

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЦЕНТР ЗАОЧНОЇ ТА ДИСТАНЦІЙНОЇ ФОРМИ НАВЧАННЯ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток для протоколювання наради з обговорення
проектних рішень»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проекування»

Виконавець роботи: студент групи ІТдн-84жт Мирончук Олександр
Юрійович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2022 р.

Науковий керівник

_____ (підпис) к.т.н., доц., Неня В.Г.
(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2022

Сумський державний університет
Центр заочної та дистанційної форми навчання
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2022 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Миرونчук Олександр Юрійович

1 Тема роботи Web-додаток для протоколювання наради з обговорення проектних рішень

керівник роботи Неня Віктор Григорович, к.т.н., доцент,

затверджені наказом по університету від «__» _____ 2022 р. _____

2 Строк подання студентом роботи «__» _____ 2022 р.

3 Вхідні дані до роботи результати аналізу предметної області, вимоги державних стандартів

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області, актуальність досліджуваної задачі, перелік вимог до Web-додатку для протоколювання наради з обговорення проектних рішень, огляд існуючих програмних продуктів для вирішення поставлених задач, постановка задачі, програмна реалізація web-додатку, висновки, список використаних джерел, додаток а.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Схеми та рисунки по теоретичному матеріалу та зображенням функціоналу застосунку діаграми, ілюстрації інтерфейсу веб-додатку, презентація для захисту дипломної роботи.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 7 жовтня 2021р

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз предметної області	17.11.21	
1.1.	Актуальність досліджуваної задачі	17.11.21	
1.2.	Перелік вимог до Web-додатку для протоколювання наради з обговорення проектних рішень	26.11.21	
1.3.	Огляд існуючих програмних продуктів для вирішення поставлених задач	07.12.21	
2.	Постановка задачі	15.12.21	
2.1.	Призначення та мета Web-додатку	15.12.21	
2.2.	Цільова аудиторія	22.12.21	
2.3.	Вимоги до Web-додатку	30.12.21	
2.4.	Вимоги до видів забезпечення	05.01.22	
3.	Програмна реалізація web-додатку	15.02.22	
3.1.	Засоби та технології розробки	15.02.22	
3.2.	Моделювання роботи web-додатку для протоколювання наради з обговорення проектних рішень	20.04.22	
3.3.	Створення та перший запуск проекту	25.04.22	
3.4.	Загальне представлення об'єктів	15.05.22	

Студент _____
(підпис)

Мирончук О.Ю.

Керівник роботи _____
(підпис)

к.т.н., доц. Неня В. Г.

РЕФЕРАТ

АНОТАЦІЯ

На наступних окремих аркушах АННОТАЦІЯ і ABSTRACT

Тема роботи «Web-додаток для протоколювання наради з обговорення проектних рішень»

Пояснювальна записка – 73 сторінки, кількість розділів 3, кількість рисунків - 21, кількість таблиць – 2, кількість використаних джерел – 21.

Метою дипломного проекту є розробка Web-додатку для протоколювання наради з обговорення проектних рішень.

Проведено аналіз існуючих програмних продуктів, які вирішують подібні задачі.

Під час роботи над курсовою роботою досліджено предметну область, для якої призначений веб-застосунок.

У роботі розглянуто web-додаток для обговорення, спілкування та зображення графічних зображень різних форматів.

В результаті роботи розроблено Web-додаток з виконанням усіх поставлених вимог.

Ключові слова: Visual Studio Code, HTML, JavaScript, SVG, SCSS, CSS, React, Redux.

ЗМІСТ

Вступ.....	6
Розділ 1 Аналіз предметної області.....	8
1.1 Актуальність досліджуваної задачі.....	8
1.2 Перелік вимог до Web-додатку для протоколювання наради з обговорення проектних рішень	10
1.3 Огляд існуючих програмних продуктів для вирішення поставлених задач.....	13
Розділ 2 Постановка задачі.....	17
2.1 Призначення та мета Web-додатку.....	17
2.2 Цільова аудиторія.....	17
2.3 Вимоги до Web-додатку	18
2.3.1 Основні вимоги до WEB- додатку.....	18
2.4 Вимоги до видів забезпечення	26
Розділ 3 Програмна Реалізація web-додатку	27
3.1 Засоби та технології розробки	27
3.2 Моделювання роботи web-додатку для протоколювання наради з обговорення проектних рішень	31
3.3 Створення та перший запуск проекту Web-додатку.....	34
3.4 Загальне представлення об'єктів.	36
Висновки	39
Список використаних джерел	40
Додаток А.....	42
Додаток В.....	56

ВСТУП

У на час є багато різних месенджерів, соціальних мереж, форумів для обговорення різних питань, сервісів відео-телефонного зв'язку та інших веб-додатків і сайтів. З кожним роком таких веб-застосунків та сайтів стає все більше і більше.

Часто потрібен такий додаток, що дозволив би нам не тільки спілкуватися в чаті чи по відеозв'язку, а який дає змогу для графічного відображення та пояснення. Який дає змогу намалювати та зобразити свої думки і донести їх іншим.

Такий, який замінить вашу оффлайн конференцію на онлайн та навіть стане краще за неї.

Як було сказано вище, цей веб-застосунок дає змогу спілкуватися з вашими знайомими, друзями, рідними, колегами по роботі чи іншими людьми, але його значна перевага в тому, що він дає можливість не просто спілкуватися, а відображати свої думки в графічному редакторі, який дає змогу малювати та створювати SVG –зображення з подальшою можливістю редагування іншими користувачами, які мають до нього доступ.

Так його функціонал дуже схожий до звичайного форуму, але це більше ніж просто форум, тому,що він відкриває для вас більше можливостей і функціоналу та інструментів.

Він змінить ваші звичні та нудні конференції по обговоренню, на справжній прорив.

Виглядає все це дуже просто, як звичайна дошка для малювання у кабінеті чи залі на якій ви щось пишете чи малюєте, але цю дошку може бачити безмежна кількість людей та кожен має можливість її редагувати не встаючи зі свого місця та не перебуваючи з вами в одному приміщенні.

Тому цей додаток настільки крутий і кращий за інші аналоги.

Масштабована векторна графіка (SVG) — це формат векторного зображення на основі XML для двовимірної графіки з підтримкою

інтерактивності та анімації. Специфікація SVG є відкритим стандартом, розробленим Консорціумом World Wide Web Consortium (W3C) з 1999 року.

Зображення SVG визначаються у форматі векторної графіки та зберігаються у текстових файлах XML. Таким чином, зображення SVG можна масштабувати за розміром без втрати якості, а файли SVG можна шукати, індексувати, створювати сценарії та стискати. Текстові файли XML можна створювати та редагувати за допомогою текстових редакторів або векторних графічних редакторів, і вони відтворюються найбільш часто використовуваними веб-браузерами. – [12]

Метою даного дослідження є розробка **«Web-додатку для протоколювання наради з обговорення проектних рішень»**.

Основні задачі полягають в тому, щоб :

- дослідити предметну область використання web-додатку для протоколювання наради з обговорення проектних рішень;
- розглянути аналоги подібного продукту.
- зробити висновки.
- розробити додаток.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність досліджуваної задачі

У наш час є багато різних додатків для спілкування, обговорення, створення різних SVG-зображень, у яких є схожий функціонал. Але значна перевага цього додатку в тому, що він поєднує в собі весь цей функціонал в одному місці, іншими словами – додаток для спілкування, обговорення та малювання в одній коробці.

Проаналізуємо, чим веб-додаток відрізняється від звичайного сайту.

1.1.1 Інтерактивність

Першим пунктом, з якого слід розпочати диференціацію «веб-додаток і веб-сайт», є інтерактивність. Веб-сайт надає візуальний і текстовий вміст, який користувач може бачити і читати, але не впливає жодним чином. У випадку веб-програми користувач може не тільки читати вміст сторінки, але й керувати даними на цій сторінці. Взаємодія відбувається у формі діалогу: користувач натискає кнопку або надсилає форму та отримує відповідь зі сторінки. Ця відповідь може мати форму завантаження документа, онлайн-чату, електронного платежу тощо.

Ілюстративним прикладом інтерактивності веб-додатків є програма онлайн-банкінгу, яка виконує транзакції на основі введених даних клієнта. Подібну функціональність можна знайти в інтернет-магазині, що дозволяє відвідувачам здійснювати пошук у каталозі та миттєво купувати товари. Ще одним вражаючим прикладом є соціальні мережі. Вони підключають користувачів через чати та блог-платформи, генерують вміст каналів на основі вподобань користувачів і дозволяють майже необмежено ділитися вмістом, не кажучи вже про їхні вбудовані міні-додатки для розваги користувачів.

Проблема в тому, що сьогодні рідко можна зустріти веб-сайт без натяку на інтерактивність. Сучасні веб-сайти зазвичай містять невеликі елементи веб-додатків. Наприклад, веб-сайт ресторану може містити віджет Google Maps,

який показує маршрут до цього ресторану. Однак у випадку з веб-сайтами баланс між інформаційним змістом та інтерактивністю зміщується в бік першого. Типовий веб-сайт містить набагато менше інтерактивних елементів, ніж інформаційний зміст, і користувач зазвичай проводить більшу частину часу на веб-сайті, читаючи, переглядаючи або слухаючи. З веб-додатками ситуація протилежна, оскільки їх основна функціональність заснована на взаємодії.

1.1.2 Інтеграція

Інтеграція означає об'єднання різних компонентів для створення більш комплексної системи. І веб-сайти, і веб-додатки можна інтегрувати з іншим програмним забезпеченням (CRM, ERP тощо). Все-таки інтеграція більш характерна для веб-додатків, оскільки їхня складна функціональність часто вимагає взаємодії з додатковими системами.

Візьмемо інтеграцію бізнес-веб-додатка (скажімо, електронного магазину) із системою CRM (Customer Relationship Management). CRM зберігає всі дані клієнтів в одному місці, забезпечуючи легкий доступ до них для співробітників. Інтеграція дозволить автоматично збирати дані користувача веб-додатку та зберігати їх у CRM. Таким чином, ваша команда отримає доступ до повного набору даних про клієнтів, їхні запити, спілкування та відгуки. Це дозволяє вивчати поведінку клієнтів і купівельні звички, а також швидше врегулювати їхні претензії. Більше того, будь-які зміни в даних клієнта будуть миттєво відображені в CRM. Завжди залишаючись в курсі своїх уподобань клієнтів, ви зменшите показники відтоку та збільшите продажі.

Веб-сайт також можна інтегрувати з CRM. Це дозволяє надавати користувачам більш персоналізований контент. Однак для веб-сайту це скоріше рідко реалізована функція, ніж частина основної функціональності.

1.1.3 Аутентифікація

Аутентифікація – це процедура, яка передбачає введення логіна та пароля користувача для отримання доступу до системи. Це необхідно для веб-програм, які потребують будь-якої особистої інформації. Облікові записи користувачів

мають бути захищені, щоб запобігти несанкціонованому доступу та витоку конфіденційних даних.

Веб-додатки здебільшого вимагають аутентифікації, оскільки вони пропонують набагато ширший спектр можливостей, ніж веб-сайти. Розглянемо приклад соціальних мереж. При реєстрації ви створюєте обліковий запис і отримуєте унікальний ідентифікаційний номер. Система попередить вас, якщо ваш логін і пароль слабкі. Якщо ви залишите їх без змін, хакери можуть отримати доступ до вашого облікового запису та вкрати вашу інформацію, а також дратувати інших користувачів небажаними листами під вашим іменем.

Для інформаційних веб-сайтів автентифікація не є обов'язковою. Користувачеві може бути запропоновано зареєструватися, щоб отримати доступ до додаткових опцій, недоступних для незареєстрованих відвідувачів сайту. Наприклад, ви можете переглядати новини та пропоновані статті на новинному веб-сайті, не реєструючись. Однак, якщо ви хочете залишити коментар, вам доведеться увійти. Таким чином користувачі підтверджують свою особу, дозволяючи системі блокувати спамерів.

Як бачите, і веб-сайти, і веб-додатки можуть вимагати аутентифікації. Однак для веб-додатків це є обов'язковим із міркувань безпеки.

1.2 Перелік вимог до Web-додатку для протоколювання наради з обговорення проектних рішень

Базовий функціонал додатку повинен включати:

- Можливість створення нового користувача та авторизацію вже зареєстрованого.
- Створення нових дописів та їх подальша можливість редагування
- Створення у ваших дописах SVG-зображень.

- Можливість визначення доступу для інших користувачів редагувати вже існуючі ілюстрації.

- Коментування дописів.
- Обговорення проблеми.
- Пошук в додатку.
- Сортування та фільтрація дописів по різних параметрах(дата, популярність, активність, актуальність, категорія).
- Створення спеціальних міток для позначення типу допису.
- Веб-сервер для зберігання даних додатку.

