

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)

(рівень вищої освіти)

на тему **Розробка інформаційного веб-ресурсу з використанням мови
програмування PHP**

Виконав: студентка 4 курсу, групи 6.1210-пзс

спеціальності 121 Інженерія програмного

забезпечення

(код і назва спеціальності)

освітньої програми Програмне забезпечення систем

(код і назва освітньої програми)

А.А.Вахновська

(ініціали та прізвище)

Керівник доцент, к.т.н. О.М. Міхайлуца

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дискус»

Р.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти _____ перший (бакалавський) _____
Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код та назва)
Освітня програма _____ Програмне забезпечення систем _____
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Т. В. Критська
“ 01 ” березня _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Вахновській Анні Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційного веб-ресурсу з використанням мови програмування PHP
керівник роботи _____ Михайлуца Олена Миколаївна, к.т.н., доцент _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від _____ від 26.12.2023 №2215-с _____
2. Строк подання студентом кваліфікаційної роботи _____ 13.06.2024 _____
3. Вихідні дані дипломної роботи
 - комплект нормативних документів ;
 - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - огляд та збір літератури стосовно теми дипломної роботи;
 - огляд та аналіз існуючих рішень та аналогів;
 - дослідження проблеми створення інформаційних веб-ресурсів ;
 - створення програмного продукту та його опис;
 - перелік вимог для роботи програми;
 - тестування програмної системи та розробка висновків та пропозицій.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____ слайдів презентації _____

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.02.2024**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	01.02-10.02.24	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.02-12.02.24	виконано
3	Аналіз існуючих методів рішення	13.02-14.02.24	виконано
4	Аналіз методів розробки інформаційних систем	15.02-20.02.24	виконано
5	Аналіз сучасних систем управління базами даних	21.02-26.02.24	виконано
6	Узгодження подальших дій з науковим керівником	27.03-28.03.24	виконано
7	Проектування інформаційного веб-застосунку	29.03-05.04.24	виконано
8	Програмна реалізація веб-застосунку	06.04-16.04.24	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.04-19.04.24	виконано
10	Реалізація користувацького інтерфейсу для комп'ютерної системи	20.04-01.05.24	виконано
11	Тестування за стосунку	02.05-7.05.24	виконано
12	Оформлення звіту	8.05-12.06.24	виконано
13	Оформлення презентації. Отримання рецензій від опонентів.	13.06-17.06.24	виконано

Студент _____ А.А. Вахновська
(підпис) (прізвище та ініціали)Керівник роботи _____ О.М. Міхайлуца
(підпис) (прізвище та ініціали)**Нормоконтроль пройдено**Нормоконтролер _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок — 64

Рисунків — 28

Джерел — 15

Вахновська А. А. Розробка інформаційного веб-ресурсу з використанням мови програмування PHP : кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник О. М. Михайлуца. Запоріжжя : ЗНУ, 2024. 66 с.

В сучасному світі онлайн ризики стають все далі більшою проблемою для підлітків в Інтернет-просторі. Вони повинні знати усі загрози, з якими вони можуть зіткнутися, та шляхи уникнення цих проблем або, якщо це сталося – як виправити становище. Батьки мають знати, як захистити дітей від онлайн ризиків, як з ними поговорити про це, та у випадку, якщо це сталося – як допомогти підлітку вирішити проблему.

Мета і завдання роботи – розробка веб-ресурсу, у якому буде приємний дизайн для підлітків та зрозумілий дизайн для батьків. Це розробка сторінок з інформацією про онлайн ризики, розробка форми, на якій підлітки зможуть анонімно писати психологу, якщо у них з'явилися питання або в них є проблеми, з якими вони зіткнулися та хочуть їх позбутися. Крім того, до складу повинні входити івенти, на яких професіонали будуть розповідати про онлайн ризики, та адміністративна панель для психологів, на якій вони бачать емейл та питання для відповіді користувачам.

У процесі підготовки було проведено порівняльний аналіз існуючих технологій по розробці схожих веб-ресурсів. На основі цього аналізу було розроблено веб-ресурс про онлайн ризики серед підлітків. Як результат створено сайт, на якому можна прочитати про онлайн ризики, задати запитання психологу та зареєструватися на івент, метою якого є показ підліткам та дорослим усі ризики сучасного інформаційного світу та шляхи попередження та подолання можливих негараздів.

Ключові слова: *веб-ресурс, PHP, онлайн ризики.*

ABSTRACT

Pages – 64

Drawings – 28

Sources – 15

Vakhnovska A. A. Development of an informational web resource using the PHP programming language: qualifying thesis of the bachelor of specialty 121 "Software engineering" / Science. manager O. M. Mikhailutsa. Zaporizhzhia: ZNU, 2024. 66 p.

In today's world, online risks are becoming more and more of a problem for teenagers in the Internet space. They should know all the threats they may face and how to avoid them or, if they do, how to fix the situation. Parents should know how to protect their children from online risks, how to talk to them about it, and in case it happened, how to help the teenager solve the problem.

The goal and task of the work is to develop a web resource that will have a pleasant design for teenagers and an understandable design for parents. This is the development of pages with information about online risks, the development of a form on which teenagers can anonymously write to a psychologist if they have questions or problems that they have faced and want to get rid of. In addition, the composition should include events where professionals will talk about online risks, and an administrative panel for psychologists, where they can see email and questions for users to answer.

In the process of preparation, a comparative analysis of existing technologies for the development of similar web resources was carried out. Based on this analysis, a web resource on online risks among teenagers was developed. As a result, a site was created where you can read about online risks, ask questions to a psychologist and register for an event, the purpose of which is to show teenagers and adults all the risks of the modern information world and ways to prevent and overcome possible troubles.

Keywords: *web resource, PHP, online risks.*

Зміст

ВСТУП	7
1 ДОСЛІДЖЕННЯ СИСТЕМ АНАЛОГІВ.....	12
1.1 Огляд існуючих рішень.....	12
1.2 Аналіз зарубіжних додатків.....	19
2 ОПИС ТЕХНОЛОГІЙ.....	22
2.1 Обґрунтування баз даних.....	25
2.2 Порівняльний аналіз PHP та Java.....	26
2.3 Переваги та недоліки системи яка застосована.....	28
3 РОЗРОБКА ІНФОРМАЦІЙНОГО ВЕБ-РЕСУРСУ.....	30
3.1 Вимоги до веб-ресурсу та користувачів.....	31
3.2 Архітектура системи	32
3.3 Інтерфейс системи	34
3.4 Розробка інтерфейсу інформаційного веб-ресурсу.....	54
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

ВСТУП

Мета цієї роботи зосереджена на допомозі дітям та підліткам розібратися як жити в реальному світі, оскільки зараз через пандемію COVID–19 та війну реальне життя ніби повністю перенеслося в онлайн світ. В онлайн світі ми робимо зараз майже усе: працюємо, вчимося, розмовляємо по різноманітним чатам або по відео зв'язку, можемо замовити через Інтернет майже все, що потрібно: різноманітні послуги або товари, проявляємо свою індивідуальність показуючи та розказуючи про своє життя. Тому актуальність у створенні онлайн ресурсу про ризики серед підлітків дуже актуальна та важлива для аудиторії, аби кожен зрозумів як оберігати себе та своїх близьких в цей важкий час.

Підлітки найбільш вразливі у цій ситуації, оскільки під час пандемії багато що перенеслось в онлайн середовище: навчання, спілкування, розваги, оскільки вони не мали змоги вийти на двір чи піти до школи, вони мали сидіти вдома задля свого здоров'я. Ця пандемія створила все нові і нові виклики для підлітків через збільшення вразливості до онлайн ризиків, таких як кіберзагроза, домагання, кібербулінг і інших форм негативного впливу.

Під час війни в Україні негативні психологічні аспекти у підлітків стають особливо важливими. Підлітки можуть відчувати стрес, тривогу, переживання через нестабільність, що робить їх більш схильними до використання онлайн світу, у якому вони ховаються від реального світу. Водночас, це може збільшити їхню небезпеку в онлайн світі.

За даними іноземних та українських психологів важливо розуміти, що підлітки використовують онлайн комунікацію для збереження соціальних зв'язків та вираження своєї унікальності. Проте це також може призвести до негативних наслідків та проблем, якщо не приділяти достатньо уваги безпеці підлітків в онлайн світі.

Як пишуть Емі Орбен, керівник програми, науковий співробітник відділу пізнання та мозку MRC Кембриджського університету,

Великобританія, та Сара–Джейн Блейкмор, професор психології та когнітивної нейронауки на факультеті психології Кембриджського університету, Великобританія, «За останнє десятиліття у підлітків різко зросли депресія, тривога та суїцидальність. Це безпосередньо залежить від кількості часу, яку молоді люди проводять в Інтернеті. Частково через побоювання, що між цими тенденціями існує зв'язок, уряди в усьому світі змушені робити більше для регулювання технологічних компаній». За цією статтею ми можемо дізнатися, що навіть британські психологи пишуть, що в сіті є така проблема в сучасному онлайн світі. Це є актуальною проблемою на яку неможливо закривати очі та заперечувати існування в сучасному онлайн світі. Проблема, навпаки, існує та дуже потужно впливає на психіку підлітків, а онлайн ризики існують взагалі всюди і замовчування їх є справжнім злочином.

Навчання та підтримка з боку батьків, вчителів та громадськості стають найважливішими чинниками в розв'язанні цих проблем. Потрібно акцентувати увагу на цифровій грамотності, психологічній підтримці та розвитку навичок взаємодії в онлайн світі. Треба спілкуватися з підлітками про те, які небезпеки їх чекають в онлайн світі, розповідати, як їх уникнути, та, якщо щось сталось, треба підтримати та допомогти позбутися цієї проблеми. Очевидно те, що замовчування проблеми під час спілкування з підлітками надалі викличе острах та замкнутість, пошук радників у своєму середовищі і не приведе к її вирішенню.

Як пише Американська Психологічна Асоціація онлайн світ допомагає підліткам у здоровому розвитку, але також цей онлайн світ може створювати онлайн ризики. Психологічні дослідження показують, що надзвичайно важливо зосередитися на тому, що підлітки бачать в онлайн світі, як вони використовують соціальні мережі та який тип контенту вони використовують. Батьки розуміють свою дитину як ніхто інший, вони точно знають, що з побаченого в онлайн світі буде гарним досвідом, а що навпаки нашкодить підлітку.

Тому батьки повинні:

- Скоротити використання соціальних мереж, які включають в себе підрахунок лайків, оскільки це дає негативний вплив на підлітка, та зробити так, щоб підліток спав восьмигодинний сон для забезпечення здорового розвитку.
- Щотижня розмовляти з підлітком про те, що він бачив в онлайн світі, аби підліток почував прихисток від батьків замість засудження. Треба питати їх, як вони зрозуміли опублікований контент, та можна ставити гіпотетичні запитання, як вони будуть реагувати на ту чи іншу ситуацію в онлайн світі.
- Обговорювати з підлітками як і чому батьки використовують онлайн світ та соціальні мережі. Батьки можуть теж собі скоротити використання соціальних мереж, щоб діти теж наслідували приклад батьків. Треба обговорювати відпочинок від соціальних мереж та онлайн світу всією сім'єю, щоб діти розуміли, які труднощі та спокуси ви відчуваєте, коли тривалий час не сидите в онлайн світі.
- Батьки мають спостерігати за дитиною та турбуватися про те, що дитині в онлайн світі загрожує небезпека або знищується емоційний та психологічний стан. Якщо вони не можуть контролювати скільки дитина сидить в онлайн світі, то краще за все звернутися до спеціаліста з психічного здоров'я, який допоможе знайти більш здорові способи спілкування в онлайн світі.

Аудиторія цього веб-ресурсу буде складатися з підлітків та дорослих (батьки та вчителі), які хочуть зрозуміти, що робити та як допомогти один одному у цей важкий час та не потрапити на шахраїв. Підлітки зможуть прочитати, які потенційні ризики можуть їх спіткати на кожному кроці життя, зможуть зареєструватися на івенти в яких вони дізнаються ще більше інформації, та у випадку проблеми зможуть звернутись до підтримки або до батьків. Для дорослих буде дуже корисно дізнатись, які онлайн ризики існують в нашому світі та як не потрапити самим та уберегти підлітків від цього або як вийти із цієї проблеми якщо це трапилось.

