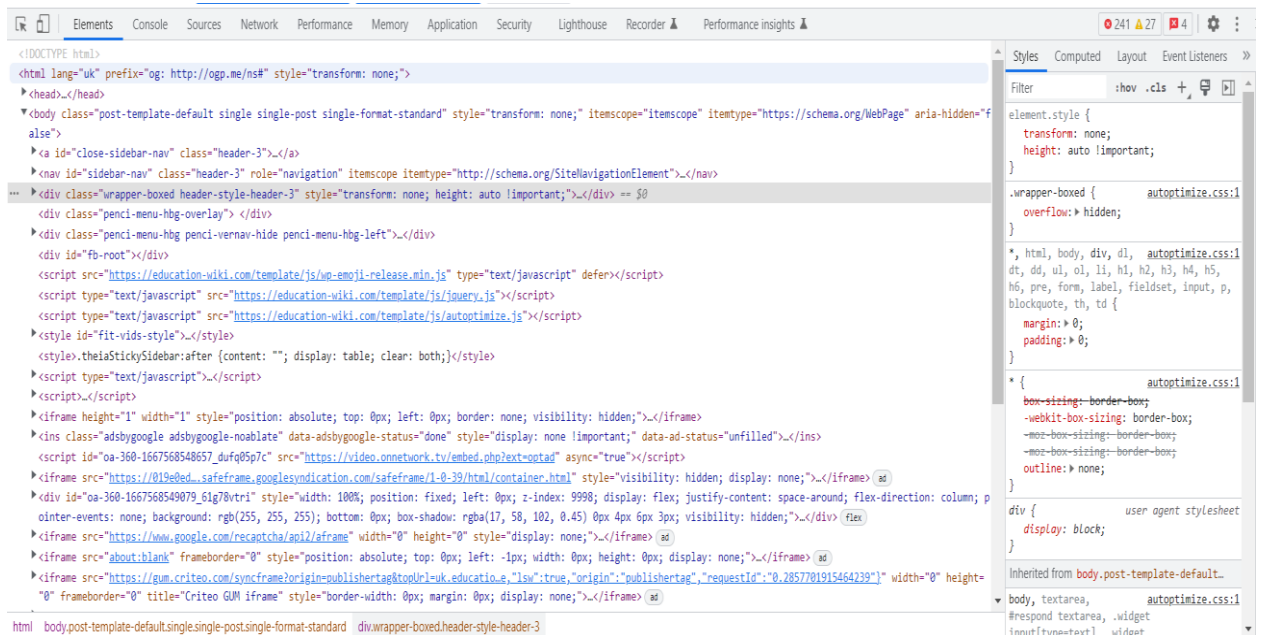


Рис.1.5. Приклад відкритого коду

Даний вид коду – клієнтський, відповідає на HTTP запити, тобто реагує на дії клієнтів: перегортання сторінок, скрол по сторінці, клацання на кнопки, ввід тексту, тощо, і одразу після реагування запит компілюється з браузером. Зазвичай такий код пишуть за допомогою таких мов програмування, як: HTML та JavaScript.

Другий код – серверний, ми його не можемо побачити, та не можемо на нього якось вплинути, він ніби «за» тим кодом, що ми бачимо тож його називають – «backend code». Такий код працює лише на запити конкретних URL і пишеться на: Python, PHP, Java, C# та інших мовах програмування.

Шлях запитів URL:

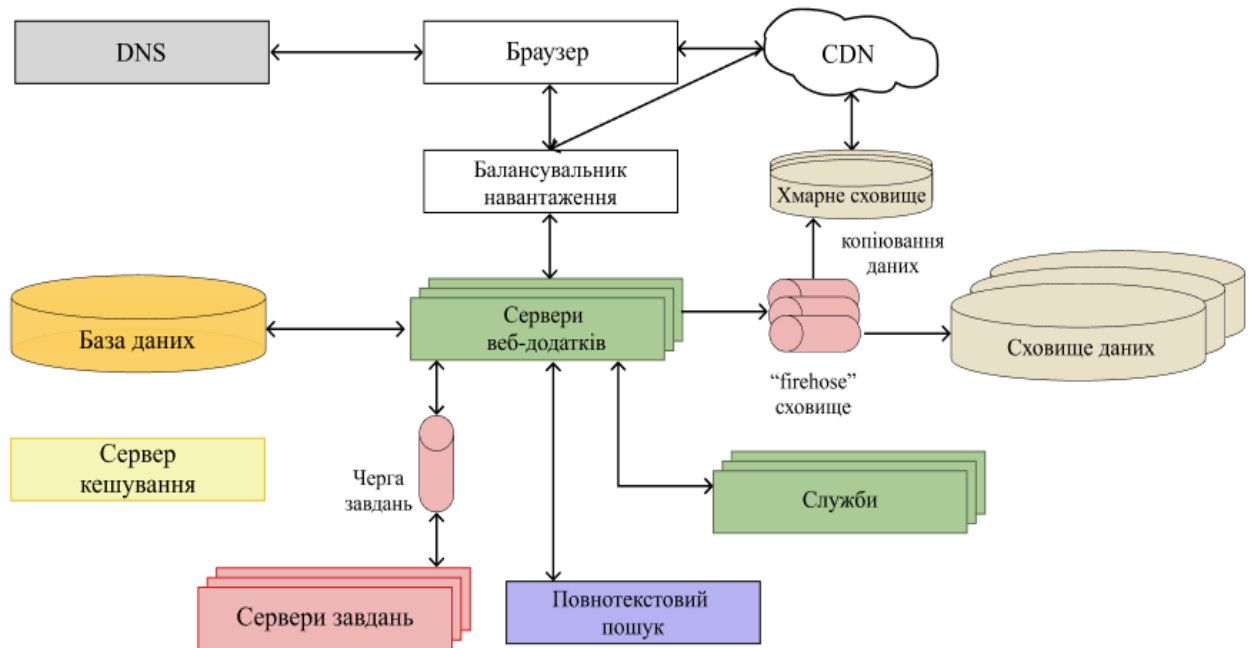


Рис.1.6. Шлях запиту користувача

1. DNS - «Domain Name System» (система доменних імен) – базова технологія, яка на рівні DNS забезпечує пошук пари з доменного імені та IP-адреси, наприклад, uk.education-wiki.com чи 87.128.81.120, це дозволяє відправити запит на сервер який відповідає за дані домені ім'я. IP -адреси – це цифри які допомагають знайти між собою сервер та девайс з якого надходив запит, а потім допомагає обмінюватись інформацією між ними. Взагалі, домені імена почали використовувати для того, аби користувачеві не потрібно було запам'ятовувати велику кількість наборів цифр, а просто би вводили маленьке ім'я, яке б асоціювалось з потрібним йому додатком. Отже, DNS це ніби книга контактів вашого телефону, але в браузері і для сайтів або додатків.

Схема роботи системи DNS (рис. 1.7.) :

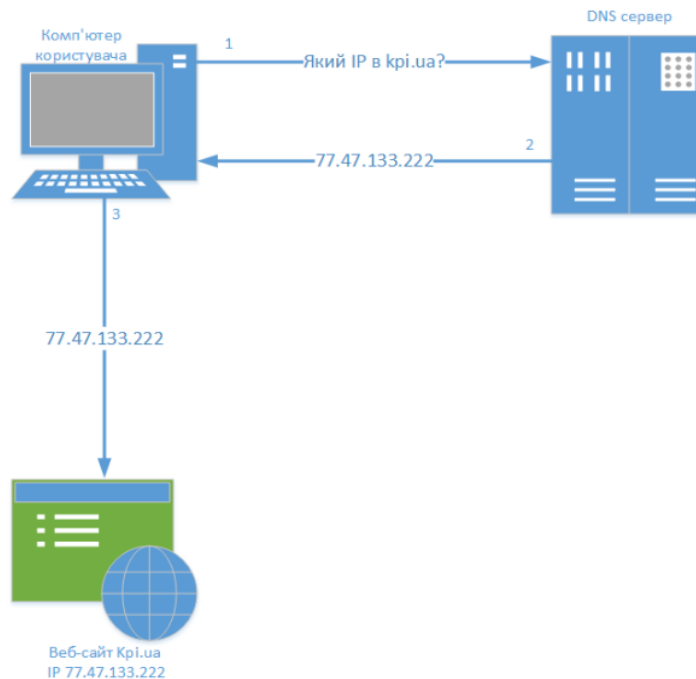


Рис. 1.7 Спрощена схема роботи системи DNS

1. Комп'ютер користувача звертається до DNS серверу аби отримати потрібний йому IP адресу.
2. Відповідь з DNS серверу повертається користувачеві.
3. Користувач вже потім звертається до потрібного сайту/додатку за отриманою IP адресою.
4. Розпочинається мережеве з'єднання (HTTP, SPDY).

TCP (Transmission Control Protocol) – це мережевий протокол, все що він робить – передає данні інтернету між клієнтом та серверами. Перевагами протоколів TCP є те що:

- незалежність від обладнання;
- не губить дані під час передачі;
- підтвердження, тобто вимагає підтвердження від сторони, яка приймає і за необхідності надсилає дані;
- присвоює номери пакетів, які передає;
- якщо передаються великі файли, то ділять файли на невеликі за розміром та передають по черзі.

HTTP (Hyper Text Transfer Protocol) – протокол передачі гіпертексту, спочатку був розроблений для передачі файли які б містили в собі посилання на інші файли, документи, але наразі використовується для передачі різноманітних файлів. Даний вид протоколу використовується у клієнто-серверних додатках, так клієнт формує свій запит: намагається відправити смс, залишити коментар, шукає щось на сторінці та ін., при цих діях запит відправляється на сервер, там він обробляється і повертається із відповіддю в зворотному напрямку - клієнту.

SPDY («speedy») – вид протоколу, який винайшла компанія Google для покращення протоколу HTTP, завдяки тому що були виправлені декілька помилок, а саме: мультиплексування даних, стиснення заголовків HTTP та встановлення пріоритетів запитів, швидкість роботи протоколів підвищилась.

2. Балансувальник навантаження – наразі дуже потрібна річ, адже через періодичні відключення світла, зв'язку в рігеонах, можливі поламки серверу, у розробці додатків почали використовувати одразу декілька серверів, аби при відключенні одного з них інший брав на себе все навантаження та роботу, аби зв'язати ці декілька серверів між собою. Як раз використовують балансувальник навантаження – це ті пристрої якими можливо зробити горизонтальне масштабування, тобто розподілити роботу поміж серверів, вони дивляться за тим щоб всі сервери працювали, а якщо десь помилка задання, вони перекидають запити на інші сервери.

Одним з найвідоміших у світі представником компаній які використовують балансувальники навантаження – це компанія Google, у світі більше мільйону серверів, які працюють постійно і при вимкненні одного його можуть легко замінити інші, саме це гарантує постійну та безперервну роботу їх додатку.

3. Сервери веб-додатків, даний вузол схеми робить всю бізнес-логіку, яка обробляє всі запити користувача і відправляє у відповідь відповідну сторінку HTML. Для оптимізування своєї роботи, вони звертаються до баз даних, серверів кешування, службам пошуку. Для написання серверів, та їх

роботи пишеться код на таких мовах програмування як: Node.js, Ruby, PHP, Scala, Java, C#, .NET, а також використовуються MVC-фреймворк (Express для Node.js, Ruby on Rails, Play для Scala, Laravel для PHP і т. д.).

4. Бази даних – вмістилище де зберігається вся інформація потрібна для роботи веб додатку. Так якщо це додаток для продажу товарів/ послуг то в базі даних буде зберігатись перелік товару, перелік потенційних клієнтів, ціна за товар, фото або відео товарів, його id. Також завдяки базам даних ми можемо доповнювати список товарів, оновлювати, щось видаляти і якщо код для сайту написаний гарно, то при зміні цих даних, сайт оновиться без потреби змінювати код додатку. Серед баз даних є ті що належать лише одному серверу, такі бази даних називають – ізольованими. Взагалі для створення бази даних використовують лише два підходи : SQL и NoSQL.

SQL (Structured Query Language) – «мова структурованих запитів». Дана мова була створена ще у 1970 роках аби розробити єдиний стандарт у світі аби зберігати набори даних. Дані у таблицях зберігаються в таблицях і для того щоб при обирання одного виду товару, підтягувалася вся інформація в тому числі з інших таблиць розробляють ключі, які і служать зв'язником між таблицями. Ключами можуть бути як числа так і буквені вирази.

Так наприклад, задано дві таблиці (рис.1.8) : «Працівники» та «Зарплата», дані таблиці пов'язані між собою завдяки зовнішньому ключу - «Табельний номер» в таблиці «Працівники» та такому ж ключу в таблиці «Зарплата».

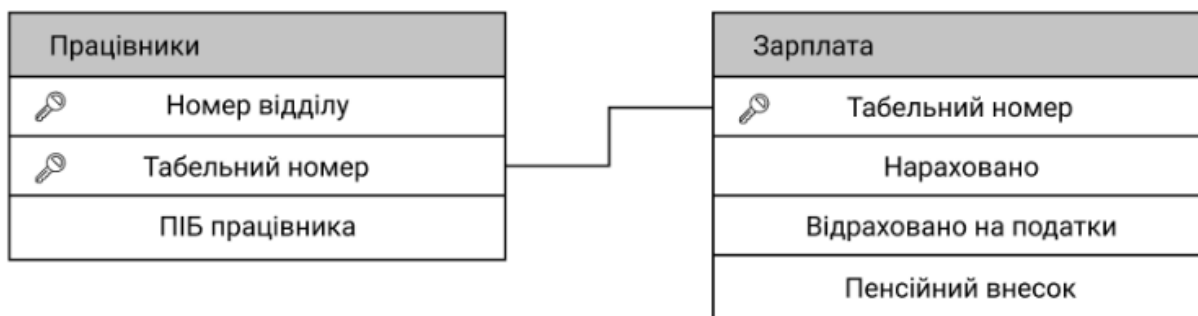


Рис 1.8. Приклад таблиць в SQL

NoSQL, в свою чергу, перекладається як «не-SQL» і є оновленою версією SQL, в якій запропоновані нові технології баз даних. Він призначений для обробки величезної кількості інформації, яку можуть генерувати великі веб-додатки. Перевага NoSQL полягає в тому, що з ним ви можете масштабувати свою базу даних горизонтально та вертикально, тоді як у SQL є проблеми.

5. Кеш-сервер - це звичайне сховище даних, у форматі ключ-значення, що дозволяє практично не витратити час на пошук необхідної інформації. Як правило, програми використовують цю функцію, щоб заощадити час, кешуючи часто використовувані запити в базі даних. Наприклад, програма може кешувати результати запитів до бази даних, HTML для заданої URL-адреси тощо. Прикладами з життя можуть бути:

- Instagram кешує більшість ваших даних, тобто ваших підписників і підписників, ваші дописи або дописи ваших друзів, «прямі» повідомлення, дані на вашій домашній сторінці.

- Google кешує результати пошуку для більш популярних запитів, наприклад, не шукає «самка» щоразу замість «сезон розмноження ангорських кроликів».

6. Велика кількість веб-додатків повинні асинхронно виконувати певну роботу у фоновому режимі незалежно від вхідного запиту користувача. Наприклад, більшість пошукових систем сканують Інтернет асинхронно, оновлюючи дані пошуку «на ходу».

Існує кілька типів архітектур, які дозволяють виконувати асинхронну роботу, але найбільш популярним і використовуваним методом є «черга завдань». Він включає: чергу завдань, які потрібно виконати, і один або кілька серверів, які оброблятимуть завдання в черзі.

У черзі завдань зберігається список завдань, які потрібно виконати асинхронно. Щоразу, коли програмі потрібно виконати певне завдання, її потрібно виконати після певної дії користувача або просто за розкладом, вона додається до списку завдань. Найкращий і найпростіший спосіб організувати

черги – «першим прийшов, першим вийшов», але більшість веб-програм потребують балансування черги. Таким чином, вони повинні працювати в пріоритетній черзі, щоб завершити операцію з фіксованим часом якомога швидше. Так, наприклад, сервер працює і виконує завдання в черзі, коли надходять завдання з більш високим пріоритетом і важливістю, вони спочатку виконують їх, а потім продовжують роботу в звичайній черзі.

7. Повнотекстовий пошук — це техніка, яка використовується майже всіма веб-додатками, які підтримують функцію текстового пошуку, і програма повертає більш відповідні результати.

Повнотекстовий пошук іноді використовується в деяких типах баз даних, таких як підтримка MySQL, але зазвичай вони запускають окрему службу, яка обчислює та зберігає інвертовані індекси та забезпечує інтерфейс запитів. Зараз найпопулярнішими платформами повнотекстового пошуку є Elasticsearch, Sphinx або Apache Solr.

