

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

До захисту допускається

Завідувач кафедри

Скарга-Бандурова

І.С.

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

Веб-сервіс моніторингу денного харчового раціону

---

---

---

Освітньо-кваліфікаційний рівень “бакалавр”  
Спеціальність 123 – “комп’ютерна інженерія”

Керівник проекту:

\_\_\_\_\_ (підпис)

М. Є. Щербакова

\_\_\_\_\_ (ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_ (підпис)

Я. О. Критська

\_\_\_\_\_ (ініціали, прізвище)

Студент:

\_\_\_\_\_ (підпис)

В. І. Коннов

\_\_\_\_\_ (ініціали, прізвище)

Група:

КІ - 13аД

---

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерної інженерії  
Освітньо-кваліфікаційний  
рівень бакалавр  
Напрямок підготовки 6.050102 Комп'ютерна інженерія  
(шифр і назва)  
Спеціальність \_\_\_\_\_  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
\_\_\_\_\_ І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Коннов Віктора Іллїча

(прізвище, ім'я, по батькові)

1. Тема роботи Веб-сервіс моніторингу денного харчового раціону

керівник проекту (роботи) Щербакова Марина Євгеніївна, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "15" травня 2017 р. № 124/48

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи Розробка веб-сервісу моніторингу денного харчового раціону

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Провести аналіз розробки та технічних вимог на проектування веб-сервісу ; виконати проектування веб-сервісу денного харчового раціону Розробка рекомендацій по охороні праці та пожежній безпеці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я. А., асистент	22.05.2017	27.05.2017

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Аналіз і обґрунтування необхідності розробки КЛМ	До 22.05.2017	
2	Аналіз технічних вимог на проектування КЛМ	До 25.06.2017	
3	Складання плану до виконання дипломного проекту	До 26.05.2017	
4	Написання першого розділу	До 28.05.2017	
5	Написання другого розділу	До 30.05.2017	
6	Написання третього розділу	До 01.06.2017	
7	Написання четвертого розділу	До 03.06.2017	
8	Виконання та оформлення розділу з охорони праці	До 05.06.2017	
9	Написання вступу, висновків	До 08.06.2017	
10	Виправлення зауважень	До 09.06.2017	
11	Захист дипломного проекту	19.06.2017 (Відповідно до графіку)	

Студент \_\_\_\_\_

( підпис )

Коннов В. І.

(прізвище та ініціали)

Керівник \_\_\_\_\_

( підпис )

Щербакова М. Є.

(прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту бакалавра: 51 сторінка, 12 рисунків, 12 таблиць, 16 бібліографічних джерел, 2 додатки.

Об'єкт розробки: веб-сервіс моніторингу денного харчового раціону.

Мета роботи: розробка веб-сервісу моніторингу денного харчового раціону.

В проекті виконано:

- 1) аналіз засобів розробки та постановка технічного завдання;
- 2) Проектування веб-сервісу денного харчового раціону;
- 3) аналіз небезпечних та шкідливих виробничих факторів та розроблені заходи з техніки безпеки.

Отримано наступні результати:

В результаті виконання дипломного проекту було спроектовано та реалізовано веб-сервіс моніторингу денного харчового раціону.

**ХАРЧОВИЙ РАЦІОН, ВЕБ-СЕРВІС, БАЗА ДАНИХ, РНР.**

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

## ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ .....	7
1.1 Вибір засобів розробки.....	7
1.1.1 Мова розмітки HTML .....	7
1.1.2 Мова розмітки CSS .....	8
1.1.3 Мова сценаріїв JavaScript.....	9
1.1.4 Мова програмування серверної частини сайту PHP .....	10
1.1.5 Мова структурованих запитів SQL .....	12
1.2 Технічне завдання на розробку.....	14
1.2.1 Постановка завдання на розробку програмного забезпечення .....	14
1.3 Висновки .....	14
2 ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЕННОГО ХАРЧОВОГО РАЦІОНУ ..	15
2.1 Проектування бази даних.....	15
2.1.1 Огляд таблиць БД .....	16
2.2 Розробка структури сайту .....	21
3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	32
3.1 Загальні питання з охорони праці .....	32
3.1.1 Правові та організаційні основи охорони праці .....	33
3.1.2 Організаційно-технічні заходи з безпеки праці.....	34
3.2 Аналіз стану умов праці .....	35
3.2.1 Вимоги до приміщення .....	35
3.2.2 Вимоги до організації робочого місця.....	35
3.2.3 Навантаження та напруженість процесу праці.....	37
3.3 Виробнича санітарія .....	37
3.3.1 Аналіз небезпечних та шкідливих факторів при розробці виробу... ..	37
3.3.2 Пожежна безпека .....	39
3.3.3 Електробезпека.....	40

3.4 Гігієнічні вимоги до параметрів виробничого середовища.....	41
3.4.1 Мікроклімат.....	41
3.4.2 Освітлення.....	42
3.4.3 Вентилювання.....	44
3.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій.....	44
5.5.1 Розрахунок захисного заземлення.....	46
3.6 Висновки.....	50
ПЕРЕЛІК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ.....	52
ДОДАТОК А.....	54

## ВСТУП

У сучасному світі масова культура пропагує здоровий спосіб життя набирає популярність дотримання людьми здорового способу життя.

Можливість цифрової оцінки величини енергії, що витрачається, а також оцінки споживаної їжі однієї і тієї ж одиницею виміру (калорія) дозволили вирішити важливу задачу по встановленню адекватності харчування виробленим енерговитратами. Крім того, перехід на калорійне обчислення дозволив досить об'єктивно оцінювати кількісну сторону харчування і в необхідних випадках оперативно вносити відповідні корективи в харчування, збільшуючи або зменшуючи його калорійність.

Значимими факторами є дотримання раціону і графіка споживання їжі, для цього людині необхідно вести контроль і облік споживаної нею їжі.

Так само слід враховувати потреби до раціону у людей яким призначені лікувальні дієти. Правильно організоване і побудоване на сучасних наукових засадах раціональне і дієтичне харчування забезпечує нормальний перебіг процесів росту і розвитку організму, збереження здоров'я і працездатності людини.

Спростити цю задачу можливо створивши веб-сервіс, з допомогою якого можливо буде вести контроль денного харчового раціону.

В дипломному проекті передбачається розробити веб-сервіс за допомогою якого користувач зможе: вносити дані про спожиту їжу, отримувати інформацію про спожиті калорії, зберігати інформацію про спожиту їжу, вводити и редагувати дані про себе.

## 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Вибір засобів розробки

#### 1.1.1 Мова розмітки HTML

HTML (HyperText Markup Language — Мова розмітки гіпертекстових документів) — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.



### 1.1.2 Мова розмітки CSS

CSS - це мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. Назва походить від англ. Cascading Style Sheets - каскадні таблиці стилів.

Простіше кажучи, мова CSS призначений для того, щоб надавати необхідний зовнішній вигляд HTML-документами.

Додання зовнішнього вигляду документів HTML - це хоч і найпопулярніший, однак лише окремий випадок застосування мови CSS, тому що з його допомогою можна надавати вид і документам інших типів: XHTML, SVG і XUL.

Метою створення CSS було відділення опису логічної структури веб-сторінки від її зовнішнього вигляду. Як ви вже знаєте, для опису структури використовується HTML, для опису ж того, як ця логічна структура буде виглядати, відповідає як раз CSS.

Роздільне опис логічної структури та подання документа дозволяє більш гнучко управляти зовнішнім виглядом документа і мінімізувати обсяг повторюваного коду, який би неминуче виникав при використанні HTML для опису зовнішнього вигляду документа.

За допомогою CSS веб-розробник може задавати для сторінки і окремих її елементів різні гарнітури та розміри шрифту, кольору елементів, відступи елементів один від одного, розташування окремих блоків на сторінці і т.д.

Зрозуміло, для того, щоб використовувати CSS для додання зовнішнього вигляду HTML-документу, потрібно цей документ якимось

пов'язати зі стилями, тобто «Повідомити» HTML-документу, що він буде оформлений за допомогою CSS.

Для цього існують різні способи підключення CSS до документа, які дають браузеру знати, що на сторінку в цілому, або до якихось окремих її елементів повинно бути застосовано стилеве оформлення.

Таблиці стилів можуть розташовуватися як безпосередньо всередині того, документа, до яких вони будуть застосовуватися, так і перебувати в окремому файлі, що має розширення .css.

CSS-файл це звичайний текстовий файл. У ньому пишуться спеціальні інструкції, що описують зовнішній вигляд елемента та його позиціонування на сторінці а також коментарі (довільні пояснення щодо написаних інструкцій).

### **1.1.3 Мова сценаріїв JavaScript**

JavaScript — динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні сервера (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

Структурно JavaScript можна представити у вигляді об'єднання трьох чітко помітних одна від одної частин:

- ядро (ECMAScript),
- об'єктна модель браузера (Browser Object Model або BOM (en)),
- об'єктна модель документа (Document Object Model або DOM).

Якщо розглядати JavaScript в відмінних від браузера середовищах, то об'єктна модель браузера і об'єктна модель документа можуть не підтримуватися.

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа. На противагу цьому ряд авторів знаходять BOM і DOM тісно взаємопов'язаними.

#### **1.1.4 Мова програмування серверної частини сайту PHP**

PHP (Hypertext Preprocessor — PHP: гіпертекстовий препроцесор), попередня назва: Personal Home Page Tools — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення, тобто вихідний код розповсюджується .

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє

використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

PHP це наступник продукту під назвою PHP / FI. Створене в 1994 році Расмусом Лердорфом, найперше втілення PHP було простим набором CGI-скриптів, написаних на мові програмування Сі. Спочатку використовуючи їх для відстеження відвідувань свого веб-резюме, він назвав цей набір скриптів "Personal Homepages Tools" ( "Інструменти для персональних домашніх сторінок"), але більш часто згадувалася назва "PHP Tools". Згодом було потрібно все більше поліпшень функціональності, і Расмус переписав PHP Tools, створивши більш велику і багату реалізацію. Ця нова реалізація була здатна взаємодіяти з базами даних і багато іншого, що створювало фреймворк, за допомогою якого користувачі могли створювати прості динамічні веб-додатки, такі як гостьові книги. У червні 1995 року Расмус »відкрив вихідний код PHP Tools громадськості, що дозволило розробникам використовувати його на свій розсуд. Це також дало можливість користувачам виправляти помилки в коді і покращувати його.

Область застосування PHP достатньо обширна і різноманітна. Проте існує безліч інших мов програмування, здатних вирішувати схожі задачі. PHP дуже простий у вивченні. Досить ознайомитися лише з основними правилами синтаксису і принципами його роботи, і можна починати писати власні програми, причому братися за такі завдання, вирішення яких на іншій мові вимагало б серйозної підготовки.[16]

### 1.1.5 Мова структурованих запитів SQL

SQL (structured query language - «мова структурованих запитів») - формальна непроцедурна мова програмування, застосовується для створення, модифікації та управління даними в реляційній базі даних, керованої відповідною системою управління базами даних (СУБД).

Перша версія SQL була розроблена на початку 1970-х років у IBM. Ця версія мала назву SEQUEL і була призначена для обробки та пошуку даних, що містилися в реляційній базі даних IBM, System R. Мова SQL надалі була стандартизована Американськими Держстандартами (ANSI) в 1986. На початку, SQL була запланована як мова запитів і управління даними, а пізніші модифікації SQL створені продавцями системи управління базами даних, які додали процедурні конструкції, control-of-flow команд і розширення мов. З випуском стандарту SQL:1999, такі розширення були формально запозичені як частина мови SQL через Persistent Stored Modules (SQL/PSM).

Мова SQL поділяється на кілька частин:

- DDL (Data Definition Language) - робота зі структурою бази,
- DML (Data Manipulation Language) - робота з рядочками,
- DCL (Data Control Language) - робота з правами,
- TCL (Transaction Control Language) - робота з транзакціями.

SQL Data Manipulation Language (DML - мова керування даними) SQL призначений для виконання запитів. Крім того в SQL входить синтаксис для поновлення, вставки і знищення даних. Цей синтаксис разом з командами поновлення формує мову управління даними (DML):

- SELECT - отримує дані з таблиці БД;

- UPDATE - оновлює дані в таблиці БД;
- DELETE - знищує дані в таблиці БД;
- INSERT INTO - вставляє нові дані в таблицю БД;

SQL Data Definition Language (DDL - мова визначення даних) – DDL є частиною SQL, яка управляє створенням і видаленням таблиць в БД, Крім того, за допомогою DDL ми можемо призначати індекси (ключові слова), налагоджувати взаємозв'язки між таблицями і накладати обмеження на таблиці БД.

Найважливішими командами DDL є такі команди:

- CREATE TABLE - створення нової таблиці;
- ALTER TABLE - зміна існуючої таблиці;
- DROP TABLE - видалення таблиці;
- CREATE INDEX - створення індексу (ключового слова для полегшення пошуку);
- DROP INDEX - видалення індексу;[16]

## **1.2 Технічне завдання на розробку**

### **1.2.1 Постановка завдання на розробку програмного забезпечення**

Головним завданням було створення сервісу за допомогою якого можливо буде вести облік споживаної їжі з огляду на специфічні потреби користувача (призначена дієта і т.д.). Для цього потрібно реалізувати:

- авторизацію користувача;
- особистий кабінет в якому буде можливість вказувати інформацію про себе і свої потреби, вести облік споживаних продуктів харчування;
- інформаційні сторінки;
- зручний інтерфейс для доступу до даних користувача і даними з обліку споживання їжі.

## **1.3 Висновки**

В даному розділі був проведений аналіз і постановка завдання, а так само вибір засобів розробки. В якості засобів для реалізації веб-сервісу були обрані найбільш підходящі технології: HTML, CSS, JavaScript, PHP, SQL.

## 2 ПРОЕКТУВАННЯ ВЕБ-СЕРВІСУ ДЕННОГО ХАРЧОВОГО РАЦІОНУ

### 2.1 Проектування бази даних

Для зберігання інформації про користувачів, прийомах їжі, продуктах харчування, дієти. Спроектвана схема бази даних яка була реалізована в СУБД MySQL (рис. 2.1).

