Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавра

зі спеціальності 121 – Інженерія програмного забезпечення

На тему: Клієнт-серверний веб-застосунок для формування комплексного харчування при здійсненні туристичних походів на певний період часу

Засвідчую, що в цій кваліфікаційній роботі немає запозичень із праць інших авторів без відповідних посилань.

Студент гр. ІПЗ-21-СК

_____ /В.М. Скірчак/

Керівник
кваліфікаційної роботи _____ / А. М. Стрюк /

Завідувач кафедри _____ / А. М. Стрюк /

Кривий Ріг

2024

Криворізький національний університет

Факультет: Інформаційних технологій

Кафедра: Моделювання та програмного забезпечення

Ступінь вищої освіти: бакалавр

Спеціальність: 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри

_____ А. М. Стрюк

«___» _____ 20__ р

ЗАВДАННЯ

на кваліфікаційну роботу

студенту групи ІПЗ-21-СК Скірчаку Владиславу Миколайовичу

1. Тема: Клієнт-серверний веб-застосунок для формування комплексного харчування при здійсненні туристичних походів на певний період часу затверджено наказом по КНУ № 22 від «__» _____ 2024 р
2. Термін подання студентом закінченої роботи: «10» червня 2024 р.
3. Вихідні дані по роботі: автоматизувати процес розрахунків кількості продуктів для подорожі, для зменшення вірогідності впливу людського фактору на процес розрахунків.
4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): виконати аналіз предметної області та існуючих аналогів, сформувати список функціональних вимог до застосунку, розробити структуру баз даних, спроектувати архітектуру застосунку, здійснити програмну реалізацію розробленого застосунку.
5. Перелік ілюстративного матеріалу: алгоритми використання, блок-схеми розроблених алгоритмів, діаграма бази даних, знімки екранних форм.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	Огляд літератури за тематикою та збір даних	
2	Проведення аналізу предметної області та існуючих аналогів	
3	Проектування функціональної частини застосунку	
4	Проектування архітектури застосунку, його основних компонентів, вибір інструментів	
5	Підготовка матеріалів першого розділу роботи	
6	Підготовка матеріалів другого розділу роботи	
7	Розроблення програмного забезпечення для формування комплексного харчування	
9	Підготовка матеріалів третього розділу роботи	
10	Оформлення пояснювальної записки	

Дата видачі завдання: «__» _____ 2024 р.

Студент _____ / В. М. Скірчак /

Керівник роботи _____ / А. М. Стрюк/

РЕФЕРАТ

ТУРИЗМ, АНАЛІЗ ВИМОГ, МЕТОДИ ДОСЛІДЖЕННЯ АКТУАЛЬНОСТІ ЗАСТОСУНКУ, РОЗРОБКА ТА ТЕСТУВАННЯ ПРОТОТИПУ

Пояснювальна записка: 89 с., 10 табл., 18 рис., 0 дод., 9 джерел.

Туризм, як вид активного відпочинку та дозвілля, щорічно залучає мільйони людей з усього світу. Відкриття нових країн, знайомство з різними культурами, мальовничі краєвиди – все це робить подорожі незабутніми. Проте, планування подорожі, особливо тривалої, може бути пов'язане з низкою труднощів, однією з яких є організація харчування. Складання раціону, враховуючи тривалість подорожі, маршрут, тип активності, харчові переваги та бюджет, може стати справжнім викликом.

Саме тому розробка клієнт-серверного web-застосунку, який автоматизує процес планування харчування в туристичних подорожах, є актуальною та має значну практичну цінність. Такий застосунок може значно полегшити життя мандрівників, позбавивши їх від необхідності самостійно підбирати їжу та розраховувати необхідну кількість продуктів.

Метою даної дипломної роботи є розробка web-застосунку для планування харчування в туристичних подорожах, який буде максимально зручним та функціональним для користувачів. Для досягнення поставленої мети необхідно виконати наступні завдання: проаналізувати існуючі аналоги, визначити потреби мандрівників, розробити архітектуру та дизайн, реалізувати застосунок та провести тестування.

Для досягнення поставлених завдань у дипломній роботі були використані такі методи дослідження: аналіз літератури, опитування користувачів, прототипування застосунку та його тестування.

ABSTRACT

TOURISM, REQUIREMENTS ANALYSIS, APPLICATION RELEVANCE RESEARCH METHODS, PROTOTYPE DEVELOPMENT AND TESTING

Explnatory note: 89p, 10 tab., 18 fig., 0 app., 9 references

Tourism, as a form of active recreation and leisure, attracts millions of people from all over the world every year. Discovering new countries, getting to know different cultures, and seeing picturesque landscapes makes traveling unforgettable. However, planning a trip, especially a long one, can be fraught with a number of difficulties, one of which is organizing meals. Creating a meal plan that takes into account the duration of the trip, route, type of activity, food preferences, and budget can be a real challenge.

That is why the development of a client-server web application that automates the process of meal planning for tourist trips is relevant and has significant practical value. Such an application can greatly facilitate the life of travelers by eliminating the need to select food and calculate the required amount of food on their own.

The purpose of this thesis is to develop a web application for meal planning in tourist travel, which will be as convenient and functional as possible for users. To achieve this goal, it is necessary to perform the following tasks: analyze existing analogues, determine the needs of travelers, develop architecture and design, implement the application and conduct testing.

To achieve these objectives, the thesis used research methods such as literature analysis, user surveys, application prototyping and testing.

ЗМІСТ

ВСТУП

Актуальність дослідження використання мікросервісної архітектури на прикладі розробки програмного забезпечення логістичної компанії полягає в тому, що в сучасному світі логістичні компанії мають велику кількість даних та ресурсів, які необхідно ефективно управляти для забезпечення оптимальної роботи та задоволення потреб клієнтів. Використання мікросервісної архітектури може допомогти компанії забезпечити більшу гнучкість, масштабованість та швидкість розробки програмного забезпечення.

Метою роботи є аналіз сучасного стану речей у області проектування програмного забезпечення в умовах змінних вимог від ринку. Ставиться задача розглянути мікросервісну архітектуру у контексті порівняння з монолітною. Також для закріплення результатів дослідження створено програмне забезпечення логістичної компанії.

Результати виконаної роботи можуть використовуватись як поштовх для подальшого, більше глибокого вивчення теми мікросервісної архітектури, її особливостей та тонкощів застосування. Також результати роботи включають в себе аналіз вимог до програмного забезпечення логістичної компанії що може бути корисно для проектування подібної системи поза межами навчальних цілей.

1 СУЧАСНИЙ СТАН І АКТУАЛЬНІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1.1 Аналіз професійної області проблеми

В представленій кваліфікаційній роботі головною проблематикою виступає автоматизація розрахунків ваги і калорійності, та процес створення харчової розкладки.

Щоб більш глибоко розібратися в цій проблемі ми проаналізували процес створення харчової розкладки для звичайної групи туристів, що збираються у групову подорож.

Зазвичай, організація групової подорожі, як і створення для цієї подорожі харчової розкладки, робота однієї відповідальної за це людини і як не дивно, незважаючи на всю розповсюдженість такої сфери як туризм, розрахунки їжі, що необхідно взяти в подорож, в повсякденному житті, робляться за допомогою звичайного калькулятора та блокнота.

Для створення харчової розкладки, лист блокнота умовно розподіляється на блоки по дням, які в свою чергу мають розбиття на сніданок, обід та вечерю. В кожну комірку записуються продукти, що буде використано на цьому прийомі їжі, записується їх назва, вага, та приблизна калорійність (це важливо, якщо людина ще не достатньо обізнана в створені меню на дні подорожі, та погано орієнтується в калорійності тих чи інших продуктів). Після створення розкладки, необхідно підрахувати загальну вагу кожного окремого інгредієнта на всю подорож, та перемножити її на загальну кількість людей, для яких створюється розкладка, адже після розрахунків необхідно закупити указані продукти для подальшого розподілення по дням. Список продуктів також зберігається в блокноті або на будь-якому іншому носії даних окремо від створюваної розкладки.

Таким чином, ми визначили розрахунки, що необхідно автоматизувати: вагу кожного продукту на всю подорож з урахуванням кількості учасників, калорійність та вагу кожного окремого меню та харчової розкладки в цілому. Також, для зручності користувача виправдане створення списку продуктів із

заздалегідь визначеними параметрами калорійності та ваги стандартної порції, які можна використати для створення харчової розкладки.

Також уважно вивчивши предметну область ми визначили яка вхідна та вихідна інформація повинна оброблятись та зберігатись на серверній частині проекту.

Інформація про вхідні дані:

- інформація про керівників подорожей;
- інформація про походи;
- інформація про харчові розкладки;
- інформація про продукти.

На основі поставленого мною завдання вихідною інформацією буде:

- сумарна вага продуктів на день/подорож;
- сумарна калорійність продуктів на день/подорож;
- сумарна вага кожного окремого продукту на всю подорож для подальшої закупівлі;
- список подорожей відповідно до обраних фільтрів;
- список продуктів відповідно до обраних фільтрів;
- список керівників та їх подорожей відповідно до обраних фільтрів;

Кожна розкладка зберігає в собі дані про продукти, що було додано до неї, з урахуванням ваги порції, дня та типу прийому їжі. Сумарна вага та калорійність вираховується виходячи з ваги та калорійності продуктів похода помноженої на кількість учасників подорожі. Калорійність стандартної порції продукту (ніде не зберігається, але виводиться для користувача біля ваги кожного продукту в списку) вираховується із калорійності продукту на 100

грам, та ваги стандартної порції.

Отже, виходячи з вищесказаного, ми можемо підвести підсумок, що ідеальний варіант створення клієнт-серверного веб-застосунку для створення розкладок харчування в подорож є написання web-вузла для клієнтської частини, з динамічною системою керування каталогами та збереженням інформації в базі даних для серверної частини проєкта.

1.2 Аналіз існуючих рішень

Для реалізації задачі автоматизації розрахунків та створення харчової розкладки в подорож на ринку програмно-апаратних комплексів існує лише декілька рішень, що хоча б в деякій мірі задовольняють вищезазначеним умовам.

Кожний з даних програмних продуктів має певний перелік функцій, але кожен з них має ряд значних недоліків, що не дозволяють в повній мірі використати їх для поставленої нами задачі.

Більшість з таких рішень виглядає не набагато складніше, аніж звичайний розлінований лист блокнота. Реалізація в сукупності своїй зводиться до таблиці зі списком продуктів, інтерфейс таких програм складно одразу зрозуміти, а для користування необхідно мати певні навички роботи з формулами.

Другою не менш важливою проблемою даних продуктів є необхідність робити власноруч деякі з розрахунків, наприклад підраховувати загальну кількість того чи іншого продукту для подальшої закупівлі, або перераховувати зазначену вагу на кількість учасників групової подорожі, що в свою чергу може з високою вірогідністю призвести до похибки через значних вплив людського фактору при здійснені розрахунків. В свою чергу, це може призвести до нестачі певних продуктів, що для автономної подорожі може стати критичним фактором який неодмінно вплине на враження від запланованої мандрівки. Так само, хоча і не настільки критично на вражені від мандрівки може вплинути більша порівняно із запланованою кількість продуктів, адже вага рюкзака неодмінно збільшиться, що може призвести до болі у деяких місцях тіла через перенавантаження, в першу чергу це стосується плечей та спини, що у більшості своїй напружуються під час перенесення такої значної ваги.

Наостанок, більшість зі знайдених програмних рішень розраховані

виключно на людей у значній мірі досвідчених у складанні харчових розкладок. Майже в усіх не має ані зазначеної калорійності продуктів, ані готового списку продуктів зі стандартною вагою та калорійністю, ані прикладів створених розкладок більш досвідчених керівників, що не дозволяє людині без принаймні якогось досвіду долучитися до створення власних подорожей.

Отже, нами був проведений пошук програмних рішень поставленої проблеми подібного призначення. В процесі аналізу існуючих аналогічних рішень, нами було знайдено лише один аналог, що відповідає поданій темі та заданим критеріям:

Програма для розрахунку харчування у багатоденному поході RACION представлена на рисунку 1.1.

		МЕНЮ	*Хибины*
			1995 г.
	ВАРИАНТ 1	**ВАРИАНТ 2**	**ВАРИАНТ 3**
	Завтрак	***Завтрак***	***Завтрак***
	Каша рисовая с изюмом	Каша пшенная с изюмом	Каша овсяная с изюмом
	Сухарь	Сухарь	Сухарь
	Сыр	Колбаса	Колбаса
	Чай с сахаром (кофе)	Чай с сахаром (кофе)	Чай с сахаром (кофе)
	Обед	**Обед**	**Обед**
	Суп	Бульон с вермишелью	Суп
	Сухарь	Сухарь	Сухарь
	Колбаса	Сыр	Сало
	Халва	Щербет	Калорийная колбаса
	Чай с сахаром	Чай с сахаром	Чай с сахаром
	Перекус	**Перекус**	**Перекус**
	Карамель	Карамель	Карамель
	Сухофрукты	Сухофрукты	Сухофрукты
	Ужин	**Ужин**	**Ужин**
	Вермишель с тушенкой	Рис с тушенкой	Гречка с тушенкой
	Сухарь	Сухарь	Сухарь
	Мармелад	Конфеты	Печенье
	Чай с сахаром	Чай с сахаром	Чай с сахаром

Рисунок 1.1 – Робота програми RACION

"Racion" – програма для планування меню для багатоденної автономної подорожі надає можливість розрахувати кількість продуктів харчування, необхідних групі в поході тривалістю до 26 днів з урахуванням складеного меню та норм харчування. Якщо маршрут розбитий на "кільця", між якими організовуються закидання, "Racion" дасть змогу розрахувати кількість продуктів

на кожне з них (до трьох "кілець"). Крім цього, "Racion" розраховує середню вагу добового раціону та середнє навантаження на учасника походу для кожного кільця та маршруту загалом.

Головними недоліками даної програми є:

– відсутність інтуїтивно зрозумілого інтерфейсу;

– відсутність зазначеної калорійності продуктів;

– відсутність початкового списку продуктів з яких можна створити свою харчову розкладку;

– відсутність прикладу створення харчових розкладок попередніми користувачами та керівниками подорожей;

– можливості планування харчової розкладки обмеженні 26 днями та лише трьома варіантами меню.

Отже, ми виявили головні недоліки програмних рішень схожого типу, і на основі проведеного аналізу було прийнято рішення створити свій програмний комплекс для вирішення поставленої задачі.

1.3 Формулювання актуальності і завдань роботи

Для автоматизації створення харчових розкладок у подорожі та подальшої реалізації автоматизованої клієнт-серверної системи нами була поставлена мета спроектувати веб-сайт для більш зручного та інтуїтивно-зрозумілого користування розробленими алгоритмами.

З метою реалізації даного завдання, ми розробили загальні вимоги та перелік основних функцій, котрі повинна виконувати панель керування проекту:

Щоб кожен міг без проблем зайти та скористатися веб-ресурсом, потрібно грамотне його налаштування– цим займається адміністратор сайту. До його обов'язків входить підтримка сайту в робочому стані і саме адміністратор відповідає за його безпеку.

Функції адміністратора сайту, які він виконує, можна поділити на три групи:

Модерація – адміністратор на громадських мережевих ресурсах має розширені права, завдяки чому він може спілкуватися з користувачами через форму зворотного зв'язку і стежити за тим, щоб не порушувалися правила сервісу

або інтернет-ресурсу. Якщо таке станеться, адміністратор може винести попередження користувачеві, заблокувати або видалити його повідомлення;

Технічна підтримка – це всі моменти, пов'язані з технічним обслуговуванням. Адміністратор працює з доменами та хостингом і відповідає за мережеву безпеку, а також займається просуванням сайту.

Інформаційна підтримка – все, що стосується змісту та контенту: наповнення, текст, графіка та ін. Адміністратор постійно взаємодіє з іншими фахівцями: контент-менеджерами, дизайнерами, розробниками та верстальниками. Кожен робить свій внесок у створення «живого» веб-ресурсу.

Отже, роль адміністратора дуже важлива для підтримки працездатності веб-ресурсу. Через це нами було вирішено розробити адміністративний програмний інтерфейс, котрий має виконувати наступні функції:

– надавати можливість додавання видалення та редагування даних про подорожі та меню харчування;

– надавати можливість обліку даних щодо закупки необхідних продуктів.

– надавати можливість менеджменту даних про походи;

Також було прийнято рішення розробити серверну частину, що має виконувати наступні функції:

– зберігати та надавати інтерактивний доступ до даних про створені подорожі;

– зберігати та надавати інтерактивний доступ до користувацьких даних.

– проводити розрахунки ваги як всього меню, так і окремих його частин з урахуванням кількості учасників.

– проводити розрахунки калорійності продуктів та окремих частин меню з урахуванням кількості учасників.

Наостанок, необхідно розробити клієнтський програмний інтерфейс, котрий має виконувати наступні функції:

– надавати можливість ідентифікації та реєстрації користувачів;

– надавати можливість перегляду всіх публічних подорожей та їх меню харчування;

– надавати можливість перегляду даних про меню як у текстовому вигляді, так і у вигляді списків.

Додатково веб-застосунок повинен володіти наступним функціональними можливостями:

– перевіряти вхідні дані на коректність та відповідність умовам;
– мати зручний та зрозумілий інтерфейс;
– підтримувати стабільне з'єднання з джерелом даних.

2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ, АЛГОРИТМІВ, ФУНКЦІОНАЛЬНИХ СХЕМ, СХЕМ БАЗ ДАНИХ

2.1 Алгоритм вирішення прикладної задачі.

Поняття «Алгоритм» - концептуальна основа різноманітних процесів обробки інформації. Саме наявність відповідних алгоритмів і забезпечує можливість автоматизації. Алгоритм визначає послідовність дій, що забезпечує одержання необхідного результату з вихідних даних.

Важливість моделювання алгоритму дій користувача розробки програмного забезпечення має кілька причин. По-перше, це дозволяє розробникам більш глибоко зрозуміти потреби та очікування користувачів, а також їхню взаємодію з програмним продуктом. Це може допомогти уникнути помилок і спростити процес розробки, створюючи більш задовільний досвід користувача.

По-друге, моделювання алгоритму дій користувача допомагає у проектуванні інтерфейсу користувача, визначаючи оптимальні способи взаємодії між користувачем та програмним продуктом. Це дозволяє створити більш інтуїтивні та зручні інтерфейси, покращуючи задоволеність користувачів та знижуючи можливість помилок.

По-третє, моделювання алгоритму дій користувача дозволяє проводити тестування програмного продукту на ранніх етапах розробки, що може виявити проблеми в взаємодії користувача і додати корективи до релізу. Це сприяє підвищенню якості програмного забезпечення та зниженню витрат на подальше доопрацювання та виправлення помилок.

Таким чином, моделювання алгоритму дій користувача є важливим аспектом розробки програмного забезпечення, який сприяє створенню більш задовільного досвіду користувача, оптимізації інтерфейсу користувача та підвищенню якості програмного продукту.

Для вирішення поставленої нами задачі нами було розроблено головний алгоритм створення харчової розкладки звичайним користувачем, що представлено на рисунку 2.1.