Найважливішими вимогами до цієї роботи є забезпечення зацікавленості підлітків інформацією з цього сайту, щоб вони не виходили з нього з думками: навіщо я на нього зайшов, не зрозумілий дизайн та інформація. У цій роботі повинен бути дуже зручний, сучасний та зрозумілий дизайн, який на підсвідомості зрозумілий кожному, інформація написана дуже зрозуміло та не розмашисто, немає непотрібної інформації – тільки те, що дійсно може знадобитись аудиторії. Дизайн зроблений сучасним, щоб усі не мали сумнівів, що цей сайт новий та на ньому написані сучасні онлайн ризики з котрими може зіткнутися кожен, а не незрозуміла та застаріла інформація.

На цьому сайті міститься інформація про онлайн ризики, як вони проявляються та що з ними робити, як називаються найпопулярніші з них. На цьому сайті є зворотній зв'язок, де підлітки та дорослі можуть залишати свої питання та їм на пошту прийде повідомлення з текстом. Також на сайті є реєстрація на івенти, на яких підлітки будуть опанувати та розказувати про онлайн світ та онлайн ризики та пояснюють, як вони повинні себе застерігати, щоб уникнути всіх ризиків

Мета дослідження

Розробка інформаційного веб-ресурсу з використанням мови програмування PHP.

Завдання дослідження

Розглянути існуючі аналоги веб-ресурсів про онлайн ризики. Проаналізувати, які рішення та технології підходять для розробки інформаційного веб-ресурсу та вибрати відповідні.

Об'єкт дослідження

Об'єктом дослідження є процес розробки повноцінного веб-ресурсу.

Практичне значення одержуваних результатів

Практична значимість роботи представляється у розробці веб-ресурсу, який допоможе підліткам та дорослим розібратися з онлайн ризиками, як їх уникати та що робити, якщо з ними зіткнулися.

Глосарій

Івент – це публічний захід, який спеціально організований чи спланований для досягнення певної мети. Івенти бувають різних типів: конференції, фестивалі, виставки, клуби за інтересами, тощо. Основною метою івентів є привернення уваги до певного продукту спільноти, а також участь і взаємодія з певною аудиторією, яка зацікавлена в цьому.

Кіберзагроза – це потенційна загроза в онлайн світі, яка виникає внаслідок недбалості або зловживання у сфері інформаційних комунікативних технологій та призводить до порушення безпеки конфіденційності та доступності інформації або комп'ютерних систем.

Домагання – це неприпустимий тиск або настирливе вимагання з метою отримання чогось від іншої особи, часто відбувається проти волі або з використанням загрози шантажу чи інших негативних методів.

Кібербулінг – це форма цілеспрямованого насильства або психологічного тиску, яка виникає в Інтернеті. Кібербулінг часто проявляється через образи, погрози, розповсюдження неправдивої інформації або приниження. Він може мати серйозні наслідки для психічного здоров'я жертви, викликаючи депресію, тривогу та втрату самооцінки.

Депресія – це психологічний розлад, який впливає на функціонування людини, характеризується втратою інтересу до життя, емоційною відстороненістю, почуттям пригніченості та іншими симптомами.

Тривога – це стан переляку, невпевненості та непокою, напруження, який може бути спровокований різними ситуаціями в житті. Тривога може виникати через особисті проблеми, робочі труднощі або непередбачувані обставини. Вона здатна суттєво впливати на психічне і фізичне здоров'я

людини. Розуміння причин та вміння керувати цим станом є важливими для підтримки гармонії та спокою в житті.

Суїцидальність – це схильність або намір особи вчинити самогубство. Це серйозний психічний стан, який може бути спровокований депресією, тривогою, страхами та різними іншими факторами.

Тенденції – це нахили або схильності до певних дій, поведінки, думок, які можуть бути спостережені у людей чи груп. Тенденції можуть відобразити загальні настрої суспільства або культурні зміни. Вони часто впливають на прийняття рішень та формування моди, звичаїв і соціальних норм. Аналіз тенденцій допомагає передбачати майбутні зміни та адаптуватися до нових умов.

Соціальні мережі – це веб-платформи де користувачі можуть обмінюватися інформацією, спілкуватися, створювати та розповсюджувати контент спільно працювати над проектом тощо. Такі мережі часто використовуються для комунікації та взаємодії з іншими людьми

Булінг – це форма психологічного насильства, яке проявляється у використанні агресивної поведінки спрямованої проти слабкіших або беззахисних людей зазвичай у шкільному середовищі серед підлітків. Це можуть бути образливі слова, фізичне насильство, викривлення інформації про жертву, відокремлення від соціальних груп, тощо. Булінг може мати серйозні наслідки для психічного та емоційного здоров'я постраждалих осіб.

1 ДОСЛІДЖЕННЯ СИСТЕМ АНАЛОГІВ

1.1 Огляд існуючих рішень

- Сайт Stop–sexting це громадська організація, зроблена за підтримки Київстар. Дуже гарний громадський проект який несе в собі дуже багато різних аспектів які можуть допомогти дітям підліткам та дорослим

розібратися в різних онлайн ризиках. На рисунку 1.1 та 1.2 зображено інтерфейс сайту аналогу Stop–sexting.

Плюси:

- Має приємний та зрозумілий дизайн є вкладки для підлітків та батьків.
- Є чат бот для допомоги підліткам та батькам. Вони викладають дуже багато інформації про те як і що робити щоб уникнути онлайн ризиків. Психологи дуже добре продумали все вони розповідають дітям що якщо вони зіткнулися з цією проблемою то вони не єдині хто з цим зіткнувся що з цим робити, вони зробили сучасний дизайн який с перших секунд на сайті захоплює око та хочеться його листати далі.
- На веб-ресурсі є комікс для підлітків в якому можна прочитати як дівчинка зіткнулась з проблемою і як вона її вирішила.
- Цей веб-ресурс має соціальні мережі що в наш час дуже добре тому що всі сидять в них та можуть підписатися та впізнати багато нового просто зайшовши на них в соціальних мережах.
- Гра для комп'ютера в якій і діти і дорослі зможуть подивитись як підліток постраждав від спілкування в Інтернеті можуть підсліджувати його дії та примати рішення про подальші дії які можуть вплинути на життя. В цій грі підлітки зможуть зрозуміти що не треба робити в онлайн світі та як вирішити, які дії робити далі якщо зіткнулися з проблемою, а батьки зможуть зрозуміти як вберегти підлітка від цих ризиків та допомогти якщо є проблема.

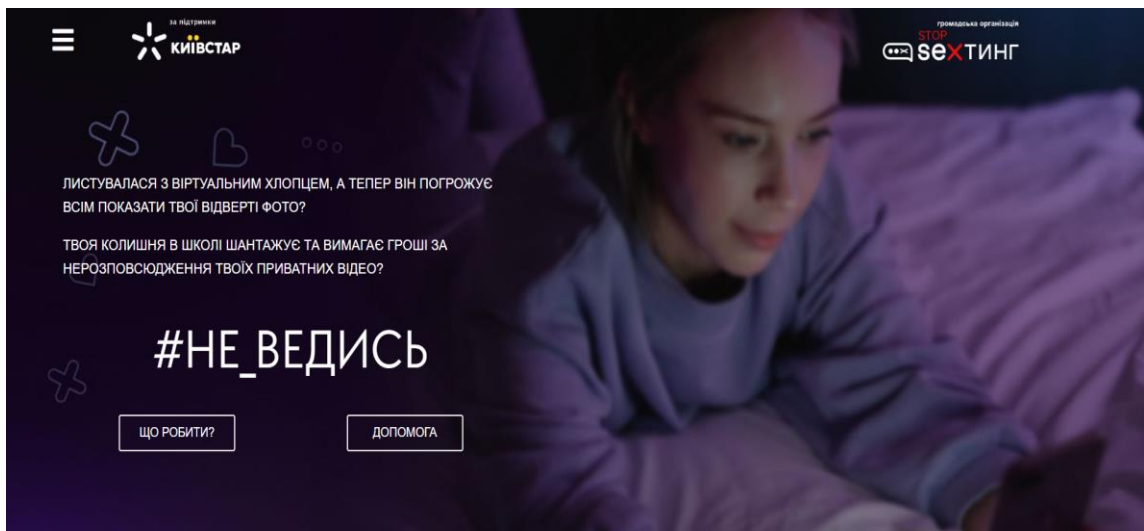


Рисунок 1.1 — Інтерфейс Stop-sexting

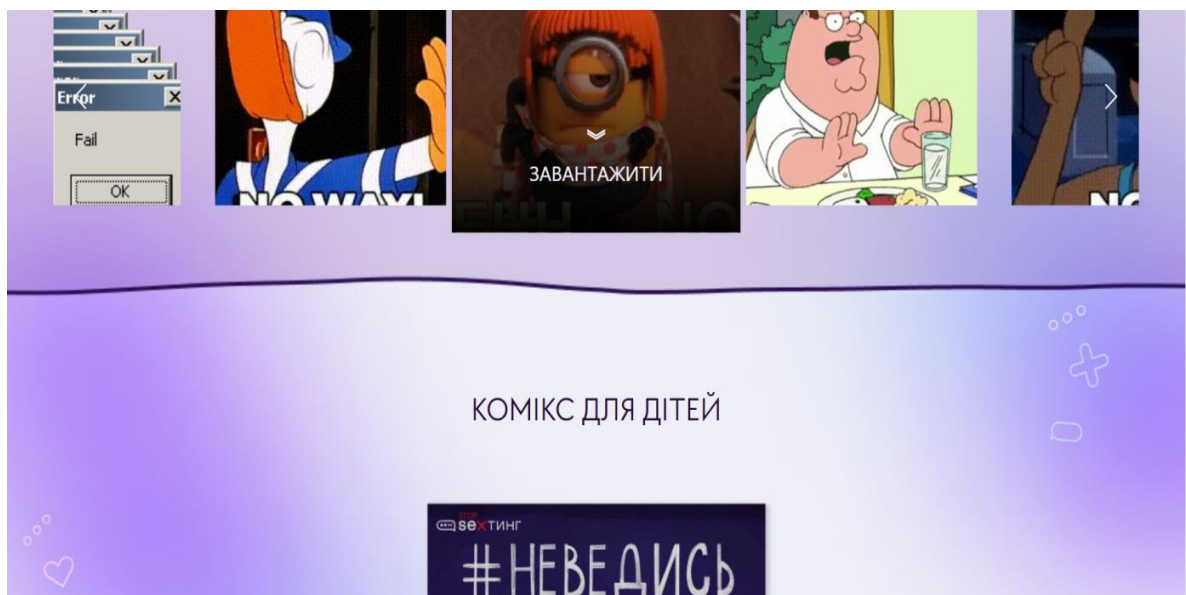


Рисунок 1.2 — Інтерфейс Stop-sexting

- Міністерство освіти і науки в Україні зробили як вкладку про безпеку дітей в Інтернеті. Ця сторінка несе в собі дуже багато інформації яка може бути корисною якщо її всю прочитати та запам'ятати. На рисунку 1.3 та 1.4 зображено інтерфейс сайту аналогу Міністерство освіти і науки в Україні.

Плюси:

- На цьому сайті можна прочитати дуже багато про безпеку в онлайн світі, багато посилань на онлайн ресурси по безпеці.

- Чат бот на який можна перейти по посиланню в Telegram чи Viber який зможе допомогти та розібратися в проблемі яка сталась с підлітком та розповість що робити.
- На веб-ресурсі є дуже корисну функцію для людей з порушення зору, весь текст стає більшим та зручніший для людей з порушенням зору.
- Має приємний дизайн.

Мінуси:

- Не дуже зрозумілий дизайн тому що дуже багато саме посилань та якщо дуже часто переходити по посилання то можна і забути з яким питанням звернулися та що конкретно шукають, в нашому світі с дуже великою психологічною напругою це може дратувати та не всі захочу прочитати сайт до кінця або заходити на нього знову.