Дизайн веб-додатків є важливим етапом створення веб-додатку. Він зосереджується на зовнішньому вигляді веб-програми. Етап проектування охоплює кілька різних аспектів, включаючи дизайн інтерфейсу користувача (UI), зручність використання (UX), створення контенту та графічний дизайн

Веб-додаток повинен відповідати таким загальним стандартам:

- Повинен бути простим та інтуїтивно зрозумілим у використанні для цільової аудиторії.
- Повинен функціонувати логічним чином для цільової аудиторії.
- Повинен використовувати стилі, які є послідовними в усій програмі та на пов'язаному веб-сайті, зокрема:
 - Використання великої літери (наприклад, регістр заголовка та регістр речення).
 - Використання пунктуації (наприклад, послідовне використання/не використання двокрапки на етикетках).
 - Повідомлення про помилки мають відобразитися в однаковому місці та в однаковому стилі.
 - Послідовне використання будь-яких позначень веб-документів (наприклад, PDF, DOC тощо).
 - Макет/відстань (наприклад, пробіл між міткою поля та елементом керування введенням).
 - Описові назви та описи метаданих.
- Повинен дотримуватися кращих галузевих практик.

- Елементи керування формою, які недоступні, необхідно приховати – не використовувати неактивні елементи керування.

1.2.1 Чуйний дизайн

Загальнодоступні веб-додатки мають бути чутливими. Додаток має правильно змінювати розмір і працювати на мобільних пристроях.

1.2.2 Навігація

Кожна веб-сторінка в програмі повинна мати одне або кілька посилань або кнопок керування, які дозволяють користувачеві переміщатися вперед і назад у програмі без використання кнопки «Назад» або інших функцій навігації браузера.

1.2.3 Перевірка полів введення форми

Поля форми мають бути перевірені, щоб переконатися, що обов'язкові поля заповнені, числові поля містять числові дані, а введені дані правильно відформатовані (наприклад, дата, адреса електронної пошти).

1.3 Огляд існуючих програмних продуктів для вирішення поставлених задач

Зараз є дуже велика кількість додатків і різних платформ для спілкування таких соціальні мережі, форуми, месенджери тощо.

Ось найактуальніші з них зараз у наш час:

Google Meet (раніше відомий як **Hangouts Meet**) — це служба відеозв'язку, розроблена компанією Google. Це один із двох додатків, які замінюють Google Hangouts, а інший є Google Chat. Він також має намір замінити Google Duo для споживачів на мобільних пристроях. Після того, як у лютому 2017 року Google офіційно випустив програму для iOS лише за запрошенням, у березні 2017 року Google офіційно запустив Meet. Сервіс був представлений як додаток для відеоконференцій до 30 учасників, який описується як зручний для підприємств. версія Hangouts. Він був запущений з веб-додатком, додатком для Android та iOS.

Хоча Google Meet представив вищезгадані функції для оновлення оригінальної програми Hangouts, деякі стандартні функції Hangouts були застарілі, зокрема перегляд відвідувачів і чат одночасно. Кількість відеоканалів, дозволених за один раз, також зменшено до 8 (у той час як у макеті «плиток» можна відобразити до 4 каналів), пріоритетним є ті учасники, які останнім часом використовували свій мікрофон. Крім того, такі функції, як вікно чату, були змінені, щоб накладати відеоканали, а не змінювати розмір останніх відповідно до розмірів. Hangouts планувалося припинити в першій половині 2021 року.

Під час пандемії COVID-19 використання Meet зросло в 30 разів у період із січня по квітень 2020 року, при цьому 100 мільйонів користувачів щодня отримували доступ до Meet, у порівнянні з 200 мільйонами щоденних використання Zoom станом на останній тиждень квітня 2020 року Google призупинив свій звичайний 60-хвилинний ліміт для неоплачених облікових записів. – [12]

Zoom Meetings (зазвичай скорочений до **Zoom** і стилізований під **zoom**) — це запатентована програма для відеотелефонії, розроблена компанією Zoom Video Communications. Безкоштовний план дозволяє одночасно брати участь до 100 учасників з обмеженням у 40 хвилин. Користувачі мають можливість оновити, підписавшись на платний план. Найвищий план підтримує до 1000 одночасних учасників для зустрічей тривалістю до 30 годин.

Під час пандемії COVID-19 відбулося значне збільшення використання Zoom для віддаленої роботи, дистанційної освіти та онлайн-соціальних стосунків. Збільшення призвело до того, що Zoom став 5-м за кількістю завантажуваних мобільних додатків у світі в 2020 році – 477,3 мільйона завантажень.

Zoom спочатку був заснований у 2011 році. Його штаб-квартира розташована в Сан-Хосе, Каліфорнія. Zoom також має офіси в Європі, Азії та Австралії.

21 серпня 2012 року була запущена бета-версія Zoom, яка могла проводити конференції лише з 15 відеоучасниками. 25 січня 2013 року була випущена версія 1.0 програми зі збільшенням кількості учасників на конференцію до 25. До кінця першого місяця у Zoom було 400 000 користувачів, а до травня 2013 року він зріс до понад мільйона користувачів. Після початку пандемії COVID-19 до лютого 2020 року, Zoom отримав 2,22 мільйона користувачів у 2020 році – більше користувачів, ніж він зібрав за весь 2019 рік при цьому ціна акцій компанії зросла на 35 відсотків. У березні 2020 року додаток Zoom було завантажено 2,13 мільйона разів. У квітні 2020 року Zoom мав понад 300 мільйонів учасників щоденних зустрічей. 24 серпня 2020 року Zoom зазнав масових відключень протягом кількох годин до відновлення роботи. – [12]

Skype — це власна телекомунікаційна програма, що керується Skype Technologies, підрозділом Microsoft, найбільш відомим завдяки відеотелефонії на основі VoIP, відеоконференціям і голосовим дзвінкам. Він також має обмін миттєвими повідомленнями, передачу файлів, дебетові дзвінки на стаціонарні

та мобільні телефони (через традиційні телефонні мережі) та інші функції. Skype доступний на різних настільних, мобільних та ігрових консольних платформах.

Skype був створений Нікласом Зеннстрьомом, Янусом Фріісом та чотирма естонськими розробниками і вперше випущений у серпні 2003 року. У вересні 2005 року eBay придбала Skype за 2,6 мільярда доларів. У вересні 2009 року Silver Lake, Andreessen Horowitz та Інвестиційна рада Канадського пенсійного плану купили 65% Skype за 1,9 мільярда доларів у eBay, оцінивши бізнес у 2,92 мільярда доларів. У травні 2011 року Microsoft купила Skype за 8,5 мільярдів доларів і використала його для заміни свого Windows Live Messenger. Станом на 2011 рік більшість команди розробників і 44% усіх співробітників підрозділу були в Таллінні та Тарту, Естонія.

Спочатку Skype містив гібридну однорангову і клієнт-серверну систему.

У травні 2012 року він повністю працював на супервузлах, керованих Microsoft у 2017 році він змінився з однорангової служби на централізовану службу на базі Azure.

Станом на березень 2020 року Skype використовували 100 мільйонів людей принаймні раз на місяць і 40 мільйонів людей щодня. Під час пандемії COVID-19 Skype втратив значну частину своєї частки ринку через Zoom. – [12]

Microsoft Teams — це запатентована платформа ділового спілкування, розроблена компанією Microsoft, як частина сімейства продуктів Microsoft 365. Teams переважно конкурує з подібним сервісом Slack, пропонуючи чат у робочому просторі та відеоконференції, зберігання файлів та інтеграцію додатків. Teams замінює інші платформи обміну бізнес-повідомленнями та спільною роботою, які керуються Microsoft, зокрема Skype для бізнесу та Microsoft Classroom. Під час пандемії COVID-19 Teams та інше програмне забезпечення, таке як Zoom і Google Meet, викликав великий інтерес, оскільки багато зустрічей перейшли у віртуальне середовище. Станом на 2021 рік він має близько 250 мільйонів користувачів щомісяця. – [12]

Discord — це соціальна платформа VoIP та миттєвих повідомлень.

Користувачі мають можливість спілкуватися за допомогою голосових дзвінків, відеодзвінків, текстових повідомлень, медіа та файлів у приватних чатах або в рамках спільнот, які називаються «серверами». Сервер — це набір постійних чат-кімнат і голосових каналів, доступ до яких можна отримати за допомогою посилань із запрошенням. Discord працює на Windows, macOS, Android, iOS, iPadOS, Linux та у веб-браузерах. Станом на 2021 рік, сервіс має понад 350 мільйонів зареєстрованих користувачів і понад 150 мільйонів активних користувачів щомісяця. — [12]

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ

2.1 Призначення та мета Web-додатку

Застосунок дає змогу обговорювати різні питання з вашими знайомими, друзями, рідними, колегами по роботі чи іншими людьми, але його значна перевага в тому, що він дає можливість не просто спілкуватися, а відобразити свої думки в графічному редакторі, який дає змогу малювати та створювати SVG – зображення з подальшою можливістю редагування іншими користувачами, які мають до нього доступ.

Метою даного дослідження є розробка web-додатку для протоколювання наради з обговорення проектних рішень.

2.2 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

1. Програмісти.
2. Менеджери.
3. Дизайнери.
4. Блогери
5. Студенти
6. Школярі
7. Викладачі
8. Офісні працівники
9. Аналітики
10. Тестувальники
11. Організації у яких часто відбуваються конференції.

2.3 Вимоги до Web-додатку

2.3.1 Основні вимоги до WEB- додатку

Веб-додаток повинен мати вигляд звичайного сайту-форуму, але розділеного на кілька частин з можливістю авторизації, додаванням та редагуванням дописів, та рисунків у цих дописах, коментуванням інших публікацій користувачів.

Користувач повинен мати можливість створення та редагування SVG-зображень, за допомогою панелі інструментів в додатку. Інтерфейс має бути інтуїтивно зрозумілим, дозволяти нанесення довільних ліній та мати набір для використання готових геометричних фігур в редакторі таких як лінії, квадрати, круги, овали тощо та можливість переміщати їх за необхідності, видаляти та редагувати, додавати текст.

2.3.2 Структура додатку

Застосунок складається з наступних елементів:

1. У шапці додатку знаходиться логотип додатку зліва та блок для авторизація або профіль користувача з права із кнопкою для створення публікації.
2. В основному контейнері з лівого боку знаходиться сайдбар який відкривається/закривається (за замовчуванням закритий) в якому знаходиться навігаційне меню додатку.
3. По центру знаходиться блок для відображення усіх постів.
4. Блок поста складається з таких даних: ім'я користувача, дата додавання, аватар, заголовок, текст, кількість переглядів та коментарів, популярність посту, мітки, по центру знаходиться блок із створеним SVG або власним зображенням.
5. На іншій сторінці знаходиться блок для створення дописів.

2.3.3 Блок Авторизації

Містить посилання: LOGIN та SIGNUP на сторінки в яких знаходяться форми для авторизації.

Перша сторінка – це сторінка для реєстрації нових користувачів.

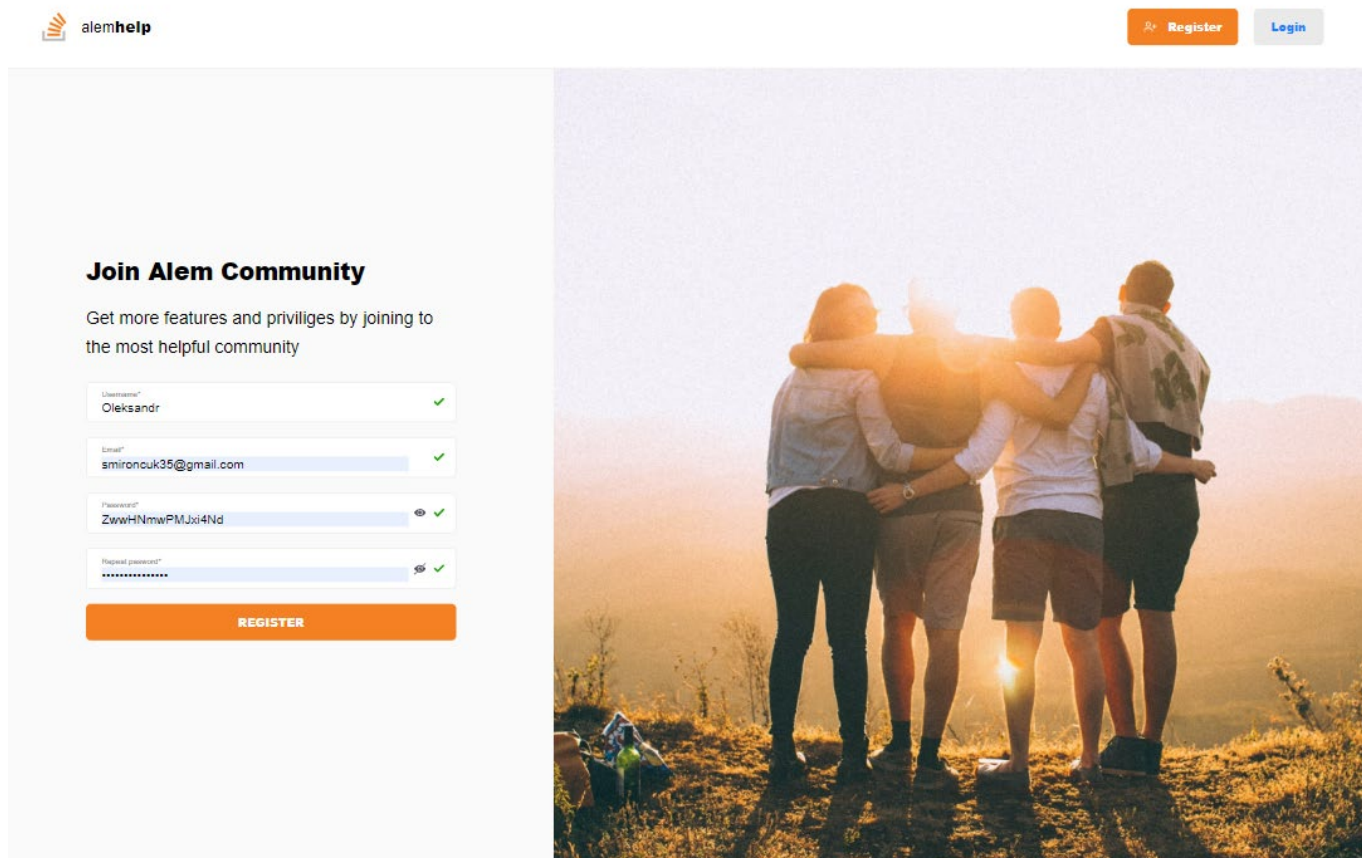


Рисунок 1 - Сторінка авторизації

Друга сторінка – це сторінка авторизації для зареєстрованих користувачів.