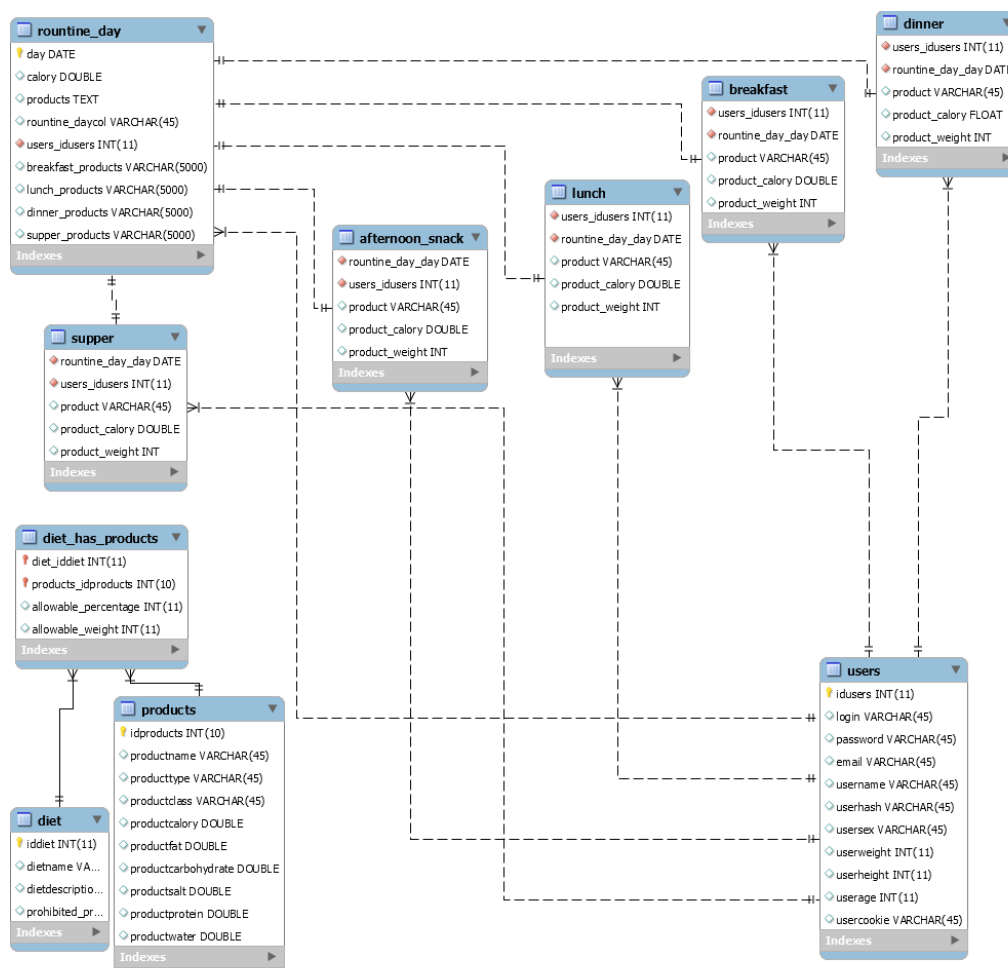


Рисунок 2.1 – Схема бази даних сервісу у вигляді ER-діаграми



### 2.1.1 Огляд таблиць БД

Таблиця – це сукупність пов'язаних даних, що зберігаються в структурованому вигляді в базі даних, є видом уявлення реляційного відносини. Складається з стовпців і рядків, перетин стовпця і рядка називається осередком. Таблиця може містити обмежене число стовпців і будь-яке число рядків. У кожному рядку повинно бути хоча б 1 унікальне значення, яке називається первинним ключем.

В таблиці 2.1 представлені описи всіх таблиць спроектованої бази даних.

Таблиця 2.1 – Опис таблиць бази даних

Назва таблиці бази даних	Опис призначення таблиці
users	Містить інформацію про користувачів.
products	Містить інформацію про продукти харчування.
diet	Містить опис дієт.
diet_has_products	Таблиця зв'язує таблиці diet і products.
routine_day	Містить інформацію про дні розпорядку користувачів.
breakfast	Містить інформацію про продукти які вжиті на сніданок.
lunch	Містить інформацію про продукти які вжиті на другий сніданок.
dinner	Містить інформацію про продукти які вжиті на обід.
afternoon_snack	Містить інформацію про продукти які вжиті на полуденок.
supper	Містить інформацію про продукти які вжиті на вечерю.

Для зберігання інформації про користувачів була створена таблиця «users» (таблиця 2.2).

Таблиця 2.2 – Опис стовпців таблиці «users»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
idusers	int	Ціле число.	Унікальний ідентифікатор користувача, первинний ключ.
login	varchar	рядок змінної довжини	Логін користувача. Використовується для авторизації на сайті.
password	varchar	Рядок змінної довжини.	Пароль користувача. Використовується для авторизації на сайті.
email	varchar	Рядок змінної довжини.	Електронна пошта користувача.
username	varchar	Рядок змінної довжини.	Ім'я користувача.
userhash	varchar	Рядок змінної довжини.	COOKIES користувача. Використовується для аутентифікації.
usersex	varchar	Рядок змінної довжини.	Містить інформацію про стать клієнта.
userweight	int	Ціле число.	Містить інформацію про вагу користувача
userheight	int	Ціле число.	Містить інформацію про зріст користувача.
userage	int	Ціле число	Містить інформацію про вік користувача.

Для зберігання інформації про продукти була створена таблиця «products» (таблиця 2.3).

Таблиця 2.3 – Опис стовпців таблиці «products»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
idproducts	int	Ціле число	Унікальний ідентифікатор продукту, первинний ключ.
productname	varchar	Рядок змінної довжини.	Містить назву продукту.
producttype	varchar	Рядок змінної довжини.	Містить тип продукту.
productclass	varchar	Рядок змінної довжини.	Містить клас продукту.

## Продовження таблиці 2.3 – Опис стовпців таблиці «products»

productcalory	double	Число з плаваючою комою.	Містить інформацію про калорійність продукту (калорій / 100гр)
productfat	double	Число з плаваючою комою.	Містить інформацію про вміст жиру в продукті.
productcarbohydrate	double	Число з плаваючою комою.	Містить інформацію про вміст вуглеводів в продукті.
productsalt	double	Число з плаваючою комою.	Містить інформацію про вміст солі в продукті.
productprotein	double	Число з плаваючою комою.	Містить інформацію про вміст білку в продукті.
productwater	double	Число з плаваючою комою.	Містить інформацію про вміст води в продукті.

Для зберігання інформації про дієти була створена таблиця «diet» (таблиця 2.4).

Таблиця 2.4 – Опис стовпців таблиці «diet»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
iddiet	int	Ціле число.	Унікальний ідентифікатор дієти, первинний ключ.
dietname	varchar	Рядок змінної довжини.	Містить назву дієти.
dietdescription	mediumtext	Може зберігати не більше 16 777 215 символів.	Містить інформацію про дієту.
prohibited_products	varchar	Рядок змінної довжини.	Містить інформацію про заборонені продукти.

Для зберігання інформації про денний розпорядок була створена таблиця «routine\_day»(таблиця 2.5).

Таблиця 2.5 – Опис стовпців таблиці «routine\_day»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
day	date	Дата	Унікальний ідентифікатор дня, первинний ключ.
calory	double	Число з плаваючою комою.	Містить кількість калорій спожитих за день.
products	text	Може зберігати не більше 65 535 символів.	Містить список продуктів спожитих за день.
routine_daycol	double	Число з плаваючою комою.	Містить число калорій спожитих за день.
users_idusers	int	Ціле число.	Містить унікальний ідентифікатор користувача, зовнішній ключ таблиці users.
breakfast_products	varchar	Рядок змінної довжини.	Містить список продуктів спожитих за сніданок.
lunch	varchar	Рядок змінної довжини.	Містить список продуктів спожитих за другий сніданок.
dinner	varchar	Рядок змінної довжини.	Містить список продуктів спожитих за обід.
supper	varchar	Рядок змінної довжини.	Містить список продуктів спожитих за вечерю.

Для зв'язку таблиць «products» і «diet» за правилом багато до багатьох була створена розподільна таблиця «diet\_has\_products»(таблиця 2.6).

Таблиця 2.6 – Опис стовпців таблиці «diet\_has\_products»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
diet_iddiet	int	Ціле число.	Містить унікальний ідентифікатор дієти, зовнішній ключ від таблиці diet.

## Продовження таблиці 2.6 – Опис стовпців таблиці «diet\_has\_products»

products_idproducts	int	Ціле число.	Містить унікальний ідентифікатор продукту харчування, зовнішній ключ від таблиці products.
allowable_percentage	int	Ціле число.	Містить число яке вказує сумісність продукту з дієтою
Allowable_wieght	int	Ціле число.	Містить число яке вказує допустиму вагу продукта.

Для зберігання інформації про спожиту їжу на сніданок, другий сніданок, обід, полуденок та вечерю були створені таблиці: «breakfast», «lunch», «dinner», «afternoon\_snack», «supper». Ці таблиці мають однакові стовбці.(таблиця 2.7)

Таблиця 2.7 – Опис стовпців таблиць «breakfast», «lunch», «dinner», «afternoon\_snack», «supper»

Ім'я стовпця таблиці	Тип даних	Пояснення до типу даних	Опис призначення стовпця
users_idusers	int	Ціле число.	Містить унікальний ідентифікатор користувача, зовнішній ключ від таблиці users.
rountine_day_day	date	Дата.	Містить дату дня прийому їжі, зовнішній ключ від таблиці routine_day .
product	varchar	Рядок змінної довжини.	Містить назву продукта.
product_calory	double	Число з плаваючою комою.	Містить кількість калорій з урахуванням ваги продукта
product_weight	int	Ціле число.	Містить число ваги продукту.

## 2.2 Розробка структури сайту

Розробка структури сайту велася відповідно до необхідного функціоналу, а саме:

- а) головна сторінка, вона ж сторінка авторизації
- б) особистий кабінет який надає доступ до наступних розділів сайту::
  - 1) інформація про користувача;
  - 2) редагування інформації користувача;
  - 3) введення інформації про спожиту їжі;
  - 4) вивід інформації про спожиту їжу;
- в) інформацію про дієти



Рисунок 2.2 – Структурна схема сайту.

На головній сторінці знаходиться форма авторизації, а так само гіперпосилання на сторінки з інформацією, реєстрацією та контактними даними (рис. 2.3).

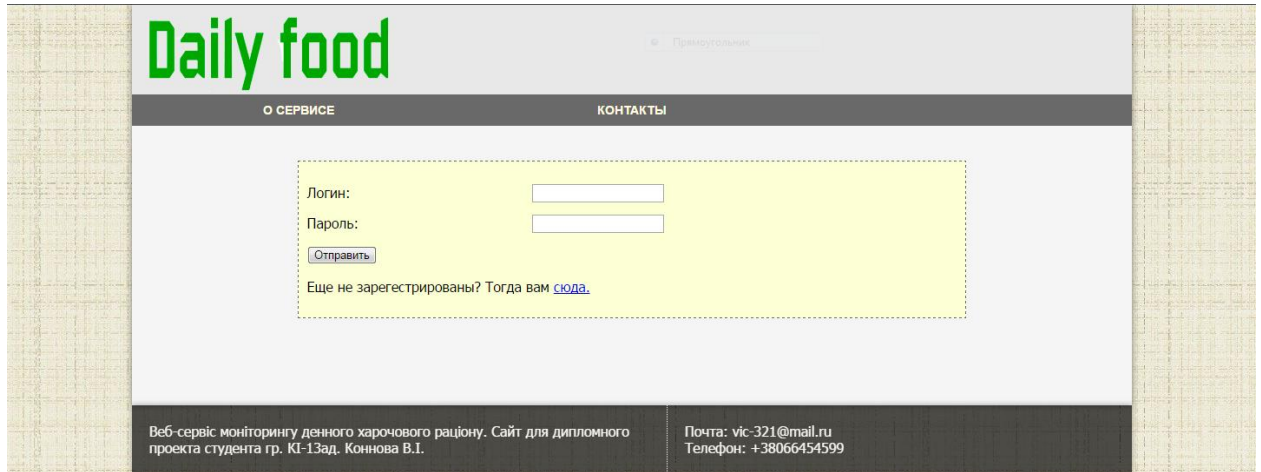


Рисунок 2.3 – Головна сторінка сервісу.

При введенні логіна і пароля форма відправляє дані на обробку в файл testreg.php.

Код файлу testreg.php :

```
<?php
include 'includes/testlog.php';
include 'includes/functions.php';
if (isset($_POST['login'])) { $login = $_POST['login']; if ($login == '') {
unset($login);} }
if (isset($_POST['password'])) { $password=$_POST['password']; if ($password == '') {
unset($password);} }
if (empty($login) or empty($password)) {
exit ("Вы ввели не всю информацию, вернитесь назад и заполните все поля!");
}
$login = clean($login);
$password = clean($password);
$query = mysqli_query($link, "SELECT * FROM `users` WHERE `login` = '$login' AND
`password` = '$password'");
$user_data = mysqli_fetch_assoc($query);
if($login == $user_data['login'] && $password == $user_data['password']) {
    $_SESSION['password']=$user_data['password'];
    $_SESSION['login']=$user_data['login'];
    $hash = md5($login.$password).'fx452';
    mysqli_query($link, "UPDATE `users` SET `userhash` = '$hash' WHERE `login` =
'$login'");
    setcookie('login', $login, time()+60*60*24*7);
    setcookie('userhash', $hash, time()+60*60*24*7);
}
echo "<html><head><meta http-equiv='Refresh' content='0';
URL=index.php'></head></html>";
```

?>

Отримані з форми дані порівнюються з даними з бази даних, якщо значення співпали то користувач отримує для користувача генеруються COOKIES, і їх актуальна версія для зберігання зберігається на сервері в базі даних з таблиці «users» в стовпці «userhash».