8. Деякі додатки, коли їх масштабують, представляють деякі служби, створені спеціально для роботи в представленнях індивідуальна заявка. Вони невидимі, але програми та інші служби взаємодіють з ними. Так, наприклад, у додатку Storyblocks є кілька сервісів операцій і планування:

- надавати послуги інтерфейсу оплати кредитними картками;
- служби вмісту, які зберігають метадані для відео, аудіо та зображень, а також надають інтерфейс для зберігання та перегляду історії зберігання;
- сервіс, який зберігає дані користувача для всіх веб-додатків компанії, що забезпечує крос-функціональність і створює більш уніфікований досвід користувача;
- служба, яка надає інтерфейс для перетворення документа з кодом HTML на документ із розширенням PDF.

9. Довголіття компанії на ринку часто залежить від того, як вона обробляє дані. Усі великомасштабні програми, які досягають певного масштабу, звертаються до способу організації збору, зберігання та аналізу даних. Як правило, робота проходить у три етапи:

1. Як правило, програми надсилають дані про дії користувача до сховища «пожежного шланга», яке забезпечує потоковий інтерфейс для прийому та обробки даних. Часто необроблені дані трансформуються або доповнюються та переміщуються до інших «пожежних» сховищ.

2. Початкові та кінцеві дані зберігаються в хмарному сховищі. Так, наприклад, є відомий сервіс Firehose, який дозволяє зберігати необроблені дані в хмарному сховищі - «S3», який дуже простий у використанні і зрозумілий.

3. Оброблені, змінені та завершені дані завантажуються в сховище даних для аналізу. Початківцям у розробці веб-додатків рекомендується AWS Redshift, а більші компанії вже використовують рішення Oracle або інші альтернативні технології. Якщо набір даних занадто великий, для аналізу може знадобитися технологія NoSQL MapReduce, наприклад Hadoop.

10. Хмарне сховище — це простий і масштабований спосіб зберігання й обміну даними через Інтернет. Його можна використовувати для зберігання та доступу до всього, що може зберігатися в локальній файловій системі, за допомогою функцій RESTful API через HTTP.

11. CDN (Content Delivery System) - перекладено на систему доставки контенту. Саме ця технологія дозволяє швидше надсилати статичні файли з розширеннями HTML, CSS і JavaScript, а також зображення. Він поширює інформацію з багатьох серверів терміналів по всьому світу, щоб користувачі не завантажували різні ресурси з них замість вихідного сервера. Так, наприклад, якщо основний сервер знаходиться в США, а ви перебуваєте в Україні, то для того, щоб надати достовірну інформацію про ваш запит, за допомогою програми ваш комп'ютер зв'яжеться з сервером, який знаходиться ближче до вас і він вже підключатися до головного сервера та отримувати інформацію з головного сервера за вашим запитом.

12. Amazon Kinesis Data Firehose або сховище пожежних шлангів — це найпростіший і найнадійніший спосіб завантажувати потоки даних в озеро даних, сховище та сервісну аналітику. Ця служба дозволяє змінювати, зберігати та надсилати поточкові дані на Amazon S3, Amazon Redshift, Amazon

Elasticsearch Service, а також на загальні HTTP-адреси та постачальників таких послуг, як-от Datadog, New Relic. Сервіс повністю масштабується залежно від пропускної здатності потоку даних і не потребує постійного керування. Amazon Kinesis Data Firehose дозволяє створювати пакети, а також стискати, змінювати та шифрувати потоки даних перед їх збереженням, зменшуючи обсяг використовуваної пам'яті та забезпечуючи певний рівень безпеки.

1.5. Порівняння сайтів аналогів

1.5.1. Інтернет-магазин «Антошка»

Даний веб-додаток є першим з запропонованих при введенні у пошукову строку Google «магазин дитячих іграшок» (рис.1.9.). Магазин «Антошка» є одним з найвідоміших мережових магазинів іграшок. У додатку є багато переваг, наприклад:

- строка пошуку, для того щоб користувачу було легко знаходити необхідний товар;
- наявність бокового каталогу який допомагає користувачеві швидко обрати саме той розділ який йому потрібен;
- здатність перемикає сторінку на ту мову яка зручна користувачеві;
- на головній сторінці одразу видно акції які діють в магазині.

Та в веб-додатку також одразу кидається недолік – це неприємно червоний колір додатку, він не розполагає до того щоб довго сидіти на сайті та обирати потрібний тобі товар.

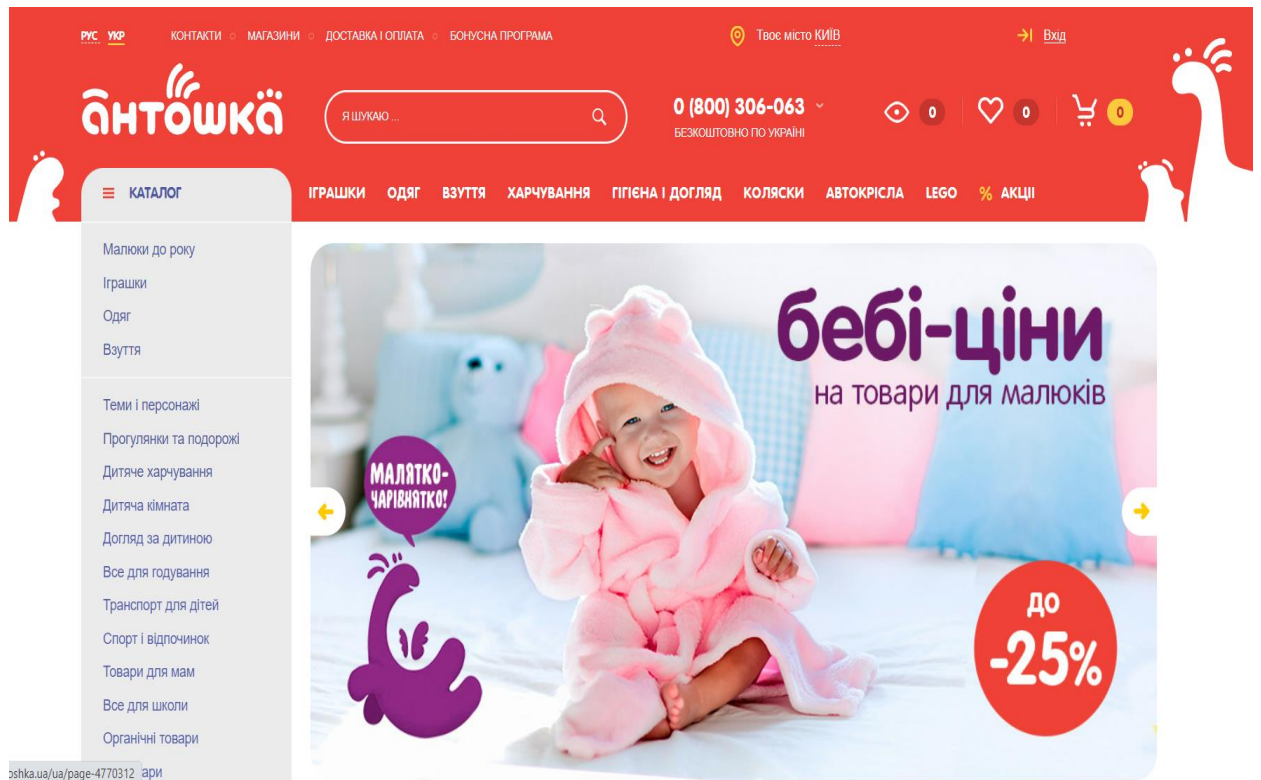


Рис. 1.9. Інтернет-магазин «Антошка»

1.5.2. Інтернет-магазин «Будинок Іграшок»

Веб –додаток магазину «Будинок іграшок» є другим у пошуковій системі Google (рис.1.10). В даного веб-сайти є безліч переваг:

- зрозуміла назва, по назві користувач одразу може зрозуміти з чим саме пов'язаний веб-сайт;
- наявність пошукової строки, одразу можна знайти що потрібно;
- одразу прописан контактний номер телефону, для зв'язку з гарячey лінією, якщо у користувача виникають якісь запитання;
- є фільтри які допомагають обрати товар на зручну вам ціну.

Та є й свої недоліки:

- не зручні фото товару, адже фон товару та головної сторінки зливаються і якомсь ріже око;
- застарілий дизайн, адже використовуються неактуальні кольори, все дуже незграбно.

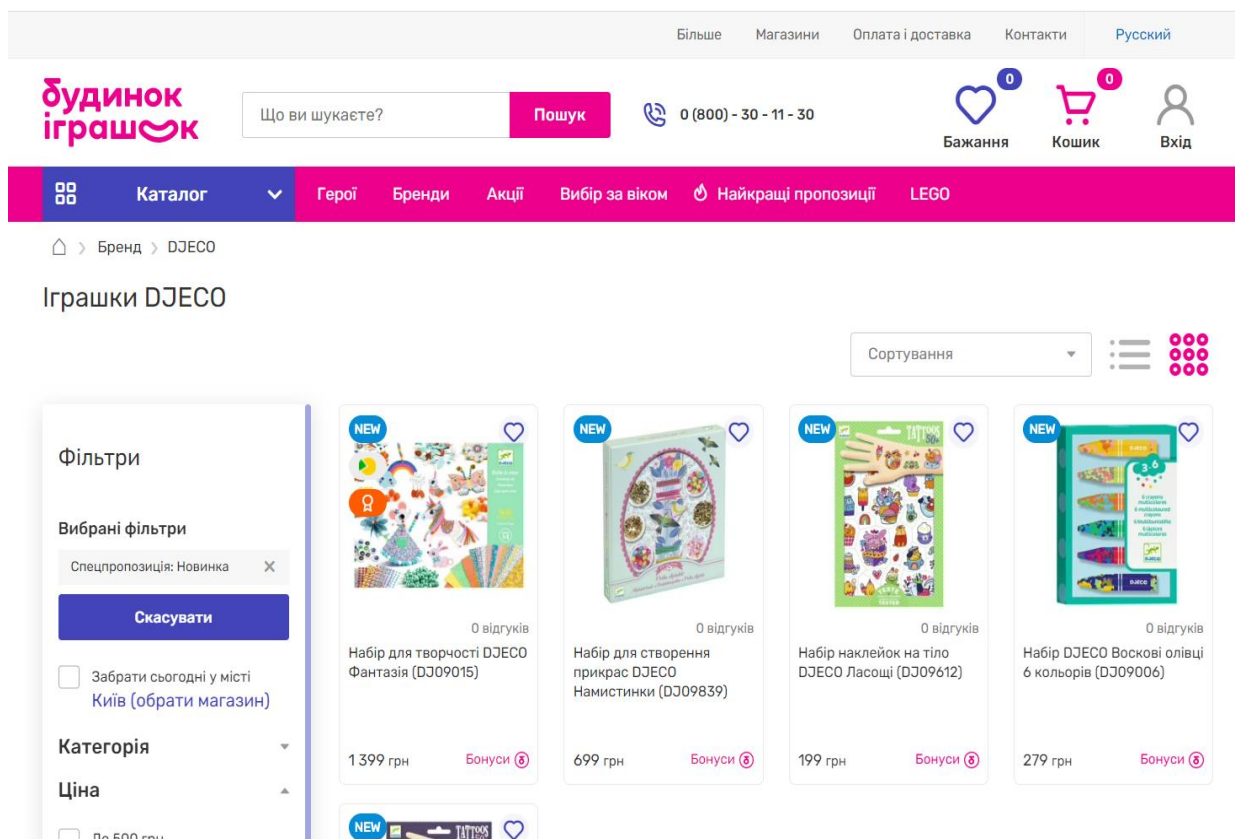


Рис. 1.10. Інтернет-магазин «Будинок іграшок»

1.5.3. Інтернет-магазин «NickToys»

Реклама веб-додатку магазину «NickToys» впливає одразу при пошуку «магазин іграшок Україна» (рис.1.11). У сайті є як переваги так і недоліки, до переваг відносяться:

- зручно дивитись фото товару, адже можна наводячи на фото товару він одображається одразу з декількох сторін, тобто не потрібно переходити окремо на сторінку товару;
- одразу є впливаюче вікно, де можна задати свої питання оператору магазину або написати свій мобільний номер для того щоб вам передзвонили;
- зверху веб-додатку є закріплена панель з віном для пошуку товару, каталогом, номером для звернень, а також одразу відображається корзина та кількість товару, яку ви вже обрали на сайті.

До недоліків можна віднести, те що сайт не є стильним та сучасним, інформація не поділена, тобто на стільки погано підібрані кольори для фону веб-додатку та фону товару або іншої інформації, що вона ніби спливається в очах.

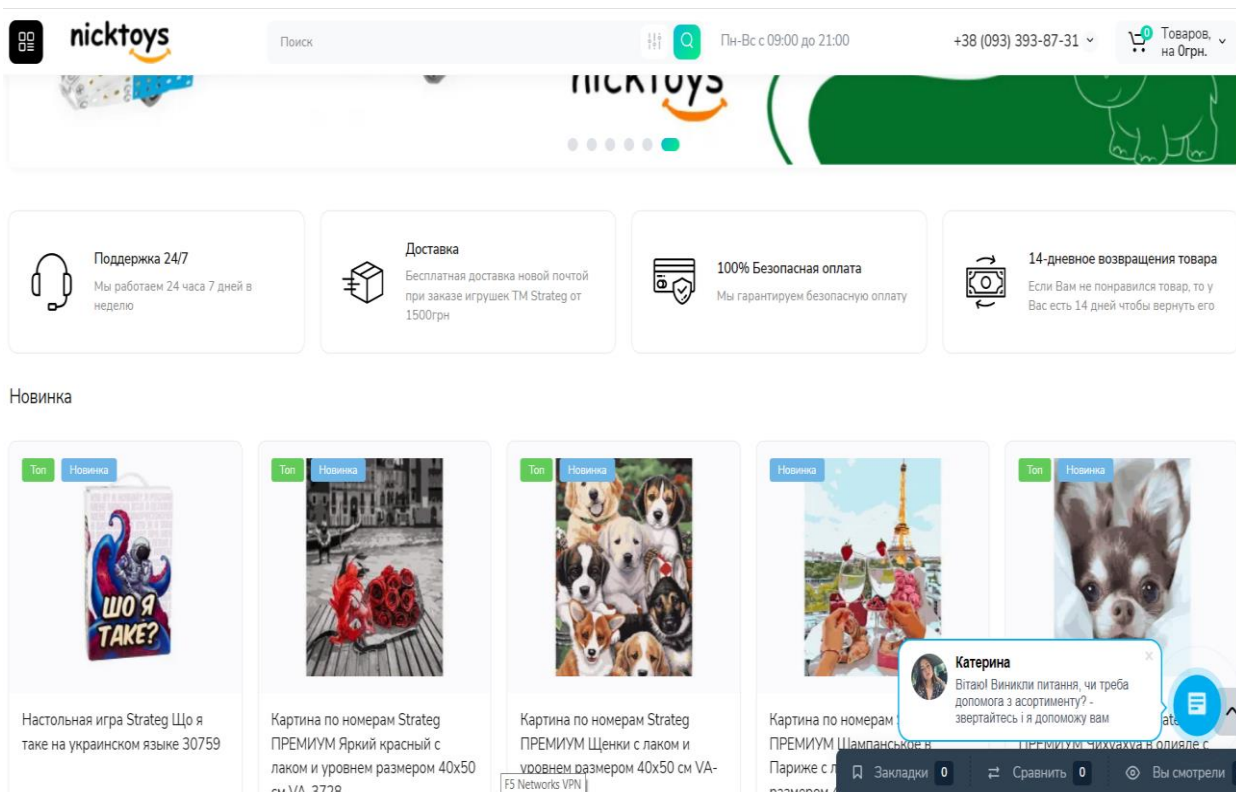


Рис. 1.11. Інтернет-магазин «NickToys»

1.5.4. Інтернет-магазин «MyPlay»

Ще одним з популярних веб-додатків в категорії іграшки є «MyPlay»(рис.1.12.). Даний сайт, як на мене, є прикладом того, яким сайт бути не повинен, так, він є кольоровим, але є багато візуальних недоліків. Наприклад:

- сайт є занадто кольоровим і це дуже відволікає при перегляді сайту, адже дуже складно зосередити свою увагу на чомусь одному;
- у веб-додатку дуже багато реклами, яку потрібно гортати, чи закривати;

- перегортати перелік товару не дуже зручно, адже при бажанні натиснути на кнопку, то одразу напливає інформація стосовно товару який розташований вище над кнопкою;
- для замовлення товару треба авторизуватися в додатку й пройти великий путь для авторизації.

