

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

Інститут (факультет) Автоматизації і комп'ютерних систем  
Кафедра Інформаційних систем

«До захисту в ЕК»

Директор інституту (декан факультету)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис)

Форсюк А. В.  
(прізвище та ініціали)

«   » \_\_\_\_\_ 2021 р.

\_\_\_\_\_  
(підпис)

Чумаченко С.М.  
(прізвище та ініціали)

«   » \_\_\_\_\_ 2021 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: Розроблення інформаційної системи підтримки обслуговування дрібногуртових клієнтів ПрАТ «Білоцерківський консервний завод»

Виконав: здобувач 4 курсу, групи 4

Степаненко Ілля Ігорович

(прізвище, ім'я, по батькові повністю)

\_\_\_\_\_  
(підпис)

Керівник Мошенський Андрій Олександрович

(прізвище, ім'я та по батькові повністю)

\_\_\_\_\_  
(підпис)

Консультанти

\_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент

Смітюх Я. В.

(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

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

Здобувач \_\_\_\_\_

(підпис)

Київ – 2021 р.

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних систем

Освітній ступінь Бакалавр

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

(назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інформаційних систем

Чумаченко С.М.

“ ” 2021 року

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Степаненко Іллі Ігоровича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи підтримки обслуговування дрібногуртових клієнтів ПрАТ «Білоцерківський консервний завод»

керівник роботи Мошенський Андрій Олександрович,

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

затверджені наказом закладу вищої освіти від “29” 04 2021 року № 248-к

2. Строк подання здобувачем роботи 31 травня 2021 р.

3. Вихідні дані до роботи Дані про працівників, їх посадові інструкції, історія кадрових операцій, електронні документи

льної записки (перелік питань, які потрібно розробити) Організаційна та функціональна схема ПрАТ «Білоцерківський консервний завод», детальна робота відділу кадрів

5. Перелік графічного матеріалу

1. Організаційна структура ПрАТ «Білоцерківський консервний завод»

2. Функціональна модель діяльності відділу кадрів

3. Логічна та фізична моделі відділу кадрів

4. Сгенерована база даних

5. Інтерфейс користувача ІС діяльності відділу кадрів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
I	Доцент Мошенський Андрій Олександрович	01.03.21	04.04.21
II	Доцент Мошенський Андрій Олександрович	01.03.21	29.04.21
III	Доцент Мошенський Андрій Олександрович	01.03.21	20.05.21
IV	Доцент Мошенський Андрій Олександрович	01.03.21	27.05.21
Висновки	Доцент Мошенський Андрій Олександрович	01.03.21	29.05.21

7. Дата видачі завдання 01 березня 2021 р.

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Вивчення та проведення системного аналізу ПрАТ «Білоцерківський консервний завод»	13.03.2021	<i>Виконано</i>
2.	На основі проведених досліджень було проведено відображення існуючій організаційної структури та окремо відділу кадрів	01.04.2021	<i>Виконано</i>
3.	Складання функціональної моделі діяльності відділу кадрів с деталізацією в три рівні.	04.04.2021	<i>Виконано</i>
4.	Виявлення недоліків та вузьких місць для автоматизації процесів відділу кадрів	21.04.2021	<i>Виконано</i>
5.	Створення моделі та узгодження її із замовником ІС	29.04.2021	<i>Виконано</i>
6.	Програмна реалізація запланованих функцій з метою поліпшення щоденних робіт відділу кадрів	01.05.2021	<i>Виконано</i>
7.	Дослідження та опис заходів з охорони праці	27.05.2021	<i>Виконано</i>
8.	Оформлення та фіксування кваліфікаційної роботи	29.05.2021	<i>Виконано</i>
9.	Створення презентації на основі пояснювальної записки.	31.05.2021	<i>Виконано</i>

**Здобувач**

\_\_\_\_\_ ( підпис )

**Степаненко І.І.**

\_\_\_\_\_ (прізвище та ініціали)

**Керівник роботи**

\_\_\_\_\_ ( підпис )

**Мошенський А.О.**

\_\_\_\_\_ (прізвище та ініціали)

## ЗМІСТ

<b>ВСТУП</b> .....	8
<b>РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ ПРАТ БІЛОЦЕРКІВСЬКИЙ КОНСЕРВНИЙ ЗАВОД</b> .....	10
<b>1.1. Структура підприємства ПрАТ Білоцерківський консервний завод</b> 10	
<b>1.2. Організаційна структура ПрАТ Білоцерківський консервний завод         роль і взаємодія підрозділів</b> .....	11
<b>1.3. Задачі автоматизації</b> .....	12
<b>1.4. Аналіз нинішнього стану комп'ютеризації ПрАТ Білоцерківський         консервний завод</b> .....	13
<b>1.5. Огляд існуючих рішень для розв'язання виявлених проблем</b> .....	18
<b>1.5. Обґрунтування доцільності проектування й розроблення (назва         проекту)</b> .....	21
<b>1.6. Розрахунок економічного ефекту від впровадження системи</b> .....	22
<b>РОЗДІЛ 2. ОГЛЯД МЕТОДИК СТРУКТУРУВАННЯ ДАНИХ ТА РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ</b> .....	29
<b>2.1. Поняття веб-сайта та їх класифікація</b> .....	29
<b>2.2. Етапи розробки веб-сайта</b> .....	30
<b>2.3. Огляд засобів структурування даних та їх використання в         інформаційній системі</b> .....	33
<b>2.4. Методи табличного верстання інтернет-сайтів</b> .....	38
<b>2.5. Використання методу float для створення Web-додатку</b> .....	39
<b>2.6. Модульна система верстки у веб-дизайні</b> .....	40
<b>РОЗДІЛ 3. ОПИС КОМПЛЕКСНОГО ВИРІШЕННЯ ЗАДАЧІ</b> .....	42
<b>3.1. Проектування БД</b> .....	42
<b>3.2. Інтерфейс користувача</b> .....	43
<b>3.3. Алгоритмізація та реалізація комплексу задач автоматизації</b> .....	45
<b>3.2.1. Розробка макету десктоп-версії web-ресурсу</b> .....	46

<b>1.4. Технічне та системне забезпечення розробки.....</b>	<b>50</b>
<b>1.4.1. Налаштування серверу .....</b>	<b>50</b>
<b>3.4.2. Розробка та верстка сайту.....</b>	<b>52</b>
<b>3.5. Розробка форми замовлення продукту .....</b>	<b>61</b>
<b>РОЗДІЛ 4. ОХОРОНА ПРАЦІ .....</b>	<b>64</b>
<b>ВИСНОВКИ .....</b>	<b>70</b>
<b>БІБЛІОГРАФІЧНИЙ СПИСОК .....</b>	<b>72</b>

## АНОТАЦІЯ

Кваліфікаційна робота «Розроблення інформаційної системи підтримки обслуговування дрібногуртових клієнтів ПрАТ Білоцерківський консервний завод» розроблена Степаненко І.І., складається з 78 сторінок, 5 таблиць, 8 рисунків, 13 сторінок графічного матеріалу, 35 літературних джерел.

