

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Медіасистем та технологій
Рівень вищої освіти перший (бакалаврський)
Спеціальність 186 Видавництво та поліграфія
Тип програми Освітньо-професійна
Освітня програма Видавничо-поліграфічна справа
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри МСТ _____
(підпис)
« 23 » травня 2022 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові Дорофєєвій Катерині Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка платформи для проведення онлайн-виставки

Затверджена наказом по університету від 21 травня 2022 р. № 558 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20 червня 2022 р.

3. Вихідні дані до роботи

Тип електронного видання – веб-сайт;

Вид сайту – веб-сервіс;

Засіб розповсюдження – мережа Інтернет;

Вид інформації у виданні – текстова, графічна.


4. Перелік питань, що потрібно опрацювати в роботі

Вступ; 1 Аналіз завдання на кваліфікаційну роботу; 2 Аналітичний огляд досягнень у проектуванні та застосуванні електронних видань; 3 Вибір інструментальних засобів розробки сайту; 4 Проектування інформаційної структури і створення навігації сайту; 5 Розробка графічного дизайну та верстка веб-сайту; 6 Розробка серверної частини веб-сайту 7 Тестування сайту; 8 Економічна частина; Висновки; Перелік посилань; Додатки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Мета, задачі, актуальність; Етапи проектування сайту; Вибір програмних та інструментальних засобів розробки; Структура веб-платформи; Розробка графічного дизайну та верстка веб-сайту; Серверна частина; Тестування та публікація веб-сайту; Плакат з економічної частини.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	ст. викл. Чеботарьова І.Б.		18.06.2022
Економічна частина	проф. Полозова Т.В.		19.06.2022

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Дослідження сучасних досягнень у розробці та використанні сайтів	17.05.22	виконано
2	Аналіз завдання на розробку web-платформи	18.05.22	виконано
3	Вибір програмних та інструментальних засобів розробки	20.05.22	виконано
4	Проектування інформаційної структури веб-сайту	26.05.22	виконано
5	Проектування навігації сайту	30.05.22	виконано
6	Розробка графічного дизайну	01.06.22	виконано
7	Розробка серверної частини	10.06.22	
8	Верстка веб-сайту	12.05.22	виконано
9	Тестування та публікація сайту	15.05.22	виконано
10	Економічна частина	17.05.22	виконано
11	Оформлення пояснювальної записки	19.06.22	виконано
12	Оформлення графічної частини	19.06.22	виконано


Дата видачі завдання 23 травня 2022 року

Студент


(підпис)

Дорофєєва К.С.

Керівник роботи


(підпис)

ст. викл. Чеботарьова І.Б.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 57 с., 15 рис., 4 табл., 1 дод., 25 джерел.

ВЕБ-ПЛАТФОРМА, САЙТ ДЛЯ ОПИТУВАНЬ, ОНЛАЙН-ВИСТАВКИ,
ВЕБ-ДИЗАЙН, JS, REACT, CSS.

Метою кваліфікаційної роботи бакалавра є проектування й розробка web-платформи, призначеної для проведення онлайн-виставок з можливістю розміщення мультимедійних файлів та проведення опитування.

Об'єкт дослідження – технологія проектування інформаційних ресурсів з можливістю адміністрування.

В ході виконання кваліфікаційної роботи бакалавра були розроблені інформаційна структура, модульні сітки й макети сайту, розроблена навігація сайту, зверстані сторінки сайту й адміністративної панелі, розроблена вся відповідна функціональність згідно до вимог, пропонованих до web-платформи онлайн-виставки.

В процесі розробки було виконано тестування сайту в різних популярних браузерях і на пристроях з різним дозволом екрану. Результатом є готовий і протестований сайт.

Також зроблено економічне обґрунтування проєкту та розрахована ціна сайту.

ABSTRACT

The explanatory note contains: 57 p., 15 pic., 4 tabl., 1 app., 25 sources.

WEB PLATFORM, SURVEY SITE, ONLINE EXHIBITIONS, WEB DESIGN, JS, REACT, CSS.

The purpose of the bachelor's qualification work is to design and develop a web-platform designed for online exhibitions with the ability to place multimedia files and conduct surveys.

Object of research – technology of designing information resources with the possibility of administration.

During the bachelor's qualification work the information structure, modular grids and site layouts were developed, site navigation was developed, site and administrative panel pages were turned over, all relevant functionality was developed in accordance with the requirements of the online exhibition web-platform.

During the development process, the site was tested in various popular browsers and on devices with different screen resolutions. The result is a ready and tested site.

The economic substantiation of the project and the calculated price of the site are also made.

ЗМІСТ

	С.
ВСТУП.....	7
1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Мета та задачі web-платформи	9
1.2 Аналіз існуючих рішень.....	10
2 АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ У ПРОЄКТУВАННІ ТА ЗАСТОСУВАННІ ЕЛЕКТРОННИХ ВИДАНЬ	14
3 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ САЙТУ	19
3.1 Вибір редактора коду	19
3.2 Вибір технологій для фронт-енд.....	22
3.2.1 Використання бібліотеки	22
3.2.2 Використання каскадних таблиць стилів	27
3.3 Вибір технологій для бек-енд.....	27
3.3.1 Використання фреймворку	28
3.3.2 Використання баз даних.....	29
4 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ТА СТВОРЕННЯ НАВІГАЦІЇ САЙТУ.....	31
5 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ І ВЕРСТКА ВЕБ-САЙТУ	34
5.1 Використання принципів чуйного дизайну	34
5.2 Застосування стилів.....	37
6 РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ВЕБ-САЙТУ	41
7 ТЕСТУВАННЯ САЙТУ	44
8 ЕКОНОМІЧНА ЧАСТИНА	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
ДОДАТОК А CSS код сайту.....	Error! Bookmark not defined.

ВСТУП

Складно переоцінити значення Інтернету у сучасному світі, адже саме завдяки йому люди отримали чудові умови для навчання, роботи та, звичайно, розваг. Саме в мережі можна знайти неосяжну кількість корисної та цікавої інформації, поспілкуватися з друзями у соціальних мережах, знайти хороший фільм чи музичні композиції.

У наш час кількість сайтів збільшується з неймовірною швидкістю, саме тому головним завданням, що ставиться при розробці сайту – зробити його максимально привабливим для потенційних користувачів та надати йому індивідуальності. Ще 15 років тому можна було залучити досить велику кількість відвідувачів на сайт, не прикладаючи для його створення великих зусиль і без особливих матеріальних витрат. Зараз же, в умовах зростання конкуренції, функціональність сайту, його стиль і запам'ятовуваність грають дуже велику роль. Тому для досягнення успіху важливим є правильний підхід до вибору стратегії та виконавців.

Метою роботи є розробка web-платформи, призначеної для проведення онлайн-виставок з можливістю розміщення мультимедійних файлів та проведення опитування. Необхідність цієї розробки виникла у зв'язку з тим, що починаючи з 2016 року кафедра «Медіасистеми та технології» Харківського національного університету щорічно проводить міжнародну науково-технічну конференцію «Поліграфічні, мультимедійні та web-технології», в рамках якої проводиться виставка студентських робіт учасників конференції. Але з 2020 року пандемія внесла свої коригування – виставка проводиться онлайн. Сайт конференції не підтримує функції розміщення та опитування, тому реалізувати цей захід на базі існуючого сайту не має можливості. Все це і визначило актуальність кваліфікаційної роботи бакалавра – розробка web-платформи для проведення виставок.

Перед тим як приступити до створення практичної частини, необхідно чітко виділити основні цілі, функції, завдання та аудиторію сайту. Для

виявлення недоліків та ще невирішених проблем на ринку необхідно виконано аналіз існуючих рішень. Це допоможе реалізувати цікавий сайт з оптимальною функціональністю.

Оскільки метою є створення сайту для розміщення, перегляду та участі у голосуваннях, було виділено основні програмні компоненти, які необхідні для виконання завдань сайту, а саме:

- можливість створення опитувань;
- підтримку завантаження графічних матеріалів (картинки різних форматів, відео);
- попередній перегляд поданого матеріалу перед публікацією опитування;
- функцію зупинки голосування та перегляду результатів;
- можливість перегляду існуючих опитувань з панелі;
- сторінку з більш детальною інформацією щодо опитування з можливістю завантаження на пристрій розміщених файлів;
- функцію участі в опитуванні.

При виконанні практичної частини увага була спрямована на декілька аспектів. По-перше це інформативність, текстовий матеріал був окремо проілюстрований з метою полегшення сприйняття тексту. До проілюстрованих матеріалів відносяться не лише картинки, а й схеми та скріншоти. По-друге це сучасність та адаптивність під мобільні пристрої різних форматів, адже зараз неможливо уявити популярний сайт який неможливо використовувати з телефону. По-третє були враховані правила щодо користування сайтом за допомогою клавіатури.

Отже, на сьогодні немає зручного сайту для проведення онлайн-виставок, є тільки загально направлені платформи для створення опитувань. Саме тому проектування та розробка подібного сервісу є дуже актуальною.

1 АНАЛІЗ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Мета та задачі web-платформи

Після появи пандемії та декількох періодів вимушеної ізоляції у суспільства з'явилися нові потреби до мережі Інтернет. Якщо раніше основною діяльністю людей в Інтернеті були розваги, то зараз користувачам потрібно замінити основні види діяльності на їх онлайн-альтернативи. Таким чином похід до супермаркету замінили сервіси «BoltFood» та «Uklon», похід до кіно поступився місцем платформами для стрімінгу, ранкову зустріч в офісі з колегами замінив ранковий стендап у Скайпі. Багато галузей вже знайшли свої альтернативи, а ось мистецтво все ще поступається у цьому. Наразі не існує сервісу який намагався б змінити підхід до проведення виставок. Саме тому метою роботи було обрано створення сайту для проведення онлайн-виставок.

У кваліфікаційній роботі бакалавра необхідно створити вузько спрямований сервіс для розміщення онлайн-виставок, з можливістю створення онлайн-голосувань.

Мета роботи – розробка web-платформи, призначеної для проведення онлайн-виставок з можливістю розміщення мультимедійних файлів та проведення опитування.

Розроблений додаток повинен реалізувати наступні функції:

- можливість створення опитувань;
- підтримку завантаження графічних матеріалів (картинки різних форматів, відео, презентації);
- попередній перегляд поданого матеріалу перед публікацією опитування;
- функцію зупинки голосування та перегляду результатів;
- можливість перегляду існуючих опитувань з функцією пошуку та сортування;

– підтримку сторінки з більш детальною інформацією щодо опитування з можливістю завантаження на пристрій розміщених файлів.

Основною функцією, яка повинна бути реалізована, є можливість проведення оцінювання глядачами експонатів онлайн-виставки. Тобто опитування з фіксацією результатів.

Веб-сайт повинен бути виконаний з урахуванням адаптивності під різні пристрої. Необхідно врахувати і той факт, що сайт буде розглядатися з різних браузерів, тому потрібно провести тестування на широко-використовуваних браузерах.

Веб-платформа повинна складатися з про-панелі та головного порталу. Про-панель повинна мати функцію створення голосування з нуля, копіювання вже існуючого голосування, вибору типу голосування, загрузку графічних файлів, перед погляд на сторінку перед опублікуванням опитування. На головній сторінці повинні бути відображені 8-10 останніх доданих робіт. При переході на кожен з них повинна відображатися актуальна інформація про роботу. Сторінка голосування повинна містити актуальну інформацію, мати функцію закриття активного голосування, та мати можливість редагування голосування. Головний портал повинен реалізувати функції необхідні для участі в опитуванні. А саме сторінку для перегляду та участі.

1.2 Аналіз існуючих рішень

Однією з найголовніших вимог до сучасних веб-додатків є їх адаптивність та спрямованість на використання за допомогою скрін-рідера [1]. Чим ретельніше опрацьовані всі дрібниці, тим вища якість сервісу. Веб-дизайн повинен відповідати спрямованості та основним завданням, що стоять перед ним. Сайт має бути простим у освоєнні, зрозумілим для відвідувачів. Також про якість сайту говорить робота системи навігації, швидкість завантаження сторінок, доступність інформації. Якщо користувачеві не

доводиться витратити час на пошук даних, якщо сторінки завантажуються миттєво, а всі посилання працюють бездоганно, сайт має гідну якість. Наведені критерії дозволяють оцінити якість сайту з позиції користувача. Їх урахування необхідно при створенні веб-ресурсу, який матиме корисні функції та цінний зміст. Такий сайт сподобається людям, й відповідно, матиме всі шанси на плідний розвиток.