Рисунок 1.3 — Інтерфейс Міністерство освіти і науки в Україні

- Довідкові матеріали для проведення заняття "Моя перша подорож безпечним онлайн-простором" для дітей 4-6 років
- Урок "Моя суперсила - безпека в Інтернеті" для 1-2 класу
- Урок "#Не_ведусь: ми – герої безпеки в Інтернеті" для 3-4 класу
- Урок "#не_ведусь" для 5-6 класів
- Квест "#не_ведусь" для 7-8 класів
- Урок "(Не)дитячі стосунки онлайн" для 7-11 класів
- Урок "Інтимні селфі в інтернеті" для 7-11 класів
- Квест "#не_ведусь" для 9-11 класів

- Довідник "Рекомендації щодо онлайн-безпеки для педагогічних працівників"

ВСЕУКРАЇНСЬКА КАМПАНІЯ ПРОТИ КІБЕРБУЛІНГУ DOCUDAYS UA

У межах всеукраїнської кампанії проти кібербулінгу, за сприяння МОН, було проведено **DOCU/ТИЖДЕНЬ проти булінгу** для учнів 5–7 та 8–11 класів. Загалом в проєкті взяли участь 81 000 учнів, 1609 вчителів і 320 шкіл.

Всеукраїнська кампанія проти кібербулінгу створила **відео** "Mr. Nox" і **буклет** для батьків і дітей, який пояснює, як розпізнати кібербулінг і куди звертатися дітям і батькам дітей, які опинились у ситуації кібербулінгу.

Docudays UA дає можливість створити кіноклуб на базі закладу освіти. Кіноклуб дасть змогу отримати доступ, зокрема, і до фільмів із теми цькування, як-от "**Булер**". Зареєструвати кіноклуб у закладі освіти можна [за посиланням](#).

Відео (Матеріали до обговорення)

[Кібербулінг. Як я перестала бути жертвою](#)

Рисунок 1.4 — Інтерфейс Міністерство освіти і науки в Україні

- Дія. Освіта це розробила Дія як вкладники для батьків про безпеку дітей в Інтернеті для батьків. Дія попіклувалась за батьків щоб вони мало змогу допомогти дітям розібратися та самими бути в курсі нових загроз в онлайн світі. На рисунку 1.5 та 1.6 зображено інтерфейс сайту аналогу Дія. Освіта.

Плюси:

- Має приємний та зрозумілий дизайн.
- На запитання в коментарях відповідають люди які працюють в Дія.
- Розроблено відео уроки та після того як їх продивитись всі відео можна пройти тест який покаже як добре прослухана інформація та якщо його добре скласти можна отримати сертифікат.

Мінуси:

- Коментарі в кінці сторінки вибиваються з облого дизайну вони не дуже гармонічно виглядають.
- Немає чат бота або гарячої лінії на яких зможуть допомогти щоб розібратися з проблемою або розповісти що робити якщо з цією проблемою зіткнулися.

- Немає ніякої інформації для дітей щоб пояснити про онлайн ризики з якими вони можуть зіткнутися.

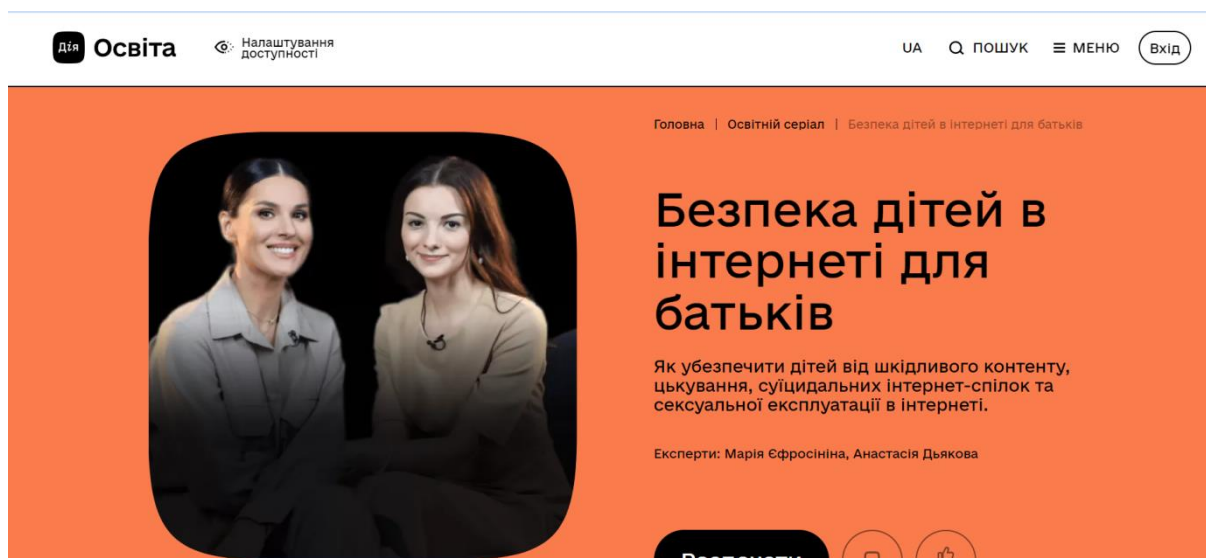


Рисунок 1.5 — Інтерфейс Дія. Освіта



Рисунок 1.6 — Інтерфейс Дія. Освіта

- Центр кращого Інтернету це сайт аналог. Який може допомогти дітям розібратися та розповісти про онлайн загрози в інформаційному світі. На рисунку 1.7 та 1.8 зображено інтерфейс сайту аналогу Центр кращого Інтернету.
Плюси:

- На цьому сайті дуже багато інформації про онлайн ризики.
- Має розробили приємний.
- Цей веб-ресурс має дуже детальну версію для тих людей які мають порушення зору можна вбирати не тільки розмір шрифту але якщо у людини якась із форм дальтонізму то можна вибирати колір фону та тексту, можна зробити щоб картинки були або їх не було або щоб картинка стала повністю чорно білою, також голос який в цьому режимі розповідає куди нажали та що відбувається та ще багато чого є для людей з різними порушеннями зору.

Мінуси:

- Не дуже зрозумілий дизайн, не дуже зрозумілий тому що в ньому все дуже близько друг к другу, накидано одне на одне, занадто багато інформації на моніторі.
- Немає гарячої лінії або чат бота які зможуть допомогти якщо буде якась проблема.



Рисунок 1.7 — Інтерфейс Центр кращого Інтернету

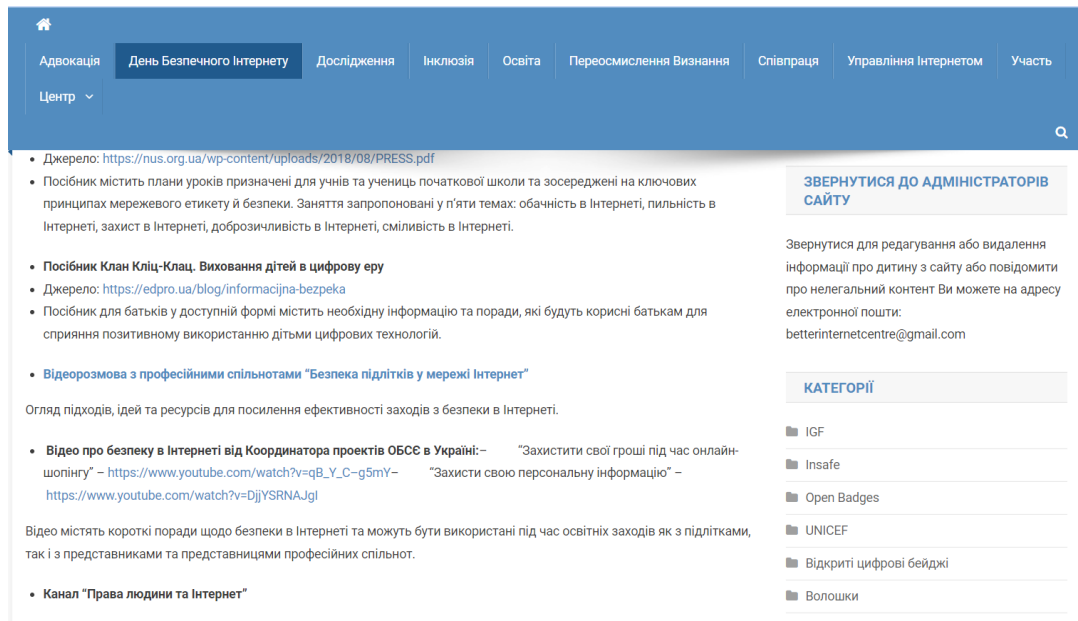


Рисунок 1.8 — Інтерфейс Центр кращого Інтернету

1.2 Аналіз зарубіжних додатків

- **INTERLAND** від компанії Google. Вона навчає дітей онлайн світу як в ньому жити та захищати себе від онлайн ризиків. Підліткам та дітям дуже цікаво та весело грати впізнаючи щось нове про онлайн світ. Просто граючи в гру вони будуть запам'ятовувати правила безпеки в Інтернеті та не робити цих помилок в майбутньому в цій грі представленні чотири гори в яких є рівні які і навчають підлітка правилам безпеки та розважають. Перша гора називається Добре королівство і ній навчають дітей бути добрими та допомагати один одному а якщо вони бачать поганців то блокувати їх та боятися розповідати про них дорослим. Друга гора називається Реальна ріка і в ній дітей навчають що таке шахрайство та які ознаки шахрайства існують, навчають відрізняти правду від підробки та навчають що таке фітінг як повідомити про нього та як уникнути. Третя гора називається Розумна гора і в ній дітей навчають уважно слідкувати та думати чим та з ким вони діляться інформацією, вони зрозуміють силу та наслідки які будуть із спільним використанням, навчаться зберігати конфіденційну

інформацію при собі та будуть розуміти що нікому не можна давати свої особисті данні та розголошувати особисті данні інших людей. Четверта гора називається Вежа скарбів і в ній діти навчаться розуміти яку інформацію потрібно зберігати в безпеці від шахраїв, навчаться зберігати свою особисту інформацію в безпеці та навчаться встановлювати пароль зрозуміють що в паролі потрібні поєднання різних регістрів, різних мов, цифр та символів, зрозуміють що пароль повинен добре запам'ятовуватись але бути складним для шахраїв. В кінці гри дитина чи підліток отримає сертифікат про те що він пройшов гру та тепер знає як бути в онлайн світі та як себе захищати. На рисунку 1.9 та 1.10 зображено інтерфейс сайту аналогу INTERLAND.



Рисунок 1.9 — *Інтерфейс INTERLAND*

Плюси:

- Дуже цікава та красива гра на сайті.
- Дуже хороший та яскравий інтерфейс.
- В ній все управління йде через клавіатуру на комп'ютері чи ноутбуці.
- Після кожного рівня йде міні тест на то як діти запам'ятали що і коли треба робити.

Мінуси:

- Для батьків нічого немає в цій грі але я вважаю що якщо вони дадуть просто можливість пограти своїй дитині в цю гру або заохотять разом пограти то дитина сама все зрозуміє та запам'ятає як захищати себе в Інтернеті та жити в цьому онлайн світі що зараз є навколо нас.



Рисунок 1.10 — Інтерфейс INTERLAND

Отже, провівши аналіз систем аналогів зрозуміло що онлайн ризики стали великою проблемою в онлайн світі. Такі ресурси мають бути що б аудиторія різного віку могла розібратися та не попадати на ці загрози. Ці сайти надають інформаційну допомогу підліткам та дорослим, тому що всім треба зрозуміти які зараз існують проблеми, що проблеми існують не тільки в офлайн житті а й онлайн світі. Також такі сайти мають великий спектр різних функцій для різних людей щоб всі все зрозуміли та могли допомогти один одному.

Проведений аналіз сайтів аналогів онлайн ризиків дозволяє виділити їх сильні та слабкі сторони. Зрозуміло що найпопулярніший сайт володіє візуально привабливим дизайном, великим та якісним контентом. За врахуванням найкращих сайтів аналогів можна зробити хороший зрозумілий та привабливий веб-ресурс по онлайн ризикам.

2 ОПИС ТЕХНОЛОГІЙ

У розробці мого веб-ресурсу «Онлайн ризики серед підлітків» було застосовано багато сучасних технологій для забезпечення ефективного та безпечної платформи для підлітків та дорослих. У веб-технологіях та фреймворках я використовувала популярні мови програмування такі як HTML, CSS JavaScript та PHP.

HTML (Hypertext Markup Language) — це мова розмітки гіпертексту, необхідна для розміщення різних елементів на веб-сторінці, таких як текст, зображення, таблиці та відео. Коли ви відвідуєте сайт, браузер завантажує HTML файл, що містить інформацію про структуру і вміст сторінки. HTML формує візуальну основу сайту, визначаючи розташування і базовий дизайн елементів, але не може самостійно «запускати» сайт.