У кваліфікаційній роботі наведені результати дослідження діяльності ПрАТ Білоцерківський консервний завод для дітей» вцілому і об'єкта автоматизації – В даному випадку мова йде про підтримку обслуговування дрібногуртових клієнтів, шляхом створення web-додатку. Для реалізації функцій, описаних у функціональній моделі, здійснено вибір програмного забезпечення, описано комп'ютерну мережу, розроблена інструкція користувача, описані заходи з охорони праці для робочого місця працівника ПЕОМ.

Функціональна модель діяльності головного технолога розроблена за допомогою CASE-засобу AllFusion ERwin Process Modeler.

Логічна та фізична моделі бази даних розроблені з використанням CASE-засобу аналізу, проектування AllFusion ERwin Process Modeler. В ході виконання роботи були розглянуті основні поняття проектування макетів та скетчів сайту, розглянуто різні підходи до створення інформаційної системи у вигляді web-ресурсу, зокрема, через конструктори створення сайтів, підходи до впровадження новітніх API-технологій з метою візуалізацій процесу і в загальному – проаналізовано їх позитивний і негативний вплив на проект кожного процесу. Практичним аспектом роботи стало створення відповідного web-ресурсу для досягнення поставленої мети роботи.

***КЛЮЧОВІ СЛОВА: WEB-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА, WEB-РЕСУРС, КОНСТРУКТОР САЙТІВ, СИСТЕМА ПІДТРИМКИ***

## ABSTRACT

Qualification work "Development of information system to support the service of small wholesale customers of PJSC Bila Tserkva Cannery" developed by Stepanenko II, consists of 78 pages, 5 tables, 8 figures, 13 pages of graphics, 35 references.

The qualifying work presents the results of a study of the activities of PJSC Bila Tserkva Cannery for Children "in general and the object of automation - In this case, we are talking about supporting the service of small customers by creating a web-application. To implement the functions described in the functional model, the software is selected, the computer network is described, the user manual is developed, the labor protection measures for the PC employee's workplace are described.

The functional model of the chief technologist's activity was developed with the help of the CASE-tool AllFusion ERwin Process Modeler.

Logical and physical database models are developed using CASE-analysis tool, designed by AllFusion ERwin Process Modeler. During the work the basic concepts of designing layouts and sketches of the site were considered, different approaches to creating an information system in the form of a web-resource, in particular, through designers of creating sites, approaches to implementing the latest API-technologies to visualize the process and in general - analyzed their positive and negative impact on the project of each process. A practical aspect of the work was the creation of an appropriate web-resource to achieve the goal.

**KEY WORDS:** *WEB-ORIENTED INFORMATION SYSTEM, WEB-RESOURCE, WEBSITE DESIGNER, SUPPORT SYSTEM*

## ВСТУП

Актуальність дослідження полягає в тому, що зростання можливостей бізнесу в мережі Інтернет та розвиток глобальної інформаційної інфраструктури створюють принципово нову ситуацію в економіці. Тому надзвичайно важливо створювати якісні, прості в застосуванні та головне унікальні інформаційні системи, котрі можуть допомагати малому та великому бізнесу у впровадженні web-технологій до їх підприємств.

У даний час Інтернет - це один з найрозвиненіших і найефективніших платформ для ведення бізнесу. При ефективному використанні усіх можливостей рівень доходу як і рівень комфорту клієнтів значно зростає, що змушує відмітити, що компаніям необхідне власне представництво в мережі Інтернет.

Першочерговим завданням для підприємства сьогодні є вихід на інтернет-ринок, що дозволяє залучити нових клієнтів та не втратити вже існуючих постійних клієнтів, охопити більший відсоток ринку, за рахунок моніторингу популярності товарів, цін, рекламних компаній конкурентів та фінансових уподобань інвесторів. Інтенсивний розвиток можливостей глобальної мережі і використання її як інструменту ведення бізнесу висунуло нові вимоги як до створення web-ресурсів, так і до просування web-сайтів у глобальному пошуку.

**Мета роботи:** обґрунтування та розробка web-сайту через конструктори створення сайтів, як перспективного та ефективного способу впровадження інформаційної системи на підприємстві.

**Об'єкт дослідження:** процес проектування web-сайту для інформаційної підтримки процесу обслуговування клієнтів.

**Предмет дослідження:** засоби створення web-сайту для процесу обслуговування клієнтів.

**Методи дослідження:** аналіз методів створення web-ресурсів, використання принципів інформаційного моделювання, застосування системного підходу та структурно-функціонального методу, аналіз та опис різних методів створення баз даних (далі БД).

**Основні завдання дослідження:**

1. Аналіз роботи та використання API-технологій.
2. Огляд інструментальних засобів для використання технологій взаємодії модулів.
3. Розробка макетів створюваного сайту.
4. Розробка web-сайту з використанням конструктора для створення сайтів.

## **РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ ПРАТ БІЛОЦЕРКІВСЬКИЙ КОНСЕРВНИЙ ЗАВОД**

### **1.1. Структура підприємства ПрАТ Білоцерківський консервний завод**

ПрАТ Білоцерківський консервний завод – це одне з найбільших підприємств по виробництву консерв у Київському регіоні. Нині виробнича потужність заводу складає 25 млн банок на рік і спеціалізується на промисловій переробці овочів, фруктів і м'яса. На сьогоднішній день асортимент продукції підприємства становить 87 видів консерв, які з успіхом продаються не тільки в Україні, але й в Німеччині, США, Ізраїлі, Канаді, Греції, Прибалтиці, Вірменії та Азербайджані.