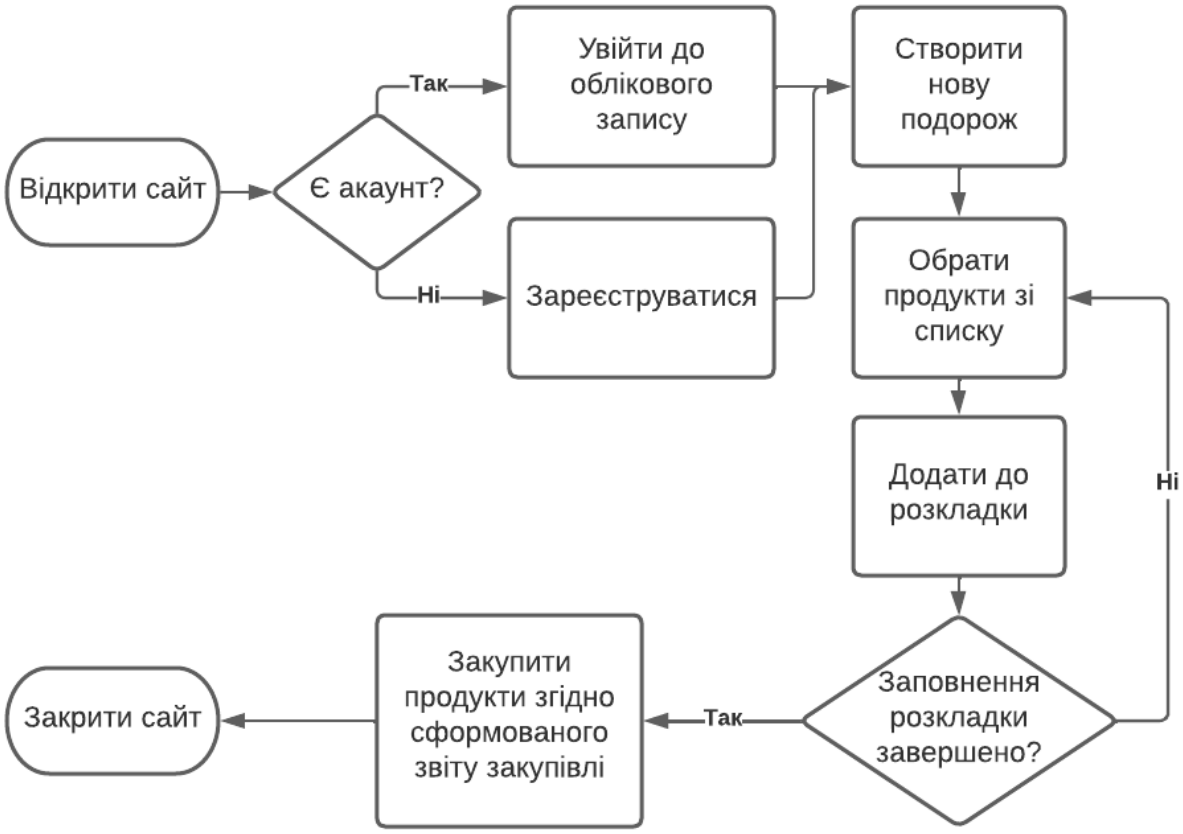

Рисунок 2.1 – Алгоритм створення харчової розкладки

На даному алгоритмі детально показано процес створення звичайним користувачем харчової розкладки. Першим етапом, очевидно, виступає процес відкриття сайту для подальшої роботи, після якого настає процес реєстрації або авторизації користувача. Після цього необхідно створити подорож для подальшого внесення продуктів харчування у майбутні меню. Після внесення в спеціальні поля назви, плану калорійності, кількості учасників та дат похода користувач може перейти до безпосереднього створення харчової розкладки. Даний процес супроводжується багаторазовим внесенням різнотипних продуктів зі списку до певних днів та типів прийому їжі. Після завершення роботи над заповненням харчової розкладки користувач повинен мати змогу подивитися звітні розрахунки для подальшої закупівлі продуктів в подорож.

Не менш важливим для ефективної роботи сайту э процес оновлення та додавання продуктів до базового списку, доступ до керування яким має бути лише у адміністратора для уникнення плутанини. Процес створення нового продукту показано на рисунку 2.2.

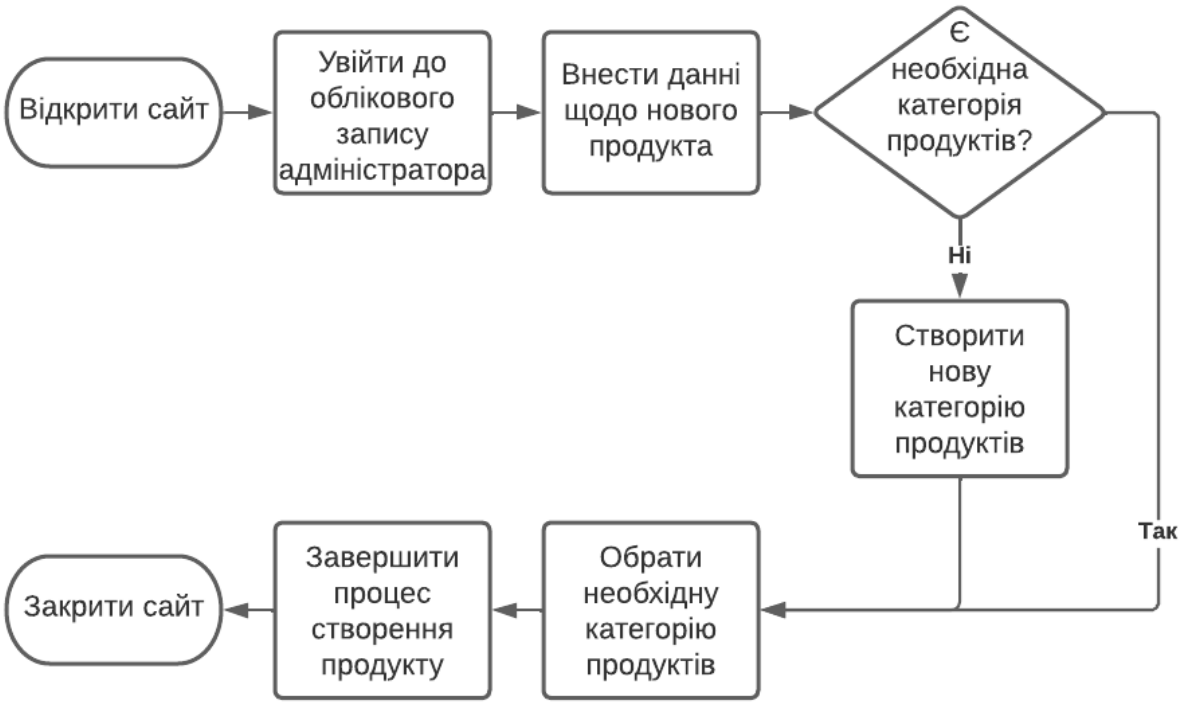

Рисунок 2.2 – Алгоритм створення нового продукту

На даному алгоритмі можна побачити процес створення нового продукту адміністратором сайту. Для цього необхідно увійти до облікового запису адміністратора та почати створення нового продукту: ввести дані щодо назви, калорійності та стандартної ваги порції, обрати категорію до якої належить продукт, за необхідності створити нову, що задовільняють потреби. На сам кінець завершити процес створення нового продукту та закрити сайт.

На відміну від авторизованих користувачів, незареєстрований користувач може лише переглянути існуючі та вже створенні іншими користувачами подорожі. Але даний процес також відіграє важливу роль в заохоченні та збагачені знаннями з досвіду досвідчених мандрівників щодо створення харчових розкладок. Процес ознайомлення незареєстрованого користувача зі створеними іншими користувачами харчовими розкладками зображено на рисунку 2.3.

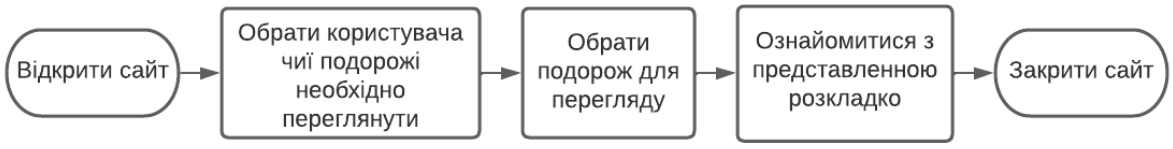

Рисунок 2.3 – Алгоритм ознайомлення незареєстрованого користувача з харчовими розкладками створеними іншими користувачами

Отже, нами було розглянуто та розроблено говні алгоритми дій

користувачів з різним рівнем доступу під час користування сайтом для створення харчових розкладок в подорож.

2.2 Проєктування та моделювання бази даних клієнт-серверного застосунку.

Клієнт-серверний застосунок для створення харчових розкладок в поход проектується з динамічною системою керування вмістом та збереженням інформації в базі даних, таким чином нам необхідно визначитися з даними, що будуть зберігатися в базі даних нашого проєкту.

Попередньо було детально проаналізовано міжпредметну область. В результаті проведеного аналізу було визначено, що:

Про кожну подорож відомі її назва, кількість учасників, дата завершення подорожі, дата її початку та план калорійності. Розкладка кожної подорожі передбачає певний набір продуктів. Про кожен продукт відомі його назва, калорійність на 100 грам, та вага стандартної порції. Якщо продукт використовується в будь-якій харчовій розкладці, слід зазначати кількість цього інгредієнта, день та тип меню до якого його було додано.

Кожен продукт належить рівно до однієї категорії (наприклад, фрукти, овочі, крупи, м’ясо, тощо), про яку відома її назва. Створює та редагує інформацію про подорожі користувач, про якого відомі його ім’я та рівень прав доступу (користувач або адміністратор), користувачі з рівнем доступу «адміністратор» також мають змогу редагувати загальний список продуктів. Назви продуктів, категорій продуктів, подорожей та імена користувачів вважайте унікальними.

Спираючись на опис міжпредметної області виділимо сутності та їх атрибути.

1. «Подорож» - містить інформацію про назву подорожі, її кількість учасників, план калорійності, продукти, що буде взято у подорож, користувача, що створив подорож, дату початку та завершення подорожі.
2. «Користувач» - містить інформацію про ім’я та рівень доступу користувача.

3. «Продукт» - містить інформацію про назву продукту, його калорійність на 100 г, масу стандартної порції в грамах, та до якої категорії продуктів він належить.
4. «Категорія продукту» - містить інформацію про назву категорії продукту.
5. «Розкладка» - містить інформацію про продукт, що буде взято в подорож, день на який його планується взяти, тип меню, кількість продукту в грамах та подорож, до якої відноситься цей продукт.

Розглянемо визначення описових атрибутів сутностей і ключів:

Опис атрибутів сутності «Користувачі» наведено в табл. 1, «ID_Користувача» є ключовим, тому що однозначно визначає екземпляр сутності.

Таблиця 2.1 - Атрибути сутності «Користувач»

Назва атрибуту	Опис атрибуту
ID_Користувача	Унікальний ідентифікатор користувача
Ім’я	Ім’я користувача
Рівень доступу	Рівень доступу користувача

Опис атрибутів сутності «Подорожі» наведено в табл. 2, «ID_Подорожі» є ключовим, тому що однозначно визначає екземпляр сутності.

Таблиця 2.2 - Атрибути сутності «Подорож»

Назва атрибуту	Опис атрибуту
ID_Подорожі	Унікальний ідентифікатор подорожі
План калорійності	Кількість калорій, що планується витрачати в день в грамах
Кількість учасників	Кількість учасників подорожі
Назва	Назва подорожі
Дата початку подорожі	Дата початку подорожі
Дата кінця подорожі	Дата кінця подорожі
ID_Користувача	Унікальний ідентифікатор користувача

Опис атрибутів сутності «Розкладка» наведено в табл. 3, «ID_Розкладки» є ключовим, тому що однозначно визначає екземпляр сутності.

Таблиця 2.3 - Атрибути сутності «Розкладка»

Назва атрибуту	Опис атрибуту
ID_Розкладки	Унікальний ідентифікатор розкладки
День	День в який додано продукт
Тип меню	Тип меню (сніданок/обід/вечеря) до якого додано продукт
Кількість продукту	Кількість продукту, що було додано, в грамах
ID_Подорожі	Унікальний ідентифікатор подорожі
ID_Продукту	Унікальний ідентифікатор продукту

Опис атрибутів сутності «Продукти» наведено в табл. 4, «ID_Продукту» є ключовим, тому що однозначно визначає екземпляр сутності.

Таблиця 2.4 - Атрибути сутності «Продукт»

Назва атрибуту	Опис атрибуту
ID_Продукту	Унікальний ідентифікатор продукту
Назва	Назва продукту
Калорійність на 100 г	Калорійність продукту на 100 грам маси
Маса стандартної порції	Маса стандартної порції продукту
ID_Категорії_продукту	Унікальний ідентифікатор категорії продукту

Опис атрибутів сутності «Категорії продуктів» наведено в табл. 5, «ID_Категорії_продукту» є ключовим, тому що однозначно визначає екземпляр сутності.

Таблиця 2.5 - Атрибути сутності «Категорія продукту»

Назва атрибуту	Опис атрибуту
ID_Категорії_продукту	Унікальний ідентифікатор категорії продукту
Назва	Назва категорії продукту

Визначимо зв’язки, що виникають між сутностями.

1. «Користувач» - «Подорож» (рис.2.4). Одному екземпляру сутності «Користувач» відповідає декілька екземплярів сутності «Подорож», так як створювати подорож може лише один користувач, та у одного користувача

може бути створено декілька подорожей. Причому у зв’язку «створює» для сутності «Користувач» - сутність «Подорож» є необов’язковою з множинністю багато, і навпаки для сутності «Подорож» - сутність «Користувач» є обов’язковою з множинністю один. Отже, встановлюється зв’язок «один-до-багатьох».

Рисунок 2.4 – Зв’язок «Користувач» - «Подорож»

2. «Подорож» - «Продукт» (рис.2.5). Кільком екземплярам сутності «Подорож» відповідає кілька екземплярів сутності «Продукт», так як в одну подорож можна взяти одразу декілька продуктів, та один продукт можна взяти одразу у декілька подорожей. Причому у зв’язку «составлена з» мають також бути вказані: день, тип меню та кількість взятого продукту. Також у цьому зв’язку для сутності «Подорож» - сутність «Продукт» є необов’язковою з множинністю багато, теж саме і для сутності «Продукт» - сутність «Подорож». Отже, встановлюється зв’язок «багато-до-багатьох».

Рисунок 2.5 – Зв’язок «Подорож» - «Продукт»

3. «Продукт» - «Категорія продукту» (рис.2.6). Кільком екземплярам сутності «Продукт» відповідає один екземпляр сутності «Категорія продукту», так як один продукт може належати лише до однієї категорії, а в одній категорії може бути декілька продуктів. Причому у зв’язку «належить» для сутності «Продукт» - сутність «Категорія продукту» є обов’язковою з множинністю один, а для сутності

«Категорія продукту» - сутність «Продукт» є обов’язковою з множинністю багато. Отже, встановлюється зв’язок «один-до-багатьох».

Рисунок 2.6 — Зв’язок «Продукт» - «Категорія продукту»

Виходячи з наведеного опису зв’язків, що виникають між сутностями, складемо загальну концептуальну ER-модель предметної області (рис. 2.7).

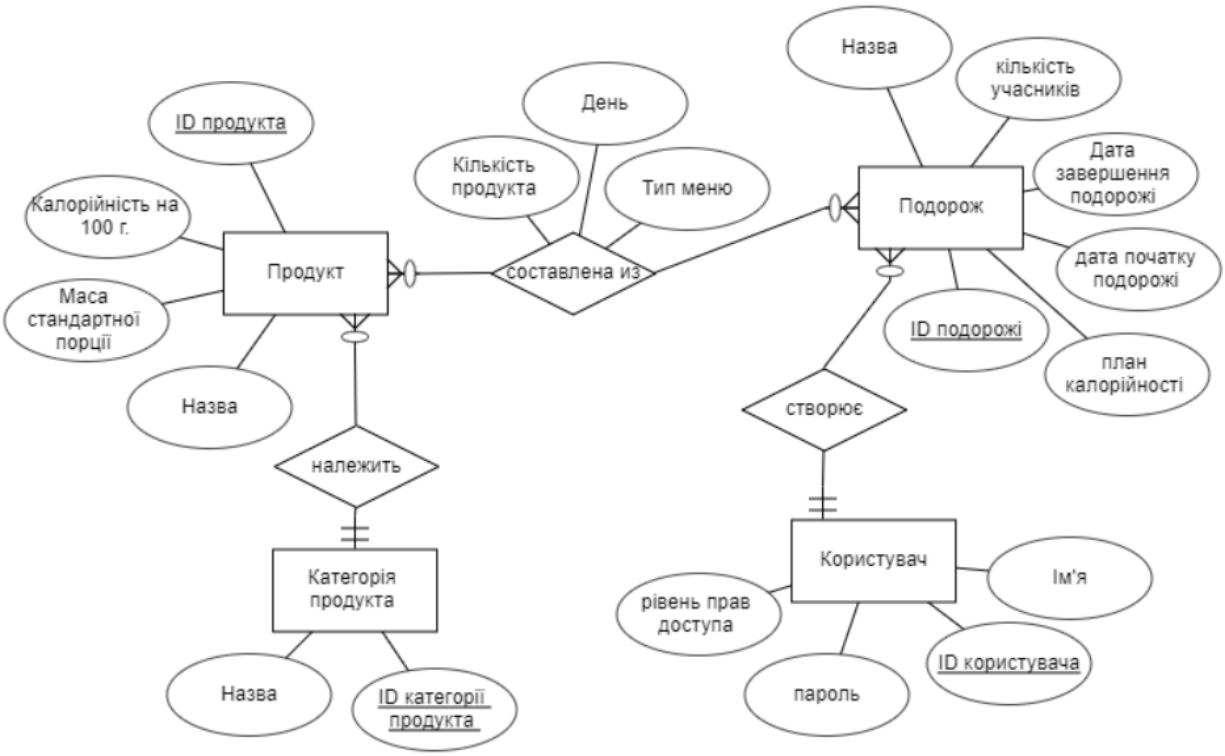

Рисунок 2.7 – Концептуальна модель бази даних проєкту

На основі всіх функціональних залежностей, та концептуальної ER-моделі предметної області побудованої вище, складемо підсумкову реляційну модель бази даних «Клієнт-серверного веб-застосунку для формування комплексного харчування при здійсненні туристичних походів на певний період часу» (рис. 2.8)

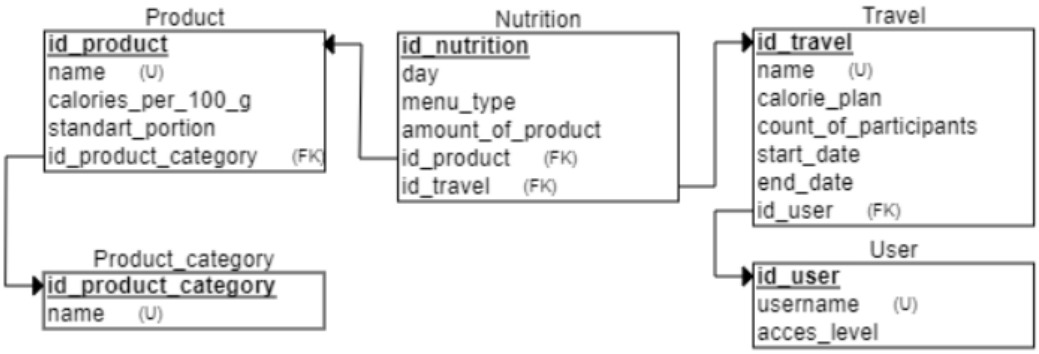

Рисунок 2.8 – Реляційна модель бази даних проєкту

Отже, нами було спроєктовано та змодельована база даних нашого клієнт-серверного застосунку.