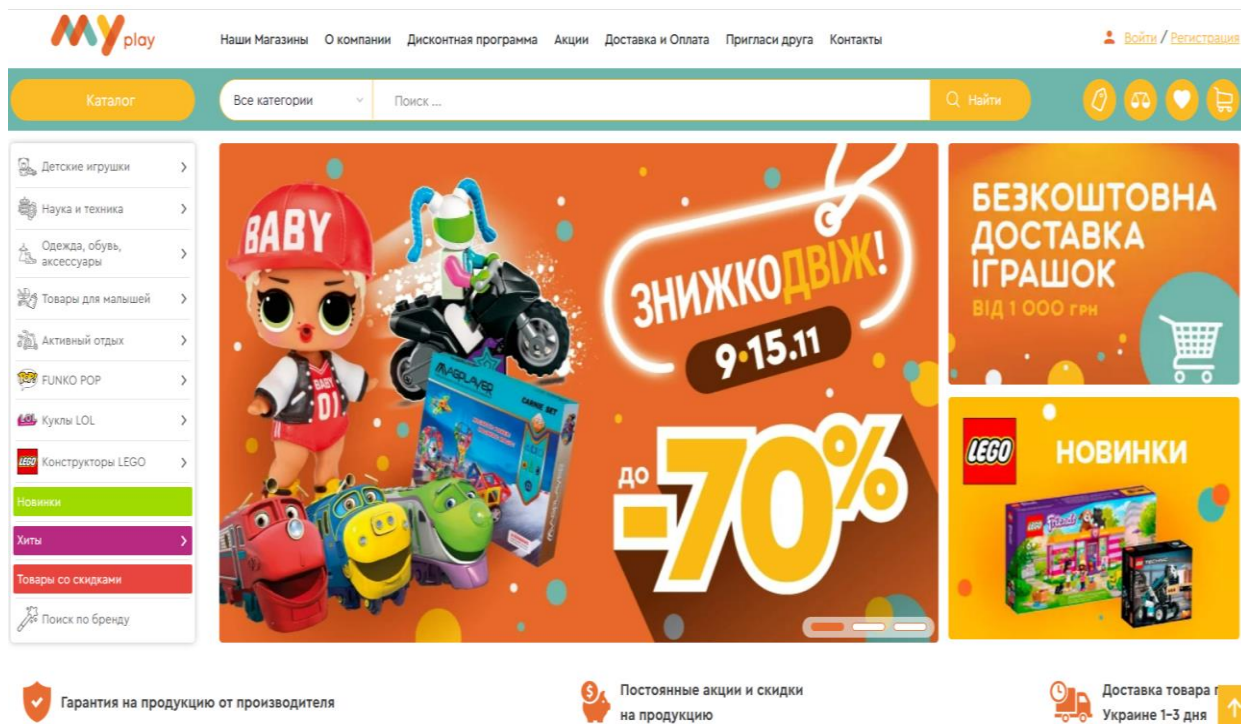


Рис. 1.12. Интернет-магазин «MyPlay»

ВИСНОВКИ ДО РОЗДІЛУ 1

В першому розділі демонструються концепції загальної веб-програми та принципи її роботи. Ми бачимо, що вони відрізняються за складністю, але, незважаючи на це, усі вони дуже прості у впровадженні та використанні, тому наразі веб-додатки, особливо для продажу чогось, є гарним рішенням для покращення економіки будь-якого бізнесу.

Був проведений одноранговий аналіз додатків, щоб зробити висновки щодо створення веб-додатку для продажу іграшок. Наприклад, додаток має бути простим для розуміння клієнтами, на сторінці має бути невелика кількість товарів, продукти мають бути посередині екрана для зручного перегляду, в додатку має бути невелика кількість реклами, адже вона відволікає від перегляду, поивнні використатись тематичні та сучасні кольори, але й не повинно бути забагато кольорів, щоб веб-додаток виглядав якось наляписто, а додаток має бути створено відповідно до нових тенденцій.

З веб-додатків, наведених у цьому розділі, можна виділити веб-додаток магазину «Антошка» та «NickToys», в обох є свої недоліки, однак є переваги які хочеться виділити. Нариклад, в веб-додатку магазину «Антошка» майже немає реклами, використовується лише пара основних кольорів, що приємно для ока, а в додатку «NickToys» є одразу боковий каталог, де дуже зручно обрати потрібний розділ, товар при перегляданні можна побачити з різних сторні, що допомагає краще зрозуміти що саме ти будеш купувати.

РОЗДІЛ 2

АНАЛІЗ ТА ПОРІВНЯННЯ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ WEB-ДОДАТКУ

2.1. Розробка макету

Першим кроком у створенні веб-додатку чи веб-сайту є створення макета. Щоб зробити це правильно та відповідати сучасним технологіям, потрібно розуміти дизайн, його види: UI дизайн та UX дизайн.

UX (User Experience) перекладається як «Взаємодія з користувачем». Цей тип дизайну відповідає за результат, поведінку, функціональність, доступність програми, а також почуття та емоції, які вона викличе у користувача. Чим простіший і приємніший інтерфейс, тим швидше та легше клієнтам буде досягати своїх цілей у вашій програмі.

Користувацький досвід — це всі контакти та взаємодії користувача з продуктом або послугою компанії. Ми можемо зробити висновок, що UX стосується:

- допомогти користувачеві вирішити його проблеми;
- логіка роботи програми або сайту;
- приємний естетичний і чіткий зовнішній вигляд;
- задоволення від роботи з інтерфейсом.

У цьому дизайні головне знати своїх клієнтів, їхні вподобання, розуміти, чого вони хочуть, вивчати звички, створювати прототипи, відкривати нові шляхи для користувачів і тестувати їх.

Кафедра КІТ				НАУ 22 10 20 000 ПЗ			
	ПІБ			Аналіз та порівняння обраних технологій для web-додатку	Літ.	Аркуш	Аркушів
Виконала	Фатькова О.О.						20
Керівник	Боровик В.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						

UI (User Interface англійською мовою), що перекладається як «Інтерфейс користувача», працює не тільки з графічним інтерфейсом, а й з дотиком, голосом або звуком.

Дизайн інтерфейсу користувача включає робочі етапи візуальної та графічної частин: кнопок, меню, анімації, значків, ілюстрацій, шрифтів тощо.

Дизайнери інтерфейсу користувача відповідають за те, як виглядає інтерфейс продукту та як клієнти взаємодіють з його елементами. Для цього необхідно ефективно організувати елементи інтерфейсу та підтримувати єдиний стиль і логіку їх взаємодії.

Вимоги до якісного дизайну UX/UI:

- така ж простота використання в усьому додатку;
- бажання бути таким же, але хочеться мати свої фішки;
- прогнозування інформаційних відповідей на відповіді;
- розробити закритий процес вирішення;
- надає просту можливість скасувати операцію;
- мінімізуйте вплив на короткочасну пам'ять.

Для того, щоб створити макет, недостатньо знати основи дизайну, потрібен також сервіс або додаток для його розробки. В даний час дуже популярним і відомим онлайн-сервісом є Figma. Головною перевагою цієї програми є співпраця в реальному часі.

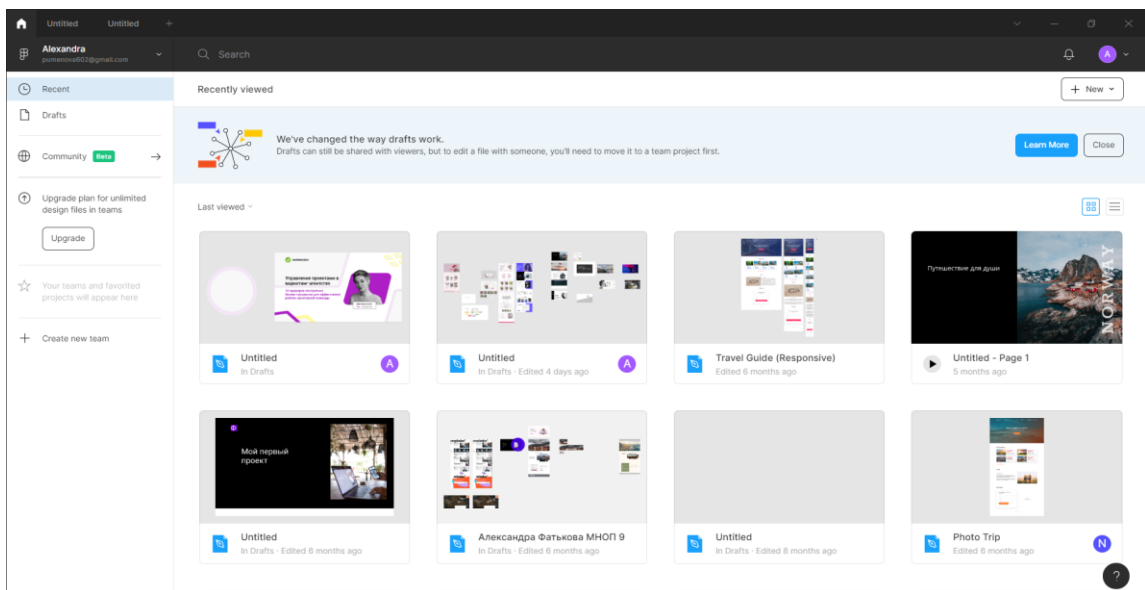


Рис 2.1. Інтерфейс сервісу Figma

Ця послуга доступна за передплатою, але є й безкоштовні плани. Figma має офлайн-версію для Windows, macOS, Linux. Її можна використовувати як для створення спрощених прототипів інтерфейсів, так і для детальної розробки дизайну інтерфейсу, веб-додатку чи сайту.

На початку створення сервісу це не давало надії, але з часом розробникам вдалося привернути до своїх проєктів увагу користувачів з усього світу, і на даний момент капіталізація компанії становить понад 2 мільярди доларів.

2.2. HTML

HTML означає мову розмітки гіпертексту. Це стандартна мова розмітки, яка використовується для створення веб-сторінок. Він дозволяє створювати та будувати розділи, абзаци та посилання за допомогою елементів HTML (будівельних блоків веб-сторінок), таких як теги та атрибути.

HTML має багато застосувань, а саме:

- веб-розробка. Розробники використовують HTML-код для проєктування того, як браузері відображають елементи веб-сторінки, такі як текст, гіперпосилання та мультимедійні файли;
- інтернет-навігація. Оскільки HTML широко використовується для вставки гіперпосилань, користувачі можуть легко переходити та вставляти посилання між пов'язаними сторінками та веб-сайтами;
- веб-документація. HTML дозволяє впорядковувати та формувати документи подібно до Microsoft Word.

Також варто зазначити, що HTML не вважається мовою програмування, оскільки він не може створювати динамічні функції. Тепер він вважається офіційним веб-стандартом. Консорціум Всесвітньої павутини (W3C) підтримує та розробляє специфікацію HTML і надає регулярні оновлення.

Історія розробки HTML почалась завдяки Тіму Бернерсу-Лі в 1990 році, який того ж року розробив браузер і серверне програмне забезпечення для

запропонованої системи, але його компанія CERN не відразу підтримала його проект.

Наприкінці 1991 року Тім Бернерс-Лі опублікував перший опис HTML-тегів в Інтернеті під назвою «HTML-теги», в якому описано 20 елементів оригінального HTML-тегу на додаток до тегу гіперпосилання, оскільки тег належить до документів CERN. , лише 13 з них використовуються.

Протягом наступних двох десятиліть Консорціум Всесвітньої павутини (W3C) продовжуватиме розвиватися до міжнародних стандартів кодування. У 1995 році була випущена друга версія HTML, за якою протягом наступних років було кілька оновлень.

Подальший розвиток під егідою IETF стикається з конкуруючими інтересами. Специфікація HTML була схвалена Консорціумом W3C з 1996 року та враховує доповнення розмітки, впроваджені компаніями-браузерами. Однак у 2000 році HTML став міжнародним стандартом (ISO/IEC 15445:2000).

Остання специфікація HTML, опублікована W3C наприкінці 1999 року, називається "HTML 4.01 Recommendation". Усі суперечливі питання та помилки в цій специфікації офіційно визнані в Списку друкарських помилок 2001 року.

Загалом, HTML дозволяє визначати різні типи елементів, які забезпечують функціональність документа: текстові фрагменти, списки, таблиці, зображення, гіперпосилання тощо із заданими параметрами форматування. Елементи HTML оголошуються за допомогою команди розмітки, яка називається лейбл (від англ. label - мітка). Під час відображення документа браузер інтерпретує всі теги HTML у тексті документа.

HTML — це мова розмітки для фраз, які описують, як програмісти використовують код для розмітки тексту. Такі мови розмітки, як HTML і XML, читаються людиною, на відміну від машинної мови, яка записана в шістнадцятковому або двійковому коді.

Використовуючи HTML для створення сторінок на веб-сайті, автори використовують кілька ключових компонентів або властивостей, які можуть

бути прочитані будь-яким Інтернет-браузером, який відвідує веб-сайт. Багато з цих атрибутів будуть поєднані з «початковим тегом» і «кінцевим тегом» (рис.2.2).

Більш уживаними тегами є:

- `<html> </html>` - вказує програму перегляду сторінок, що це HTML документ;
- `<head></head>` - Визначає розташування, яке містить різну інформацію, але не відображається в тілі документа. Тут ви розміщуєте теги заголовка документа та теги пошукової системи;
- `<body></body>` - Визначте видиму частину документа;
- `<div></div>` - Поділяє код на смислові блоки;
- `<title></title>` - Визначає назву сторінки;
- `< textarea ></ textarea >` - Визначає багаторядковий елемент керування введенням (область тексту);
- `` - Допомогає виділити слово або фразу серед всього тексту та надати деякі особливості;
- `
</br>` - Робить шрифт тексту жирним.

```
<div class="block1">Lorem ipsum dolor sit amet, consectetur  
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet  
dolore magna aliquam erat volutpat.</div>  
<div class="block2">Ut wisis enim ad minim veniam, quis nostrud  
exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex  
ea commodo consequat.</div>
```

Рис.2.2. Приклад використання тегів

Наразі всі версії мови гіпертексту використовують HTML 5, стандарт мови, який зараз використовується більшістю веб-сайтів в Інтернеті. Приклад коду з використанням HTML 5:

```
123.html x index.html — Рассылка макеты\41 mail) Phot
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Квас</title>
6   </head>
7   <body>
8     <div class="page-header">
9       <div class="header-top">
10        <div class="logo">
11          
12        </div>
13        <ul>
14          <li><a href="#">История</a> </li>
15          <li><a href="#">Производство</a> </li>
16          <li><a href="#">Ассортимент</a> </li>
17          <li><a href="#">Где купить</a> </li>
18        </ul>
19      </div>
20      <br>
21      <p>Крафтовый квас<br>
22      <p>всему голова!</p>
23    </div>
24  </body>
25 </html>
```

Рис.2.3. Приклад коду HTML

2.3. CSS

Каскадні таблиці стилів (CSS) — це мова таблиць стилів, яка використовується для опису подання документів, написаних мовами розмітки, такими як HTML або XML, включаючи діалекти SVG, MathML або XML, такі як XHTML. CSS, як HTML і JavaScript, є наріжною технологією Всесвітньої павутини.

CSS спрямований на досягнення розділення вмісту та презентації, включаючи макет, кольори та шрифти. Це розділення може покращити доступність вмісту; забезпечити більшу гнучкість і контроль у специфікації функцій презентації; дозволити використовувати форматування кількох веб-сторінками шляхом визначення відповідного CSS в окремих файлах .css, тим самим зменшуючи складність і дублювання структурованих вміст і ввімкнення кешування файлів .css для покращення швидкості завантаження сторінок між сторінками, які спільно використовують файл і його формат.

Розділення формату та вмісту також дає змогу відтворювати ту саму сторінку розмітки в різних стилях для різних методів відтворення, таких як екран, друк, мовлення (через мовний браузер або програму зчитування з

екрана) та шрифт Брайля. Тактильні пристрої. CSS також має правила для альтернативного форматування, якщо доступ до вмісту здійснюється на мобільному пристрої.

Каскад імен походить із зазначеної схеми пріоритету, щоб визначити, яке правило стилю застосовувати, якщо певному елементу відповідає більше ніж одне правило. Ця каскадна модель пріоритетів передбачувана.

Специфікація CSS підтримується Консорціумом Всесвітньої павутини (W3C). Тип Інтернет-медіа (тип MIME) `text/css` зареєстровано для використання з CSS відповідно до RFC 2318 (березень 1998). W3C надає безкоштовну службу перевірки CSS для документів CSS.

Крім HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, простий XML, SVG і XUL. Один файл CSS можна приєднати до кількох сторінок одночасно, що полегшує життя розробникам, оскільки той самий код не потрібно писати кілька разів. Приклад коду написаний на CSS:

```
body {
  margin: 0px;
  padding: 0px;
  background: #16a085;
  font-family: 'Raleway', sans-serif;
  font-size: 11pt;
  font-weight: 400;
  color: #363636;
}

h1, h2, h3 {
  margin: 0;
  padding: 0;
}

p, ol, ul {
  margin-top: 0px;
}

p {
  line-height: 190%;
}
```

Рис. 2.3. Приклад коду CSS

Переваги CSS:

1. Структура. Сторінки стають більш структурованими, оскільки інформація про стиль відокремлена від основного коду.
2. Швидкість дії. Оскільки стилі можна використовувати на кількох сторінках одночасно, браузері можуть кешувати цю інформацію замість того, щоб завантажувати її для всіх сторінок кожного разу.
3. Сумісність. Вміст файлів CSS можна оптимізувати для кількох типів пристроїв.
4. Легко обслуговувати. Щоб змінити стилі відразу на кількох сторінках для деяких вибраних елементів, потрібно встановити глобальну зміну і написати для неї потрібні стилі.
5. Швидкість. Ви можете написати код CSS один раз і використовувати його на кількох HTML-сторінках.