Варто зазначити, що HTML не є мовою програмування. Це мова розмітки, яка повідомляє браузеру, як відображати веб-сторінки, які ви відвідуєте. HTML може бути як простим, так і складним, залежно від бажання веб-дизайнера. Він складається з набору елементів, які використовуються для організації та форматування контенту, щоб надати йому певного вигляду і поведінки.

Елементи HTML включають огорожуючі теги, які можуть перетворювати текст або зображення на посилання, робити текст курсивом, змінювати розмір шрифту і виконувати багато інших функцій. Ці теги використовуються для того, щоб різні частини контенту відображалися або функціонували певним чином на веб-сторінці.

CSS (Cascading Style Sheets) — це мова, яка використовується для опису зовнішнього вигляду документа. Вона відповідає за те, як виглядають веб-сторінки: колір фону і декоративних елементів, розмір і стиль шрифтів. CSS працює разом з іншою мовою розмітки — HTML, який відповідає за розміщення елементів на сторінці.

CSS — це код, який використовується для стилізації веб-сторінки. Знання основ CSS допоможе вам зрозуміти, що необхідно для початку роботи з цією мовою. Як і HTML, CSS не є мовою програмування і не є мовою розмітки, а представляє собою мову таблиці стилів.

CSS дозволяє вибірково застосовувати стилі до елементів у документах HTML. Це означає, що за допомогою CSS можна задавати різні візуальні характеристики для елементів веб-сторінки, покращуючи її зовнішній вигляд і зручність використання.

JavaScript це мова програмування, яка додає інтерактивні елементи на ваш веб-сайт. Його функції можуть покращити зручність взаємодії користувача з веб-ресурсом: від оновлення стрічки новин у соціальних мережах до відображення анімації та інтерактивних карт. JavaScript використовується розробниками для створення інтерактивних веб-сторінок і дозволяє створювати ігри, обробляти натискання кнопок і введення даних у форми, а також застосовувати динамічні стилі та анімацію.

JavaScript (скорочено "JS") — це потужна і динамічна мова програмування, яка застосовується до HTML-документів, забезпечуючи динамічну інтерактивність на веб-сторонках. JavaScript є мовою програмування для розробки скриптів, що виконуються на стороні клієнта, що робить його однією з базових технологій у всесвітній мережі Інтернет.

JavaScript неймовірно універсальний і легко засвоюється новачками. З досвідом ви зможете розробляти ігри, анімовану 2D і 3D графіку, повнофункціональні застосунки з базами даних та багато іншого. Наприклад, карусель зображень, випадаюче по кліку меню і динамічно змінювані кольори елементів на веб-сторінці, які ви бачите під час перегляду сторінок в Інтернеті, виконані за допомогою JavaScript. Ця мова сама по собі компактна, але дуже гнучка, і безліч інструментів, створених на її основі, значно розширюють її можливості з мінімальними зусиллями.

Усі три мови які я використовую для частини користувача HTML, CSS та JavaScript завжди використовуються в спільності для створення інтерактивних цікавих та привабливих веб-ресурсах на яких цікаво сидіти.

У частині яку не бачить звичайний користувач мого веб-ресурсу тобто бази даних я використовую MySQL та PHP вони забезпечують надійність та швидкість роботи с великими обсягами даних. Вони дозволяють зручно взаємодіяти з користувачем, зберігати данні та налаштування на моєму веб-ресурсі.

MySQL (Structured Query Language) — це система управління базами даних (СУБД), що дозволяє створювати і контролювати бази даних. Основна її функція — робота з реляційними базами даних, для чого використовується стандартизована мова запитів SQL. MySQL допомагає організувати дані та ефективно взаємодіяти з ними.

Основні завдання, які вирішує MySQL, включають коректне розподілене зберігання даних на сервері, швидкий пошук потрібної інформації з використанням мови SQL, а також ідентифікацію, обробку, перетворення і відправлення даних. За допомогою цієї СУБД можна створювати, редагувати і видаляти записи в базі даних, забезпечуючи гнучкість і керованість даними.

MySQL є високоефективною та надійною СУБД для веб-середовища. Вона ідеально підходить для сайтів та веб-застосунків, які потребують виконання простих транзакцій. MySQL забезпечує стабільну і швидку роботу з невеликими та середніми обсягами даних, що робить її оптимальним вибором для багатьох проєктів.

PHP (Hypertext Preprocessor) — це популярна мова програмування загального призначення з відкритим вихідним кодом. Він спеціально розроблений для веб-розробки і дозволяє впроваджувати код безпосередньо в HTML, що робить його зручним для створення динамічних веб-сторінок і застосунків.

Основна область застосування PHP — написання серверних скриптів. Він може виконувати всі завдання, притаманні іншим CGI програмам, такі як

обробка даних форм, генерування динамічних сторінок, а також відправка і прийом cookies. Проте можливості PHP цим не обмежуються; він надає широкий спектр функцій для веб-розробки, включаючи взаємодію з базами даних та управління сесіями.

Спочатку PHP створювався для веб-застосунків, але з часом еволюціонував у мову програмування загального призначення. Завдяки динамічній типізації, код на PHP легко читати і розуміти, що прискорює процес розробки. Будучи інтерпретованою мовою, PHP дозволяє вносити зміни і тестувати їх у реальному часі без необхідності компіляції, що робить його ідеальним інструментом для швидкої та гнучкої веб-розробки.

Тому MySQL та PHP можуть зробити потужний вплив на створення мого веб-ресурсу та допомогти зробити зручні для підлітків форми які вони будуть заповнювати щоб дізнатися про питання які їх цікавлять або зареєструватися на якийсь з івентів.

2.1 Обґрунтування баз даних

Обидва MySQL і Microsoft SQL Server є потужними системами керування базами даних, кожна з яких має свої переваги і недоліки. При виборі між ними варто враховувати кілька ключових факторів.

MySQL, як відомий відкритий джерела база даних, відомий своєю безкоштовною ліцензією та широким спектром можливостей. Він здатний працювати ефективно на великих об'ємах даних, особливо в середовищах з великою кількістю одночасних запитів. Завдяки своїй відкритості, MySQL має велику спільноту розробників, що забезпечує велику кількість ресурсів та підтримку для користувачів.

З іншого боку, Microsoft SQL Server відомий своєю масштабованістю та високою стійкістю до великих навантажень. Він надає різноманітні інструменти та функції для роботи з даними, включаючи розширену систему безпеки та підтримку бізнес процесів. Особливо в корпоративному

середовищі, де вимагається висока стабільність і безпека, Microsoft SQL Server може бути перевагою.

Коли я розглядаю вибір між MySQL та Microsoft SQL Server для мого проекту, я приходжу до висновку, що MySQL більш підходить для моїх потреб. Оскільки у мене обмежений бюджет і проект потребує ефективної роботи з великим обсягом даних, MySQL з його відкритою ліцензією є привабливим варіантом. Відкритий характер MySQL робить його доступним для проектів будьякого розміру, особливо для стартапів та малих підприємств. Додатково, MySQL має швидкодіючу систему та добре справляється з великим обсягом одночасних запитів, що є важливим для мого проекту, який буде мати високий трафік користувачів.

2.2 Порівняльний аналіз PHP та Java

PHP і Java — два з найбільш поширених мов програмування у світі. Обидві мови призначені в основному для розробки веб-додатків, але вони також використовуються в інших галузях, таких як розробка додатків для мобільних пристроїв і корпоративних систем.

Одним з ключових відмінностей між PHP і Java є їх синтаксис і типізація. PHP зазвичай вважається простішим для вивчення і використання завдяки своєму простому та менш формальному синтаксису. Він динамічно типізований, що означає, що змінні не потребують оголошення типу даних. Тоді як Java має більш жорсткий і формальний синтаксис, а також є статично типізованою мовою, що вимагає явного вказання типу даних змінних. Це робить Java більш передбачуваною та підходящою для створення великих проектів з високими вимогами до надійності.

Ще одним важливим аспектом є продуктивність. Java, як компільована мова, часто вважається більш продуктивною порівняно з PHP, яка інтерпретується. Однак сучасні оптимізації та JIT-компіляція можуть

покращити продуктивність PHP. У будь-якому випадку вибір між цими мовами залежить від конкретних вимог проекту та бажаної продуктивності.

Крім того, масштабованість та надійність також важливі при виборі між PHP і Java. PHP часто використовується для створення невеликих і середніх веб-додатків, тоді як Java більш підходить для великих корпоративних проектів, де вимагається висока надійність і масштабованість.

Залежно від екосистеми та підтримки, обидві мови мають активні спільноти розробників і велику кількість інструментів та фреймворків для роботи. PHP має широке застосування в веб-розробці завдяки популярним фреймворкам, таким як Laravel і Symfony. Java також має обширну екосистему з фреймворками, такими як Spring і Hibernate, які широко використовуються для створення великих корпоративних додатків.

У контексті моїх потреб, PHP є більш підходящим вибором для мого проекту. Завдяки своїм веб-орієнтованим можливостям, швидкості розробки, великій спільноті розробників і низьким витратам, PHP дозволяє ефективно створювати веб-додатки і забезпечує доступність ресурсів для підтримки проекту.

Також для створення бази даних я також використовувала XAMPP Control Panel та PHPMyAdmin які допомогли мені зробити сервер розробити та підключити базу даних.

XAMPP Control Panel — це панель управління через яку можна вклучати та відключати хост для того щоб сервер працював.

Для повноцінного функціонування сайту можна використовувати збірку веб-сервера, яка містить всі необхідні додатки і не потребує окремого налаштування кожного компонента. Така збірка дозволяє розробляти та тестувати проекти на локальному комп'ютері без необхідності постійного підключення до мережі. Завдяки цьому розробники можуть працювати без високошвидкісного Інтернету та хостингу.

Проект є крос-платформним, що означає його стабільну роботу на всіх популярних операційних системах, таких як Windows, Linux і Mac OS. Це

робить його зручним і універсальним інструментом для розробників, незалежно від використовуваної платформи.

Однією з найвідоміших збірок локального сервера є ХАМРР. Цей інструмент популярний завдяки високій якості, широкому набору функцій та обширній інформаційній підтримці. Примітно, що ХАМРР користується більшою популярністю серед західних розробників, хоча причини цього залишаються незрозумілими.

РНРMyAdmin — це веб-застосунок який допомагає с адміністрацією СУБД MySQL. В ньому можна зробити базу даних та таблиці, легко продивлятися вміст таблиць та легко підключитись до них для веб-сайту.

PhpMyAdmin дозволяє адмініструвати сервер MySQL через браузер, запускати SQL- команди та переглядати вміст таблиць і баз даних. Це застосування популярне серед веб-розробників, оскільки спрощує керування СУБД MySQL без необхідності безпосереднього введення SQL-команд.

PhpMyAdmin є веб-застосунком, який надає зручний графічний інтерфейс для керування базами даних. Його встановлюють на сервер, де зберігається сайт, і потім відкривають у браузері для віддаленого адміністрування СУБД. Це дозволяє віддавати команди натисканням кнопок у веб-інтерфейсі, замість написання їх вручну в терміналі.

PhpMyAdmin призначена для проектів, які використовують веб-сервер Apache, мову РНР та системи MySQL або MariaDB. Процес встановлення і налаштування може незначно відрізнятись в залежності від операційної системи. За стосунок постійно оновлюється і вдосконалюється, при цьому зовнішній вигляд панелі керування залишається майже незмінним. PhpMyAdmin має відкритий вихідний код і поширюється безкоштовно, що разом із багатофункціональністю та зручним інтерфейсом робить його дуже популярним.

2.3 Переваги та недоліки системи яка застосована

Переваги:

- Користуватися дружелюбність: мій веб-ресурс спрямований на підлітків та він розроблений з акцентом на простоту та дружелюбність тому що кожен хто зайде на мій веб-ресурс не важливо підлітки чи дорослі повинні інтуїтивно зрозуміти цей сайт як і де знаходиться інформація, як заповнити форми з питаннями чи як зареєструватися на івент та провести корисно на ньому час. Головна мета мого за стосунку це забезпечити зрозумілий доступ до інформації про онлайн ризики зокрема для аудиторії яка може будуть менш досвідченою в онлайн світі щоб вони навчилися в ньому жити та знали які ризи існують як їх запобігти та якщо це сталось то як позбутися цієї проблеми.
- Актуальність Інформації: Система регулярно буде оновлюватись щоб відображати найновіші онлайн ризики для підлітків щоб користувачі завжди знали що нового придумали шахраї та як цього запобігти. Це дозволить тримати інформацію на платформі завжди актуальною та відповідною до сучасного онлайн світу.
- Безпека та середовищ: В моєму веб-ресурсі застосовані анонімні форми в яких можна написати не своє справжнє ім'я чи зробити нову пошту щоб отримати повідомлення про івент або відповідь на питання. Там не має різниці хто пише або записується важливо щоб людина отримала відповідь на своє питання або лист що вона зареєструвалась на івент важливо щоб людина зрозуміла які є ризики про неї ніхто не буду збирати базу даних та якимось шкодити їй тільки спробують допомогти. В цьому й полягає безпека в моєму веб-ресурсі все анонімно і ніхто нічого не буде вимагати.