Галузь додатків, спрямованих на створення і розміщення опитувань є досить поширеною. Більшість з таких веб-додатків безкоштовні та не відрізняються унікальним дизайном або функціональністю. Однією з характерних рис таких програм є неупорядковане та надлишкове відображення інформації, адже невелика кількість систем використовує системи аналітики вибору. Як правило, на головній сторінці відображається досить багато реклами, а невпорядкованість поданої інформації ускладнює пошук потрібної інформації, що руйнує перше враження та миттєво відлякує користувачів.

Деякі платформи розрізняють користувачів та надають обмежену функціональність в залежності від рівня доступу, у той час як не зареєстровані користувачі не можуть користуватись додатком. Все це вносить корективи у можливості функціоналу та у розуміння користувачем щодо його вживання.

На сьогоднішній день в мережі Інтернет існує велика кількість сайтів для розміщення опитувань. Список прикладів представлений в таблиці 1.1.

Таблиця 1.1 – Популярні сервіси

Веб-застосунок	Коментар
Poll-maker[2]	Безкоштовний ресурс, не розділяє користувачів
Doodle[3]	Сервіс з можливістю платної підписки
DirectPoll[4]	Безкоштовний ресурс, не розділяє користувачів
Pollunit[5]	Має декілька варіантів підписки

Далі розглянемо та порівняємо інформацію подану у таблиці 1.1.

Poll-maker [2] – це попри безліч переваг безкоштовний ресурс що дозволяє використання у вільному доступі. Має функцію додавання тексту та картинок в опитування. Головною перевагою та відмінною рисою цієї платформи є впорядкована інформація щодо зібраних голосів подана у вигляді графіків. Відсутня сторінка для перегляду вже створених голосувань, з цього можна зробити висновок що сайт спрямований на поширення тільки окремого матеріалу через посилання. Сервіс має зручну та продуману навігацію, досить якісний дизайн та задовольняє критерії доступності.

Doodle [3] – веб-додаток спрямований на платну підписку, незареєстрований користувач має досить обмежені можливості. Основною функцією цього сервісу є створення опитувань. Окрім цього користувачі з про-підпискою можуть влаштовувати відео-конференції. Сайт задовольняє умовам користування за допомогою клавіатури та має зручний для використання дизайн та навігацію.

DirectPoll [4] – аналог сервісу Poll-maker[2], але має досить багато недоліків. По-перше він не має можливості додавання графічних матеріалів в опитування, що досить обмежує його спрямованість та використання. По-друге має застарілий дизайн та не задовольняє вимогам щодо вживання за допомогою скрін-рідера, що у свою чергу зменшує показник його якості та робить неможливим його використання людям з обмеженими можливостями.

Pollunit [5] – веб-сайт, який має декілька варіантів підписки, включаючи безкоштовну. Має функцію додавання тексту, дат, файлів, таблиць та картинок в опитування, з можливістю перед поглядом на виставлений матеріал до опублікування та вибору як найкраще подати матеріал (таблиця або список). Веб-застосунок має сучасний дизайн та задовольняє умовам користування за допомогою клавіатури та має зручний для використання дизайн та навігацію.

Після аналізу існуючих рішень можна зробити висновок що ресурсів для створення опитувань, спрямованих на оцінку графічного матеріалу, досить невелика кількість. Більшість з них спрямована на платну підписку та

не дозволяє реалізувати бажані функції. Саме тому проектування та розробка веб-сайту для проведення виставок з можливістю голосування є актуальними. Перед розробником є мета у досягненні декількох цілей. По-перше це розробка сайту, який буде задовольняти вимогам сучасних браузерів та мати унікальний дизайн з добре продуманою навігацією та упорядкованою інформацією. По-друге проектування веб-додатку для залучення користувачів до мистецтва, а авторів до розвитку та отримання можливості представити себе та свою творчість.

Тобто необхідно розробити web-платформу, призначену для проведення онлайн-виставок, яка буде інтегрована на сайт міжнародної конференції «Поліграфічні, мультимедійні та web-технології» й використана для проведення виставки студентських робіт в наступному році.

Для реалізації поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні тенденції веб-дизайну;
- виконати аналіз аналогів;
- обґрунтувати вибір технологій при створенні подібних сайтів;
- розробити концепцію сайту;
- визначити структуру сайту;
- спроектувати серверну та клієнтську частину;
- розробити дизайн web-платформи та верстку;
- виконати тестування сайту;
- зробити економічне обґрунтування проєкту.

2 АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ У ПРОЄКТУВАННІ ТА ЗАСТОСУВАННІ ЕЛЕКТРОННИХ ВИДАНЬ

Класифікація веб-сайтів.

Веб-сайти класифікують за функціями, які вони виконують. Вважають, що веб-сайти з найбільш гладким візуальним дизайном є найбільш складними з точки зору функціональності, але часто буває зворотне. Наприклад, гігант інтернет-торгівлі Amazon – не найкрасивіший сайт, але він розроблений для виконання великого діапазону функцій.

1. Інформаційні сайти.

Інформаційний сайт – досить великий віртуальний масив інформації, що включає безліч різноманітних тематичних розділів меншого розміру, або деяку кількість самостійних проєктів; є для клієнта основним джерелом інформації, що нагадує енциклопедію або спеціалізований журнал [6].

2. Корпоративні інформаційні web-сайти.

Корпоративний інформаційний web сайт – необхідний для автоматизації внутрішнього документообігу, обліку показників компанії, управління персоналом, може бути оснащений функціями обміну інформацією між віддаленими філіями; корпоративний сайт позитивно впливає на репутацію та імідж компанії; дизайн має відповідати фірмовому стилю компанії; найчастіше включає «адміністраторську частину» для створення та зміни контенту, що дозволяє менеджеру або секретареві компанії додавати або змінювати новини, інформаційні статті, довідкову та іншу інформацію на сайті; використовується мінімум графіки, основний акцент йде на текст [7].

3. Корпоративні іміджеві web-сайти.

Корпоративний іміджевий сайт – ідеально підходить для забезпечення іміджевої присутності в Мережі; сайт служить для надання докладної інформації про компанію, історію торгової марки, відомостей про надання послуг або товарів, що надаються; корпоративний сайт зазвичай містить

стрічку новин компанії, засоби публікації інформації про рекламні та торгові акції, інформацію для преси та інші відомості; нерідко корпоративні сайти поєднують інформацію про компанію із каталогом продукції; використовується ексклюзивний оригінальний дизайн, що вигідно представляє компанію; використання нестандартних ідей та рішень в оформленні, процентне співвідношення текст/графіка = 50/50.

4. Блоги.

Блог – це сайт, що представляє собою інтернет-щоденник, або журнал, який ведеться на кшталт стрічки новин, при цьому на кожен тему блогу відвідувачі можуть залишати свої коментарі-повідомлення, доступні з головної сторінки відповідної теми. Блог зазвичай дуже часто оновлюється, тому він може набагато частіше індексуватися пошуковими машинами, ніж звичайні сайти. Блог часто використовується для спілкування людей об'єднаних будь-якими спільними інтересами, і в цьому чимось нагадує форум [8].

5. Сайти-візитки.

Зазвичай складається з кількох сторінок і має унікальний, але простий та функціональний дизайн; ідеально підходить для компаній, які хочуть розмістити інформацію про себе та свої послуги в Інтернеті; основні розділи сайту: «Про компанію», «Продукція або послуги», «Прайс-листи», «Контактна інформація»; сайт-візитка використовується підприємствами, організаціями та приватними особами [9]. У сайта-візитки нестандартна структура, яка поділяє його на кілька розділів, в яких містяться дані про компанію, які продаються товари, послуги, що надаються, контактні дані. Виділяє кілька переваг таких «компактних» Інтернет-ресурсів:

- висока швидкість створення;
- змістовний і короткий виклад всієї інформації;
- залучення партнерів і клієнтів;
- невисока вартість;
- зручний і простий в управлінні.

6. Комерційні сайти.

Комерційні ресурси – одна з найбільш поширених категорій сайтів. Їх основне призначення – продаж товарів інтернет-користувачам. Комерційні сайти діляться на кілька підкатегорій, найбільш поширеними з яких є:

- сайт-вітрина;
- промо-сайти;
- інтернет магазини;
- сайти-візитки.

Сайт-вітрина – вид сайту, основним призначенням якого є не продаж готової продукції, а лише її реклама. Найбільшого поширення сайти візитки отримали серед компаній і підприємств, які виробляють товари. З його допомогою неможливо здійснити операцію купівлі-продажу, так як на сайті-вітрині тільки надається докладна інформація про товар. Але іноді, на них зазначаються місця, де можна купити зацікавив товар [10].

Промо-сайти – інтернет-ресурси, на яких користувачеві також рекламуються певні послуги або товари, але особливу увагу зупиняючи на їх перевагах. Вони характеризуються досить простою структурою, невеликим обсягом (до 10 сторінок) і нетривалого тривалістю життя. На сторінках сайту містяться різні графічні матеріали, контактні дані компанії і т. д. Якщо необхідно провести масштабну компанію певного товару – промо-сайт стане прекрасним рішенням проблеми [10]

Інтернет-магазини призначені продавати різні категорії товарів в мережі в режимі онлайн. Відвідавши такий сайт, кожен користувач всесвітньої павутини може вибрати що цікавить його товар, дізнатися про нього докладну інформацію і, в разі необхідності, здійснити замовлення не виходячи з дому і не відриваючись від комп'ютера. Дуже часто компанії пропонують користувачам всі свої лінійки товарів на одному сайті. Значно

рідше створюються кілька ресурсів, на яких представлена одна певна категорія поширюваної продукції.

Під каталогом розуміється сайт із вітриною з товарів, але без можливості додати їх до кошика та оформити замовлення. Купівля товарів з каталогу здійснюється через e-mail, телефон чи інші засоби зв'язку з власником ресурсу. Варто зауважити, що на даний момент сайт-каталог – це не найкраще рішення для бізнесу (виняток є каталоги з дорогими товарами, такими як спецтехніка, автомобілі і т.д.). Клієнтам нині потрібно, щоб було просто та зручно. А що простого і зручного в тому, що йому доводиться шукати можливість зв'язатися з власником каталогу, витратити час на спілкування з ними і тільки після проходження цього нелегкого шляху клієнт буде мати можливість отримати те, що йому потрібно. В цілому, це не найуспішніший вид сайту для розвитку онлайн-бізнесу.

Інтернет-магазини створюються будь-якими торговими організаціями, які здійснюють торговельну діяльність та прагнуть розширити коло потенційних покупців і можливості свого сервісу.

7. Веб-сервіси.

Сервіс зазвичай сконцентрований на вирішенні якоїсь однієї або декількох однотипних потреб клієнта. Найкращим прикладом будуть онлайн-консультанти. Напевно, кожен із нас зустрічав на сучасних сайтах спливаючі вікна зі зверненням менеджерів або пропозицією ввести свій номер, щоб нам швидко передзвонили та проконсультували. Так ось, такі онлайн-помічники і є прикладом сервісу – видом сайтів, які надають послуги із задоволення конкретної потреби користувача. Ті ж хмарні хостинги (Dropbox), відеохостинги (Youtube), різні послуги для онлайн-конвертації файлів і т.д. – є яскравим прикладом гарного та корисного сервісу. Сервісів в інтернеті існує безліч, щодня з'являються нові [11]. Для того, щоб створити свій успішний сервіс, потрібно лише зрозуміти, чого не вистачає людям. Згадайтеся, як вирішити чийось проблему, стати корисним для людей – і цей сервіс неодмінно чекає на успіх. Звичайно, за умови, що він буде

розроблений та введений в експлуатацію професіоналами. Оскільки хорошу ідею без грамотної технічної реалізації найчастіше очікує на провал.

Що ж вибрати за такої різноманітності видів сайтів. Вибір типу сайту насправді набагато простий, ніж здається. Все залежить від функціонального призначення. Тобто спочатку необхідно визначитися з метою сайту, а вже потім з його типом, структурою та функціями.

Якщо ви хочете продавати чи пропонувати послуги, у такому випадку інтернет-магазин або лендінг до ваших послуг. Також можна подивитися у бік каталогу. Але, слід зауважити, каталог набагато менш зручний порівняно з повноцінним інтернет-магазином, за винятком деяких видів бізнесу.

Якщо сайт покликаний інформувати клієнтів найбільш зручним та корисним буде корпоративний сайт або портал. За бажанням до цих типів сайтів можна додати форум або блог – це корисні інструменти у сфері підтримки та інформування користувачів.