2.4. JavaScript

JavaScript — це динамічна мова комп'ютерного програмування. Він легкий і найчастіше використовується як частина веб-сторінок, а його реалізація дозволяє сценаріям на стороні клієнта взаємодіяти з користувачами та створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно-орієнтованими функціями.

Спочатку JavaScript називався LiveScript, але Netscape змінив назву на JavaScript, ймовірно, через захоплення Java. JavaScript вперше з'явився в Netscape 2.0 у 1995 році як LiveScript. Common Language Core вбудовано в Netscape, Internet Explorer та інші веб-браузери.

Специфікація ECMA-262 визначає стандартну версію основної мови JavaScript:

- JavaScript — це полегшена інтерпретована мова програмування;
- спеціально розроблений для створення веб-додатків;

- доповнює та інтегрується з Java;
- доповнюйте та інтегруйте з HTML;
- відкритий і кросплатформний;
- JavaScript на стороні клієнта.

JavaScript на стороні клієнта є найпоширенішою формою мови. Сценарій має бути включений у HTML-документ або посилання на нього, щоб браузер міг інтерпретувати код.

Це означає, що веб-сторінки не обов'язково мають бути статичними HTML, але можуть містити програми, які взаємодіють з користувачем, керують браузером і динамічно генерують вміст HTML.

Двигуни JavaScript на стороні клієнта пропонують багато переваг перед традиційними сценаріями CGI на стороні сервера. Наприклад, ви можете використовувати JavaScript, щоб перевірити, чи користувач ввів дійсну адресу електронної пошти в поле форми.

Код JavaScript виконується, коли користувач надсилає форму, і всі записи надсилаються на веб-сервер, лише якщо вони дійсні.

JavaScript можна використовувати для запису ініційованих користувачем подій, таких як натискання кнопок, навігація за посиланням та інші дії, які явно чи неявно ініціюються користувачем.

До переваг JavaScript можна віднести:

1. Менше взаємодії з сервером – ви можете перевірити введені користувачем дані перед відправкою сторінки на сервер. Це економить трафік сервера, що означає менше навантаження на ваш сервер.

2. Миттєвий зворотний зв'язок з відвідувачами – їм не потрібно чекати перезавантаження сторінки, щоб побачити, чи не забули вони щось ввести.

3. Підвищення інтерактивності. Ви можете створювати інтерфейси, які реагуватимуть, коли користувач наводить на них мишу або активує їх за допомогою клавіатури.

4. Розширений інтерфейс. Ви можете використовувати JavaScript, щоб включити такі елементи, як компоненти перетягування та повзунки, щоб забезпечити розширений інтерфейс для відвідувачів вашого веб-сайту.

Недоліки JavaScript:

1. JavaScript на стороні клієнта не дозволяється читати або записувати файли. Це зарезервовано з міркувань безпеки.

2. JavaScript не можна використовувати у веб-додатках, оскільки він не підтримується.

3. JavaScript не має багатопоточності або багатопроцесорності.

4. JavaScript не може закрити вкладки, створені не за допомогою JavaScript.

Неможливо зробити міждоменні запити, тобто неможливо отримати доступ до сторінок, закріплених на іншому домені. Навіть якщо сторінки з різних доменів відкриваються одночасно, але в різних вкладках браузера, код JavaScript, що належить одному домену, не може отримати доступ до інформації про веб-сторінки з іншого домену. Але це також має «плюс», оскільки гарантує безпеку інформації, доступ до якої має лише власник веб-сайту чи веб-додатку;

Немає доступу до файлів, розташованих на комп'ютері користувача або за межами самої веб-сторінки, за винятком файлів cookie.

Подібним чином JavaScript — це полегшена інтерпретована мова програмування, яка дозволяє вбудовувати інтерактивність у статичні сторінки HTML.

Однією з головних переваг JavaScript є те, що він не потребує дорогих інструментів розробки. Ви можете почати з простого текстового редактора, наприклад Блокнота. Оскільки це інтерпретована мова в контексті веб-браузера, вам навіть не потрібно купувати компілятор. Приклад коду (рис.2.5):

```

const car1 = {
  maker: 'Ford',
  model: 'Fiesta',
  drive() {
    console.log(`Driving a ${this.maker} ${this.model} car!`)
  }
}
const anotherCar = {
  maker: 'Audi',
  model: 'A4'
}
car1.drive.bind(anotherCar)()
//Driving a Audi A4 car!

const car2 = {
  maker: 'Ford',
  model: 'Fiesta'
}
const drive = function(kmh) {
  console.log(`Driving a ${this.maker} ${this.model} car at ${kmh} km/h!`)
}
drive.call(car2, 100)

```

Рис. 2.5. Приклад коду JavaScript

Щоб полегшити наше життя, різні постачальники придумали дуже хороші інструменти редагування JavaScript. Деякі з них:

- Microsoft FrontPage - Microsoft розробила популярний редактор HTML під назвою FrontPage. FrontPage також надає веб-розробникам багато інструментів JavaScript, які допомагають створювати інтерактивні веб-сайти;
- Macromedia Dreamweaver MX - Macromedia Dreamweaver MX є дуже популярним редактором HTML і JavaScript серед професійних веб-розробників. Він надає кілька зручних попередньо скомпільованих компонентів JavaScript, добре інтегрується з базами даних і відповідає новим стандартам, таким як XHTML і XML;
- Macromedia HomeSite 5 — HomeSite 5 — це популярний редактор HTML і JavaScript від Macromedia для ефективного керування особистими веб-сайтами.

Сьогодні JavaScript від Netscape і JScript від Microsoft відповідають стандарту ECMAScript, хоча обидві мови все ще підтримують функції, які не є частиною стандарту.

2.5. Порівняння фреймворків

Фреймворк — це платформа, яка визначає структуру веб-додатку. Його функціональність дозволяє мережевим ресурсам взаємодіяти з серверами.

На сайтах написаних без фреймворку, контент зберігається на сервері. Тому при завантаженні нового матеріалу сайту потрібно перезавантажити сторінку. Перевага фреймворку є постійні блоки, які зберігаються від однієї конфігурації до іншої це дозволяє вам претендувати на миттєвий зворотній зв'язок з користувачами, коли додається новий вміст.

Цього принципу можна дотримуватися при створенні односторінкових веб-додатків, електронної комерції, хмарних сервісів і багатьох соціальних мереж. Так, здійснювати переходи користувача з одного розділу в інший будуть миттєвими і перевантаження сторінки не відбуватиметься.

Переваги використання фреймворків без змін:

- фреймворк повністю безкоштовний і має відкритий вихідний код;
- використовуються вбудовані шаблони для полегшення створення коду;
- вища якість проекту з меншою кількістю рядків коду;
- швидкісна розробка, яка досягається за рахунок відкритості документації та численних форумах.

Багато приховано труднощів за реалізацією проекту на основі класичного JS і HTML, що призводять до великої кількості фреймворків на ринку. Але ми роздивимось лише три React, Vue, Angular

2.5.1. React

Сьогодні інтерфейсні фреймворки та бібліотеки стають невід'ємною частиною сучасної веб-розробки. React.js — це інтерфейсна бібліотека, яка поступово стала домінуючою структурою сучасної веб-розробки в спільноті JavaScript.

Фреймворк React.js — це фреймворк і бібліотека JavaScript із відкритим кодом, розроблена Facebook. Він використовується для швидкого й

ефективного створення інтерактивних інтерфейсів користувача та веб-додатків із набагато меншим кодом, ніж звичайний JavaScript.

У React ви розробляєте додатки, створюючи повторно використовувані компоненти, які можна вважати окремими кубиками Lego. Ці компоненти є окремими частинами кінцевого інтерфейсу, які, зібрані, утворюють повний інтерфейс користувача програми.

Основна роль React у додатку полягає в обробці рівня перегляду додатка так само, як V у шаблоні Model-View-Controller (MVC), забезпечуючи найкращу та найефективнішу продуктивність візуалізації. Замість того, щоб розглядати весь UI як єдине ціле, React.js заохочує розробників розбивати ці складні UI на окремі багаторазово використовувані компоненти, які утворюють будівельні блоки всього UI. У той же час фреймворк ReactJS поєднує швидкість і ефективність JavaScript з більш ефективними методами маніпулювання DOM для швидшого відтворення веб-сторінок і створення високодинамічних і адаптивних веб-додатків.

Ще в 2011 році Facebook мав величезну базу користувачів і зіткнувся з непростим завданням. Він сподівається надати користувачам більш багатий досвід користувача, створивши більш динамічний, чуйний і ефективний інтерфейс користувача.

Джордан Волке, один із розробників програмного забезпечення Facebook, створив React для цього. React спрощує процес розробки, надаючи більш організований і структурований спосіб побудови динамічних та інтерактивних інтерфейсів користувача з компонентами, які можна використовувати повторно.

Першою його використала стрічка новин Facebook. Завдяки своєму революційному підходу до маніпулювання DOM та користувальницького інтерфейсу React різко змінив підхід Facebook до веб-розробки та швидко став популярним в екосистемі JavaScript після того, як його випустили для спільноти з відкритим кодом.

Зазвичай ви робите запит до веб-сторінки, ввівши її URL у веб-браузері. Потім ваш браузер надсилає запит на сторінку, яку він відображає. Якщо ви клацнете посилання на цій веб-сторінці, щоб перейти на іншу сторінку сайту, на сервер надсилається новий запит для отримання нової сторінки.

Цей шаблон завантаження вперед і назад між браузером (клієнтом) і сервером продовжується для кожної нової сторінки чи ресурсу, до якого ви намагаєтесь отримати доступ на своєму веб-сайті. Цей типовий метод завантаження веб-сайту працює добре, але розгляньте веб-сайт, який дуже керується даними. Попереднє завантаження всієї веб-сторінки було б зайвим і створило б погану взаємодію з користувачем. Крім того, коли дані змінюються в традиційних програмах JavaScript, потрібно вручну маніпулювати DOM, щоб відобразити ці зміни. Ви повинні визначити, які дані змінилися, і оновити DOM, щоб відобразити ці зміни, що призведе до повного перезавантаження сторінки.

React використовує інший підхід, дозволяючи вам створювати так звані односторінкові програми (SPA). Односторінкова програма завантажує лише один документ HTML за першим запитом. Потім він використовує JavaScript для оновлення певного розділу, вмісту або основного вмісту веб-сторінки, який потрібно оновити.

Цей шаблон відомий як маршрутизація на стороні клієнта, оскільки клієнту не потрібно перезавантажувати всю веб-сторінку, щоб отримати нову сторінку кожного разу, коли користувач робить новий запит. Натомість React перехоплює запит і лише отримує та змінює ті частини, які потрібно змінити, не запускаючи повне перезавантаження сторінки. Такий підхід забезпечує кращу продуктивність і більш динамічний досвід користувача.

React покладається на віртуальний DOM, який є копією справжнього DOM. Щоразу, коли стан даних змінюється, віртуальний DOM React негайно перезавантажується, щоб відобразити цю нову зміну. Потім React порівнює віртуальний DOM із справжнім DOM, щоб побачити, що саме змінилося.

Тоді React знаходить найдешевший спосіб виправити фактичний DOM без відтворення фактичного DOM. У результаті компоненти React та інтерфейс користувача можуть дуже швидко відображати зміни, оскільки вам не потрібно перезавантажувати всю сторінку під час кожного оновлення.

2.5.2 Angular

На даний момент загальна продуктивність обладнання користувача збільшується додано, щоб логіку програми можна було виконувати в браузері. Це веде до створення односторонньої програми або SPA, розробка його доручена Angular. У Angular не має контролерів або моделей перегляду, натомість деякі компоненти складаються з шаблонів, класів і метаданих. Існуючий Angular фокусувався на шаблоні MVC, тоді як Angular2 повністю залежить від ієрархії компонентів.

Переваги використання Angular:

- спрощене двостороннє прив'язування даних, тобто Angular дозволяє прив'язувати дані до HTML за допомогою виразів, а директиви Angular дозволяють розробникам розширювати функціональність HTML і створювати нові дизайни. Маніпуляції DOM і код прив'язки даних складено в прості елементи, які можна швидко та легко вставити в HTML шаблони;

- вміння керувати ІІ та ВЕ, тобто Angular входить до пакету MEAN, який також включає MongoDB, Express.js і Node.js. Таким чином, це дозволяє вам керувати інтерфейсом і сервером вашого проекту використовуючи JS.

- модульна концепція, Angular будується відповідно до першочергових методів, тому фреймворк найкраще для розробки зверху вниз. Модульна концепція Angular дозволяє спростити розподіл роботи між різними командами у великих проектах. Найменша кількість коду є пріоритетом, тому програми Angular, зазвичай компактний і його легко редагувати.

Недоліком Angular є надлишковість, тобто він не призначений для роботи з DOM з великим обсягом даних, оскільки він покладається на "брудну

перевірку" керування змінами DOM (будь-які зміни змінних потрібно оновлювати DOM). Це не проблема для більшості сайтів, але редактори GUI та відеоігри можуть лежати.

2.5.3.Vue

Vue — це прогресивний фреймворк для створення власних інтерфейсів. На відміну від монолітних фреймворків, Vue створено для адаптації реалізовувати крок за кроком. Його суть полягає в основному у розв'язанні завдання рівень перегляду, який спрощує інтеграцію з іншими бібліотеками, і існуючий проект. З іншого боку, Vue також ідеально підходить для створення складних односторінкових додатки (SPA, Single-Page Applications), якщо працювати сучасними інструментами та іншими бібліотеками.

Переваги використання Vue

- детальна документація, Vue.js має чудову документацію щодо швидкості навчання розробників і економія часу на розробку програм базовими знаннями HTML і JavaScript;
- адаптивність, Vue.js надає швидкий перехід на інші фреймворки, тому що він схожий на Angular і React за дизайном та архітектурою;
- велика інтеграція, Vue.js можна використовувати для створення однієї сторінки програми та більш складні веб-інтерфейси найважливіше невеликі інтерактивні частини можна легко інтегрувати в існуючі частини інфраструктури без негативного впливу на систему в цілому;
- малий розмір, Vue.js може важити близько 20 КБ і підтримувати свою швидкість, гнучкість, що дозволяє досягти вищої продуктивності, порівняно з іншими фреймворками;
- низький бар'єр входу. Vue вважається одним із найпростіших у використанні фреймворків, тим самим залучаючи нових розробників.

2.6. Вибір текстового редактору

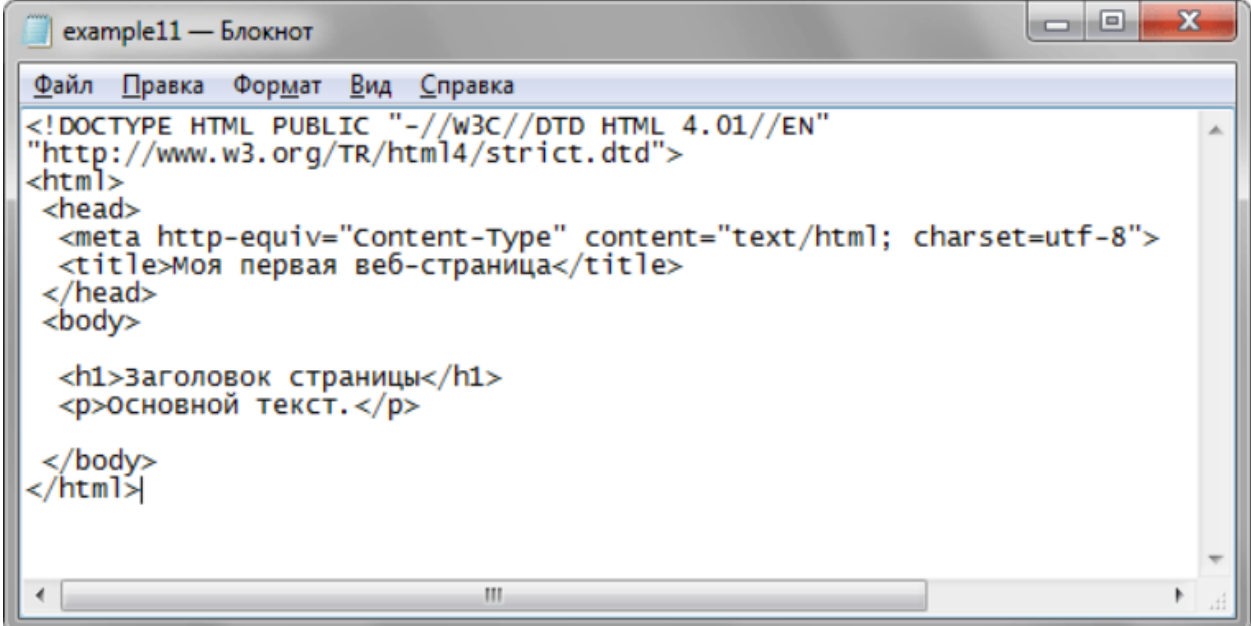
Програмне забезпечення для редагування тексту розроблено, щоб допомогти користувачам читати та писати код. В ідеалі для веб-розробників є інструменти в текстовому редакторі, які допомагають користувачам швидко писати та читати вихідний код. На відміну від інших популярних текстових процесорів, текстові редактори не мають основних функцій. Програма не додає форматування до тексту, а зосереджується на функціях редагування звичайного тексту. Вибираючи найкращий текстовий редактор для веб-розробки, програма текстового редактора має спростити відображення та редагування простих текстових файлів, щоб збільшити розвиток проекту.