Недоліки:

- Можливі технічні проблеми: в залежності від сервера можуть виникнути технічні проблеми, такі як відмова сервера або мережеві перебої. Це

може вплинути на доступність ресурсу та вимагати невідкладних технічних втручань.

- **Необхідність регулярних оновлень:** для забезпечення безпеки система вимагає регулярних оновлень включаючи нові сторінки та заходи безпеки. Процес оновлення може також призвести до тимчасового відключення ресурсу для користувачів.
- **Можливість вразливостей коду:** написання коду без належних заходів безпеки може призводити до вразливостей, які шахраї можуть використовувати для атак на сервер або несанкціонованого доступу. Треба ретельно тестування та аудит коду є важливим етапом для забезпечення стійкості та надійності системи від шахраїв.

Незважаючи на недоліки постійна увага до безпеки та регулярні оновлення можуть забезпечити стабільну та ефективно працездатність мого веб-ресурсу який незважаючи на це зможе навчити людей як правильно жити в Інтернет світі та не попадатися на онлайн ризики від шахраїв.

У процесі аналізу основних інструментів для створення веб-сайтів та характеристик технологій, використаних у створенні тематичного блогу, було виявлено використання HTML для розробки структури сторінок, CSS для стилізації та оформлення, а також JavaScript для інтерактивності та динамічної поведінки. У ролі мови програмування на серверному боці був обраний PHP, який відзначається ефективністю та простотою використання. Базою даних виступала MySQL, що є потужною системою управління реляційними базами даних, забезпечуючи надійне зберігання та управління структурованими даними для розробників, що використовують їх для створення та масштабування програм. Використання таких технологій забезпечує якісну та продуктивну розробку веб-сайту з можливістю оптимізації продуктивності та підвищення масштабованості.

3 РОЗРОБКА ІНФОРМАЦІЙНОГО ВЕБ-РЕСУРСУ

3.1 Вимоги до веб-ресурсу та користувачів

Головною метою аналізу вимог та користувачів полягала в тому щоб з'ясувати потреби та очікування користувачів цього веб-ресурсу а також вимог до мого веб-застосунку. На рисунку 3.1 зображено Use Case Diagrams яка показує користувачів цього веб-ресурсу.

Тож я з'ясувала що основним користувачем будуть підлітки, та їх батьки. Тому що саме цим людям потрібно розібратися з онлайн ризиками саме підлітки все більш до цього схильні. Цей веб-ресурс був спеціально розроблений для того щоб користувачі розуміли які на часі є проблеми з якими вони можуть зіткнутися та що робити.

Також користувачами мого веб-ресурсу будуть адміністратори та психологи які будуть відповідати на пошту користувачам відслідковувати яким користувачам потрібна допомога та надавати її.

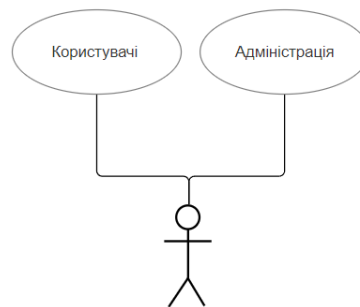


Рисунок 3.1 — *Use Case Diagrams*

Таким чином проаналізувавши які користувачі будуть заходити та користуватися цим веб-ресурсом можна виділити декілька пунктів:

- Сучасний дизайн — він потрібен щоб зацікавити підліткам щоб їм було комфортно та приємно читати цей веб-ресурс
- Адаптивність — в сучасному світі всі сидять в телефонах різних марок та різних діагоналей дисплеїв , різних планшетах ноутбуках та

комп'ютерах у всіх них різні параметри тому треба щоб сайт легко адаптувався під різні екрани

- Зрозумілий інтерфейс — тому що окрім підлітків на ресурс також будуть заходити дорослі батьки вчителі тощо. Тому проаналізувавши потрібен зрозумілий інтерфейс на якому легко зрозуміти всім куди нажимати щоб подивитись інформацію.
- Короткі та змістові тексти — було зрозуміло що зараз люди не люблять читати довгі та не цікаві тексти, нікому не цікаво дочитувати до кінця тому що змінилось покоління зараз всі зайняті та довгі тести ніхто не буде сидіти та читати. Всі люблять короткі та зрозумілі тести щоб було все коротко та чітко щоб це не займало багато часу.
- Анонімність — зараз всі дуже хвилюються над анонімністю тому щоб задати питання треба тільки пошта яку побачить психолог та відповідь. Це зроблено для того щоб не переживали що хтось вкраде данні імена пошту яка проблема у цього користувача. Тому що сама проблема повинна бути анонімною користувачі не повинні хвилюватись коли вони пишуть та чекають відповіді.

Тому роблячи висновок з цього аналізу можна зрозуміти який повинен бути ресурс щоб допомагати підліткам з їх проблемами, які користувачі будуть та що для них важливо та потрібно.

3.2 Архітектура системи

Розробка інформаційного веб-ресурсу з використанням мови програмування PHP представлена у цій роботі, підрозділяється на дві частини:

- Клієнтський шар (frontend)
- Серверний шар (backend)

Клієнтський шар (Frontend)

Цей шар відповідає за взаємодію з користувачем і включає наступні технології:

HTML: Використовується для структури сторінок. Визначає розмітку та контент. Створення сторінок з елементами, такими як заголовки, параграфи, форми, кнопки тощо.

CSS: Використовується для стилізації сторінок. Визначає зовнішній вигляд і оформлення. Визначення стилів для елементів сторінки, таких як колір, шрифти, розміри, розташування.

JavaScript: Використовується для додавання інтерактивності та динамічного контенту на сторінках. Обробка подій (наприклад, кліки, слайдер)

Серверний шар (Backend)

Цей шар відповідає за обробку логіки та взаємодію з базою даних. Основна технологія тут - PHP:

PHP: Використовується для написання серверної логіки, обробки питань від користувачів, взаємодії з базою даних і генерації HTML-сторінок на сервері.

Приклад функціональності:

Обробка форм: Отримання даних форми від користувача, і збереження даних у базі даних.

Генерація сторінок: Динамічне створення HTML-контенту на основі даних із бази даних.

Аутентифікація: Управління реєстрацією, входом і виходом користувачів.

База даних

Цей шар відповідає за зберігання даних і взаємодію з ними. Використовувана база даних — MySQL:

MySQL: СУБД для зберігання даних, таких як користувачі, пости, коментарі тощо.

Приклад роботи з базою даних:

Запити: Написання SQL — запитів для витягування, вставки, оновлення та видалення даних.

База даних інтерфейсу зображена на рисунку 3.2

quest users	quest submissions	quest quest
id : int	id : int	id : int
full_name : varchar(255)	name : varchar(100)	first_name : varchar(255)
login : varchar(255)	email : varchar(100)	email : varchar(60)
email : varchar(255)	phone : varchar(100)	comment : varchar(200)
password : varchar(255)		
avatar : varchar(500)		

Рисунок 3.2 — База даних за стосунку

3.3 Інтерфейс системи

Коли адміністратор або психолог хочуть зайти в адміністративну панель то вони заходять на головну сторінку та нажимають в самому низу на кнопку яку майже не видно та переходять на вікно входу, вони там бачать що можуть ввести логин та пароль за яким зареєструвалися, код цього вікна наведено у лістингу1

Лістинг 1 Вхід до профілю адміністратора

```
<?php
session_start();

if (isset ($_SESSION['user'])) {
    header('Location: profile.php');
}
?>
```

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Авторизація та реєстрація</title>
    <link rel="stylesheet" href="css/main.css">
</head>
<body>
<!-- Форма авторизації -->
    <form action="vendor/signin.php" method="post">
        <label> Логін </label>
        <input type="text" name="login"
placeholder="Введіть свій логін">
        <label>Пароль</label>
        <input type="password" name="password"
placeholder=" Введіть пароль">
        <button type="submit">Войти</button>
    <p>
        У вас ще нема аккаунта? - <a
href="/auth/register.php">Зареєструйтесь</a>!
    </p>
    <?php
        if (isset($_SESSION['message'])) {
            echo '<p class="msg"> ' .
$_SESSION['message'] . ' </p>';
        }
        unset($_SESSION['message']);
    ?>
    </form>
</body>
</html>

```

Детальний опис коду

Початок сеансу: `session_start()` ініціює сесію або відновлює існуючу. Це необхідно для зберігання інформації про авторизованого користувача.

Перевірка авторизації: Якщо сесія вже містить інформацію про користувача (`$_SESSION['user']`), відбувається перенаправлення на сторінку профілю (`profile.php`).

Форма авторизації: Містить поля для введення логіну та паролю, а також кнопку для відправки форми. Дані форми відправляються на обробку до скрипту `vendor/signin.php`.

Сценарій обробки авторизації

Сценарій `vendor/signin.php` виконує наступні кроки:

Приймає дані з форми.

Перевіряє правильність логіну та паролю.

У разі успіху створює сесію для користувача та перенаправляє його до адміністративної панелі.

У разі невдачі відображає повідомлення про помилку.

Після того як адміністратор натиснув кнопку увійти якщо все добре то форма з'єднується з базою даних де він порівнює логін та пароль та якщо все вірно то виводить з бази даних було вказано ПІБ, емейл та аватар на іншу сторінку цей код наданий у лістинг 2

Лістинг 2 Перевірка логіну та паролю

```
<?php
    session_start();
    require_once 'connect.php';
    $login = $_POST['login'];
    $password = md5($_POST['password']);
    $check_user = mysqli_query($connect, "SELECT * FROM
`users` WHERE `login` = '$login' AND `password` =
'$password'");
```

```
if (mysqli_num_rows($check_user) > 0) {
    $user = mysqli_fetch_assoc($check_user);

    $_SESSION['user'] = [
        "id" => $user['id'],
        "full_name" => $user['full_name'],
        "avatar" => $user['avatar'],
        "email" => $user['email']
    ];

    header('Location: ../profile.php');
} else {
    $_SESSION['message'] = 'Не верный логин или
пароль';

    header('Location: ../index.php');
}
?>
<pre>
<?php
print_r($check_user);
print_r($user);
?>
</pre>
```

Детальный опис коду

Початок сесії

Кожен PHP-скрипт, що працює з сесіями, повинен починатися з виклику `session_start()`. Це дозволяє зберігати та відновлювати дані користувача між різними сторінками. У нашому випадку це необхідно для зберігання інформації про авторизованого користувача.

Підключення до бази даних

Далі, за допомогою `require_once 'connect.php'`, підключається файл, що встановлює з'єднання з базою даних. Це забезпечує доступ до бази даних для подальших операцій.

Отримання даних форми

Дані, введені користувачем у форму на сторінці входу, передаються через метод POST. Ми зчитуємо ці дані та зберігаємо їх у змінних `$login` та `$password`. Важливо зауважити, що пароль хешується за допомогою функції `md5()` перед перевіркою в базі даних.

Запит до бази даних

Запит до бази даних виконується за допомогою функції `mysqli_query()`. Ми шукаємо користувача з введеними логіном та паролем у таблиці `users`. Якщо такий користувач існує, функція `mysqli_num_rows()` поверне більше нуля рядків.

Перевірка результату запиту

Якщо знайдено користувача, дані про нього зберігаються у масиві сесії `$_SESSION['user']`. Це дозволяє зберігати інформацію про користувача для подальшого використання на інших сторінках.

Перенаправлення користувача

У разі успішної авторизації користувач перенаправляється на сторінку профілю за допомогою `header('Location: ../profile.php')`. Якщо ж введено невірні дані, встановлюється сесійне повідомлення про помилку і користувач перенаправляється назад на сторінку входу

Якщо логін та пароль вірні то адміністратора відкривається форма на якій він може перейти до питань які задали користувачі або перейти на

вкладнику івенти де прописано хто зареєструвався на івент, код наведено в лістингу 3.

Лістинг 3 Профіль користувача.