Якщо ви збираєтеся надавати якусь конкретну послугу через інтернет, сервіс буде чудовим рішенням у цьому випадку. Хорошим рішенням буде суміщення лендингу та сайту-сервісу (лендінг – для хорошої конверсії, а сам сервіс буде в окремому електронному кабінеті користувача). Зручно та ефективно.

Варто зауважити, що класифікація сайтів – це річ дещо умовна. Як правило, більшість інтернет-ресурсів мають ознаки різних видів сайтів. Грамотне поєднання функціоналу, структури та дизайну інтернет-представництва – ось що справді важливо.

В кваліфікаційній роботі бакалавра розроблено сайт – web-сервіс, основна функція якого проведення опитування. Він призначений для проведення онлайн-виставки художніх робіт, тому повинен мати лаконічний дизайн, щоб не заважати оцінювати ці роботи, а також надавати повне адміністрування для розміщення робіт та підведення підсумків голосування.

3 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ САЙТУ

Інструментів для розробки сайту існує на сьогоднішній день велика кількість. В даному розділі необхідно підібрати оптимальні інструментальні засоби для front-end та back-end розробки.

3.1 Вибір редактора коду

Десяток-другий років тому веб-сайти робили за допомогою рук і звичайного текстового редактора. У міру того, як змінюється веб-розробка, з'являються нові фреймворки, плагіни та програми для оптимізації написання коду. Додаткові інструменти спрощують робоче середовище та підвищують продуктивність. Ми використовуємо системи контролю версій для спільного управління проектами з відкритим вихідним кодом та бібліотекою. Середовище розробки відіграє важливу роль у процесі створення сайту. Вибір ідеального текстового редактора може бути складним, але здійсненним завданням, яке вимагає застосування емпіричного дослідження.

Notepad++. Найпростіший текстовий редактор з підсвічуванням синтаксису та вкладками. Підтримує безліч мов програмування та технологій. Відмінно підійде фахівцю-початківцю.

Основні функції:

- створення закладок;
- автоматичне копіювання та збереження створених файлів;
- підтримка та конвертація стандартних кодувань тексту;
- користувачі можуть самостійно встановити правила підсвічування синтаксису;
- підтримка та конвертація стандартних кодувань тексту;
- користувачі можуть самостійно встановити правила підсвічування синтаксису;
- встановлення різних корисних плагінів;

- налаштування та перевизначення «гарячих» клавіш;
- перевірка орфографії та шифрування тексту;
- може керувати більшими обсягами даних.

Brackets. У Adobe є багато професійних продуктів, які використовуються Frontend-розробниками та дизайнерами. Brackets – один із них. Цей безкоштовний кросплатформовий редактор з відкритим вихідним кодом базується на JavaScript, HTML5 та CSS3, тому в роботі орієнтований саме на них. Якщо у вас є базове розуміння JavaScript, саме з цього редактора потрібно починати роботу. Тому що саме він надасть усі необхідні інструменти для розширення. Редактор потребує деяких доопрацювань, але в цілому досить активно розвивається і користується популярністю у розробників. Причому кожен користувач може внести свої зміни та створити додаткові розширення.

Основні функції:

- підказки при редагуванні JS, CSS та HTML-файлів;
- підтримка перегляду коду в браузері реального часу (Live Preview);
- синхронізація з FTP-сервером;
- підтримка та автоматична компіляція CSS до less-файлів;
- надає вибір шрифтів (з можливістю їх доповнення), які можна використовувати під час роботи з текстом;
- робота із системою контролю версій – Git.

Visual Studio Code. Visual Studio Code це редактор вихідного коду, що вже традиційно лідирує в рейтингу популярності розробників. Він створений компанією Microsoft і доступний для Windows, Linux та Mac OS. Цей редактор простий у використанні та суттєво прискорює процес кодингу. Серед його додаткових переваг варто відзначити велику спільноту та відмінну підтримку.

Основні особливості:

- автодоповнення (Intellisense);
- імпортовані модулі;

- інтеграція з GitHub;
- широкий вибір тем;
- відмінні вбудовані шаблони;
- інтегровані інструменти для тестування коду;
- доступна система контролю версій.

WebStorm. Це не звичайний текстовий редактор, а інтегроване середовище розробки JavaScript, CSS & HTML. Її автори – компанія JetBrains, та сама, що створила улюблене студентами JavaRush середовище IntelliJ IDEA. WebStorm є одним з найрозумніших і найпопулярніших редакторів. Підтримує основні фреймворки: AngularJS, React та Meteor та служить для створення програми на Node.js. WebStorm досить складне середовище, але при цьому воно відмінно підходить тим, кому необхідно безліч розширень.

Основні особливості:

- ефективно працює з проектами різного розміру завдяки інтеграції із системою контролю версій Git, GitHub, Mercurial. Це дозволяє «комітіти» файли, переглядати зміни і вирішувати конфлікти прямо в IDE;
- LiveEdit – показує попередні зміни без перезавантаження сторінки (працює з браузером Google Chrome);
- аналіз коду в процесі редагування;
- рефакторинг та налагодження коду на Javascript та Node.js додатків;
- підтримка мов ECMAScript, TypeScript, CoffeeScript та Dart;
- створення програм за допомогою шаблону Node.js Express;
- запуск та налагодження з фреймворками на зразок Mocha, Karma та Protractor. Тестувати можна в самому редакторі або в зручному деревоподібному меню, з якого можна повернутися до тексту;
- повністю підтримуються препроцесори LESS, Sass, SCSS, Stylus;
- швидкий пошук у вікні терміналу;
- структуризація проектів, у яких задіяно кілька бібліотек Javascript;
- підтримка та робота з Angular 2.

Через популярність та найбільш зручними технічно виявились редактори Visual Studio Code та WebStorm. Через те що WebStorm немає безкоштовної можливості використання, було обрано Visual Studio Code як редактор коду.

3.2 Вибір технологій для фронт-енд

JavaScript залишається найпопулярнішою мовою програмування, якщо судити з результату опитування Stack Overflow [12] та дослідження від GitHub [13], саме тому вона була обрана для усього додатку.

3.2.1 Використання бібліотеки

Для досвідченого розробника немає істотної різниці в тому, який фреймворк використовувати, просто потрібен деякий час, щоб звикнути до нового. Кожен фреймворк має свої плюси і мінуси, а це означає, що при розробці продукту потрібно зробити правильний вибір для кожного окремого випадку. Щоб полегшити та прискорити цей процес було обрано два найпопулярніших фреймворка та виявлено загальні переваги та недоліки кожного з них.

Angular – це середовище JavaScript MVVM, засноване в 2009 році, яке відмінно підходить для створення інтерактивних веб-додатків.

Переваги Angular [14]:

- Angular використовується разом з Typescript [15], він має виняткову підтримку для цього;
- Angular-language-service забезпечує інтелектуальні можливості та автозаповнення шаблону HTML-компонента;
- нові функції, такі як generation Angular, що використовують бібліотеки npm з CLI, та розробка компонентів, що використовує Angular;

- детальна документація, що дозволяє розробнику отримати всю необхідну інформацію. Однак це потребує більше часу для навчання;
- одностороння прив'язка даних, що забезпечує виняткову поведінку програми, що зводить до мінімуму ризик можливих помилок;
- MVVM (Model-View-ViewModel), яка дозволяє розробникам працювати окремо над тим самим розділом програми, використовуючи той самий набір даних;
- впровадження залежностей від компонентів, пов'язаних з модулями та модульністю в цілому;
- структура та архітектура, спеціально створені для великої масштабованості проекту.

Недоліки:

- різноманітність різних структур (Injectables, Components, Pipes, Modules тощо) ускладнює вивчення порівняно з React і Vue.js, які мають тільки «Component»;
- досить повільна продуктивність з огляду на різні показники, однак це можна легко вирішити, використовуючи так званий ChangeDetectionStrategy, який допомагає вручну контролювати процес рендерингу компонентів.

Компанії, що використовують Angular: Microsoft, Autodesk, McDonald's, UPS, Cisco Solution Partner Program, AT&T, Apple, Adobe, GoPro, ProtonMail, Clarity Design System, Upwork, Freelancer, Udemy, YouTube, Paypal, Nike, Google, Telegram, Weather, iStockphoto, AWS, Crunchbase.

React– це бібліотека JavaScript, розроблена Facebook у 2013 році, яка відмінно підходить для створення сучасних додатків будь-якого розміру та масштабу.

Переваги React [16]:

- легко вивчити завдяки простому дизайну використанню JSX (HTML-подібний синтаксис) для шаблонів і дуже докладної документації.

Розробники витрачають більше часу на написання сучасного JavaScript і менше турбуються про код, специфічний для фреймворку;

- дуже швидка завдяки реалізації React Virtual DOM і різним оптимізаціям рендерингу;

- відмінна підтримка рендерингу на стороні сервера, що робить його потужною платформою для контент-орієнтованих програм;

- першокласна підтримка Progressive Web App (PWA) завдяки генератору додатків «create-react-app»;

- прив'язка даних є односторонньою, що означає менше небажаних побічних ефектів;

- Redux [17], найпопулярніша платформа для управління станом додатків у React, її легко вчити та використовувати;

- React реалізує концепції функціонального програмування (FP), створюючи простий у тестуванні та дозволяє багаторазово використовувати код;

- програми можуть бути створені за допомогою TypeScript або Facebook Flow, що мають вбудовану підтримку JSX;

- перехід між версіями, як правило, дуже простий: Facebook надає кодові модулі для автоматизації більшої частини процесу;

- навички, отримані React, можуть бути застосовані до розробки на React Native.

Недоліки React – змішування шаблонів з логікою (JSX) може спантеличити деяких розробників при перших знайомствах з React.

Компанії, що використовують React: Facebook, Instagram, Netflix, New York Times, Yahoo, Khan Academy, Whatsapp, Codecademy, Dropbox, Airbnb, Asana, Atlassian, Intercom, Microsoft, Slack, Storybook та багато інших.

Треба зазначити що недоліків у React виявилось значно менше, тому саме його було обрано для розробки цуи-платформи для опитування.

Для того щоб використовувати React, треба спочатку установити npx [18] через термінал за допомогою команди 'npm і npx', якщо ще немає npm на машині то потрібно скористатися необхідними інструкціями для установки з

сайту Node.js [19]. Тепер можна починати налаштування проекту, для цього потрібно використати команду 'npx create-react-app golos', де golos це назва папки проекту, після чого буде автоматично створена структура проекту, яка представлена на рисунку 3.1.

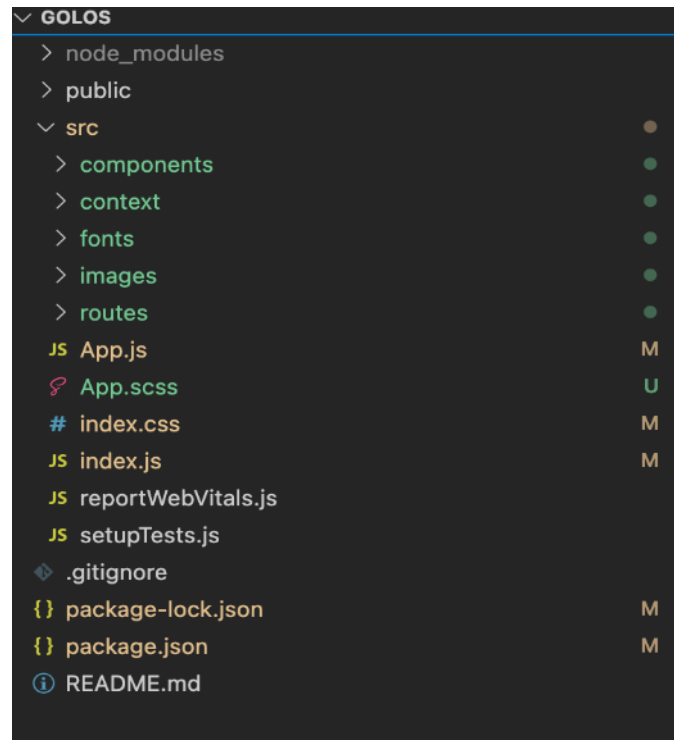


Рисунок 3.1 – Структура проекту

Розглянемо структуру проекту більш детально.

По-перше та найголовніше що містить проєкт – це файли налаштувань, такі як README.md, package.json, package-lock.json, .gitignore.