2.3 Проєктування інтерфейсу клієнт-серверного застосунку

При розробці фронтенд частини було використано HTML, CSS та JavaScript. Для полегшення створення адаптивного дизайну зваженим рішенням стало використання фреймворку Bootstrap версії 3.3.7. За таких умов, фронтенд частина не буде використовувати зайві ресурси та її виконання буде найбільш ефективним.

На основі спроєктованих нами раніше алгоритмів, нами було розроблено інтерфейс клієнт-серверного застосунку зі створення харчових розкладок в подорож. Основною метою було спроєктувати інтерфейс таким чином, щоб вся необхідна інформація була розбита на категорії, що в свою чергу, візуально мали б розбиття на менші блоки. Виходячи з цього, нами було прийнято рішення зробити розбиття на категорії за допомогою пунктів меню, між якими вільно може переходити користувач, в свою чергу розбиття на блоки з різноплановою інформацією в рамках одного пункту меню буде організовано за допомогою внутрішньої HTML розмітки сторінки. Приклад спроєктованого інтерфейсу представлено на рисунку 2.9 на якому зображено головну сторінку майбутнього сайту.

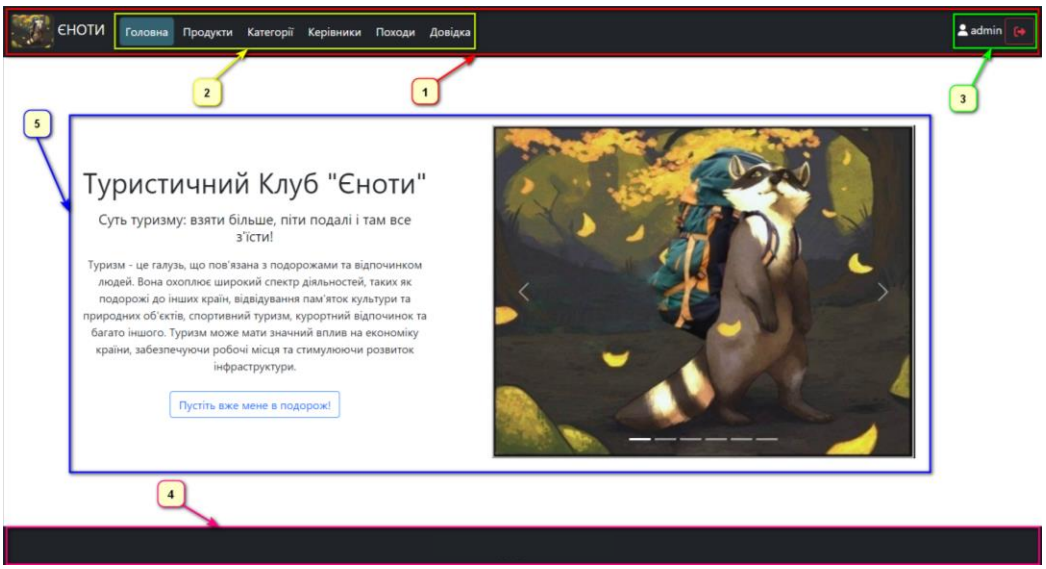

Рисунок 2.9 – Інтерфейс головної сторінки сайту

На кожній сторінці сайту присутні такі умовні блоки як:

1. Header.
2. Пункти меню.
3. Кнопка реєстрації/автентифікації.
4. Footer.
5. Блок з основною інформацією.

Також нами було розроблено інтерфейс основних пунктів меню сайту необхідних для ефективного створення харчових розкладок в подорож:

Пункт меню «Походи» (рис. 2.10) призначений для керування списком своїх подорожей, а також для перегляду харчових розкладок створених іншими керівниками подорожей. Основними елементами даної сторінки виступають:

1. Список подорожей із додатковою інформацією відносно керівника, що створив подорож, запланованої кількості учасників, початкової та кінцевої дат подорожі, плану калорійності та її статусу проходження.

2. Кнопки керування власними подорожами.

3. Елементи для здійснення швидкого пошуку шляхом сортування та пошуки за допомогою ключових слів.

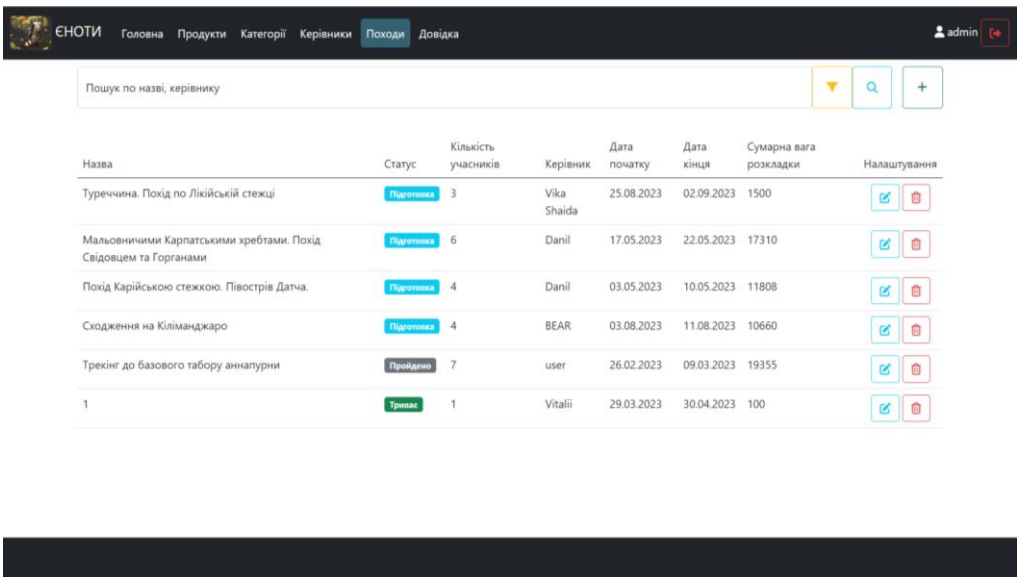

Рисунок 2.10 – Інтерфейс сторінки сайту пункту меню «Походи»

Пункт меню «Розкладка» (рис. 2.11) призначений для безпосереднього створення харчової розкладки. Основними елементами даної сторінки виступають:

1. Таблиця для відображення меню на кожен день подорожі.
2. Список продуктів, що можна додати до харчової розкладки.
3. Елементи навігації між днями меню подорожі.
4. Елемент для відображення про назву подорожі.
5. Кнопка для перегляду звітної інформації, щодо подорожі.

Також, для більш зручного користування розробленим застосунком, елемент, зо слугує для переходу між пунктами основного меню було зроблено рухливим, завдяки чому користувач матиме змогу повністю сконцентруватися на процесі створення харчової розкладки.

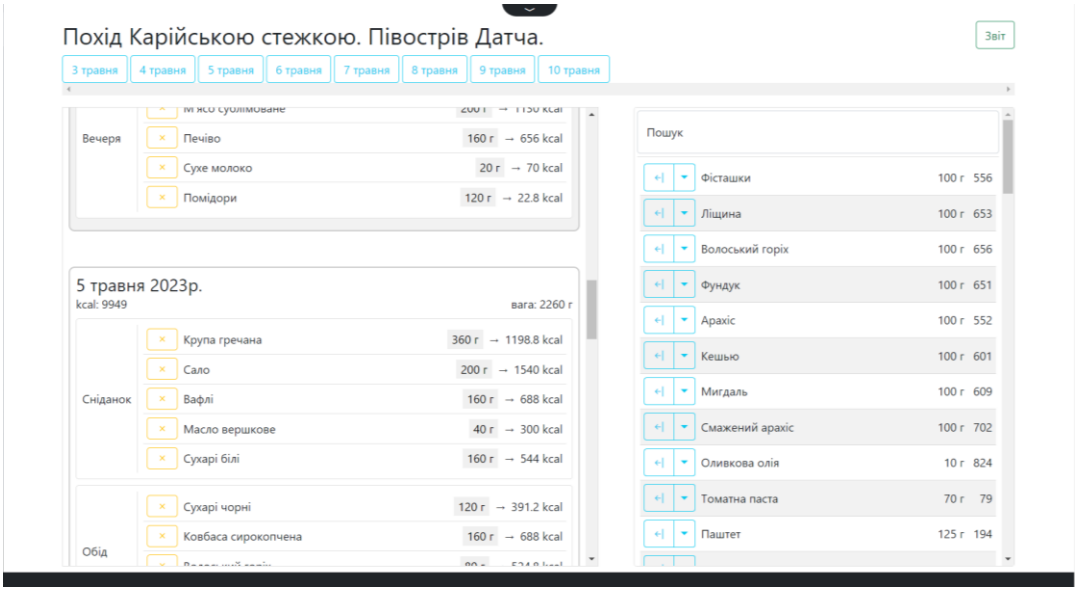

Рисунок 2.11 – Інтерфейс сторінки сайту пункту меню «Розкладка»

Пункт меню «Звіт» (рис. 2.12) призначений для формування списку продуктів для подальшої їх закупівлі в подорож, кожен пункт списку представляє собою сумарну вагу певного продукту на весь час запланованої подорожі. Основними елементами даної сторінки виступають:

1. Елемент з інформацією про назву подорожі.
2. Список продуктів та їх сумарна вага необхідних для закупівлі.

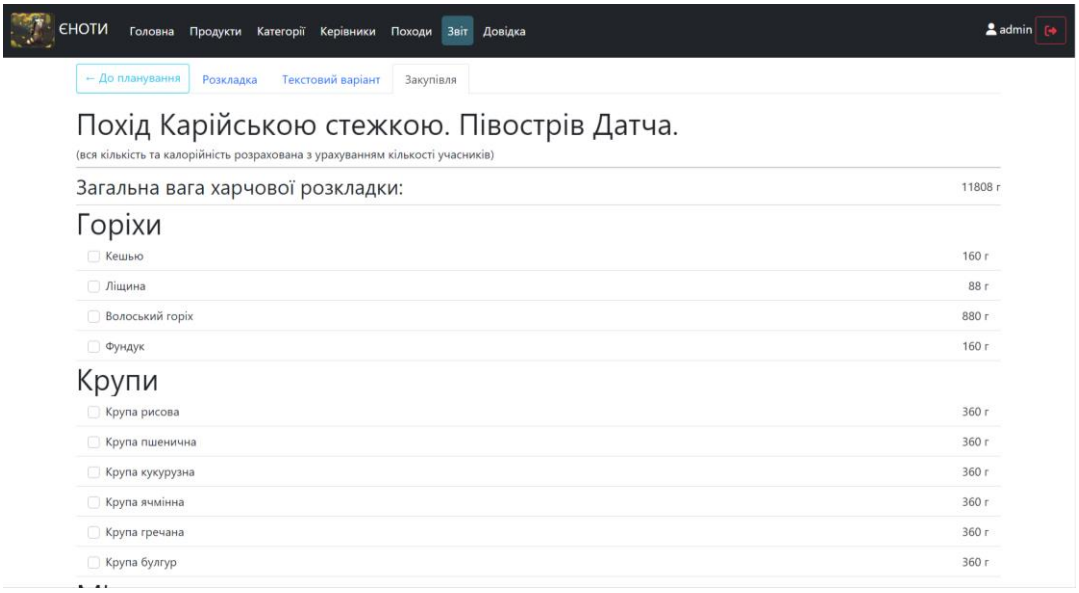

Рисунок 2.12 – Інтерфейс сторінки сайту пункту меню «Звіт»

Пункт меню «Довідка» (рис. 2.13) призначений для ознайомлення користувача з правилами користування розробленим застосунком. Основним елементом даної сторінки виступає блок із довідковою інформацією, щодо користування застосунком.

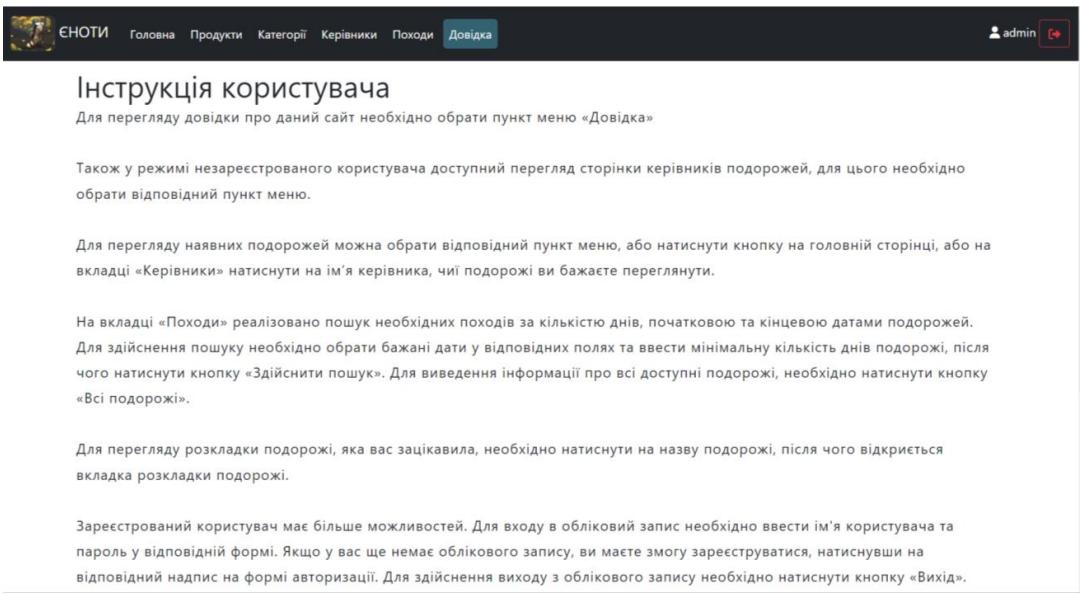

Рисунок 2.13 – Інтерфейс сторінки сайту пункту меню «Довідка»

Отже, нами було розроблено інтерфейс користувача для клієнт-серверного застосунку зі створення харчових розкладок в подорож.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Програмна реалізація бази даних клієнт-серверного застосунку

На підставі підсумкової реляційної моделі, опишемо таблиці бази даних «tourist_club», які були реалізовані в СУБД Oracle MySQL Server, за допомогою phpMyAdmin.

Таблиця 3.1 – Проєкт таблиці nutrition - «Розкладка»

Назва атрибуту	Тип даних	Розмір	Індексування
id_nutrition	int(11)	11	Так
day	int(11)	11	Ні
menu_type	int(11)	11	Ні
amount_of_product	int(11)	250	Ні
id_product	int(11)	11	Так
id_travel	int(11)	11	Так

Таблиця 3.2 – Проєкт таблиці product - «Продукт»

Назва атрибуту	Тип даних	Розмір	Індексування
id_product	int(11)	11	Так
name	varchar(50)	50	Так
calories_per_100_g	int(11)	11	Ні
standart_portion	int(11)	11	Ні
id_product_category	int(11)	11	Так

Таблиця 3.3 – Проєкт таблиці product_category - «Категорія продукта»

Назва атрибуту	Тип даних	Розмір	Індексування
id_product_category	int(11)	11	Так
name	varchar(50)	50	Так

Таблиця 3.4 – Проєкт таблиці travel - «Подорож»

Назва атрибуту	Тип даних	Розмір	Індексування
start_date	date		Ні
end_date	date		Ні
id_user	int(11)	11	Так
name	varchar(250)	250	Так
id_travel	int(11)	11	Так
calorie_plan	int(11)	11	Ні
count_of_participants	int(11)	11	Ні

Продовження таблиці 3.4

Таблиця 3.5 – Проєкт таблиці user - «Користувач»

Назва атрибуту	Тип даних	Розмір	Індексування
id_user	int(11)	11	Так
username	varchar(50)	50	Так
acces_level	varchar(50)	50	Ні
user_password	varchar(40)	40	Ні
user_hash	varchar(100)	100	Ні

При розробці бази даних клієнт-серверного web- застосунку одним із основних завдань - є вивід даних з бази даних на web-сторінки застосунку. Для цього виконані дії:

1. Встановлене з’єднання з базою даних:

```
// змінні для створення з'єднання з БД
$hostname = "localhost";
$username = "vikasova";
$password = "kva";
$dbName = "tourist_club";
// створення з'єднання з БД
$connection = mysqli_connect($hostname, $username, $password,
$dbName)
or die ("Не можу встановити з’єднання");
```

2. Виконана вибірка даних:

```
$query = "select product.id_product, product.name as prod_name,
product.calories_per_100_g, product.standart_portion,
product_category.name as prod_cat_name from product_category,
```

```
product WHERE product.id_product_category =
product_category.id_product_category ORDER by product.name"; //
вибірка з таблиці «Продукти» та «Продуктові категорії»
$query = "select * from user"; // вибірка з таблиці
«Користувачі»
$query = "SELECT id_travel, name, count_of_participants,
start_date, end_date, username, travel.id_user as idUser FROM
travel, user WHERE user.id_user = travel.id_user "; // вибірка з
таблиці «Подорожі»
$query = "SELECT * FROM nutrition WHERE id_travel = $idTravel";
// вибірка з таблиці «Розкладка»
```

Завданням дипломного проєкту передбачена зміна табличних даних за допомогою базового синтаксису виразів INSERT, UPDATE, DELETE та SELECT.

Додавання даних в проєкті відбувається за допомогою INSERT:

```
$qry = "INSERT INTO product (name, calories_per_100_g,
standart_portion, id_product_category) VALUES ('$name',
$calories_per_100_g, $standart_portion, $id_product_category)";
```

Оновлення даних в проєкті відбувається за допомогою UPDATE:

```
$qry = "UPDATE product SET name = '$name', calories_per_100_g =
$calories_per_100_g, standart_portion = $standart_portion,
id_product_category = $id_product_category WHERE id_product =
$id_product";
```

Видалення даних в проєкті відбувається за допомогою DELETE:

```
$qry =  "DELETE from product WHERE id_product =
".$_GET['delete']
```

Отже, нами було описано програмну реалізацію бази даних нашого клієнт-серверного застосунку.

3.2 Програмна реалізація основного алгоритму клієнт-серверного застосунку

У більшості своїй програмна реалізація основного алгоритму клієнт-серверного застосунку реалізована завдяки SQL-запитів.