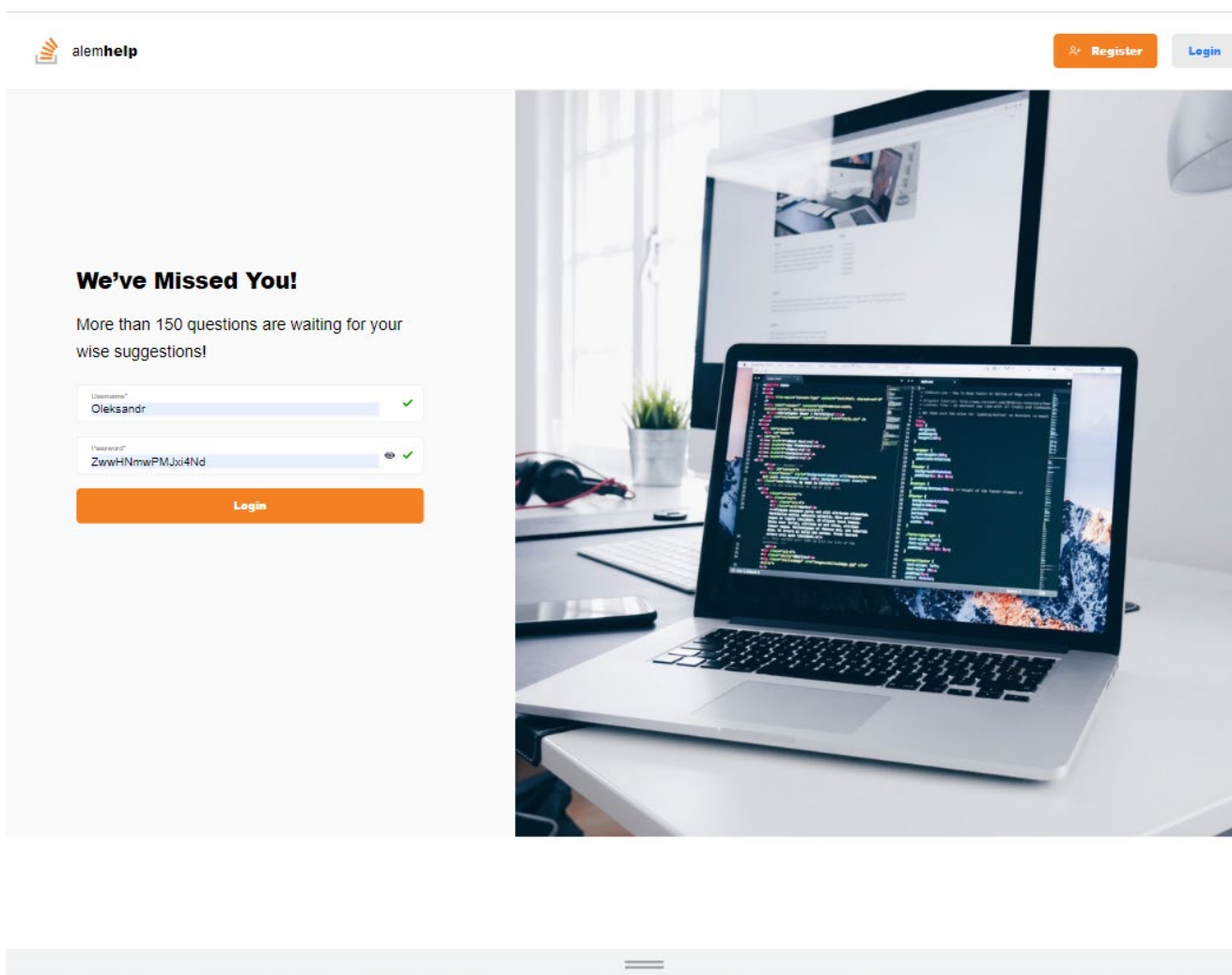


Рисунок 2 - Сторінка авторизації

2.3.4 Шапка додатку

Має 2 стани:

Не авторизований:



Рисунок 3 – Шапка додатку (не авторизований)

Авторизований:



Рисунок 4 - Шапка додатку (авторизований)

2.3.5 Блок для відображення усіх постів.

Складається з таких даних: ім'я користувача, дата додавання, аватар, заголовок, текст, кількість переглядів та коментарів, популярність посту, мітки, по центру знаходиться блок із створеним SVG або власним зображенням.

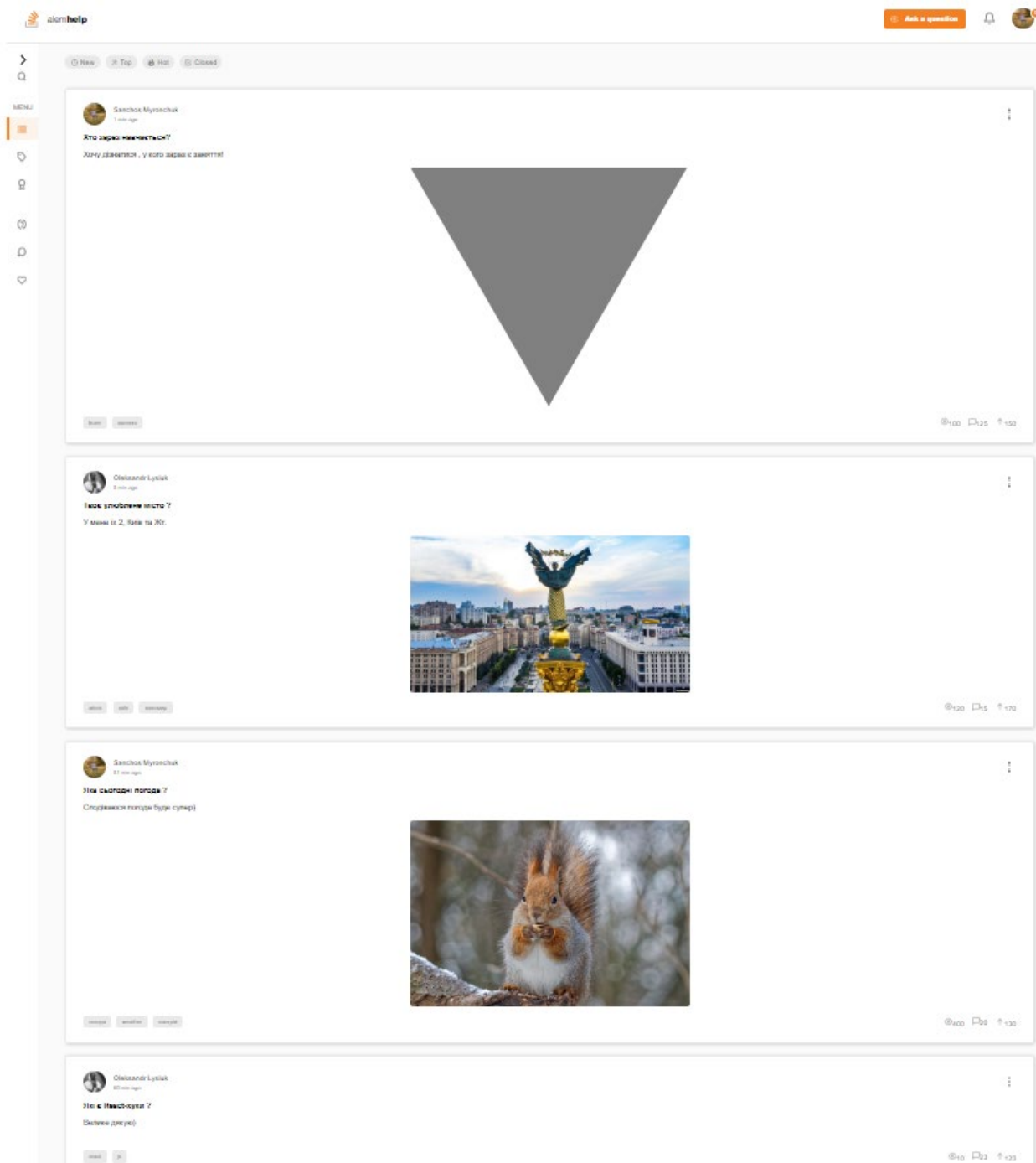


Рисунок 5 - Блок для відображення усіх постів

2.3.6 Сайдбар

Цей блок складається з пошуку та 3 основних пунктів меню і 3, які є доступними тільки для авторизованих користувачів.

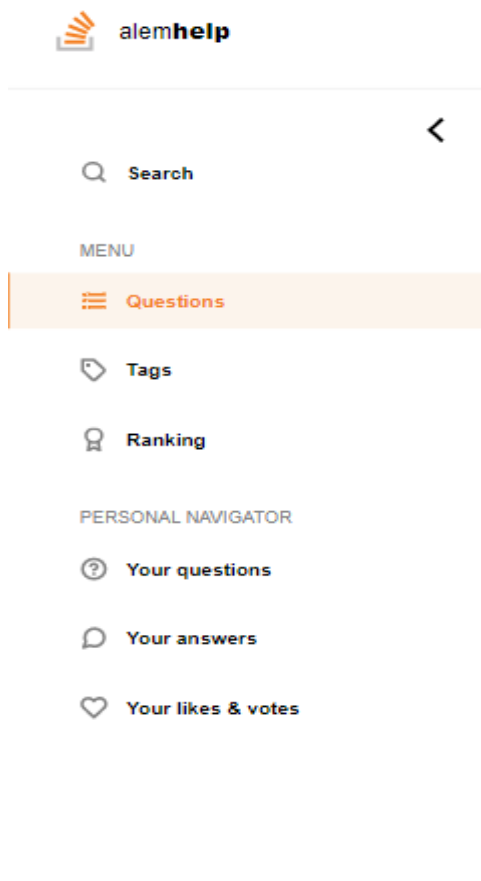


Рисунок 6 – Сайдбар (розгорнутий)

Сайдбар у згорнутому вигляді:

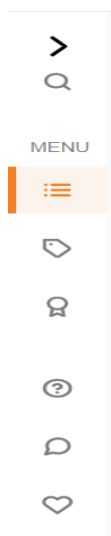


Рисунок 7 - Сайдбар (згорнутий)

2.3.7 Блок для створення дописів

Складається з 3 полів для даних посту та 3 кнопок:

Вибір категорії, заголовок та текст допису

Кнопки збереження публікації, публікування, та додавання існуючого або створення свого зображення.

Рисунок 8 - Блок для створення дописів

2.3.8 Графічний редактор додатку:

Відкривається на кнопку «Add Image» або «Edit Image».

Складається з панелі інструментів, яка містить готовий набір геометричних фігур в якому ми можемо вибирати і переносити в робочу область будь які об'єкти за допомогою кліку по потрібній нам фігурі та затиснувши на полотні праву клавішу миші визначити розмір фігури. В панелі є також можливість вибрати інструменти для довільного малювання, додавання тексту в робочу область, вибору кольору для фігури чи інструменту, визначення кольору об'єкта та інші.

Робоча область редактора містить біле полотно, в якій ми і відображаємо весь наш контент рисунка.

Права частина відображає поточний розмір полотна та дозволяє нам вибрати із списку розмір робочої області, задати, фіксовану висоту і ширину, так і залишити її довільною.

Маємо змогу також визначити колір фону чи встановити його повністю прозорим для нашої SVG.