По-друге треба зазначити що існує дві високорівневі папки: public та src. Public це папка яка автоматично генерується при процесі build у проєкта. Src це папка, в якій розробник працює, додає елементи, навігацію та стилі. Вона є основою для компіляції, та містить 5 додаткових папок. Перша з них це components, вона складається з різних компонентів, які будуть використовуватись у сайті. Серед таких компонентів можна виділити кнопку, табку, інпут, дропдаун та інші. Слід зазначити що компоненти повинні бути досить гнучкими, щоб мати можливість використання у різних частинах

додатку. Друга папка це context, яка містить обгортку для передачі даних між різними компонентами (рис.3.2).

```
src > context > ApplicationContext.jsx > ApplicationProvider
1  import React, { useState, useContext } from 'react';
2
3  > const setToolsWrapper = (setContext) => ({-
9  });
10
11 export const ApplicationProvider = ({ children }) => {
12   const [context, updateContext] = useState({
13     newPoll: {},
14   });
15
16   const handleUpdateContext = (data) => updateContext((prevContext) => ({
17     ...prevContext, ...data
18   }));
19
20   const optimizedSetTools = React.useCallback(setToolsWrapper(updateContext), [updateContext]);
21
22 > const toolsState = React.useMemo(() => ({-
25
26 > React.useEffect(() => {-
34
35   return (
36     <ApplicationContext.Provider value={toolsState}>
37       {children}
38     </ApplicationContext.Provider>
39   );
40 };
41
42 > const ApplicationContext = React.createContext({-
45
46 export const useApplicationContext = () => useContext(ApplicationContext);
47
```

Рисунок 3.2 – Розроблена обгортка контексту

Папка fonts містить файли з розширенням .otf, що містять необхідні шрифти. Окрім шрифтів папка містить файл, що робить їх доступними у всіх файлах стилів (рис.3.3).

```
@font-face {
  font-family: 'Gill Sans Bold Italic';
  src: local('Gill Sans Bold Italic'), local('Gill Sans Bold Italic'),
  url('./Gill Sans Bold Italic.otf') format('otf');
  font-style: normal;
}

@font-face {
  font-family: 'Gill Sans Bold';
  src: local('Gill Sans Bold'), local('Gill Sans Bold'),
  url('./Gill Sans Bold.otf') format('otf');
  font-style: normal;
}

@font-face {
  font-family: 'Gill Sans Heavy';
  src: local('Gill Sans Heavy'), local('Gill Sans Heavy'),
  url('./Gill Sans Heavy.otf') format('otf');
  font-style: normal;
}

@font-face {
  font-family: 'Gill Sans Italic';
  src: local('Gill Sans Italic'), local('Gill Sans Italic'),
  url('./Gill Sans Italic.otf') format('otf');
  font-style: normal;
}

@font-face {
  font-family: 'Gill Sans Light Italic';
  src: local('Gill Sans Light Italic'), local('Gill Sans Light Italic'),
  url('./Gill Sans Light Italic.otf') format('otf');
  font-style: normal;
}
```

Рисунок 3.3 – Налаштування шрифтів

Папка `images` містить необхідні статичні графічні матеріали.

Папка `routes` необхідна для зазначення роутів додатка.

3.2.2 Використання каскадних таблиць стилів

Написання CSS – це основа з якої починають усі розробники, але вже досвідчені інженери замислюються о проблемах та як їх вирішувати. Коли таблиця стилів стає величезною, її стає складно обслуговувати, у такому випадку препроцесори стають у нагоді. Одним з таких інструментів і є Sass. Він дозволяє використовувати функції недоступні в самому CSS, наприклад змінні, вкладеності, міксини, успадкування та інші приємні речі, що повертають зручність у написання стилів. Як тільки ви починаєте користуватися Sass, він обробляє ваші файли стилів з розширенням `.scss` і зберігає його як простий CSS-файл, який вже використовується у браузері. Перш ніж використовувати Sass, необхідно його налаштувати у проекті. Для цього існує бібліотека `sass`. Як і всі бібліотеки її можна додати за допомогою пакетного менеджера, а саме за допомогою команди «`npm i sass`». Після чого для використання функціональності представленої пакетом Sass вам необхідно додавати розширення `.scss` кожному файлу стилів.

3.3 Вибір технологій для бек-енд

Будучи найпопулярнішою мовою програмування, JavaScript також є однією з найуніверсальніших технологій розробки програм, дозволяючи створювати якісні програми для багатьох платформ.

Так як було обрано саме JavaScript як основну мову розробки програмного забезпечення кваліфікаційної роботи, треба обрати найзручніший серед існуючих варіант виконання серверної частини.

3.3.1 Використання фреймворку

Серед js фреймворків існує не так багато варіантів для використання. Було виділено два фреймворки на які має сенс звернути увагу.

NodeJS. Насправді Node.js – це не фреймворк або бібліотека, а середовище виконання, яке базується на Chrome V8 JavaScript. Проаналізуємо сильні та слабкі сторони Node.js.

Переваги:

- швидка обробка запитів та ефективна подієва модель;
- ідеальний вибір для мікро-сервісної архітектури;
- багата екосистема;
- підтримка IT-гігантів;
- повна підтримка JSON.

Недоліки:

- низька продуктивність під час роботи з важкими обчислювальними завданнями;
- залежність від зворотного виклику.

Deno JS.

Це той самий Node.js, але з підтримкою експериментальних функцій, серед них:

- імпорт за URL;
- import maps;
- відмова від загального переліку залежностей.

Відрізнення Deno JS та Node.js можна віднести як до переваг як і до недоліків залежно від використання. Deno JS все ще має слабку підтримку серед ком'юніті що є дуже залежним недоліком. Так як він заснований на тій самій основі як і Node.js було вирішено віддати перевагу більш стабільному – а саме Node.js.

Для використання Node.js не потрібно виконувати додаткові дії. Все що потрібно, то мати його встановленим на комп'ютері. Для того щоб створити новий проект було викликано команду 'npm init' та власноруч створено структуру папки (рис 3.4).

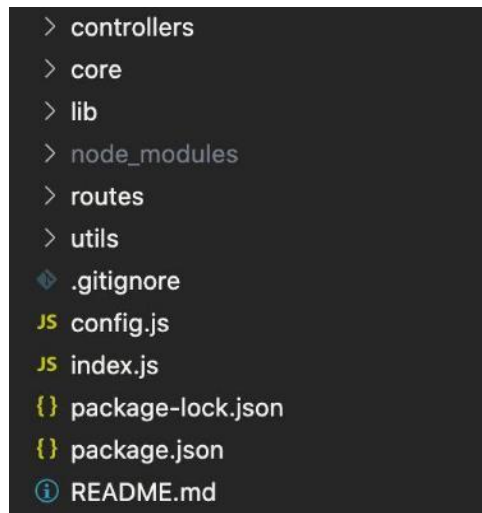


Рисунок 3.4 – Структура проекту

3.3.2 Використання баз даних

MySQL – одна з систем управління базами даних, що найбільш використовуються, саме тому її було обрано для використання.

MySQL управляє реляційними базами даних, тобто такими, у яких таблиці пов'язані між собою. MySQL працює за принципом клієнт-сервер. Комп'ютер користувача (клієнт) надсилає запит. Сервер баз даних його обробляє та надає відповідь. Ось тому часто можна почути поняття MySQL-сервер. Це сервер, де зберігається база даних. Система MySQL написана мовами програмування C та C++. Для роботи MySQL використовується мова структурованих запитів SQL.

SQL (Structured Query Language) – це мова програмування, за допомогою якої можна керувати інформацією: додавати, модифікувати, видаляти та отримувати дані. Запити до бази даних формуються мовою SQL. SQL використовується у MySQL. Багато СУБД (реляційні системи

управління базами даних) використовують цю мову для роботи з даними.

Наприклад: Microsoft SQL Server; PostgreSQL; Oracle Database; SQLite.

Серед СУБД було обрано PostgreSQL.

Структура таблиць БД представлена на рисунку 3.5.

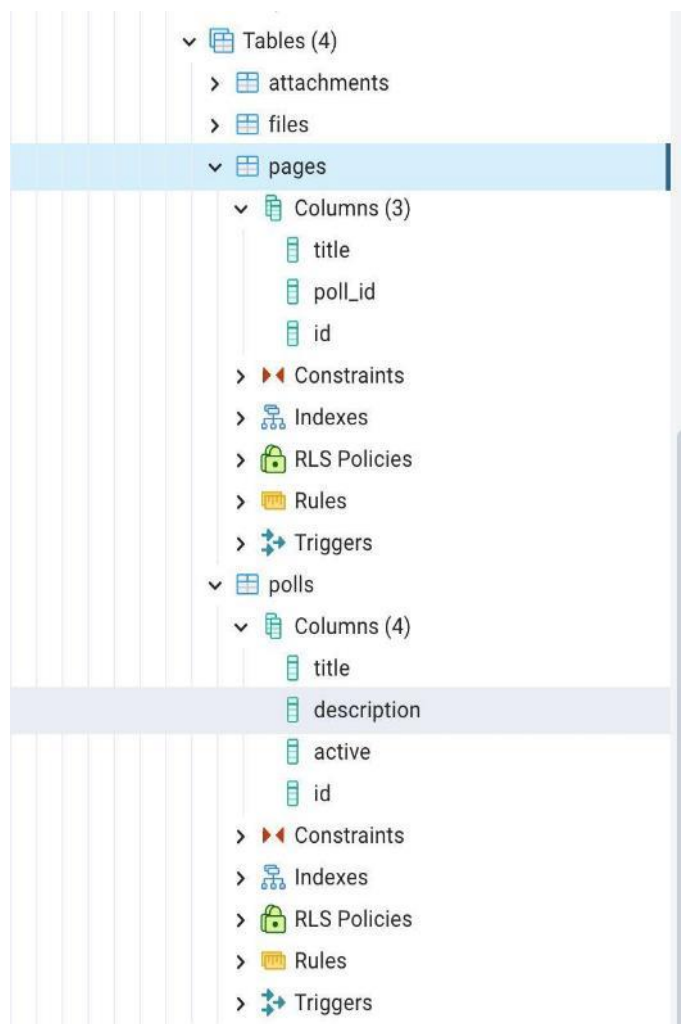


Рисунок 3.5 – Структура таблиць

4 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СТРУКТУРИ ТА СТВОРЕННЯ НАВІГАЦІЇ САЙТУ

З погляду SEO, кожен розроблений веб-сайт повинен мати картку. Карта дозволяє Google та іншим пошуковим системам краще сканувати, оскільки забезпечує сканери схемою змісту сайту. Дозволяючи сканерам пошукових систем краще сканувати сайт, можна підвищити рейтинг результатів і залучити більше трафіку. Це до тих пір, поки контент добре підготовлений та привабливий для веб-користувачів.

Схеми веб-сайтів важливі не тільки для пошукових машин, вони також можуть бути корисними для користувачів, які шукають певну сторінку. У той час як карти можуть бути корисні як пошукачам, так і користувачам, різні варіанти передбачають різні формати файлів: XML для пошукових систем та HTML для користувачів.

У даній роботі було визначено такі сторінки:

- головна сторінка;
- створення голосування;
- превью голосування;
- редагування голосування;
- сторінка для участі у голосуванні;
- про окремого учасника у голосуванні;
- вхід;
- лендінг сторінка.

На «Головній» сторінці розташовуються такі структурні блоки: додані опитування, навігація. За допомогою навігації є можливість перейти до перегляду голосів окремих опитувань, редагування опитувань, деактивації опитувань, створення опитувань. Головна сторінка доступна тільки авторизованим користувачам.

Створення голосування ділиться на кілька етапів.

Перший – це сторінка заповнення інформації про голосування.

Другий – це вибір типу голосування.

Третій – створення категорій для голосування та завантаження графічних матеріалів до кожної з них.

Четвертий – завантаження інформації про окремі роботи.

Сторінка перед погляду на голосування призначена для внесення остаточних корективів та не відрізняється від сторінки редагування.

Сторінка участі у голосуванні доступна усім та слугує єдиною можливою функціональністю для неавторизованих користувачів.

Сторінка «Вхід» дає користувачеві можливість залогінитися на сайті. Вхід користувача здійснюється за допомогою введення електронної пошти та пароля.

Лендінг сторінка виконує функції сторінки привітання з однією можливою функцією входу до порталу.

Схема навігації для не авторизованих користувачів представлена на рисунку 4.1.



Рисунок 4.1– Схема навігації для не авторизованих користувачів

Схема навігації авторизованих користувачів представлена на рис. 4.2.

В ході виконання даного розділу було виділено основні інформаційні блоки, спроектована навігація сайту, доступність цих сторінок для користувачів різних рівнів доступу.