Для доступу до всіх сторінок окрім сторінки авторизації та реєстрації користувач повинен бути авторизованим, тому на всіх цих сторінках є система аутентифікації яка реалізована наступним чином:

```
include'db_connect.php';
if (isset($_COOKIE['login']) && isset($_COOKIE['userhash'])) {
$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]');");
$user_data = mysqli_fetch_assoc($user_data);
$log_OK = false;
if ($user_data['userhash'] == $_COOKIE['userhash'] && $user_data['login'] ==
$_COOKIE['login']) {$log_OK = true;
} else {$log_OK = false;}
```

При натисканні на посилання «сюди» виводиться сторінка реєстрації(рисунок 2.4)

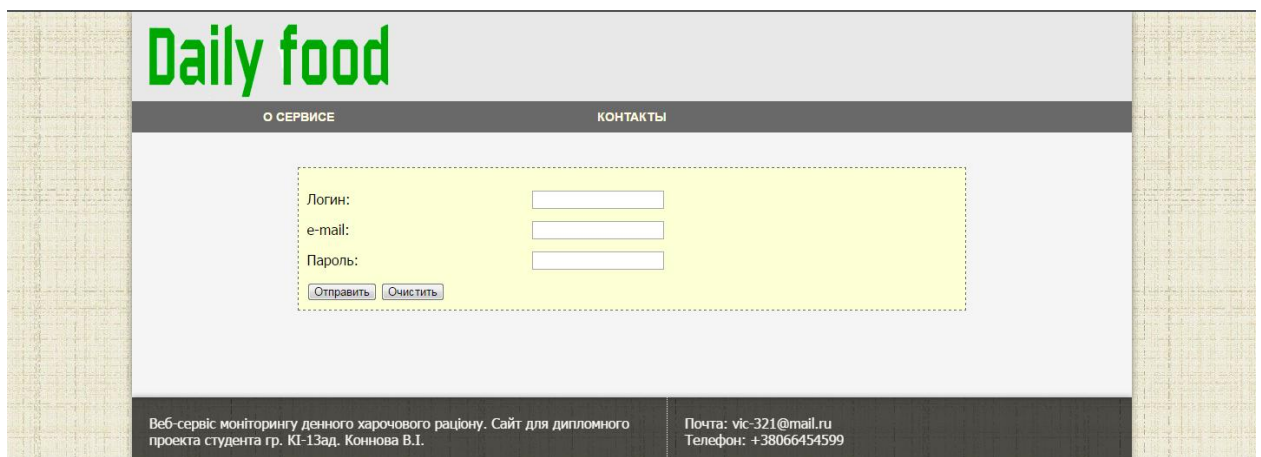


Рисунок 2.4 – Сторінка реєстрації

Поля введення реєстрації перевіряються і фільтруються. У разі невідповідності правилам буде видано попередження(рис. 2.5). Реєстрація буде успішною якщо не виникло ніяких попереджень.



Рисунок 2.5 – Перевірка форми реєстрації

Перевірка правильності введення виконується рhp скриптом на стороні сервера. Нижче представлений код який обробляє форму:

```

if (isset($_POST['reset'])) unset($_POST);
if (!empty($_POST)) {
    $_POST['login'] = clean($_POST['login']);
if (!check_length($_POST['login'], 2, 20)) {$error[0] .= "Длина логина должна быть
от 2 до 20 символов. ";}

$_POST['password'] = clean($_POST['password`']);
if (!check_length($_POST['password'], 8, 20)) {$error[2] .= "Длина пароля должна
быть от 8 до 20 символов. ";}

$email = mysqli_query($link, "SELECT * FROM `users` WHERE `email` = \"$_POST[email]\"
"); //Занят до бази даних для перевірки на зайнятість email-а
$email = mysqli_fetch_assoc($email);

$_POST['email'] = clean($_POST['email']);
if ($email['email'] == $_POST['email']) {$error[1]="данный email уже занят !";}

if(empty($error)) {
    mysqli_query($link, "INSERT INTO `users` (`idusers`, `login`, `password`,
`email`) VALUES(\"\", \"$_POST[login]\", \"$_POST[password]\", \"$_POST[email]\");
    echo '<script> window.location.href = "/index.php"</script>';
}
}

```

В наведеному кодi використовувалася для користувача функція clean за допомогою якої з отриманих даних форми вилучалися зайві пробіли, теги і екрануючі символи:

```

function clean($value = "") {
    $value = trim($value);
    $value = stripslashes($value);
    $value = strip_tags($value);
}

```

```

$value = htmlspecialchars($value);
return $value;
}

```

Якщо користувач вже зареєстрований і виконав вхід то він потрапляє на сторінку «Приватний кабінет»(рис. 2.6)

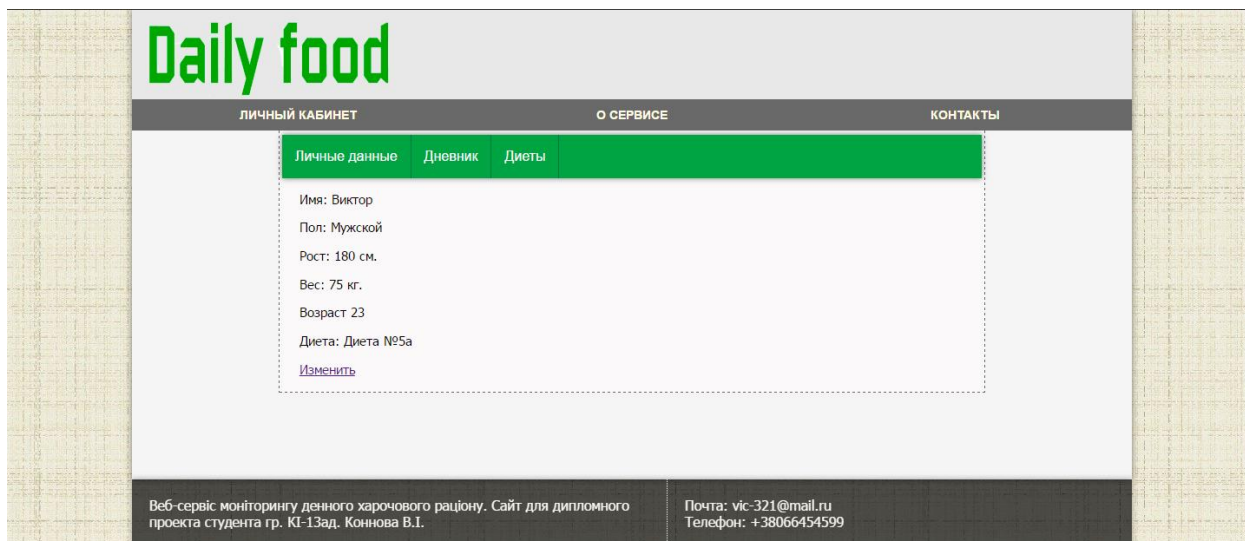


Рисунок 2.6 – Сторінка приватного кабінету

На сторінці показані дані користувача. Користувач при може їх змінити перейшовши за гіперпосиланням "змінити" на сторінку редагування особистих даних(рисунок 2.7).

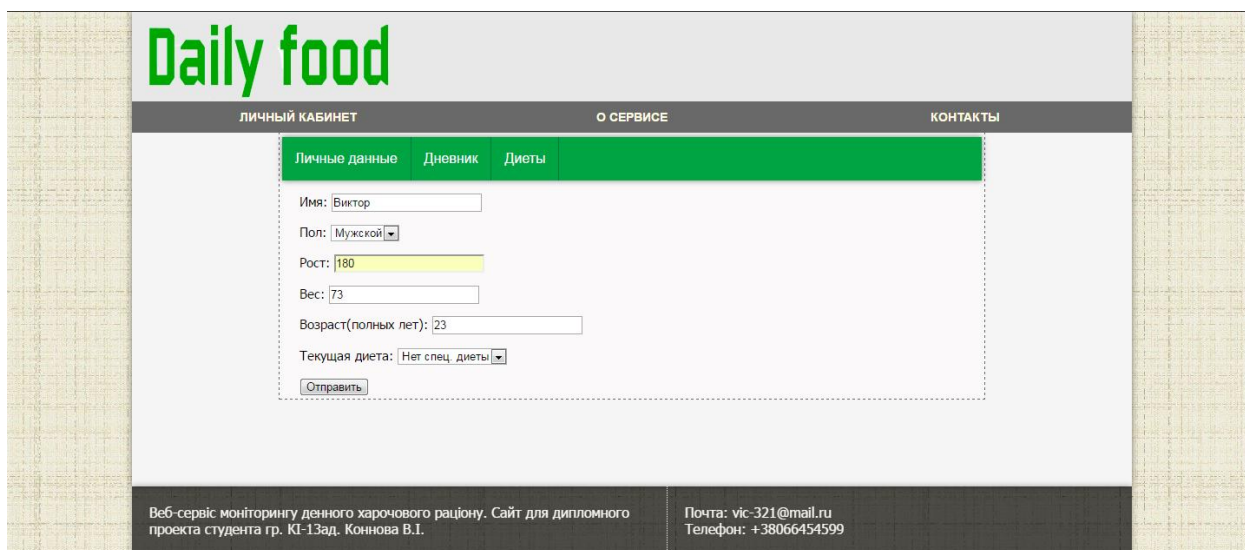


Рисунок 2.7 – Сторінка редагування даних.

На сторінці редагування даних користувач може ввести або змінити таке дані: ім'я, стать, зріст, вагу, вік, поточну дієту.

На сторінці щоденника в пункті «внести запис» користувач може ввести дані про спожиту їм їжі: вибрати найменування продукту, вказати масу (в грамах) час прийому (розділене на сніданок, другий сніданок, обід, полуденок і вечеря) і дату.(рис. 2.8)

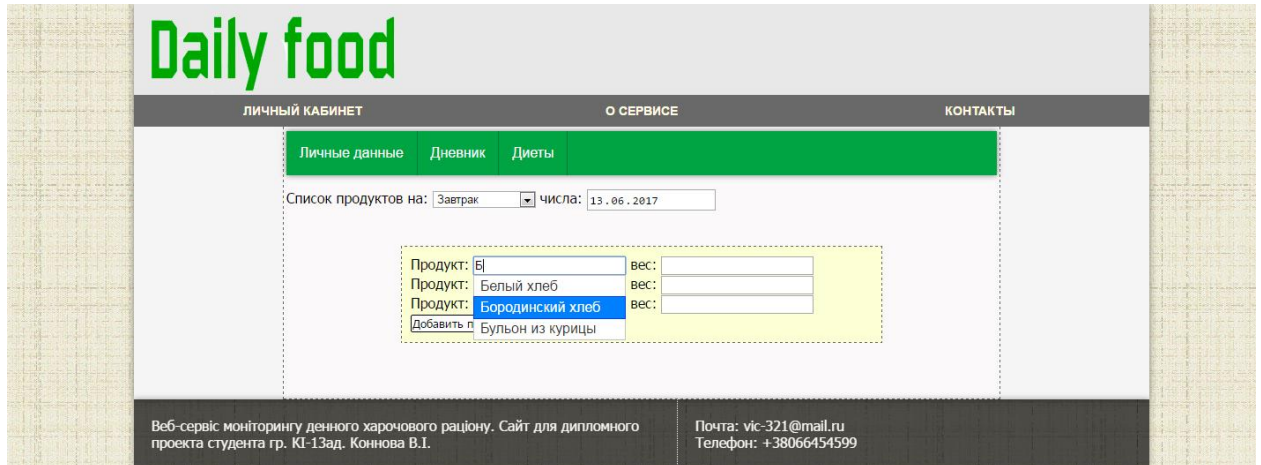


Рисунок 2.8 – Сторінка ведення харчового щоденника

Для того щоб не вводити дані по 1 продукту за 1 раз і для того що б у формі спочатку не було зайвих невикористання полів засобами JavaScript було реалізовано додавання нового рядка з полями вводу(рис. 2.9).

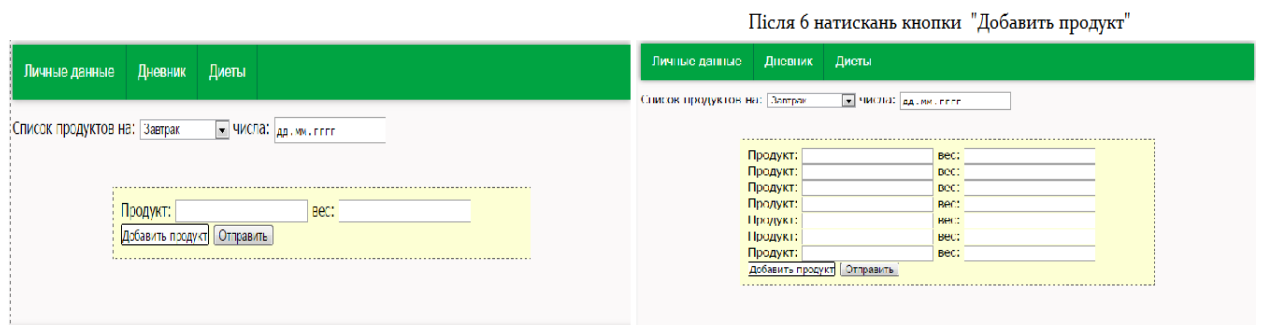


Рисунок 2.9 – Приклад роботи скрипта.

Код JavaScript:

```
var nr = 1;
function additem() {
    div_elem = document.getElementById('ration_add_div');
```

```

div_elem.innerHTML = div_elem.innerHTML + ' Продукт: <input type="text" name="item_'
+ rr + '" onfocus="search_product(this.name,this.value);" > вес: <input type="text"
name="weight_' + rr + '"> <br>';
    rr++;
}

```

Принцип роботи скрипта полягає в тому що при натисканні кнопки "додати продукт" викликається функція `additem()` в якій: `div_elem = document.getElementById ('ration_add_div');` - отримання id блоку в якому додаються рядки, `div_elem.innerHTML = div_elem.innerHTML + 'Продукт: <input type = "text" name = "item_' + rr + '" onfocus = "search_product (this.name, this.value) ; " > Вага: <input type = "text" name = "weight_ '+ rr +'"> <br> ';` - додавання елемента безпосередньо в блок через властивість `innerHTML`.

При введенні назви продукту для зручності в випадний список динамічно засобами AJAX з бази даних завантажуються назви продуктів які містять введені в поле вводу символи(рис. 2.8).