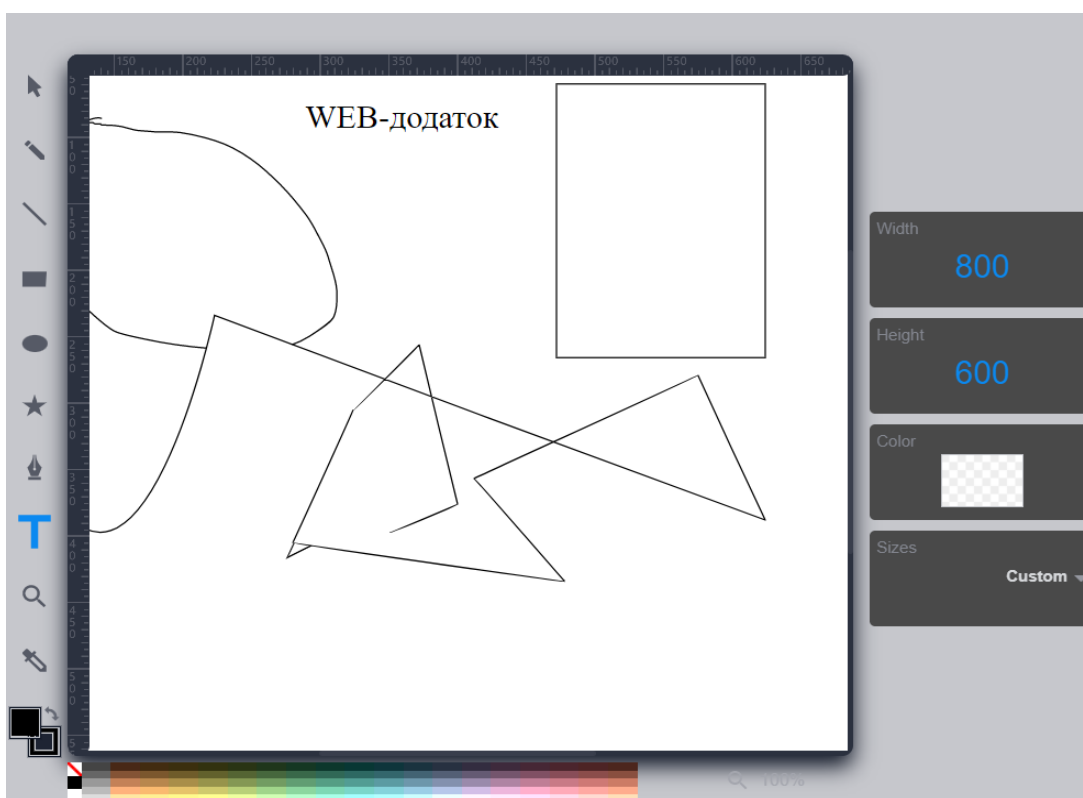


Рисунок 9 - Графічний редактор

2.3.9 Блок сортування:

Складається з 4 параметрів:

New – дата створення

Top – популярність

Hot – важливість

Closed - актуальність

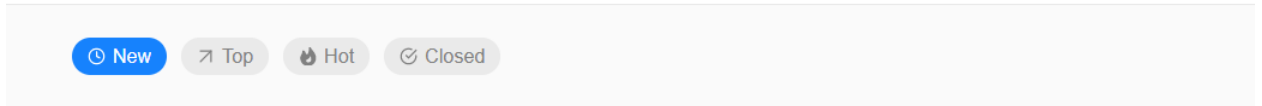


Рисунок 10 - Сортування

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням:

- HTML
- CSS
- SCSS
- JavaScript
- React.js
- Redux

2.4.2 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка Javascript і cookies.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

3.1 Засоби та технології розробки

Додаток розробляється у середовищі Visual Studio Code.

Visual Studio Code — це легкий, але потужний редактор вихідного коду, який працює на вашому робочому столі та доступний для Windows, macOS та Linux. Він поставляється з вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов (наприклад, C++, C#, Java, Python, PHP, Go) і середовищ виконання (наприклад, .NET і Unity).

VS Code — це безкоштовний редактор коду, який працює в операційних системах macOS, Linux і Windows. – [19]

Розробка проекту відбувається за допомогою **HTML, CSS, SCSS, JavaScript** бібліотек **React, Redux, StyledComponents** та інших додаткових.

Мова **розмітки гіпертексту** або **HTML** є стандартною мовою розмітки для документів, призначених для відображення у веб-браузері. Цьому можуть допомогти такі технології, як каскадні таблиці стилів (CSS) і мови сценаріїв, такі як JavaScript.

Веб-браузери отримують документи HTML з веб-сервера або з локального сховища і перетворюють документи на мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично та оригінально вклав підказки для зовнішнього вигляду документа. - [12]

CSS - це мова таблиць стилів, яка використовується для опису подання документа, написаного мовою розмітки, як-от HTML. CSS є наріжною технологією всесвітньої мережі, поряд з HTML і JavaScript.

CSS розроблено для того, щоб уможливити розділення презентації та вмісту, включаючи макет, кольори та шрифти. Це поділ може покращити доступність вмісту ; забезпечити більшу гнучкість і контроль у специфікації характеристик презентації; дозволити кільком веб-сторінкам спільно використовувати форматування, вказавши відповідний CSS в окремому

файлі.css, що зменшує складність і повторюваність структурного вмісту; і ввімкніть кешування файлу.css, щоб покращити швидкість завантаження сторінки між сторінками, які спільно використовують файл, та його форматування.

Розділення форматування та вмісту також дає можливість представити одну й ту саму сторінку розмітки в різних стилях для різних методів візуалізації, наприклад, на екрані, у друкованому вигляді, голосом (через мовленнєвий браузер або програму зчитування з екрана), а також на основі шрифту Брайля. тактильні пристрої. CSS також має правила альтернативного форматування, якщо доступ до вмісту здійснюється на мобільному пристрої.

Каскадність назв походить від заданої схеми пріоритету, щоб визначити, яке правило стилю застосовується, якщо більше одного правила відповідає певному елементу. Ця каскадна схема пріоритетів є передбачуваною.

Специфікації CSS підтримуються Консорціумом World Wide Web Consortium (W3C). Тип Інтернет-медіа (тип MIME) `text/css` зареєстровано для використання з CSS згідно RFC 2318 (березень 1998 р.). W3C надає безкоштовну службу перевірки CSS для документів CSS.

На додаток до HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, звичайний XML, SVG і XUL. - [12]

Sass (скорочення від *синтаксично чудових таблиць стилів*) — це мова сценаріїв препроцесора, яка інтерпретується або компілюється в каскадні таблиці стилів (CSS). SassScript - це сама мова сценаріїв.

Sass складається з двох синтаксисів. Оригінальний синтаксис, який називається «синтаксис із відступом», використовує синтаксис, подібний до Haml. Він використовує відступ для розділення кодових блоків і символи нового рядка для розділення правил. Новий синтаксис «SCSS» (Sassy CSS) використовує блочное форматування, як у CSS. Він використовує дужки для позначення блоків коду і крапку з комою для розділення правил всередині

блоку. Синтаксис з відступом і файли SCSS традиційно мають розширення.sass і.scss відповідно.

CSS3 складається з серії селекторів і псевдоселекторів, які групують правила, які до них застосовуються. Sass (у більшому контексті обох синтаксисів) розширює CSS, надаючи кілька механізмів, доступних у більш традиційних мовах програмування, зокрема в об'єктно-орієнтованих мовах, але вони недоступні самому CSS3. Коли SassScript інтерпретується, він створює блоки правил CSS для різних селекторів, визначених файлом Sass. Інтерпретатор Sass перекладає SassScript в CSS. Крім того, Sass може контролювати файл.sass або.scss і переводити його у вихідний файл.css щоразу, коли файл.sass або.scss зберігається.

Синтаксис із відступом є метамовою. SCSS є вкладеною метамовою, оскільки дійсний CSS є дійсним SCSS з такою ж семантикою.

SassScript надає такі механізми: змінні, вкладення, міксини та успадкування селектора. - [12]

JavaScript — **JS**, є мовою програмування, яка є однією з основних технологій всесвітньої мережі, поряд з HTML і CSS. Понад 97% веб- сайтів використовують JavaScript на стороні клієнта для поведінки веб-сторінки, часто включають сторонні бібліотеки. Усі основні веб-браузери мають спеціальний механізм JavaScript для виконання коду на пристроях користувачів.

JavaScript — це мова високого рівня, яка часто скомпільована «точно вчасно », яка відповідає стандарту ECMAScript. Він має динамічну типізацію, орієнтацію на прототип і першокласні функції. Це багатопарадигмальний, підтримує керовані подіями, функціональні та імперативний стилі програмування. Він має інтерфейси прикладного програмування (API) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних та об'єктною моделлю документа. (DOM).

Стандарт ECMAScript не містить жодних засобів введення/виводу (I/O), таких як мережа зберігання чи графічні засоби. На практиці веб-браузер або інша система виконання надає API JavaScript для введення-виводу.

Механізми JavaScript спочатку використовувалися лише у веб-браузерах, але тепер вони є основними компонентами деяких серверів і різноманітних програм. Найпопулярнішою системою виконання для такого використання є Node.js. - [12]

React - JavaScript-бібліотека для створення користувацьких інтерфейсів.

Декларативний

React дозволяє безболісно створювати інтерактивні інтерактивні інтерфейси. Створюйте прості представлення для кожного стану у програмі, і React ефективно оновлюватиме й відобразить потрібні компоненти, коли ваші дані змінюються.

Декларативні уявлення роблять код більш передбачуваним і легшим для налагодження.

Компонентний

Створюйте інкапсульовані компоненти, які керують своїм власним станом, а потім компонуйте їх, щоб створити складні інтерфейси.

Оскільки логіка компонентів написана на JavaScript, а не на шаблонах, можна легко передавати розширені дані через додаток і зберігати стан поза межами DOM. - [4]

Redux - це передбачуване сховище стану для доданого JavaScript.

Він допомагає писати додатки, які ведуть передбачувано, працюють у різних оточеннях (клієнтських, серверних та нативних додатків) і які легко тестувати. Крім того, він надає відмінні можливості для розробників, такі як редагування коду в реальному часі в поєднанні з налагоджувачем у часі.

Redux використовується разом із React або будь-якою іншою view-бібліотекою. Це малесенька бібліотека (2kB, включаючи залежності).

Весь глобальний стан програми зберігається в дереві об'єктів всередині одного стору (**store**). Єдиний спосіб змінити дерево станів — викликати екшен (**action**), об'єкт, що описує те, що сталося, і відправити його до стору.

Щоб визначити, як стан оновлюється у відповідь на екшен, потрібно писати чисті функції- редутори (reducer), які обчислюють новий стан на основі старого стану та екшена. - [7]

StyledComponents

Використовуючи теговані шаблонні літерали (недавнє доповнення до JavaScript) і потужність CSS, styled-components дозволяє писати фактичний код CSS для стилізації компонентів. Він також виконує видалення зіставлення між компонентами та стилями – використання компонентів як низькорівневої конструкції стилю не може бути простіше. - [9]

3.2 Моделювання роботи web-додатку для протоколювання наради з обговорення проектних рішень

Концептуальне моделювання застосування WEB -додатку для протоколювання наради з обговорення проектних рішень виконаємо на основі аналізу організації робіт підрозділу підприємства з проектування продукції.

Нарада, як колективний орган, повинна бути продуктивною. Інакше немає сенсу її організувати та проводити. Отримане на нараді деяке проектне рішення фіксується у протокалі наради. Ці рішення формується на основі проектних пропозицій учасників наради. Запропоновані проектні рішення не можуть бути довільними. Вони повинні відповідати як об'єкту проектування, так і встановленим вимогам до об'єкту і можливостей підприємства-виробника. Дотримання державних стандартів, вимог законодавства передбачається по замовчуванню. Учасники наради повинні буди зосереджені на суті вирішуваної задачі і якомога швидше генерувати змістовні проектні рішення. Фіксацію самих пропозицій, як роботу не творчу, але кропітку, пропонується передати спеціальному web-додатку.

Ці положення зафіксуємо у контекстній діаграмі проведення наради з приводу вирішення проектної задачі (рисунок 11).

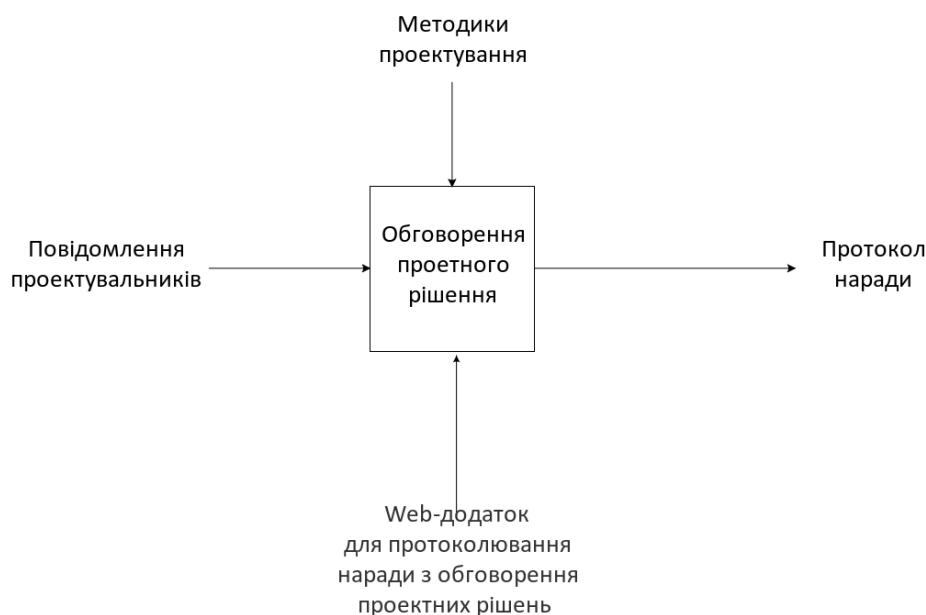


Рисунок 11 – Контекстна діаграма моделювання роботи наради з вирішення проектних задач

Діаграма функціонування WEB -додатку для протоколювання наради з обговорення проектних рішень (рисунок 12). Цей контекст передбачає, що найбільш розповсюдженими формами виразу людських пропозицій є текстова та графічна. Відтак на WEB –додаток покладаємо обробку одночасну текстової та графічної інформації.