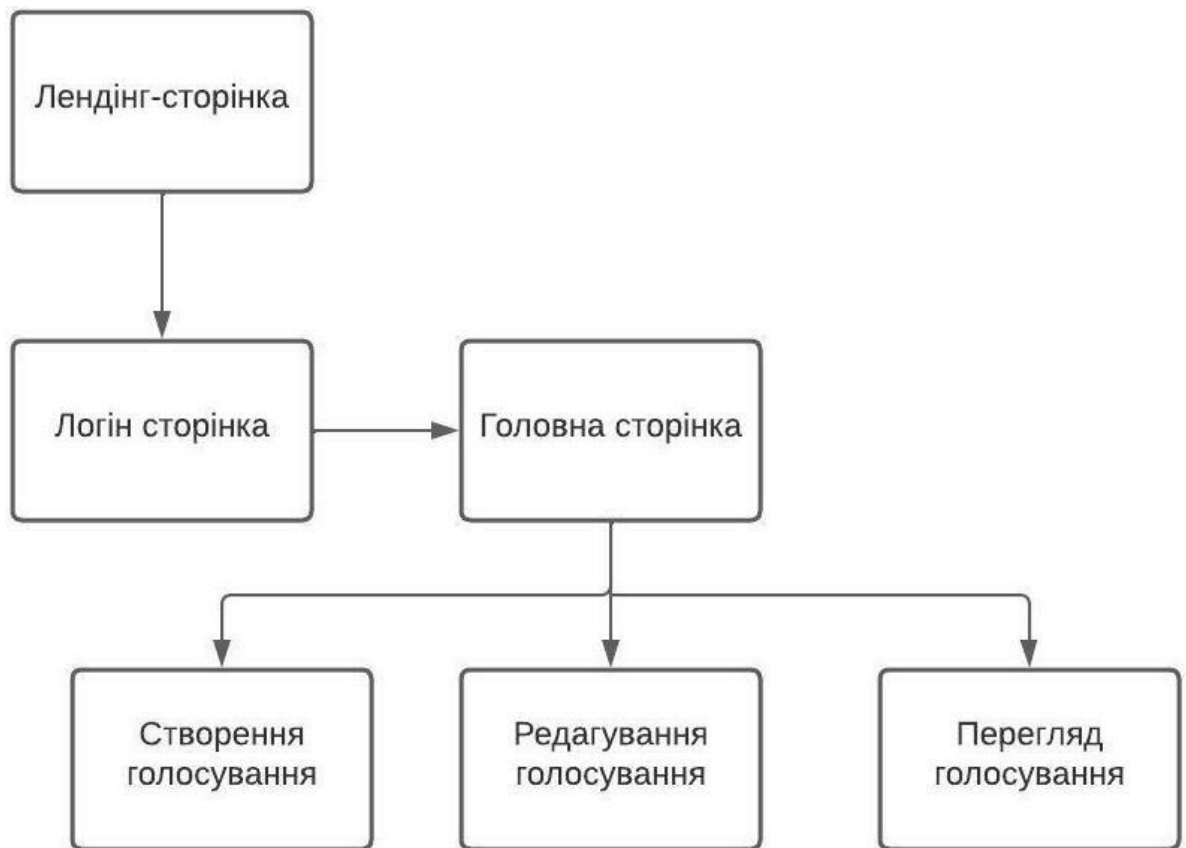


Рисунок 4.2 – Схема навігації для авторизованих користувачів

5 РОЗРОБКА ГРАФІЧНОГО ДИЗАЙНУ І ВЕРСТКА ВЕБ-САЙТУ

Під час розробки сайту важливим етапом є проектування та створення доступного для різних вікових груп користувачів веб-ресурсу. Маючи інформаційну структуру, потрібно розробити наочний інтерфейс.

Веб-дизайн включає графічний дизайн, дизайн взаємодій із користувачем, дизайн інтерфейсу, пошукову оптимізацію та створення контенту. Ці елементи визначають, як виглядає та працює сайт на різних пристроях.

Головною метою графічного дизайну є візуальний зв'язок з людиною за допомогою текстової та графічної інформації. Добре виконане візуальне подання дозволяє зосередити увагу з боку користувача, впливаючи насамперед на емоційну складову.

Саме через виділені характеристики дизайн посідає важливе місце у процесі розробки. З цієї точки зору сайт – це сукупність елементів єдиної системи. Для зв'язування всіх сторінок сайту було прийнято рішення використати єдину стилістику.

5.1 Використання принципів чуйного дизайну

Чуйний дизайн – це підхід, коли контент і макет веб- та мобільних додатків змінюється відповідно до поведінки користувача та середовища.

Загалом дизайн реагує на те, як користувач взаємодіє з ним. Це означає, що незалежно від розміру екрана, типу пристрою або платформи для читання вмісту, інформація буде добре відображатися на всіх носіях.

Оскільки користувачі, як правило, отримують доступ до вмісту різними способами на безлічі різних пристроїв, потреба у чуйному дизайні більша, ніж будь-коли.

Перед широким впровадженням чуйного дизайну дизайнер інтерфейсу користувача міг би створити тільки один макет веб-сайту, і він буде

запущений у виробництво і випущений у світ для всіх користувачів. Існували стандартні розміри, такі як 800 x 600 або 1024 x 786, до яких дотримувалися дизайнери. Якщо ви використовували більший екран для перегляду або менший, дизайн залишався б того ж розміру. Проблема такого підходу полягає у тому, що він передбачає, що один макет підходить всім пристроям.

Є багато елементів, які становлять чуйний веб-сайт, крім простої зміни розмірів. Інші елементи, які слід враховувати:

- типографіка;
- навігація;
- продуктивність.

«Веб-дизайн – це 95% типографіки» – Олівер Райхенштейн [20]. Оскільки типографіка відіграє таку важливу роль у дизайні веб-сайтів та мобільних додатків, має сенс, щоб шрифти також були адаптивними. Адаптивна типографіка змінюватиметься за розміром щодо розміру вікна перегляду.

Мобільні пристрої не дуже широкі, тому для забезпечення гарного інтерфейсу необхідно використовувати інші засоби відображення інформації, що економлять місце. Так наприклад з навігацією має сенс додавання бургер-меню, де будуть знаходитись посилання меню навігації.

Хоча багато елементів, які становлять чуйний дизайн, є наочними, існують також back-end елементи, що підвищують продуктивність адаптивного дизайну. Наприклад, якщо ви створили 15 окремих файлів HTML і CSS для кожного з ваших пристроїв, ви зіткнетеся з проблемами продуктивності. Кількість запитів HTTP від пристрою до сервера різко збільшить час завантаження. А повільний час завантаження може призвести до того, що на вашому сайті залишиться менше людей.

Приклад розроблених екранів наведено на рисунках 5.1 та 5.2.

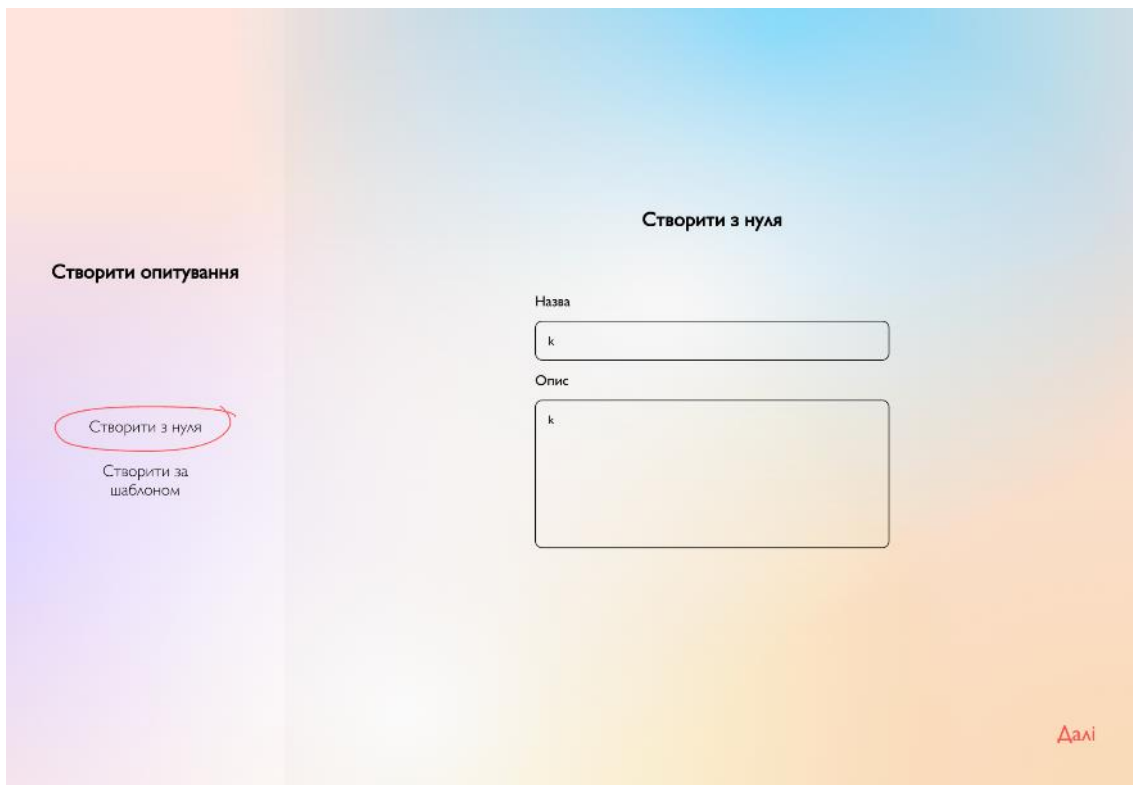


Рисунок 5.1 – Приклад екрану створення голосування (створити з нуля)

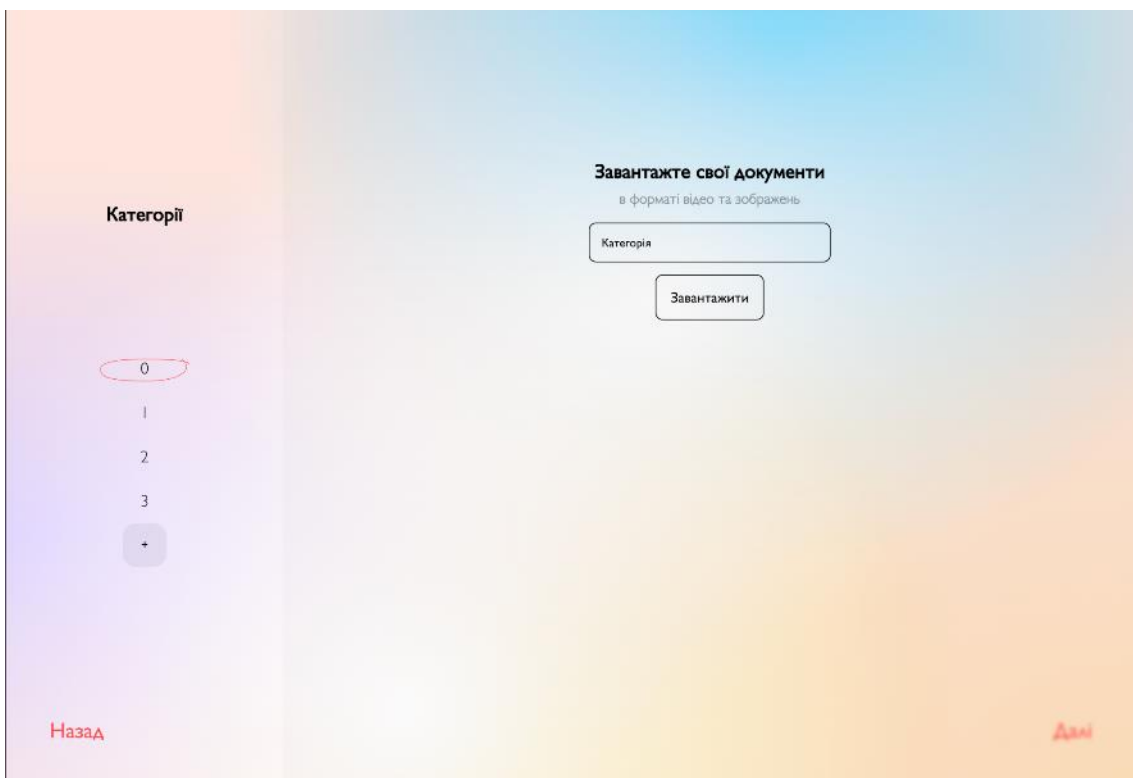


Рисунок 5.2 – Приклад екрану створення голосування (за категоріями)

5.2 Застосування стилів

В інтернеті існує чимало готових рішень, застосування яких спростить завдання і скоротить час розробки. Для досягнення цілей проекту було вирішено не використовувати готові рішення, щоб створити більш оптимальний варіант. Основні функції Scss, які використовувалися в проекті:

- змінні;
- наслідування;
- анімації;
- контрольні точки.