Опишемо основні запити, що були реалізовані в нашому проєкті:

SQL-код запита «сумарна вага продуктів на меню/день/подорож»:

```
$query = "SELECT SUM(amount_of_product) as sum FROM nutrition
WHERE id_travel = $idTravel"; // запит на сумарну вагу продуктів
подорожі
$query = "SELECT SUM(amount_of_product) as sum FROM nutrition
WHERE id_travel = $idTravel and day = $i"; // запит на сумарну
вагу продуктів дня
```

```
$query = "SELECT SUM(amount_of_product) as sum FROM nutrition
WHERE id_travel = $idTravel and day = $i and menu_type = $j"; //
запит на сумарну вагу продуктів меню
```

SQL-код запита «сумарна калорійність продуктів на меню/день»:

```
$query_day = "SELECT * FROM nutrition WHERE id_travel =
$idTravel and day = $i"; // запит на всі продукти певного дня
$query_menu = "SELECT * FROM nutrition WHERE id_travel =
$idTravel and day = $i and menu_type = $j" // запит на всі
продукти перного меню
$query =  "SELECT calories_per_100_g FROM product WHERE
id_product = $kva" // запит на калорійність певного продукта
$calocies_by_day += $row['amount_of_product'] / 100 *
$row2['calories_per_100_g']; // підрахунок калорійності певної
кількості продукта
```

SQL-код запита «сумарна вага кожного продукту на всю подорож»:

```
$query = "SELECT SUM(amount_of_product) as kva FROM nutrition
WHERE id_product = $idProduct and id_travel = $idTravel"; //
запит на сумарну вагу кожного продукта на всю подорож
```

SQL-код запита «продукти певної категорії»:

```
$query = "select product.id_product, product.name as prod_name,
product.calories_per_100_g, product.standart_portion,
product_category.name as prod_cat_name from product_category,
product WHERE product.id_product_category =
product_category.id_product_category and
product.id_product_category = $kva ORDER by product.name"; //
запит на продукти певної категорії
```

SQL-код запита «пошук по назві інгредієнту»:

```
$query = "select product.id_product, product.name as prod_name,
product.calories_per_100_g, product.standart_portion,
product_category.name as prod_cat_name from product_category,
product WHERE product.id_product_category =
product_category.id_product_category and product.name = '$kva'
ORDER by product.name"; // запит на продукт з певною назвою
```

SQL-код запита «вивести список походів та сумарну вагу продуктів, для обраного проміжку дат»:

```
$query = "SELECT SUM(amount_of_product) as sum FROM nutrition
WHERE id_travel = $idTravel"; // запит на сумарну вагу розкладки
подорожі
$query = "SELECT id_travel, name, count_of_participants,
start_date, end_date, username, travel.id_user as idUser FROM
travel, user WHERE user.id_user = travel.id_user  and
travel.start_date >= '$kva'  and travel.end_date <= '$kva1'"; //
запит на подорожі, дата початку яких більша, або рівна даті
зазначені в пошуковому запросі, та дата кінця якої менше, або
дорівнює даті кінця подорожі зазначеній в пошуковому запиті
```

SQL-код запита «список походів, тривалість яких більше або дорівнює кількісті заданих днів»:

```
$query = "SELECT id_travel, name, count_of_participants,
start_date, end_date, username, travel.id_user as idUser FROM
travel, user WHERE user.id_user = travel.id_user"; // запит на
інформацію про кожну подорож
$count_of_days = 1 + date_diff(new DateTime($row1['end_date']),
new DateTime($row1['start_date']))->days; // визначення
тривалості подорожі.
```

Отже, нами було програмно реалізовано основний алгоритм клієнт-серверного застосунку.

3.3 Розробка багаторівневого доступу користувачів клієнт-серверного застосунку

Багаторівневий доступ користувачів до нашого клієнт-серверного застосунку реалізовано на основі рольового розмежування доступу. Це дає змогу визначити більш чіткі і зрозумілі для користувачів інформаційної системи рамки, щодо їх можливостей відносно даного застосунку.

Рольове розмежування доступу є розвитком дискреційної моделі розмежування доступу; при цьому права доступу суб'єктів системи на об'єкти групуються з урахуванням специфіки їх застосування. Це дає змогу визначити більш чіткі і зрозумілі для користувачів інформаційної системи правила розмежування доступу. На основі рольового розмежування доступу, в тому числі, може бути реалізовано мандатне розмежування доступу. Схему моделі рольового розмежування доступу зображено на рис. 2.14:

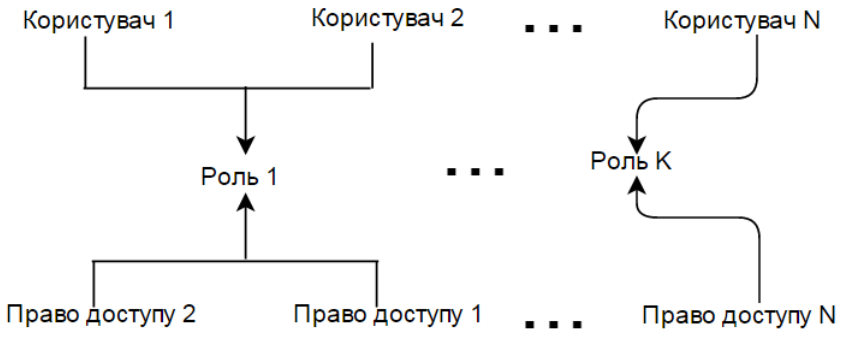

Рисунок 2.14 — Зовнішній вигляд схеми рольового розмежування доступу

Програмна реалізація реєстрації користувача має наступний вигляд:

```
// якщо була натиснута кнопка реєстрації
```

```
if(isset($_POST['register_button'])){
    $err = [];
     // перевірка на коректність вводу логіна
    if(!preg_match("/^[a-zA-Z0-9а-яА-
Я]+$/",$_POST['username_reg'])){
        $err[] = "Логін може складатись тільки з букв
англійського та українського алфавіту і цифр";  }
    // перевірка на коректність вводу логіна
    if(strlen($_POST['username_reg']) < 3 or
strlen($_POST['username_reg']) > 30){   $err[] = "Логін повинен
бути не менше 3-х символів і не більше 30";  }
     // зміні для перевірки на наявність логіна в базі даних
     $kva = $_POST['username_reg'];
     $query = mysqli_query($connection, "SELECT * FROM user
WHERE username = '$kva'");
// перевірка на наявність логіна в базі даних
    if(mysqli_num_rows($query) > 0){
        $err[] = "Користувач з таким логіном вже існує в базі
даних";}
     // перевірка на коректність повторного вводу пароля
     if($_POST['user_password_reg'] !=
$_POST['user_password_2_reg']){
        $err[] = "Повторно введений пароль не співпадає з
початковим";  }
// перевірка на наявність помилок
    if(count($err) == 0){
// змінні для реєстрації нового користувача
        $username_reg = $_POST['username_reg'];
        $password = $_POST['password'];
// створення запита до бази даних
        mysqli_query($connection,"INSERT INTO user (username,
acces_level, user_password) VALUES ('$username_reg', 'U',
'$password')");
        header("Location: members.php"); exit();
    } else {
// вивід помилок, що виникли при реєстрації
        print "<b>При реєстрації виникли наступні
помилки:</b><br>";
        foreach($err AS $error){
            print $error."<br>";
        }}}
```

Програмна реалізація авторизації користувача має наступний вигляд:

```
// якщо була здійснена авторизація
if(isset($_POST['autorization_button'])){
    $kva = $_POST['login']; // отримання змінних
    $query = mysqli_query($connection,"SELECT * FROM user where
username='$kva' LIMIT 1");// формування запиту
    $data = mysqli_fetch_assoc($query);
// якщо правильно введений пароль
    if($data['user_password'] == $_POST['password']){
        $hash = md5(generateCode(10)); // створення хєш кода
```

```
        mysqli_query($connection, "UPDATE users SET
user_hash='".$hash."' ".$insip." WHERE
id_user='".$data['id_user']."'");// здійснення запита
        setcookie("id", $data['id_user'], time()+60*60*24*30,
"/");
        setcookie("hash", $hash, time()+60*60*24*30, "/", null,
null, true);
          // отримання необхідних змінних
          $query = mysqli_query($connection,"SELECT * FROM user
where username='$kva' LIMIT 1");
          $user = mysqli_fetch_assoc($query);
          // заповнення змінної сесії значеннями
          $_SESSION['auth'] = true;
          $_SESSION['acces_level'] = $user['acces_level'];
          $_SESSION['id_user'] = $user['id_user'];
          // перехід на головну сторінку
        header("Location: members.php"); exit();
    } else{ // попередження про помилку
        print "Ви ввели неправильний логін/пароль"; }
Програмна реалізація розмежування прав доступу користувачів:
<!--якщо користувач має права доступу адміністратора, йому
надається доступ до вкладки продуктів-->
<? if ( $_SESSION['acces_level']== 'A'){?>
<li> <a href="products.php">Продукти</a> </li> <? }?>
```

Отже, нами було розроблено багаторівневий доступ до клієнт-серверного застосунку для створення харчових розкладок в подорож.

3.4 Розробка інструкції користувача

Для зручності користування розробленим веб-застосунком нами було розроблено інструкцію користувача, що давала б вичерпну інформацію щодо користування сайтом.

Скопіювати папку OpenServer до кореневого каталогу диску C: чи D:

Запустити від імені адміністратора OpenServer x86 або OpenServer x64, в залежності від версії операційної системи. В треї з’явиться піктограма прапорця. Натиснути правою кнопкою миші та обрати пункт «Запустити». Якщо все гаразд, прапорець повинен стати зеленого кольору. Знову натискаємо правою кнопкою миші на прапорець та обираємо Мої сайти>localhost, після цього перед вами з’явиться головна сторінка сайту (рис. 2.15).

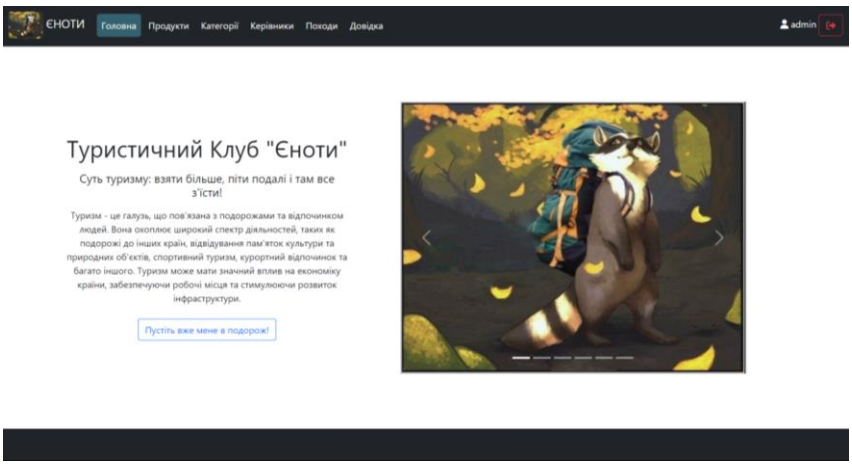

Рисунок 2.15 – Головна сторінка сайту

Наступним кроком буде реєстрація, або, якщо акаунт вже було попередньо створено, автентифікація. Для цього необхідно натиснути на кнопку в правій верхній частині сайту (рис. 2.16). Після заповнення необхідної інформації та її перевірки буде створено, або здійснено вхід до облікового запису користувача.

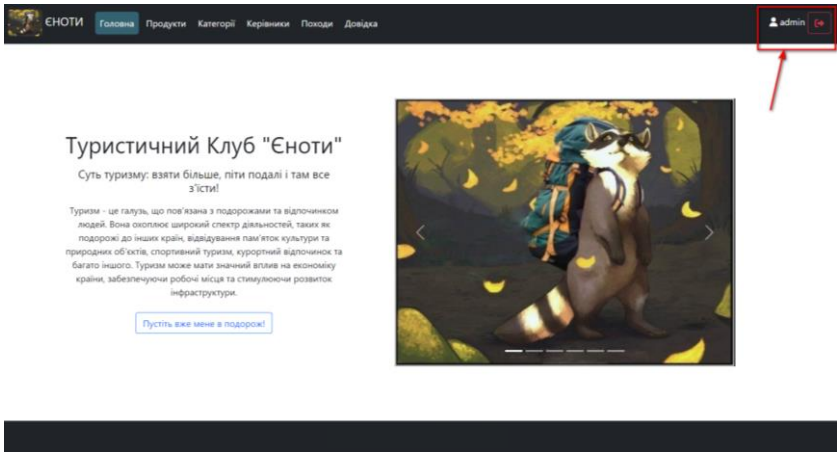

Рисунок 2.16 – Інтерфейс сторінки сайту пункту меню «Подорожі»

Після входу до облікового запису, користувач має змогу створити свій власний похід, для якого буде створюватися майбутня харчова розкладка (рис. 2.17). Для цього необхідно перейти до пункту меню «Походи» та натиснути кнопку «+» в правій верхній частині екрану, після чого необхідно буде ввести необхідну інформацію до відповідних полів та натиснути кнопку «Створити». Після даних маніпуляцій на вищезгаданій вкладці з’явиться назва та короткий опис створеної вами подорожі.

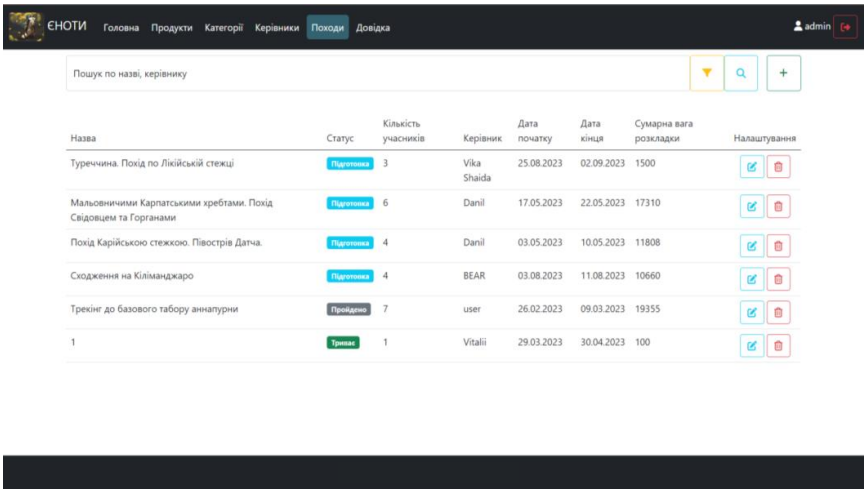

Рисунок 2.18 – Пункт меню «Подорожі»

Для безпосереднього створення харчової розкладки необхідно натиснути на попередньо створений похід, після чого буде здійснено перехід до вкладки створення харчової розкладки (рис. 2.19), де для зручності користувача буде доступно список продуктів, з яких можна створити харчову розкладку, також для спрощення пошуку усі продукти можна відсортувати за категоріями та знайти за допомогою ключових слів.

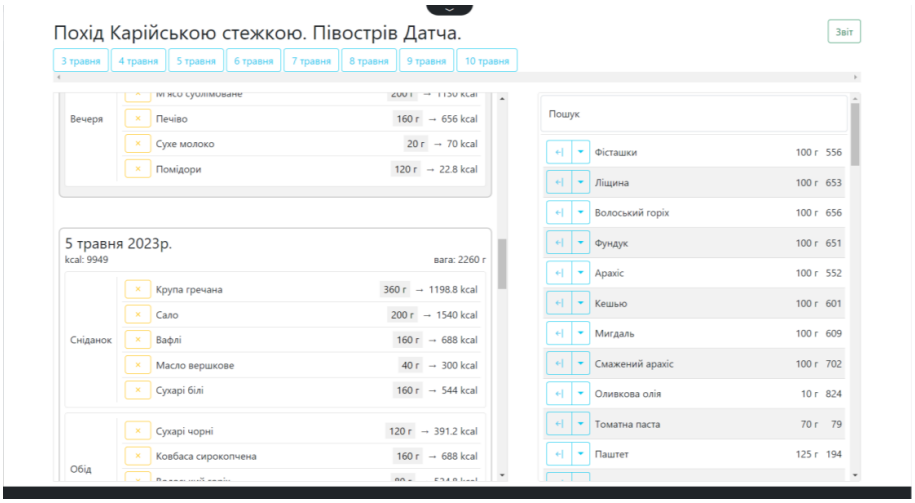

Рисунок 2.19 – Пункт меню «Розкладка»

Також для користувача доступна таблиця харчової розкладки, що створюється, та кнопка для оформлення звітності щодо продуктів, які необхідно закупити в подорож (рис. 2.20).

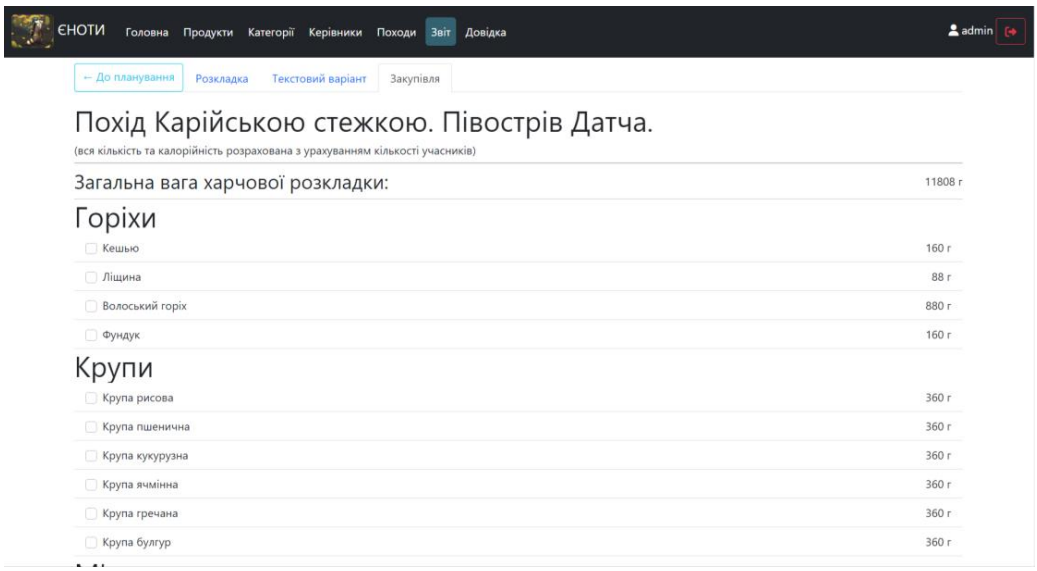

Рисунок 2.20 – Пункт меню «Звіт»

У разі виникнення запитань користувач завжди може скористатися довідкою, що розташована на сайті у вигляді окремого пункту меню (рис. 2.21).