Рисунок 12 – Контекстна діаграма роботи web-додатку

Декомпозиція контекстної діаграми WEB-додатку для протоколювання наради з обговорення проектних рішень показана на рисунку 13.

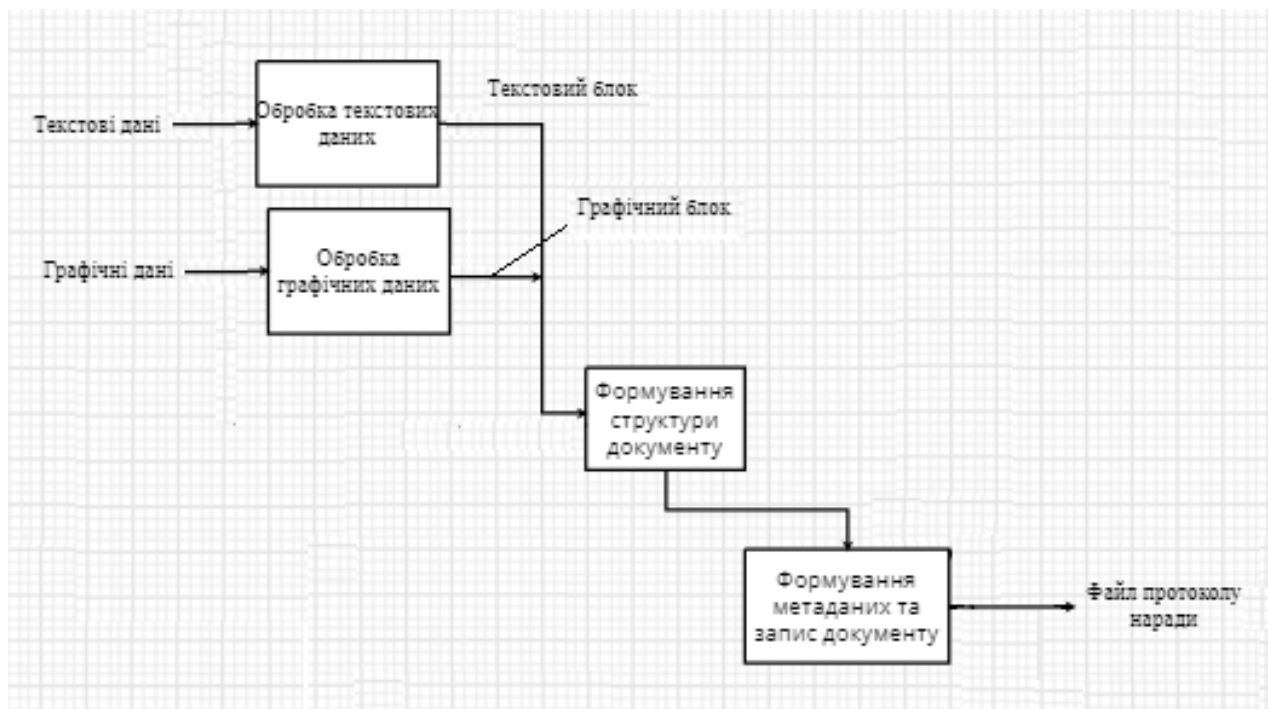


Рисунок 13 – Декомпозиція контекстної діаграми роботи web-додатку

Процедури визначених задач та учасники наради взаємодіють так, як показано на рисунку 14.

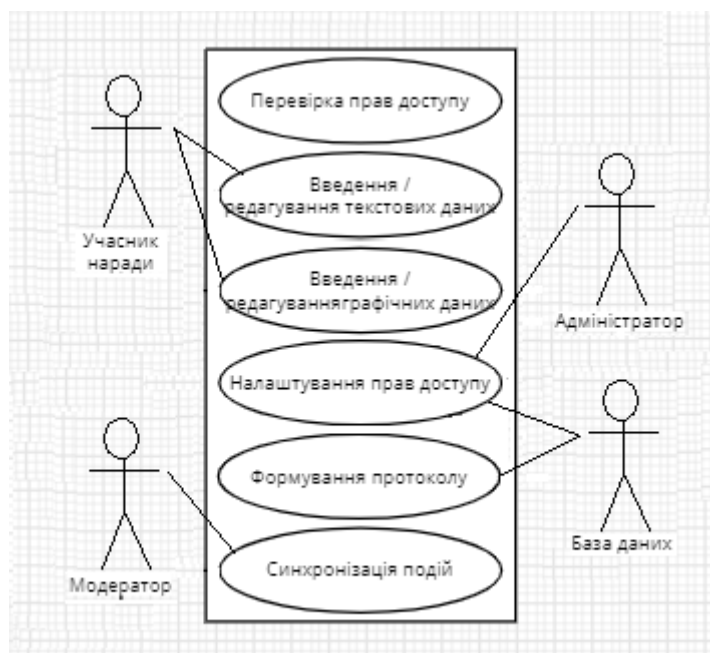


Рисунок 14 – Діаграма варіантів використання web-додатку

Розроблені моделі дозволяють перейти до створення Web-додатку для протоколювання наради з обговорення проектних рішень.

3.3 Створення та перший запуск проекту Web-додатку

Для створення React додатку використовується інструмент **Create React App**.

Create React App — це комфортне середовище для **вивчення React**, а також це найкращий шлях щоб почати будувати **нові односторінкові додатки** за допомогою React.

Він встановлює осередок для розробки таким чином, щоб ви могли використовувати найновіші можливості JavaScript, робить розробку комфортнішою, а також оптимізує ваш додаток для продакшну.

Для працездатності, потрібно встановити на комп'ютер **Node** та **npm** не нижче версій **8.10** та **5.6** відповідно. - [4]

Для створення проекту потрібно виконати наступну команду:

npm create-react-app my-app

Для того , щоб запуснути додаток потрібно перейти у папку проекту та виконати команду ***npm start***

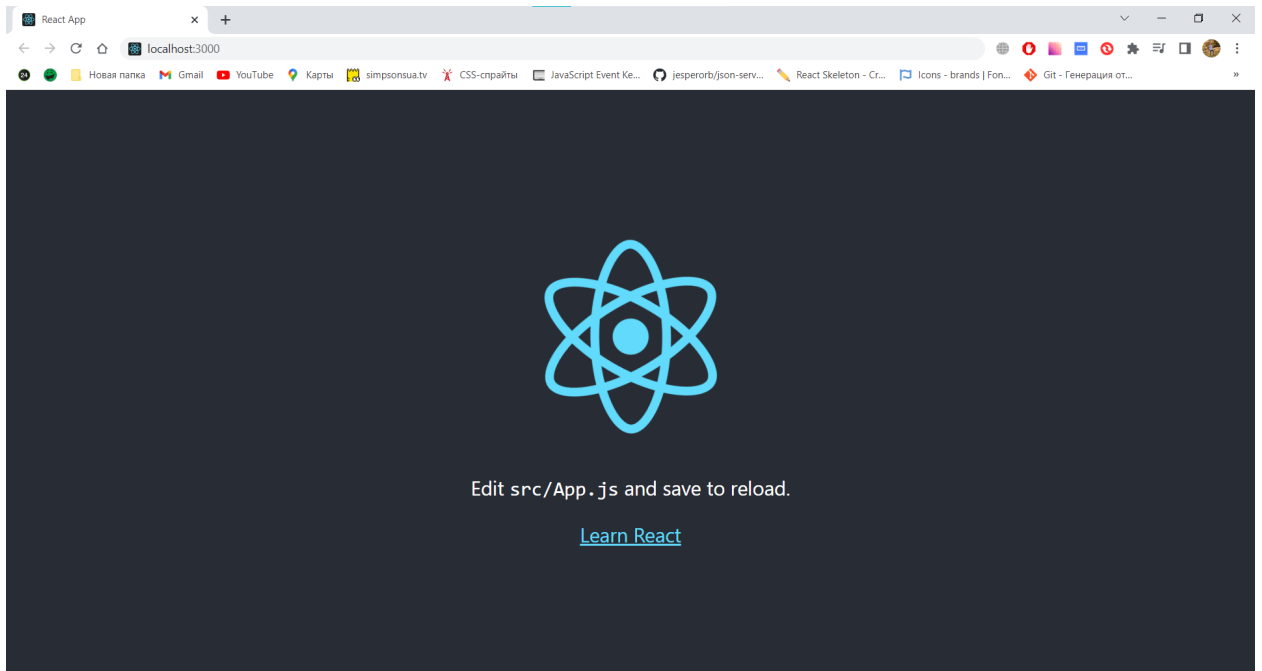


Рисунок 15 – Запуск React-додатку

Проект буде розгорнуто на локальному сервері на комп'ютері.

За замовчуванням порт локального хосту: `http://localhost:3000/` (рисунок 16).

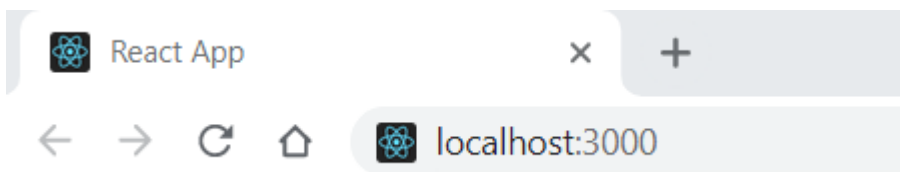


Рисунок 16 – Локальний хост

Якщо ця адреса вже зайнята, буде запропоновано використати іншу вільну (рисунок 17).

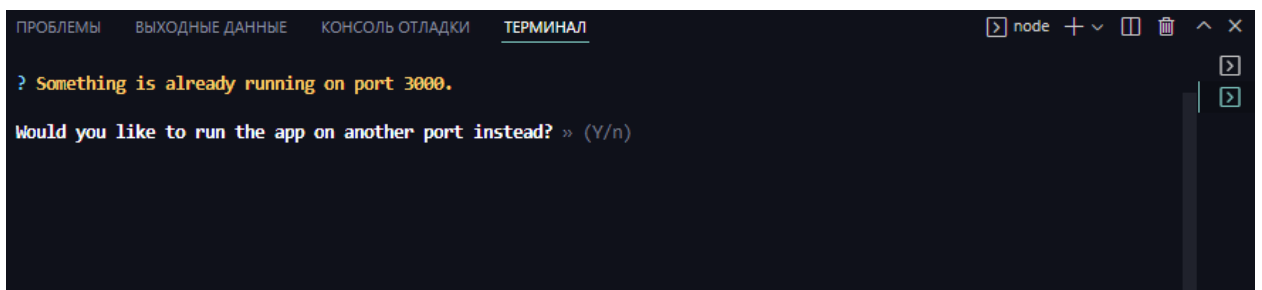


Рисунок 17 – Командний рядок (зміна адреси)

За бажанням її можна змінити на іншу, наприклад: `http://localhost:5000/`

3.4 Загальне представлення об'єктів.

Веб-застосунок складається з кількох частин основних частин: Шапки (Header), яка в свою чергу складається з двох інших блоків – логотипу та блоку авторизації з двома посиланнями, які ведуть на сторінки реєстрації та авторизації або блоку авторизованого користувача з ім'ям та аватаром із випадаючим меню і кнопкою для створення публікації.

Основного блоку розділеного на дві частини.

В першій частині зліва знаходиться сайдбар з полем пошуку, та меню із трьох основних пунктів, та трьох додаткових, які доступні тільки для авторизованих юзерів.

В другій частині міститься контент-блок, в якому знаходяться усі публікації користувачів, з можливістю сортування за різними параметрами (дата, популярність, актуальність, важливість).

Публікація складається з аватару автора, псевдоніму або ім'я, заголовку, тексту запитання чи проблеми, мітки, кількості переглядів, коментарів та популярність.

Щоб редагувати зображення, яке створив користувач або завантажив готове потрібно натиснути 3 крапки і кнопку «Edit Image».

При натисканні на кнопку відкривається редактор.

Він складається з панелі інструментів, яка містить готовий набір геометричних фігур в якому ми можемо вибирати і переносити в робочу область будь які об'єкти за допомогою кліку по потрібній нам фігурі та затиснувши на полотні праву клавішу миші визначити розмір фігури. В панелі є також можливість вибрати інструменти для довільного малювання, додавання тексту в робочу область, вибору кольору для фігури чи інструменту, визначення кольору об'єкта та інші.

Робоча область редактора містить біле полотно, в якій ми і відображаємо весь наш контент рисунка.

Права частина відображає поточний розмір полотна та дозволяє нам вибрати із списку розмір робочої області, задати, фіксовану висоту і ширину, так і залишити її довільною.

Маємо змогу також визначити колір фону чи встановити його повністю прозорим для нашої SVG.

інші.

Робоча область редактора містить біле полотно, в якій ми і відображаємо весь наш контент рисунка.

Загальний вигляд додатку показано на рисунку 18.