Білоцерківський міськрайонний центр зайнятості протягом багатьох років співпрацює з ПАТ «Білоцерківським консервним заводом», який є одним із провідних підприємств переробки сільськогосподарської сировини в Київському регіоні. На цьому підприємстві постійно затребувані безробітні для сезонних та оплачуваних громадських робіт.

В 2009 році з ПАТ «Білоцерківський консервним заводом» був заключений договір на сезонні роботи на 100 чоловік та 2 договори на оплачувані громадські роботи на 110 чоловік, де працювали безробітні виконували підсобні роботи по сортуванню та закладенню овочів у банки. Представники підприємства ПАТ «Білоцерківського консервного заводу» беруть активну участь в заходах, які проводить служба зайнятості, а саме: семінари для представників кадрових служб, засіданнях круглих столів, конференціях для роботодавців, ярмарках вакансій.

Залежно від наявності основних і допоміжних процесів розрізняють підприємства з комплексною та спеціалізованою структурами виробництва.

## **1.2. Організаційна структура ПрАТ Білоцерківський консервний завод роль і взаємодія підрозділів**

Підприємства з комплексною виробничою структурою мають усю сукупність основних та допоміжних цехів, а зі спеціалізованою структурою – лише частину. Підприємства зі спеціалізованою структурою поділяють на:

- підприємства механоскладального типу, які отримують заготовки від інших підприємств;
- підприємства складального типу, які випускають продукцію з деталей, вузлів та агрегатів, що виготовляються на інших підприємствах;
- підприємства заготовочного типу, що спеціалізуються на виробництві заготовок;
- підприємства, спеціалізовані на виробництві окремих деталей.

На виробничу структуру підприємства впливає ряд факторів. Головні з них:

- галузева належність, номенклатура продукції, її техніко- економічні особливості, використані ресурси;
- тип виробництва, рівень спеціалізації і кооперування;
- структура засобів праці, технології;
- ступінь складності конструкції і технологічність продукції;
- організація обслуговування обладнання, його ремонт і модернізація;

- мобільність виробничого процесу, тобто його спроможність оперативно переходити на випуск нової продукції;

- характер відтворювального процесу в підрозділах підприємства.

Ці фактори визначають специфіку виробничої структури підприємства в різних галузях. Ринкові умови господарювання потребують передусім нової виробничої структури, відтворення фондів підприємства ПрАТ Білоцерківський консервний завод. На наступному рисунку представлена у вигляді схеми організаційна структура нашого підприємства.

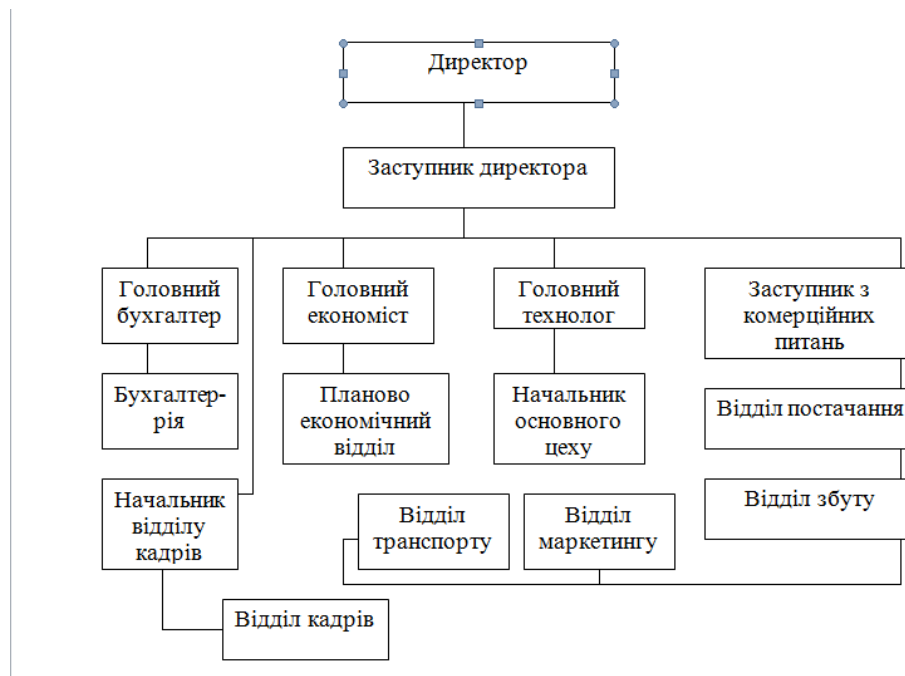


Рис. 1.1. Організаційна структура ПрАТ Білоцерківський консервний завод

### 1.3. Задачі автоматизації

На сьогоднішній день практично кожна організація має власний веб-сайт. В умовах використання сучасних інформаційних технологій – це необхідний чинник існування, що дозволяє розширити поле рекламної діяльності і привернути тим самим додаткових клієнтів.

Створення і розробка сайтів включає:

1. Затвердження первинного технічного завдання на розробку сайта.
2. Визначення структурної схеми сайту - розташування розділів, контенту і навігації.
3. Веб-дизайн - створення графічних елементів макету сайту, стилів і елементів навігації.
4. Розробка програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті.
5. Тестування і розміщення сайту в мережі Інтернет.

#### **1.4. Аналіз нинішнього стану комп'ютеризації ПрАТ Білоцерківський консервний завод**

Обмін інформацією в Інтернет здійснюється за допомогою протоколів прикладного рівня, що реалізують той або інший прикладний сервіс (пересилку файлів, гіпертекстової інформації, пошта і так далі). Одним з найбільш молодих і популярних сервісів Інтернет, розвиток якого і привело до сплеску популярності самої Інтернет, стала World Wide Web (WWW), заснована на протоколі HTTP (Hyper Text Transfer Protocol - протокол передачі гіпертекстової інформації). Гіпертекстові документи, представлені в WWW, мають одну принципову відмінність від традиційних гіпертекстових документів - зв'язки, в них що використовуються, не обмежені одним документом, і більш того, не обмежені одним комп'ютером. Для підготовки гіпертекстових документів використовується мова HTML (Hyper Text Markup Language – мова розмітки гіпертекстових документів), що надає широкі можливості по форматуванню і структурній розмітці документів, організації зв'язків між різними документами, засоби включення графічної і мультимедійної інформації. HTML-документи є видимими за допомогою спеціальної програми - браузера. Найбільшого поширення в даний час набули браузери MozillaFirefox і Internet Explorer

компанії Microsoft (MSIE). Реалізації MozillaFirefoxдоступні практично для всіх сучасних програмних і апаратних платформ, реалізації MSIE доступні для всіх Windows платформ, Macintosh і деяких комерційних Unix-систем.