#### Код JavaScript:

```

prod_arr = "";
function get_prod(str) {
    if (str.length == 0) {
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                arr = this.responseText;
                prod_arr = arr.split(',');
            }
        };
        $( function() {
            var availableTags = prod_arr;
            $( "#tags" ).autocomplete({
                source: availableTags
            });
        });
    }
};
xmlhttp.open("GET", "search_prod_ajax.php?search_str=" + str, true);
xmlhttp.send();
}
}

```

Суть роботи скрипта полягає в тому щоб за подією `onkeyup` відправляється поточне значення поля вводу на сервер для обробки, після чого отримуємо з сервера рядок в якій потрібні результати розділені комою, за допомогою функції `split ()` переформатуємо рядок в масив. Далі отриманий

масив передається функції autocomplete яка і створює випадний список з варіантами вибору продуктів. autoComplete - це плагін для бібліотеки JQuery.

#### Код PHP:

```
<?php
header('Content-Type: text/plain; charset=utf-8');
include'includes/db_connect.php';
$available_products = array();
$req_str = $_REQUEST['search_str'];
if($req_str != "")
{
    if($query = mysqli_query($link,"SELECT * FROM `products` WHERE `productname`
RLIKE \"\$req_str\" LIMIT 10 "))
    {
        while($available_products = mysqli_fetch_assoc($query))
            $str .= "$available_products[productname], ";
    }
    $str = substr($str, 0, (mb_strlen($str) - 2) );
}
echo $str;
unset($str);
unset($req_str);
unset($available_products);
?>
```

Даний php скрипт приймає дані від форми по GET запити, далі йде звернення до бази даних і шукаються в ній продукти в імені яких є отримані від форми символи. Далі формується рядок з назв продуктів які задовольнили умови пошуку, після чого цей рядок відправляється користувачеві.

Сторінка «Вивести історію» містить форму вводу з полями вибору дати і часового проміжку (сніданок, другий сніданок, обід, полуденок, вечеря). Після вибору дати і часового проміжку користувачеві виводяться дані про спожиту їм їжу.(рис. 2.10)

**Daily food**

ЛИЧНЫЙ КАБИНЕТ      О СЕРВИСЕ      КОНТАКТЫ

Личные данные    Дневник    Диеты

Выбор даты:   
 Выбор временного промежутка:

Данные на: 2017-06-13 - Завтрак

Продукт: Белый хлеб
Вес: 400 гр.
Калорийность: 1066

---

Продукт: Арахис жаренный
Вес: 100 гр.
Калорийность: 624

---

Продукт: Гречневая каша
Вес: 350 гр.
Калорийность: 462

---

Всего калорий: 2152

Данные на: 2017-06-13 - Полдник

Продукт: Белый хлеб
Вес: 666 гр.
Калорийность: 1774.89

---

Продукт: Арахис жаренный
Вес: 777 гр.
Калорийность: 4848.48

---

Всего калорий: 6623.37

Рисунок 2.10 – Сторінка «Вивести історію»

Код відображення форми вводу:

```

if($_GET['history_of_eat'] == 1)
{
echo <<<EOF

<form name="day_ration_add" action="private_office.php?history_of_eat=1"
method="POST" class="history_form">

Выбор даты: <input type="date" name = "select_date"><br>
Выбор временного промежутка:
<select name="type_of_food_intake"><option>Целый
день</option><option>Завтрак</option><option>Второй
завтрак</option><option>Обед</option><option>Полдник</option><option>Ужин</option></s
elect>

<br>

<input type="submit" value="Получить данные">

</form>

EOF;

```

Код виводу даних про спожиту їжу за весь день:

```

if(isset($_POST['select_date']) && isset($_POST['type_of_food_intake'])) {
$type_of_food_intake = $_POST['type_of_food_intake'];
if($type_of_food_intake == "Целый день")
{
foreach ($type_a_food_intake_assoc_array as $key => $value) {

```

```

        $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$key]` WHERE `routine_day_day`='$current_day'");

        $empty_cheker = mysqli_fetch_assoc(mysqli_query($link,"SELECT *
FROM `$type_a_food_intake_assoc_array[$key]` WHERE
`routine_day_day`='$current_day'"));
        if($history_of_food&&$empty_cheker){
            echo " <div class='history_form'>
            <br>Данные на: $current_day - $key";
            while($data_array = mysqli_fetch_assoc($history_of_food))
        {
                echo "<div class = 'product_report'>
                Продукт: $data_array[product] <br>
                Вес: $data_array[product_weight] гр. <br>
                Калорийность: $data_array[product_calory]
                </div>";
            }
            $summ_calory = summ_calory(mysqli_query($link,"SELECT * FROM `$value` WHERE
`routine_day_day`='$current_day'"));
            echo "Всего калорий: $summ_calory </div>";
        }
    }
}

```

### Код виводу даних за 1 часовий проміжок

```

        $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$type_of_food_intake]` WHERE
`routine_day_day`='$current_day'");
        if($history_of_food){
            echo " <div class='history_form'>
            <br>Данные на: $current_day - $type_of_food_intake";
            while($breakfast_array = mysqli_fetch_assoc($history_of_food))
        {
                echo "<div class = 'product_report'>
                Продукт: $breakfast_array[product] <br>
                Вес: $breakfast_array[product_weight] гр. <br>
                Калорийность: $breakfast_array[product_calory]
                </div>";
            }
            $summ_calory = summ_calory(mysqli_query($link,"SELECT *
FROM `breakfast` WHERE `routine_day_day`='$current_day'"));
            echo "Всего калорий: $summ_calory </div>";
        }
    }
}

```

Повний код скрипта відображення цієї сторінки міститься в файлі `private_office.php` (Додаток А).

Сторінка «Дієти» містить таку інформацію про дієти як назву та опис(рис. 2.11).

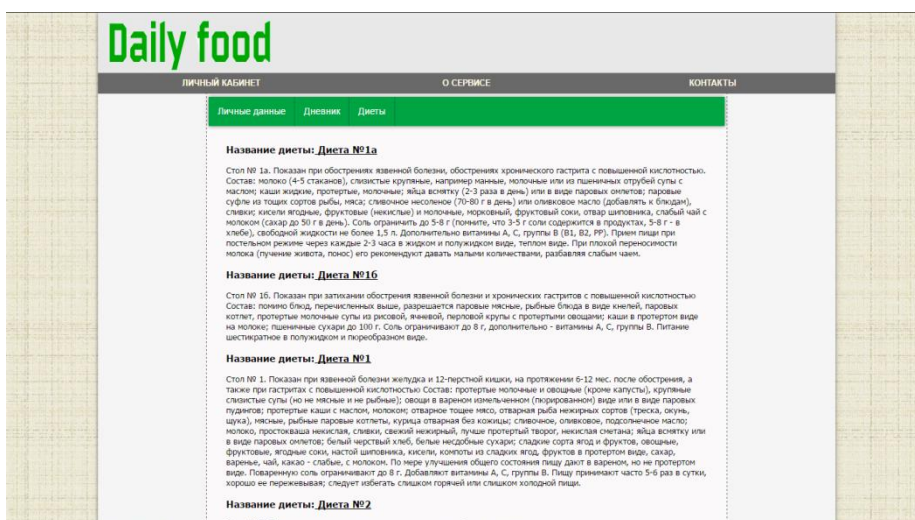


Рисунок 2.11 – Сторінка «Дієти»

Ділянка коду відповідає за відображення списку дієт:

```
if($_GET['diets'] == 1)
{
    $get_info_of_diets = mysqli_query($link, "SELECT * FROM `diet` ");

    echo '<div id="diets_div">';
    if($get_info_of_diets) {
        while($d_row = mysqli_fetch_assoc($get_info_of_diets))
        {
            echo "<h3>Название диеты:<span class='diet_span'>";
            $d_row[dietname]</span></h3>";
            echo "<p> $d_row[dietdescription]</p>";
        }
    } else {
        echo "Инофрмация о диетах отсутствует!";
    }
    echo "</div>";
}
```

## 2.3 Висновки

В даному розділі була складена і реалізована схема бази даних, розроблено веб-інтерфейс користувача, викорістовуючі сучасні веб-технології: HTML5, CSS3, JavaScript, AJAX. Також була реалізована логічна структура веб-сервісу у вигляді PHP сценаріїв.



### **3 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даного проекту бакалавра було розробка веб-сервісу моніторингу денного харчового раціону. Так як в процесі проектування виконувалось у домашніх умовах, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера на якому буде розроблено веб-сервіс.

#### **3.1 Загальні питання з охорони праці**

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними

речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган (далі роботодавець). Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Роботодавець повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація, небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

### **3.1.1 Правові та організаційні основи охорони праці**

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Державна політика в галузі охорони праці визначається відповідно до Конституції України Верховною Радою України і спрямована на створення

належних, безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Відповідно до статті 3 Закону України «Про охорону праці» (далі – Закону) законодавство про охорону праці складається з Закону, Кодексу законів про працю України, Закону України "Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності" та прийнятих відповідно до них нормативно-правових актів, норм міжнародного договору (ратифіковані Конвенції і Рекомендації МОТ, директиви Європейської Ради).

### **3.1.2 Організаційно-технічні заходи з безпеки праці**

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнагляддохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [5].

Обов'язковими вимогами враховане наступне:

– не слід допускати до роботи осіб, що в установленому порядку не пройшли навчання, інструктаж та перевірку знань з охорони праці, пожежної безпеки та цих Правил.

– на підприємстві/організації, де експлуатуються ЕОМ з відео дисплейними терміналами (ВДТ) і периферійними пристроями (ПП), розробляється інструкція з охорони праці відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнагляддохоронпраці від 29.01.98 N 9, зареєстрованого в Міністерстві юстиції України 07.04.98 за N 226/2666 [7].

– ознайомлення з правилами безпеки праці, одержання відповідних інструктажів засвідчується у журналі інструктажів.

### 3.2 Аналіз стану умов праці

Робота над створенням локальної комп'ютерної мережі проходить в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

#### 3.2.1 Вимоги до приміщення

Геометричні розміри приміщення зазначені у таблиці 5.1.

Таблиця 5.1 – розміри робочого місця

Параметр	Значення
Довжина, м	3
Ширина, м	3
Висота, м	2,5
Площа, м <sup>2</sup>	15
Об'єм, м <sup>3</sup>	37,5

Згідно до санітарних норм мікроклімату виробничих приміщень [7] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм – не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

#### 3.2.2 Вимоги до організації робочого місця

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця [8] (табл. 5.2) і

відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 5.2 – Характеристика робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷800
Висота простору для ніг, мм	730	не менше 600
Ширина простору для ніг, мм	660	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	470	400 ÷500
Ширина сидіння, мм	400	не менше 400
Глибина сидіння, мм	400	не менше 400
Висота поверхні спинки, мм	600	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷800

У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону).

За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений переносним вуглекислотним вогнегасником ВВК-5 .

Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

### **3.2.3 Навантаження та напруженість процесу праці**

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви тривалістю 15 хв через кожну годину роботи;

## **3.3 Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

### **3.3.1 Аналіз небезпечних та шкідливих факторів при розробці виробу**

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 3.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання, які встановлюють вимоги безпеки до обладнання

робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U = +220\text{В} \pm 5\%$ ;
- робочий струм  $I = 2\text{А}$ ;
- споживана потужність  $P = 600\text{ Вт}$ .

Таблиця 3.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Нормативні документи
Фізичні		
- підвищена температура поверхонь обладнання	експлуатація ЕОМ, серверного обладнання для роботи	[7]
- підвищена або знижена вологість повітря	-//-	[7]
- підвищена або знижена рухливість повітря	-//-	[7]
- підвищений рівень напруги електричної мережі	-//-	[10] [12]
- підвищений рівень статичної електрики	-//-	[10]
- підвищена напруженість електромагнітного поля	-//-	[9]
- недостатність природного світла	порушення умов праці (вимог до приміщень)	[9]
- недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	[10]
Психофізіологічні		

### Продовження таблиці 3.3 – Аналіз небезпечних і шкідливих виробничих факторів

-нервово-психічна перевантаження	Розумова робота над проектом	[2] [8]
- фізичні (статичне – сидіння)	порушення умов праці та організації робочого часу	[2]

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 [8]. За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом, яких можуть бути: принтер, сканер та інші джерела виділення багатьох хімічних речовин - напр., озону, оксидів азоту та аерозолів високодисперсних частинок тонера), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

#### 3.3.2 Пожежна безпека

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування. Небезпека загоряння пов'язана з особливістю комп'ютерів – із значною кількістю щільно розташованих на монтажній платі і блоках електронних вузлів і схем, електричних і комутаційних кабелів, резисторів, конденсаторів, напівпровідникових діодів і транзисторів. Надійна робота окремих елементів і мікросхем в цілому забезпечується тільки в певних інтервалах температури, вологості і при заданих електричних



параметрах. При відхиленні реальних умов експлуатації від розрахункових можуть виникнути пожежонебезпечні ситуації.

Для гасіння пожеж в офісному приміщенні пропонується використовувати порошкові або вуглекислотні вогнегасники, так як вони є універсальними.

Виникнення пожежі можливе, якщо на об'єкті є горючі речовини, окиснювач і джерела запалювання. Вірогідність пожежної небезпеки приймається значною, якщо ймовірна взаємодія цих трьох чинників. Горючими компонентами є: будівельні матеріали для акустичної і естетичної обробки приміщень, перегородки, підлоги, двері, ізоляція силових, сигнальних кабелів і т.д.

### **3.3.3 Електробезпека**

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три-провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві

рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

### **3.4 Гігієнічні вимоги до параметрів виробничого середовища**

#### **3.4.1 Мікроклімат**

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Іа. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають нормам [7] і наведені в табл. 5.4.

Таблиця 5.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С°	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-витяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [7]. Рівні позитивних і негативних іонів у повітрі мають відповідати [7]. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання.

Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату.

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

### 3.4.2 Освітлення

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає ДБН В.2.5-28:2015. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

#### Розрахунок освітлення

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \times S_n, \quad (5.1)$$

де  $S_b$  – площа віконних прорізів,  $\text{м}^2$ ;

$S_n$  – площа підлоги,  $\text{м}^2$ .

$$S_n = a \cdot b = 5 \cdot 3 = 15 \text{ м}^2,$$

$$S = 1/10 \cdot 15 = 1,5 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S=1,5 \text{ м}^2$ .