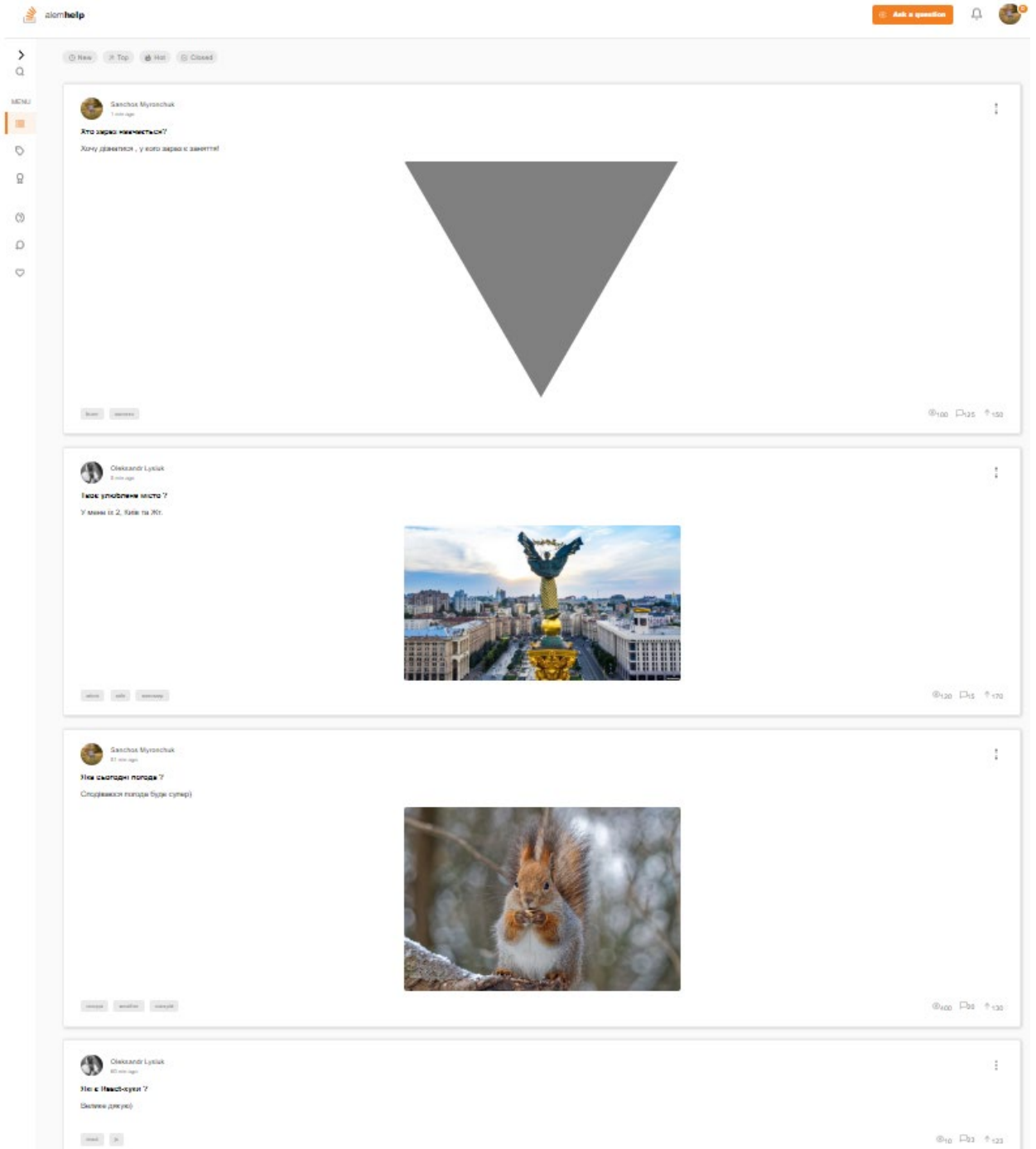


Рисунок 18 – Загальний вигляд Web-додатку для протоколювання наради

ВИСНОВКИ

В результаті реалізації даного проекту було проаналізовано предметну область використання web-додатку для протоколювання наради з обговорення проектних рішень, розглянуто аналоги подібного продукту та останні публікації з подібною проблематикою, сформульовано та поставлено задачі на проектування та створення програмного продукту, визначено цільову аудиторію, засоби та технології для розробки, вимоги до видів забезпечення, побудовано повноцінну структуру додатку.

Було розроблено застосунок який дає змогу обговорювати та протоколювати різні проектні рішення, створювати SVG-зображення в графічному редакторі, з подальшою можливістю їх редагування.

В кінцевому результаті маємо web-додаток для протоколювання наради з обговорення проектних рішень.

У подальшому web-додаток може удосконалюватися у напрямку розпізнавання аудіо потоку і включення цього тексту у протокол наради.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дубовик А.В. Web на практиці. CSS, HTML, JavaScript, MySQL, PHP для fullstack-розробників. / А.В.Дубовик, Е.В.Никольський, А.П. Кириченко – Видавництво Наука і Техніка, 2021 – 432 с.
2. Никсон Р. Створюємо динамічні веб-сайти з допомогою PHP, MySQL, JavaScript, CSS і HTML5/ Р.Никсон. 5-е вид. – СПб: Пітер, 2019 – 816 с.
3. Фрэйн Б. HTML5 і CSS3. Розробка сайтів для будь-яких браузерів і пристроїв. 2-е вид. — СПб.: Пітер, 2017. — 272 с.
4. React [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://uk.reactjs.org/docs/getting-started.html> Дата доступу: 7.04.2022.
5. React Waypoint – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://www.npmjs.com/package/react-waypoint> Дата доступу: 7.04.2022.
6. Стефанов С. React.js. Швидкий старт / Стоян Стефанов. – СПб: Пітер, 2016. – 304 с.
7. Redux [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://redux.js.org/introduction/getting-started> Дата доступу: 7.04.2022.
8. API React – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://v5.reactrouter.com/web/guides/quick-start> Дата доступу: 8.04.2022.
9. Використання шаблонів літералів – [Електронний ресурс]:[Сайт] – Режим доступу: вільний <https://styled-components.com/docs/tooling> Дата доступу: 8.04.2022.
10. Axios- клієнт HTTP – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://axios-http.com/uk/docs/intro> Дата доступу: 8.04.2022.
11. JavaScript – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://uk.javascript.info>— Дата доступу: 9.04.2022..
12. Socket.IO – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://socket.io/get-started/chat> Дата доступу: 9.04.2022.

13. – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://uk.wikipedia.org/> Дата доступу: 9.04.2022.
14. Ресурси для розробників, від розробників – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://developer.mozilla.org> Дата доступу: 9.04.2022.
15. Підручник HTML – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <http://htmlbook.ru/samhtml> Дата доступу: 11.04.2022.
16. Сайт для тих, хто вивчає веб-технології та створює сайти – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://html5book.ru/> Дата доступу: 11.04.2022.
17. Font Awesome – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://fontawesome.com> Дата доступу: 11.04.2022.
18. Векторні значки і наклейки – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://www.flaticon.com> Дата доступу: 12.04.2022.
19. Visual Studio Code – [Електронний ресурс]:[Сайт] –Режим доступу: вільний <https://code.visualstudio.com/docs> Дата доступу: 12.04.2022.
20. Bruce Johnson. Visual Studio Code. End-to-End Editing and Debugging Tools for Web Developers – John Wiley & Sons P&T, 2019 – 192 Pages.
21. Alessandro Del Sole Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux – Apress, 2019 – 221 Pages.

Додатки

ДОДАТОК А

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Технічне завдання
на створення програмного продукту**

**«Web-додаток для протоколювання наради з обговорення проектних
рішень»**

1. Призначення та мета Web-додатку

1.1 Призначення web-додатку

Застосунок дає змогу обговорювати різні питання з вашими знайомими, друзями, рідними, колегами по роботі чи іншими людьми, але його значна перевага в тому, що він дає можливість не просто спілкуватися, а відобразити свої думки в графічному редакторі, який дає змогу малювати та створювати SVG –зображення з подальшою можливістю редагування іншими користувачами, які мають до нього доступ.

1.2 Мета створення web-додатку

Метою даного дослідження є розробка web-додатку для протоколювання наради з обговорення проектних рішень.

1.3 Цільова аудиторія

У цільовій аудиторії інформаційної системи можна виділити наступні групи:

12. Програмісти.
13. Менеджери.
14. Дизайнери.
15. Блогери
16. Студенти
17. Школярі
18. Викладачі

19. Офісні працівники
20. Аналітики
21. Тестувальники
22. Організації у яких часто відбуваються конференції.

2. Вимоги до Web-додатку

2.1 Основні вимоги до WEB- додатку

Веб-додаток повинен мати вигляд звичайного сайту-форуму, але розділеного на кілька частин з можливістю авторизації, додаванням та редагуванням дописів, та рисунків у цих дописах, коментуванням інших публікацій користувачів.

Користувач повинен мати можливість створення та редагування SVG-зображень, за допомогою панелі інструментів в додатку. Інтерфейс має бути інтуїтивно зрозумілим, дозволяти нанесення довільних ліній та мати набір для використання готових геометричних фігур в редакторі таких як лінії, квадрати, круги, овали тощо та можливість переміщати їх за необхідності, видаляти та редагувати, додавати текст.

2.2 Структура додатку

Застосунок складається з наступних елементів:

1. У шапці додатку знаходиться логотип додатку зліва та блок для авторизація або профіль користувача з права із кнопкою для створення публікації.

2. В основному контейнері з лівого боку знаходиться сайдбар який відкривається/закривається (за замовчуванням закритий) в якому знаходиться навігаційне меню додатку.

3. По центру знаходиться блок для відображення усіх постів.

4. Блок поста складається з таких даних: ім'я користувача, дата додавання, аватар, заголовок, текст, кількість переглядів та коментарів, популярність посту, мітки, по центру знаходиться блок із створеним SVG або власним зображенням.

5. На іншій сторінці знаходиться блок для створення дописів.

2.3 Блок Авторизації

Містить посилання: LOGIN та SIGNUP на сторінки в яких знаходяться форми для авторизації.

Перша сторінка – це сторінка авторизації для зареєстрованого користувач.

Друга сторінка – це сторінка для реєстрації нових користувачів.

2.4 Шапка додатку

Шапка додатку має 2 стани:

Не авторизований та авторизований:

2.5 Блок для відображення усіх постів.

Складається з таких даних: ім'я користувача, дата додавання, аватар, заголовок, текст, кількість переглядів та коментарів, популярність посту, мітки, по центру знаходиться блок із створеним SVG або власним зображенням.

2.6 Сайдбар

Цей блок складається з пошуку та 3 основних пунктів меню і 3, які є доступними тільки для авторизованих користувачів.

2.7 Блок для створення дописів

Складається з 3 полів для даних посту та 3 кнопок:

Вибір категорії, заголовок та текст допису

Кнопки збереження публікації, публікування, та додавання існуючого або створення свого зображення.

2.8 Графічний редактор додатку

Відкривається на кнопку «Add Image» або «Edit Image».

Складається з панелі інструментів, яка містить готовий набір геометричних фігур в якому ми можемо вибирати і переносити в робочу область будь які об'єкти за допомогою кліку по потрібній нам фігурі та затиснувши на полотні праву клавішу миші визначити розмір фігури. В панелі є також можливість вибрати інструменти для довільного малювання, додавання тексту в робочу область, вибору кольору для фігури чи інструменту, визначення кольору об'єкта та інші.

Робоча область редактора містить біле полотно, в якій ми і відображаємо весь наш контент рисунка.

Права частина відображає поточний розмір полотна та дозволяє нам вибрати із списку розмір робочої області, задати, фіксовану висоту і ширину, так і залишити її довільною.

Маємо змогу також визначити колір фону чи встановити його повністю прозорим для нашої SVG.

2.9 Блок сортування:

Блок сортування має з 4 параметрів:

New – дата створення

Top – популярність

Hot – важливість

Closed – актуальність

3. Вимоги до видів забезпечення

3.3.1 Вимоги до інформаційного забезпечення

Реалізація сайту відбувається з використанням:

- HTML
- CSS
- SCSS
- JavaScript
- React.js
- Redux

3.3.2 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка Javascript і cookies.

ДОДАТОК Б

Планування робіт

У на час є багато різних месенджерів, соціальних мереж, форумів для обговорення різних питань, сервісів відео-телефонного зв'язку та інших веб-додатків і сайтів. З кожним роком таких веб-застосунків та сайтів стає все більше і більше.

Часто потрібен такий додаток, що дозволив би нам не тільки спілкуватися в чаті чи по відеозв'язку, а який дає змогу для графічного відображення та пояснення. Який дає змогу намалювати та зобразити свої думки і донести їх іншим.

Такий, який замінить вашу оффлайн конференцію на онлайн та навіть стане краще за неї.

Є рішення - це «Web-додаток для протоколювання наради з обговорення проектних рішень».

Як було сказано вище, цей веб-застосунок дає змогу спілкуватися з вашими знайомими, друзями, рідними, колегами по роботі чи іншими людьми, але його значна перевага в тому, що він дає можливість не просто спілкуватися, а відображати свої думки в графічному редакторі, який дає змогу малювати та створювати SVG –зображення з подальшою можливістю редагування іншими користувачами, які мають до нього доступ.