```
<?php
session_start();
if (!$_SESSION['user']) {
    header('Location: /');
}
?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Авторизація та реєстрація</title>
    <link rel="stylesheet" href="assets/css/main.css">
</head>
<body>
    <!-- Профіль -->
    <form>
        
        <h2 style="margin: 10px 0;"><?=
$_SESSION['user']['full_name'] ?></h2>
        <a href="#"><?= $_SESSION['user']['email'] ?></a>
        <a href="vendor/logout.php"
class="logout">Выход</a>
        <a href="CRUD/index1.php" class="link">Питиння </a>
```

```

        <a href="CRUD1/index.php" class="link">Івенти</a>
    </form>
</body>
</html>

```

Детальний опис коду

Початок сесії та перевірка аутентифікації

Кожен скрипт, що працює з сесіями, повинен починатися з виклику `session_start()`. Це дозволяє зберігати та відновлювати дані користувача між різними сторінками. У цьому випадку, якщо користувач не авторизований, він перенаправляється на головну сторінку.

HTML-структура профілю

HTML-структура профілю містить інформацію про адміністратора та надає посилання на різні секції.

Відображення інформації про користувача

В тілі HTML-документа ми використовуємо дані сесії для відображення аватара, повного імені та email-адреси користувача.

Посилання на інші секції

Також є посилання для виходу з профілю (logout) та для переходу до секцій "Питання" та "Івенти". Це дозволяє адміністратору швидко переключатися між різними функціональними частинами системи.

Якщо адміністратор натисне кнопку питання то він перейде до сторінки де відображаються всі емейли питання на які треба відповідати, код наведено в лістингу 4.

Лістинг 4. Показ таблиці на сторінці

```
<?php
```



```
session_start();

if (!$_SESSION['user']) {
    header('Location: /');
}

/*
 * Подключаем файл для получения соединения к базе данных
 (PhpMyAdmin, MySQL)
 */

require_once 'config/connect.php';

?>

<!doctype html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Products</title>

</head>

<style>

    th, td {

        padding: 10px;

    }

    th {

        background: #606060;

        color: #fff;

    }

    td {

        background: #b5b5b5;

    }

}
```

```
body {  
    background-color: #606060;  
}  
</style>  
<body>  
    <table>  
        <tr>  
            <th>ID</th>  
            <th>Email</th>  
            <th>Коментар</th>  
        </tr>  
        <?php  
            /*  
            * Робимо виборку зі всіх рядків із таблиці  
"products"  
            */  
            $products = mysqli_query($connect, "SELECT *  
FROM `quest`");  
            /*  
            * Перетворюємо отримані дані на нормальний  
массив  
            */  
            $products = mysqli_fetch_all($products);  
            /*  
            * Перебираємо масив і рендер HTML з даними з  
массиву  
            * Ключ 0 - id  
            * Ключ 1 - email
```

```

* Ключ 3 - Коментар
*/
foreach ($products as $product) {
    ?>
        <tr>
            <td><?= $product[0] ?></td>
            <td><?= $product[1] ?></td>
            <td><?= $product[3] ?></td>
        </tr>
    <?php
    }
    ?>
</table>
</body>
</html>

```

Детальний опис коду

Початок сесії та перевірка аутентифікації

Як і в попередніх прикладах, скрипт починається з виклику `session_start()`, що дозволяє використовувати сесійні дані для перевірки автентичності користувача. Якщо користувач не авторизований, його перенаправляють на головну сторінку.

Підключення до бази даних

Для отримання даних з бази даних необхідно підключити файл з налаштуваннями з'єднання з базою. Це робиться за допомогою `require_once 'config/connect.php'`.

HTML-структура сторінки

HTML-структура сторінки містить заголовок та стилі для таблиці, що відображає запитання. Стилi задають оформлення таблиці та її елементів, таких як комірки і заголовки.

Запит до бази даних та відображення даних

З бази даних робиться вибірка всіх рядків з таблиці quest, яка містить запитання від користувачів. Отримані дані перетворюються на звичайний масив за допомогою `mysqli_fetch_all($products)`. Потім кожен рядок масиву перебирається у циклі `foreach`, і дані відображаються у вигляді таблиці.

Закриття HTML-структури

HTML-документ завершується закриттям таблиці, тіла та HTML-документу.

Якщо є новий співробітник то він має натиснути на кнопку реєстрації та заповнити форму щоб мати свій акаунт та продивлятися бази даних, код наведено в лістингу 5.

Лістинг 5. Реєстрація нового адміністратора.

```
<?php
    session_start();
    if (isset($_SESSION['user'])) {
        header('Location: profile.php');
    }
?>
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Авторизация и регистрация</title>
```

```

    <link rel="stylesheet" href="assets/css/main.css">
</head>
<body>
    <!-- Форма регистрации -->
    <form action="/auth/vendor/signup.php" method="post"
enctype="multipart/form-data">
        <label>ПІБ</label>
        <input type="text" name="full_name"
placeholder="Введіть своє повне ім'я">
        <label>Логін</label>
        <input type="text" name="login" placeholder="
Введіть свій логін">
        <label>Пошта</label>
        <input type="email" name="email" placeholder="
Введіть адресу своєї пошти">
        <label>Зображення профілю</label>
        <input type="file" name="avatar">
        <label>Пароль</label>
        <input type="password" name="password"
placeholder=" Введіть пароль">
        <label>Підтверження пароля</label>
        <input type="password" name="password_confirm"
placeholder="Підтвердіть пароль">
        <button type="submit">Войти</button>
    <p>
        У вас вже є аккаунт? - <a
href="/">авторизуйтеся</a>!
    </p>
    <?php
        if (isset($_SESSION['message'])) {

```

```

        echo '<p class="msg"> ' .
$_SESSION['message'] . ' </p>';
    }
    unset($_SESSION['message']);
?>
</form>

</body>
</html>

```

Детальний опис коду

Початок сесії та перевірка автентичності

Скрипт починається з виклику `session_start()`, що дозволяє використовувати сесійні дані. Якщо користувач вже авторизований (сесія існує), його перенаправляють на сторінку профілю.

HTML-структура сторінки

HTML-структура містить заголовок, стилі та форму реєстрації. Форма містить поля для введення ПІБ, логіну, електронної пошти, зображення профілю, пароля та підтвердження пароля. Форма також має кнопку для відправки даних на сервер та посилання для переходу до сторінки авторизації, якщо у користувача вже є обліковий запис.

На головній сторінці веб-ресурсу є слайдер з восьми слайдів, де кожен слайд містить зображення та посилання на іншу сторінку. Слайдер має кнопки для переходу до наступного та попереднього слайдів, пагінацію, а також підтримує керування за допомогою колеса миші та клавіатури. Бібліотека `Swiper` надає гнучкість і дозволяє легко налаштувати слайдер під різні потреби, код наведено в лістингу 6.

Лістинг 6. Слайдер

```
<div class="swiper mySwiper">
  <div class="swiper-wrapper">
    <div class="swiper-slide"><a
href="page1.html">
      
    </a></div>
    <div class="swiper-slide"><a
href="page2.html">
      </a></div>
    <div class="swiper-slide"><a
href="page3.html">
      </a></div>
    <div class="swiper-slide"><a
href="page4.html">
      </a></div>
    <div class="swiper-slide"><a
href="page5.html">
      </a></div>
    <div class="swiper-slide"><a
href="page6.html">
      </a></div>
    <div class="swiper-slide"><a
href="page7.html">
      </a></div>
    <div class="swiper-slide"><a
href="page8.html">
```

```

        </a></div>

</div>

<div class="swiper-pagination"></div>

<div class="swiper-button-next"></div>

<div class="swiper-button-prev"></div>

</div>

</div>

<script>

    var swiper = new Swiper(".mySwiper", {
        spaceBetween: 30,
        loop: true,
        navigation: {
            nextEl: ".swiper-button-next",
            prevEl: ".swiper-button-prev",
        },
        pagination: {
            el: ".swiper-pagination",
        },
        mousewheel: true,
        keyboard: true,
    });

</script>

```

Опис коду

Опис структури HTML:

Контейнер слайдера:

`<div class="swiper mySwiper">`: Головний контейнер для слайдера, який містить всі слайди та елементи управління.

Обгортка слайдів:

`<div class="swiper-wrapper">`: Контейнер, що містить всі слайди.

Окремі слайди:

`<div class="swiper-slide">`: Кожен окремий слайд. В даному випадку, всередині кожного слайда є посилання (`<a>`) на іншу сторінку та зображення (``).

Елементи управління:

`<div class="swiper-pagination"></div>`: Контейнер для пагінації (точок, які показують кількість слайдів та поточний слайд).

`<div class="swiper-button-next"></div>`: Кнопка для переходу до наступного слайда.

`<div class="swiper-button-prev"></div>`: Кнопка для переходу до попереднього слайда.

Опис налаштувань JavaScript:

Ініціалізація Swiper:

`var swiper = new Swiper(".mySwiper", { ... });`: Створюється новий об'єкт Swiper, який ініціалізує слайдер для елемента з класом `.mySwiper`.

Параметри слайдера:

`spaceBetween: 30`: Встановлює відстань у 30 пікселів між слайдами.

`loop: true`: Вмикає циклічний режим, де після останнього слайда відображається перший.

Навігація:

`navigation: { nextEl: ".swiper-button-next", prevEl: ".swiper-button-prev" }`:

Встановлює елементи для кнопок навігації "вперед" і "назад".

Пагінація:

`pagination: { el: ".swiper-pagination" }`: Визначає елемент для пагінації.

Керування за допомогою миші та клавіатури:

`mousewheel: true:` Дозволяє перегортання слайдів за допомогою колеса миші.

`keyboard: true:` Дозволяє керувати слайдером за допомогою клавіатури (стрілки вліво та вправо).

Коли користувач хоче задати запитання він може заповнити форму яка є майже на всіх сторінках та форма відправить данні в базу даних щоб на це питання відповіли , код наведено в лістингу 7.

Лістинг 7. Відправка питань з форми

```
<?php
// Включение отображения ошибок для отладки
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
// Установка заголовка для JSON
header('Content-Type: application/json');
// Подключение к базе данных
$dbc = mysqli_connect('localhost', 'root', '12345',
'quest');
// Проверка подключения к базе данных
if (!$dbc) {
    http_response_code(500);
    echo json_encode(['message' => 'Database connection
failed']);
    exit;
}
// Получение данных из запроса
$data = json_decode(file_get_contents('php://input'),
true);
```

```

$name = $data['name'] ?? null;

$email = $data['email'] ?? null;

$comment = $data['comment'] ?? null;

// Проверка наличия данных
if (!$name || !$email || !$comment) {

    http_response_code(400);

    echo json_encode(['message' => 'Name, email, and
comment are required']);

    exit;

}

// Запрос на вставку данных
$query = "INSERT INTO quest (first_name, email, comment)
VALUES ('$name', '$email', '$comment)";

// Выполнение запроса и проверка результата
if (mysqli_query($dbc, $query)) {

    http_response_code(201);

    echo json_encode(['message' => 'Ваше питання
відправлено чекайте на відповідь']);

} else {

    http_response_code(500);

    echo json_encode(['message' => 'Error saving
feedback']);

}

// Закриття підключення к базі даних
mysqli_close($dbc);

?>

```

Опис коду

Відображення помилок

```
ini_set('display_errors', 1); та ini_set('display_startup_errors', 1);:
```

Включення відображення помилок для налагодження.

```
error_reporting(E_ALL);:
```

 Встановлення рівня звітності про всі помилки.

```
Заголовок для JSON
```

```
header('Content-Type: application/json');:
```

 Встановлення заголовка для відправки відповіді у форматі JSON.

```
Підключення до бази даних
```

```
$dbc = mysqli_connect('localhost', 'root', '12345', 'quest');:
```

 Підключення до бази даних.

```
Підключається до бази даних MySQL. Параметри:
```

```
localhost: Сервер бази даних.
```

```
root: Ім'я користувача для підключення.
```

```
12345: Пароль користувача.
```

```
quest: Назва бази даних.
```

```
Перевірка підключення до бази даних
```

```
if (!$dbc) { ... }:
```

 Перевірка підключення до бази даних. Якщо підключення не вдалося, повертається відповідь з кодом 500 і повідомленням про помилку.

