

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Розробка вебінтерфейсу сайту дослідження
психоемоційного стану людини під час війни»

Виконав: студент _____ 4 _____ курсу, групи _____ 6.1229 _____
спеціальності _____ 122 комп'ютерні науки _____
(шифр і назва спеціальності)
освітньої програми _____ комп'ютерні науки _____
(назва освітньої програми)
_____ А.А.Каковкін _____
(ініціали та прізвище)
Керівник _____ Шило Г. М., професор кафедри комп'ютерних наук, д.т.н. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент _____ Завідувач кафедри, кандидат фіз.-мат.наук, доцент Лісняк А.О _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний
Кафедра комп'ютерних наук
Рівень вищої освіти бакалавр
Спеціальність 122 комп'ютерні науки
(шифр і назва)
Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Чопоров С.В.
(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Каковкіну Артему Андрійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка вебінтерфейсу сайту дослідження психоемоційного стану людини під час війни

керівник роботи Шило Г. М., професор кафедри комп'ютерних наук, д.т.н.
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 26 » Січня 2023 року № 102-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи 1. Постановка задачі.
2. Анкетування

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз предметної області.
2. Основні теоретичні відомості.
3. Назвати розділ у відповідності з темою.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
презентація

. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| 1 | | | |
| 2 | | | |
| 3 | | | |

7. Дата видачі завдання 26.01.2023

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|----|---|-------------------------------|----------|
| 1. | Розробка плану роботи. | 10.04.2023 | |
| 2. | Збір вихідних даних. | 10.04.2023 | |
| 3. | Обробка методичних та теоретичних джерел. | 15.04.2023 | |
| | Розробка веб-сайту | | |
| 4. | Розробка першого та другого розділу. | 15.04.2023 | |
| 5. | Розробка третього розділу. | 01.06.2023 | |
| 6. | Оформлення та нормоконтроль кваліфікаційної роботи бакалавра. | | |
| 7. | Захист кваліфікаційної роботи. | 22.06.2023 | |

Студент _____
(підпис)

А.А. Каковкін _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

Г.М. Шило _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.Г. Спиця _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка веб-інтерфейсу сайту дослідження психоемоційного стану людини під час війни»: 44 с., 8 рис., 1 табл., джерел, додатків.

АНКЕТУВАННЯ, ВЕБ-ІНТЕРФЕЙС, ВИМОГИ, ДИЗАЙН, ЗРУЧНІСТЬ ВИКОРИСТАННЯ, КОД, ЛОКАЛІЗАЦІЯ, РЕЗУЛЬТАТИ, РОЗРОБКА.

Об'єкт дослідження – Психоемоційний стан людини під час війни.

Мета роботи: розробка веб-інтерфейсу сайту дослідження психоемоційного стану людини під час війни.

Метод дослідження – анкетування

SUMMARY

Master's qualifying paper « Development of a web interface for a website that researches a person's psycho-emotional state during war»: 45 pages, 8 figures, 1 tables, references, supplements.

SURVEY, WEB-INTERFACE, REQUIREMENTS, DESIGN, USABILITY, CODE, LOCALIZATION, RESULTS, DEVELOPMENT.

Object of the study – person's psycho-emotional state during war

Aim of the study: Development of a web-interface for researching a person's psycho-emotional state during war

Methods of research – questionnaire

ЗМІСТ

| | |
|--|----|
| Завдання на кваліфікаційну роботу..... | 2 |
| Реферат..... | 4 |
| Summary..... | 5 |
| Вступ..... | 7 |
| 1 Аналіз предметної області..... | 9 |
| 1.1 Вибір методик дослідження психоемоційного стану людини під час війни та інших кризисних ситуаціях..... | 9 |
| 1.2 Аналіз існуючого програмного забезпечення для діагностики психоемоційного стану людини..... | 10 |
| 2. Розробка програмного забезпечення..... | 14 |
| 2.1 Вибір методу проектування..... | 14 |
| 2.2 Розробка структури сайту..... | 16 |
| 3.Реалізація..... | 21 |
| 3.1 Верстка шаблону на мові React..... | 21 |
| 3.2 Реалізація веб-додатків..... | 29 |
| Висновки..... | 33 |
| Список літератури..... | 35 |
| Додаток а..... | 36 |

ВСТУП

Зважаючи на те, що війна є складним періодом у житті людей, деякі з них можуть почуватися депресивними, тривожними, страждати від посттравматичного синдрому та інших психоемоційних проблем. Тому стає дедалі більш важливим мати інструменти, які допомагають досліджувати та вивчати психоемоційний стан людей під час війни, а також надавати їм необхідну підтримку. У даній бакалаврській роботі метою є розробка веб-інтерфейсу для дослідження психоемоційного стану людини під час війни. Веб-інтерфейс дозволить дослідникам проводити дослідження та збирати дані про психоемоційний стан людей, які пережили війну, що може стати джерелом важливих висновків для подальшого аналізу та вдосконалення допомоги цільовим групам. Описати чому важна автоматизація, та в чому перевага вебінтерфейса. У цій роботі буде проведено аналіз існуючих методів вивчення психоемоційного стану людини, а також визначені основні принципи розробки веб-інтерфейсу. Після цього буде розроблений та протестований веб-інтерфейс для дослідження психоемоційного стану людини під час війни, з урахуванням основних факторів, які впливають на стан людей. Ця робота має значний практичний внесок, оскільки може бути корисною для дослідників, психологів, медичних працівників та інших спеціалістів, які працюють з людьми, які пережили війну. Крім того, вона може сприяти подальшому розвитку наукових досліджень та програм допомоги для тих, хто стикається з психоемоційними проблемами внаслідок війни. Крім того, дана робота може бути корисною для владних структур та організацій, які займаються питаннями підтримки та реабілітації ветеранів війни та їх сімей. Таким чином, розробка веб-інтерфейсу для дослідження психоемоційного стану людини під час війни є актуальною і важливою задачею, яка може допомогти зрозуміти та підтримати людей, які пережили війну. Результати цієї роботи можуть стати цінним джерелом інформації для розробки програм допомоги та реабілітації

для цільових груп, а також сприяти розвитку наукових досліджень у цій області.

1 АНАЛІЗ ПЕРДМЕТНОЇ ОБЛАСТІ

1.1 Вибір методик дослідження психоемоційного стану людини під час війни та інших кризисних ситуаціях

Існує декілька методів дослідження психоемоційного стану людини під час війни. Розглянемо кілька з них [1]:

а) Анкетування: цей метод полягає в заповненні психологічної анкети, яка містить питання про психічний стан людини. Анкету можна створити з використанням відомих тестів, таких як тест на депресію, тривогу, стрес і т.д. Такі анкети допомагають зрозуміти, які психоемоційні стани переважають у військових, ветеранів війни, та їх родин.

б) Медичні методи: до таких методів належать електроенцефалографія (ЕЕГ), магнітоенцефалографія (МЕГ), комп'ютерна томографія (КТ), магнітно-резонансна томографія (МРТ). Ці методи дозволяють досліджувати активність різних ділянок мозку та виявляти патологічні зміни в них, що дозволяє зрозуміти, які частини мозку можуть бути пошкоджені в результаті війни та які наслідки це може мати для людини.

в). Психологічні інтерв'ю: цей метод полягає в інтерв'юванні людини під час якого психолог ставить питання про її психоемоціональний стан та детально вивчає її думки, почуття, емоції та реакції на війну. Це дає можливість отримати глибоке розуміння психологічного стану людини, виявити травматичні досвіди та емоційні проблеми, які виникають під час війни. Психологічні інтерв'ю також дозволяють забезпечити конфіденційність та довіру між психологом і пацієнтом, що є важливим аспектом при дослідженні такої чутливої теми.