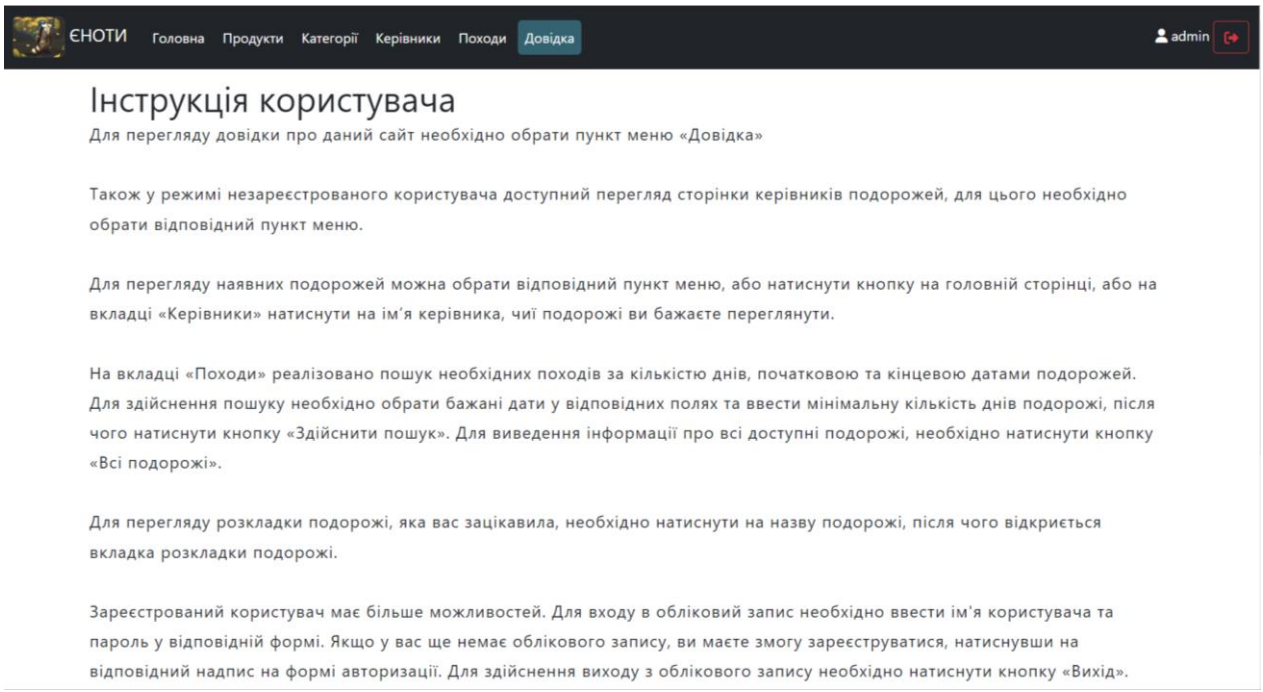

Рисунок 2.21 – Пункт меню «Довідка»

ВИСНОВКИ

В даній кваліфікаційній роботі було розглянуто процес розробки клієнт-серверного веб-застосунку для формування комплексного харчування при туристичних походах. Основною метою проекту було автоматизувати процес розрахунку необхідної кількості продуктів для зменшення впливу людського фактору на цей процес, що є особливо важливим для забезпечення успішності та комфортності тривалих подорожей.

У ході виконання роботи було здійснено аналіз предметної області, виявлено основні проблеми та труднощі, з якими стикаються туристи при плануванні харчування. Визначено, що традиційні методи розрахунку кількості продуктів за допомогою блокнота та калькулятора є неефективними та можуть призвести до помилок через людський фактор. Було проведено аналіз існуючих рішень на ринку, виявлено їх недоліки та визначено, що вони не відповідають повною мірою вимогам до автоматизації цього процесу.

В роботі було розроблено структуру баз даних, архітектуру застосунку та його інтерфейс, що забезпечують зручність користування та ефективність розрахунків.

Основними досягненнями роботи стали автоматизація розрахунків ваги та калорійності продуктів, створення зручного інтерфейсу для користувачів, а також забезпечення можливості зберігання та обробки даних на серверній частині проекту. Розроблений застосунок дозволяє користувачам швидко та точно планувати харчування на весь період подорожі, зменшуючи ризик помилок та підвищуючи загальний комфорт та безпеку туристичних походів.

Таким чином, виконана робота має значну практичну цінність і може бути використана як основа для подальшого вдосконалення та розвитку систем планування харчування в туристичних подорожах. Результати роботи можуть бути застосовані для створення аналогічних систем в інших галузях, де необхідно автоматизувати процеси розрахунку ресурсів та зменшити вплив людського фактору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги та монографії:

1. Іванов І. І., Петров П. П. Основи туристичного харчування. Київ: Видавництво "Турист", 2020.
2. Сидоров С. С. Інформаційні системи та технології. Харків: Видавництво "Наука і Техніка", 2018.

Наукові статті:

3. Коваленко А. А., Тарасенко В. В. Автоматизація процесу планування харчування для туристичних походів. Журнал "Інформаційні Технології", 2021, №4, С. 45-53.
4. Петренко М. М. Геоінформаційні системи в туризмі: сучасні тенденції та перспективи. Вісник Київського національного університету, 2019, №2, С. 78-85.

Інтернет-ресурси:

5. Туристичний портал України. Планування харчування для походів. [Електронний ресурс] – Режим доступу: http://www.tourismportal.ua – Дата звернення: 15.05.2024.
6. Веб-сайт для туристів Гід по харчуванню в походах. [Електронний ресурс] – Режим доступу: http://www.hikingfoodguide.com – Дата звернення: 20.05.2024.

Навчальні посібники:

7. Смирнов О. О. Веб-технології та веб-дизайн. Навчальний посібник. Львів: Видавництво Львівського університету, 2017.
8. Гончаренко Д. Д. Розробка клієнт-серверних додатків. Київ: Видавничий дім "Освіта", 2019.

Документація та технічні звіти:

9. Технічний звіт про використання геоінформаційних систем в туризмі. Інститут геоінформаційних технологій, 2022.

ДОДАТОК А – КОД ЗАСТОСУНКУ

Index

```
<?php
    require 'assets/php/Partials.php';
    session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php Partials::Head('Головна');  ?>

    <style>
        .column-first{
            width: 40%;
        }
        .column-second{
            width: 60%;
        }
        @media screen and (max-width: 800px) {
            .main-container {
                margin-top: 20px;
                margin-bottom: 20px;
                flex-direction: column-reverse!important;
            }
            .column-first, .column-second{
                width: 100%;
            }
        }

    </style>
</head>
<body>
    <?php
        Partials::NavBar('index', $_SESSION['user']);
    ?>

    <div class="main-container container content gap-3 d-flex
flex-row justify-content-center text-center align-items-center">
            <div class="column-first d-flex flex-column gap-3">
                <h1>Туристичний Клуб "Єноти"</h1>
                <h5>Суть туризму: взяти більше, піти подалі і
там все з'їсти!</h5>
                Туризм - це галузь, що пов'язана з подорожами та
відпочинком людей. Вона охоплює широкий спектр діяльностей,
таких як подорожі до інших країн, відвідування пам'яток культури
та природних об'єктів, спортивний туризм, курортний відпочинок
та багато іншого. Туризм може мати значний вплив на економіку
країни, забезпечуючи робочі місця та стимулюючи розвиток
інфраструктури.
```

```
                <div class="mt-2"><a href="hiking.php"
class="btn btn-outline-primary">Пустіть вже мене в
подорож!</a></div>
            </div>
        <div class="column-second d-flex justify-content-
center">
            <div style="max-width: 600px;"
id="carouselExampleIndicators" class="carousel slide">
                <div class="carousel-indicators">
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="0"
class="active" aria-current="true" aria-label="Slide
1"></button>
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="1" aria-
label="Slide 2"></button>
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="2" aria-
label="Slide 3"></button>
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="3" aria-
label="Slide 4"></button>
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="4" aria-
label="Slide 5"></button>
                    <button type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide-to="5" aria-
label="Slide 6"></button>
                </div>
                <div class="carousel-inner">
                    <div class="carousel-item active">
                        <img class="d-block"
src="assets/img/enot.png" alt="Тут мав бути ЄНОТіше □" >
                    </div>
                    <div class="carousel-item">
                        <img class="d-block"
src="assets/img/carousel_item1.png" alt="Тут мав бути ЄНОТіше □"
>
                    </div>
                    <div class="carousel-item">
                        <img class="d-block"
src="assets/img/carousel_item2.png" alt="Тут мав бути ЄНОТіше □"
>
                    </div>
                    <div class="carousel-item">
                        <img class="d-block"
src="assets/img/carousel_item3.png" alt="Тут мав бути ЄНОТіше □"
>
                    </div>
                    <div class="carousel-item">
                        <img class="d-block"
src="assets/img/carousel_item4.png" alt="Тут мав бути ЄНОТіше □"
>
```

```
                    </div>
                    <div class="carousel-item">
                        <img class="d-block"
src="assets/img/carousel_item5.png" alt="Тут мав бути ЄНОТіше □"
>
                    </div>
                </div>
                <button class="carousel-control-prev"
type="button" data-bs-target="#carouselExampleIndicators" data-
bs-slide="prev">
                    <span class="carousel-control-prev-icon"
aria-hidden="true"></span>
                    <span class="visually-
hidden">Previous</span>
                </button>
                <button class="carousel-control-next"
type="button" data-bs-target="#carouselExampleIndicators" data-
bs-slide="next">
                    <span class="carousel-control-next-icon"
aria-hidden="true"></span>
                    <span class="visually-hidden">Next</span>
                </button>
            </div>

        </div>
    </div>

    <?php
        Partials::Footer();
    ?>
</body>
</html>
```

Categories

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
    /* встановлення зв'язку з бд */
    $db = new ClubDB();
    Partials::Head('Категорії');
    /* перевірка режиму сторінки */
    if ($_GET['mode']=='editing' && $_GET['cid']){
        $categoryEdit = $db->getCategory($_GET['cid']);
        $_GET['category_id'] = $categoryEdit->getId();
        $_GET['category_name'] = $categoryEdit->getName();
```

```
    }

    ?>
</head>
<body>
<?php
Partials::NavBar('categories', $_SESSION['user']);
?>
<div class="main-container container content gap-3 d-flex flex-
column pt-2 pb-2 ">
    <div class="d-flex flex-row gap-3">
        <form action="categories.php" method="POST" class="d-
flex flex-row w-100">
            <div class="form-floating mb-3 flex-grow-1">
                <input type="text" class="form-control"
id="floatingInput" name="search_name" placeholder="Пошук по
назві, категорії" value="<?= $_POST['search_name']?>">
                <label for="floatingInput">Пошук по
назві</label>
            </div>
            <?php if($_POST['search_name']): ?>
                <a style="height: 58px; width: 58px;"
href="categories.php" class="btn btn-outline-info d-flex
justify-content-center align-items-center">
                    <script
src="https://cdn.lordicon.com/qjzruarw.js"></script>
                    <lord-icon

src="https://cdn.lordicon.com/akuwjdzh.json"
                        trigger="hover"
                        style="width:25px;height:25px">
                    </lord-icon>
                </a>
            <?php endif; ?>
            <button type="submit" style="height: 58px; width:
58px;" class="btn btn-outline-info"><i class="fa-solid fa-
magnifying-glass"></i></button>
        </form>
        <a style="height: 58px; width: 58px;"
href="categories.php?mode=adding" class="btn btn-outline-success
d-flex justify-content-center align-items-center"><i class="fa-
solid fa-plus"></i></a>
    </div>
    <div style="max-height: calc(100vh - 250px); overflow:
auto;">
        <table class="table table-hover">
            <thead>
            <tr>
                <td>Назва</td>
                <?php if ($_SESSION['user']['access_level'] ==
'A'): ?>
                    <td class="text-center">Налаштування</td>
                <?php endif; ?>
```

```
            </tr>
            </thead>
            <tbody>
            <!-- формуємо список категорій-->
            <?php foreach (($_POST['search_name'])?$db-
>searchCategory($_POST['search_name']):$db->getCategories() as
&$category):?>
                <tr>
                    <td><a class="d-block"
href="products.php?category_search=<?=$category->getId();?>"><?=
$category->getName(); ?></a></td>
                    <?php if ($_SESSION['user']['access_level']
== 'A'): ?>
                        <td class="text-center">
                            <a class="btn btn-outline-info"
href="categories.php?mode=editing&cid=<?php echo $category-
>getId(); ?>"><i class="fa-solid fa-pen-to-square"></i></a>
                            <?php if($category->getId()!='12'):
?>
                                <a class="btn btn-outline-
danger" href="categories.php?del=true&cid=<?php echo $category-
>getId(); ?>"><i class="fa-regular fa-trash-can"></i></a>
                            <?php endif;?>
                        </td>
                    <?php endif; ?>
                </tr>
            <?php endforeach;?>
            </tbody>
        </table>

    </div>
</div>


<!-- Додавання нового продукту -->
<script>
    /* функція перевірки вхідних даних */
    function checkDataAndSubmit(){

document.getElementById("category_name").classList.remove("is-
invalid")

        if
(document.getElementById("category_name").value.trim().length
=== 0){

document.getElementById("category_name").classList.add("is-
invalid")
            return;
        }

        if
(document.getElementById("category_name").value.trim().length >
```

```
50){

document.getElementById("category_name").classList.add("is-
invalid")
            return;
        }
        document.getElementById("modalAdd").submit();
    }
</script>
<button style="height: 58px; width: 58px;" type="button"
id="btnOpenModal" class="btn btn-outline-success" data-bs-
toggle="modal" data-bs-target="#addModal" hidden></button>
<div class="modal fade" id="addModal" tabindex="-1" data-bs-
backdrop="static" data-bs-keyboard="false" aria-
labelledby="addModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel"><?= ($_GET['mode']=='adding')?'Додавання
нової':'Редагування' ?> категорії</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="assets/php/api.php"
name="category_add" method="post" id="modalAdd"  class="modal-
body">
                <input type="text" name="category_mode"
value="<?=
($_GET['mode']=='adding')?'category_add':'category_edit' ?>"
hidden readonly>
                <input type="text" name="category_id" value="<?=
$_GET['category_id']?>" hidden readonly>
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="category_name" name="category_name" placeholder="...."
value="<?= $_GET['category_name']?>">
                    <label for="category_name">Назва
категорії</label>
                </div>
            </form>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
onclick="checkDataAndSubmit()"><?=
($_GET['mode']=='adding')?'Додати':'Редагувати' ?></button>
            </div>
        </div>
    </div>
</div>
<?php
Partials::Footer();
```

```
/* редагування категорії */
if (($_GET['mode']=='editing' && $_GET['cid']) ||
$_GET['mode']=='adding'){
    Partials::SimClick('btnOpenModal');
}
/* видалення категорії */
if ($_GET['del']=='true' && $_GET['cid']){
    if ($db->getCategory($_GET['cid']) != null)
        Partials::DeleteDialog('Видалення', 'Ви впевнені, що
бажаєте видалити категорію - '.$db->getCategory($_GET['cid'])-
>getName(),
'assets/php/api.php?category_del=true&cid='.$_GET['cid']);
}
?>
</body>
</html>
```

Feedback

```
<?php
require 'assets/php/Partials.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php Partials::Head('Зв\'язок');  ?>

    <style>
        .column-first{
            width: 40%;
        }
        .column-second{
            width: 60%;
        }
        @media screen and (max-width: 800px) {
            .main-container {
                margin-top: 20px;
                margin-bottom: 20px;
                flex-direction: column-reverse!important;
            }
            .column-first, .column-second{
                width: 100%;
            }
        }

    </style>
</head>
<body>
<?php
Partials::NavBar('feedback', $_SESSION['user']);
?>
```

```
<div class="main-container container content gap-3 d-flex flex-
row-reverse justify-content-center text-center align-items-
center">

    <div class="column-second d-flex justify-content-center">
        <form id="contact-form">
            <div class="row">
                <div class="col-6">
                    <div class="form-floating mb-3">
                        <input type="text" class="form-control"
id="first-name" placeholder=" ">
                        <label for="first-name">Ім'я</label>
                    </div>
                </div>
                <div class="col-6">
                    <div class="form-floating mb-3">
                        <input type="text" class="form-control"
id="last-name" placeholder=" ">
                        <label for="last-name">Прізвище</label>
                    </div>
                </div>
            </div>
            <div class="form-floating mb-3">
                <input type="email" class="form-control"
id="email" placeholder=" ">
                <label for="email">Пошта</label>
            </div>
            <div class="form-floating mb-3">
                <textarea class="form-control" placeholder=" "
id="message" style="height: 100px"></textarea>
                <label for="message">Повідомлення</label>
            </div>
            <div class="d-flex justify-content-end">
                <button type="submit" class="btn btn-outline-
success mb-3">Надіслати</button>
            </div>
            </form>
    </div>
    <div class="column-first d-flex flex-column gap-3">
        <h1>Зв'яжіться з нами</h1>
        <h5>Хочете зв'язатися з нами? Заповніть форму із
запитом. </h5>
    </div>
</div>
<script>
    document.getElementById('contact-form').onsubmit = (e) =>{
        e.preventDefault();
        let url = "mailto:skirchak.vlad@gmail.com?subject=Форма
зв'язку";

url+="&body=Доброго%20дня,%20я%20"+document.getElementById('firs
t-name').value+'%20'+document.getElementById('last-
```

```
name').value+'.%0A';

url+="Моя%20пошта:%20"+document.getElementById('email').value+"%
0A";
        url+=document.getElementById('message').value;
        window.location.href = url;
    }