Текстові редактори доступні на різних серверах для різних цілей. Вибираючи найкращий текстовий редактор для веб-розробки, можуть бути доступні онлайн-версії для завантаження та пробні версії для нових користувачів. У більшості випадків веб-розробники використовують текстові редактори з більш розширеними функціями, ніж ті, що доступні в Блокноті. Альтернативні текстові редактори можуть виводити більш розширені функції, такі як підсвічування синтаксису та кольорове кодування тексту на основі мови програмування/мови розмітки, на якій вони написані. Інші розширені функції включають розширену підтримку файлів, покращені інструменти пошуку та заміни, клавіші редагування вертикального вибору та порівняння програмної документації. На щастя, Блокнот уже постачається з Windows. Блокнот, однак, є базовим текстовим редактором зі стандартними функціями та меншою кількістю опцій користувача.

2.6.1 Блокнот

Блокнот - розроблений для Microsoft Windows, — це універсальний текстовий редактор, також відомий як редактор звичайного тексту, який дозволяє користувачам комп'ютерів створювати документи(рис.2.7). Блокнот

бере свій початок у 1983 році, коли програма працювала як програма MS-DOS на основі миші. Як універсальний блокнот, історія програми ілюструє час, коли досвідчені програмісти розподіляли ресурси для створення багатогранних електронних таблиць. Наразі оновлена Microsoft додаток містить у своєму Microsoft Store. У результаті програмне забезпечення для редагування тексту оновлюється частіше.



```
example11 — Блокнот
Файл  Правка  Формат  Вид  Справка
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Моя первая веб-страница</title>
  </head>
  <body>

    <h1>Заголовок страницы</h1>
    <p>Основной текст.</p>

  </body>
</html>
```

Рис.2.6. Приклад написання коду в блокноті

У деталях Блокнот містить багато функцій, корисних для веб-розробки. Популярний текстовий редактор для веб-розробників, програма може друкувати файли та легко налаштовувати верхні, нижні колонтитули та поля у файлах налаштування сторінки. Крім того, у верхніх і нижніх колонтитулах розміщуються дати, імена файлів та інша важлива інформація. Різні коди складаються з амперсанда ("&"), після якого йде літера для внесення необхідних коригувань. Крім того, Блокнот містить просту вбудовану функцію журналювання, яка вставляє мітки часу у файли кінця рядка, починаючи з ".log".

Крім того, Блокнот допомагає користувачам установлювати параметри форматування та стилю. Зокрема, програми текстових редакторів вбудовують типи шрифтів і приймають лише текст у форматі CF_TEXT. Зазвичай, коли ви

копіюєте документи з веб-сторінки та вставляєте їх у електронний лист, Блокнот створює коди стилів на основі відформатованого «тексту». Процес включає форматування тексту, тимчасово вставленого в блокнот. Другим кроком є копіювання та вставка документа в інші програми ще раз у скороченому форматі.

Переваги використання блокнота:

Блокнот залишається простим у використанні текстовим редактором для компаній і розробників. Файли зберігаються з розширенням ".txt" без форматування або тегів стилю для легкого редагування системних файлів для використання в середовищі DOS.

Перевагою цієї програми є те, що вона швидко завантажується та обробляє, особливо на апаратному забезпеченні з недостатньою потужністю. За замовчуванням більшість просунутих текстових редакторів, включаючи Блокнот, містять функцію пошуку та заміни для визначення ключових слів користувача. Навіть базові текстові редактори, такі як стандартний Блокнот Windows, включають перенос рядків і слів. Однак розширені текстові редактори часто містять більше налаштованих параметрів упаковки, ніж традиційні програми. Розширений редактор корисний під час перегляду рядка тексту без прокручування по горизонталі. Крім того, Блокнот підтримує мови, що підвищує доступність для користувачів. Крім того, Блокнот виконує типові функції розгортання та згортання, окреслення та приховування коду, що часто називають обтіканням текстом. Текстові редактори дозволяють користувачам приховувати або показувати частини коду або тексту для легшого читання.

Недоліки використання Блокнота:

Блокнот, з іншого боку, не має розширених інструментів редагування тексту і наразі надає базові можливості обробки тексту для стандартної версії та підтримує служби пошуку та заміни для новішої версії. Хоча Блокнот надає простий текстовий редактор для більшості користувачів, обмеження Блокнота спонукають більшість розробників додатків використовувати альтернативні текстові редактори.

2.6.2. Visual Studio Code

Visual Studio Code — це безкоштовний, легкий, але потужний редактор вихідного коду для робочого столу та веб-версії для Windows, macOS, Linux і Raspberry Pi (рис.2.8). Він має вбудовану підтримку JavaScript, TypeScript і Node.js, а також багату екосистему розширень для інших мов програмування (таких як C++, C#, Java, Python, PHP і Go), середовища виконання (наприклад, .NET і Unity), середовища (такі як Docker і Kubernetes) і хмари (такі як Amazon Web Services, Microsoft Azure і Google Cloud Platform).

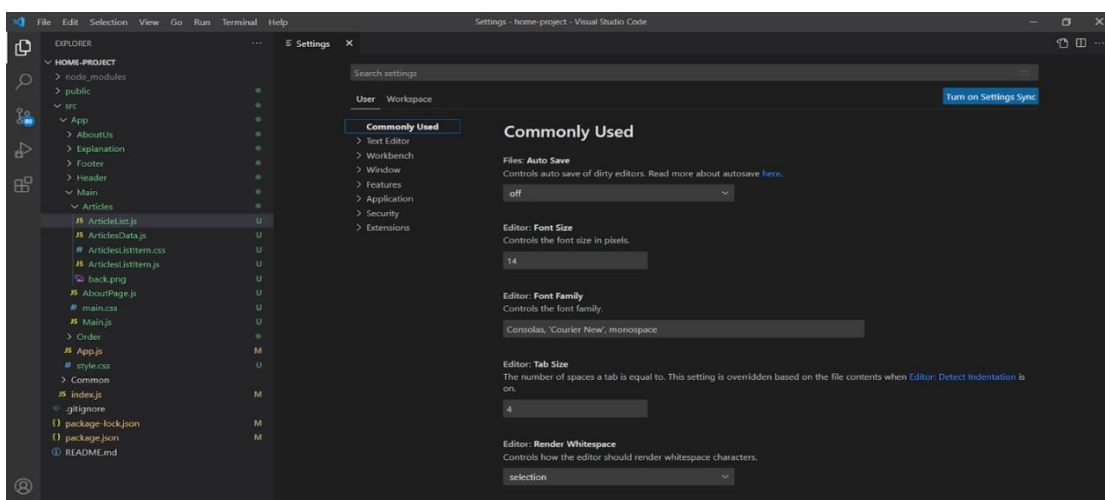


Рис.2.7. Вигляд Visual Studio Code

На додаток до загальної ідеї щодо легкості та швидкості запуску та запуску Visual Studio Code має вдосконалення Code IntelliSense для змінних, методів і модулів імпорту; графічне налагодження; потужні функції редагування, такі як стрічка, редагування кількома курсорами, підказки параметрів тощо, швидка навігація та рефакторинг коду, а також вбудований контроль джерела, включаючи підтримку Git. Більшість із них адаптовані з технологій Visual Studio.

Власний код Visual Studio створено за допомогою Electron, Node.js, TypeScript і протоколу Language Server Protocol і оновлюється щомісяця. Багато розширень оновлюються так часто, як це необхідно. Різноманітність підтримки залежить від мови програмування та її розширень, від простого підсвічування синтаксису та дужок до налагодження та рефакторингу. Якщо

мовний сервер недоступний, ви можете використовувати шейдери TextMate, щоб додати базову підтримку бажаної мови.

Код у репозиторії кодів Visual Studio є відкритим кодом за ліцензією MIT. Сам продукт Visual Studio Code знаходиться під стандартною ліцензією на продукт Microsoft, оскільки він має невеликий набір налаштувань Microsoft. Хоча є комерційна ліцензія, вона безкоштовна.

Visual Studio Code має мільйони активних користувачів, не лише в Microsoft. Наприклад, багато користувачів VS Code є розробниками в Google або Facebook. Багато користувачів VS Code є дослідниками даних, які хочуть отримати повну підтримку мови Python або R під час використання Jupyter Notebooks.

Розробникам подобається простота використання Visual Studio Code як редактора та його здатність перевіряти синтаксис, завершувати код, виконувати рефакторинг коду, налагоджувати та передавати в репозиторій. Розробникам хмар і контейнерів подобаються можливості віддаленого керування VS Code і його явна підтримка основних хмар, Docker і Kubernetes. Розробники, які працюють у командах, люблять інтеграцію Git від VS Code.

2.6.2 Sublime

Sublime — швидкий кросплатформенний текстовий редактор. На даний момент він є одним з найпопулярніших редакторів. Перша версія Sublime була створена в 2008 році тільки для Windows (рис.2.8). У цьому середовищі були можливі: пошук файлів, підтримка гарячих клавіш, панель швидкого доступу для перегляду файлів, об'єднання файлів в один проект.

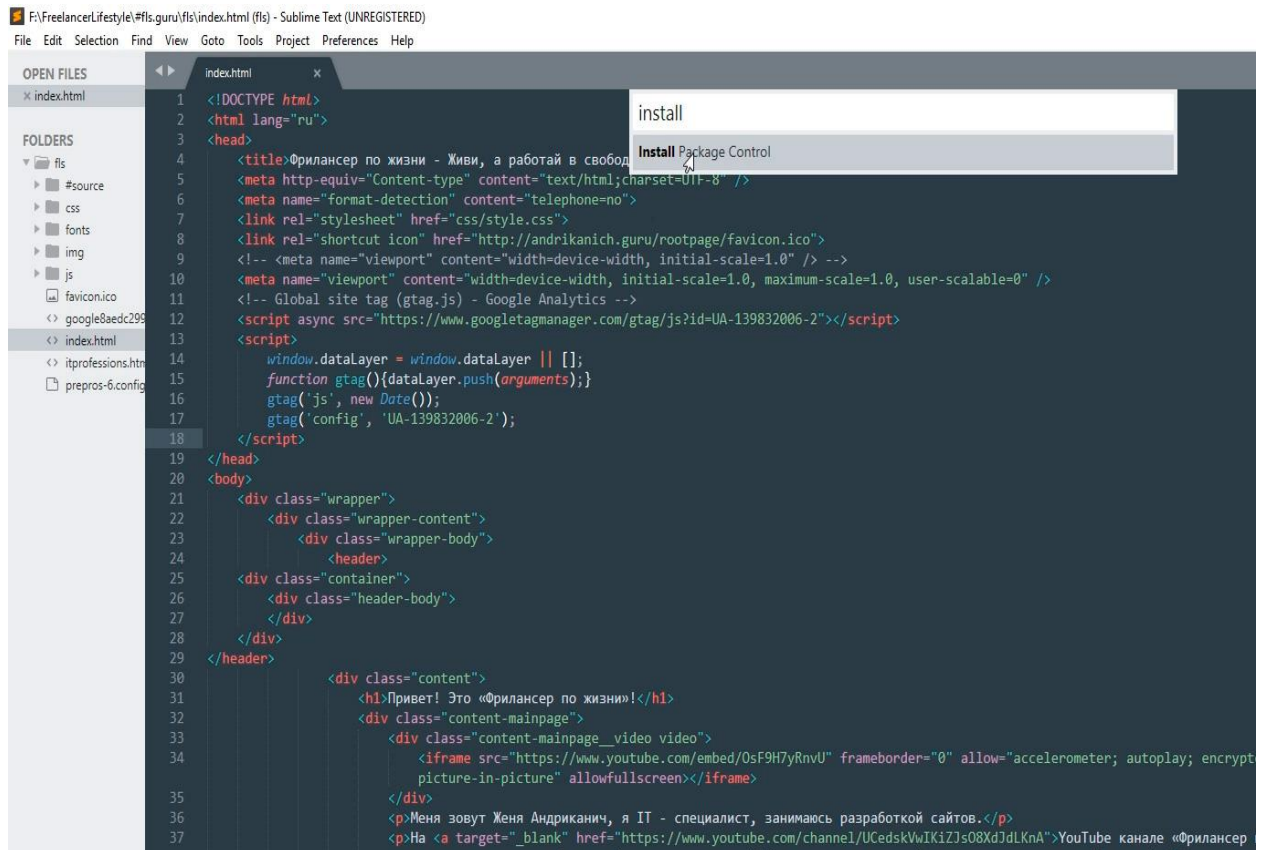


Рис. 2.8. Интерфейс редактору Sublime

Чотири роки потому вийшла друга версія Sublime, редактор став більш відкритим, доступним для Linux і Mac, а також були додані нові функції: підсвічування синтаксису Go, швидкі переходи до різних розділів коду та швидкі зміни проекту.

Через рік, у 2013 році, вийшло третє видання, яке допрацьовувалося і допрацьовувалося до 2018 року. Наразі її запрошуватимуть розробники, серед покращень яких можна виділити: збільшення швидкості запуску програми, покращений інтерфейс, використання Python 3 як плагінів, індексування файлів для пошуку анонсів змінних, функцій і класів, зміна плагінів в окремому процесі Швидше завантаження, можливість повернення курсору в попереднє положення.

На даний момент Sublime Text 4 знаходиться в стадії активної розробки і доступна лише деяким користувачам, але вони стверджують, що ця версія дуже швидка, а прокручування в проектах стало більш плавним.

Недоліки Sublime:

1. Швидкість, оскільки ця програма важча за Visual Studio Code і розроблена для великих проектів. Текстовий редактор може відкривати та працювати з кількома файлами, дозволяючи користувачам перехресно перевіряти свої файли для швидшого створення проекту та кращої організації.

2. Ціна, є безкоштовний період використання – місяць, але його може не вистачити для початківців аби розібратися з тим, чи підходить їм Sublime, чи зручно з ним працювати, чи вистачає їм швидкості, а вже повна ліцензія коштує 80 доларів.

3. Відсутня така функція, як автоматичне налагодження, проте як Visual Studio Code має її і тим самим полегшує життя розробників.

Коли ви вперше відкриваєте Sublime, він виглядає як простий редактор. Це тому, що ідея цієї програми полягає в тому, що користувач сам може додати те, що йому потрібно для роботи. Хоча це легко для досвідчених розробників, початківці можуть ще не знати, що їм потрібно, і витратять багато часу, щоб це зрозуміти.

ВИСНОВКИ ДО РОЗДІЛУ 2

У цьому розділі описано прийоми, які будуть використані для розробки веб-додатку для продажу еко-іграшок для маленьких дітей. Розглянули можливість використання такого додатка, як Figma, для створення макетів сторінок, адже даний додаток є безкоштовним та є автозберігання. В даному розділі приведено опис мови програмування JavaScript та її бібліотеку React показали їх переваги чи , а також порівняння різних видів редакторів, таких як: Блокнот, Visual Studio Code та Sublime, текстові редактори для створення проектів, розкриває їхні плюси та мінуси.

Вибір впав на Visual Studio Code, оскільки він безкоштовний та має головну перевагу – продуктивність, тоді як в Блокноті ми не можемо побачити свої помилки, працювати з терміналами, або ж поширювати свій код поміж декількох кодерів одразу або Sublime, який займає більше часу для завантаження бібліотеки в проект і, як правило, займає більше часу для запуску проекту. Вибір впав на Visual Studio Code, оскільки він безкоштовний та має головну перевагу – продуктивність, тоді як в Блокноті ми не можемо побачити свої помилки, працювати з терміналами, або ж поширювати свій код поміж декількох кодерів одразу або Sublime, який займає більше часу для завантаження бібліотеки в проект і, як правило, займає більше часу для запуску проекту.

РОЗДІЛ 3 ОПИС ПРОТОТИПУ WEB-ДОДАТКУ МАГАЗИНУ ДИТЯЧИХ ІГРАШОК

3.1 Створення макету

Для створення макету я обрала програму Figma, даний додаток є безкоштовним, може працювати без використання інтернету та там є постійні автозбереження, це дуже допомагає особливо з постійними відключеннями електроенергії. Створення макету починається зі створення рамки, поля, стандартного розміру 1600 x 900 пікселів(рис.3.1).