HTML-документ складається з тексту, що є змістом документа, і *тегів*, що визначають його структуру і зовнішній вигляд при відображенні браузером. Простий html-документ виглядає таким чином:

```
<html >
<head >
<title>Название</title>
</head>
<body >
<p>Тело документа
</body>
</html>
```

Як видно з прикладу, тег є ключовим словом, поміщеним в кутові дужки. Розрізняють одинарні теги, як, наприклад, `<p >`, і парні, як `<body> </body>`, в останньому випадку дія тега розповсюджується тільки на текст між його відкриваючою і закриваючою дужкою. Теги також можуть мати параметри - наприклад, при описі сторінки можна задати колір фону, колір шрифту і т.д.: `<Body bgcolor="white" text="black">`.

Текст всього документа полягає в теги `<html >`, сам документ розбивається на дві частини - заголовок і тіло. Заголовок описується тегами `<head>`, в яких можуть бути включені назва документа (за допомогою тегов `<title>`) і інші параметри, що використовуються браузером при відображенні документа. Тіло документа поміщене в теги `<body>` і містить власне інформацію, яку бачить користувач. За відсутності тегів форматування весь текст виводиться у вікно браузера суцільним потоком, переклади рядків, пропуски і табуляції

розглядаються як пробільні символи, декілька пробільних символів, що йдуть підряд, замінюються на один. Для форматування використовуються наступні основні теги:

`<p>` - початок нового абзацу, може мати параметр, що визначає вирівнювання:

`<p align=right>`;

`<br>` - переклад рядка в межах поточного абзацу;

`<u></u>` - виділення тексту підкресленням

Посилання на інший документ встановлюється за допомогою тега `<a href="URL ">...</a>`, де URL - повна або відносна адреса документа. При цьому текст, ув'язнений в тег `<a>`, зазвичай виділяється підкресленням і кольором, і після натискання мишею по цьому посиланню браузер відкриває документ, адреса якого вказана в параметрі href. Графічні зображення вставляються в документ за допомогою тега ``.

**Динамічний HTML** (Dynamic HTML, DHTML) не є якоюсь особливою мовою розмітки сторінок. Це всього лише термін, вживаний для позначень HTML-сторінок з динамічно змінним вмістом.

Реалізація DHTML “стоїть” на трьох “китах”: безпосередньо HTML, каскадні таблиці стилів і мови сценаріїв. Ці три компоненти DHTML зв'язано між собою об'єктною моделлю документа (DOM, Document Object Model), що є по суті інтерфейсом прикладного програмування (API). DOM зв'язує воедино три перераховані компоненти, додаючи простому документу HTML нову якість – можливість динамічної зміни свого вмісту без перевантаження сторінки.

Об'єктна модель документа робить всі елементи сторінки програмованими об'єктами. З її допомогою через мови сценаріїв можна дістати доступ і управляти всім, що є в документі. Кожен елемент HTML доступний як індивідуальний об'єкт, а це означає, що можна змінювати значення будь-якого параметра будь-якого тега HTML-сторінки, і, як наслідок, документ дійсно стає динамічним.

Будь-яка дія користувача (натискання кнопкою миші, переміщення миші у вікні браузера або натиснення клавіші клавіатури) об'єктною моделлю документа трактується як подія, яка може бути перехоплена і оброблена процедурою сценарію.

На сьогоднішній день вже всім фахівцям в області веб-технологій стало очевидне, що існуючих стандартів передачі даних по Internet недостатньо. Формат HTML, ставши свого часу проривом в області відображення вмісту вузлів Internet, вже не задовольняє всім необхідним на даний момент вимогам. Він дозволяє описати те, яким чином повинні бути відображені дані на екрані кінцевого користувача, але не надає ніяких засобів для ефективного опису передаваних даних і управління ними.

Крім того, каменем спотикання для багатьох компаній, що займаються розробкою програмного забезпечення, є необхідність сумісного використання різних компонентів, забезпечення їх взаємодії, можливості обміну даними між ними.

До недавнього часу не існувало стандарту, що надає засоби для інтелектуального пошуку інформації, обміну даними, адаптивної обробки отримуваних даних.

Вирішенням всіх описаних вище проблем став затверджений в 1998 році міжнародною організацією W3C мова XML. XML (eXtensible Markup Language) - це розширювана мова розмітки, призначена для опису в текстовій формі структурованих даних. Цей текстовий (text-based) формат, багато в чому схожий з HTML, розроблений спеціально для зберігання і передачі даних.

XML дозволяє описувати і передавати такі структуровані дані, як:

1. Окремі документи.
2. Метадані, що описують вміст якого-небудь вузла Internet.

3. Об'єкти, що містять дані і методи роботи з ними (наприклад, елементи управління ACTIVEX або об'єкти Java).

4. Окремі записи (наприклад, результати виконання запитів до баз даних)

5. Всілякі веб-посилання на інформаційні і людські ресурси Internet (адреси електронної пошти, гіпертекстові посилання і ін.)

Дані, описані на мові XML, називаються XML-документами. Мова XML легко читана і достатньо проста для розуміння. Якщо Ви були знайомі з HTML, то навчитися складати XML-документи не складе для Вас ніяких труднощів.

Початковий текст XML-документа складається з набору XML-елементів, кожен з яких містить початковий і кінцевий теги. Кожна пара тегів представляє частину даних. Тобто, як і HTML, мова XML для опису даних використовує теги. Але, на відміну від HTML, XML дозволяє використовувати необмежений набір пар тегів, кожна з яких представляє не то, як увязнені в неї дані повинні виглядати, а то, що вони означають.

Будь-який елемент XML-документа може мати атрибути, характеристики, що уточнюють його. Атрибут - це пара ім'я = "значення", яка задається при визначенні елемента в початковому тегу.