Загальні зміни розташування блоків були явно задані контрольними точками. Для досягнення цілей було обрано підхід Mobile-first.

Mobile-first – Концепція «Mobile First Design» розповідає про проектування спочатку для мобільних пристроїв, а вже потім для настільних пристроїв.

Застосування цієї технології дозволяє: отримувати користувачеві важливий зміст сторінки в першу чергу; забезпечувати зручний інтерфейс для навігації в мобільному екрані; підтримувати швидке завантаження сторінок при низькій швидкості підключення; споживати мінімальну кількість web-ресурсів для відображення основного вмісту, що веде до економії мобільного Internet трафіку [21]. Загалом це означає, що медіа-запити використовують мінімальну ширину. Ми використовуємо медіа-запити для додавання або перезапису стилів для встановленої точки зупину та більше. Приклад існуючих контрольних точок:

- 760 та більші;
- 1170 та більші.

Приклад медіа запитів наведено на рисунках 5.3 та 5.4.

Для розробки веб-сайту було обрано дванадцяти-стовпчикову сітку з відступами між елементами кратними чотирьом пікселям. Розроблена модульна сітка наведена на рисунку 5.4-5.5.



Рисунок 5.3 – Приклад екрану з розширенням мобільного телефону

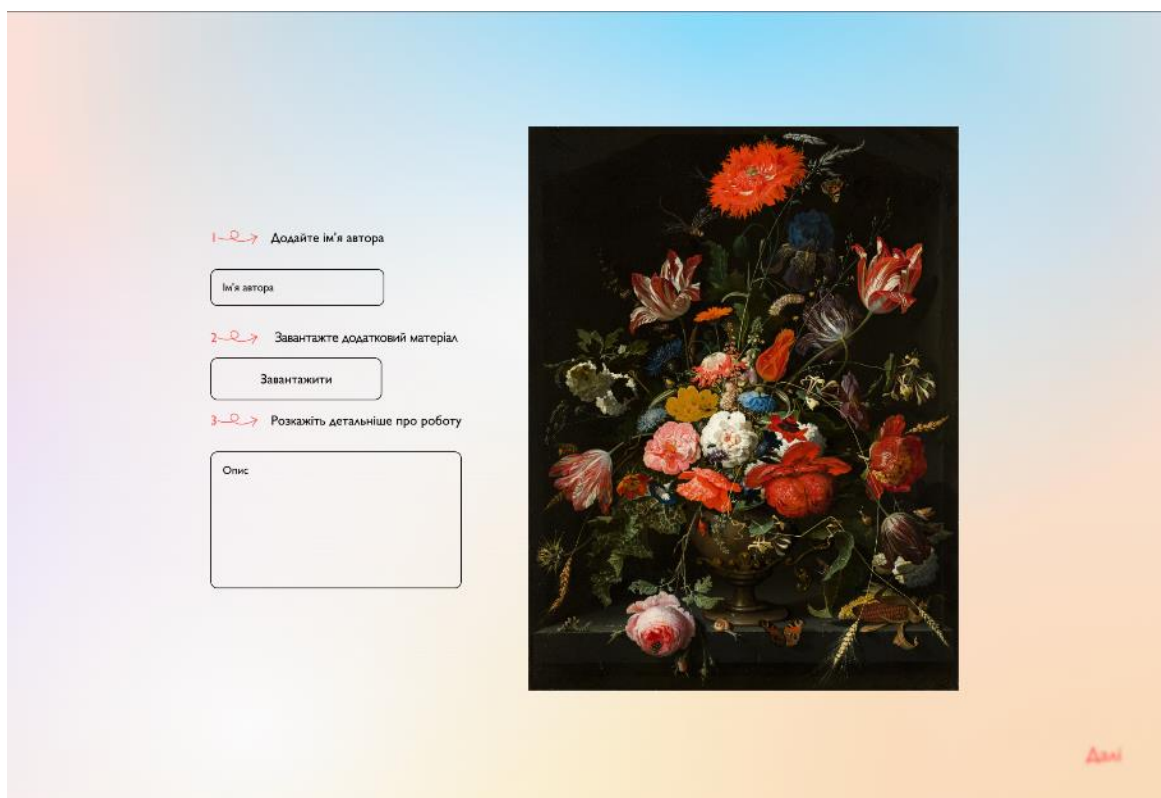


Рисунок 5.3 – Приклад екрану з розширенням екрану комп'ютера

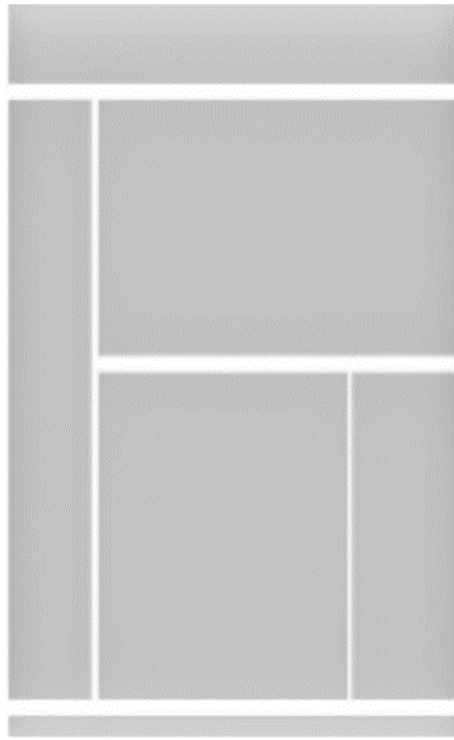


Рисунок 5.4 – Приклад модульної сітки для мобільного телефону

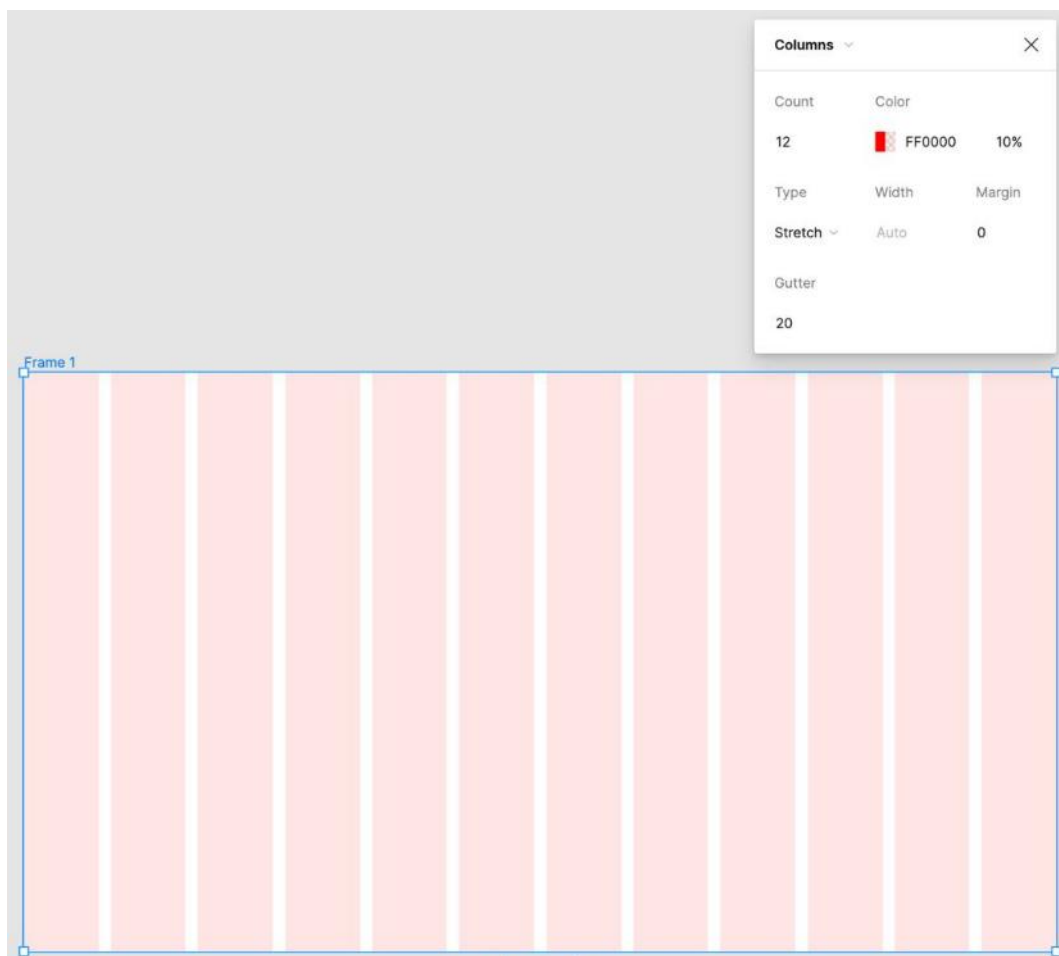


Рисунок 5.5 – Приклад модульної сітки для комп'ютера

В результаті виконання поданого розділу було обрано модульну сітку, розроблена система навігації а також здійснено верстку основних сторінок сайту. Верстка веб-ресурсу була розроблена без додавання сторонніх бібліотек. Дизайн адаптується залежно від розміру та типу екрана. Повторювані елементи, наприклад, навігаційна панель, мають однакове положення на всіх сторінках, що спрощує сприйняття структури сайту. Елементи які з'являються на декількох сторінках, наприклад, кнопки навігації, розташовані незалежно від контенту на одному місці всіх роутів.

6 РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ВЕБ-САЙТУ

Модель MVC – Model-view-controller рис. 6.1 («модель-вид-контролер») – схема використання декількох шаблонів проектування, за допомогою яких програмне забезпечення, клієнтський інтерфейс та клієнтська взаємодія поділені на три компоненти, так що зміна одного з компонентів не значно впливає на інші. Ця схема доволі часто використовується для побудови структурної системи при переході від гіпотези до використання в певному діапазоні. Веб-додаток, як правило, складається з набору контролерів, моделей і видів. Ідеєю цього ставлення є отримання ізольованих компонентів, тобто поділ відмінних можливостей.

MVC Architecture Pattern

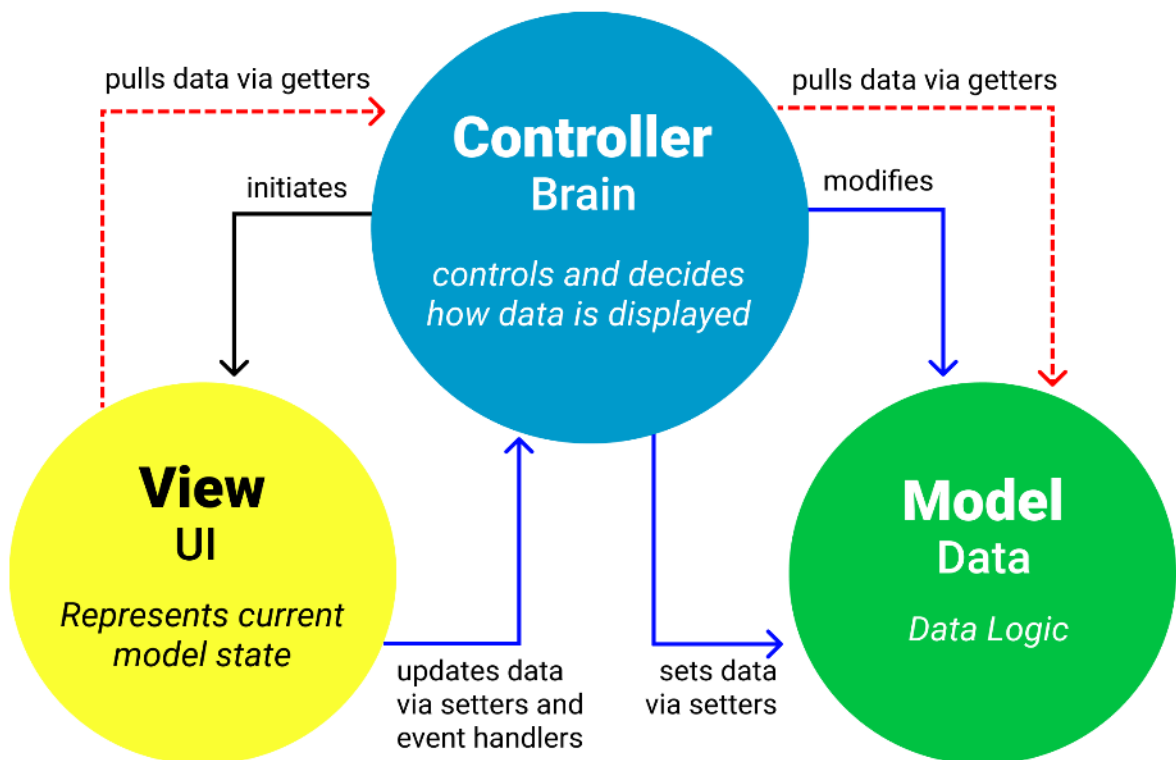


Рисунок 6.1 – MVC структура

Контролер (Controller) управляє запитом користувача (отримані як запитів HTTP GET чи POST, коли користувач натискає на елементи

інтерфейсу до виконання різних дій). Його основна функція – викликати та координувати дію необхідних ресурсів та об'єктів, необхідних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель завдання і вибирає відповідний вид.