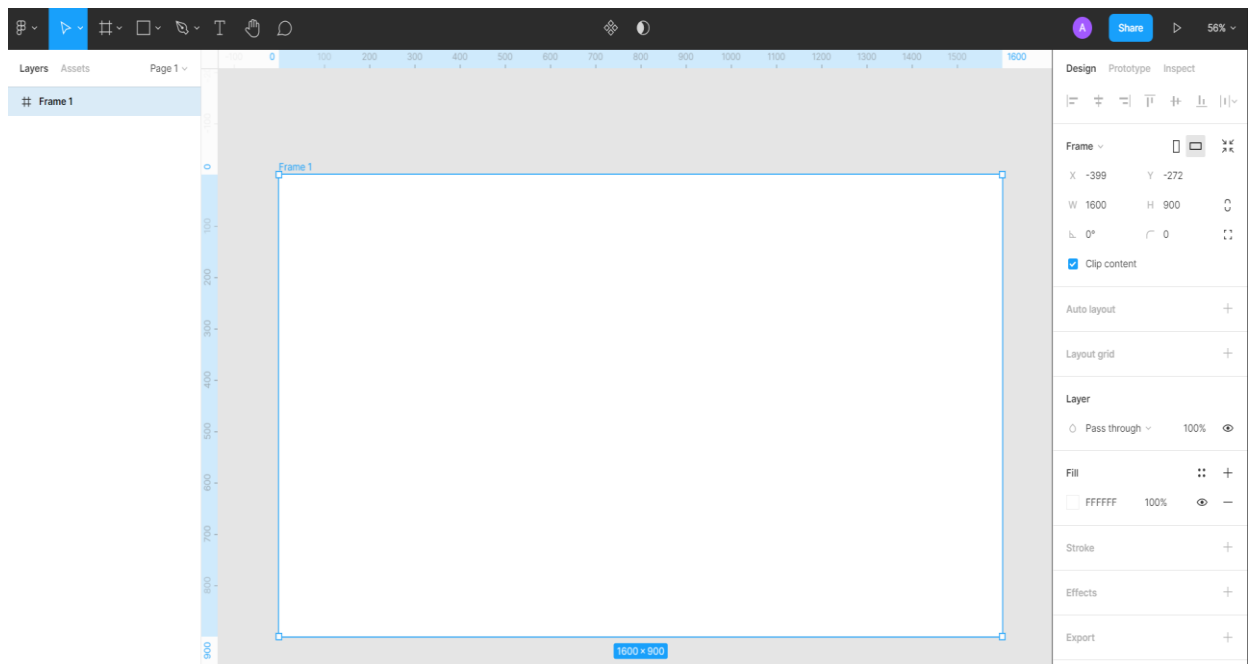


Рис.3.1. Створення фрейму

Кафедра КІТ				НАУ 22 34 80 000 ПЗ			
	ПІБ			Опис прототипу web- додатку магазину дитячих іграшок	Літ.	Аркуш	Аркушів
Розроб.	Фатькова О.О.						21
Керівник	Боровик В.М.				ТП-215М - 122		
Н.Контр.	Толстікова О.В.						

Відразу створюємо сітку з 12 стовпчиків з відступами в рамці по обидві сторони від 160 пікселів для зручності розміщення деталей (рис 3.2).

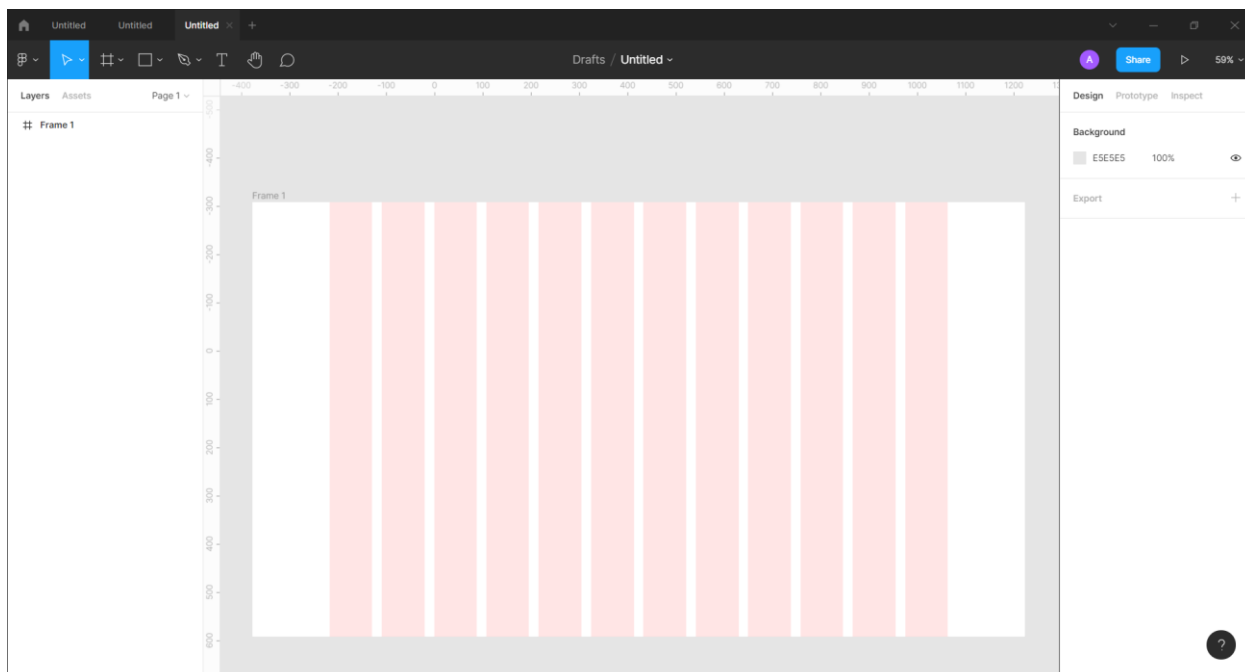


Рис. 3.2. Створення сітки

Наступним етапом є підбирання кольорів які будуть використатись по всьому веб-додатку, адже кольори повинні бути приємними для ока користувача, а також кольори повинні повністю відображати ваш маганчик, тож якщо в мене веб-додаток пов'язан з еко-іграшками, кольори повинні бути сонячні адже жовті кольори асоціюються зі світлом, добром, теплом, а це саме ті асоціації які повинні визиватись при погляді на дитячу іграшку, з зеленими відтінками щоб підкреслювати еко походження іграшок, тож це є головним етапом(рис.3.3).



Рис.3.3. Палітра для сайту

Для кожного додатку головним є створення логотипу, свій логотип я зробила простим та лаконічним, використовуючи кольори з обраної палітри, зелений та жовтий, та дві літери «Е» та «Т», як аббревіатуру від назви магазину «Есо-Тоys» (рис.3.4).



Рис.3.4. Логотип

Наступним етапом є створення фону для нашого веб-додатку, робимо його максимально нейтральним та щоб він не забрав всю увагу на себе, але й підкреслював те чим самим ми займаємось та використовуючи ті кольори та їх відтінки які ми взагалі обрали для палітри веб-додатку. Для фону я обрала зробити плями кольорів з обраної палітри для того щоб якось розбавити однотонний фон, але й не забирати увагу від фотографії обраної для головної сторінки. Для того щоб створити плями в Figma ми повинні завантажити плагін Blob, запустивши плагін ми побачимо екран для вибору форми для плям (рис.3.5.).

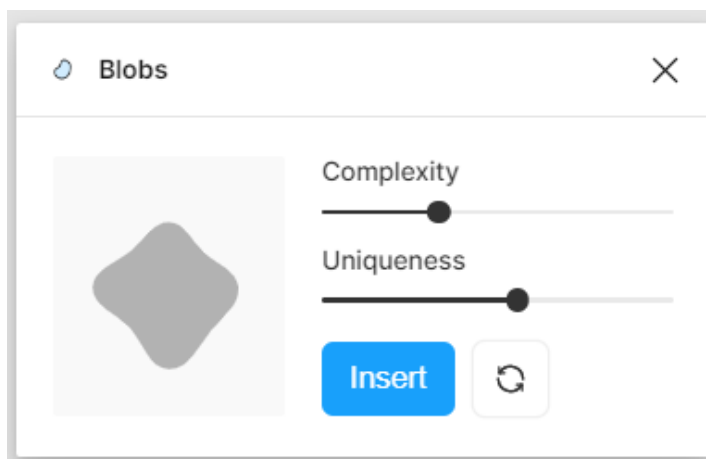


Рис.3.5. Плагін Blobs

Після того як я обрала форму для клякс, я починаю підбирати для них колір, щоб вони не забирали весь акцент на себе зі сторінки та товарів які є в нашому маганчику. (рис.3.6.):

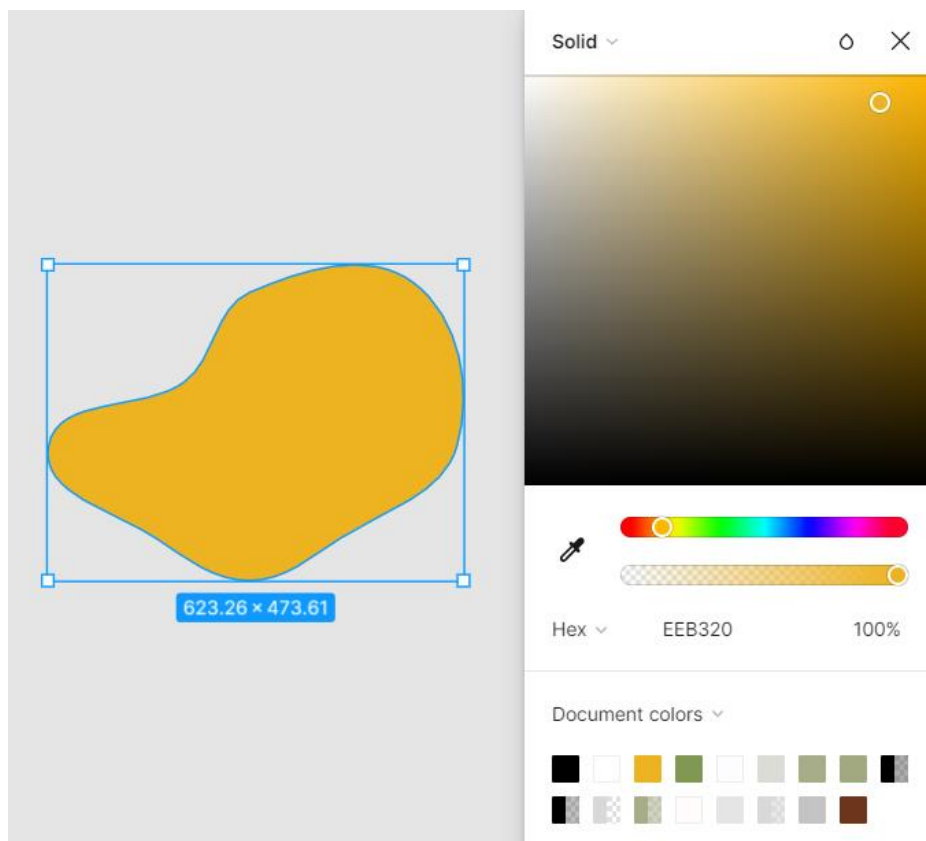


Рис.3.6. Вибір кольору для плями

Коли вже всі плями зроблені та обрані кольори для них, наступним є те, що їх потрібно розмістити так щоб вони не була одна на одній, тобто щоб для ока було приємно дивитись (рис.3.7.). Наступним кроком для стало те, щоб яось різнообразити свій дизайн головної сторінки, було обрано додати токи лінії для фону, які б підкреслювали наші клякси та й яось розбавляли наш фон, для цього ми використовуємо функціонал Фігми, а саме «Pen» і починаю малювати різні форми замикаючи останні точки між собою, та приблизно повторюючи форму наших клякс (рис.3.8.).



Рис.3.7. Створення плям для фону

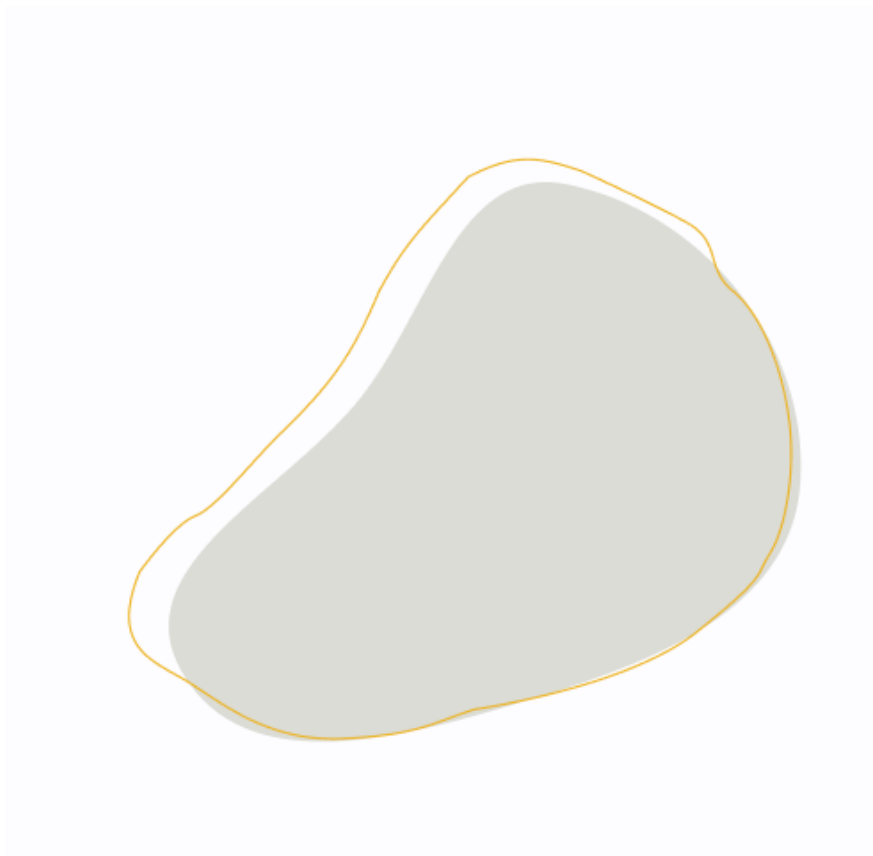


Рис.3.8. Створення ліній для фону

Наступним етапом є поєднати наші плями, лінії та логотип (рис.3.9.)



Рис. 3.9. Фон веб-додатку

Для розуміння користувача що саме представляє з себе наш додаток ми повинні помістити якусь картинку, а саме одну іграшку, яка є в асортименті магазину, додатки інформацію про наш магазинчик, а саме написати чим саме ми займаємось, кнопку для замовлення при натисканні якої користувач одразу переходить до початку нашого каталогу з іграшками. А також ми розмістили кнопку для спливаючого меню та головне – назву веб додатку – «Eco-toys» для назви був обраний незвичний шрифт – «Indie Flower» для того аби при передивлянні веб-додатку назва привертала на себе увагу і користувач розумів саме в якому додатку він знаходиться (рис.3.10.).



Рис.3.10. Додавання інформації на головну сторінку

Наступним етапом є створення меню-каталогу для полегшення користування веб-додатком користувачам. Воно буде показуватись як тільки користувач натисне на позначку меню на головному меню. Для його закриття треба буде натиснути на три горизонтальні білі лінії в правому верхньому кутку меню(рис.3.11.). В саме меню ми вставляємо такі розділи як:

- О нас;
- Гризунки;
- Бізібук;
- Слімбуси;
- Подарункові бокси;
- Відгуки;
- Оплата;
- Контакти.

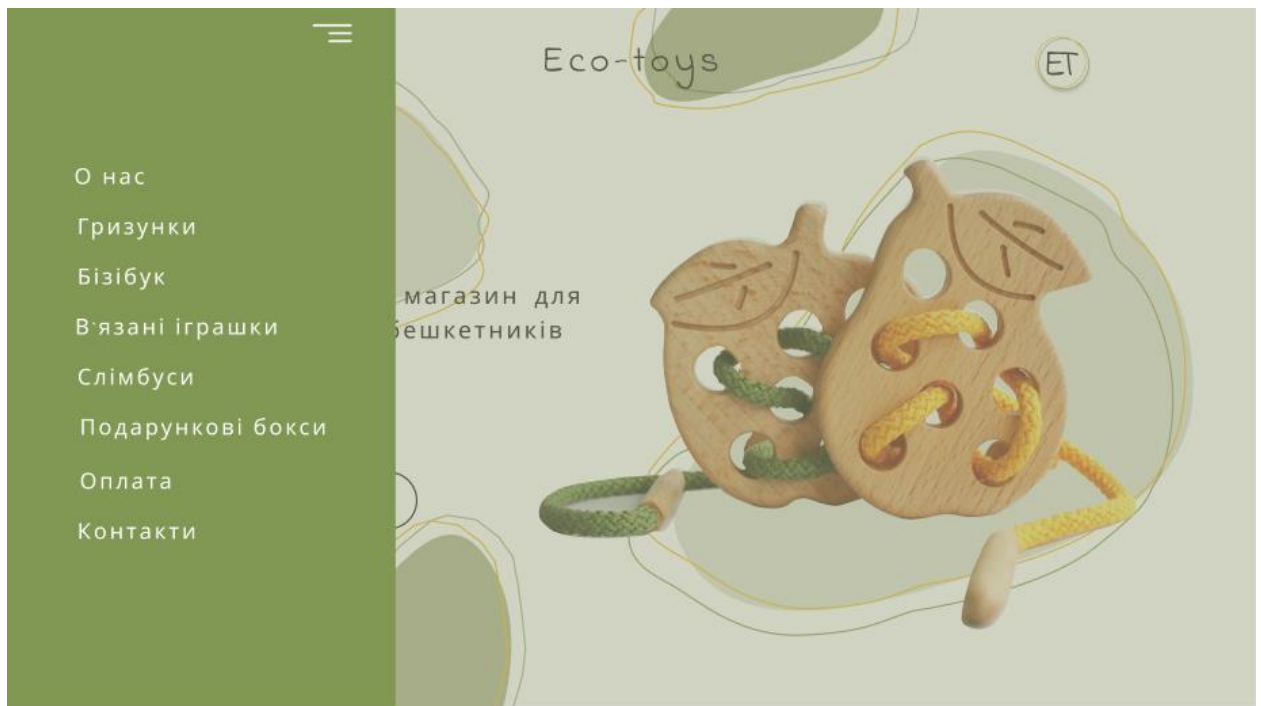


Рис.3.9. Меню

Далі йде розробка не менш важливих частей веб-додатку, а саме окремі сторінки відділів, перша йде «О нас». Вона буде включати в себе інформацію створення та мета данного магазинчику, фотографію дитини з нашою іграшкою, логотип, назву веб-додатку та функцію переходу в меню (рис.3.10.).