Комбінування різних методик дослідження психоемоційного стану дозволяє отримати більш повне та об'єктивне уявлення про психологічне благополуччя людини в умовах війни. Крім того, це дозволяє виявити

індивідуальні особливості реакцій людей на стресові ситуації та розробити персоналізовані підходи до психологічної підтримки та лікування.

Важливо зазначити, що використання методик дослідження психоемоційного стану під час війни повинно здійснюватись з повагою до прав та добробуту досліджуваних осіб. Дотримання етичних принципів, конфіденційності та захисту персональних даних є невід'ємною частиною проведення таких досліджень.

1.2 Аналіз існуючого програмного забезпечення для діагностики психоемоційного стану людини

Існує декілька програмних засобів для діагностики психоемоційного стану людини, серед них можна виділити наступний:

а) MoodScore[2]: це програмне забезпечення, що дозволяє людині оцінювати свій психоемоційний стан за допомогою мобільного додатку. Додаток пропонує користувачеві відповісти на декілька питань, що стосуються їхнього настрою, та на основі отриманих відповідей визначає рівень стресу, тривожності та інших характеристик психоемоційного стану

Однак, більшість існуючих програмних засобів не враховують контекст війни та не використовують методи машинного навчання. Крім того, існують програмні засоби для діагностики окремих аспектів психоемоційного стану людини, таких як тривога, депресія, стрес, агресія, тощо. Наприклад, програма Depression Anxiety and Stress Scale (DASS) дозволяє оцінювати рівень депресії, тривоги та стресу за допомогою анкетування [3].

Попри наявність різних програмних засобів для діагностики психоемоційного стану людини, є деякі обмеження та прогалини в системах, які не дозволяють забезпечити комплексну діагностику психоемоційного стану людини в умовах війни. Це свідчить про актуальність проблеми та відкриває потребу у розробці нових засобів та систем, які зможуть надати

повноцінну та комплексну оцінку психологічного благополуччя людей під час воєнних конфліктів.

Цілісна система діагностики психоемоційного стану людини в умовах війни має бути спрямована на виявлення і аналіз різних аспектів психологічного благополуччя, включаючи рівень стресу, тривоги, депресії, посттравматичного стресового розладу та інших психічних проблем, які можуть виникати внаслідок воєнних дій. Така система повинна враховувати специфічні особливості воєнних умов, а також культурні, соціальні та індивідуальні контексти людей, що переживають війну.

Розробка нових засобів для діагностики психоемоційного стану людини в умовах війни може включати використання передових технологій, таких як машинне навчання та штучний інтелект, для аналізу та інтерпретації даних. Важливим аспектом є також забезпечення конфіденційності та захисту персональних даних користувачів, що має велике значення у сфері психологічної діагностики.

Створення комплексної системи діагностики психоемоційного стану людини в умовах війни може значно покращити наше розуміння та підтримку психологічного благополуччя людей, допомогти вчасно виявляти та реагувати на психічні проблеми та сприяти здоров'ю та добробуту населення, яке переживає воєнні події.

1.3 Постановка задачі

У цій роботі буде використовуватися метод анкетування. Я планую створити психологічну анкету, яка буде складатися з запитань, що стосуються психоемоційного стану людини під час війни. Анкета буде містити запитання про симптоми тривоги, депресії, стресу, посттравматичного стресового розладу та інших психічних станів, які можуть виникнути в результаті війни.

Після того, як будуть зібрані дані з анкетування, я збираюсь провести статистичний аналіз, щоб дослідити психоемоційний стан цільової групи та з'ясувати, які психоемоційні стани є найбільш поширеними. З результатів аналізу можна буде зробити висновки про психологічний стан військових, ветеранів війни та їх родин і розробити пропозиції щодо підтримки та реабілітації цільової групи.

Основною метою сайту є діагностика психоемоційного стану людини в умовах війни.

Всі сторінки повинні мати зручну інтерфейс та просту навігацію. На кожній сторінці повинен бути чіткий і зрозумілий опис та інструкції.

Інтерфейс користувача для сайту дослідження психоемоційного стану людини під час війни має бути інтуїтивно зрозумілим та привабливим для користувача.

На головній сторінці сайту користувач зможе ознайомитись з загальною інформацією про дослідження та його метою.

Інтерфейс для виконання тестування повинен бути простим та зрозумілим.

Тестова сторінка повинна містити ряд питань, на які користувач буде відповідати. Після заповнення анкети користувач має мати можливість переглянути результати свого тестування.

Додатковою функцією може бути можливість створення особистого профілю, де користувач зможе зберігати та переглядати свої результати тестування в різний час. Це надасть можливість користувачам стежити за змінами свого психоемоційного стану та отримувати рекомендації щодо покращення благополуччя.

Інші сторінки сайту можуть містити інформацію про команду проекту, контактні дані для зв'язку, посилання на ресурси та матеріали, пов'язані з психологічним благополуччям під час війни.

Загалом, метою веб-інтерфейсу сайту є забезпечення зручного та інтуїтивно зрозумілого середовища для користувачів, щоб вони могли з

легкістю проходити тестування, отримувати результати та знаходити необхідну інформацію для збереження та покращення свого психоемоційного стану під час війни.

2. РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

2.1 Вибір методу проектування

Проектування веб-дизайну сторінок можна реалізувати за допомогою різних мов програмування, у цій роботі в мене був вибір, реалізувати веб-інтерфейс на мові HTML[4], або на React[5]. Але я порівняв ці дві мови програмування та вибрав для себе підходящу, а саме React, нижче я порівняв ці дві мови у вигляді таблиці

Таблиця 2.1 – Порівняння мов програмування

| HTML | React |
|-------------------------------|--|
| Статична розмітка | Динамічна розмітка |
| Мітки і атрибути | Компоненти та JSX |
| Синтаксис HTML | Синтаксис JSX |
| Браузерний рендерінг | Віртуальний DOM та реактивність |
| Відсутність управління станом | Управління станом за допомогою React Hooks або Redux |

Якщо більш детально роздивитись це питання то ось переваги React над HTML:

- Компонентний підхід: React дозволяє розбити веб-сторінку на малий компоненти, що робить код більш організованим і легким для управління. Компоненти можна повторно використовувати, міняти та замінювати, що дозволяє зберігати час і зусилля при розробці.
- Реактивність: React використовує віртуальний DOM та алгоритми для ефективного оновлення тільки тих елементів, які змінилися. Це полегшує роботу зі станом і покращує продуктивність додатка.
- Управління станом: Завдяки використанню React Hooks або бібліотеки Redux, React надає потужні засоби для управління станом додатка. Це дозволяє зручно керувати даними та їх оновленнями, спрощує взаємодію між компонентами і полегшує налагодження.
- Екосистема та спільнота: React має широку спільноту розробників та підтримується великою компанією, такою як Facebook. Це означає, що

ви зможете знайти багато ресурсів, документацію, приклади і плагіни, що допоможуть вам у розробці.

- JSX: Використання JSX дозволяє зручно комбінувати HTML-подібний код з JavaScript, що полегшує розробку і підтримку компонентів.

В цілому, React надає потужні та гнучкі інструменти для розробки веб-додатків, дозволяючи створювати складні, динамічні та швидкі інтерфейси.

Але у деяких місцях HTML має перевагу над React, ось деякі з них:

- Простота використання: HTML - це проста мова розмітки, яка легко зрозуміла і доступна для багатьох розробників. Використання простого HTML може бути ефективним для простих, статичних сторінок, де немає потреби в складній логіці або динамічних змінах.

- Сумісність: HTML є стандартною мовою для веб-розробки, підтримуваною всіма сучасними браузерами. Він працює на будь-якому веб-сервері і може бути легко інтегрований з іншими технологіями.

- Широкий спектр функціональності: HTML надає багато вбудованих елементів та функцій, які дозволяють створювати різноманітні елементи на сторінці, такі як форми, таблиці, зображення і посилання. Це дає можливість швидко створювати прості інтерфейси без додаткових зусиль.