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 5 м, ширина 5 м, світильниками ЛПО2П, оснащеними лампами типа ЛБ (дві по 80 Вт) з світловим потоком 3200 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (5.2):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M}, \quad (5.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 15$  м<sup>2</sup>;

$Z$  – поправочний коефіцієнт світильника (1,1 для люмінесцентних ламп);

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 3200лм (для ЛБ-40-2).

Підставивши числові значення у формулу (5.2), отримуємо:

$$n = \frac{300 \times 15 \times 1.1 \times 1.5}{3200 \times 0.575 \times 2} = 2,018$$

Приймаємо освітлювальну установку, яка складається з 2-х світильників, які складаються з 2-х люмінесцентних ламп загальною потужністю 40 Вт, напругою – 220 В.

### **3.4.3 Вентилювання**

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

### **3.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:

а) якщо об'єм приміщення  $20 \text{ м}^3$ , то потрібно подати не менш як  $30 \text{ м}^3/\text{год}$  повітря;

б) якщо об'єм приміщення у межах від 20 до 40 м<sup>3</sup>, то потрібно подати не менш як 20 м<sup>3</sup>/год повітря;

в) якщо об'єм приміщення становить понад 40 м<sup>3</sup>, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

- зниження рівня шуму та вібрації:

а) у джерелі виникнення, шляхом застосування раціональних конструкцій, нових матеріалів і технологічних процесів;

б) звукоізолювання устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;

в) використання засобів індивідуального захисту).

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;

- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;

- не тягнути за мережевий кабель, щоб витягти вилку з розетки;

- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;

- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;

- не залишати включені електроприлади без нагляду;

- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;

- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Від ураження струмом застосовують різні електричні захисні засоби:

а) Ізолюючі – ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.

б) Основні – володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою.

в) Запобіжні – володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, при чому при використанні основних захисних засобів достатньо застосування одного запобіжного захисного засобу.

### 5.5.1 Розрахунок захисного заземлення

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом № [4], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача  $\eta$  – це відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів  $\eta_v$  в залежності від розміщення заземлювачів та їх кількості знаходиться в межах

0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача  $\eta_c$ .

Послідовність розрахунку:

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (5.3)$$

де  $R_{пр.з.}$  – опір природних заземлювачів;  $R_d$  – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_d$ .

Підставивши числові значення у формулу (5.3), отримуємо:

$$R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $\rho_{розр.}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $\rho_{розр.в.}$ , і горизонтальних  $\rho_{розр.г.}$ , Ом·м за формулою:

$$\rho_{розр.} = \Psi \cdot \rho \quad (5.4)$$

де  $\Psi$  – коефіцієнт сезонності для вертикальних заземлювачів і кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $\rho_{розр.в.} = 1,7$  і горизонтальних  $\rho_{розр.г.} = 5,5$  Ом·м.

$$\rho_{розр.в.} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{розр.г.} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_B$ , Ом, за (4.).

$$R_B = \frac{\rho_{розр.в.}}{2 \cdot \pi \cdot l_B} \cdot \left( \ln \frac{2 \cdot l_B}{d_{ст}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_B}{4 \cdot t - l_B} \right), \quad (5.5)$$

де  $l_B$  – довжина вертикального заземлювача (для труб – 2 – 3 м;  $l_B = 3$  м);

$d_{ст}$  – діаметр стержня (для труб – 0,03 – 0,05 м;  $d_{ст} = 0,05$  м);



$t$  – відстань від поверхні землі до середини заземлювача, яка визначається за ф. (5.6):

$$t = h_E + \frac{1_E}{2}, \quad (5.6)$$

де  $h_B$  – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м};$$

$$R_b = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

- 1) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_B$ :

$$n = \frac{2R_E}{R_D} = \frac{2 \times 18,5}{4} = 9,25 \quad (5.7)$$

$\Gamma$  визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки  $\eta_B = 0,57$  (табличне значення).

- 2) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $\eta_B$ , шт:

$$n = \frac{2 \cdot R_E}{R_D \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} \approx 16 \quad (5.8)$$

- 3) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_C$ , м:

$$l_C = 1,05 \cdot L_B \cdot (n_B - 1), \quad (5.9)$$

де  $L_B$  – відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$  м);  
 $n_B$  – необхідна кількість вертикальних заземлювачів.

$$l_C = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}.$$

Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_{\Gamma}$ , Ом:

$$R_{\Gamma} = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot 1_c} \cdot \ln \frac{2 \cdot 1_c^2}{d_{\text{см}} \cdot h_{\Gamma}}, \quad (5.10)$$

де  $d_{\text{см}}$  – еквівалентний діаметр смуги шириною  $b$ ,  $d_{\text{см}} = 0,95b$ ,  $b = 0,15$  м;

$h_{\Gamma}$  – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  – довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_{\Gamma} = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

4) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$  відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ .

Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$ .

Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг.}} = \frac{R_E \cdot R_{\Gamma}}{R_E \cdot \eta_c + R_{\Gamma} \cdot \eta_E \cdot \eta_E} \leq R_d, \quad (5.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{\text{заг.}} < 4$  Ом, а саме:

$$R_{\text{заг.}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинно накопичуватися сміття, непотрібний папір, мотлох та ін. речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі.

### **3.6 Висновки**

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника.

Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

## ВИСНОВКИ

У данному дипломному проекті на базі аналізу основних принципів прийому їжі, і дієт, був розроблений і реалізований веб-сервіс моніторингу денного харчового раціону, з використанням таких засобів як PHP, HTML, CSS і SQL. Був розроблений зручний інтерфейс для доступу до функцій сервісу і його даних.

Даний веб-сервіс може бути корисним для людей які стежать за своїм здоров'ям, харчовим раціоном а так само зацікавити таких, які раніше про це не замислювалися.

## ПЕРЕЛІК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Закон України «Про охорону праці»;
2. НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин»;
3. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»;
4. НПАОП 0.00-4.15-98 «Положення про розробку інструкцій з охорони праці»;
5. НПАОП 40.1-1.01-97 «Правила безпечної експлуатації електроустановок»;
6. НАПБ Б.02.005-2003 Типове положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України;
7. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» ;
8. ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»;
9. ГОСТ 12.1.044-89 «ССБТ. Вогнестійкість. Номенклатура показників і методи їх визначення»;
- 10.ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення, занулення».
- 11.ГОСТ 12.1.006-84 «ССБТ. Електромагнітні поля радіочастот»;
- 12.ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів. Норми якості електричної енергії в системах електропостачання загального призначення»;
- 13.ДБН В.2.5-28:2015 «Державні Будівельні Норми України. Природне і штучне освітлення»;
- 14.НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою».

15. Вікіпедія: вільна енциклопедія URL:<https://uk.wikipedia.org/wiki/HTML>  
(дата звернення: 11. 06. 2017)
16. Линн Бейли. Вивчаємо PHP і MySQL .Москва, Ексмо, 2010. 785 с.

## ДОДАТОК А

### Лістинг програми

#### Файл index.php:

```

<?php error_reporting(-1);
Error_Reporting(E_ALL & ~E_NOTICE);
include 'includes/header.php';
    include 'includes/testlog.php';
?>
<?php
include 'includes/functions.php';
//include 'includes/testlog.php';
if($log_OK==false) {
echo <<<EOD
<div class = "input_forms">
<form action='testreg.php' method='POST'>
<p>    Логин: <input name='login'></p>
<p>    Пароль:      <input name='password' type='password'> </p>
                <input type='submit' value='Отправить' class = "submit_button">
            </form>
            <p>Еще не зарегистрированы? Тогда вам
<a href="/register.php">сюда.</a> </p>
<div style="clear:both;"></div>

</div>
EOD;
} else {
    echo "hello $_COOKIE[login] </br>";

echo '<a href = "private_office.php">Личный кабинет</a></br>';
echo '<a href="logout.php">Выйти</a>';
}
include 'includes/footer.php';
?>

```

#### Файл logout.php:

```

<?php
setcookie('login', '', time() - 3600);
setcookie('userhash', '', time() - 3600);
unset($_COOKIE);
unset($_SESSION);
echo "<html><head><meta http-equiv='Refresh' content='0';
URL=index.php'></head></html>";
?>

```

#### Файл private\_office.php

```

<?php
Error_Reporting(E_ALL & ~E_NOTICE);

//include 'includes/testlog.php';
include 'includes/functions.php';
include 'includes/testlog.php';
$username = clean($_POST['username']);
$usersex = clean($_POST['usersex']);
$userheight = clean($_POST['userheight']);
$userweight = clean($_POST['userweight']);
$userage = clean($_POST['userage']);
$type_a_food_intake_assoc_array = array (
    "Завтрак" => "breakfast",

```

```

"Второй завтрак" =>"lunch",
"Обед" =>"dinner",
"Полдник" => "afternoon_snack",
"Ужин" => "supper"
);
$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]';");
    $user_data = mysqli_fetch_assoc($user_data);
if($user_data['userdiet']==0) $user_data['userdiet'] = "Нет спец. диеты";
if(!$log_OK) header('Location: index.php');
include 'includes/header.php';
?>
<div class = "content">
<div class = "content_menu">
<ul class = "menu-th">
<li><a href="/private_office.php?user_info=1">Личные данные</a></li>

<li><a href="/private_office.php?ration=1">Дневник</a>
    <ul>
        <li><a href="/private_office.php?ration=1">Внести запись</a></li>
        <li><a href="/private_office.php?history_of_eat=1">Вывести
историю</a></li>

    </ul>
</li>
<li><a href="/private_office.php?diets=1">Диеты</a></li>
</ul>
</div>
</div>
<?php

$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]'");
$user_data = mysqli_fetch_assoc($user_data);
if (isset($_POST['dietname'])) $diet_info = mysqli_query($link, "SELECT * FROM `diet`
WHERE `dietname` = '$_POST[dietname]'");
if($diet_info) $diet_info = mysqli_fetch_assoc($diet_info);
if(isset($_POST['select_date'])&&!empty($_POST['select_date'])){ $current_day =
$_POST['select_date'];} else{ $current_day = date("Y-m-d");}
if(empty($username)) $username = $user_data['username'];
if(empty($usersex)) $usersex = $user_data['usersex'];
if(empty($userheight)) $userheight = $user_data['userheight'];
if(empty($userweight)) $userweight = $user_data['userweight'];
if(empty($userage)) $userage = $user_data['userage'];
if(empty($_POST['dietname'])) $diet_info['iddiet'] = $user_data['userdiet'];

mysqli_query($link, "UPDATE `users` SET `username` = '$username', `usersex` =
'$usersex', `userheight` = '$userheight', `userweight` = '$userweight', `userage` =
'$userage', `userdiet` = '$diet_info[iddiet]' WHERE `login` = '$_COOKIE[login]'");

if($_GET['user_info'] == 1) {
    if(empty($_GET['change_data'])) {

        $getdietname = mysqli_query($link, "SELECT * FROM `diet` WHERE `iddiet` =
$user_data[userdiet]");
        if($getdietname) {$getdietname = mysqli_fetch_assoc($getdietname);} else
{$getdietname['dietname'] = "диеты нет (обычный рацион)";}

        echo <<<EOD
<div class = "in_c">
    <p> Имя: $user_data[username] </p>

```



```

        <p> Пол: $user_data[usersex] </p>
        <p> Рост: $user_data[userheight] см. </p>
        <p> Вес: $user_data[userweight] кг.</p>
        <p> Возраст $user_data[userage] </p>
        <p> Диета: $getdietname[dietname] </p>
        <p> <a href = "private_office.php?change_data=1">Изменить</a> </p>
    </div>
EOD;
    }

    unset($_GET['user_info']);
}
if($_GET['change_data'] == 1) {
    echo '
    <div class = "in_c">
    <form action="private_office.php?user_info=1" method="POST">
    <p> Имя: <input type="text" name = "username"></p>
    <p> Пол: <select name = "usersex"> </p>
    <option>Мужской</option>
    <option>Женский</option>
    </select>
    </br>
    <p> Рост: <input type="text" name = "userheight"></p>
    <p> Вес: <input type = "text" name = "userweight"></p>
    <p> Возраст(полных лет): <input type = "text" >

';
    $get_diets_names = mysqli_query($link, "SELECT * FROM `diet` ");
    $i=0;
    $d_name_arr = array();
    while($arr = mysqli_fetch_assoc($get_diets_names))
    {
        $d_name_arr[$i] = $arr['dietname'];
        $i++;
    }
    echo "<p>Текущая диета: <select name = 'dietname' >
    <option>Нет спец. диеты</option>";
    foreach($d_name_arr as $value)
    {
        echo "<option>$value</option>";
    }
    ?>
    </select> </p>
    <input type="submit" value="Отправить">
    </form>
    </div>
    <?php
    }
    unset($_GET['change_data']);
    if($_GET['ration'] == 1) {
        //инициализация дня
        $exist_day_usr = mysqli_fetch_assoc(mysqli_query($link, "SELECT * FROM `routine_day`
        WHERE `users_idusers` = '$user_data[idusers]' AND `day` = '$current_day'"));
        if (empty($exist_day_usr))
        {
            mysqli_query($link, "INSERT INTO `routine_day` (`day`, `users_idusers`) VALUES
            ('$current_day', '$user_data[idusers]') ; ");
        }

        $type_a_food_intake = $type_a_food_intake_assoc_array[$_POST['type_of_food_intake']];
        //debug