Модель (Model) – це інформація та правила, які використовуються для роботи з інформацією, що відповідає концепції адміністрування програми. У будь-якій програмі повна структура моделюється як інформація, яка обробляється певним чином. Модель дає контролеру представлення інформації, яку запитав клієнт (повідомлення, сторінка книги, колекція фотографій тощо). Відображення інформації буде однаковим, незалежно від того, як нам потрібно відобразити його клієнту. Таким чином ми вибираємо будь-який доступний перегляд для відображення інформації. Модель містить головну критичну частину обґрунтування застосування. Контролер містить загальне організаційне обґрунтування самої програми.

Вид (View) надає різні способи передачі інформації, отриманої в результаті демонстрації. Загалом може бути багато різноманітних поглядів але контролер обирає, який з них найкраще підходить для поточних обставин.

Загалом увесь цикл:

- браузер звертається до контролера;
- контролер звертається до моделі;
- модель звертається до б/д;
- дані обробляються та попадають до моделі;
- модель передає інформацію до контролеру;
- контролер передає цю інформацію в вигляд;
- вигляд рендериться в браузері.

Оскільки контролер є частиною URL-адреси браузер взаємодіє лише з ним. Клієнт може отримати дані тільки з контролера. Контролер в свою чергу формує вимоги клієнта, та є середньою особою для показу вигляду в браузері.

Інший компонент є дискреційним, але дуже важливим при розробці програмного забезпечення. Цим важливим елементом є база даних. У середині системи концепції MVC визнається, що моделі повинні працювати з базою даних. Час від часу такий відхід від правил може бути надзвичайно цінним з точки зору досягнення узгодженості коду та розуміння програми. Гіпотетично вид є дискреційним компонентом також у сюжеті MVC. Можна реалізувати всю функціональність без використання моделей у межах додатку. У цьому випадку клієнт буде підключатися до бази даних із записів контролера та вида.

7 ТЕСТУВАННЯ САЙТУ

Основна мета тестування – перевірка функціонування ресурсу на відповідність до пред'явлених вимог. Здавалося б, як можна перевірити сайт або програмне забезпечення на помилки, якщо спочатку ніхто не користується проектом? Все просто. Тестувальники спеціально створюють штучні ситуації, які можуть у майбутньому виникнути у роботі з ресурсом. Проводиться аналіз поведінки ресурсу на запропонованих умовах. Коли фахівець виявляє баг (професійна назва помилки), він передає свій звіт проєкт-менеджеру, який надалі розподіляє роботу зі свого усунення серед інших учасників проєкту. Коли помилки усувають сайт знову тестують. Тестування буде проводитись доти, доки сайт не стане ідеальним. Нижче наведені пункти, які обов'язково потрібно пройти при тестуванні веб-сайту. Так можна швидко виявити явні і поширені баги і виправити їх перед запуском в інтернет.

Тестування на помилки здійснюється різними методами і щоб домогтися коректного функціонування сайту, фахівці створюють певний план дій.

Етапи тестування веб-проєктів

- тестування функціональності;
- перевірка usability сайту (зручність користування);
- тест продуктивності;
- перевірка на безпеку;
- тестинг інтерфейсу, UI Testing.

Функціональне тестування сайту – це вид перевірки, що виявляє неправильну роботу функціоналу програми. Чек лист функціонального тестування:

- перевірка правильності роботи основних функцій ресурсу;
- виявлення посилань, що ведуть до однієї сторінки;
- коректність внутрішніх посилань;

– перевірка власних форм. Сюди входять: додавання коментарів у блог зворотний зв'язок та інше;

– наскільки контент сторінки вплине на цільову аудиторію і чи вплине взагалі, притягне до відповідальності.

Юзабіліті (usability) тестування. Даний етап тестування дозволяє перевірити, наскільки зручний сайт для користувача, наскільки легко знайти ту чи іншу інформацію. Словом, комфортність виконання бажаних дій. Основна мета тестування usability:

- визначити, чи ваш сайт для користувача зрозумілий, чи зручний;
- зрозуміти, наскільки зручна навігація;
- з'ясувати, яке враження створюється у користувача;
- оцінити, що може бути зайвим на ресурсі.

Існують такі види тестування сайту на юзабіліті.

Навігаційне тестування. Тут фахівець перевіряє чи всі сторінки, кнопки та поля зрозумілі користувачеві. Чи є доступ до головної сторінки та меню зі усіх інших сторінок.

Тестування контенту. Спеціаліст перевіряє наявність граматичних помилок, наскільки контент інформативний, чи мають картинки та відео потрібні розміри та якість, чи всі заголовки проставлені коректно.

Зручність користування. Тестувальник оцінює наскільки зрозуміла структура веб-програми і чи є зайві компоненти на ресурсі (перевіряються всі сторінки). Важливим для цієї перевірки є тестування на продуктивності. Це необхідно для того, щоб з'ясувати, чи витримає певне навантаження система. Визначити оперативність та стабільність роботи, адже сайт має витримувати високі навантаження.

Чек лист тестування на продуктивність:

- проведення стрес-тестування для визначення поведінки сайту за межами очікуваного робочого навантаження;
- вивчення поведінки сайту зі збільшенням робочого навантаження, так зване навантажувальне тестування;

- перевірка на здатність роботи протягом або трохи довше за прийнятний період. Тестування стабільності;
- тест на продуктивність ресурсу (об'ємне тестування) за рахунок збільшення обсягів бази даних;
- тестування паралелізму, де перевіряється швидкість роботи ресурсу при одночасному переживанні на ньому великої кількості користувачів;
- тест на швидкість завантаження сторінок;
- перевірка коректності роботи сайту при безперервному додатковому навантаженні, тестування на витривалість.

Під час розробки було написано інтеграційні тести за допомогою технології Cypress [22].

Приклад інтеграційних тестів наведено на рисунку 7.1

```

70     });
71   });
72 });
73
74 it('should display page footer', () => {
75   cy.server()
76     .request('/api/config')
77     .then((res) => {
78     const config = res.body as IConfig;
79
80     cy.get('p:contains(Still need help)').should('exist');
81     cy.get('a:contains(Find your rep)').should('exist');
82     cy.get('a:contains(Find your rep)')
83       .should('have.attr', 'target', '_blank')
84       .and('have.attr', 'href', config.repFinder.url);
85   });
86 });
87
88 it('should not display "Back to Top" button if the page is not scrolled \
89 and should display the button on the page scroll', () => {
90   cy.get('#app-content').scrollTo(0, 0);
91   cy.getByDataTestId('back-to-top-btn').should('not.be.visible');
92
93   cy.get('#app-content').scrollTo(0, 201);
94   cy.getByDataTestId('back-to-top-btn').should('be.visible');
95 });
96
97 it('should focus first item\'s link in the product list on "Back to Top" button click', () => {
98   cy.getByDataTestId('back-to-top-btn').click();
99
100   cy.getByDataTestId('title-heading')
101     .first()
102     .should('be.focused');
103 });
104

```

Рисунок 7.1 – Приклад інтеграційних тестів

Більшість інструментів наскрізного тестування засновані на Selenium, тому всі вони мають однакові проблеми. Щоб Cypress був іншим, розробники побудували нову архітектуру з нуля. Тоді як Selenium виконує віддалені команди через мережу, Cypress працює в тому ж циклі, що й програма. Cypress тестує все, що працює у веб-браузері. Вся архітектура, що оточує Cypress, створена для роботи з сучасними фреймворками JavaScript.

Є сотні проектів, які використовують новітні фреймворки React, Angular, Vue, Elm тощо. Cypress також однаково добре працює на старих серверних сторінках або програмах.

Також було написано unit тести за допомогою бібліотеки react-testing-library [22].

Приклад тестів наведено на рисунку 7.2.

```
24 describe('Function that checks equality of objects', () => {
25   it('should return false if objects have the same width but different keys and values', () => {
26     const result = deepEqual(sameWidthSplit[0], sameWidthSplit[1]);
27     expect(result).toEqual(false);
28   });
29
30   it('should return false if objects have the same keys but different values', () => {
31     const result = deepEqual(withDifferentValuesSplits[0], withDifferentValuesSplits[1]);
32     expect(result).toEqual(false);
33   });
34
35   it('should return false if one of the objects is undefined', () => {
36     const result = deepEqual(withDifferentValuesSplits[0], _undefined);
37     expect(result).toEqual(false);
38   });
39
40   it('should return false if one of the objects is null', () => {
41     const result = deepEqual(withDifferentValuesSplits[0], _null);
42     expect(result).toEqual(false);
43   });
44
45   it('should return false if null and date are compared ', () => {
46     const result = deepEqual(_null, _date);
47     expect(result).toEqual(false);
48   });
49
50   it('should return true if objects are identical', () => {
51     const result = deepEqual(equalSplits[0], equalSplits[1]);
52     expect(result).toEqual(true);
53   });
54 });
55
```

Рисунок 7.2 – Приклад unit тестів

8 ЕКОНОМІЧНА ЧАСТИНА

В рамках даної кваліфікаційної роботи бакалавра розробляється сайт для проведення опитувань.

Метою роботи є проектування web-платформи, яка призначена для проведення онлайн-виставок. Розроблений додаток буде інтегрований на сайт міжнародної конференції «Поліграфічні, мультимедійні та web-технології» і використаний для проведення виставки студентських робіт в 2023 році.

При розробці веб-сайту було відокремлена інформаційна ціль – розповісти і показати роботи учасників у голосуваннях. Також цей сайт має адміністративну частину, яка дозволяє розміщувати роботи та керувати процесом голосування [23]. Розглянемо переваги проектного сайту:

- проект виконаний згідно з документації та підтримує рівень АА щодо використання сайту за допомогою читача екрану;
- розробник використовував найпопулярніші технології. Серед них Reactjs та Nodejs;
- дизайн проекту виконувався згідно з сучасними тенденціями.

Розглянемо конкурентне середовище. Якщо інша компанія пропонує той самий продукт, як і ви, то ця компанія є вашим прямим конкурентом. Прямі конкуренти пропонують одні й ті самі товари за різною ціною та різною якістю обслуговування. Щоб зрозуміти, чому конкуренти можуть випереджати вас, необхідно проаналізувати їх роботу. Знаючи які ключові слова вони використовують для збільшення трафіку ви маєте можливість оптимізувати свій контент. Пошукові системи дають кожному сайту шанс зайняти перше місце в пошуковій видачі. Якщо ваші конкуренти займають високі позиції в пошуковій видачі Google, якість їх ключових слів та зусилля з пошукової оптимізації мають бути чудовими. Технології також можуть стати величезним фактором, чому ви не можете йти у ногу зі своїми

конкурентами. Їх контент повинен бути більш оптимізований для пошукових систем, тому що вони використовують більш просунуті інструменти.

У результаті аналізу технологій які використовують веб-конкуренти було виділено їх недоліки:

- використання застарілих технологій, таких як Php;
- надання переваги статиці, ніж динаміці;
- ігнорування семантики та маленька увага до використання користувачами з обмеженими можливостями;
- зневаження зовнішніх факторів.

Також було виділено їх переваги:

- велика область використання;
- широка модель функціональності.

Спостереження за інструментами і плагінами, що використовувались конкурентами, дали уявлення про тактику, що застосовується. Це було враховано під час проектування та розробки web-платформи. Розроблений сайт відповідає вимогам сучасності та використовує найкращі технології, отже, можна говорити про його конкурентоспроможність.

Він буде дуже вдалим доповненням до головного сайту міжнародної науково-технічної конференції «Поліграфічні, мультимедійні та web-технології».

Розглянемо джерела економії, доходу та джерела фінансування [24].