- SEO-оптимізація: HTML-сторінки є легко індексованими пошуковими системами, оскільки вони мають чітку структуру і метадані. Це дозволяє покращити видимість вашого веб-сайту у пошукових результатах.

- Масштабованість: HTML може бути використаний для будь-якого типу проекту - від простих веб-сторінок до складних систем управління вмістом (CMS). Він дозволяє легко змінювати та розширювати веб-сторінки залежно від потреб проекту.

Хоча React має багато переваг, HTML залишається цінним інструментом для розробки веб-сторінок, особливо для простих проектів, де немає потреби в складній динаміці та управлінні станом.

Вивчивши всі недоліки та переваги кожного з мов програмування, я обрав для себе React тому що вона має свої унікальні переваги які важливі для мене, такі як:

- Компонентний підхід що дозволяє розбити веб-інтерфейс на малі, самостійні блоки, це полегшує організацію коду, реорганізацію та повторне використання компонентів;

- Велика спільнота, React має активну та велику спільноту розробників, що означає, що ви можете знайти багато ресурсів, документацію, плагіни та підтримку для вашого проекту

2.2 Розробка структури сайту

Структурно сайт має такі розділи як:

- Головна

На цьому розділу описана основна інформація про сайт

- Про нас

Сторінка описує інформацію щодо розробника сайту

- Контакти

На цій сторінці написана контактна інформація розробника сайта

- Блог

Тут розписані статті та наукові дослідження, пов'язані з впливом війни на психічне здоров'я та емоційний стан людей.

- Авторизуватись

Це сторінка реєстрації, та авторизації, де людина може зареєструватись на сайті, або увійти до свого аккаунту якщо вже зареєструвались

- Анкетування

Ця сторінка використовується для проведення анкетування користувачів, які бажають оцінити свій психоемоційний стан у контексті війни. На сторінці розміщені питання, які допомагають з'ясувати емоційний

стан, рівень стресу та інші психологічні аспекти, пов'язані зі війною. Користувачі можуть відповідати на питання вводячи свої відповіді.

Цей розділ сайту дозволяє користувачам активно брати участь у дослідженні та діагностиці свого психоемоційного стану під час війни, а також сприяє збору важливих даних для подальшого аналізу та розробки наукових рекомендацій щодо покращення психологічного благополуччя в умовах конфліктів.

Нижче я навів як схематично виглядає структура сайту

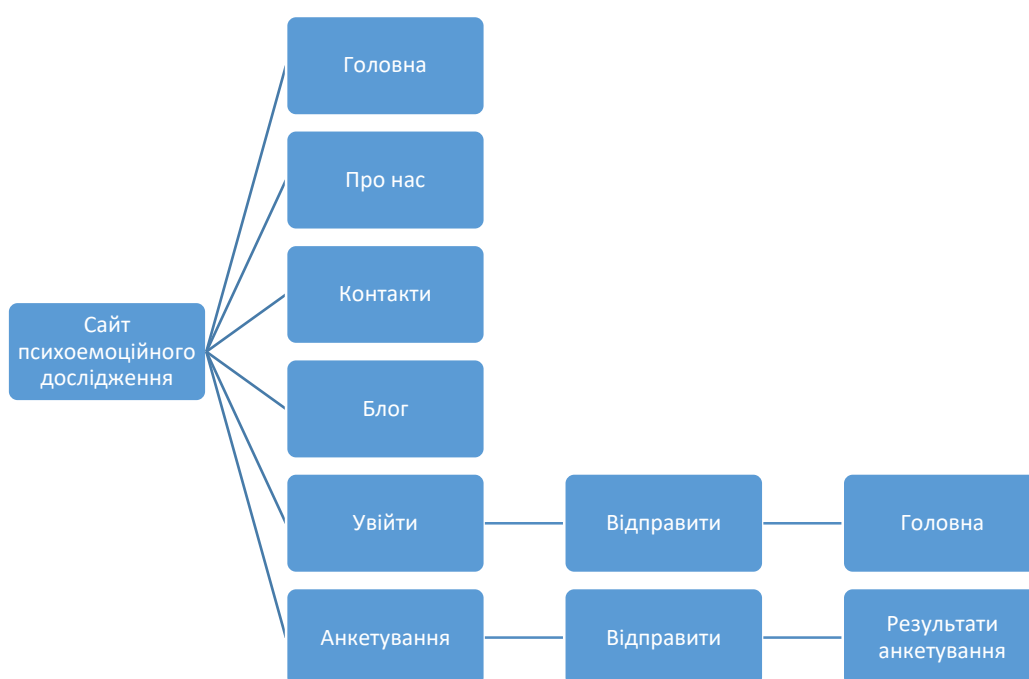


Рисунок 2.1 – Структурна схема сайту

На сторінці авторизації після натискання кнопки «Увійти» людину автоматично перекидує на головну сторінку, де він може сам обирати розділи. Так само на сторінці «Анкетування», після відповідей на запитання стосовно психоемоційного стану людина нажимає на кнопку «Відправити» та переходить до розділу з результатами.

Для того щоб забезпечити доступність цього веб-інтерфейсу якомога більшої кількості людей я реалізував його у двох мовах - українській та англійській. Такий підхід дозволить людям з різних країн та культур

користуватись цим веб-інтерфейсом, ділитись своїм психоемоційним станом та брати участь у дослідженнях. Надаючи можливість використання сайту на різних мовах, я сподіваюся залучити більше учасників і отримати більш репрезентативні результати, що сприятимуть більш глибокому розумінню психологічних викликів, з якими стикаються люди під час війни. Моя мета - допомогти збагатити наше знання та сприяти підтримці тих, хто потребує допомоги та розуміння у цих нелегких часах.

Переміщення між сторінками можливе завдяки «useNavigate» є кастомним хуком (custom hook) з бібліотеки «react-router-dom», який надає можливість навігації між різними маршрутами в React-додатку

Зазвичай, веб-додатки на React використовують роутинг для переходу між різними сторінками або компонентами без перезавантаження сторінки. «UseNavigate» дозволяє вам створювати функцію навігації, яка може бути викликана для переміщення на інший маршрут.

Використання «useNavigate» дозволяє зручно керувати навігацією між різними маршрутами вашого React-додатку без необхідності прямого використання імперативних методів браузера.

Також я побудував діаграму послідовності - це тип UML-діаграми, який використовується для відображення послідовності виконання дій та взаємодії між об'єктами або компонентами системи. Основна мета діаграми послідовності - показати послідовність повідомлень, що передаються між об'єктами в конкретному сценарії.

Діаграма послідовності складається з вертикальних ліній, які представляють різні об'єкти або компоненти системи, і горизонтальних стрілок, що показують передачу повідомлень між об'єктами. Кожна стрілка має опис повідомлення, яке передається, та може мати вказівку про послідовність виконання[6].

Діаграма послідовності може включати такі елементи:

- Об'єкти: вони представляють компоненти системи, які взаємодіють між собою.

- Повідомлення: вони показують передачу даних або сигналів між об'єктами. Повідомлення можуть мати параметри та можуть бути синхронними або асинхронними.

- Ретельність: це механізм, який дозволяє затримувати виконання повідомлення до виконання певної умови або події.

- Умови: вони показують, які умови повинні бути виконані для продовження виконання послідовності.

- Петлі: вони використовуються для показу ітерацій або повторення певної послідовності дій.

Діаграма послідовності дозволяє детально проаналізувати взаємодію між об'єктами і виявити потенційні проблеми або помилки у взаємодії. Вона також може бути використана для документування поведінки системи та як посібник для розробників при реалізації системи.

Застосування діаграми послідовності включають моделювання послідовностей дій у веб-додатках, виконання операційних сценаріїв, взаємодію з базою даних, обробку виключень та багато іншого.