Так його функціонал дуже схожий до звичайного форуму, але це більше ніж просто форум, тому, що він відкриває для вас більше можливостей і функціоналу та інструментів.

Він змінить ваші звичні та нудні конференції по обговоренню, на справжній прорив.

Виглядає все це дуже просто, як звичайна дошка для малювання у кабінеті чи залі на якій ви щось пишете чи малюєте, але цю дошку може бачити

безмежна кількість людей та кожен має можливість її редагувати не встаючи зі свого місця та не перебуваючи з вами в одному приміщенні.

Тому цей додаток настільки крутий і кращий за інші аналоги.

Масштабована векторна графіка (SVG) — це формат векторного зображення на основі XML для двовимірної графіки з підтримкою інтерактивності та анімації. Специфікація SVG є відкритим стандартом, розробленим Консорціумом World Wide Web Consortium (W3C) з 1999 року.

Зображення SVG визначаються у форматі векторної графіки та зберігаються у текстових файлах XML. Таким чином, зображення SVG можна масштабувати за розміром без втрати якості, а файли SVG можна шукати, індексувати, створювати сценарії та стискати. Текстові файли XML можна створювати та редагувати за допомогою текстових редакторів або векторних графічних редакторів, і вони відтворюються найбільш часто використовуваними веб-браузерами. -[12]

Деталізація мети проекту методом SMART. Для збільшення конкурентноздатності та успішності проекту необхідно коректно визначити його мету, що можна зробити за допомогою SMART-методу. Деталізований результат методу зображено в таблиці Б.1.

Планування змісту робіт. Ключовим результатом проекту, що розбиває командну роботу на відслідковувані та керовані частини – це WBS (Work Breakdown Structure – Ієрархічна структура робіт). Іншими словами, це візуальний план компонентів проекту, які згруповано ієрархією. Також WBS надає необхідний вигляд для оцінки термінів та контролю за графіком виконання.

Структурно, на найвищому рівні розміщено кінцевий результат проекту. На другому рівні знаходяться різноманітні дії та заходи для досягнення мети проекту. Таким чином декомпозиція відбувається до моменту, коли дії стають атомарними і мають чіткий та однозначний результат і мають одного виконавця для якого можна розрахувати витрати на працю і часові затрати. На рисунку Б.1 зображено WBS з розробки додатку.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Розробити всі компоненти web-додатку та протестувати їх роботу
Measurable (вимірювана)	Розробити визначені компоненти до кінця 4 курсу, використовуючи мінімальний об'єм ресурсів необхідний для виконання завдання.
Achievable (досяжна, узгоджена)	Для розробки проекту потрібні знання HTML, CSS, SCSS-препроцесора, мови програмування JavaScript і її бібліотек: React та Redux
Relevant (реалістична)	Створений web-додаток буде надавати функціонал для спілкування, публікування дописів з можливістю створювати власні SVG-зображення та подальше їх редагування.
Time-framed (обмежена в часі)	Термін досягнення мети проекту визначено за навчальним планом до кінця червня 2022 року.

Планування структури виконавців. Наступним кроком після декомпозиції процесів є розробка організаційної структури виконавців або ще OBS, яка зображується як графічна структура, що відображає учасників проекту відповідальних за певний етап реалізації.

До відповідальних осіб відносяться співробітники, що виконують елементарну роботу зазначену в OBS.

На рисунку Б.2 зображено організаційну структуру планування проекту

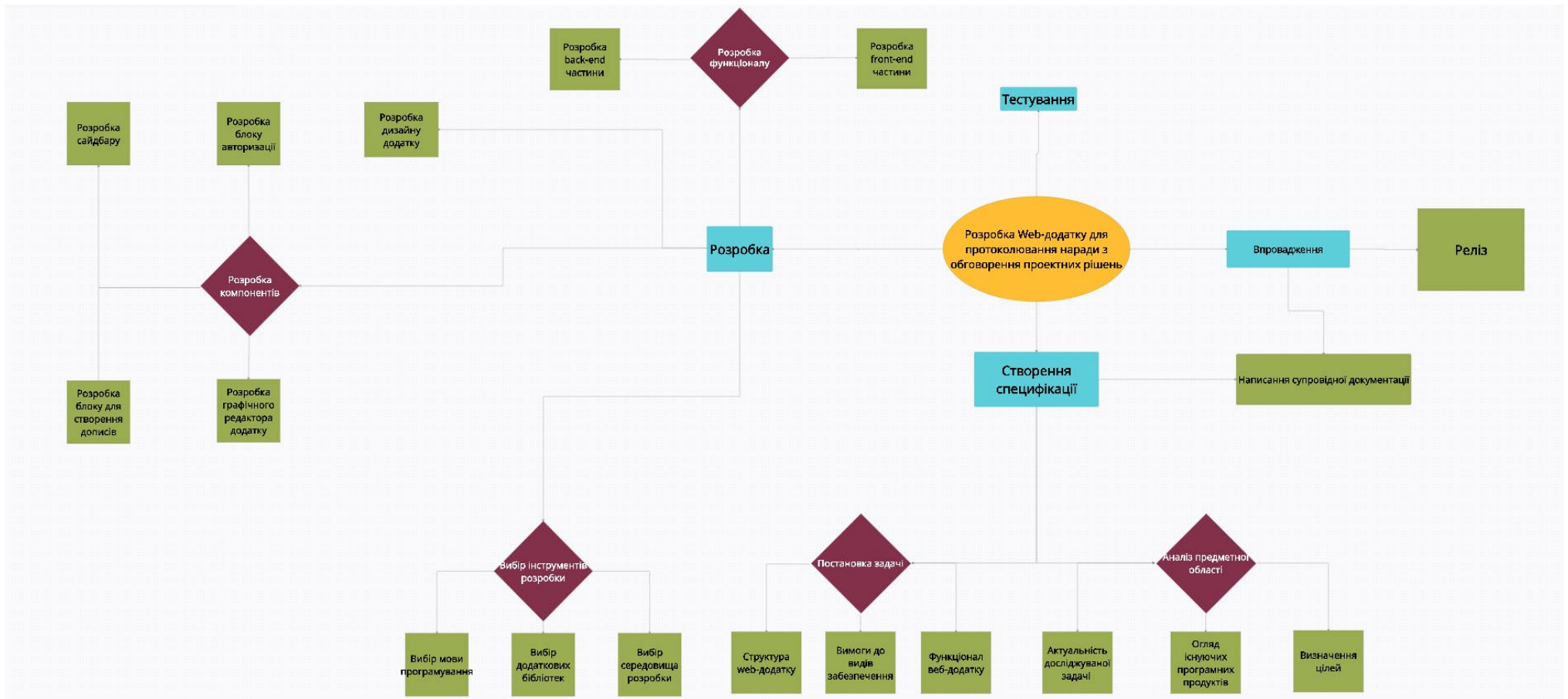


Рисунок Б.1 – WBS-структура робіт проекту

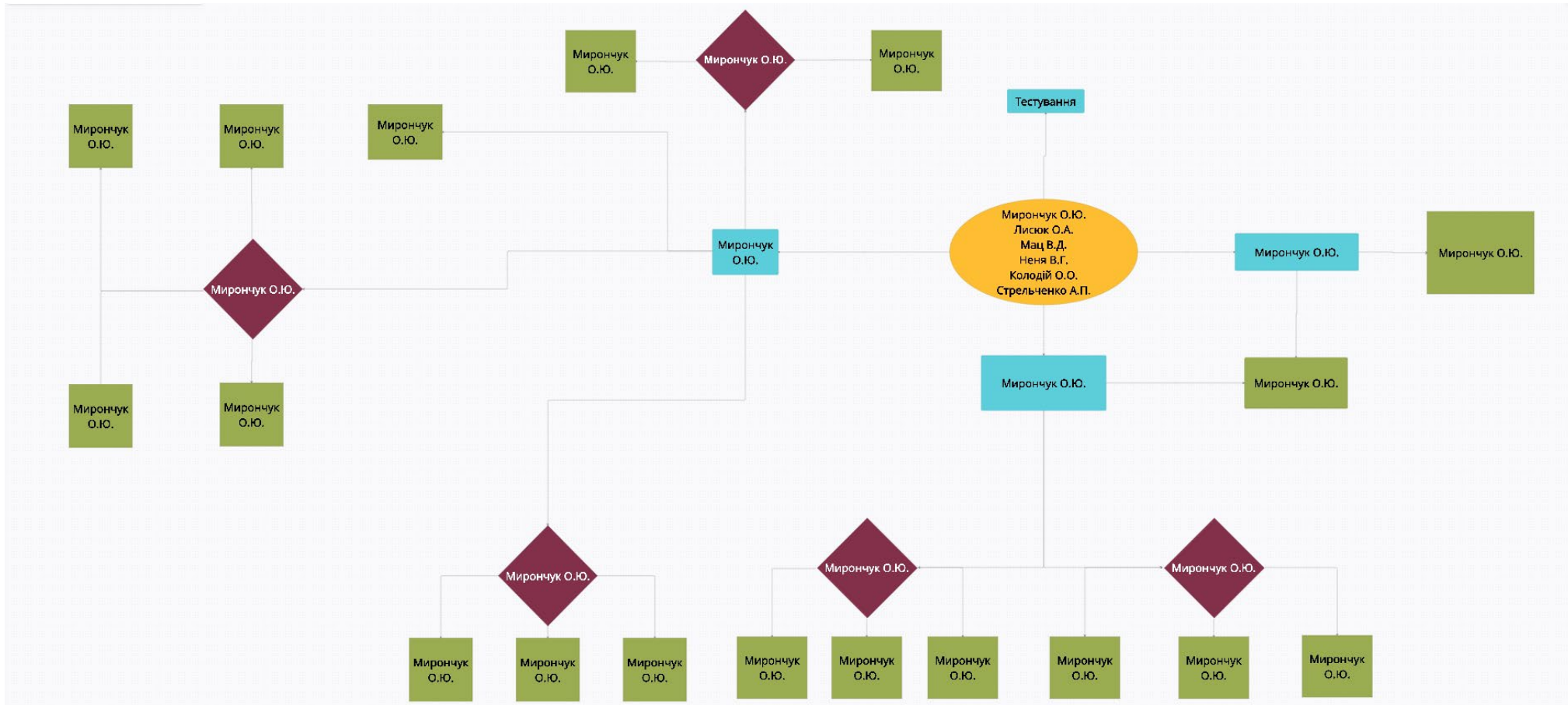


Рисунок Б.2 – OBS-структура робіт проекту

Список виконавців даного проекту описано в таблиці Б.2.

Таблиця Б.2 – Виконавці проекту.

Роль	Ім'я	Проектна роль
Розробник	Мирончук О.Ю.	Виконує розробку модулів.
Проектувальник	Мирончук О.Ю.	Відповідає за проектування та структуру програмного забезпечення.
Тестувальники	Мирончук О.Ю. Лисюк О.А. Мац В.Д. Неня В.Г. Колодій О.О. Стрельченко А.П.	Здійснює тестування дизайну та функціональності створених компонентів.
Керівник проекту	Неня В.Г	Видає завдання на розробку проекту.
Менеджер проекту	Мирончук О.Ю.	Відповідальний за розподіл ресурсів та виконання завдань

Діаграма Ганта. Побудова календарного графіку (діаграми Ганта) є одним з найважливіших етапів планування проекту, що виглядає як розбиття виконання робіт з реальним розподілом дат. Завдяки їй можливо отримати достовірне уявлення про тривалість процесів з обмеженнями у ресурсах, урахуванням вихідних днів та свят. Календарний графік проекту представлено на рисунку Б.3.