```
http_response_code(500);:
```

 Встановлює код відповіді 500 (Internal Server Error).

```
echo json_encode(['message' => 'Database connection failed']);:
```

 Виводить повідомлення про помилку у форматі JSON.

```
exit;:
```

 Зупиняє виконання скрипта.

```
Отримання та перевірка даних
```

```
$data = json_decode(file_get_contents('php://input'), true);:
```

 Отримання даних з запиту.

```
$name = $data['name'] ?? null;, $email = $data['email'] ?? null;, $comment = $data['comment'] ?? null;:
```

 Витягнення даних з запиту.

`if (!$name || !$email || !$comment) { ... }`: Перевірка наявності всіх необхідних даних. Якщо якийсь поле не заповнене, повертається відповідь з кодом 400 і повідомленням про помилку.

Перевірка наявності даних:

`if (!$name || !$email || !$comment) { ... }`: Перевіряє, чи всі необхідні дані присутні. Якщо будь-яке з полів порожнє або відсутнє, виконується наступний блок.

`http_response_code(400);`: Встановлює код відповіді 400 (Bad Request).

`echo json_encode(['message' => 'Name, email, and comment are required']);`:

Виводить повідомлення про помилку у форматі JSON.

`exit;`: Зупиняє виконання скрипта.

Вставка даних

`$query = "INSERT INTO quest (first_name, email, comment) VALUES ('$name', '$email', '$comment')";`: Формування SQL запити для вставки даних.

Виконання запити та перевірка результату:

`if (mysqli_query($dbc, $query)) { ... }`: Виконання SQL запити і перевіряє, чи був він успішним..

Якщо запит виконався успішно:

`http_response_code(201);`: Встановлює код відповіді 201 (Created).

`echo json_encode(['message' => 'Ваше питання відправлено чекайте на відповідь']);`: Виводить повідомлення про успішне збереження у форматі JSON.

Якщо запит не виконався:

`http_response_code(500);`: Встановлює код відповіді 500 (Internal Server Error).

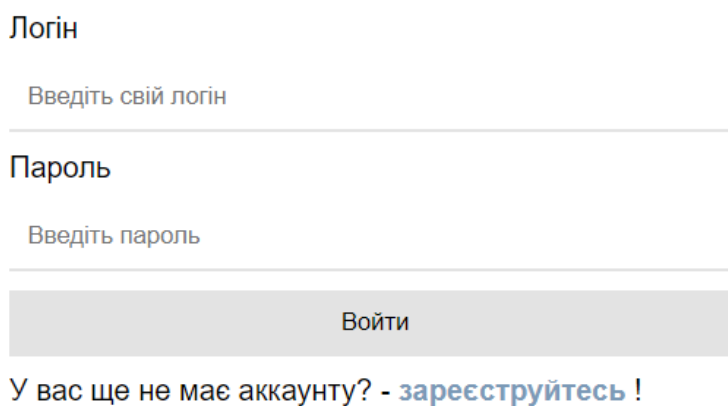
`echo json_encode(['message' => 'Error saving feedback']);`: Виводить повідомлення про помилку у форматі JSON.

Закриття підключення

`mysqli_close($dbc);`: Закриття підключення до бази даних після завершення операцій.

3.4 Розробка інтерфейсу інформаційного веб-ресурсу

Для того щоб адміністратор та психолог могли зайти та подивитись на базу даних були розроблено декілька сторінок , на рисунок 3.3 показано Аутентифікацію та вхід на яких адміністратор або психолог можуть ввести свій логін та пароль та вони зможуть перейти у свій профіль.



Логін

Введіть свій логін

Пароль

Введіть пароль

Войти

У вас ще не має акаунту? - [зареєструйтесь!](#)

Рисунок 3.3 — Сторінка входу

Якщо заходить новий співробітник то треба натиснути кнопку реєструватись та сайт перейде на іншу сторінку реєстрації, на рисунку 3.4 показана сторінка реєстрації.

ПІБ

Введіть своє повне ім'я

Логін

Введіть свій логін

Пошта

Введіть адресу своєї пошти

Зображення профілю

Файл не выбран

Пароль

Введіть пароль

Підтвердження пароля

Підтвердіть пароль

У вас вже є аккаунт? - [авторизуйтесь!](#)

Рисунок 3.4 — Сторінка реєстрації.

Коли адміністратор чи психолог заїде у свій профіль він побачить свій профіль та дві кнопки щоб перейти на бази даних котрі їх цікавлять, на рисунку 3.5 зображено профіль користувача.



вахновская анна андреевна

anny2013zp@gmail.com

Івенти

Питиння

Выход

Рисунок 3.5 — Профіль користувача

Кликнувши на кнопку питання відкриється таблиця з баз даних в якій буде відображатися ID,Email та коментар на якому можна подивитись та потім відповісти на пошту що людина не зрозуміла чи з чим зітнулась,на рисунку 3.6 зображено таблицю з питаннями

ID	Email	Коментар
3	шма	павтпаєрвбона
5	шма	павтпаєрвбона
6	шма	павтпаєрвбона
7	шма	павтпаєрвбона
8	шма	павтпаєрвбона
9	шма	павтпаєрвбона
10	шма	павтпаєрвбона
11	шма	павтпаєрвбона
12	шма	павтпаєрвбона
13	шма	павтпаєрвбона
14	шма	павтпаєрвбона

Рисунок 3.6 — Сторінка з таблицею

Це одна зі сторінок на яких представлена інформація про один з онлайн ризиків, на картинках 3.7 та 3.8 показана одна зі сторінок. У заголовку на кожній сторінці написано про який ризик та ілюстрація яка допомагає створити візуальний зв'язок з темою і робить сторінку більш привабливою для користувачів.

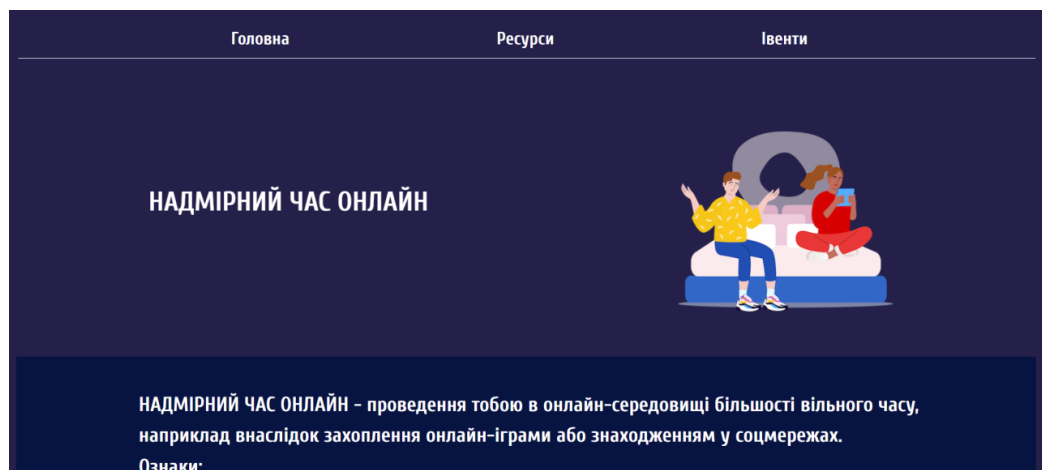


Рисунок 3.7 — Одна із сторінок про онлайн ризики

- Пропуски занять через комп'ютерні ігри та час в онлайні;
- Проведення часу за гаджетами вночі;
- Відсутність інших захоплень, окрім гаджетів;
- Нівелювання спілкуванням з іншими людьми;
- Інтернет набуває для тебе надцінного значення;
- Агресивне реагування на намагання оточуючих обмежити час в онлайні;
- Прояви агресивності, дратівливості із-за неможливості скористатись інтернетом;
- Втрата інтересу до соціального життя;
- Виправдання власної поведінки і пристрасті;
- Часті безпричинні зміни настрою;
- Хвороблива і неадекватна реакція на критику, зауваження, поради;
- Наростаюча опозиційність до батьків, родичів, старих друзів;
- Перенесення спілкування у кіберпростір.

Рисунок 3.8 — Одна із сторінок про онлайн ризики

Форма, яка розміщена внизу кожної сторінки веб-сайту. Вона призначена для того, щоб користувачі могли задавати питання, які виникли після прочитання матеріалів. Питання можуть бути відправлені фахівцям (психологам), які нададуть відповідь, на рисунку 3.9 зображена форма відправки.

Якщо в тебе залишилися питання ти можеш задати їх фахівцю

Задати питання

Ім'я*

Email*

Питання*

[Відправити](#)

© Вахнівська Анна 2023

Рисунок 3.9 — Форма відправки

Коли користувач тільки переходить по посиланню то йому відкривається головна сторінка на якій він побачить слайдер зі всіма сторінками з інформацією про онлайн ризики, на рисунках 3.10, 3.11 та 3.12 зображено головна сторінка веб-ресурсу.