Рис.3.10. Сторінка «О нас»

Наступною є сторінка з гризунками (рис. 3.11.), в магазинчику є велика кількість гризунків, тож потрібно зробити ще й перемикач між сторінками (рис. 3.12.):



Рис.3.11. Перша сторінка розділу «Гризунки»

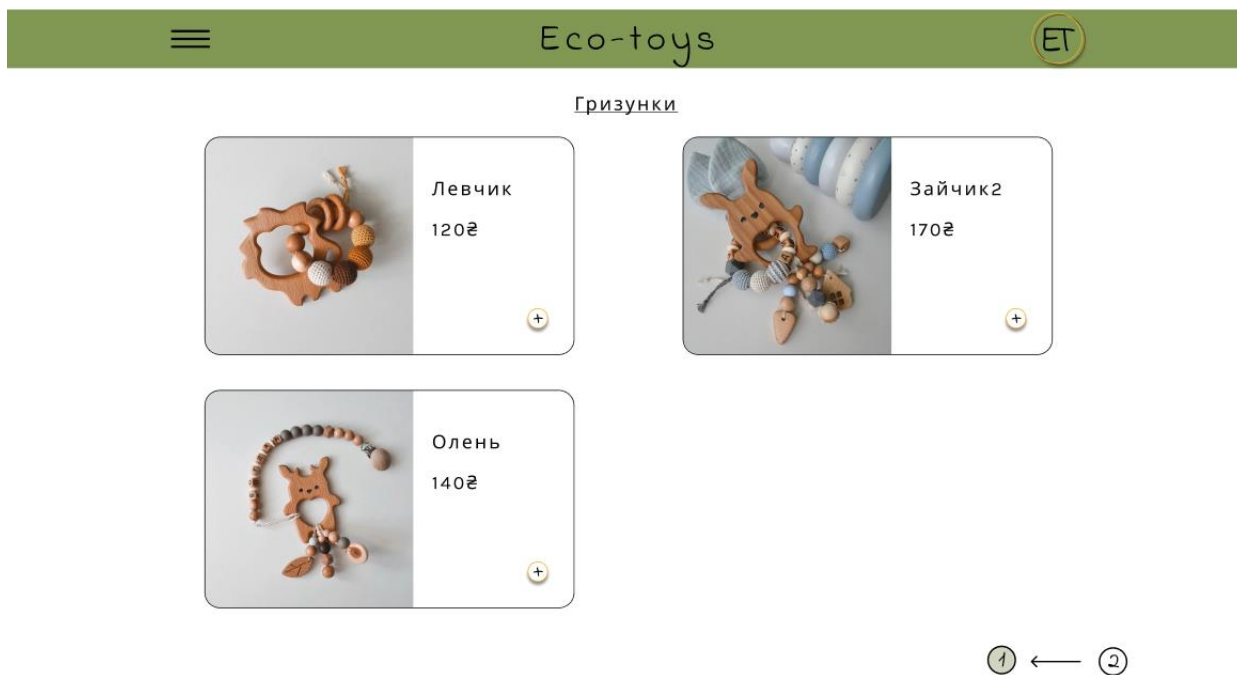


Рис.3.12. Друга сторінка розділу «Гризунки»

Сторінка з бізібуками (рис.3.13.):



Рис.3.13. Сторінка «Бізібуки»

Сторінка з в'язаними іграшками (рис.3.14.):



Рис.3.14. Сторінка «В'язані іграшки»

Сторінка додатку зі слінгбусами (рис.3.15):



Рис.3.15. Сторінка «Слімбуси»

Сторінки для подарункових боксів (рис.3.16) та (рис.3.17):



Рис.3.16. Перша сторінка «Подарункові бокси»



Рис.3.17. Друга сторінка «Подарункові бокси»

Сторінка з відгуками (рис.3.14):



Рис.3.18. Сторінка «Відгуки»

Сторінка з Оплатою, на цій сторінці ви одразу побачите ті самі іграшки які ви обрали, та загальну суму до сплати (рис.3.19):



Рис.3.19 Сторінка «Оплата»

Сторінка для вводу даних зі своєї картки для оплати ваших товарів (рис. .):

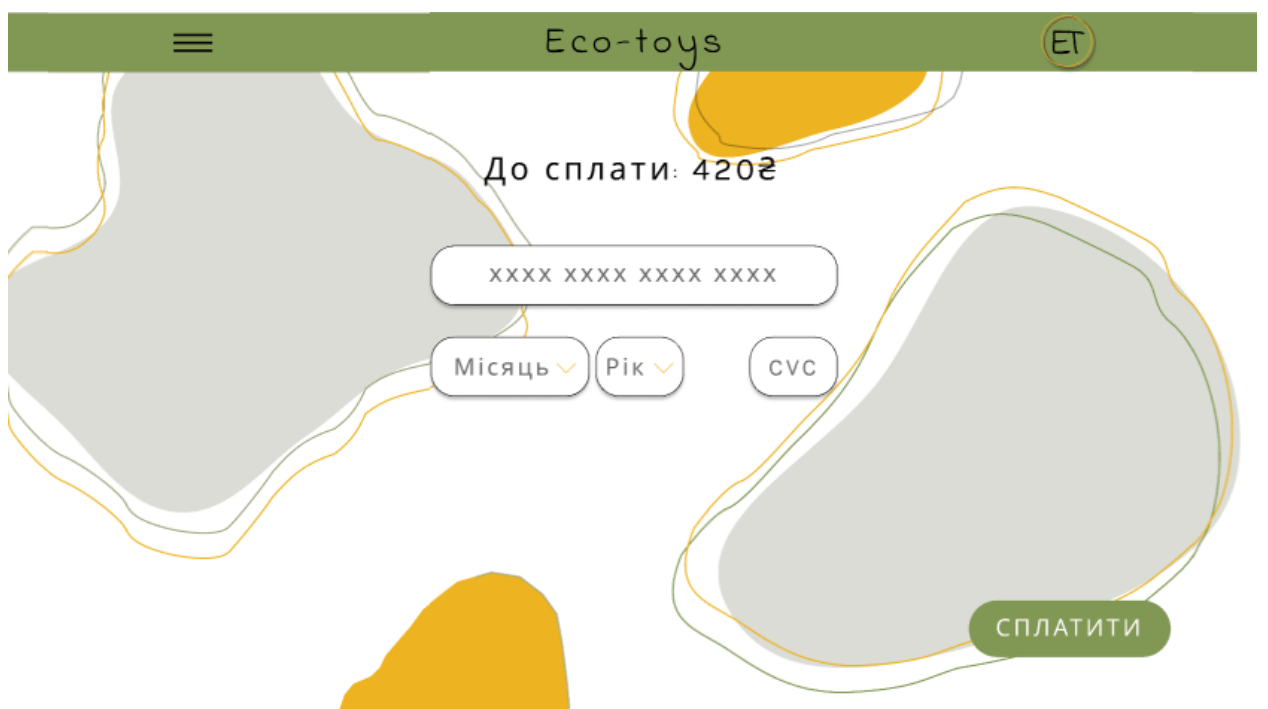


Рис.3.20. Сторінка для вводу даних картки

Після вводу даних та успішної верифікації у вас одразу з'явиться повідомлення про успішно оплату, та те що з вами зв'яжеться оператор (рис.3.21) :



3.2 Верстка макету

Для того, щоб розпочати верстати макет перш за все потрібно встановити програмне забезпечення, а саме Visual Studio Code. Його можна знайти просто ввівши назву в пошукову строку Google назву цього додатку, одразу покажеться офіційна сторінка, де ви зможете обрати яку саме версію скачати, на iOS чи Windows. Даний додаток є повністю безкоштовним, і завантаження займає лише 5 хвилин.

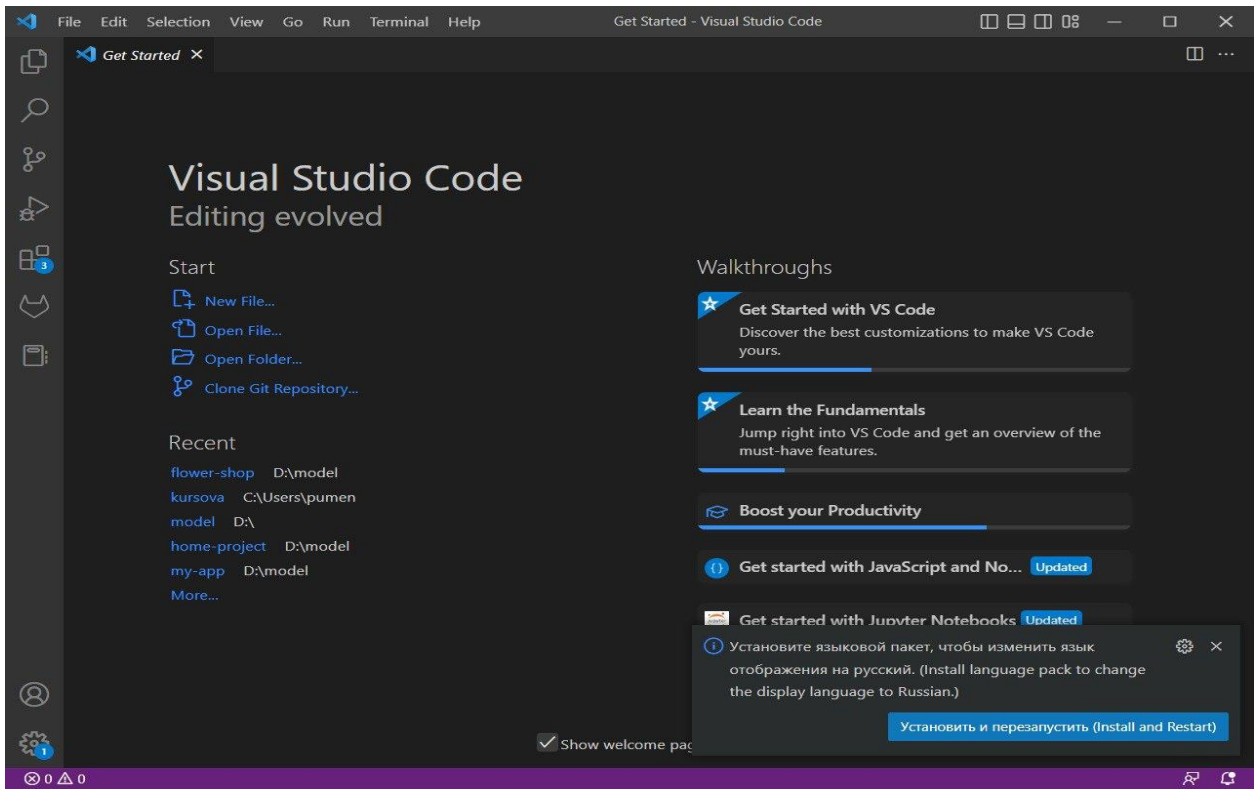


Рис. 3.24. Розпочаток роботи в Visual Studio Code

Для початку роботи в програмі та для початку створення якогось проекту, потрібно відкрити термінал в Visual Studio Code(рис.3.25)

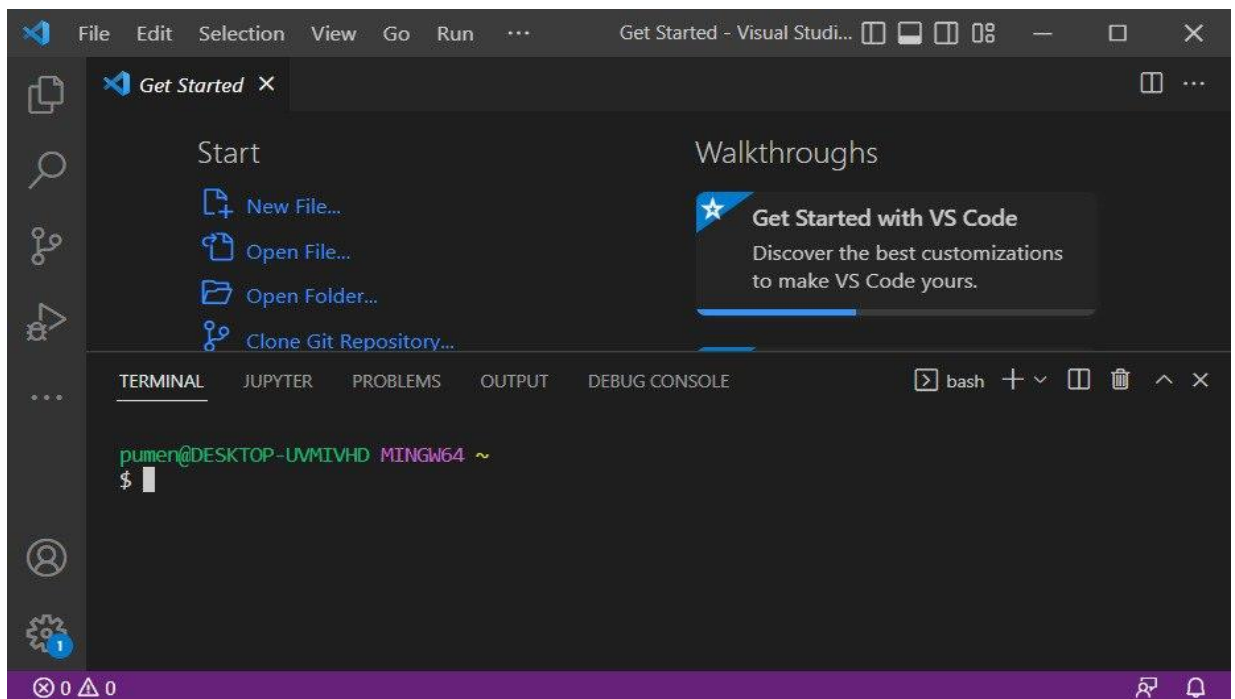


Рис. 3.25. Розпочаток роботи в Visual Studio Code

В терміналі необхідно ввести команди, для того щоб на вашому комп'ютері створилась папка зі всіма необхідними файлами (рис.3.26):

```
npx create-react-app my-app
cd my-app
npm start
```

Рис. 3.26. Команди для початку проекту

Після того як ми пропишемо ці команди, почекавши поки проект буде завантажуватись ми зможемо побачити у боковій панелі такі папки:

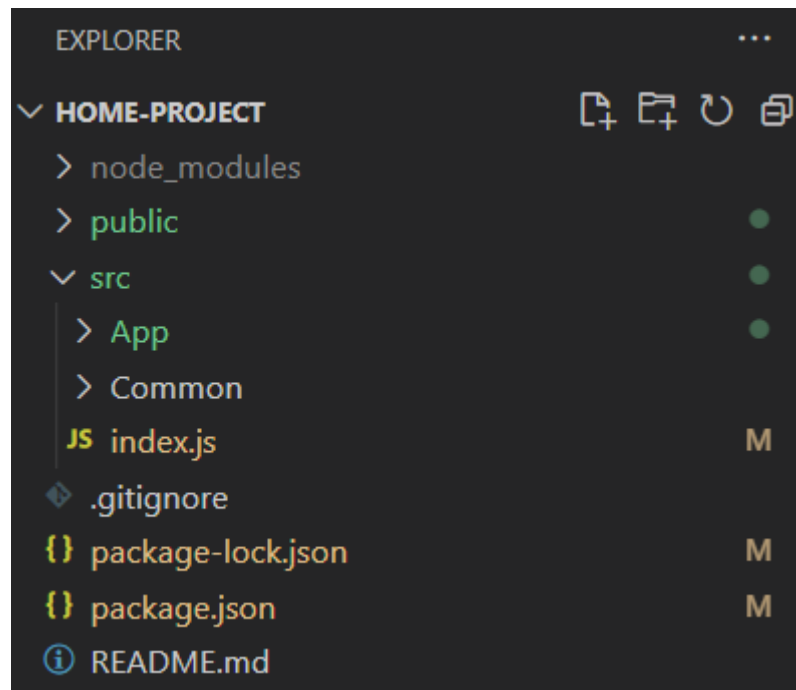


Рис. 3.27. Бокова панель

Наступним кроком у створенні сайтів буде розбити сайт на блоки, а саме:

- Main;
- Menu;
- About us;
- Rodents;
- BusyBooks;

- Slimbuses;
- SurpriseBox;
- Reviews;
- Payment;
- Contacts.