Принцип розширюваності мови XML полягає в можливості використання необмеженої кількості пар тегів, визначуваних творцем XML-документа.

Принцип незалежності визначення внутрішньої структури документа від способів представлення цієї інформації полягає у відділенні даних від процесу їх обробки і відображення. Таким чином, отримані дані можна використовувати відповідно до потреб клієнта, тобто вибирати потрібне оформлення, застосовувати необхідні методи обробки.

Управляти відображенням елементів у вікні програми-клієнта (наприклад, у вікні браузера) можна за допомогою спеціальних інструкцій - стильових таблиць XSL (eXstensible Stylesheet Language). Ці таблиці XSL дозволяють визначати

оформлення елемента залежно від його місцезнаходження усередині документа, тобто до двох елементів з однаковою назвою можуть застосовуватися різні правила форматування. Крім того, мовою, лежачою в основі XSL, є XML, а це означає, що таблиці XSL більш універсальні, а для контролю коректності складання таких стильових таблиць можна використовувати DTD-описання або схеми даних, розглянуті нижче.

### **1.5. Огляд існуючих рішень для розв'язання виявлених проблем.**

Формат XML, в порівнянні з HTML, має невеликий набір простих правил розбору, який дозволяє розбирати XML-документи, не вдаючись до яких-небудь зовнішніх описів використовуваних XML-елементів. У загальному випадку XML-документи повинні задовольняти наступним вимогам:

1. Кожен відкриваючий тег, що визначає деяку частину даних в документі, обов'язково повинен супроводжуватися таким, що закриває, тобто, на відміну від HTML, не можна опускати закриваючі теги.

2. Вкладеність тегів в XML строго контролюється, тому необхідно стежити за порядком проходження відкриваючих і закриваючих тегів.

3. У XML враховується регістр символів.

4. Вся інформація, розташована між початковим і кінцевим тегами, розглядається в XML як дані, і тому враховуються всі символи форматування (тобто пропуски, переклади рядків, табуляції не ігноруються, як в HTML).

У XML існує набір зарезервованих символів, які повинні бути задані в XML-документі тільки спеціальним чином. Багато фахівців розглядають XML як нову технологію інтеграції програмних компонентів. Основними перевагами використання XML є:

1. Інтеграція даних з різних джерел. XML можна використовувати для об'єднання різнорідних структурованих даних на середньому рівні тривірневих веб-систем, баз даних.

2. Локальна обробка даних. Отримані дані у форматі XML можна розбирати, обробляти і відображати безпосередньо на клієнтові без додаткових звернень до сервера.

3. Перегляд і маніпулювання даними в різних розрізах. Отримані дані можуть оброблятися і бути видимими клієнтом різними способами залежно від потреб кінцевого користувача.

4. Можливість часткового оновлення даних. За допомогою XML можна оновлювати тільки ту частину структурованих даних, яка була змінена, а не всю структуру цілком.

Всі ці переваги роблять XML незамінним інструментом для розробки гнучких засобів пошуку інформації в базах даних, могутніх тривірневих веб-додатків, а також додатків, що підтримують транзакції. Іншими словами, за допомогою XML можна формувати запити до баз даних різних структур, що дозволяє здійснювати пошук інформації в численних несумісних один з одним базах даних. Використання XML на середньому рівні тривірневих веб-додатків дозволяє здійснювати ефективний обмін даними між клієнтами і серверами систем електронної комерції.

Крім того, мова XML може використовуватися як засіб для опису граматики інших мов і контролю правильності складання документів.

Інструменти обробки даних, отриманих у форматі XML, можуть бути розроблені в середовищі Visual Basic, Java або C++.

Історія **RНР** починається осінню 1994 року. Коли Расмус Лердорф (Rasmus Lerdorf) почав працювати над тим, що згодом стало RНР, єдиною метою, яка була у нього в думках, з'ясувати, хто читає його резюме. У той час, будучи незалежним підрядчиком, Лердорф розсилав потенційним працедавцям своє міні-резюме з URL посиланням на його повну версію. Щоб стежити за відвідувачами, він створив CGI скрипт на Perl-і, який вставлявся як спеціальний тег в HTML код

його сторінки, і збирав інформацію про відвідувачів. Щоб справити враження на потенційних працедавців, він дозволив будь-якому відвідувачеві сторінки проглядати збирану статистику відвідин. [1]

Він назвав цей код для збору статистики «PHP-Tools for Personal Home Page», оскільки сам використовував його на своїй персональній домашній сторінці (personal home page). Декілька чоловік поцікавилися тим, як вони могли б отримати цей інструмент, і Лерддорф ухвалив рішення надати його іншим особам. «Це диво програмного забезпечення. Ви можете дати це і проте залишити це собі», - дотепно відмітив Лерддорф. У той час руху Open Source не існувало. Тоді воно назвалося freeware. Ближче до кінця 1995 року Лерддорф відкрив для людей перший список розсилки по PHP, щоб можна було обмінюватися ідеями, виправленнями помилок і кодом.

Найпоширеніший веб-сервер в світі - це **Apache**. За даними компанії Netcraft, загальне число веб-узлів, що працюють під його управлінням, до кінця 1998 р. досягло 2 млн. (55% загального числа вузлів) і постійно росте. Для порівняння: на долю серверів Microsoft доводиться 25%, Netscape - 7%. Будучи безкоштовною відкритою програмою, призначеною для безкоштовних же Unix-систем (FREEBSD, Linux і ін.), Apache по функціональних можливостях і надійності не поступається комерційним серверам, а широкі можливості конфігурації дозволяють побудувати його для роботи практично з будь-якою конкретною системою. Існують локалізації сервера для різних мов, у тому числі і для російської.

Історично склалося так, що російські тексти в Internet можуть бути представлені в різних кодуваннях, з яких найбільш поширені koі8-г (або просто koі8) і Windows-1251: з першою працюють більшість серверів і робочих станцій під управлінням Unix, друга є стандартною для всіх версій Windows. Оскільки кодування Windows-1251, природно, застосовується на переважній більшості

клієнтських машин, частка тих, хто подорожує по російській частині WWW, використовуючи koI8, не перевищує зараз 5%. Проте в цьому кодуванні зберігаються документи на багатьох Unix-серверах, в ній найчастіше передаються поштові повідомлення і практично завжди - листи в телеконференції, з нею ж працюють багато російськомовних каналів IRC (до речі, аббревіатура KOI розшифровується як "код обміну інформацією"). Щоб вирішити проблеми, що виникають при неспівпаданні кодувань тексту на сервері і клієнтській машині, і був створений російський модуль APACHE-RUS для веб-сервера Apache.