</script>
<?php
Partials::Footer();
?>
</body>
</html>
```

Food

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
    /* встановлення зв'язку з бд */
    $db = new ClubDB();
    //отримуємо похід
    $travel = $db->getTravel($_GET['hid']);
    Partials::Head('Походи');
    ?>
    <script>

    </script>
    <style>
        @import url("assets/bootstrap-
icons/font/fonts/bootstrap-icons.woff");
        @import url("assets/bootstrap-
icons/font/fonts/bootstrap-icons.woff2");
        .hide-show{
            position: absolute;
            left: 0;
            right: 0;
            top: -74px;
            z-index: 999999;
            transition: all .5s;
        }

        .hide-show::before{
            position: relative;
            top: 45px;
            left: calc(50% - 50px);
```

```
            width: 80px;
            height: 20px;
            border-radius: 0 0 20px 20px;
            background: #212529;
            display: flex;
            justify-content: center;
            font-family: 'Bootstrap-icons';
            content: "\F279";
            align-items: center;
            color: white;
            cursor: pointer;
        }

        .hide-show:hover{
            position: absolute;
            left: 0;
            right: 0;
            top: 0;
        }

        body:first-child{
            position: absolute;
            top: 100vh;
            left: 0;
            right: 0;
        }

        .product{

        }
        .product-weight, .product-caloric{
            min-width: 30px;
            text-align: right;
        }
        .product-title{
            white-space: nowrap; /* Prevents text from wrapping
*/
            overflow: hidden; /* Hides any text that overflows
the container */
            text-overflow: ellipsis; /* Displays an ellipsis
when text is truncated */
        }
        .text-ellipsis{
            white-space: nowrap; /* Prevents text from wrapping
*/
            overflow: hidden; /* Hides any text that overflows
the container */
            text-overflow: ellipsis; /* Displays an ellipsis
when text is truncated */
        }
        .day-part{
            min-width: 75px;
            border-right: 1px solid #DFDFDF;
```

```
            display: flex;
            align-items: center;
            cursor: pointer;
        }
        .active-day-part{
            border: 2px solid lightgreen;
            background: #EAF6EA;
        }

        .hiking-days > li:nth-child(even) {
            background-color: #f2f2f2;
        }
        .hiking-days > li:nth-child(odd) {
            background-color: #ffffff;
        }
        .products-container > li:nth-child(even) {
            background-color: #f2f2f2;
        }

        .hiking-days,.products-container > li:nth-child(odd) {
            background-color: #ffffff;
        }

        #days-container{
            <?php if ($_SESSION['user']['uid']==$travel-
>getUserId() || $_SESSION['user']['access_level']=='A'):?>
                max-width: 60%;
            <?php else:?>
                max-width: 100%;
            <?php endif;?>
        }
        #products-div{
        <?php if ($_SESSION['user']['uid']==$travel->getUserId()
|| $_SESSION['user']['access_level']=='A'):?>
            max-width: 40%;
        <?php else:?>
            display: none!important;
        <?php endif;?>

        }

        @media screen and (max-width: 1200px){
            .hide-show{
                top: -95px;
            }
            .hide-show::before{
                top: 85px;
            }
        }

        @media screen and (max-width: 900px){
            .product-weight{
                display: none;
```

```
            }
        }
        @media screen and (max-width: 800px){
            .content{
                flex-direction: column!important;
            }
            #days-container{
                max-width: 100%;
            }
            #products-div{
                max-width: 100%;
            }
        }

        @media screen and (max-width: 600px) {
            .day-part-card{
                flex-direction: column!important;
            }

            .day-part{
                border-right: none!important;
                border-bottom: 1px solid #DFDFDF!important;
            }
        }
    </style>
</head>
<body class="d-flex flex-column">
<?php
Partials::NavBar('food', $_SESSION['user']);
?>
<div class="main-container container-xl content gap-3 d-flex
flex-grow-1 flex-column pt-2 pb-2 ">
    <input type="number" name="hiking_id" id="hiking_id"
value="<?= $_GET['hid']; ?>" readonly hidden>
    <input type="number" name="hiking_user_id"
id="hiking_user_id" value="<?= $travel->getUserId(); ?>"
readonly hidden>
    <input type="number" name="access_level" id="access_level"
value="<?= ($_SESSION['user']['access_level']=='A')?'1':'0' ?>"
readonly hidden>
    <input type="number" name="user_id" id="user_id" value="<?=
$_SESSION['user']['uid']; ?>" readonly hidden>
    <div class="basic-info mt-3">
        <div class=" d-flex flex-row justify-content-between mb-
2">
            <h2 class="product-title"><?= $travel-
>getName();?></h2>
            <div>
                <a href="zvit.php?hid=<?= $_GET['hid']; ?>"
class="btn btn-outline-success">Звіт</a>
            </div>
        </div>
        <div class="hiking-navigation d-flex flex-row gap-1"
```

```
style="overflow-x: scroll;" id="hiking-navigation">
        </div>
    </div>
    <div class="content d-flex flex-row gap-md-5 flex-grow-1">
        <ul class="hiking-days card flex-grow-1 list-group list-
group-flush flex-grow-1 p-2 gap-5" id="days-container"
style="overflow: auto; max-height: calc(100vh - 170px);">

        </ul>
        <div class="products card flex-grow-1 d-flex flex-column
gap-1 p-1" id="products-div" style="overflow: auto; max-height:
calc(100vh - 170px);">
            <div class="form-floating sticky-top bg-white">
                <input type="text" class="form-control"
id="search_product" placeholder="name@example.com">
                <label for="search_product">Пошук</label>
            </div>
            <ul class="products-container list-group list-group-
flush flex-grow-1 p-1" id="products-container">
            </ul>
        </div>
    </div>
</div>
<button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#portion-modal" id="btn-show-
modal-portion" hidden></button>
<div class="modal fade" id="portion-modal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel">Введіть вагу порції на одного</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-weight" placeholder="вага">
                    <label for="portion-weight">Вага</label>
                </div>
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-caloric" placeholder="вага" readonly>
                    <label for="portion-caloric">Калорії</label>
                </div>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
id="cancel-modal" data-bs-dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
id="add-non-standard-portion">Додати</button>
```

```
            </div>
        </div>
    </div>
</div>
<button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#portion-modal-update" id="btn-
show-modal-portion-update" hidden></button>
<div class="modal fade" id="portion-modal-update" tabindex="-1"
aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel">Введіть вагу порції на одного</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-weight-update" placeholder="Вага">
                    <label for="portion-weight-
update">Вага</label>
                </div>
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-caloric-update" placeholder="Калорії" readonly>
                    <label for="portion-caloric-
update">Калорії</label>
                </div>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
id="cancel-modal-update" data-bs-
dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
id="add-update-portion">Зберегти</button>
            </div>
        </div>
    </div>
</div>
<button type="button" class="btn btn-primary" data-bs-
toggle="modal" data-bs-target="#portion-modal-ultra-add"
id="btn-show-modal-portion-ultra-add" hidden></button>
<div class="modal fade" id="portion-modal-ultra-add" tabindex="-
1" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel">Введіть вагу порції на одного</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
```

```
            </div>
            <div class="modal-body d-flex flex-column gap-1">
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-weight-ultra-add" placeholder="вага">
                    <label for="portion-weight-
update">Вага</label>
                </div>
                <div class="form-floating">
                    <input type="number" class="form-control"
id="portion-caloric-ultra-add" placeholder="вага" readonly>
                    <label for="portion-caloric-
update">Калорії</label>
                </div>
                <div class="form-floating">
                    <select class="form-select" id="portion-
menu-type-ultra-add" aria-label="Floating label select example">
                        <option value="NONE" selected>Відкрийте
це меню</option>
                        <option value="1">Сніданок</option>
                        <option value="2">Обід</option>
                        <option value="3">Вечеря</option>
                    </select>
                    <label for="floatingSelect">Оберіть
меню</label>
                </div>
                <div class="portion-days-ultra-add d-flex flex-
wrap gap-1 border-2 p-1 " id="ultra-days">

                </div>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
id="cancel-modal-ultra" data-bs-
dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
id="add-ultra-portion">Додати</button>
            </div>
        </div>
    </div>
</div>
<script src="assets/js/hiking.js"></script>
<?php
Partials::Footer();
?>
</body>
</html>
```

Hiking

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
require 'assets/php/functions.php';
```

```
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
        /* встановлення зв'язку з бд */
        $db = new ClubDB();
        Partials::Head('Походи');
        if ($_GET['member_search']) $_POST['search_name'] =
$_GET['member_search'];
    ?>
    <script>
        function redirect(ref){
            window.location.href = ref;
        }
    </script>
    <style>
        .content-table{
            max-height: calc(100vh - 250px);
            overflow-y: auto;
        }

        @media screen and (max-width: 1280px) {
            .date-filter{
                flex-direction: column!important;
            }
        }
        @media screen and (max-width: 1025px) {
            .main-container{
                margin: 0;
                max-width: 100%!important;
            }
        }

        @media screen and (max-width: 970px) {
            .count-filters-my {
                flex-direction: column!important;
            }
        }

        @media screen and (max-width: 800px) {
            .filters{
                flex-direction: column!important;
                gap: 5px;
            }
            .content-table{
                max-height: 100%;
            }
        }

        @media screen and (max-width: 770px) {
```

```
            .filters-second{
                flex-direction: column!important;
                gap: 5px;
            }
            .main-container{
                margin: 0;
                max-width: 100%!important;
            }
            .overflow-table{
                display: none;
            }
        }
    </style>
</head>
<body>
<?php
Partials::NavBar('hiking', $_SESSION['user']);
?>

<div class="main-container container content gap-3 d-flex flex-
column">
    <div class="d-flex flex-row gap-3">
        <div class="d-flex flex-column w-100">
            <form action="hiking.php" method="POST" class="d-
flex flex-row w-100">
                <div class="form-floating mb-3 flex-grow-1">
                    <input type="text" class="form-control"
id="search_name" name="search_name" placeholder="Пошук по назві,
керівнику" value="<?= $_POST['search_name']?>">
                    <label for="search_name">Пошук по назві,
керівнику</label>
                </div>
                <button style="height: 58px; width: 58px;"
class="btn btn-outline-warning  d-flex justify-content-center
align-items-center" type="button" id="btn-filters" data-bs-
toggle="collapse" data-bs-target="#collapseExample" aria-
expanded="false" aria-controls="collapseExample">
                    <i class="fa-solid fa-filter"></i>
                </button>
                <?php if($_POST['search_name']): ?>
                    <a style="height: 58px; width: 58px;"
href="hiking.php" class="btn btn-outline-info d-flex justify-
content-center align-items-center">
                        <script
src="https://cdn.lordicon.com/qjzruarw.js"></script>
                        <lord-icon

src="https://cdn.lordicon.com/akuwjdzh.json"
                                trigger="hover"
                                style="width:25px;height:25px">
                        </lord-icon>
                    </a>
                <?php endif; ?>
```

```
                <button type="submit" style="height: 58px;
width: 58px;" class="btn btn-outline-info"><i class="fa-solid
fa-magnifying-glass"></i></button>
                <a style="height: 58px; width: 58px; margin-
left: 15px;" href="hikingaddedit.php?mode=adding" class="btn
btn-outline-success d-flex justify-content-center align-items-
center"><i class="fa-solid fa-plus"></i></a>
            </form>
            <div class="collapse" id="collapseExample">
                <div class="card card-body">
                    <form action="hiking.php" method="post"
id="search_filters" name="search_filters">
                        <div class="filters d-flex flex-row
justify-content-between">
                            <div class="filters-second d-flex
flex-row gap-3 align-items-center">
                                <div class="card card-body d-
flex flex-row gap-3 date-filter">
                                    <div class="form-floating ">
                                        <input type="date"
class="form-control" id="search_dateStart"
name="search_dateStart" placeholder="00.00.0000" value="<?=
$_POST['search_dateStart'];?>">
                                        <label
for="search_dateStart">Дата початку</label>
                                    </div>
                                    <div class="form-floating ">
                                        <input type="date"
class="form-control" id="search_dateEnd" name="search_dateEnd"
placeholder="00.00.0000" value="<?=
$_POST['search_dateEnd'];?>">
                                        <label
for="search_dateEnd">Дата кінця</label>
                                    </div>
                                </div>
                                <div class="count-filters-my d-
flex flex-row gap-3 align-items-center">
                                    <div class="card card-body
d-flex flex-row">
                                        <div class="form-
floating">
                                            <select class="form-
select" id="search_modeCount" name="search_modeCount"
style="min-width: 100px" aria-label="Floating label select
example">
                                                <option <?=
($_POST['search_modeCount']=="=")?'selected':''; ?>
value="=">=</option>
                                                <option <?=
($_POST['search_modeCount']==">")?'selected':''; ?>
value=">">&#62;</option>
                                                <option <?=
($_POST['search_modeCount']==">=")?'selected':''; ?>
```

```
value=">=">&#8805;</option>
                                                    <option <?=
($_POST['search_modeCount']=="<=")?'selected':''; ?>
value="<=">&#8804;</option>
                                                    <option <?=
($_POST['search_modeCount']=="<")?'selected':''; ?>
value="<">&#60;</option>
                                                </select>
                                                <label
for="search_modeCount">Режим</label>
                                            </div>
                                            <div class="form-
floating ">
                                                <input type="number"
class="form-control" id="search_minCount" name="search_minCount"
placeholder="Пошук по назві, керівнику" value="<?=
$_POST['search_minCount']?>">
                                                <label
for="search_minCount">Кількість учасників</label>
                                            </div>
                                        </div>
                                        <div class="form-check form-
switch">
                                            <input class="form-
check-input" type="checkbox" role="switch"
id="search_checkOnlyMy" name="search_checkOnlyMy" <?=
($_POST['search_checkOnlyMy'])?'checked':''; ?>>
                                            <label class="form-
check-label" for="search_checkOnlyMy">Тільки мої походи</label>
                                        </div>
                                    </div>
                                </div>
                                <div class="d-flex align-items-
center justify-content-center"><button type="submit" class="btn
btn-outline-info">Застосувати фільтри</button></div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
        <div class="content-table">
            <?php
            //отримання даних
            $travels = $db->getTravelsWithSumNut();
            $ar = array_filter($travels, function ($val){
                return $val->getUserId() ==
$_SESSION['user']['uid'];
            });
            if ($_POST['search_name']) $travels = $db-
>searchTravel($_POST['search_name']);
            if ($_POST['search_dateStart'] ||
$_POST['search_dateEnd'] || $_POST['search_minCount'] ||
```

```
$_POST['search_checkOnlyMy']) $travels = $db-
>searchTravelFilters($_POST['search_dateStart'],
$_POST['search_dateEnd'], $_POST['search_minCount'],
($_POST['search_checkOnlyMy']=='on')?'1':'',$_SESSION['user']['u
id'], $_POST['search_modeCount']);
        ?>
        <table class="table table-hover">
            <thead>
            <tr>
                <td>Назва</td>
                <td>Статус</td>
                <td class="overflow-table">Кількість
учасників</td>
                <td>Керівник</td>
                <td class="overflow-table">Дата початку</td>
                <td class="overflow-table">Дата кінця</td>
                <td class="overflow-table">Сумарна вага
розкладки</td>
                <?php if ($_SESSION['user']!=null && !empty($ar)
|| $_SESSION['user']['access_level'] == 'A'): ?>
                    <td class="text-center">Налаштування</td>
                <?php endif;?>
            </tr>
            </thead>
            <tbody>
            <!-- формування таблиці походів -->
            <?php foreach ($travels as &$travel):
                $status = "";
                if (new DateTime($travel->getStartDate()) > new
DateTime()){
                    $status = '<span class="badge bg-info text-
bg-info">Підготовка</span>';
                }else
                if (new DateTime($travel->getEndDate()) < new
DateTime()){
                    $status = '<span class="badge bg-secondary
text-bg-secondary">Пройдено</span>';
                }else{
                    $status = '<span class="badge bg-success
text-bg-success">Триває</span>';
                }
                ?>
                <?php if ($travel->getPublic()==1 ||
$_SESSION['user']['access_level']=='A' || $travel->getUserId()
== $_SESSION['user']['uid']): ?>
                    <tr
onclick="redirect('<?='food.php?hid='.$travel->getId();?>')"
style="cursor: pointer;">
                        <td><div class="text-ellipsis"><?=
$travel->getName(); ?></div></td>
                        <td><?= $status; ?></td>
                        <td class="overflow-table"><?= $travel-
>getCountParticipants(); ?></td>
```

```
                        <td><?= $db->getUserById($travel-
>getUserId())->getUsername(); ?></td>
                        <td class="overflow-table"><?php
$dateStart = new DateTime($travel->getStartDate()); echo
$dateStart->format('d.m.Y');?></td>
                        <td class="overflow-table"><?php
$dateStart = new DateTime($travel->getEndDate()); echo
$dateStart->format('d.m.Y');?></td>
                        <td class="overflow-table"><?= $travel-
>getSumNut(); ?></td>
                        <?php if
($_SESSION['user']['access_level'] == 'A' || $travel-
>getUserId() == $_SESSION['user']['uid']): ?>
                            <td class="text-center">
                                <a class="btn btn-outline-info"
href="hikingaddedit.php?mode=editing&hid=<?php echo $travel-
>getId(); ?>"><i class="fa-solid fa-pen-to-square"></i></a>
                                <a class="btn btn-outline-
danger" href="hiking.php?del=true&hid=<?php echo $travel-
>getId(); ?>"><i class="fa-regular fa-trash-can"></i></a>
                            </td>
                        <?php else: ?>
                            <?php if ($_SESSION['user']!=null &&
!empty($ar)): ?>
                                <td></td>
                            <?php endif;?>
                        <?php endif;?>
                    </tr>
                <?php endif;?>
            <?php endforeach;?>
            </tbody>
        </table>

    </div>
</div>

<?php
Partials::Footer();
/* видалення походу */
if ($_GET['del']=='true' && $_GET['hid']){
    if ($db->getTravel($_GET['hid']) != null)
        Partials::DeleteDialog('Видалення', 'Ви впевнені, що
бажаєте видалити похід - '.$db->getTravel($_GET['hid'])-
>getName(),
'assets/php/api.php?hiking_del=true&hid='.$_GET['hid']);
}
/* перевірки на фільтри */
if ($_POST['search_dateStart'] || $_POST['search_dateEnd'] ||
$_POST['search_minCount'] || $_POST['search_checkOnlyMy'])
Partials::SimClick("btn-filters");
?>
</body>
</html>
```

Info

```
<?php
require 'assets/php/Partials.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php Partials::Head('Довідка');  ?>
</head>
<body>
<?php
Partials::NavBar('info', $_SESSION['user']);
?>

<div class="main-container container content gap-3 d-flex flex-
column">
    <div>
        <h1>Інструкція користувача</h1>
        <p style="font-size: 120%; letter-spacing: 1px; line-
height: 180%;">
            Для перегляду довідки про даний сайт необхідно
обрати пункт меню «Довідка»
            <br><br>Також у режимі незареєстрованого користувача
доступний перегляд сторінки керівників подорожей, для цього
необхідно обрати відповідний пункт меню.
            <br><br>Для перегляду наявних подорожей можна обрати
відповідний пункт меню, або натиснути кнопку на головній
сторінці, або на вкладці «Керівники» натиснути на ім’я
керівника, чиї подорожі ви бажаєте переглянути.
            <br><br>На вкладці «Походи» реалізовано пошук
необхідних походів за кількістю днів, початковою та кінцевою
датами подорожей. Для здійснення пошуку необхідно обрати бажані
дати у відповідних полях та ввести мінімальну кількість днів
подорожі, після чого натиснути кнопку «Здійснити пошук». Для
виведення інформації про всі доступні подорожі, необхідно
натиснути кнопку «Всі подорожі».
            <br><br>Для перегляду розкладки подорожі, яка вас
зацікавила, необхідно натиснути на назву подорожі, після чого
відкриється вкладка розкладки подорожі.
            <br><br>Зареєстрований користувач має більше
можливостей. Для входу в обліковий запис необхідно ввести ім'я
користувача та пароль у відповідній формі. Якщо у вас ще немає
облікового запису, ви маєте змогу зареєструватися, натиснувши на
відповідний надпис на формі авторизації. Для здійснення виходу з
облікового запису необхідно натиснути кнопку «Вихід».
            <br><br>Для зареєстрованого користувача поступні
функції створення, редагування та видалення подорожей, додавання
та видалення до харчової розкладки харчових продуктів,
редагування ваги взятих продуктів, а також перегляд інформації
щодо звіту для закупки продуктів. Зауважимо, що все вище
```

```
перераховане доступно лише для власноруч створених подорожей, та
не розповсюджується на подорожі створені іншими користувачами.
        </p>
        <?php if ($_SESSION['user']['access_level']== 'A'){?>
        <p style="font-size: 120%; letter-spacing: 1px; line-
height: 180%;">
            Для адміністратора передбачені такі додаткові
функції як: перегляд повних даних про зареєстрованих
користувачів, модифікація та видалення даних про них (рис. 37)
та доступ до вкладки з продуктами, на якій своєю чергою
адміністратору доступне додавання, редагування та видалення
харчових продуктів (рис. 36). Також на вкладці «Продукти»
реалізовано пошук по назві та по категорії продукту, для
використання якого необхідно заповнити відповідні поля та
натиснути кнопку «Здійснити пошук».
        </p><?php }?>
    </div>
</div>