Для фірми-розробника джерелом доходу є кошти підприємства-замовника. Серед витрат фірми можна виділити бюджет на заробітну плату працівникам. Також до витрат враховують передбачені законодавством оплачувані вихідні, лікарняні та материнські виплати. Тобто, така фірма заробляє відсоток за працю кожного розробника на проекті підприємства-замовника.

Для підприємства-замовника джерелом економії можуть бути цілі діджиталізації, тобто заміни ручної праці, а також зосередження нової

аудиторії до свого продукту. Витрати підприємства складаються з перелічених у контракті з фірмою-розробником коштів.

Розглянемо порядок проектування веб-сайту.

Розробка веб-сайту потребує [25]:

– початкового етапу, на якому формується уява про майбутній продукт;

– основного етапу, який поєднує розробку дизайну, програми та оформлення документації;

– заключного етапу, який характеризується презентацією продукту.

Здійснимо розрахунок собівартості і ціни розробки веб-сайту. У собівартість розробки веб-сайту входять наступні статті витрат:

- основна заробітна плата;
- додаткова заробітна плата;
- єдиний соціальний внесок;
- інші витрати.

У розробці додатка беруть участь 4 фахівці – проджект-менеджер, аналітик даних, UI/UX дизайнер та розробник. Заробітна плата проджект-менеджера становить 27 000 грн./міс. [1], аналітика даних – 27 000 грн./міс.[2], UI/UX дизайнера – 13 500 грн./міс.[3], розробника – 29 700 грн./міс[4].

Знаючи, що місяць складається із 160 годин, можна розрахувати годинну ставку кожного працівника:

$$27\,000/160 = 168,75 \text{ грн./год.}$$

$$27\,000/160 = 168,75 \text{ грн./год.}$$

$$13\,500/160 = 84,375 \text{ грн./год.}$$

$$29\,700/160 = 185,625 \text{ грн./год.}$$

Основні етапи створення сайту зайняли 15 днів. Розрахунок основної заробітної плати:

$$607,5 * 8 * 15 = 72\,900 \text{ грн.}$$

Додаткова заробітна плата – це винагорода за працю понад установлені норми, за трудові успіхи. Вона включає доплати, надбавки, гарантійні та компенсаційні виплати, передбачені чинним законодавством; премії, пов'язані з виконанням виробничих завдань і функцій. У даному випадку додаткова заробітна плата становить 10 % від основної:

$$72\,900 * 0,1 = 7\,290 \text{ грн.}$$

Розрахунок загальних витрат на заробітну плату зведено у таблиці 8.1.

Таблиця 8.1– Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконавець		Годинна ставка	Тривалість виконання, дні	Заробітна плата, грн.
		Кількість	Посада			
1. Початковий	Аналіз середовища та розробка портрету користувача	1	Дизайнер	84,375	2	1 350
	Створення документації	1	Бізнес-аналітик	168,75		2 700
	Технічні дослідження та настройка гіт простору	1	Розробник	185,625		2 970
	Менеджмент команди	1	Менеджер	168,75		2 700
2. Основний етап	Розробка дизайну	1	Дизайнер	84,375	10	6 750
	Розробка компонентів та розміщення на просторі тестування. Комплексне тестування сайту.	1	Розробник	185,625		14 850
	Оформлення документації	1	Бізнес-аналітик	168,75		13 500
	Менеджмент команди	1	Менеджер	168,75		13 500
3. Заключний етап	Розміщення на фінальному просторі.	1	Розробник	185,625	3	4 455
	Супровід на	1	Бізнес-	168,75		4 050

	фінального простору		аналітик			
	Менеджмент команди	1	Менеджер	168,75		4 050
	Презентація продукту	1	Дизайнер	84,375		2 025
Разом					15	72 900
Додаткова заробітна плата						7 290
Усього						80 190

Ставка єдиного соціального внеску становить 22 % від величини основної та додаткової заробітної плати:

$$80\,190 * 0,22 = 17\,641,8 \text{ грн.}$$

Витрати на електроенергію розраховуються виходячи зі споживаної потужності устрою і тарифу на електроенергію. У даному випадку передбачається використання 4 комп'ютерів з потужністю 0,7 кВт/год. Вартість однієї кВт/год електроенергії прийнято у розмірі 1,619 грн. Час використання електроенергії в процесі розробки:

$$8 * 15 = 120 \text{ години.}$$

Отже, плата за електроенергію складе:

$$0,7 * 1,619 * 120 * 4 = 543,984 \text{ грн.}$$

Витрати на обслуговування ЕОМ визначаються з вартості ЕОМ і часу її експлуатації, після закінчення якого, вона підлягає заміні (звичайно цей час не перевищує 3-х років), протягом року ЕОМ використовує 254 робочих дні.

$$(4200 / (3 * 8 * 254)) * 120 = 82,68 \text{ грн.}$$

Проект впроваджується на 1 компанію, тому собівартість розробки

$$98\,458,46 / 1 = 98\,458,46 \text{ грн.}$$

Розрахуємо суму прибутку від реалізації розробки (виходячи з рівня рентабельності 40 %):

$$98\,458,46 * 0,4 = 39\,383,4 \text{ грн.}$$

Розрахуємо ціну розробки сайту без податку на додану вартість (ПДВ):

$$98\,458,46 + 39\,383,4 = 137\,841,85 \text{ грн.}$$

Розрахуємо суму ПДВ, вона рівна 20% від ціни:

$$137\,841,85 * 0,2 = 27\,568,37 \text{ грн.}$$

Розрахуємо ціну сайту з урахуванням ПДВ:

$$137\,841,85 + 27\,568,37 = 165\,410,219 \text{ грн.}$$

Результати розрахунків наведено у таблиці 8.2.

Таблиця 8.2 – Розрахунок витрат на розробку та ціни веб-сайту

Стаття витрат	Сума, грн.
Основна заробітна плата	72 900
Додаткова заробітна плата	7 290
Єдиний соціальний внесок	17 641,8
Витрати на обслуговування ЕОМ	82,68
Витрати на електроенергію	543,984
Собівартість розробки сайту	98 458,46
Прибуток (рівень рентабельності 40 %)	39 383,4
Ціна без ПДВ	137 841,85
Податок на додану вартість (ПДВ)	27 568,37
Ціна з урахуванням ПДВ	165 410,219

Таким чином, виходячи з виконаних розрахунків повна вартість розробки web-платформи для опитувань складе 165 410,219 грн. Термін виконання всіх етапів розробки становить 15 днів для команди з одного проджект-менеджера, аналітика даних, UI/UX дизайнера і розробника. Очікувана сума прибутку складе 39 383,4 грн.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра розроблена web-платформа, яка призначена для проведення онлайн-виставок. Розроблений додаток буде інтегрований на сайт міжнародної конференції «Поліграфічні, мультимедійні та web-технології» і використаний для проведення виставки студентських робіт в 2023 році.

На початку проектування було здійснено аналіз існуючих аналогів, та виявлено їх переваги та недоліки. Основною проблемою є те, що всі аналогічні рішення призначені для загальних задач і не можуть використовуватись для вирішення поставлених в роботі задач.

Тому на етапі технічного завдання були поставлені задачі щодо реалізації web-платформи для проведення онлайн-виставок, а також виділені основні функції цього додатка. Також проаналізовані та обрані необхідні інструментальні засоби розробки сайту – редактора коду, технології для фронт-енд, використання бібліотек, каскадних таблиць стилів та БД.

Виконана розробка інформаційної структури і навігації сайту для різних користувачів, виконано проектування карти сайту. Також розроблено графічний дизайн і здійснена верстка сайту.

Для реалізації всього функціоналу та адміністрування web-платформи розроблена серверна частина сайту. Проведено його тестування.

В результаті виконання кваліфікаційної роботи бакалавра був реалізований сайт для розміщення, перегляду та участі у голосуваннях, який виконує наступні функції:

- створення опитувань;
- підтримка завантаження графічних матеріалів;
- попередній перегляд матеріалу перед публікацією опитування;
- зупинка голосування та перегляду результатів;
- перегляд існуючих опитувань з панелі;

– перегляд сторінки з результатами опитування та завантаження їх на різні пристрої;

– участь в опитуванні.

В економічній частині було обґрунтовано витрати на розробку веб-сайту, оцінено ринки збуту та конкурентів. Розроблений сайт відповідає всім поставленим на початковому етапі проектування вимогам. Повна вартість розробки web-платформи для опитувань складе 165 410,219 грн.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Веб-доступність. Що варто знати кожному Front-end розробнику і дизайнеру. URL: <https://dou.ua/lenta/articles/web-accessibility/> (дата звернення: 07.05.2022).
2. How to Make a Poll. URL: <https://www.poll-maker.com/> (дата звернення: 07.05.2022).
3. Create online polls fast with the Doodle poll maker. URL: <https://doodle.com/poll-maker> (дата звернення: 07.05.2022).
4. Directpoll. URL: <https://www.directpoll.com/> (дата звернення: 07.05.2022).
5. Simplify decisionmaking with PollUnit. URL: <https://pollunit.com/en> (дата звернення: 07.05.2022).
6. Chto takoe ynformatsyonniy sait. URL: <https://semantica.in/blog/chto-takoe-informacziionnyj-sajt.html> (дата звернення: 07.05.2022).
7. Корпоративний сайт: що це, особливості розробки URL: <https://xn--90aamhdбасрq0s.xn--j1amh/teoriya/korporativnyy-sayt-shcho-tse-osoblyvosti-rozrobky> (дата звернення: 07.05.2022).
8. Sotnik S., & et al.. (2022). Key Directions for Development of Modern Expert Systems. International Journal of Engineering and Information Systems (IJEAIS), 6(5), 4-10.
9. Сторінка-вітрина. URL: <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0-%D0%B2%D1%96%D1%82%D1%80%D0%B8%D0%BD%D0%B0> (дата звернення: 07.05.2022).
10. Промо-сайт. URL: <https://websait.uz.ua/posluhy/stvorenniya-sajtiv/promo-sajt/> (дата звернення: 15.05.2022).
11. Deineko, Zh., & et al.. (2022). Usage and Application Prospects QR Codes. International Journal of Engineering and Information Systems (IJEAIS), 6(7), 40-48..
12. Geographical distribution of active users. URL: <https://octoverse.github.com/#geographical-distribution-of-active-users> (дата звернення: 20.05.2022).
13. The modern web developer's platform. URL: <https://angular.io/> (дата звернення: 20.05.2022).

14. TypeScript is JavaScript with syntax for types. URL: <https://www.typescriptlang.org/> (дата звернення: 25.05.2022).

15. React. A JavaScript library for building user interfaces. URL: <https://reactjs.org/> (дата звернення: 01.06.2022).

16. Redux A Predictable State Container for JS Apps. URL: <https://redux.js.org/> (дата звернення: 01.06.2022).

17. Npx. URL: <https://www.npmjs.com/package/npx> (дата звернення: 01.06.2022).

18. Download the Node.js source code or a pre-built installer for your platform, and start developing today. URL: <https://nodejs.org/en/download/> (дата звернення: 01.06.2022).

19. The web has evolved. Finally, testing has too. URL: <https://www.cypress.io/> (дата звернення: 01.06.2022).

20. Вовк О.В., Черемський Р.А. Інфографіка як ефективний засіб навчання // Системи обробки інформації. 2017. Випуск 4 (150). С. 199-205.

21. Доценко Д.В., Чеботарьова І.Б. Розробка web-сайту за принципом Mobile First Design // Поліграфічні, мультимедійні та web-технології: матеріали школи-семінару VII Міжнародної науково-технічної конференції (20 квітня 2022, м. Харків). Харків, 2022. Т. 2. С. 40-42.

22. React Testing Library. URL: <https://testing-library.com/docs/react-testing-library/intro/> (дата звернення: 01.06.2022).

23. Дорофєєва К.С., Чеботарьова І.Б. Розробка WEB-платформи для проведення онлайн-виставок // Поліграфічні, мультимедійні та web-технології: матеріали молодіжної школи-семінару VII Міжнародної наук.-техн. конф., 17-21 травня 2022 р. Харків: ХНУРЕ, 2022. Т. 2. С. 37-39.

24. Полозова Т.В. Методичні вказівки до виконання економічної частини кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 186 Видавництво та поліграфія усіх форм навчання. Харків: ХНУРЕ, 2022. 47 с.

25. Методичні вказівки з виконання кваліфікаційної роботи для студентів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти спеціальності 186 «Видавництво та поліграфія» за освітньою програмою «Видавничо-поліграфічна справа» / В.П. Ткаченко, А.В. Бізюк, О.В. Вовк, І.М. Єгорова, В.Ф. Челомбїтько. Харків: ХНУРЕ, 2020. 68 с.