Отже, на підставі поставленого технічного завдання була вибрана система управління вмістом CMS Opencart, оскільки саме ця система відповідає всім вимогам технічного завдання.

Безперечно, є багато аналогів які можна використовувати замість CMS Opencart, але можливість та вартість допрацювань, орієнтовність на електронну комерцію, можливість інтеграцій – роблять його фаворитом в моєму виборі.

Для розробки дизайну була вибрана програма Sublime Text, оскільки дана програма дозволяє візуально споглядати за створенням дизайну для сайту після відкриття документу як формат html разом з форматом css. На головній сторінці було розміщено логотип веб-сайту, основне меню сайту (для навігації по його структурі), форму аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових клієнтів).

### **1.5. Обґрунтування доцільності проектування й розроблення (назва проекту).**

Враховуючи поставлені вище завдання, автор вирішив, що найбільш оптимальним засобом для розробки такого роду електронної комерції – HTML+система управління вмістом Opencart.

Opencart — написана на мові PHP і така, що використовує як сховище змісту базу даних MySQL. Opencart є, захищена ліцензією GNU. Однією з головних особливостей Opencart є відносна простота управління при практично безмежних можливостях і гнучкості при виготовленні сайтів.

CMS Opencart включає різні інструменти для виготовлення веб-сайту. Важливою особливістю системи є мінімальний набір інструментів при початковій установці, який збагачується в міру необхідності. Це знижує те, що захирашується адміністративна панель непотрібними елементами, а також знижує навантаження на сервер і економить місце на хостингу.

### 1.6. Розрахунок економічного ефекту від впровадження системи

Початкові дані

1. Ступінь новизни розроблюваних задач – «В» – використання типових проектних рішень за умови їх змін.

2. Група складності алгоритму – 2.

3. Вид інформації, яка використовується:

Таблиця 1.1 – Вхідна інформація

Вид інформації	Позначення	К-сть наборів даних
Змінна інформація	ПИ	m=4
Нормативно-довідкова інформація	НСИ	n=3
Банк (база) даних	БД	p=1
Обробка в режимі реального часу	РВ	так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	ні

4. Визначаємо базові значення витрат часу на проведення передпроектного дослідження  $T_{Б1}$  і розробки системи  $T_{Б2}$ . з табл.1.3 знаходимо:

Таблиця 1.3 – Базові витрати часу на проведення передпроектного дослідження

	І стадія		ІІ стадія	
	В	Г	В	Г
1.Управління матеріально-технічним постачанням, управління збутом продукції, управління комплектацією, управління експортними та імпорнтними поставками.	95	65	230,1	230,1
2.Бухоблік, управління фінансовою діяльністю.	121	92	231,6	231,6
3.Управління організацією праці і зарплатою, управління кадрами, норми і нормативи, управління охороною праці.	97	65	249,6	249,6
4.Управління якістю продукції, управління технологічними процесами, управління стандартизацією, управління технічною підготовкою виробництва.	98	66	303,1	303,1
5.Управління транспортними перевезеннями. Управління технічним обслуговуванням виробництва, управління допоміжними службами і енергопостачанням.	113	71	272,1	272,1
6.Управління НДР та ДКР.	113	61	231,65	231,65
7.Управління науково-технічною інформацією.	91	61	195,9	195,9
8.Удосконалення документообігу і контроль виконання документів.	91	61	227,2	227,2

$T_{B1}=95\text{год.}; T_{B2.}=230,1\text{год.}$

Для І стадії робіт витрати часу складають:

$$T_1 = T_{B1}$$

Далі одержане базове значення трудомісткості робіт на ІІ стадії  $T_{B2.}$  коригується за допомогою поправочних коефіцієнтів таким чином:

$$T_2 = T_{B2} * k_{II} * k_o * k_c,$$

де  $k_{\Pi}$  – коефіцієнт, що враховує вид використаної інформації і визначається за формулою (1) з використанням даних табл. 1.4

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} = 0.7886$$

Таблиця 1.2 – Значення коефіцієнтів в залежності від виду використаної інформації

Значення коефіцієнтів $k_1$ , $k_2$ , $k_3$ в залежності від виду використаної інформації	Група складності алгоритму	Степень новизни	
		В	Г
$k_1$ (ПИ)	1	1,2	0,65
	2	1,1	0,58
	3	1,0	0,48
$k_2$ (НСИ)	1	0,65	0,4
	2	0,58	0,34
	3	0,48	0,29
$k_3$ (БД)	1	0,54	0,32
	2	0,48	0,29
	3	0,40	0,24

$k_0$  визначається з табл.3  $k_0=1,26$ .

Таблиця 1.3 – Значення коефіцієнтів в залежності від виду обробки даних

Вид обробки даних	Ступінь новизни	
	В	Г
РВ	1,26	1,1
ТОУ	1,36	1,15

$k_c$  – коефіцієнт, що враховує складність контролю вхідної та вихідної інформації і знаходиться з табл.4.  $k_c=1$

Таблиця 1.4 – Значення коефіцієнтів в залежності від складності контролю інформації

Складність контролю вхідної інформації	Складність контролю вихідної інформації	
	21	22
11	1,16	1,07
12	1,08	1,00

Отже,  $T_2 = 230,1 * 0,7886 * 1,26 * 1 = 228.63$  (люд/дн)

Таким чином, загальні витрати людської праці на проектування системи складають

$$T_{\Sigma} = T_1 + T_2 = 95 + 228.63 = 323.63 \text{ (люд/дн)}$$

Визначення Ч - чисельності виконавців.

$$Ч = T_{\Sigma} / \Phi,$$

де  $\Phi$  – кількість робочих годин на виконання проекту. Для дипломного проекту ця величина складає 530 год.