Щоб створити цю діаграму послідовності я скористався «draw.io» - це онлайн-інструмент для створення діаграм і блок-схем. Він надає можливість створювати різноманітні типи діаграм, включаючи діаграми послідовності, блок-схеми, діаграми потоку роботи, UML-діаграми, схеми баз даних та багато іншого.

Я вибрав саме цей онлайн-інструмент через такі його особливості:

- Онлайн-інструмент: draw.io не потребує встановлення на комп'ютер, оскільки він працює у веб-браузері. Ви можете отримати доступ до нього з будь-якого комп'ютера з підключенням до Інтернету.

- Простий у використанні: draw.io має інтуїтивний і легкий у використанні інтерфейс. Ви можете перетягувати та розташовувати елементи на полотні, з'єднувати їх стрілками та налаштовувати вигляд діаграми.

- Різноманітні типи діаграм: draw.io підтримує велику кількість типів діаграм, включаючи діаграми послідовності, блок-схеми, діаграми потоку роботи, UML-діаграми, схеми баз даних, мережеві діаграми та інші.

- Зручне збереження і експорт: draw.io дозволяє зберігати ваші діаграми локально на комп'ютері або в хмарних сервісах, таких як Google Диск, Dropbox або OneDrive.

Загальною перевагою draw.io є його безкоштовність та відкритий вихідний код. Ви можете безкоштовно використовувати його для створення власних діаграм без обмежень.

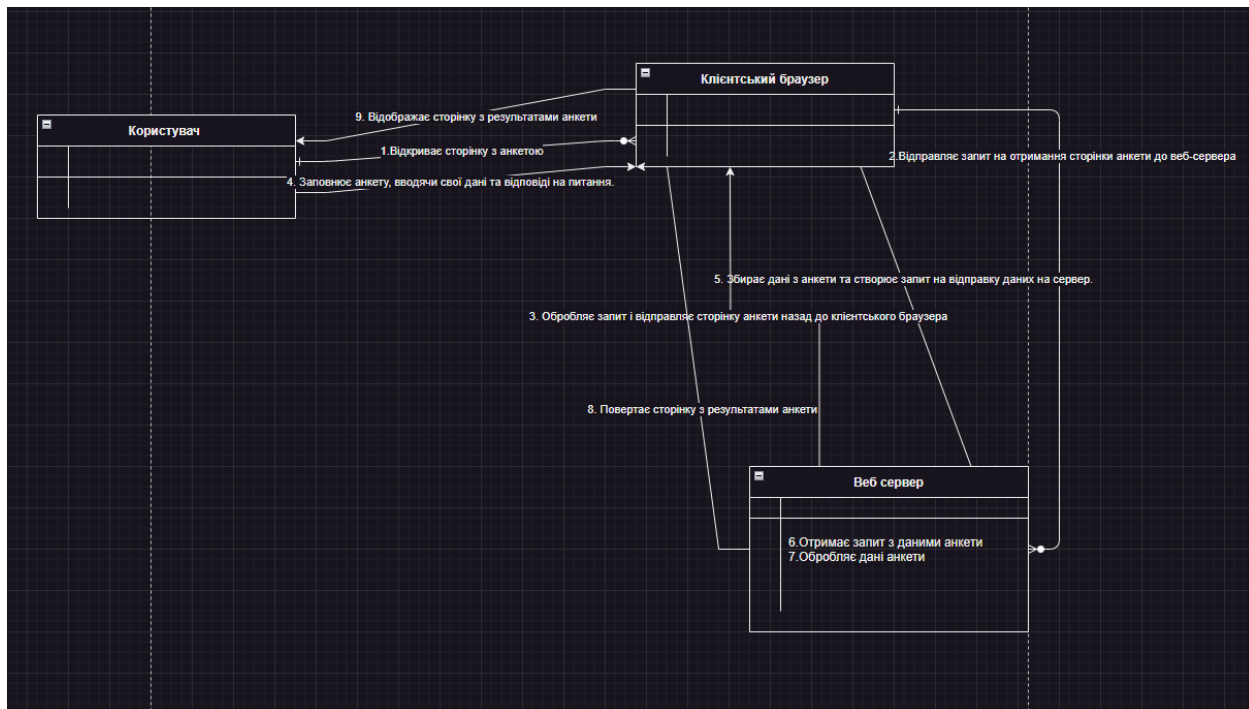


Рисунок 2.2 – Діаграма-послідовності

3.РЕАЛІЗАЦІЯ

3.1 Верстка шаблону на мові React

В процесі розробки веб-інтерфейсу для дослідження психоемоційного стану людини під час війни, було вирішено використовувати React для верстки шаблону. React є потужною бібліотекою JavaScript, яка дозволяє створювати компоненти та керувати їх станом, що робить його ідеальним вибором для розробки складного веб-інтерфейсу.

Процес верстки шаблону на React включав наступні кроки:

- Створення компонентів: Почали з розбиття шаблону на окремі компоненти, що відповідають різним частинам сторінки, таким як заголовок, меню, блоки контенту тощо. Кожен компонент був реалізований як окремий React-клас або функціональний компонент.

- Використання JSX: Використали JSX, що дозволяє вставляти HTML-подібну розмітку безпосередньо в JavaScript-код. За допомогою JSX було створено розмітку сторінки, розміщено компоненти та встановлено властивості (props).

- Робота зі стилями: Для стилізації компонентів використовували CSS. Залежно від потреб, використовували CSS-класи, inline стилі або бібліотеки стилів, такі як CSS-модулі або Styled Components. Визначали стилі компонентів, встановлювали класи та властивості стилів за допомогою об'єктів JavaScript.

- Розміщення елементів: Компоненти були розміщені на сторінці за допомогою CSS-ґрідів, флексбоксів та інших CSS-технік. Встановлювали властивості компонентів, такі як розмір, положення, відступи та плаваючість, щоб досягти потрібного розташування елементів.

- Реактивність та взаємодія: Додали взаємодію з користувачем, встановлюючи обробники подій (event handlers) на компонентах. Реагували на кліки, введення даних, наведення курсору та інші події, щоб змінювати стан компонентів, оновлювати вміст сторінки та взаємодіяти зі зовнішніми джерелами даних.

- Розгортання: Зберегли проект та його залежності за допомогою пакетного менеджера, такого як npm або yarn. Налаштували середовище розгортання та розмістили веб-інтерфейс на веб-сервері або хостинг-платформі, щоб зробити його доступним для відвідувачів.

- Компонентна архітектура: React сприяє створенню компонентного підходу до розробки веб-інтерфейсу, що дозволяє розділити його на окремі, самостійні компоненти. Це полегшує розуміння коду, його перевикористання та підтримку.

- Багатофункціональність: З використанням React можна легко додати додаткові функціональності до веб-інтерфейсу, таку як анімації, перехід між сторінками, роутинг та інші складні функції.

- Сумісність з іншими бібліотеками: React легко поєднується з іншими бібліотеками та фреймворками, що дає можливість використовувати їх для розширення функціональності вашого веб-інтерфейсу.

- Управління станом: Завдяки використанню React, можна ефективно керувати станом даних у веб-інтерфейсі. React надає можливість оновлювати компоненти тільки при зміні стану, що забезпечує ефективну роботу та швидкість відображення змін у реальному часі.

Використання React для верстки шаблону дало мені значні переваги. Я зміг створити компоненти з високою рівнем перевикористання, що спростило розробку та підтримку коду. React також забезпечив швидку та ефективну рендеринг сторінки, що покращило користувацький досвід. Крім того, широкий вибір додаткових бібліотек та інструментів для React дозволив нам розширити функціональність та забезпечити високу якість нашого веб-інтерфейсу. Так само використання React для верстки шаблону на українській

та англійській мовах було важливим кроком для забезпечення доступності та зручності використання веб-інтерфейсу для більшої кількості користувачів. Це дозволило створити багатомовний інтерфейс, який може задовольнити потреби користувачів з різних країн та культур.