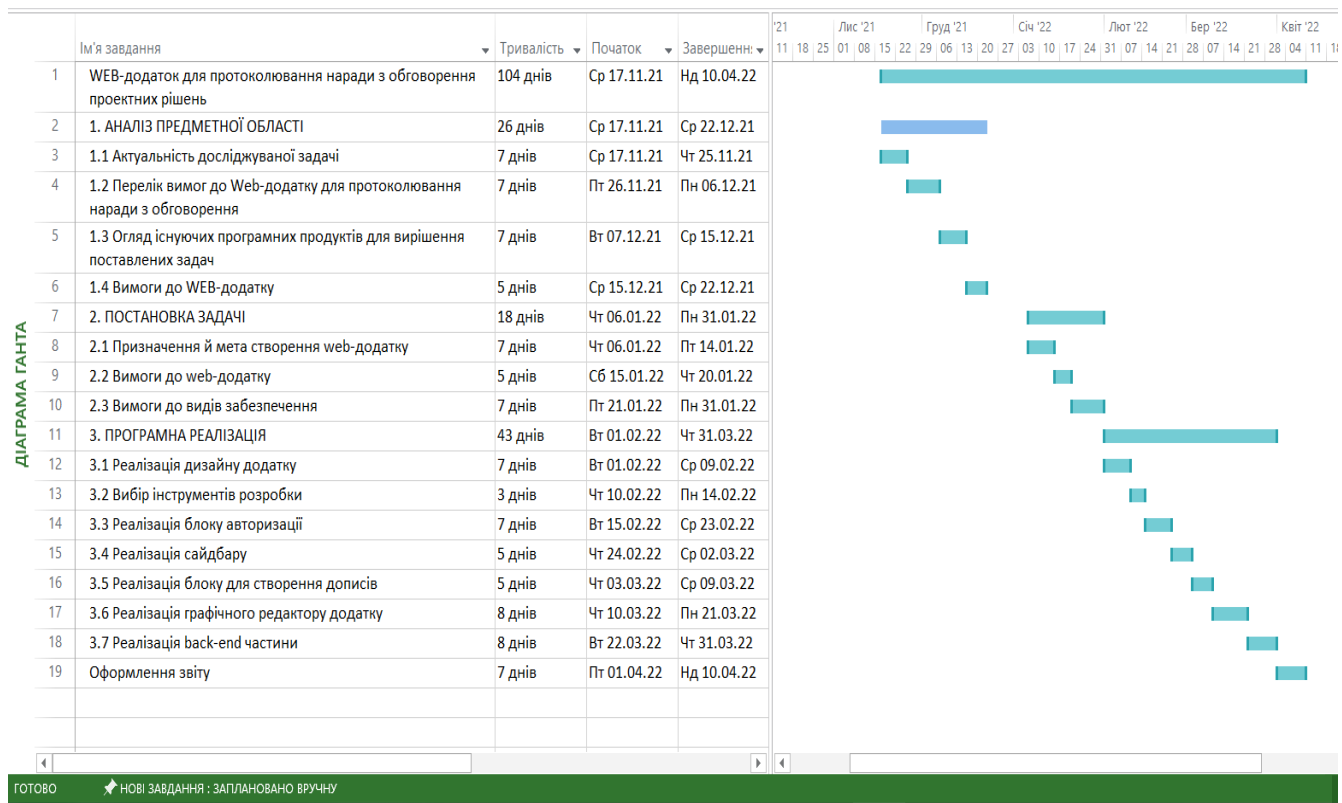


Рисунок Б.3 – Календарний графік розробки web-додатку для протоколювання наради з обговорення проектних рішень

ДОДАТОК В

Лістинг файла App.jsx

```
import React from "react";
import { Routes, Route } from "react-router-dom";

import { GlobalStyle, Container } from "./globalStyles";
import { Header } from "components";
import {
  Home,
  NotFound,
  Registration,
  Login,
  Questions,
  Tags,
  Ranking,
} from "pages";

function App() {

  return (
    <div className="App">
      <Container>
        <GlobalStyle />
        <Header />
        <Routes>
          <Route path="*" element={<NotFound />} />
          <Route path="/" element={<Home />} />
          <Route path="/" element={<Questions />} />
          <Route path="/tags" element={<Tags />} />
        </Routes>
      </Container>
    </div>
  );
}
```



```

    <Route path="/ranking" element={<Ranking/>} />
  </Route>
  <Route path="/registration" element={<Registration />} />
  <Route path="/login" element={<Login />} />
</Routes>
</Container>
</div>
);
}

```

```
export default App;
```

Лістинг файла Home.jsx

```

import React, { useState } from "react";
import { Outlet } from "react-router";

import { Main, Navbar, OpenCloseButton, Content } from "./style";
import { ReactComponent as ChevronLeft } from "assets/chevron-left.svg";
import { Search, Menu, Authorized } from "components";
import { menuItems, personalNavigatorItems } from
"components/Menu/constants";

export const Home = ({ authorized }) => {
  const [isOpenMenu, setIsOpenMenu] = useState(false);

  return (
    <Main>
      <Navbar isOpenMenu={isOpenMenu}>
        <OpenCloseButton
          isOpenMenu={isOpenMenu}

```

```

        onClick={() => setIsOpenMenu(!isOpenMenu)}
    >
        <ChevronLeft />
    </OpenCloseButton>

        <Search      isOpenMenu={isOpenMenu}
setIsOpenMenu={setIsOpenMenu} />
        <Menu      title="MENU"      menuItems={menuItems}
isOpenMenu={isOpenMenu} />
        {authorized && (
            <Menu
                title={isOpenMenu ? "PERSONAL NAVIGATOR" : ""}
                menuItems={personalNavigatorItems}
                isOpenMenu={isOpenMenu}
            />
        )}
    </Navbar>
    <Content>
        <Outlet />
    </Content>
</Main>
);
};

```

Лістинг файла Header.jsx

```
import React from "react";
import { Link } from "react-router-dom";

import { Wrapper, Logo } from "./style";
import { Authorized, Auth } from "components";

import logo from "assets/logo.svg";

export const Header = ({ authorized }) => {
  return (
    <Wrapper>
      <Link to="/">
        <Logo>
          <img src={logo} alt="Logo" />
          <h1>
            аем<b>help</b>
          </h1>
        </Logo>
      </Link>
      {authorized ? <Authorized /> : <Auth />}
    </Wrapper>
  );
};
```

Лістинг файла Questions.jsx

```
import React from "react";

import { Wrapper, Posts } from "./style";
import { Sorting, Post } from "components";

export const Questions = ({posts}) => {
  return (
    <Wrapper>
      <Sorting />
      <Posts>
        {posts.map((item, index) => (
          <Post key={item.title} {...item} />
        ))}
      </Posts>
    </Wrapper>
  );
};
```

Лістинг файла Post.jsx

```
import React from "react";

import { ReactComponent as FeatherMore } from "assets/feather_more-
vertical.svg";
import eyePost from "assets/eye-post.svg";
import comment from "assets/comment.svg";
import arrowUp from "assets/arrow-up-post.svg";

import {
  Wrapper,
```

```
Container,  
Head,  
Author,  
Info,  
UserName,  
PostContent,  
Image,  
SubMenu,  
Tags,  
Activity,  
} from "./style";
```

```
export const Post = ({ author, date, title, text, tags, activity, img }) => {  
  return (  
    <Wrapper>  
      <Container>  
        <Head>  
          <Author>  
            <img src={author.ava} alt={title} />  
            <Info>  
              <UserName>{author.name}</UserName>  
              <time dateTime="">{date} min ago</time>  
            </Info>  
          </Author>  
          <FeatherMore />  
        </Head>  
        <PostContent>  
          <h4>{title}</h4>  
          <p>{text}</p>  
          <Image>
```

```
        <img src={img} alt={img} />
      </Image>
    </PostContent>
    <SubMenu>
      <Tags>
        {tags.map((item) => (
          <li>
            <a href="#" ">{item}</a>
          </li>
        ))}
      </Tags>
      <Activity>
        <li>
          <img src={eyePost} alt="eyePost" />
          {activity[0]}
        </li>
        <li>
          <img src={comment} alt="comment" />
          {activity[1]}
        </li>
        <li>
          <img src={arrowUp} alt="arrowUp" />
          {activity[2]}
        </li>
      </Activity>
    </SubMenu>
  </Container>
</Wrapper>
);
};
```

Лістинг файла **Login.jsx**

```
import React from "react";

import { LayoutAuthPages } from "components";
import loginImage from "assets/login-image.jpg";
import { itemsInput } from "../constants";

export const Login = () => {
  return (
    <LayoutAuthPages
      title="We've missed you!"
      description="More than 150 questions are waiting for your wise
suggestions!"
      itemsInput={itemsInput}
      buttonName="Login"
      img={loginImage}
    ></LayoutAuthPages>
  );
};
```

Лістинг файла **Registration.jsx**

```
import React from "react";

import { LayoutAuthPages } from "components";
import registrationImage from "assets/registration-image.jpg";
import { itemsInput } from "../constants";

export const Registration = () => {
  return (
    <LayoutAuthPages
```

```

        title="Join Alem community"
        description="Get more features and privileges by joining to the most
helpful community"
        itemsInput={itemsInput}
        buttonName="REGISTER"
        img={registrationImage}
    </LayoutAuthPages>
);
};

```

Лістинг файла `LayoutAuthPages.jsx`

```

import React from "react";
import { Wrapper, Info, Title, Description } from "./style";
import { AuthForm } from "components/AuthForm";

export const LayoutAuthPages = ({
  title,
  description,
  itemsInput,
  buttonName,
  img,
}) => {
  return (
    <Wrapper>
      <Info>
        <Title>{title}</Title>
        <Description>{description}</Description>
        <AuthForm itemsInput={itemsInput} buttonName={buttonName} />
      </Info>
      <img src={img} alt="img" />
    </Wrapper>
  );
};

```



```
    </Wrapper>
  );
};
```

Лістинг файла index.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter } from "react-router-dom";

import App from "./App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

Лістинг файла globalStyles.js

```
import styled, { createGlobalStyle } from "styled-components";

export const GlobalStyle = createGlobalStyle`

* {
  margin: 0;
  padding: 0;
  list-style-type: none;
  text-decoration: none;
```

```
    color: black;
  }
:root{
  --roboto-font: "Roboto",sans-serif;

  --orange-color: #F48023;
  --gray-color: #EAEAEA;
  --dark-gray-color:#808080;
  --blue-color: #1682FD ;
  --bg-color: #FAFAFA;
  --error-color: #FF0000;
  --black-gray-color:#4e4e5c;
  --rose-white-color: #FCF4EC;

}
body {
  font-family: var(--roboto-font);
}

`;

export const Container = styled.div`
  min-width: 320px;
  ${"" /* max-width: 1440px; */}
  margin: 0 auto;
`;
```

Лістинг файла setFormParam.jsx

```
import * as Yup from "yup";

const setInitialValues = (formItems) => {
  return {
    ...formItems.reduce(
      (obj, { name }) => ({
        ...obj,
        [name]: "",
      }),
      {}
    ),
    // recaptcha: "",
  };
};

const setValidationSchema = (formItems) =>
  Yup.object().shape({
    ...formItems.reduce(
      (obj, { name, validation }) => ({
        ...obj,
        [name]: validation,
      }),
      {}
    ),
    // recaptcha: Yup.string().required(),
  });

export const setFormParam = (formItems) => ({
  initialValues: setInitialValues(formItems),
```

```
validationSchema: setValidationSchema(formItems),
onSubmit: (values) => {
  console.log(values);
},
});
```

Лістинг файла Button/index.jsx

```
import React from "react";

import { Wrapper } from "./style";

import { ReactComponent as UserPlus } from "assets/user-plus.svg";
import { ReactComponent as PlusCircle } from "assets/plus-circle.svg";
import { ReactComponent as ArrowUp } from "assets/arrow-up.svg";
import { ReactComponent as Message } from "assets/message-square.svg";
import { ReactComponent as Image } from "assets/image.svg";
import { ReactComponent as Send } from "assets/send.svg";

export const Button = ({
  children,
  width,
  height,
  bg,
  margin,
  color,
  icon,
  disabled,
  onClick,
}) => {
  const setIcon = (icon) => {
```

```
switch (icon) {
  case "userPlus":
    return <UserPlus />;

  case "plusCircle":
    return <PlusCircle />;

  case "arrowUp":
    return <ArrowUp />;

  case "message":
    return <Message />;

  case "image":
    return <Image />;

  case "send":
    return <Send />;

  default:
    return null;
}
};

return (
  <Wrapper
    width={width}
    height={height}
    bg={bg}
    margin={margin}
```

```
    color={color}
    disabled={disabled}
    icon={icon}
    onClick={onClick}
  >
    {icon && setIcon(icon)}
    {children}
  </Wrapper>
);
};
```

Лістинг файла Button/style.js

```
import styled from "styled-components";

export const Wrapper = styled.button`
  display: flex;
  justify-content: center;
  align-items: center;
  font-family: var(--roboto-font);
  font-size: 12px;
  line-height: 14px;
  font-weight: 900;
  margin: ${({ margin = "0" }) => margin};
  border-radius: 5px;
  outline: none;
  border: none;
  cursor: pointer;
  opacity: ${({ disabled }) => (disabled ? 0.5 : 1)};
  pointer-events: ${({ disabled }) => (disabled ? "none" : "auto")};
```

```
width: ${{ width }} => {  
  switch (width) {  
    case "XS3":  
      return "72px";  
  
    case "XS2":  
      return "78px";  
  
    case "XS":  
      return "95px";  
  
    case "S":  
      return "108px";  
  
    case "M":  
      return "113px";  
  
    case "L":  
      return "126px";  
  
    case "XL":  
      return "149px";  
  
    case "XXL":  
      return "380px";  
  
    default:  
      return "112px";  
  }  
}};
```

```
height: ${{({ height = "112px" }) => {  
  switch (height) {  
    case "S":  
      return "30px";  
  
    case "M":  
      return "38px";  
  
    default:  
      return "30px";  
  }  
}};
```

```
background-color: ${{({ bg }) => {  
  switch (bg) {  
    case "blue":  
      return "var(--blue-color)";  
  
    case "gray":  
      return "var(--gray-color)";  
  
    default:  
      return "var(--orange-color)";  
  }  
}};
```

```
color: ${{({ color }) => {  
  switch (color) {  
    case "blue":  
      return "var(--blue-color)";  
  }  
}};
```



```
    case "gray":
      return "var(--dark-gray-color)";

    default:
      return "#fff";
  }
}};

&:hover {
  opacity: 0.8;
}

&:active {
  opacity: 0.8;
  transform: translateY(1px);
}

& svg {
  margin: 0 12px 0 0;
}
`;
```