$$Ч = 323.63 / 530 = 0.6106 \approx 1$$

Оплата праці виконавців підраховується за формулою:

$$V_1' = 1 * 3 * 20000 \text{ грн.} = 60000 \text{ грн.}$$

## 2. Витрати, пов'язані з розробкою програми на ПК

### 2.1 Розрахунок річного фонду часу роботи ПК.

Дійсний річний фонд часу ПК у годинах дорівнює числу робочих годин у році для оператора, за винятком часу на технічне обслуговування і ремонт ПК ( в середньому 5 год/міс + 6 діб/рік).

$$T_{\text{ПК}} = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год.}$$

Оскільки під час виконання дипломного проекту (роботи) студент в середньому витрачає 400 год. машинного часу, то величина фонду часу ПК дорівнює

$$T_{\text{ПК}'} = T_{\text{ПК}} * 400 / 2000 = 1892 * 400 / 2000 = 378,4 \text{ год.}$$

2.2 Поточні витрати на експлуатацію визначаються по формулі:

$$V_1'' = Z_{\text{зп}} + Z_{\text{ам}} + Z_{\text{ел}} + Z_{\text{р}} + Z_{\text{мат}},$$

де  $Z_{\text{зп}}$  – заробітна плата обслуговуючого персоналу (якщо роботи виконуються не на власному ПК). Ця величина дорівнює 16800 грн.;

$Z_{\text{ам}}$  – амортизаційні відрахування, що обчислюються за формулою:

$$Z_{\text{ам}} = C_{\text{ПК}} / N_{\text{а}},$$

де  $C_{\text{ПК}}$  – балансова вартість ПК,  $C_{\text{ПК}} = 10000$  грн.

$N_{\text{а}}$  – норма амортизаційних відрахувань, яка для ПК дорівнює 5.

$$Z_{\text{ам}} = 2000 \text{ грн.}$$

Балансова вартість ПК вираховується за формулою:

$$C_{\text{ПК}} = C_{\text{р}} * (1 + K_{\text{ун}}),$$

$C_{\text{р}}$  – ринкова вартість ПК, орієнтовно 10000 грн.,

$K_{\text{ун}}$  – коефіцієнт, що враховує витрати на установку і налагодження ПК і дорівнює 0,12.

$$C_{\text{ПК}} = 11200 \text{ грн.}$$

Витрати на електроенергію, споживану ПК, визначаються за формулою:

$$Z_{\text{ел}} = P_{\text{ПК}} * T_{\text{ПК}} * C_{\text{ел}} * A,$$

де  $P_{\text{ПК}}$  – потужність ПК, орієнтовно дорівнює 0,4 квт,

$T_{\text{ПК}}$  – фонд корисного часу роботи ПК, дорівнює 400 год.,

$C_{\text{ел}}$  – вартість 1 квт електроенергії, дорівнює 1,9218 грн/кВт,

$A$  – коефіцієнт інтенсивного використання ПК, дорівнює 0,9.

$$Z_{\text{ел}} = 277 \text{ грн.}$$

$Z_p$  – витрати на поточний ремонт і технічне обслуговування ПК дорівнюють 6% від  $C_{ПК}$ .

$$Z_p = 672 \text{ грн.}$$

$Z_{\text{мат}}$  – непрямі витрати, пов'язані з експлуатацією ПК, дорівнюють 5% від  $C_{ПК}$

$$Z_{\text{мат}} = 560 \text{ грн.}$$

Отже, загальні витрати на розробку програмного забезпечення комп'ютерної системи дорівнюють

$$V_1 = V'_1 + V_1'' = 60000 + 18314 = 78314 \text{ грн.}$$

### *3. Витрати на придбання і установку ПК*

Витрати на придбання і установку ПК ( $V_2$ ) визначаються за формулою (3).

$$V_2 = 5600 \text{ грн.}$$

### *4. Витрати на підготовку приміщення*

Ці витрати залежать від стану приміщення, де буде встановлюватися ПК. Орієнтовно можна взяти  $V_3 = 10000$  грн, якщо спеціального приміщення немає. Якщо таке приміщення вже є, то  $V_3 = 0$ .

### *5. Витрати на навчання персоналу*

В середньому навчання персоналу триватиме 1 місяць, тому можна дорівняти  $V_4 = 4000$  грн.

### *6. Загальна вартість розробки і впровадження системи $V_{\Sigma}$ .*

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 = 78314 + 11200 + 0 + 4000 = 93514 \text{ грн.}$$

Оскільки норма амортизаційних витрат для комп'ютерних систем  $N_a = 5$ , то для обрахування річного економічного ефекту слід брати до розгляду величину  $V_p = V_{\Sigma} / N_a = 18702,8$  грн

Річний прибуток  $P_p$  від впровадження системи буде за рахунок збільшення контролю за сировиною внаслідок впровадження рецептурного довідника і формуванню коректних змінних виробничих завдань на основі плану

виробництва. І складе не менше 75000 грн на рік (10% більше ніж до впровадження системи) тому економічний ефект від впровадження системи складатиме :

$$K_{\text{еф}} = \Pi_{\text{р.}} / V_{\text{р.}} = 75000 / 18702,8 = 4.01$$

Термін окупності розробки дорівнює  $T_{\text{ок}} = 1 / K_{\text{еф}} = 0,25 \sim 3$  місяці.

## РОЗДІЛ 2. ОГЛЯД МЕТОДИК СТРУКТУРУВАННЯ ДАНИХ ТА РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1. Поняття веб-сайта та їх класифікація

Інформація, доступна користувачам Internet, розташовується на комп'ютерах (веб-серверах), на яких встановлено спеціальне програмне забезпечення. Значна частина цієї інформації організована у вигляді веб-сайтів. Кожен з них має своє ім'я (адреса) в Internet.

Веб-сайт – це інформація, представлена в певному вигляді, яка розташовується на веб-сервері і має своє ім'я (адреса). Для перегляду веб-сайтів на комп'ютері користувача використовуються спеціальні програми, які називаються браузерами. Залежно від того, яке ім'я (адреса) сайту ми задамо в рядку "Адреса", браузер завантажуватиме в своє вікно відповідну інформацію.

Веб-сайт складається із зв'язаних між собою веб-сторінок. Веб-сторінка є текстовим файлом з розширенням \*.htm, який містить текстову інформацію і спеціальні команди – HTML-коди, що визначають в якому вигляді ця інформація відобразатиметься у вікні браузера. Вся графічна, аудіо- і відео-інформація безпосередньо в Веб-сторінку не входить і є окремими файлами з розширеннями \*.gif, \*.jpg (графіка), \*.mid, \*.mp3 (звук), \*.avi (відео). У HTML-коді сторінки містяться тільки вказівки на такі файли.