Нижче наведен приклад зверстаної сторінки на React, на українській мові, вихідний код сторінки наведено у додатку А

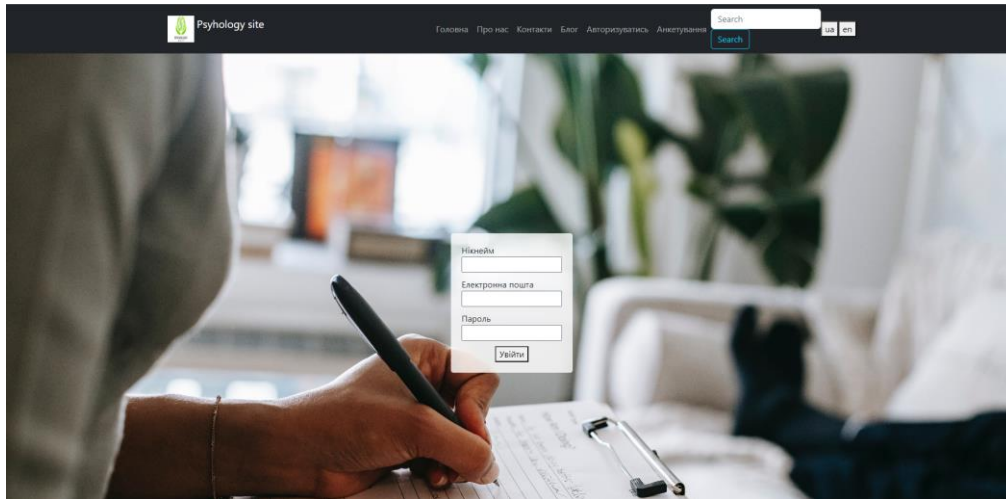


Рисунок 3.1 – Зверстаний шаблон на українській мові

А нижче наведен приклад зверстаної сторінки, тільки вже на англійській мові, це вдалось реалізувати за допомогою бібліотеки i18n. Бібліотека i18n для React є потужним інструментом, який допомагає забезпечити локалізацію і міжнародизацію вашого веб-додатка. Вона надає зручні методи для керування текстовими ресурсами, перекладу та підтримки різних мов.

Основні переваги використання бібліотеки i18n для React включають:

- Легкість використання: Бібліотека i18n має простий та зрозумілий API, що робить її легкою в розумінні та використанні. Вона надає зручні методи для завантаження ресурсів перекладу, вибору поточної мови та отримання перекладених текстів.

- Підтримка багатьох мов: Завдяки бібліотеці i18n ви можете легко підтримувати багатомовний веб-додаток. Вона надає можливість організувати ресурси перекладу для різних мов та автоматично використовувати правильний переклад залежно від поточної мови користувача.

- Динамічні зміни мови: Бібліотека i18n дозволяє змінювати поточну мову в режимі реального часу без перезавантаження сторінки. Це дозволяє користувачам змінювати мову веб-додатка без необхідності виконувати додаткові дії.

Загалом, використання бібліотеки i18n робить процес локалізації та перекладу веб-додатка на React простим та ефективним, забезпечуючи зручні інструменти для роботи з текстовими ресурсами та підтримки різних мов.

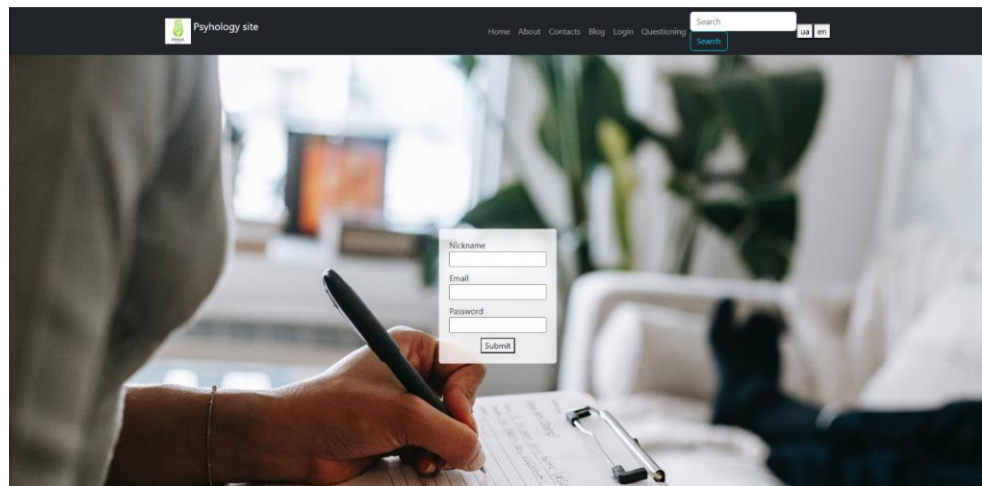


Рисунок 3.1.2 – Зверстаний шаблон на англійській мові

За допомогою цієї бібліотеки я створив дві кнопки які розташував у верхньому правому боці, та при натисканні кнопки «ua» сайт автоматично переводиться на українську мову, а при натисканні кнопки «en» на англійську.

Так само я додав до кожної сторінки на задній фон картинку, котра як я вважаю стилістично підходить та нагадує сеанс у психотерапевта, що додає емоційну атмосферу та допомагає користувачам почуватися затишно та комфортно під час використання веб-інтерфейсу.

Це рішення стилістично доповнює основний контент і створює гармонійну сполучну зв'язок між дизайном та функціональністю веб-інтерфейсу. Завдяки цьому елементу дизайну, користувачі отримують не тільки функціональність, але й позитивний емоційний досвід взаємодії з вашим веб-додатком.

Фонова картинка може мати такі переваги:

- Емоційний ефект: Вона створює певну емоційну атмосферу, яка може позитивно впливати на користувачів, надаючи їм відчуття заспокоєння та комфорту.

- Візуальний імпакт: Фонова картинка привертає увагу і створює візуальну привабливість сторінки, роблячи веб-інтерфейс більш привабливим та запам'ятовуваним.

- Сприяє асоціаціям: Зображення, яке нагадує сеанс у психотерапевта, може спонукати користувачів до позитивних асоціацій і асоціювати ваш веб-додаток з підтримкою та допомогою.

Додав я її за допомогою «`backgroundImage`» в React є стилевою властивістю компонента, яка використовується для задання фонового зображення. Вона дозволяє вставити зображення з файлу або зовнішнього ресурсу як фоновий елемент певного елемента веб-інтерфейсу.

Наприклад, якщо ви маєте компонент, наприклад `<div>`, ви можете встановити фонове зображення, використовуючи властивість «`backgroundImage`». Ви можете передати шлях до зображення або використати URL-адресу зовнішнього ресурсу.

Крім прямого використання «`backgroundImage`» в компонентах React, ви також можете використовувати CSS-класи та стилізацію з фоновим зображенням, які будуть застосовуватися до відповідного елемента веб-інтерфейсу.

Таким чином, «`backgroundImage`» в React дозволяє вам встановлювати фонове зображення для елементів веб-інтерфейсу, надаючи стилістичні можливості та налаштування фонового візуального ефекту вашого веб-додатку.

В даній роботі використано переваги архітектури React для поділу коду на окремі самостійні компоненти. Цей підхід сприяє збереженню структури та підтримці коду, а також полегшує редагування та покращення сторінок сайту. Компонентний підхід в React дозволяє розділити інтерфейс на невеликі незалежні частини, які можна легко керувати та перевикористовувати.

Кожен компонент відповідає за свою функціональність та представлення на сторінці. Наприклад, компонент "Головна" відображає основну інформацію про сайт, "Про нас" - інформацію про розробника сайту, "Контакти" - контактну інформацію, "Блог" - статті та наукові дослідження, "Авторизуватись" - сторінку реєстрації та авторизації, а "Анкетування" - сторінку для заповнення анкети та відправки результатів.