<?php
Partials::Footer();
?>
</body>
</html>
```

Members

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
    /* встановлення зв'язку з бд */
    $db = new ClubDB();
    Partials::Head('Керівники');
    /* якщо сторінка в режимі редагування отримуємо дані
користувача для редагування */
    if ($_GET['mode']=='editing' && $_GET['uid']){
        $userEdit = $db->getUserById($_GET['uid']);
        $_GET['user_id'] = $userEdit->getId();
        $_GET['user_username'] = $userEdit->getUsername();
        $_GET['user_access_level'] = $userEdit-
>getAccessLevel();
        $_GET['user_phone'] = $userEdit->getPhone();
    }

    ?>

    <script>
```

```
        /* метод перемикання режиму зміни пароля */
        function changePass(){
            if
(document.getElementById('checkBoxChangePassword').checked) {
                document.getElementById('password').value = '';
                document.getElementById('confirm-
password').value = '';
                document.getElementById('div-
password').classList.remove('hidden')
                document.getElementById('div-confirm-
password').classList.remove('hidden')
            } else {
                document.getElementById('password').value = '';
                document.getElementById('confirm-
password').value = '';
                document.getElementById('div-
password').classList.add('hidden')
                document.getElementById('div-confirm-
password').classList.add('hidden')
            }
        }

    </script>
</head>
<body>
<?php
Partials::NavBar('members', $_SESSION['user']);
?>
<div class="main-container container content gap-3 d-flex flex-
column pt-2 pb-2 ">
    <div class="d-flex flex-row gap-3">
        <form action="members.php" method="POST" class="d-flex
flex-row w-100">
            <div class="form-floating mb-3 flex-grow-1">
                <input type="text" class="form-control"
id="search_username" name="search_username" placeholder="Пошук
по назві, категорії" value="<?= $_POST['search_username']?>">
                <label for="search_username">Пошук за іменем
користувача</label>
            </div>
            <?php if($_POST['search_username']): ?>
                <a style="height: 58px; width: 58px;"
href="members.php" class="btn btn-outline-info d-flex justify-
content-center align-items-center">
                    <script
src="https://cdn.lordicon.com/qjzruarw.js"></script>
                    <lord-icon

src="https://cdn.lordicon.com/akuwjdzh.json"
                            trigger="hover"
                            style="width:25px;height:25px">
                    </lord-icon>
                </a>
```

```
            <?php endif; ?>
            <button type="submit" style="height: 58px; width:
58px;" class="btn btn-outline-info"><i class="fa-solid fa-
magnifying-glass"></i></button>
        </form>
        <!-- якщо авторизован адмін виводимо кнопки керування --
>
        <?php if ($_SESSION['user']['access_level']=='A'):?>
            <a style="height: 58px; width: 58px;"
href="members.php?mode=adding" class="btn btn-outline-success d-
flex justify-content-center align-items-center"><i class="fa-
solid fa-plus"></i></a>
        <?php endif; ?>
    </div>
    <div style="max-height: calc(100vh - 250px); overflow:
auto;">
        <table class="table table-hover">
            <thead>
            <tr>
                <td>Керівники</td>
                <!-- якщо авторизован адмін виводимо кнопки
керування -->
                <?php if ($_SESSION['user']['access_level'] ==
'A'): ?>
                    <td>Рівень доступу</td>
                    <td class="text-center">Налаштування</td>
                <?php endif; ?>
            </tr>
            </thead>
            <tbody>
            <!-- формування таблиці користувачів -->
            <?php foreach (($_POST['search_username'])?$db-
>searchUser($_POST['search_username']):$db->getUsers() as
&$user):?>
                <tr>
                    <td><a class="d-block"
href="hiking.php?member_search=<?=$user->getUserName();?>"><?=
$user->getUserName(); ?></a></td>
                    <?php if ($_SESSION['user']['access_level']
== 'A'): ?>
                        <td><?= $user->getAccessLevel(); ?></td>
                        <td class="text-center">
                            <a class="btn btn-outline-info"
href="members.php?mode=editing&uid=<?php echo $user->getId();
?>"><i class="fa-solid fa-pen-to-square"></i></a>
                            <a class="btn btn-outline-danger"
href="members.php?del=true&uid=<?php echo $user->getId(); ?>"><i
class="fa-regular fa-trash-can"></i></a>
                        </td>
                    <?php endif; ?>
                </tr>
            <?php endforeach;?>
            </tbody>
```

```
        </table>

    </div>
</div>
<script>
    function isValidPhoneNumber(phoneNumber) {
        return /^\+\d{1,3}\d{9}$/.test(phoneNumber);
    }
    /* метод перевірки вхідних даних */
    function checkDataAndSubmit(){
        document.getElementById("login").classList.remove("is-
invalid")

document.getElementById("password").classList.remove("is-
invalid")
        document.getElementById("confirm-
password").classList.remove("is-invalid")
        document.getElementById("phone").classList.remove("is-
invalid")
        document.getElementById("login-label").innerHTML =
'Логін'
        document.getElementById("login-password").innerHTML =
'Пароль'
        document.getElementById("phone-label").innerHTML =
'Телефон'
        document.getElementById("login-confirm-
password").innerHTML = 'Підтвердить пароль'

        if (document.getElementById("login").value.trim().length
=== 0){
            document.getElementById("login").classList.add("is-
invalid")
            return;
        }
        if (document.getElementById("login").value.trim().length
> 50){
            document.getElementById("login").classList.add("is-
invalid")
            return;
        }
        if (document.getElementById("phone").value.trim().length
=== 0){
            document.getElementById("phone").classList.add("is-
invalid")
            return;
        }
        if (document.getElementById("phone").value.trim().length
> 14){
            document.getElementById("phone").classList.add("is-
invalid")
            return;
        }
        if
```

```
(!isValidPhoneNumber(document.getElementById("phone").value.trim
())){
            document.getElementById("phone").classList.add("is-
invalid")
            return;
        }
        fetch(`assets/php/api.php`,
            {
                method: "POST",
                headers:{
                    "Content-Type":"application/x-www-form-
urlencoded",
                },
                body: new URLSearchParams({
                    checkLogin:
document.getElementById("login").value.trim(),
                    checkPhone:
document.getElementById("phone").value.trim(),
                    userId:
document.getElementById("user_id").value
                })
            }
        ).then(value => {
            value.json().then(response=>{
                if (response.status === 200) {
                    console.log(response);

                    if (!response.loginFree){

document.getElementById("login").classList.add("is-invalid")
                        document.getElementById("login-
label").innerHTML = 'Такий логін вже зайнятий!'
                        return;
                    }
                    if (!response.phoneFree){

document.getElementById("phone").classList.add("is-invalid")
                        document.getElementById("phone-
label").innerHTML = 'Такий телефон вже зайнятий!'
                        return;
                    }
                    if
(document.getElementById('checkBoxChangePassword').checked){
                        if
(document.getElementById("password").value.trim().length === 0){

document.getElementById("password").classList.add("is-invalid")
                            return;
                        }
                        if (document.getElementById("confirm-
password").value.trim().length === 0){
                            document.getElementById("confirm-
password").classList.add("is-invalid")
```

```
                            return;
                        }
                        if (document.getElementById("confirm-
password").value.trim() !==
document.getElementById("password").value.trim()){

document.getElementById("password").classList.add("is-invalid")
                            document.getElementById("confirm-
password").classList.add("is-invalid")

                            document.getElementById("login-
password").innerHTML = 'Паролі не співпадають!'
                            document.getElementById("login-
confirm-password").innerHTML = 'Паролі не співпадають!'
                            return;
                        }
                    }

document.getElementById("modalAdd").submit();
                }
            })
        })
    }
</script>
<button style="height: 58px; width: 58px;" type="button"
id="btnOpenModal" class="btn btn-outline-success" data-bs-
toggle="modal" data-bs-target="#addModal" hidden></button>
<div class="modal fade" id="addModal" tabindex="-1" data-bs-
backdrop="static" data-bs-keyboard="false" aria-
labelledby="addModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel"><?= ($_GET['mode']=='adding')?'Додавання
нового':'Редагування' ?> керівника</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="assets/php/api.php" method="post"
id="modalAdd"  class="modal-body">
                <input type="text" name="user_mode" value="<?=
($_GET['mode']=='adding')?'user_add':'user_edit' ?>" hidden
readonly>
                <input type="text" name="user_id" id="user_id"
value="<?= $_GET['user_id']?>" hidden readonly>
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="login" name="login" required placeholder="Kva"
value="<?=$_GET['user_username']?>">
                    <label for="login" id="login-
label">Логін</label>
                </div>
```

```
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="phone" name="phone" required placeholder="Kva"
value="<?=$_GET['user_phone']?>">
                    <label for="phone" id="phone-
label">Телефон</label>
                </div>
                <div class="form-floating mb-3">
                    <select class="form-select"
id="user_access_level" name="access_level" aria-label="Floating
label select example">
                        <option value="U" <?=
($_GET['user_access_level'] == 'U')?'selected':''?>>Користувач
(U)</option>
                        <option value="A" <?=
($_GET['user_access_level'] ==
'A')?'selected':''?>>Адміністратор (A)</option>
                        <option value="B" <?=
($_GET['user_access_level'] == 'B')?'selected':''?>>Заблокований
користувач (B)</option>
                    </select>
                    <label for="user_access_level">Рівень
доступу</label>
                </div>
                <div class="form-check form-switch mb-3">
                    <input class="form-check-input"
type="checkbox" role="switch" id="checkBoxChangePassword"
onchange="changePass()">
                    <label class="form-check-label"
for="checkBoxChangePassword">Змінити пароль?</label>
                </div>
                <div class="form-floating mb-3 hidden" id="div-
password">
                    <input type="password" class="form-control"
id="password" name="password" required placeholder="*******">
                    <label for="password" id="login-
password">Пароль</label>
                </div>
                <div class="form-floating mb-3 hidden" id="div-
confirm-password">
                    <input type="password" class="form-control"
id="confirm-password" name="password" required
placeholder="*******">
                    <label for="confirm-password" id="login-
confirm-password">Підтвердить пароль</label>
                </div>
            </form>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
onclick="checkDataAndSubmit()"><?=
($_GET['mode']=='adding')?'Додати':'Редагувати' ?></button>
```

```
            </div>
        </div>
    </div>
</div>
<?php
Partials::Footer();
/* редагування користувача */
if (($_GET['mode']=='editing' && $_GET['uid']) ||
$_GET['mode']=='adding'){
    Partials::SimClick('btnOpenModal');
}
/* видалення користувача */
if ($_GET['del']=='true' && $_GET['uid']){
    if ($db->getUserById($_GET['uid']) != null)
        Partials::DeleteDialog('Видалення', 'Ви впевнені, що
бажаєте видалити керівника - '.$db->getUserById($_GET['uid'])-
>getUsername().'? Видаляться усі його походи!',
'assets/php/api.php?user_del=true&uid='.$_GET['uid']);
}
?>

</body>
</html>
```

Product

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
session_start();
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
    /* встановлення зв'язку з бд */
        $db = new ClubDB();
        Partials::Head('Продукти');
        /* якщо сторінка в режимі редагування отримуємо дані
продукту для редагування */
        if ($_GET['mode']=='editing' && $_GET['pid']){
            $productEdit = $db->getProduct($_GET['pid']);
            $_GET['product_id'] = $productEdit->getId();
            $_GET['product_name'] = $productEdit->getName();
            $_GET['product_caloric'] = $productEdit-
>getCalories();
            $_GET['product_weight'] = $productEdit-
>getDefaultPortion();
            $_GET['product_category'] = $productEdit-
>getCategory()->getId();
        }
        if ($_GET['category_search'] && $db-
>getCategory($_GET['category_search'])!=null){
```

```
            $_POST['search_category'] =
$_GET['category_search'];
        }
    ?>
</head>
<body>
<?php
Partials::NavBar('products', $_SESSION['user']);
?>
<div class="main-container container content gap-3 d-flex flex-
column pt-2 pb-2 ">
    <div class="d-flex flex-row gap-3">
        <form action="products.php" method="POST" class="d-flex
flex-row w-100">
            <div class="form-floating mb-3 flex-grow-1">
                <input type="text" class="form-control"
id="floatingInput" name="search_name" placeholder="Пошук по
назві, категорії" value="<?= $_POST['search_name']?>">
                <label for="floatingInput">Пошук по
назві</label>
            </div>
            <div class="form-floating">
                <select class="form-select" id="floatingSelect"
name="search_category" aria-label="Floating label select
example">
                    <option value="-1">Оберіть
категорію</option>
                    <!-- заповнення випадаючого списку
категоріями-->
                    <?php foreach ($db->getCategories() as
&$category):?>
                        <option <?=
($_POST['search_category']==$category->getId())?'selected':'' ?>
value="<?= $category->getId(); ?>"><?= $category->getname();
?></option>
                    <?php endforeach;?>
                </select>
                <label for="floatingSelect">Пошук по
категорії</label>
            </div>
            <?php if($_POST['search_name'] ||
$_POST['search_category']): ?>
                <a style="height: 58px; width: 58px;"
href="products.php" class="btn btn-outline-info d-flex justify-
content-center align-items-center">
                    <script
src="https://cdn.lordicon.com/qjzruarw.js"></script>
                    <lord-icon

src="https://cdn.lordicon.com/akuwjdzh.json"
                            trigger="hover"
                            style="width:25px;height:25px">
                    </lord-icon>
```

```
                </a>
            <?php endif; ?>
            <button type="submit" style="height: 58px; width:
58px;" class="btn btn-outline-info"><i class="fa-solid fa-
magnifying-glass"></i></button>
        </form>
        <a style="height: 58px; width: 58px;"
href="products.php?mode=adding" class="btn btn-outline-success
d-flex justify-content-center align-items-center"><i class="fa-
solid fa-plus"></i></a>
    </div>
    <div style="max-height: calc(100vh - 250px); overflow:
auto;">
        <table class="table table-hover">
            <thead>
                <tr>
                    <td>Назва</td>
                    <td>Калорій на 100 г.</td>
                    <td>Стандартна порція(г)</td>
                    <td>Категорія</td>
                    <!-- якщо авторизован адмін виводимо кнопки
керування -->
                    <?php if ($_SESSION['user']['access_level']
== 'A'): ?>
                        <td class="text-
center">Налаштування</td>
                    <?php endif; ?>
                </tr>
            </thead>
            <tbody>
            <!-- формуємо список продуктів -->
            <?php foreach (($_POST['search_name'] ||
$_POST['search_category'])?$db-
>searchProduct($_POST['search_name'],$_POST['search_category']):
$db->getProducts() as &$product):?>
                <tr>
                    <td><?= $product->getName(); ?></td>
                    <td><?= $product->getCalories(); ?></td>
                    <td><?= $product->getDefaultPortion();
?></td>
                    <td><?= $product->getCategory()->getName();
?></td>
                    <?php if ($_SESSION['user']['access_level']
== 'A'): ?>
                        <td class="text-center">
                            <a class="btn btn-outline-info"
href="products.php?mode=editing&pid=<?php echo $product-
>getId(); ?>"><i class="fa-solid fa-pen-to-square"></i></a>
                            <a class="btn btn-outline-danger"
href="products.php?del=true&pid=<?php echo $product->getId();
?>"><i class="fa-regular fa-trash-can"></i></a>
                        </td>
                    <?php endif; ?>
```

```
                </tr>
            <?php endforeach;?>
            </tbody>
        </table>

    </div>
</div>
<script>
    /* метод перевірки вхідних даних */
    function checkDataAndSubmit(){

document.getElementById("product_name").classList.remove("is-
invalid")

document.getElementById("product_caloric").classList.remove("is-
invalid")

document.getElementById("product_weight").classList.remove("is-
invalid")

        if
(document.getElementById("product_name").value.trim().length ===
0){

document.getElementById("product_name").classList.add("is-
invalid")
            return;
        }
        if
(document.getElementById("product_name").value.trim().length >
50){

document.getElementById("product_name").classList.add("is-
invalid")
            return;
        }
        if
(document.getElementById("product_caloric").value.trim().length
=== 0){

document.getElementById("product_caloric").classList.add("is-
invalid")
            return;
        }
        if
(document.getElementById("product_weight").value.trim().length
=== 0){

document.getElementById("product_weight").classList.add("is-
invalid")
            return;
        }
        document.getElementById("modalAdd").submit();
```

```
    }
</script>
<button style="height: 58px; width: 58px;" type="button"
id="btnOpenModal" class="btn btn-outline-success" data-bs-
toggle="modal" data-bs-target="#addModal" hidden></button>
<div class="modal fade" id="addModal" tabindex="-1" data-bs-
backdrop="static" data-bs-keyboard="false" aria-
labelledby="addModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel"><?= ($_GET['mode']=='adding')?'Додавання
нового':'Редагування' ?> продукту</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="assets/php/api.php" name="product_add"
method="post" id="modalAdd"  class="modal-body">
                <input type="text" name="product_mode"
value="<?=
($_GET['mode']=='adding')?'product_add':'product_edit' ?>"
hidden readonly>
                <input type="text" name="product_id" value="<?=
$_GET['product_id']?>" hidden readonly>
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="product_name" name="product_name" placeholder="Масло Ферма
73%" value="<?= $_GET['product_name']?>">
                    <label for="product_name">Назва
продукту</label>
                </div>
                <div class="form-floating mb-3">
                    <input type="number" class="form-control"
id="product_caloric" name="product_caloric" placeholder="50"
value="<?= $_GET['product_caloric']?>">
                    <label for="product_caloric">Калорійність на
100 г.</label>
                </div>
                <div class="form-floating mb-3">
                    <input type="number" class="form-control"
id="product_weight" name="product_weight" placeholder="200"
value="<?= $_GET['product_weight']?>">
                    <label for="product_weight">Вага стандартної
порції</label>
                </div>
                <div class="form-floating">
                    <select class="form-select"
id="product_category" name="product_category" aria-
label="Floating label select example">
                        <!-- заповнення випадаючого списку
категоріями-->
                        <?php foreach ($db->getCategories() as
```