Після розбиття сайту на блоки, під кожний блок потрібно створити папки у папці App з тими ж іменами папки (рис.3.28).

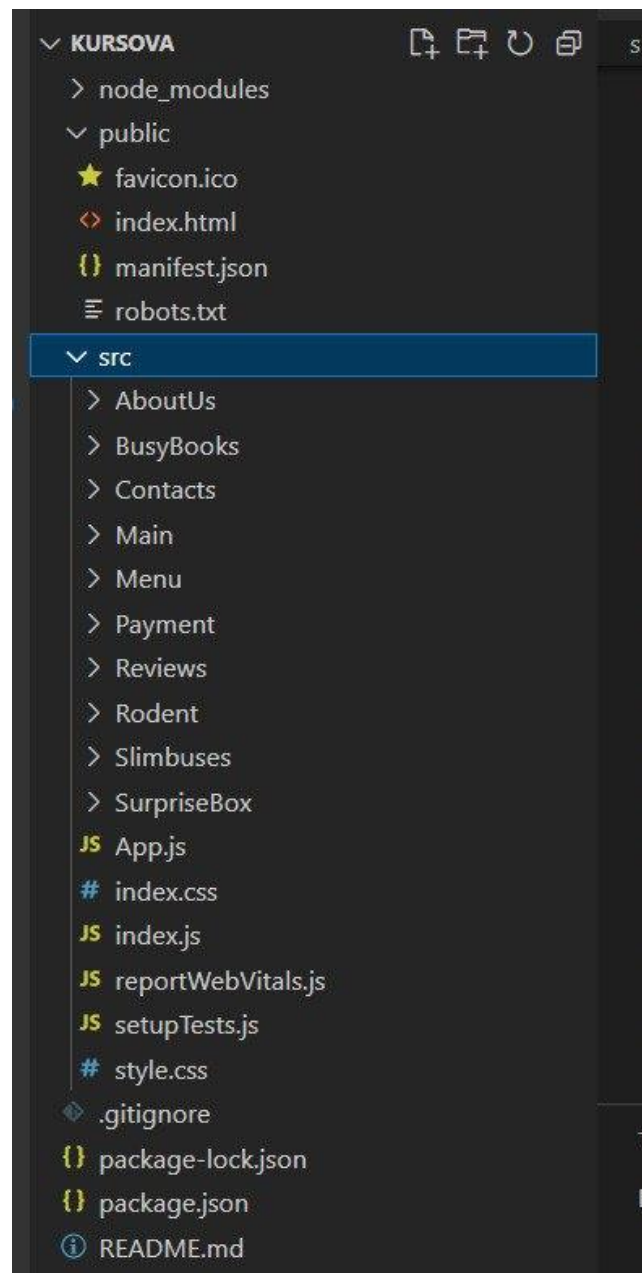


Рис.3.28. Папка App

В тільки но створених папках треба створити файли з розширеннями «.js» та «.css», тобто в папці Main треба створити Main.js та Main.css (рис. 3.29).

їх потрібно викликати в спільному файлі App.js (рис. 3.30) для того, щоб вони відображувались на сторінці.

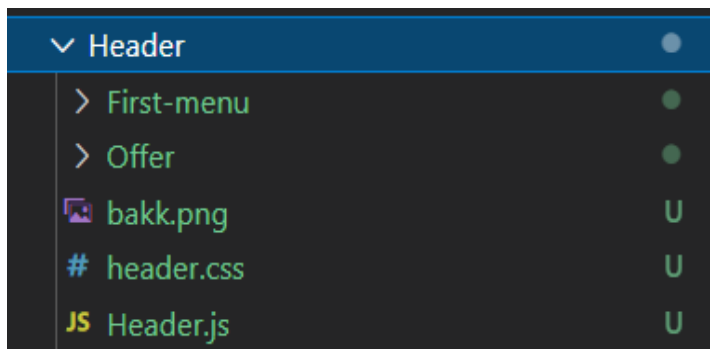


Рис. 3.29. Папка з розширеннями

```
import React from 'react';
import Header from './Header/Header'
import Explanation from './Explanation/Explanation'
import Main from './Main/Main'
import AboutUs from './AboutUs/AboutUs'
import Footer from './Footer/Footer'
import Order from './Order/Order'
import './style.css'

function App() {
  return (
    <div>
      <Header/>
      <Explanation/>
      <Main/>
      <AboutUs/>
      <Footer/>
      <Order/>
    </div>
  );
}

export default App;
```

Рис. 3.30. Файл App.js

Написання коду слід починати з самого початку і послідовно, тож почнемо з файлу Header.js, слід пам'ятати, що для більш ефективного написання коду, блок Header слід також поділити на блоки: Menu (рис.3.31.),

Text, Offer (рис. 3.32.), для кожного цього блоку ми заводимо окремі папки, в яких створюємо файли з тими ж назвами, але різними розширеннями. Для їх відображення треба їх викликати у файлі Header.js. та задати стилі цьому блоку.

```
const Menu = () => {
  return(
    <div className="header-menu">
      <p className="logo">FFlower</p>
      <ul>
        <a href="#aboutus" className="header-menu-link">Чому саме ми</a>
        <a href="#catalog" className="header-menu-link">Каталог</a>
        <a href="#delivery" className="header-menu-link">Доставка і оплата</a>
      </ul>
      <button className="header-button">
        +380675895111
      </button>
    </div>
  )
}
export default Menu
```

Рис.3.31. Блок Menu

```
const Offer = () => {
  return(
    <div className='information'>
      <div className='row'>
        <p className='big-number'>3</p>
        <div>
          <p>роки безперервної</p>
          <p>роботи і досвіду в</p>
          <p>створення букетів</p>
        </div>
      </div>
      <div className='row'>
        <p className='big-number'>10</p>
        <div>
          <p>людей в команді</p>
          <p>які шалено</p>
          <p>горять своєю справою</p>
        </div>
      </div>
      <div className='row'>
        <p className='big-number'>1000</p>
        <div>
          <p>задоволених клієнтів</p>
          <p>і красивих букетів,</p>
          <p>викликаючих захват</p>
        </div>
      </div>
    </div>
  )
}
```

Рис. 3.32. Блок Offer

```

const Header = () => {
  return(
    <div className='header'>
      <Menu/>
      <div className="text">
        <div className='main-text'>
          <p className='first-line'>Унікальні букети</p>
          <p className='second-line'>на будь-який бюджет</p>
        </div>
        <div className='second-text'>
          <p>Зберемо букет мрії і доставимо</p>
          <p className='second-text-line'>по Києву і КО за 1 годину</p>
        </div>
        <div className='second-text third'>
          <p>Залиште відгук і отримаєте</p>
          <p className='second-text-line x'>знижку 15% на ваш наступний замовлення!</p>
          <a href='#catalog' className="link">
            Обрати букет
          </a>
        </div>
      </div>
    </div>
  )
}

```

Рис.3.33. Блок Header

```

.header{
  background: url(bakk.png);
}
.text{
  margin-left: 160px;
  margin-right: 160px;
}
.main-text{
  font-family: 'Yeseva One';
  font-size: 52px;
}
.first-line{
  color: #EF6898;
  margin-top: 141px;
  margin-bottom: 0px;
}
.second-line{
  color: #383838;
  margin-top: 12px;
  margin-bottom: 28px;
}
.second-text p{
  font-family: 'Comfortaa';
  font-size: 18px;
  color: #383838;
  margin-bottom: 20px;
}

```

Рис.3.34. Стили блоку Header

По такому принципу будуть написані й інші блоки сторінки і в кінцевому варіанті папка App буде виглядати так:

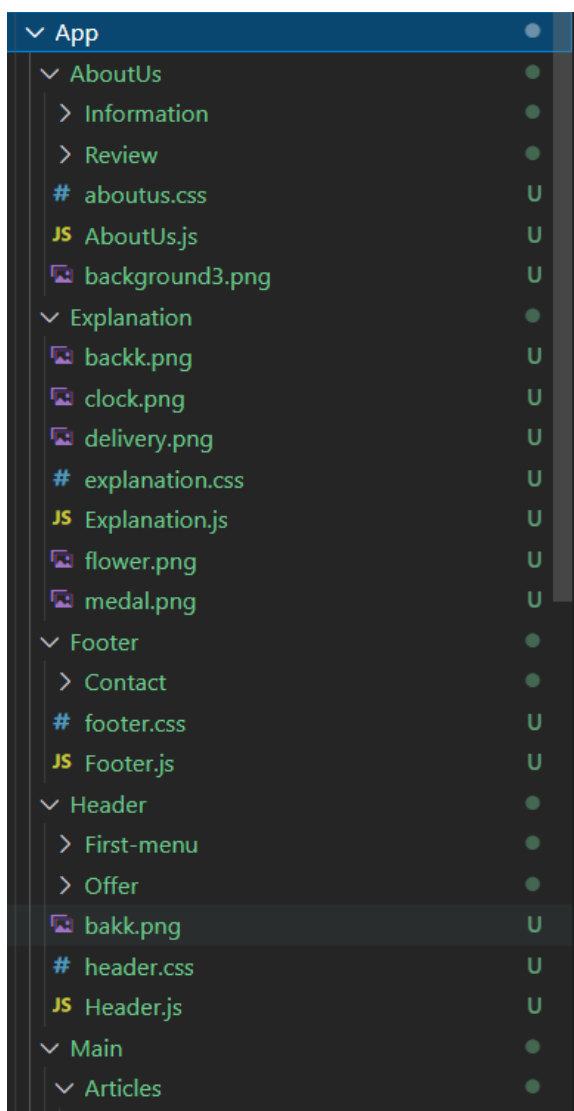


Рис.3.35. Остаточний вигляд App

Після прописання всіх блоків та елементів сторінки її треба запусити і подивитись, як воно виглядає у браузері і якщо є якісь розбіжності із макетом, то прибрати їх. Запустити проект можна прописав у терміналі код:

\$ npm start

І ми отримаємо результат.

ВИСНОВОК ДО РОЗДІЛУ 3

Даний розділ демонструє, як поетапно створити веб-додаток з продажу еко-іграшок.

Першим що потрібно зробити, це визначитися з кольорами аби вони не були занадто яскравими, а також підкреслювали приналежність кольорів до тематики веб-додатків, тож були обрані спокійні, сонячні, зеленуваті кольори.

Другим кроком є створення макету веб-додатку, для цього було обрано програму Figma. Ця програма має багато переваг, адже є автозбереження, ви можете працювати одразу на декількох пристроях, дані передаються одразу.

Остаточним кроком є написання коду, було обрано додаток Visual Studio Code. В даній програмі є багато переваг, наприклад, він є безкоштовним, є автовиправлення коду, а також можуть вискакувати підказки для полегшення написання коду.

Завдяки добре обраному програмному забезпеченню створення веб-додатку було легким та приємним.

Цей Web-додаток орієнтовано на родини в яких є маленькі діти, а наразі це кожна третя родина, тож додаток є актуальним. У проекті зроблено легкий для розуміння інтерфейс, використано дизайн за новими тенденціями, використано приємні для ока кольори, додано функціонал, який полегшить перебування користувача у веб-додатку. Завдяки чистому та легкому коду, можна розширювати асортимент додатку, доповнювати функціонал.

ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є веб-додаток з продажу еко-іграшок, а саме “Есо Toys”. Цей додаток можна вже використовувати у повсякденному житті, та застосовувати у різних середовищах.

Наразі, через ситуацію в країні, все більше мешканців цих територій вирішили покинути свої міста та домівки, щоб убезпечитися від постійних загроз безпеці, артилерійського вогню, налітаючих ракет або бомб. У такі моменти прийняття рішень, коли стрілка годинника збігає з годинами чи навіть хвилинами, люди беруть лише найнеобхідніше.

Найбільше у цій ситуації страждають діти, адже вони не розуміють що коїться і їх постійно треба заспокоювати, відволікати, а саме – іграшками, та через нажаль неможливо взяти всі із собою, через це береться маленька кількість. Тож для кваліфікаційної роботи було обрано зробити веб-додаток, який би допоміг сім’ям для швидкого вибору іграшок, а також по адекватним цінам.

На даний момент кількість користувачів Інтернету в Україні постійно зростає, а тому зростає попит на послуги Інтернету, а саме онлайн-спілкування, онлайн-покупки та читання новин. Цей період є критичним для переходу більшості галузей бізнесу в Україні на онлайн-платформи.

Завдяки Інтернету український бізнес може хоч якось пережити воєнний час, а не потрапити у великий «мінус» чи взагалі не закритися. Адже інтернет-магазин дозволяє власникам бізнесу зменшити витрати на оренду приміщення, продаж своїх товарів, розширити асортимент, а також завдяки всьому цьому вони мають можливість не втратити своїх співробітників, а також виплачувати їм заробітну плату та утримання їх у важкі часи.

Спочатку створювалися звичайні сайти які орієнтувалися як ринок роздрібної торгівлі, або web-портали, де систематизувалася корпоративна інформація та доступ до неї мала лише певна кількість людей.

З усього вище сказаного однозначно зрозуміла актуальність питання дослідження Веб-додатків та використання новітніх технологій.

В результаті виконання було проаналізовано предметну область обраної теми, розібрано як саме працюють web-додатки та проаналізовано додатки-аналоги, в яких були виявлені, як недоліки так і переваги, і останні були застосовані у дипломному проекті, а саме: новітній дизайн, розташування товарів посередині сторінки.

Для реалізації всіх завдань даного проекту було також проаналізовано й різні технології для реалізації, адже саме їх правильний підбір допомагає краще освоювати нові знання та полегшую написання коду. Так, у третьому розділі було описано покроково процес реалізації дипломного проекту. А саме те, як було створено макет для додатку, його верстка, а також додавання різних функцій, які б полегшили використання додатку для користувачів, а також для полегшення його розуміння.

Цей проект має перспективу, як серед розробників, так і серед користувачів адже його написано на мові програмування, яка постійно оновлюється та легка для вивчення і розуміння, зі сторони користувачів можна розширювати перелік товарів в додатку, доповнювати його функціонал, щоб він був більш сучасним та швидшим у використанні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HTML [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/HTML>
2. HTML - елементи [Електронний ресурс] – Режим доступу: <https://html5book.ru/html-tags/>
3. Who is narattive designer [Електронний ресурс] – Режим доступу: <https://www.nicelysaid.net/what-is-narrative-design/#:~:text=Simply%20put%2C%20narrative%20design%20is,and%20their%20experience%20is%20interactive.>
4. Особливості веб-додатків [Електронний ресурс] – Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1191.ukr.html>
5. ux-journal [Електронний ресурс] – Режим доступу: <https://ux-journal.ru/chtotakoeuxuidizajniakpopastveti-professii.html>
6. Джон Дакетт./ HTML і CSS. Розробка і дизайн веб-сайтів; [пер. з англ. М.А.Райтман]. – М.:Ексмо, 2013. – 480с. – (Світовий комп'ютерний бестселер).
7. Фрімен Е., Робсон Е./ Вивчення програмування на JavaScript. – СПб.: Пітер, 2018. – 640с.(Серія «Head First O`Reilly»).
8. Video Game Design, Computer Science [Електронний ресурс] – Режим доступу: <https://www.computerscience.org/careers/video-game-designer/>
9. Крокфорд Дуглас./ Как устроен JavaScript. – Спбю:Київ,2019. – 304 с.-(Серія «Для професіоналів»)
10. Design for Apps [Електронний ресурс] – Режим доступу: <https://www.blog.udonis.co/mobile-marketing/mobile-design>
11. Visual Studio Code [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/Visual_Studio_Code#F
12. Популярні Web-сервіси: практика використання. Айверсон Уилл.– 2005.

13. Гультяев А.К. Проектування і дизайн інтерфейсу користувача – Київ: Еко-принт, 2010. – 349 с.
14. Фельке-Моррис Т./ Велика книга веб-дизайну; пер. с англ. Н.А.Райтмана.= М.: Эксмо, 2012.- 608с.
15. Simon J.D. Prince Computer Vision: Models, Learning, and Inference / J.D. Simon Prince. – London: Cambridge University Press, 2017. – 598 s.: – Bibliogr.: s. 19-25.
16. Никсон Р./ Створюємо динамічні веб-додатки завдяки JavaScript, CSS, HTML 4 – Київ, 2016 – 788с.
17. CSS [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/CSS>
18. JavaScript [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>
19. Адам Крут/ Coding HTML CSS JavaScript Made Easy. Web, Apps and Desktop – Київ, 2016 – 788с