```

```

$select_date = $_POST['select_date'];
$product_names_arr = array();
$product_weight_arr = array();
$pattern = '_';
if(isset($_POST)) {
    $iw = 0;
    $in = 0;
    $delim = 1;
    foreach ($_POST as $key => $value) {

        if(stripos($key, "item") !== false) {
            $product_names_arr[$iw] = $value;
            $iw++;
        }
        if(stripos($key, "weight") !== false) {
            $product_weight_arr[$in] = $value;
            $in++;
        }
        $delim++;
        $i++;
    }

    $delim = 1;
    $i = 0;
}
foreach($product_names_arr as $value)
{
    echo "$value <br />";
}
foreach($product_weight_arr as $value)
{
    echo "$value <br />";
}
$id_user = mysqli_fetch_assoc(mysqli_query($link, "SELECT `idusers` FROM `users`
WHERE `idusers` = '$user_data[idusers]'"));
$id_user = $id_user['idusers'];

//будет генерироваться строка запроса в бд
$i_weight = 0;
foreach ($product_names_arr as $value) {
    $product_calory = mysqli_fetch_assoc(mysqli_query($link, "SELECT
`productcalory` FROM `products` WHERE `productname` = '$value';"));
    $total_calory = (((float)$product_weight_arr[$i_weight] *
(float)$product_calory['productcalory']) / 100.0);

    $query_string = "INSERT INTO `$type_a_food_intake` (`users_idusers`,
`routine_day_day`, `product`, `product_calory`, `product_weight`) VALUES ($id_user,
'$current_day', '$value', '$total_calory', '$product_weight_arr[$i_weight]' );";
    echo "SQL row : ".$query_string."<br>";
    $qytt = mysqli_query($link, $query_string);
    echo "<br> SQL test ".$qytt;
    $i_weight++;
}
$i_weight = 0;
unset($product_weight_arr);
unset($product_names_arr);
$summary = mysqli_query($link, "SELECT `product_calory` FROM `breakfast` WHERE
`routine_day_day` = '$current_day' AND `users_idusers` = '$user_data[idusers]'");
$summary = summ_calory($summary);
echo<<<EOF
<script type="text/javascript">

```

```

var rr = 1;
function additem() {

    form = document.forms[1];
    var arr = form.elements;
    div_elem = document.getElementById('ration_add_div');
div_elem.innerHTML = div_elem.innerHTML + ' Продукт: <input type="text" name="item_'
+ rr + '" onfocus="search_product(this.name,this.value);" > вес: <input type="text"
name="weight_' + rr + '"> <br>';

                                rr++;

    }
</script>

<form name="day_ration_add" action="private_office.php?ration=1" method="POST">
<p>Список продуктов на: <select
name="type_of_food_intake"><option>Завтрак</option><option>Второй
завтрак</option><option>Обед</option><option>Полдник</option><option>Ужин</option></s
elect> числа: <input type="date" name = "select_date">
<div class = "input_forms">
<div id = "ration_add_div" onload="XmlHttp()">
Продукт: <input type="text" name="item_0" onkeyup="get_prod(this.value)" id="tags"
/> вес: <input type="text" name="weight_0"> <br>
</div>
<div onclick="additem();" style="border-style:solid; border-radius: 2px; border-
color:black; display: inline-block; border-width: 1px; font-size:10pt; background-
color: #fdfdfd; cursor: pointer; padding:1px;">Добавить продукт</div>
<input type="submit" value='Отправить' ></input>
</form>
</div>

<script>
prod_arr = "";
function get_prod(str) {
    if (str.length == 0) {
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                arr = this.responseText;
                // alert (arr);
                prod_arr = arr.split(',');
            }
        };
        $( function() {
            var availableTags = prod_arr;
            $( "#tags" ).autocomplete({
                source: availableTags
            });
        });
    }
    xmlhttp.open("GET", "search_prod_ajax.php?search_str=" + str, true);
    xmlhttp.send();
}
</script>
EOF;
}

```

```

if($_GET['history_of_eat'] == 1)
{
echo <<<EOF

<form name="day_ration_add" action="private_office.php?history_of_eat=1"
method="POST" class="history_form">
Выбор даты: <input type="date" name = "select_date"><br> Выбор временного промежутка:
<select name="type_of_food_intake"><option>Целый
день</option><option>Завтрак</option><option>Второй
завтрак</option><option>Обед</option><option>Полдник</option><option>Ужин</option></s
elect>
<br>
<input type="submit" value="Получить данные">
</form>
EOF;

    $history_sql_arr['breakfast'] = "SELECT * FROM `breakfast` WHERE
`routine_day_day`='$_current_day';";
    $history_sql_arr['lunch'] = "SELECT * FROM `lunch` WHERE
`routine_day_day`='$_current_day';";
    $history_sql_arr['dinner'] = "SELECT * FROM `dinner` WHERE
`routine_day_day`='$_current_day';";
    $history_sql_arr['afternoon_snack'] = "SELECT * FROM `afternoon_snack` WHERE
`routine_day_day`='$_current_day';";
    $history_sql_arr['supper'] = "SELECT * FROM `supper` WHERE
`routine_day_day`='$_current_day';";

    if(isset($_POST['select_date']) && isset($_POST['type_of_food_intake'])) {

        $type_of_food_intake = $_POST['type_of_food_intake'];
        if($type_of_food_intake == "Целый день")
        {
            foreach ($type_a_food_intake_assoc_array as $key => $value) {
                $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$key]` WHERE `routine_day_day`='$_current_day'");

                $empty_cheker = mysqli_fetch_assoc(mysqli_query($link,"SELECT *
FROM `$type_a_food_intake_assoc_array[$key]` WHERE
`routine_day_day`='$_current_day'"));

                if($history_of_food&&$empty_cheker){

                    echo " <div class='history_form'>
<br>Данные на: $_current_day - $key";
                    while($data_array = mysqli_fetch_assoc($history_of_food))
                    {

                        echo "<div class = 'product_report'>
Продукт: $data_array[product] <br>
Вес: $data_array[product_weight] гр. <br>
Калорийность: $data_array[product_calory]
</div>";

                    }
                    $summ_calory = summ_calory(mysqli_query($link,"SELECT *
FROM `$value` WHERE `routine_day_day`='$_current_day';"));
                    echo "Всего калорий: $summ_calory </div>";

                }

            }

        }

    }
}

```

```

        $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$type_of_food_intake]` WHERE
`routine_day_day`='$current_day'");

        if($history_of_food){

            echo " <div class='history_form'>
<br>Данные на: $current_day - $type_of_food_intake";
            while($breakfast_array = mysqli_fetch_assoc($history_of_food))
            {
                echo "<div class = 'product_report'>
                Продукт: $breakfast_array[product] <br>
                Вес: $breakfast_array[product_weight] гр. <br>
                Калорийность: $breakfast_array[product_calory]
                </div>";
            }
            $summ_calory = summ_calory(mysqli_query($link,"SELECT *
FROM `breakfast` WHERE `routine_day_day`='$current_day'"));
            echo "Всего калорий: $summ_calory </div>";

        }
    }
    unset($_POST['select_date']);
}
if($_GET['diets'] == 1)
{
    $get_info_of_diets = mysqli_query($link, "SELECT * FROM `diet` ");
    echo '<div id="diets_div">';
    if($get_info_of_diets) {
        while($d_row = mysqli_fetch_assoc($get_info_of_diets))
        {
            echo "<h3>Название диеты:<span class='diet_span'>
$d_row[dietname]</span></h3>";
            echo "<p> $d_row[dietdescription]</p>";

        }
    } else {
        echo "Инофрмация о диетах отсутствует!";
    }
    echo "</div>";

}
?>
</div>
</div>
</div>
<?php
include 'includes/footer.php';
?>

```

### Файл register.php:

```

<?php
include 'includes/functions.php';
include 'includes/header.php';
include 'includes/testlog.php';
$error_flag = false;
$error = array();
if (isset($_POST['reset'])) unset($_POST);
if (!empty($_POST)) {
    $_POST['login'] = clean($_POST['login']);
}

```

```

if (!check_length($_POST['login'], 2, 20)) {$error[0] .= "Длина логина должна быть
от 2 до 20 символов. "};
$_POST['password'] = clean($_POST['password`']);
if (!check_length($_POST['password'], 8, 20)) {$error[2] .= "Длина пароля должна
быть от 8 до 20 символов. "};
$email = mysqli_query($link, "SELECT * FROM `users` WHERE `email` = \"$_POST[email]\"
"); //Запит до бази даних для перевірки на зайнятість email-а
$email = mysqli_fetch_assoc($email);
$_POST['email'] = clean($_POST['email']);
if ($email['email'] == $_POST['email']) {$error[1]="данный email уже знят !";}

if(empty($error)) {

    mysqli_query($link, "INSERT INTO `users` (`idusers`, `login`, `password`,
`email`) VALUES(\"\", \"$_POST[login]\", \"$_POST[password]\", \"$_POST[email]\");

    echo '<script> window.location.href = "/index.php"</script>';

}
}

?>

<div class="input_forms">
<form action="register.php" method="POST">
<span class = "error_class"><?php echo $error[0]; ?></span>
<p>Логин: <input type="text" name="login" value = "<? echo $ _POST['login']
?>"></p>
<span class="error_class"><?php echo $error[1]; ?></span>
<p> e-mail: <input type="text" name="email" value = "<? echo $_POST['email'] ?>">
</p>
<span class = "error_class"><?php echo $error[2]; ?></span>
<p>Пароль: <input type="password" name="password" value = "<? echo $_POST['password']
?>"></p>
<input type='submit' value='Отправить'>
<input type='submit' value='Очистить' onclick = "var form = document.forms[0]; for
(var i = 0; i<3; i++) {form.elements[i].value = '' ;}" name = "reset">
</form>
</div>

<?php
include'includes/footer.php';
?>

```

### Файл db\_connect.php:

```

<?php
$link = mysqli_connect("127.0.0.1", "root", "", "eat_monitoring");

if (!$link) {
    echo "Ошибка: Невозможно установить соединение с MySQL." . PHP_EOL;
    echo "Код ошибки errno: " . mysqli_connect_errno() . PHP_EOL;
    echo "Текст ошибки error: " . mysqli_connect_error() . PHP_EOL;
    exit;
}

?>

```

**Файл footet.php:**

```

</div>
<div id="footer">
<div class = "footer_left">
Веб-сервіс моніторингу денного харчового раціону. Сайт для дипломного проекту
студента гр. КІ-13ад. Коннова В.І.
</div>

<div class = "footer_right">
Почта: vic-321@mail.ru <br>
Телефон: +38066454599 <br>
</div>
<div style="clear:both;"></div>
</div>

</body>
</html>

```

**Файл functions.php:**

```

<?php
function clean($value = "") {
    $value = trim($value);
    $value = stripslashes($value);
    $value = strip_tags($value);
    $value = htmlspecialchars($value);

    return $value;
}

function check_length($value = "", $min, $max) {
    $result = (mb_strlen($value) < $min || mb_strlen($value) > $max);
    return !$result;
}

function summ_calory($result){
if($result) {

    while ($row = mysqli_fetch_assoc($result)) { // Получаем строки из результата
        $summ += $row['product_calory'];
    }
}
    return $summ;
}
?>

```

**Файл header.php:**

```

<?php
session_start();
header('Content-Type: text/html; charset=utf-8');

?>
<!DOCTYPE html>
<html>
<head>
<title><?php echo $page_name; ?></title>
<link rel="stylesheet" href="src/styles/base_style.css" type="text/css" />

```

```

<link rel="stylesheet" href="src/styles/awesomplete.css" />
<!--<script src="src/scripts/jquery-3.2.1.min.js."></script> -->
<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<link rel="stylesheet" href="/resources/demos/style.css">
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
</head>
<body>
<div id="main">

<div id="head">
    <a href ="/?main=1"></a>

        <div id="poisk">
            <form action = "search.php" method="POST">

                </form>
            </div>
        </div>
<div id="container">
    <nav id="menu1">
        <ul>
            <li>
                <?php
if($log_OK) {echo ' <li><a href="private_office.php?user_info=1" >Личный
кабинет</a>';}
                ?>

            </li>
            <li><a href="about.php" >О сервисе</a></li>
            <li><a href="contacts.php">Контакты</a></li>

        </ul>
    </nav>
</div>

```

### Файл testlog.php:

```

<?php
include'db_connect.php';
if (isset($_COOKIE['login']) && isset($_COOKIE['userhash'])) {
$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]';");
$user_data = mysqli_fetch_assoc($user_data);
$log_OK = false;
if ($user_data['userhash'] == $_COOKIE['userhash'] && $user_data['login'] ==
$_COOKIE['login']) {$log_OK = true;
}
?>

```

### Файл base\_style.css:

```

body {
    background-image: url(/src/333_fabrik-len.jpg) ;
    margin:0;
}
#main{

```



```

background-color:#f5f5f5;
margin-top:0;
height:100%;
display:block;
margin: 0 auto;
width: 80%;
min-width: 1000px;
box-shadow: 0 0 10px rgba(0,0,0,0.5);
font-family: tahoma;
}
#head{
background-color:#E8E8E8;
height: auto;
}
#container{
height:auto;
margin:0px auto;
z-index:100;
}
#menu1{
position:relative;
display:block;
width:100%;
height:auto;
z-index:10;
}
#menu1 ul{
position:relative;
display:block;
margin:0px;
padding:0px;
width:100%;
height:auto;
list-style:none;
background:#696969;
}
#menu1 > ul::after{
display:block;
width:100%;
height:0px;
clear:both;
content:" ";
}
#menu1 ul li{
position:relative;
display:block;
float:left;
width:33.33%;
height:auto;
text-align:center;
}
#menu1 ul li a{
display:block;
padding:9px 25px 0px 25px;
font-size:14px;
}

```

```

        font-family:Arial, sans-serif;
        color:#FFFFFFE0; /* Цвет текста меню */
        line-height:1.3em;
        text-decoration:none;
        font-weight:bold;
        text-transform:uppercase;
        height:36px;
        box-sizing:border-box;
    }
    #menu1 ul li > a:hover, #menu1 ul li:hover > a{
        background:#BEBEBE;
        color:#FFFFFFE0;
    }
    #menu1 ul li ul{
        position:absolute;
        top:36px;
        left:0px;
        display:none;
        width:100%;
        background:#696969;
    }
    #menu1 ul li:hover ul{display:block;}
    #menu1 ul li ul li{
        float:none;
        width:100%;
    }
    #menu1 ul li ul li a{
        display:block;
        text-transform:none;
        height:auto;
        padding:7px 25px;
        width:100%;
        box-sizing:border-box;
        border-top:1px solid #ffffff;
    }
    #menu1 ul li ul li:first-child a{border-top:0px;}
    #menu1 ul li ul li a:hover{
        background:#BEBEBE;
        color:#FFFFFFE0;
    }
}

#naiti {
    position:absolute;
    cursor: pointer;
    background: url(poisk.png) no-repeat 40% 40%;
    border: none;
    width: 25px;
    height: 20px;
    top: 12%;
    right:0%;
    outline:none;
}

div.input_forms {
    background-color:#FDFFD4;
    width: 65%;
    margin: auto;
    margin-top: 40px;
    border-style: dashed;
    border-color: #666666;
}