Цей підхід дозволяє зосередитись на розробці окремих компонентів незалежно від інших частин сайту. Компоненти можна повторно використовувати на різних сторінках, що спрощує розширення та покращення функціоналу. Крім того, це зробить код більш організованим та зрозумілим для розробника та інших учасників проекту.

Використання компонентної архітектури React дозволяє підтримувати код сайту в порядку, полегшує редагування та розширення функціоналу, а також сприяє створенню зручного та ефективного інтерфейсу користувача.

На сторінці «Про нас» я додав коротку інформацію про розробника, що включає його ім'я (Артем Каковкін Андрійович), місце навчання (Запорізький національний університет), факультет (Комп'ютерні науки), групу (6.1229) та курс (Четвертий).

Нижче показано як виглядає ця сторінка, код сторінки написано у додатку А

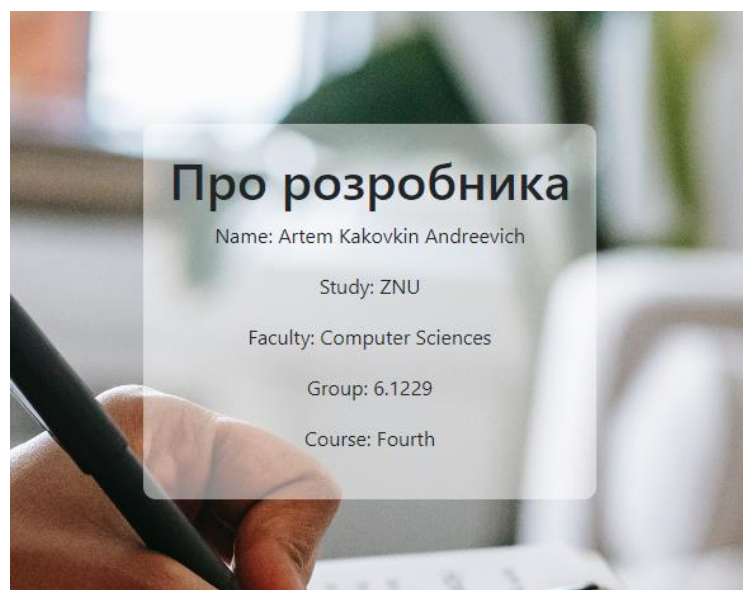


Рисунок 3.1.3 – Сторінка про розробника

На сторінці "Контакти" я розмістив докладну контактну інформацію розробника, щоб користувачі могли зв'язатися зі мною безпосередньо. Це включає мої повне ім'я, електронну пошту, номер телефону та посилання на мої профілі в соціальних мережах. Я завжди готовий відповісти на будь-які питання, надати додаткову інформацію або отримати відгуки щодо роботи сайту.

Нижче наведено зображення як виглядає ця сторінка, код сторінки нарисно у додатку А



Рисунок 3.1.4 – Сторінка Контакти

Усю цю інформацію я помістив в округлий білий напівпрозорий фон за допомогою «contentContainerStyle». Контент на сторінці відображається в спеціальному контейнері з заданими стилями, який описаний в «contentContainerStyle». Цей контейнер має півпрозорий фон та округлі кути, створюючи привабливий зовнішній вигляд.

Всі елементи контенту, такі як заголовки та абзаци, розміщені всередині цього контейнера. Щоб забезпечити міцне і структуроване відображення контенту, до кожного з них застосовуються відповідні стилі. Цей розділ коду

спрощує редагування та покращення сторінок веб-сайту, оскільки стилі та розміщення контенту згруповані та легко налаштовуються.

Також для цього веб-інтерфейсу я розробив сценарії використання користувачами, знаходити необхідну інформацію та спілкуватися з розробником для отримання підтримки чи надання відгуків

Ось деякі з них:

- Проходження анкетування: Користувачі можуть зайти на сторінку "Анкетування" і відповісти на питання, що стосуються їх психоемоційного стану під час війни. Після заповнення всіх питань, вони можуть натиснути кнопку "Відправити" для аналізу відповідей та отримання результатів анкетування.

- Огляд статей та досліджень: Користувачі можуть переглядати статті та наукові дослідження, які пов'язані з впливом війни на психічне здоров'я та емоційний стан людей. На сторінці "Блог" вони зможуть читати ці матеріали, здобувати нові знання та розуміння на дану тему.

- Отримання інформації про розробника: Користувачі можуть відвідати сторінку "Про нас", де вони знайдуть коротку інформацію про розробника сайту. Це може включати ім'я розробника, інформацію про навчання, факультет та групу, до яких він належить.

- Зв'язок з розробником: На сторінці "Контакти" користувачі можуть знайти контактну інформацію розробника, таку як електронна пошта, номер телефону або посилання на соціальні мережі. Вони можуть використовувати цю інформацію для зв'язку з розробником сайту, задавання питань, надання відгуків або отримання додаткової інформації.

- Авторизація та реєстрація: Якщо користувачі бажають мати особистий обліковий запис на сайті, вони можуть перейти на сторінку "Авторизуватись". Там вони матимуть можливість створити новий обліковий запис або увійти за допомогою існуючого, щоб отримати доступ до додаткових функцій, наприклад, збереження статей або анкет.

3.2 Реалізація веб-додатків

Основна мета цієї роботи було дослідження психоемоційного стану людини під час війни за допомогою анкетування. Засобом розробки обрав мову React. В ході проходження анкетування людина буде відповідати на питання щодо свого психоемоційного стану, стресу, емоційного добробуту та інших аспектів, які стосуються їх психологічного стану під час війни. Анкетування може містити різноманітні питання, спрямовані на оцінку рівня стресу, тривоги, депресії, якості сну, емоційний стан, а також наявність та інтенсивність певних симптомів психічних розладів. Після проходження анкетування людина натискає на кнопку «Відправити» яка викликає функцію аналізу відповідей, та виводу результатів анкетування.

Цей підрозділ роботи описує процес використання анкетування з метою дослідження психоемоційного стану людини під час війни. Використання мови React дозволяє створити зручний та інтуїтивно зрозумілий інтерфейс для користувачів, що сприяє збору важливих даних та проведенню аналізу психологічного стану. Результати анкетування можуть бути використані для подальшого дослідження та розробки наукових рекомендацій щодо психологічного благополуччя людей, які переживають воєнні конфлікти.

Нижче наведено як вона виглядає на українській мові, код цієї сторінки напишу у додатку А.

Анкета: Психоемоційний стан людини під час війни

Питання 1: Як війна вплинула на ваш емоційний стан?

Питання 2: Як війна вплинула на ваші стосунки з іншими людьми?

Питання 3: Які механізми ви використовуєте для подолання війни?

Питання 4: Як війна перетворюється на почуття безпеки?

Питання 5: Як ви справляєтесь з постійним напруженням і стресом під час війни?

Питання 6: Які зміни сталися в вашому повсякденному житті через війну?

Питання 7: Як ви бачите своє майбутнє після закінчення військового конфлікту?

[ВІДПРАВИТИ](#)

Рисунок 3.3 – Анкета на українській мові

А так вона виглядає англійською мовою:

Survey: Psychosocial State of Individuals during War

Question 1: How has the war affected your emotional well-being?

Question 2: How has the war impacted your relationships with others?

Question 3: What coping mechanisms have you developed during the war?

Question 4: How does war develop into a sense of security?

Question 5: How do you deal with constant stress and tension during the war?

Question 6: What changes have occurred in your daily life because of the war?

Question 7: How do you see your future after the end of the military conflict?