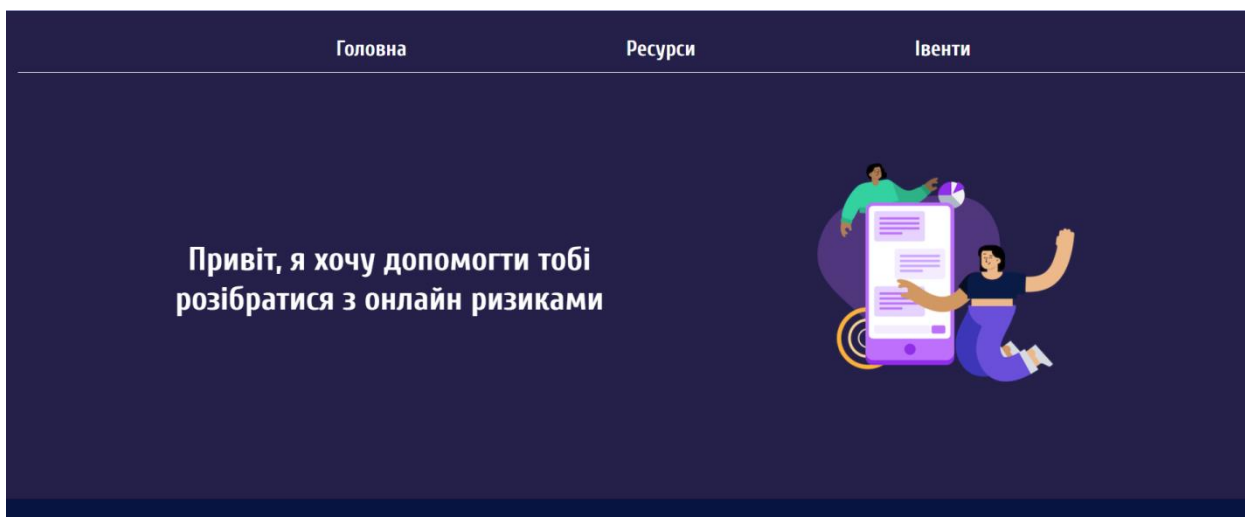


Рисунок 3.10 — Головна сторінка

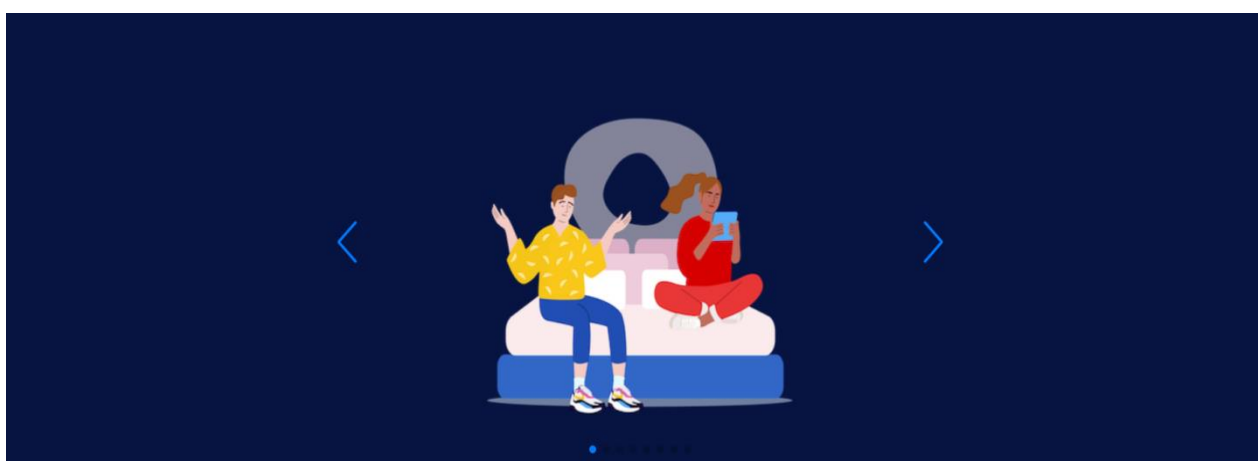


Рисунок 3.11 — Головна сторінка

Якщо в тебе залишилися питання ти можеш задати їх фахівцю

Задати питання

© Вахнівська Анна 2023

Рисунок 3.12 — Головна сторінка

Якщо користувач перейде на сторінку ресурси то він побачить список ресурсів на які він може перейти та дізнатись інформацію чи поспілкуватись у чат боті с іншими психологами, на рисунках 3.13 та 3.14 зображено сторінку ресурси.



Рисунок 3.13 — Сторінка ресурси

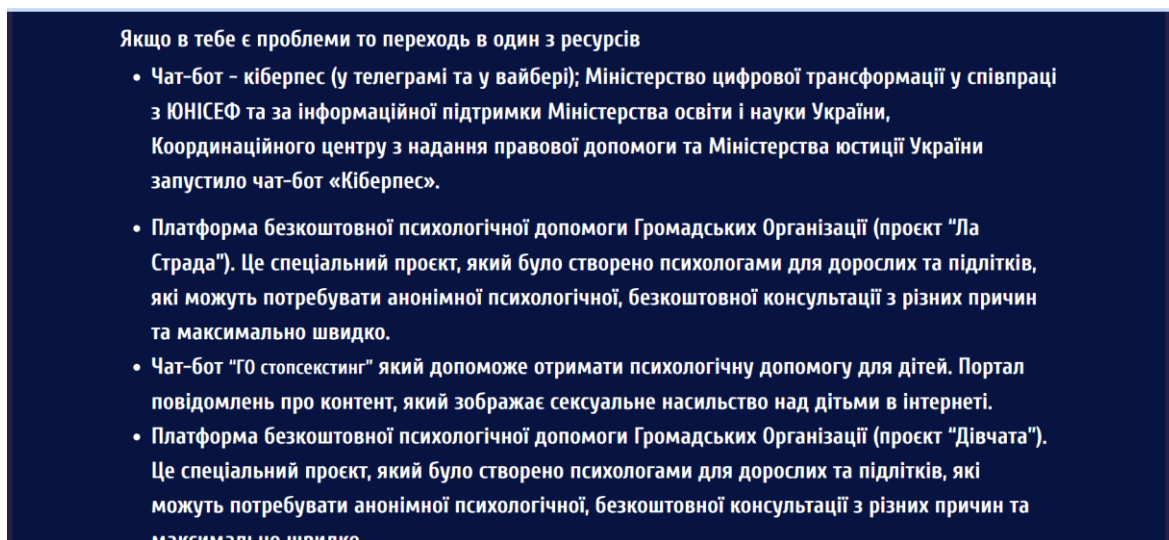


Рисунок 3.14 — Сторінка ресурси

Якщо користувач перейде на вкладнику івенти там він побачить івент опис його та зможе реєструватись якщо захоче прийти, на рисунках 3.15 та 3.16 зображено сторінку івенти.



Рисунок 3.15 — Сторінка івенти

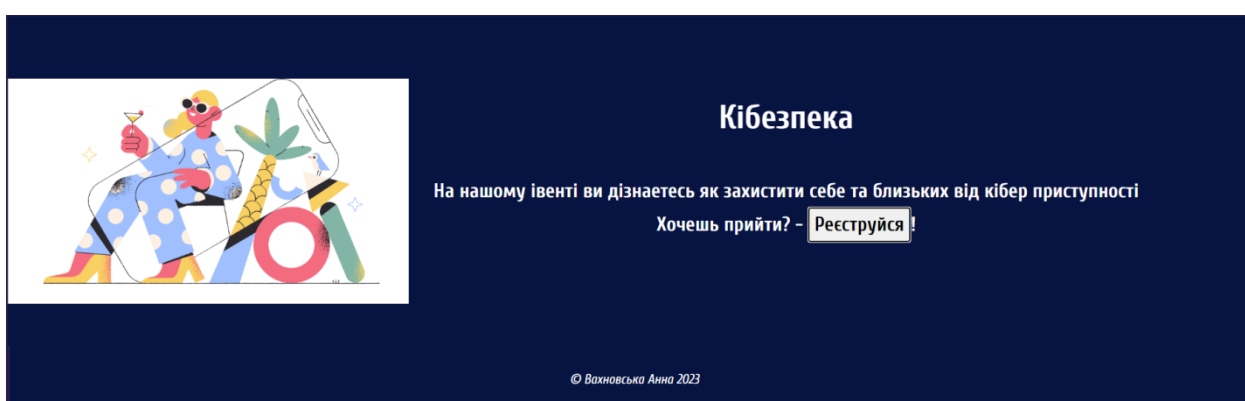


Рисунок 3.16 — Сторінка івенти

Після того як користувач натисне кнопку Реєструватися інформація додається до бази даних та адміністратор це побачить, рисунках 3.17 зображено реєстрацію на івент.

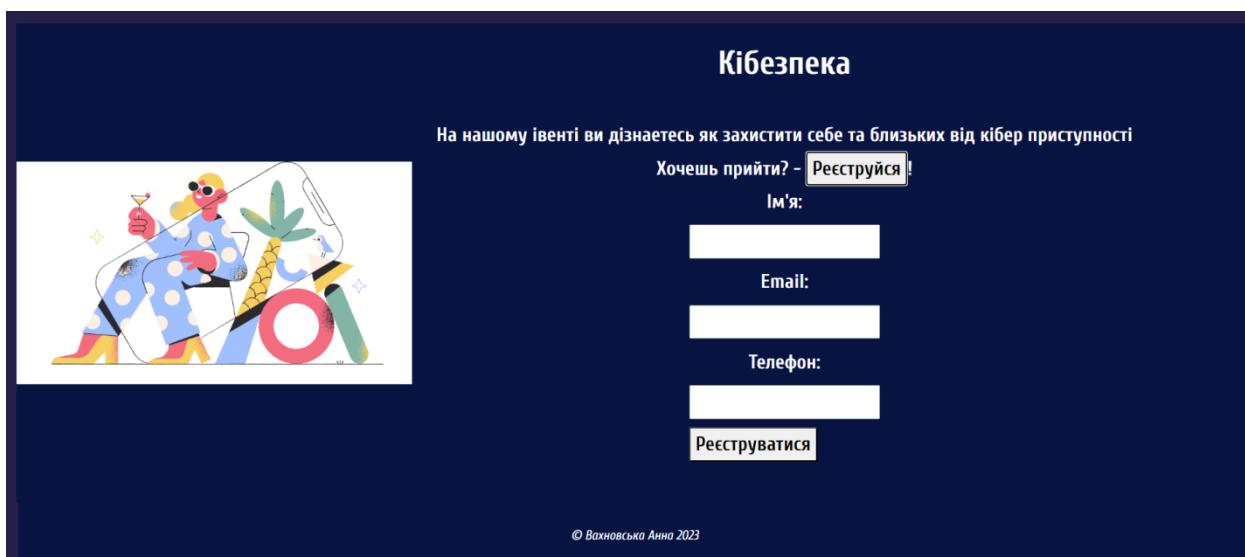


Рисунок 3.17 — Сторінка івенти

На цьому рисунку 3.18 зображено що бачить адміністратор, в адмін панелі можна перейти на івенти та побачити таблицю з бази даних в якій зазначено ім'я емайл та номер телефону.

ID	name	Email	телефон
1	анна	anny2013zp@gmail.com	
2	анна	anny2013zp@gmail.com	
3	анна	Iphone4love124@gmail.com	
4	Аня	anny2013zp@gmail.com	0997870299
5	Аня	anny2013zp@gmail.com	0997870299
6	Аня	anny2013zp@gmail.com	0997870298

Рисунок 3.18 — Таблиця в адмін панелі

ВИСНОВКИ

1. У ході реалізації кваліфікаційної роботи було проведено огляд існуючих рішень (аналогів) продукту, які стосуються онлайн ризиків серед підлітків, проаналізовано функціональні можливості та інтерфейси аналогів, виявлено актуальні потреби та проблеми у функціоналі.
2. На основі аналізу визначено мету розробки власного веб-ресурсу, який буде набагато зручніший для користувачів. Очікується, що розробка веб-ресурсу допоможе знизити онлайн ризики серед підлітків.
3. Розроблено технічне завдання, яке чітко описує функціональні вимоги, вимоги до інтерфейсів, вимоги до системи та план робіт. Спроектовано архітектуру веб-ресурсу, на якій зображено всі компоненти та їх взаємодію.
4. Розроблено базу даних, яка буде зберігати всі питання реєстрації на івенти та реєстрації користувачів.
5. Створено Інтернет-ресурс, який допомагає користувачам дізнатися нову інформацію для себе, задати питання та зареєструватися на івент.
6. Існує можливість розширення інформативних сторінок в майбутньому. Створення цікавих інтерактивів на веб-ресурсі для підлітків, аби їм було цікавіше дізнаватися нову інформацію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Orben, A., & Blakemore, S.–J. (2023). How social media affects teen mental health: A missing link. <https://doi.org/10.1038/d41586-023-00402-9> (дата звернення: 10.02.2024).
2. © 2024 American Psychological Association
750 First St. NE, Washington, DC 20002-4242
<https://www.apa.org/topics/social-media-internet/social-media-parent-tips> (дата звернення: 10.02.2024).
3. Stop-sexting <https://stop-sexting.in.ua/> Громадська організація за підтримки Київстар (дата звернення: 1.04.2024).
4. Intrland https://beinternetawesome.withgoogle.com/uk_ua/interland
Google (дата звернення: 1.04.2024).
5. День Безпечного Інтернету <https://betterinternetcentre.org> (дата звернення: 1.04.2024).
6. Дія Освіта <https://osvita.diaa.gov.ua/courses/serial-dlya-batkiv-onlayn-bezpeka-ditey> (дата звернення: 1.04.2024). Дія
7. Міністерство освіти і науки України
<https://mon.gov.ua/ua/osvita/pozashkilna-osvita/vihovna-robota-ta-zahist-prav-ditini/bezpeka-ditej-v-interneti> (дата звернення: 1.04.2024). kitsoft
8. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. посіб. Харків : ХНУМГ ім. О. М. Бекетова, 2017. 93 с. (дата звернення: 10.04.2024).
9. Richards M. Software Architecture Patterns : USA: O'Reilly Media, 2015. 350 р. (дата звернення: 15.04.2024).
10. . Мельник Р.А. Програмування веб-застосувань (фронт-енд та бек-енд): навч. посіб. Львів : Видавництво Львівська політехніка, 2018. 248 с. (дата звернення: 20.04.2024).

11. . Tweedie R. Learning FuelPHP for Effective PHP Development. Livery : Packt Publishing Ltd, 2013.104p. (дата звернення: 2.05.2024).
12. . Drouyer S. FuelPHP Application Development Blueprints. Livery : Packt Publishing Ltd, 2015. 398p. (дата звернення: 6.05.2024).
13. Ніксон Р. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML5, 2016. 510 p. (дата звернення: 10.05.2024).
14. Zandstra M. PHP Objects, Patterns, and Practice : Apress: 5th ed. 2016. 603 p. (дата звернення: 14.05.2024).
15. Єгорова І.М. Проектування та розробка Web-документів: навч. посібник. Харків: ХНУРЕ, 2018. 264 с. Feb 15, 10:17 (дата звернення: 15.05.2024).
16. Конференція молода наука https://sites.znu.edu.ua/stud-sci-soc//2009/tom_5_2024.pdf (дата звернення: 14.06.2024)

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Вахновська Анна Андріївна, студентка 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти annu2013zp@gmail.com, — підтверджую, що написана мною кваліфікаційна робота на тему «**Розробка інформаційного веб-ресурсу с використанням мови програмування PHP.**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 25.05.2024 Підпис _____ Вахновська Анна Андріївна
(студент)

Дата 25.05.2024 Підпис _____ Михайлуца Олена Миколаївна
(науковий керівник)