```

```

        border-width:1px;
        text-align: left;
        padding:10px;
    }
    .input_forms p input {
        font-family: consolas;
        float: right;

        margin-right: 45%;
    }

    .input_forms p {

    }

    .submit_button {

    }

    .content {
        height:100%;
        display:block;
        margin: 0 auto;
        width: 70%;
        min-width: 500px;
        border-style: dashed;
        border-color: #666666;
        border-width:1px;
        padding: 3px;
        background-color: #fbf9f9;
        border-top: none;

    }

    .in_c {
        margin-left: 20px;
        font-size: 11pt;
    }
    .content_menu {
        box-shadow: 0 0 10px rgba(0,0,0,0.5);
    }
    ul.menu-th {
        list-style: none;
        margin: 0;
        padding-left: 0;
        margin-top:2px;
        background:#00A442;
        height: 50px;
    }

    .menu-th a {
        text-decoration: none;
        background:#00A442;
        color:#fff;
        padding:0px 15px;
        font-family: arial;
        line-height:50px;
        display: block;

```

```

border-right: 1px solid #007742;
-moz-transition: all 0.3s 0.01s ease;
-o-transition: all 0.3s 0.01s ease;
-webkit-transition: all 0.3s 0.01s ease;

}
.menu-th a:hover {
  background:#00D355;
}
.menu-th li {
  float:left;
  position:relative;
}

.menu-th li > ul {
  position:absolute;
  top:25px;
  display:none;
}

.menu-th li:hover > ul {
  display:block;
  width:200px;
  list-style:none;
  padding-left:0px;
  padding-top:20px;
  z-index: 9999;
}
.menu-th li:hover > ul > li {
  float:none;
}

#footer {

  background-color:#070707;
  opacity: 0.7;
  margin-top:0;
  height:100%;
  display:block;
  margin: 0 auto;
  width: 80%;
  min-width: 1000px;
  box-shadow: 0 0 10px rgba(0,0,0,0.5);
  font-family: tahoma;

}

#footer .footer_left {
  float:left;
  width:50%;
  font-size: 12pt;
  border-right: 1px dotted white;
  color:white;

  padding: 20px;
}

```

```

#footer .footer_right {
    float:left;
    text-align: left;
    font-size: 12pt;
    width: 20%;
    color:white;

    padding: 20px;
}

.product_report {
    border-bottom-style: solid;
    border-bottom-width: 2px;
    border-bottom-color: #555555;
    margin :15px;
    margin-top:5px;
    margin-bottom: 0px;

    padding-bottom: 10px;
}

.history_form {
    margin :15px;
}

.autocomplete-suggestions {
    text-align: left; cursor: default; border: 1px solid #ccc; border-top: 0;
background: #fff; box-shadow: -1px 1px 3px rgba(0,0,0,.1);

    position: absolute; display: none; z-index: 9999; max-height: 254px; overflow:
hidden; overflow-y: auto; box-sizing: border-box;
}
.autocomplete-suggestion { position: relative; padding: 0 .6em; line-height: 23px;
white-space: nowrap; overflow: hidden; text-overflow: ellipsis; font-size: 1.02em;
color: #333; }
.autocomplete-suggestion b { font-weight: normal; color: #1f8dd6; }
.autocomplete-suggestion.selected { background: #f0f0f0; }

.diet_span {
    text-decoration:underline
}

#diets_div {
    margin:30px;
    font-size:11pt;
}

.error_class {

    font-size: 10pt;
    color:red;
    font-style: italic;
}

```

**Файл private\_office.php:**

```

<?php
Error_Reporting(E_ALL & ~E_NOTICE);

//include 'includes/testlog.php';
include 'includes/functions.php';
include 'includes/testlog.php';
$username = clean($_POST['username']);
$usersex = clean($_POST['usersex']);
$userheight = clean($_POST['userheight']);
$userweight = clean($_POST['userweight']);
$userage = clean($_POST['userage']);
$type_a_food_intake_assoc_array = array (
    "Завтрак" => "breakfast",
    "Второй завтрак" => "lunch",
    "Обед" => "dinner",
    "Полдник" => "afternoon_snack",
    "Ужин" => "supper"
);
$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]';");
    $user_data = mysqli_fetch_assoc($user_data);
if($user_data['userdiet']==0) $user_data['userdiet'] = "Нет спец. диеты";
if(!$log_OK) header('Location: index.php');
include 'includes/header.php';
?>
<div class = "content">
<div class = "content_menu">
<ul class = "menu-th">
<li><a href="/private_office.php?user_info=1">Личные данные</a></li>

<li><a href="/private_office.php?ration=1">Дневник</a>
    <ul>
        <li><a href="/private_office.php?ration=1">Внести запись</a></li>
        <li><a href="/private_office.php?history_of_eat=1">Вывести
историю</a></li>

    </ul>
</li>
<li><a href="/private_office.php?diets=1">Диеты</a></li>
</ul>
</div>
<?php

$user_data = mysqli_query($link, "SELECT * FROM `users` WHERE `login` =
'$_COOKIE[login]");
$user_data = mysqli_fetch_assoc($user_data);
if (isset($_POST['dietname'])) $diet_info = mysqli_query($link, "SELECT * FROM `diet`
WHERE `dietname` = '$_POST[dietname]");
if($diet_info) $diet_info = mysqli_fetch_assoc($diet_info);
if(isset($_POST['select_date'])&&!empty($_POST['select_date'])){$current_day =
$_POST['select_date'];} else{ $current_day = date("Y-m-d");}
if(empty($username)) $username = $user_data['username'];
if(empty($usersex)) $usersex = $user_data['usersex'];
if(empty($userheight)) $userheight = $user_data['userheight'];
if(empty($userweight)) $userweight = $user_data['userweight'];
if(empty($userage)) $userage = $user_data['userage'];
if(empty($_POST['dietname'])) $diet_info['iddiet'] = $user_data['userdiet'];

```

```

mysqli_query($link, "UPDATE `users` SET `username` = '$username', `usersex` =
'$usersex', `userheight` = '$userheight', `userweight` = '$userweight', `userage` =
'$userage', `userdiet` = '$diet_info[iddiet]' WHERE `login` = '$_COOKIE[login]'");

if($_GET['user_info'] == 1) {
    if(empty($_GET['change_data'])) {

        $getdietname = mysqli_query($link, "SELECT * FROM `diet` WHERE `iddiet` =
$user_data[userdiet]");
        if($getdietname) {$getdietname = mysqli_fetch_assoc($getdietname);} else
{$getdietname['dietname'] = "диеты нет (обычный рацион)";}

        echo <<<EOD
<div class = "in_c">
    <p> Имя: $user_data[username] </p>
    <p> Пол: $user_data[usersex] </p>
    <p> Рост: $user_data[userheight] см. </p>
    <p> Вес: $user_data[userweight] кг.</p>
    <p> Возраст $user_data[userage] </p>
    <p> Диета: $getdietname[dietname] </p>
    <p> <a href = "private_office.php?change_data=1">Изменить</a> </p>
</div>

EOD;
    }

    unset($_GET['user_info']);
}
if($_GET['change_data'] == 1) {
    echo '
    <div class = "in_c">
<form action="private_office.php?user_info=1" method="POST">
<p> Имя: <input type="text" name = "username"></p>
<p> Пол: <select name = "usersex"> </p>
<option>Мужской</option>
<option>Женский</option>
</select>
    </br>
<p> Рост: <input type="text" name = "userheight"></p>
<p> Вес: <input type = "text" name = "userweight"></p>
<p> Возраст(полных лет): <input type = "text" >

';
$get_diets_names = mysqli_query($link, "SELECT * FROM `diet` ");
    $i=0;
    $d_name_arr = array();
    while($arr = mysqli_fetch_assoc($get_diets_names))
    {
        $d_name_arr[$i] = $arr['dietname'];
        $i++;
    }
    echo "<p>Текущая диета: <select name = 'dietname' >
    <option>Нет спец. диеты</option>";
    foreach($d_name_arr as $value)
    {
        echo "<option>$value</option>";
    }

?>
</select> </p>

```

```

        <input type="submit" value="Отправить">

</form>
</div>

<?php
}
unset($_GET['change_data']);
if($_GET['ration'] == 1) {

        //инициализация дня

    $exist_day_usr = mysqli_fetch_assoc(mysqli_query($link, "SELECT * FROM `routine_day`
    WHERE `users_idusers` = '$user_data[idusers]' AND `day` = '$current_day'"));
    if (empty($exist_day_usr))
    {
        mysqli_query($link, "INSERT INTO `routine_day` (`day`, `users_idusers`) VALUES
        ('$current_day', '$user_data[idusers]') ; ");
    }

    $type_a_food_intake = $type_a_food_intake_assoc_array[$_POST['type_of_food_intake']];
    //debug
    $select_date = $_POST['select_date'];
    $product_names_arr = array();
    $product_weight_arr = array();
    $pattern = '_';
    if(isset($_POST)) {
        $iw = 0;
        $in = 0;
        $delim = 1;
        foreach ($_POST as $key => $value) {

            if(strpos($key, "item") !== false) {
                $product_names_arr[$iw] = $value;
                $iw++;
            }
            if(strpos($key, "weight") !== false) {
                $product_weight_arr[$in] = $value;
                $in++;
            }
            $delim++;
            $i++;
        }

        $delim = 1;
        $i = 0;
    }
    foreach($product_names_arr as $value)
    {
        echo "$value <br />";
    }
    foreach($product_weight_arr as $value)
    {
        echo "$value <br />";
    }
    $id_user = mysqli_fetch_assoc(mysqli_query($link, "SELECT `idusers` FROM `users`
    WHERE `idusers` = '$user_data[idusers]'"));
    $id_user = $id_user['idusers'];

```



```

//будет генерироваться строка запроса в бд
$i_weight = 0;
foreach ($product_names_arr as $value) {
    $product_calory = mysqli_fetch_assoc(mysqli_query($link, "SELECT
`productcalory` FROM `products` WHERE `productname` = '$value';"));
    $total_calory = (((float)$product_weight_arr[$i_weight] *
(float)$product_calory['productcalory']) / 100.0);

    $query_string = "INSERT INTO `$type_a_food_intake` (`users_idusers`,
`routine_day_day`, `product`, `product_calory`, `product_weight`) VALUES ($id_user,
'$current_day', '$value', '$total_calory', '$product_weight_arr[$i_weight]' );";
    echo "SQL row : ".$query_string."<br>";
    $qytt = mysqli_query($link, $query_string);
    echo "<br> SQL test ".$qytt;
    $i_weight++;
}
$i_weight = 0;
unset($product_weight_arr);
unset($product_names_arr);

$summary = mysqli_query($link, "SELECT `product_calory` FROM `breakfast` WHERE
`routine_day_day` = '$current_day' AND `users_idusers` = '$user_data[idusers]'");

$summary = summ_calory($summary);

echo<<<EOF
<script type="text/javascript">

var rr = 1;
function additem() {

    form = document.forms[1];
    var arr = form.elements;
    div_elem = document.getElementById('ration_add_div');

    div_elem.innerHTML = div_elem.innerHTML + ' Продукт: <input type="text" name="item_'
+ rr + '" onfocus="search_product(this.name,this.value);" > вес: <input type="text"
name="weight_' + rr + '"> <br>';

    rr++;

}

</script>

<form name="day_ration_add" action="private_office.php?ration=1" method="POST">
<p>Список                продуктов                на:                <select
name="type_of_food_intake"><option>Завтрак</option><option>Второй
завтрак</option><option>Обед</option><option>Полдник</option><option>Ужин</option></s
elect> числа: <input type="date" name = "select_date">
<div class = "input_forms">
<div id ="ration_add_div" onload="XmlHttpRequest">

```

```

Продукт: <input type="text" name="item_0" onkeyup="get_prod(this.value)" id="tags"
/> вес: <input type="text" name="weight_0"> <br>
</div>
<div onclick="additem();" style="border-style:solid; border-radius: 2px; border-
color:black; display: inline-block; border-width: 1px; font-size:10pt; background-
color: #fdfdfd; cursor: pointer; padding:1px;">Добавить продукт</div>
<input type="submit" value='Отправить' ></input>
</form>
</div>

<script>
prod_arr = "";
function get_prod(str) {
    if (str.length == 0) {

        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                arr = this.responseText;
                // alert (arr);
                prod_arr = arr.split(',');
            }
        };
        $( function() {
            var availableTags = prod_arr;
            $( "#tags" ).autocomplete({
                source: availableTags
            });
        } );

        }
    };
    xmlhttp.open("GET", "search_prod_ajax.php?search_str=" + str, true);
    xmlhttp.send();
}

</script>
EOF;
}

if($_GET['history_of_eat'] == 1)
{
echo <<<EOF

<form name="day_ration_add" action="private_office.php?history_of_eat=1"
method="POST" class="history_form">
Выбор даты: <input type="date" name = "select_date"><br> Выбор временного промежутка:
<select name="type_of_food_intake"><option>Целый
день</option><option>Завтрак</option><option>Второй
завтрак</option><option>Обед</option><option>Полдник</option><option>Ужин</option></s
elect>
<br>
<input type="submit" value="Получить данные">
</form>