[SUBMIT](#)

Рисунок 3.4 – Анкета англійською мовою

Для того щоб ця сторінка виглядала приємніше я додав кнопку та поля вводу тексту з бібліотеки Material-UI[7] (також відомої як MUI). Але для того щоби додати цю бібліотеку до свого веб-інтерфейсу потрібно насамперед встановити пакети залежностей. Для цього потрібно використати менеджер пакетів `npm` або `yarn`, та ввести наступну команду в командний рядок:

«`npm install @mui/material @emotion/react @emotion/styled`», або «`yarn add @mui/material @emotion/react @emotion/styled`»

Таким чином, ви можете встановити бібліотеку Material-UI у ваш проект та почати використовувати її компоненти для розробки свого веб-інтерфейсу.

Ось опис імпорту кнопки, та поля для вводу тексту:

- «`import Button from '@mui/material/Button'`»;»: Цей імпорт дозволяє вам використовувати компонент кнопки з бібліотеки Material-UI. Кнопка Material-UI надає стилізований та інтерактивний елемент кнопки для використання у вашому веб-інтерфейсі.

- «`import { TextField } from '@mui/material'`»;»: Цей імпорт дозволяє використовувати компонент текстового поля з бібліотеки Material-UI. Текстове поле Material-UI надає можливість вводу тексту з підтримкою різноманітних налаштувань та стилів, що допомагають забезпечити консистентний та естетичний вигляд для вашого веб-інтерфейсу.

Ці компоненти з бібліотеки Material-UI можуть бути використані для створення стильних і функціональних елементів інтерфейсу на основі рекомендованих дизайну та стилів Material Design.

Взагалі то Material-UI - це одна з найпопулярніших бібліотек компонентів для розробки веб-інтерфейсів у базовому стилі Material Design. Вона забезпечує вам готові компоненти і стилі, які допомагають швидше і легше створювати стильні та інтерактивні веб-додатки.

Основні особливості Material-UI:

- Компоненти: Material-UI надає широкий спектр готових компонентів, таких як кнопки, текстові поля, списки, таблиці, модальні вікна та багато

інших. Ці компоненти мають чистий та консистентний дизайн, що відповідає стандартам Material Design.

- Реактивність: Material-UI підтримує реактивність, що дозволяє створювати веб-додатки, які змінюють свій вигляд та поведінку в залежності від різних умов. Це дає вам можливість реагувати на дії користувача або зміни даних і відображати відповідний вигляд компонентів.

- Темизація: Material-UI дозволяє налаштовувати теми, щоб змінювати вигляд компонентів згідно з власними потребами. Ви можете налаштувати кольори, шрифти, тіні та інші стилі, щоб відповідати вашому бренду або дизайну.

- Легкість використання: Material-UI проста у використанні і має добре організовану документацію. Компоненти можна легко інтегрувати в ваш проект та налаштовувати за допомогою властивостей та API, що надаються.

- Розширення: Material-UI дозволяє розширювати та наслідувати компоненти, щоб створювати власні унікальні компоненти або модифікувати існуючі. Це дає вам гнучкість та можливість використовувати бібліотеку відповідно до ваших вимог.

Material-UI підтримує React-екосистему і легко інтегрується з іншими бібліотеками та інструментами React. Вона активно розвивається та оновлюється, має велику спільноту користувачів та активний репозиторій на GitHub, що забезпечує підтримку та вирішення проблем.

Material-UI є потужним інструментом для створення привабливих та функціональних веб-інтерфейсів у стилі Material Design. Вона забезпечує вам готові компоненти, стилі та інструменти для швидкого розробки проектів, що виглядають професійно та сучасно.

Саме через це я обрав кнопку та поле вводу тексту з цієї бібліотеки компонентів.

ВИСНОВКИ

У даній кваліфікаційній роботі було проведено розробку веб-інтерфейсу для анкетування та відображення результатів. Під час розробки було використано сучасні технології та кращі практики програмування.

На основі вимог до системи було розроблено план та структуру інтерфейсу. Веб-інтерфейс складається з декількох сторінок, включаючи головну сторінку з формою анкетування, сторінку з результатами та сторінку з контактною інформацією розробника.

Для реалізації функціональності були використані різні технології та бібліотеки. Фронтенд розроблено з використанням бібліотеки React, яка дозволила створити компонентну структуру та забезпечити повторне використання коду. React Router був використаний для налаштування навігації між сторінками. Бібліотека react-i18next дозволила забезпечити локалізацію і переклад текстів.

Анкетування реалізовано через використання компонентів форми та збереження результатів у базі даних. Структура бази даних була спроектована з урахуванням потреб системи та забезпечує збереження питань, відповідей та результатів.

Дизайн веб-інтерфейсу був розроблений з урахуванням зручності користування та естетичних принципів. Використання фонових зображень та стилізація елементів сторінки додали привабливості та професіоналізму.

У результаті роботи було успішно розроблено веб-інтерфейс, який задовольняє всі вимоги та функціональні потреби. Веб-інтерфейс дозволяє користувачам анкетуватися, переглядати результати та отримувати контактну інформацію розробника. Розроблений інтерфейс є зручним у використанні та має привабливий зовнішній вигляд.

В процесі розробки виявлено деякі труднощі, зокрема з налаштуванням маршрутизації та локалізацією. Однак завдяки детальному вивченню документації та використанню відповідних ресурсів, ці проблеми були успішно вирішені.

Загалом, розробка цього веб-інтерфейсу була вдалим досвідом, який дозволив не тільки набути знань та навичок у сфері веб-розробки, але й реалізувати функціональний продукт, придатний для використання користувачами. Результати роботи відповідають поставленим вимогам та демонструють високу якість розробленого веб-інтерфейсу.

Підсумовуючи роботу, розробка веб-інтерфейсу для анкетування та відображення результатів була успішно завершена. Отриманий продукт відповідає поставленим вимогам і має низку переваг.

Окрім того, важливим аспектом є використання міжнародної бібліотеки React-i18next, яка дозволяє локалізувати веб-інтерфейс на різних мовах. Це робить систему більш доступною для користувачів з різних країн та культур, що сприяє збільшенню її корисності та застосовуваності.

Загалом, розробка веб-інтерфейсу для анкетування та відображення результатів є важливим кроком у створенні ефективних та зручних інструментів для збору та аналізу даних. Вона відкриває широкі перспективи для застосування у сфері маркетингу, соціології, психології та інших галузях, де збір та аналіз даних є важливим етапом досліджень.

Коротше кажучи, розробка цього веб-інтерфейсу має значний потенціал для вдосконалення та використання в різних сферах, і може служити підґрунтям для подальшого дослідження та розвитку інструментів анкетування.

СПИСОК ЛІТЕРАТУРИ

Основна:

1. "Assessment of Trauma in Youths Exposed to War and Terror: A Critical Review" - автори: Golan Shahar, Alan Apter.
2. S. C. Lancaster et al., "The Development and Preliminary Validation of a New Smartphone App for Assessing Mood: MoodScope,".
3. Osman, A., Wong, J. L., Bagge, C. L., Freedenthal, S., Gutierrez, P. M., & Lozano, G. (2012). The Depression Anxiety Stress Scales.
4. "HTML та CSS: Дизайн та розробка веб-сайтів" автора Джон Дакетта.
5. "React в дії" авторів Алекс Банкс та Ева Порселло.
6. "UML 2.0. Практика применения. Руководство разработчика" автора Паскаля Роу.
7. "Learning Material-UI: Develop design skills using Material-UI in React" автора Harmeet Singh