```
&$category):?>
                                <option <?=
($_GET['product_category']==$category->getId())?'selected':'' ?>
value="<?= $category->getId(); ?>"><?= $category->getname();
?></option>
                            <?php endforeach;?>
                        </select>
                        <label for="product_category">Оберіть
категорію</label>
                    </div>
                </form>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Скасувати</button>
                    <button type="button" class="btn btn-primary"
onclick="checkDataAndSubmit()"><?=
($_GET['mode']=='adding')?'Додати':'Редагувати' ?></button>
                </div>
            </div>
        </div>
    </div>
    <?php
    Partials::Footer();
    /* редагування продукту */
    if ($_GET['mode']=='editing' && $_GET['pid']){
        Partials::SimClick('btnOpenModal');
    }
    /* додавання продукту */
    if ($_GET['mode']=='adding'){
        Partials::SimClick('btnOpenModal');
    }
    /* видалення продукту */
    if ($_GET['del']=='true' && $_GET['pid']){
        if ($db->getProduct($_GET['pid']) != null)
        Partials::DeleteDialog('Видалення', 'Ви впевнені, що бажаєте
видалити продукт - '.$db->getProduct($_GET['pid'])->getName(),
'assets/php/api.php?product_del=true&pid='.$_GET['pid']);
    }
    ?>
    </body>
    </html>
```

User

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/Helper.php';
require 'assets/php/ModelDB.php';
require 'assets/php/functions.php';
session_start()
?>
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<?php
    Partials::Head('ЄНОТИ | Профіль користувача');
?>
    <style>
        hr{
            margin-top: 1rem;
            margin-bottom: 1rem;
            border: 0;
            border-top: 1px solid rgba(0,0,0,.1);
        }
        h4 {
            font-size: 1.5rem;
        }
    </style>
    <script>
        function isValidPhoneNumber(phoneNumber) {
            return /^\+\d{1,3}\d{9}$/.test(phoneNumber);
        }
        /* метод перемикання режиму зміни пароля */
        function changePass(){
            if
(document.getElementById('checkBoxChangePassword').checked) {
                document.getElementById('password').value = '';
                document.getElementById('confirm-
password').value = '';
                document.getElementById('div-
password').classList.remove('hidden')
                document.getElementById('div-confirm-
password').classList.remove('hidden')
            } else {
                document.getElementById('password').value = '';
                document.getElementById('confirm-
password').value = '';
                document.getElementById('div-
password').classList.add('hidden')
                document.getElementById('div-confirm-
password').classList.add('hidden')
            }
        }

    </script>
    <script>
        /* метод перевірки вхідних даних */
        function checkDataAndSubmit(){

document.getElementById("login").classList.remove("is-invalid")

document.getElementById("password").classList.remove("is-
invalid")
            document.getElementById("confirm-
password").classList.remove("is-invalid")

document.getElementById("phone").classList.remove("is-invalid")
```

```
            document.getElementById("login-label").innerHTML =
'Логін'
            document.getElementById("login-password").innerHTML
= 'Пароль'
            document.getElementById("phone-label").innerHTML =
'Телефон'
            document.getElementById("login-confirm-
password").innerHTML = 'Підтвердить пароль'

            if
(document.getElementById("login").value.trim().length === 0){

document.getElementById("login").classList.add("is-invalid")
                return;
            }
            if
(document.getElementById("login").value.trim().length > 50){

document.getElementById("login").classList.add("is-invalid")
                return;
            }
            if
(document.getElementById("phone").value.trim().length === 0){

document.getElementById("phone").classList.add("is-invalid")
                return;
            }
            if
(document.getElementById("phone").value.trim().length > 14){

document.getElementById("phone").classList.add("is-invalid")
                return;
            }
            if
(!isValidPhoneNumber(document.getElementById("phone").value.trim
())){

document.getElementById("phone").classList.add("is-invalid")
                return;
            }

            fetch(`assets/php/api.php`,
                {
                    method: "POST",
                    headers:{
                        "Content-Type":"application/x-www-form-
urlencoded",
                    },
                    body: new URLSearchParams({
                        checkLogin:
document.getElementById("login").value.trim(),
                        checkPhone:
document.getElementById("phone").value.trim(),
```

```
                        userId:
document.getElementById("user_id").value
                    })
                }
            ).then(value => {
                value.json().then(response=>{
                    if (response.status === 200) {
                        console.log(response);

                        if (!response.loginFree){

document.getElementById("login").classList.add("is-invalid")
                            document.getElementById("login-
label").innerHTML = 'Такий логін вже зайнятий!'
                            return;
                        }
                        if (!response.phoneFree){

document.getElementById("phone").classList.add("is-invalid")
                            document.getElementById("phone-
label").innerHTML = 'Такий телефон вже зайнятий!'
                            return;
                        }

                        if
(document.getElementById('checkBoxChangePassword').checked){
                            if
(document.getElementById("password").value.trim().length === 0){

document.getElementById("password").classList.add("is-invalid")
                                return;
                            }
                            if
(document.getElementById("confirm-password").value.trim().length
=== 0){

document.getElementById("confirm-password").classList.add("is-
invalid")
                                return;
                            }
                            if
(document.getElementById("confirm-password").value.trim() !==
document.getElementById("password").value.trim()){

document.getElementById("password").classList.add("is-invalid")

document.getElementById("confirm-password").classList.add("is-
invalid")

                                document.getElementById("login-
password").innerHTML = 'Паролі не співпадають!'
                                document.getElementById("login-
confirm-password").innerHTML = 'Паролі не співпадають!'
```

```
                                return;
                            }
                        }

document.getElementById("modalAdd").submit();
                    }
                })
            })
        }
    </script>
</head>
<body>
<?php Partials::NavBar('user', $_SESSION['user']); ?>
<?php
    $db = new ClubDB();

    $user = $db->getUserById($_SESSION['user']['uid']);
    if ($user==null) echo '<script>window.location.href =
"assets/php/logout.php";</script>';

    /* якщо сторінка в режимі редагування отримуємо дані
користувача для редагування */

    $userEdit = $db->getUserById($_SESSION['user']['uid']);
    $_GET['user_id'] = $userEdit->getId();
    $_GET['user_username'] = $userEdit->getUsername();
    $_GET['user_access_level'] = $userEdit->getAccessLevel();
    $_GET['user_phone'] = $userEdit->getPhone();

?>
<div class="main-container container content gap-3 d-flex flex-
column" style="margin-top: 2em;">
    <div class="main-body">
        <div class="row gutters-sm">
            <div class="col-md-4 mb-3">
                <div class="card">
                    <div class="card-body">
                        <div class="d-flex flex-column align-
items-center text-center">
                            <form id="avatar-form">
                                <input type="file" id="avatar"
name="avatar" hidden>
                            </form>
                            <button class="btn"
id="change_avatar">
                                <img src="users-avatars/<?=
$user->getAvatar() ?? '/enot.png'?>" alt="Admin" id="avatar-img"
class="rounded-circle" width="200">
                            </button>
                            <div class="mt-3">
                                <h4 class="mb-3"><?php echo
$user->getUsername(); ?></h4>
                                <a href="assets/php/logout.php"
```

```
class="btn btn-info">Вийти</a>
                                <a class="btn btn-danger"
href="user.php?del=true&uid=<?php echo $user->getId();
?>">Видалити акаунт</a>
                            </div>
                        </div>
                    </div>
                </div>

            </div>
            <div class="col-md-8">
                <div class="card mb-3">
                    <div class="card-body">
                        <div class="row">
                            <div class="col-sm-3">
                                <h6 class="mb-0">Ім'я
користувача:</h6>
                            </div>
                            <div class="col-sm-9 text-
secondary">
                                <?php
                                    echo $user->getUsername();
                                ?>
                            </div>
                        </div>
                        <hr>
                        <div class="row">
                            <div class="col-sm-3">
                                <h6 class="mb-0">Телефон:</h6>
                            </div>
                            <div class="col-sm-9 text-
secondary">
                                <?php
                                    echo $user->getPhone();
                                ?>
                            </div>
                        </div>
                        <hr>
                        <div class="row">
                            <div class="col-sm-12">
                                <a class="btn btn-info "
href="user.php?mode=editing&uid=<?php echo $user->getId();
?>">Редагувати</a>
                                <button hidden type="button"
id="btn-add" onclick="document.getElementById('alert-
add').classList.add('hidden')" class="btn btn-light" data-bs-
toggle="modal" data-bs-target="#modal-add">
                                    <i class="fa-solid fa-
plus"></i>
                                </button>
                            </div>
                        </div>
                    </div>
```

```
                </div>
            </div>
        </div>

    </div>
</div>
<button style="height: 58px; width: 58px;" type="button"
id="btnOpenModal" class="btn btn-outline-success" data-bs-
toggle="modal" data-bs-target="#addModal" hidden></button>

<div class="modal fade" id="addModal" tabindex="-1" data-bs-
backdrop="static" data-bs-keyboard="false" aria-
labelledby="addModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5"
id="exampleModalLabel">Редагування</h1>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="assets/php/api.php" method="post"
id="modalAdd"  class="modal-body">
                <input type="text" name="user_mode"
value="user_edit" hidden readonly>
                <input type="text" name="user_id" id="user_id"
value="<?= $_GET['user_id']?>" hidden readonly>
                <input type="hidden" name="access_level"
id="access_level" value="<?= $_GET['user_access_level']?>"
readonly>
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="login" name="login" required placeholder="Kva"
value="<?=$_GET['user_username']?>">
                    <label for="login" id="login-
label">Логін</label>
                </div>
                <div class="form-floating mb-3">
                    <input type="text" class="form-control"
id="phone" name="phone" required placeholder="Kva"
value="<?=$_GET['user_phone']?>">
                    <label for="phone" id="phone-
label">Телефон</label>
                </div>
                <div class="form-check form-switch mb-3">
                    <input class="form-check-input"
type="checkbox" role="switch" id="checkBoxChangePassword"
onchange="changePass()">
                    <label class="form-check-label"
for="checkBoxChangePassword">Змінити пароль?</label>
                </div>
                <div class="form-floating mb-3 hidden" id="div-
password">
```

```
                    <input type="password" class="form-control"
id="password" name="password" required placeholder="*******">
                    <label for="password" id="login-
password">Пароль</label>
                </div>
                <div class="form-floating mb-3 hidden" id="div-
confirm-password">
                    <input type="password" class="form-control"
id="confirm-password" name="password" required
placeholder="*******">
                    <label for="confirm-password" id="login-
confirm-password">Підтвердить пароль</label>
                </div>
            </form>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Скасувати</button>
                <button type="button" class="btn btn-primary"
onclick="checkDataAndSubmit()">Редагувати</button>
            </div>
        </div>
    </div>
</div>

<?php
    Partials::Footer();

    /* редагування користувача */
    if (($_GET['mode']=='editing' && $_GET['uid']) ||
$_GET['mode']=='adding'){
        Partials::SimClick('btnOpenModal');
    }
    /* видалення користувача */
    if ($_GET['del']=='true' && $_GET['uid']){
        if ($db->getUserById($_GET['uid']) != null)
            Partials::DeleteDialog('Видалення', 'Ви впевнені, що
бажаєте видалити керівника - '.$db->getUserById($_GET['uid'])-
>getUsername().'? Видаляться усі його походи!',
'assets/php/api.php?user_del=true&uid='.$_GET['uid']);
    }
?>
<script>
    document.getElementById('change_avatar').onclick = () => {
        document.getElementById('avatar').click()
    }

    document.getElementById('avatar').onchange = () => {
        let formData = new
FormData(document.getElementById('avatar-form'))
        formData.append('user-id',
document.getElementById('user_id').value)
        formData.append('update-avatar', 'update-avatar')
```

```
        fetch(`assets/php/api.php`,
            {
                method: "POST",
                body: formData
            }).then(value => {
            value.json().then(json=>{
                if (json.code === 200){
                    document.getElementById('avatar-img').src =
`users-avatars/${json.avatar}`;
                }
            })
        })
    }

</script>
</body>
</html>
```

Zvit

```
<?php
require 'assets/php/Partials.php';
require 'assets/php/ModelDB.php';
session_start();
//оголошення масиву місяців
$months = array(
    1 => 'Січня',
    2 => 'Лютого',
    3 => 'Березня',
    4 => 'Квітня',
    5 => 'Травня',
    6 => 'Червня',
    7 => 'Липня',
    8 => 'Серпня',
    9 => 'Вересня',
    10 => 'Жовтня',
    11 => 'Листопада',
    12 => 'Грудня'
);
?>

<!doctype html>
<html lang="uk">
<head>
    <?php
    /* встановлення зв'язку з бд */
    $db = new ClubDB();
    //отримання походів з бд
    $travel = $db->getTravelWithSumNut($_GET['hid']);
    Partials::Head('Звіт закупівель');
```

```
    ?>
    <style>
        .list > li:nth-child(even) {
            background-color: #f2f2f2;
        }
        .list > li:nth-child(odd) {
            background-color: #ffffff;
        }
    </style>
    <script>
        //функція копіювання тексту з textarea
        function copyText(id){
            const copyButton = document.querySelector('#'+id);
            const copyIcon = document.querySelector('#copy-icon');
            copyButton.select();
            document.execCommand('copy');

            // Change icon to 'clipboard-check'
            copyIcon.classList.remove('bi-clipboard');
            copyIcon.classList.add('bi-clipboard-check');

            // Reset icon to 'clipboard' after 2 seconds
            setTimeout(() => {
                copyIcon.classList.remove('bi-clipboard-check');
                copyIcon.classList.add('bi-clipboard');
            }, 2000);
        }

    </script>
</head>
<body>
<?php
Partials::NavBar('zvit', $_SESSION['user']);
?>
<div class="main-container container content gap-3 d-flex flex-column pt-2
pb-2 ">
    <input type="number" id="hid" value="<?= $_GET['hid']?>" hidden readonly>
    <input type="number" id="uid" value="<?= $_SESSION['user']['uid']?>"
hidden readonly>
    <nav>

        <div class="nav nav-tabs" id="nav-tab" role="tablist">
            <a class="btn btn-outline-info" href="food.php?hid=<?=
$_GET['hid']?>">← До планування</a>
            <button class="nav-link active" id="nav-home-tab" data-bs-
toggle="tab" data-bs-target="#nav-home" type="button" role="tab" aria-
```

```
controls="nav-home" aria-selected="true">Розкладка</button>
            <button class="nav-link" id="nav-profile-tab" data-bs-
toggle="tab" data-bs-target="#nav-profile" type="button" role="tab" aria-
controls="nav-profile" aria-selected="false">Текстовий
варіант</button>
            <button class="nav-link" id="nav-purchase-tab" data-bs-
toggle="tab" data-bs-target="#nav-purchase" type="button" role="tab" aria-
controls="nav-purchase" aria-selected="true">Закупівля</button>
        </div>
    </nav>
    <div class="tab-content" id="nav-tabContent">
        <div class="tab-pane fade show" id="nav-purchase" role="tabpanel"
aria-labelledby="nav-purchase-tab" style="background: none;">
            <div class="d-flex flex-column gap-2">
                <h1><?= $travel->getName();?></h1>
                <h6>(вся кількість та калорійність
розрахована з урахуванням кількості
учасників)</h6>
                <hr>
                <div class="d-flex flex-row justify-content-between ">
                    <h3>Загальна вага харчової
розкладки:</h3>
                    <div class="d-flex justify-content-center align-items-
center"><?= $travel->getSumNut();?> г</div>
                </div>
                <hr>
            </div>
            <?php
                //отримання закупівель
                $p = $db->getPurchase($_GET['hid']);
                $categories = $p["cat"];
                $purchases = $p['pur'];
            ?>
            <!-- формування звіту закупівель -->
            <?php foreach ($categories as $category){ ?>
            <h1><?= $category; ?></h1>
            <ul class="list-group list-group-flush">
            <?php foreach ($purchases[$category] as $product){?>
                <li class="list-group-item">
                    <div class="d-flex flex-row gap-2 align-items-center">
                        <input class="form-check-input purchase-check"
type="checkbox" purchase-name="<?=$product['pname']?>" <?= ($db-
>getPurchaseStatus($_SESSION['user']['uid'], $_GET['hid'],
$product['pname'])==1)?'checked':'' ?>>
                        <div class="flex-grow-1">
                            <?= $product['pname'];?>
```

```
                    </div>
                    <div>
                        <?= $product['weight'];?> г
                    </div>
                </div>
            </li>
        <?php } }?>
        </ul>
    </div>
    <div class="tab-pane fade show active" id="nav-home" role="tabpanel"
aria-labelledby="nav-home-tab" style="background: none;">
        <div class="d-flex flex-column gap-2">
            <h1><?= $travel->getName();?></h1>
            <h6>(вся кількість та калорійність
розрахована з урахуванням кількості
учасників)</h6>
            <hr>
            <div class="d-flex flex-row justify-content-between ">
                <h3>Загальна вага харчової
розкладки:</h3>
                <div class="d-flex justify-content-center align-items-
center"><?= $travel->getSumNut();?> г</div>
            </div>
            <hr>
        </div>
        <ul class="list card flex-grow-1 list-group list-group-flush
flex-grow-1 p-1">
        <?php
        $start_date = new DateTime($travel->getStartDate());
        $end_date = new DateTime($travel->getEndDate());
        $interval = new DateInterval('P1D');
        $date_range = new DatePeriod($start_date, $interval, $end_date-
>modify('+1 day'));
        $day_number = 1;
        $nutritious = $db->getTravelNutrition($travel->getId());
        $formatNutritious = [];
        foreach ($nutritious as $nutrition){

$formatNutritious[$nutrition['day']][$nutrition['menu_type']][] =
Partials::nutrition($nutrition);
        }
        /*формування звіту розкладки*/
        foreach ($date_range as $date) {
            $formatDate = $date->format('j')." ".$months[$date-
>format('n')]." ".$date->format('Y');
            Partials::dayZvit($travel->getId(), $day_number, $formatDate,
```

```
$formatNutritious);
                $day_number++;
            }
            ?>
            </ul>
        </div>
        <div class="tab-pane fade" id="nav-profile" role="tabpanel" aria-
labelledby="nav-profile-tab" style="background: none;">
            <?php
            $report = trim($travel->getName()). "¥n";
            $report.= 'Вага розкладки: '.$travel->getSumNut()."
г¥n";
            $formatNutritious = [];
            foreach ($nutritious as $nutrition){

$formatNutritious[$nutrition['day']][$nutrition['menu_type']][] =
Partials::nutritionText($nutrition);
            }
            $day_number = 1;
            /*формування звіту розкладки у
текстовому форматі*/
            foreach ($date_range as $date) {
                $formatDate = $date->format('j')." ".$months[$date-
>format('n')]." ".$date->format('Y');
                $report.=Partials::dayZvitText($travel->getId(), $day_number,
$formatDate, $formatNutritious);
                $day_number++;
            }
            ?>
            <div class="text" style="min-height: calc(100vh - 210px);">
                <div class="copy-area d-flex justify-content-end mb-1 p-1"
style="background-color: #F0F0F0; border: 1px solid #CED4DA; border-radius:
5px;">
                    <button class="btn btn-outline-info copy-button"
onclick="copyText('text-zvit')"><i class="bi bi-clipboard" id="copy-
icon"></i> Копіювати</button>
                </div>
                <textarea class="form-control w-100" style="min-height:
calc(100vh - 255px);; font-family: Courier, monospace;" id="text-zvit"
readonly><?= $report ?></textarea>
            </div>

        </div>
    </div>

</div>
```

```
<?php
Partials::Footer();
?>

<script>
    //ф у н к ц і я  п е р е в і р к и  з а к у п і в л і
    const checkPurchase = () => {
        const checkboxes = document.querySelectorAll('.purchase-check');
        checkboxes.forEach(checkbox => {
            checkbox.addEventListener('change', function() {
                console.log(this.checked);
                console.log(this.getAttribute("purchase-name"))
                if (this.checked){
                    fetch(`assets/php/api.php`,
                        {
                            method: "POST",
                            headers:{
                                "Content-Type":"application/x-www-form-
urlencoded",
                            },
                            body: new URLSearchParams({
                                add_purchase: 'add_purchase',
                                uid: document.getElementById('uid').value,
                                hid: document.getElementById('hid').value,
                                pname: this.getAttribute("purchase-name")
                            })
                        }
                    ).then(value => {
                        value.json().then(response=>{
                            console.log(response);
                        })
                    })
                }else{
                    fetch(`assets/php/api.php`,
                        {
                            method: "POST",
                            headers:{
                                "Content-Type":"application/x-www-form-
urlencoded",
                            },
                            body: new URLSearchParams({
                                delete_purchase: 'delete_purchase',
                                uid: document.getElementById('uid').value,
                                hid: document.getElementById('hid').value,
                                pname: this.getAttribute("purchase-name")
                            })
```

```
                        }
                    ).then(value => {
                        value.json().then(response=>{
                            console.log(response);
                        })
                    })
                }
            });
        });
    }
    checkPurchase();
</script>

</body>
</html>
```