```

```

EOF;

    $history_sql_arr['breakfast'] = "SELECT * FROM `breakfast` WHERE
`routine_day_day`='$current_day';";
    $history_sql_arr['lunch'] = "SELECT * FROM `lunch` WHERE
`routine_day_day`='$current_day';";
    $history_sql_arr['dinner'] = "SELECT * FROM `dinner` WHERE
`routine_day_day`='$current_day';";
    $history_sql_arr['afternoon_snack'] = "SELECT * FROM `afternoon_snack` WHERE
`routine_day_day`='$current_day';";
    $history_sql_arr['supper'] = "SELECT * FROM `supper` WHERE
`routine_day_day`='$current_day';";

    if(isset($_POST['select_date']) && isset($_POST['type_of_food_intake'])) {

        $type_of_food_intake = $_POST['type_of_food_intake'];
        if($type_of_food_intake == "Целый день")
        {
            foreach ($type_a_food_intake_assoc_array as $key => $value) {
//$key pycc
                $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$key]` WHERE `routine_day_day`='$current_day'");

                $empty_cheker = mysqli_fetch_assoc(mysqli_query($link,"SELECT *
FROM
`$type_a_food_intake_assoc_array[$key]` WHERE
`routine_day_day`='$current_day'"));

                if($history_of_food&&$empty_cheker){

                    echo " <div class='history_form'>
<br>Данные на: $current_day - $key";
                    while($data_array = mysqli_fetch_assoc($history_of_food))
                    {
                        echo "<div class = 'product_report'>
Продукт: $data_array[product] <br>
Вес: $data_array[product_weight] гр. <br>
Калорийность: $data_array[product_calory]
</div>";
                    }
                    $summ_calory = summ_calory(mysqli_query($link,"SELECT *
FROM `$value` WHERE `routine_day_day`='$current_day';"));
                    echo "Всего калорий: $summ_calory </div>";
                }
            }
        }

        $history_of_food = mysqli_query($link,"SELECT * FROM
`$type_a_food_intake_assoc_array[$type_of_food_intake]` WHERE
`routine_day_day`='$current_day'");

        if($history_of_food){

            echo " <div class='history_form'>
<br>Данные на: $current_day - $type_of_food_intake";
            while($breakfast_array = mysqli_fetch_assoc($history_of_food))
            {

```

```

        echo "<div class = 'product_report'>
        Продукт: $breakfast_array[product] <br>
        Вес: $breakfast_array[product_weight] гр. <br>
        Калорийность: $breakfast_array[product_calory]
        </div>";
    }
    $summ_calory = summ_calory(mysql_query($link,"SELECT *
FROM `breakfast` WHERE `rountine_day_day`='$current_day'"));
    echo "Всего калорий: $summ_calory </div>";

}

}

unset($_POST['select_date']);
}

if($_GET['diets'] == 1)
{
    $get_info_of_diets = mysql_query($link, "SELECT * FROM `diet` ");

    echo '<div id="diets_div">';
    if($get_info_of_diets) {
        while($d_row = mysql_fetch_assoc($get_info_of_diets))
        {
            echo "<h3>Название диеты:<span class='diet_span'>
$d_row[dietname]</span></h3>";

            echo "<p> $d_row[dietdescription]</p>";

        }
    } else {
        echo "Инофрмация о диетах отсутствует!";
    }
    echo "</div>";

}
?>
</div>
</div>
</div>
<?php
include 'includes/footer.php';
?>

```

### Файл search\_prod\_ajax.php:

```

<?php
header('Content-Type: text/plain; charset=utf-8');

include'includes/db_connect.php';
$avaible_products = array();

$req_str = $_REQUEST['search_str'];

if($req_str != "")
{

```

```

        if($query = mysqli_query($link,"SELECT * FROM `products` WHERE `productname`
RLIKE \"\$req_str\" LIMIT 10 "))
    {
        while($avaible_products = mysqli_fetch_assoc($query))
            $str .= "$avaible_products[productname], ";
    }
    $str = substr($str, 0, (mb_strlen($str) - 2) );
}
echo $str;
unset($str);
unset($req_str);
unset($avaible_products);
?>

```

### Файл testreg.php:

```

<?php
include 'includes/testlog.php';
include 'includes/functions.php';
//include 'includes/header.php';
if (isset($_POST['login'])) { $login = $_POST['login']; if ($login == '') {
unset($login);} } //заношим введенный пользователем логин в переменную $login, если
он пустой, то уничтожаем переменную
if (isset($_POST['password'])) { $password=$_POST['password']; if ($password =='') {
unset($password);} }
//заношим введенный пользователем пароль в переменную $password, если он пустой, то
уничтожаем переменную

if (empty($login) or empty($password)) //если пользователь не ввел логин или пароль,
то выдаем ошибку и останавливаем скрипт
{
exit ("Вы ввели не всю информацию, вернитесь назад и заполните все поля!");
//останавливаем выполнение сценариев
}
//если логин и пароль введены,то обрабатываем их, чтобы теги и скрипты не работали,
мало ли что люди могут ввести
$login = clean($login);
$password = clean($password);
$query = mysqli_query($link, "SELECT * FROM `users` WHERE `login` = '$login' AND
`password` = '$password'");
$user_data = mysqli_fetch_assoc($query);
if($login == $user_data['login'] && $password == $user_data['password']) {
    $_SESSION['password']=$user_data['password'];
    $_SESSION['login']=$user_data['login'];
    $hash = md5($login.$password).'fx452';
    mysqli_query($link, "UPDATE `users` SET `userhash` = '$hash' WHERE `login` =
'$login'");
    setcookie('login', $login, time()+60*60*24*7);
    setcookie('userhash', $hash, time()+60*60*24*7);
}

echo "<html><head><meta http-equiv='Refresh' content='0;
URL=index.php'></head></html>";
?>

```

**ДОДАТОК Б**  
**Комп'ютерна презентація**

**Дипломний проект бакалавра на тему**  
**Веб-сервіс денного харчового раціону**

Студента групи КІ-13аД    Коннова Віктора Ілліча

Рисунок Б.1 – Титульний лист

Розробка веб-сервісу моніторингу денного харчового раціону включає в себе наступні задачі:

- Вибір засобів розробки.
- Проектування бази даних.
- Реалізація логіки сайту.
- Розробка зручного інтерфейсу.

Рисунок Б.1– Задачі розробки

## Вибір засобів розробки:

- PHP
- HTML
- CSS
- JavaScript
- SQL (СУБД - MySQL)

Рисунок Б.3 – Вибір засбір розробки



Рисунок Б.4 – Схема бази даних.

## Структура інтерфейсу

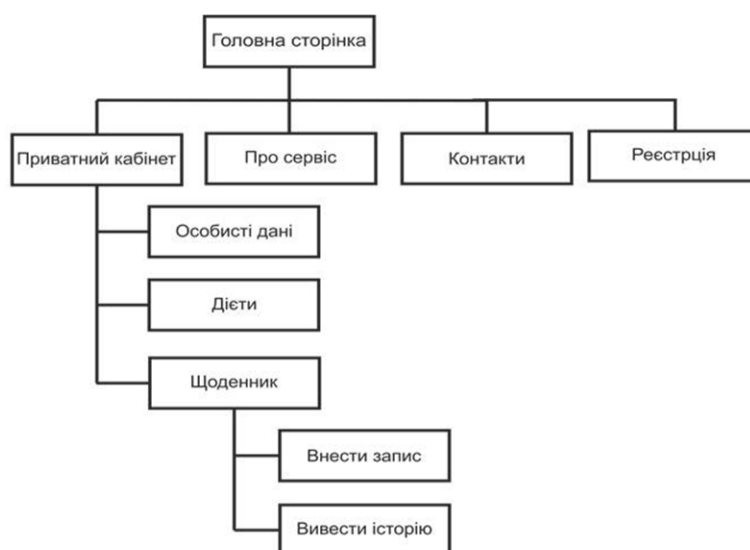


Рисунок Б.5 – Структура інтерфейсу

## Сторінка авторизації

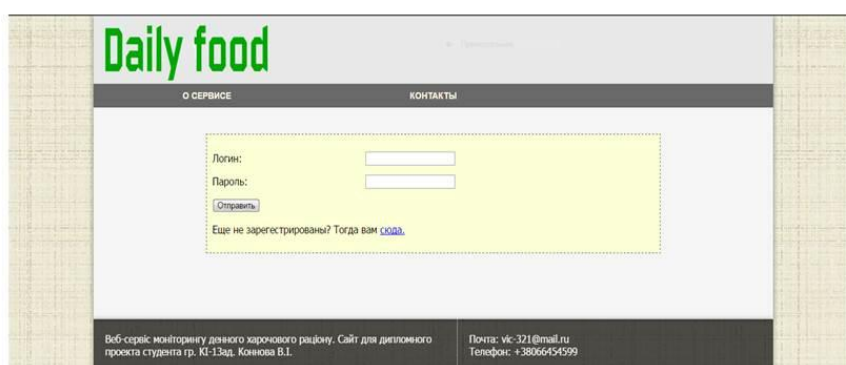


Рисунок Б.6 – Сторінка авторизації



## Сторінка реєстрації

The screenshot shows the registration page for 'Daily food'. The page has a header with the logo 'Daily food' in green. Below the logo are two navigation links: 'О СЕРВИСЕ' and 'КОНТАКТЫ'. The main content area is a yellow box containing a registration form with three input fields labeled 'Логин:', 'e-mail:', and 'Пароль:'. Below the fields are two buttons: 'Отправить' and 'Очистить'. At the bottom of the page, there is a footer with two columns of text. The left column contains: 'Веб-сервис мониторингу дневного харчового раціону. Сайт для дипломного проекта студента гр. КІ-13ад, Кожнова В.І.'. The right column contains: 'Почта: vic-321@mail.ru' and 'Телефон: +38066454599'.

Рисунок Б.7 – Сторінка реєстрації

## Сторінка інформації о користувачі

The screenshot shows the user information page for 'Daily food'. The page has a header with the logo 'Daily food' in green. Below the logo are three navigation links: 'Личный кабинет', 'О СЕРВИСЕ', and 'КОНТАКТЫ'. The main content area is a white box with a green header bar containing three tabs: 'Личные данные', 'Дневник', and 'Диеты'. Below the tabs is a list of user information: 'Имя: Виктор', 'Пол: Мужской', 'Рост: 180 см.', 'Вес: 75 кг.', 'Возраст: 23', and 'Диета: Диета №5a'. Below the information is a link labeled 'Изменить'. At the bottom of the page, there is a footer with two columns of text. The left column contains: 'Веб-сервис мониторингу дневного харчового раціону. Сайт для дипломного проекта студента гр. КІ-13ад, Кожнова В.І.'. The right column contains: 'Почта: vic-321@mail.ru' and 'Телефон: +38066454599'.

Рисунок Б.8 – Сторінка інформації о користувачі

## Сторінка редагування інформації користувача

The screenshot shows the 'Daily food' website interface. At the top, there is a navigation bar with 'Личный кабинет', 'О сервисе', and 'Контакты'. Below this, a green header contains 'Личные данные', 'Дневник', and 'Диеты'. The main content area is a form for editing user information:

- Имя: Витер
- Пол: Мужской
- Рост: 180
- Вес: 73
- Возраст(полных лет): 23
- Текущая диета: Нет спец. диеты

At the bottom of the form is a button labeled 'Отправить'. The footer contains contact information: 'Веб-сервис мониторингу дневного харчового раціону. Сайт для дипломного проекта студента гр. КИ-13ад. Ковалева В.І.' and 'Почта: vc-321@mail.ru', 'Телефон: +38066454599'.

Рисунок Б.9 – Сторінка редагування інформації користувача

## Сторінка ведення харчового раціону

The screenshot shows the 'Daily food' website interface for entering a food diary. At the top, there is a navigation bar with 'Личный кабинет', 'О сервисе', and 'Контакты'. Below this, a green header contains 'Личные данные', 'Дневник', and 'Диеты'. The main content area is a form for entering a food diary entry:

- Список продуктов на: Завтрак
- Дата: 13.06.2017

Below the form, there is a table for adding products:

Продукт:	Белый хлеб	вес:	
Продукт:	Сардинский хлеб	вес:	
Добавить:	Бульон из курицы		

The footer contains contact information: 'Веб-сервис мониторингу дневного харчового раціону. Сайт для дипломного проекта студента гр. КИ-13ад. Ковалева В.І.' and 'Почта: vc-321@mail.ru', 'Телефон: +38066454599'.

Рисунок Б.10 – Сторінка ведення харчового раціону



## Сторінка з інформацією про дієти

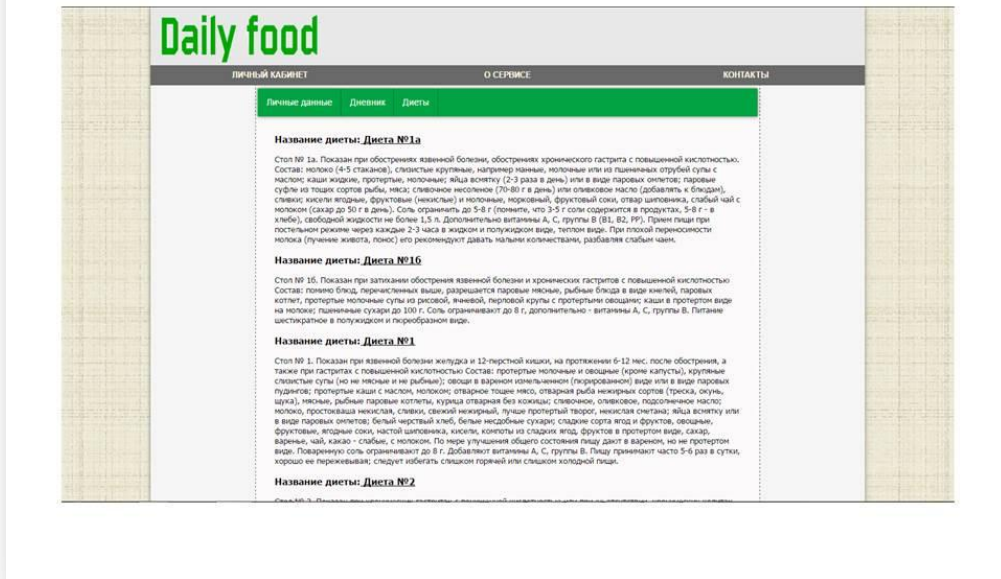


Рисунок Б.12 – Сторінка з інформацією про дієти

## Висновки

- У данному дипломному проекті на базі аналізу основних принципів прийому їжі, і дієт, був розроблений і реалізований веб-сервіс моніторингу денного харчового раціону, з використанням таких засобів як PHP, HTML, CSS і SQL. Був розроблений зручний інтерфейс для доступу до функцій сервісу і його даних.
- Даний веб-сервіс може бути корисним для людей які стежать за своїм здоров'ям, харчовим раціоном а так само зацікавити таких, які раніше про це не замислювалися.

Рисунок Б.13 – Висновки