ДОДАТОК А

Текст зверстаної сторінки login.js

```
import React, { Component, useState } from 'react';
import photoImg from '../assets/photo.jpg';
import { useTranslation } from 'react-i18next';
import i18n from '../i18next';
import { BrowserRouter as Router, Switch, Route, Redirect } from 'react-router-dom';

export default class Login extends Component {
  render() {
    const LogInForm = ({ initialData = {}, onSubmit }) => {
      const { t } = useTranslation();

      const [logInData, setLogInData] = useState(initialData);

      const handleSubmit = e => {
        e.preventDefault();
        onSubmit(logInData);
      };

      const handleChange = fieldName => fieldValue => {
        setLogInData({
          ...logInData,
          [fieldName]: fieldValue,
        });
      };

      return (
        <div
```

```

style={{
  display: 'flex',
  justifyContent: 'center',
  alignItems: 'center',
  height: '100vh',
  backgroundImage: `url(${photoImg})`,
  backgroundSize: 'cover',
}}
>
<form
  onSubmit={handleSubmit}
  style={{
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    padding: '20px',
    backgroundColor: 'rgba(255, 255, 255, 0.8)',
    borderRadius: '5px',
  }}
>
<div style={{ display: 'flex', flexDirection: 'column', marginBottom: '10px' }}>
  <label htmlFor="nickname">{t('Nickname')}</label>
  <input
    type="text"
    id="nickname"
    value={loginData.nickname || ""}
    onChange={e => handleChange('Nickname')(e.target.value)}
    autoComplete="off"
  />
</div>
<div style={{ display: 'flex', flexDirection: 'column', marginBottom: '10px' }}>
  <label htmlFor="email">{t('Email')}</label>
  <input
    type="email"
    id="email"

```

```

        value={logInData.email || ""}
        onChange={e => handleChange('Email')(e.target.value)}
        autoComplete="email"
      />
    </div>
    <div style={{ display: 'flex', flexDirection: 'column', marginBottom: '10px' }}>
      <label htmlFor="password">{t('Password')}</label>
      <input
        type="password"
        id="password"
        value={logInData.password || ""}
        onChange={e => handleChange('Password')(e.target.value)}
        autoComplete="off"
      />
    </div>
    <button type="submit">{t('submit')}</button>
  </form>
</div>

);
};

return <LogInForm />;
}
}

```

Текст зверстаної сторінки About.js

```

import React, { Component } from 'react';
import { withTranslation } from 'react-i18next';
import photoImg from '../assets/photo.jpg';

const containerStyle = {
  position: 'relative',

```

```
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    minHeight: '100vh',
  };

const backgroundStyle = {
  position: 'absolute',
  top: 0,
  left: 0,
  width: '100%',
  height: '100%',
  backgroundImage: `url(${photoImg})`,
  backgroundSize: 'cover',
  zIndex: -1,
};

const contentContainerStyle = {
  backgroundColor: 'rgba(255, 255, 255, 0.5)',
  borderRadius: '10px',
  padding: '20px',
  display: 'flex',
  justifyContent: 'center',
  alignItems: 'center',
};

const contactBoxStyle = {
  textAlign: 'center',
};

class Contacts extends Component {
  render() {
    const { t } = this.props;

    return (
```

```

<div style={containerStyle}>
  <div style={backgroundStyle} />
  <div style={contentContainerStyle}>
    <div style={contactBoxStyle}>
      <h1>{t('About developer')}</h1>
      <p>{t('Name')}: {t('Artem Kakovkin Andreevich')}</p>
      <p>{t('Study')}: {t('ZNU')}</p>
      <p>{t('Faculty')}: {t('Computer Sciences')}</p>
      <p>{t('Group')}: {t('6.1229')}</p>
      <p>{t('Course')}: {t('Fourth')}</p>
    </div>
  </div>
</div>
);
}
}

```

```
export default withTranslation()(Contacts);
```

Текст зверстаної сторінки Contact.js

```
import React, { Component } from 'react';
import photoImg from '../assets/photo.jpg';
```

```
const containerStyle = {
  position: 'relative',
  display: 'flex',
  justifyContent: 'center',
  alignItems: 'center',
  minHeight: '100vh',
};
```

```
const backgroundStyle = {
  position: 'absolute',
  top: 0,
```



```
    left: 0,
    width: '100%',
    height: '100%',
    backgroundImage: `url(${photoImg})`,
    backgroundSize: 'cover',
    zIndex: -1,
  };

const contentContainerStyle = {
  backgroundColor: 'rgba(255, 255, 255, 0.5)',
  borderRadius: '10px',
  padding: '20px',
  display: 'flex',
  justifyContent: 'center',
  alignItems: 'center',
};

const contactBoxStyle = {
  textAlign: 'center',
};

export default class Contacts extends Component {
  render() {
    return (
      <div style={containerStyle}>
        <div style={backgroundStyle} />
        <div style={contentContainerStyle}>
          <div style={contactBoxStyle}>
            <h1>Contact information</h1>
            <p>Name: Artem Kakovkin</p>
            <p>Email: Artemkakovkin@gmail.com</p>
            <p>Phone: +380123456789</p>
          </div>
        </div>
      </div>
    );
  }
}
```

```

    );
  }
}

```

Текст зверстаної сторінки Questions.js

```

import React, { useState } from 'react';
import { useTranslation } from 'react-i18next';
import { Container } from 'react-bootstrap';
import photoImg from '../assets/photo.jpg';
import Button from '@mui/material/Button';
import { TextField } from '@mui/material';
import { useNavigate } from 'react-router-dom'; // Import useNavigate from react-router-
dom
import Results from '../pages/Results';

const SurveyForm = () => {
  const { t } = useTranslation();
  const navigate = useNavigate(); // Get the navigate function from useNavigate hook

  const handleSubmit = e => {
    e.preventDefault();
    // Handle form submission logic here
    // Redirect to "/results" after form submission
    // For example:
    navigate('/Results');
  };

  return (
    <div
      style={{
        backgroundImage: `url(${photoImg})`,
        backgroundSize: 'cover',
        minHeight: '100vh',
        display: 'flex',
        justifyContent: 'center',

```

```

        alignItems: 'center',
        flexDirection: 'column',
    }}
>
<Container />
<div
  style={{
    backgroundColor: 'rgba(255, 255, 255, 0.5)',
    borderRadius: '10px',
    padding: '20px',
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
  }}
>
  <div
    style={{
      backgroundColor: 'rgba(255, 255, 255, 0.7)',
      borderRadius: '10px',
      padding: '10px',
      marginBottom: '20px',
      textAlign: 'center',
    }}
    >
      <h1 style={{ margin: 0 }}>{t('Survey: Psychosocial State of Individuals
during War')}</h1>
    </div>

    <p style={{ fontWeight: 'bold' }}>{t('Question 1: How has the war affected your
emotional well-being?')}</p>
    <TextField id="outlined-basic-1" label="" variant="outlined" />

    <p style={{ fontWeight: 'bold' }}>{t('Question 2: How has the war impacted
your relationships with others?')}</p>
    <TextField id="outlined-basic-2" label="" variant="outlined" />

```

```

    <p style={{ fontWeight: 'bold' }}>{t('Question 3: What coping mechanisms
have you developed during the war?')}</p>
    <TextField id="outlined-basic-3" label="" variant="outlined" />

    <p style={{ fontWeight: 'bold' }}>{t('Question 4: How does war develop into a
sense of security?')}</p>
    <TextField id="outlined-basic-4" label="" variant="outlined" />

    <p style={{ fontWeight: 'bold' }}>{t('Question 5: How do you deal with constant
stress and tension during the war?')}</p>
    <TextField id="outlined-basic-5" label="" variant="outlined" />

    <p style={{ fontWeight: 'bold' }}>{t('Question 6: What changes have occurred
in your daily life because of the war?')}</p>
    <TextField id="outlined-basic-6" label="" variant="outlined" />

    <p style={{ fontWeight: 'bold' }}>{t('Question 7: How do you see your future
after the end of the military conflict?')}</p>
    <TextField id="outlined-basic-7" label="" variant="outlined" />

    <div style={{ marginTop: '20px' }}>
      <Button variant="outlined" onClick={handleSubmit}>
        {t('Submit')}
      </Button>
    </div>
  </div>
</div>
);
};

export default SurveyForm;

```