Кожна сторінка веб-сайта також має свій Internet адрес, який складається з адреси сайту і імені файлу, відповідного даній сторінці. Таким чином, веб-сайт – це інформаційний ресурс, що складається із зв'язаних між собою гіпертекстових документів (веб-сторінок), розміщений на веб-сервері і такий, що має індивідуальну адресу. Подивитися веб-сайт може будь-яка людина, що має комп'ютер, підключений до Internet.

В даний час у всесвітній павутині розміщено декілька мільйонів веб-сайтів і їх число постійно росте. Це особисті сторінки, що містять інформацію про автора,

його інтереси. Їх створюють для того, щоб знайти друзів по інтересах, розширити свій кругозір, свій світ.

*Інформаційні сайти.* До них відносяться сайти учбових закладів, співтовариств по інтересах, фірм і інше.

*Сайти-портали.* До них відносяться крупні веб-сайти, які пропонують велику кількість сервісів, починаючи від каталога зображень до форумів і поштових скриньок.

*Інтернет-магазини.* Такі сайти в наш час набирають дуже велику популярність з боку, як користувачів, так і з боку підприємців. Це веб-сайти, в яких кожна людина може купити будь-який товар не відходячи від свого місця біля комп'ютера.

*Соціальні сайти.* Це веб-сайти на яких кожна людина має змогу спілкуватися зі старими друзями та знаходити нових.

## **2.2. Етапи розробки веб-сайта**

Проектування і розробка сайтів включає:

- Затвердження первинного технічного завдання розробки сайту.
- Визначення структурної схеми сайту - розташування розділів, контенту і навігації.
- Веб-дизайн - створення графічних елементів макету сайту, стилів і елементів навігації.
- Розробка програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті.
- Тестування і розміщення сайту в мережі Інтернет.

Перед автором було поставлено перше завдання: розробити структурну схему проекту веб – сайту для використання в середовищі Internet. На думку керівника диплому, веб-сайт, що розробляється, повинен володіти наступними особливостями:

- гнучкістю, зручною для адміністраторів системою управління структурою;
- веб-сайт повинен підтримувати використання графічних вставок, анімації, які повинні підсилювати емоційно-ціннісний компонент змісту, формувати мотивацію;
- для користувачів повинна бути також реалізована можливість роздрукувати будь-яку сторінку веб-сайта.

Проте головним завданням проектування було створення системи управління вмістом, яка б дозволяла вносити зміни веб – сайта для людей, які не мають навичок в розробці веб-сайтів.

Титульна сторінка (головна) будь-якого сайту повинна максимально інформативно і в стислому об'ємі відображати необхідну користувачеві інформацію про сайт. На головній сторінці необхідно помістити логотип веб-сайту, основне меню сайту (для навігації по його структурі), форму аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових клієнтів).

Оскільки сайт написаний за допомогою системи управління вмістом Opencart, база даних якої представлена у вигляді MYSQL, база даних сайта підтримуватиме ту ж структуру (Рис.1), вносячи до неї свої зміни.

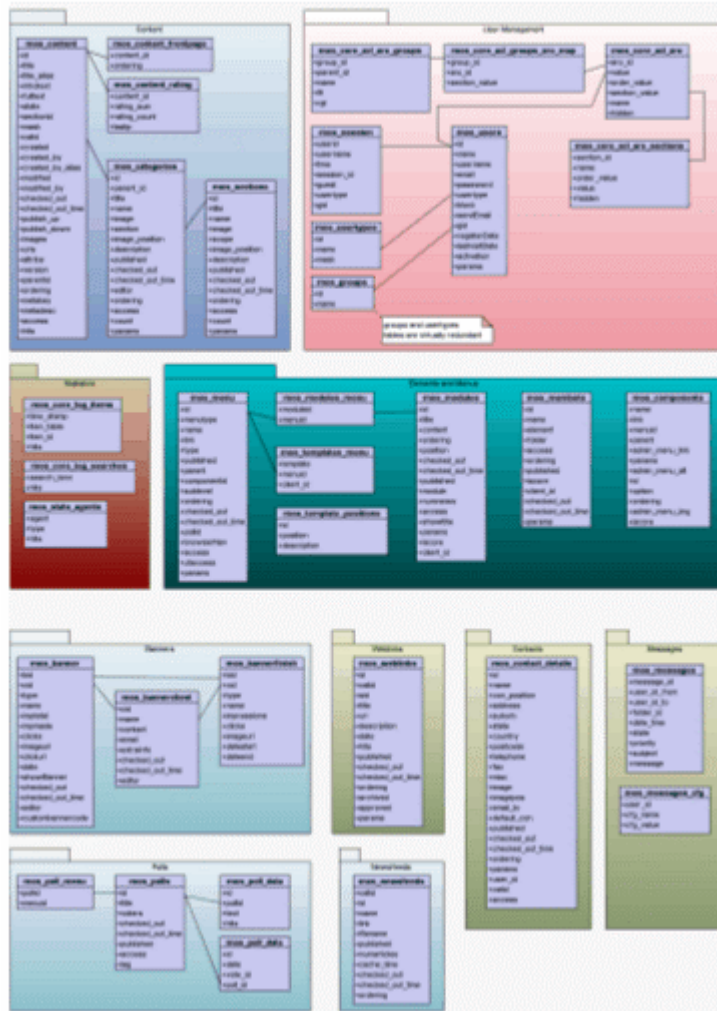


Рис. 2.1. Фізична модель БД

Адміністративна частина повинна включати: інформацію про нових користувачів, розділ налаштувань сайту, можливість адміністрування магазину, можливість виведення статистики по магазину, управління обліковими записами користувачів і інші можливості.

Адміністративна частина сайту представлена у вигляді системи управління вмістом.

### 2.3. Огляд засобів структурування даних та їх використання в інформаційній системі

Для того, щоб дані чи інформація виявились корисними, нам потрібно організувати їх у цілісну структуру. Використовується два типи баз даних: реляційні та нереляційні. У реляційній базі дані зберігаються в окремих таблицях, які зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість поєднувати при виконанні запиту дані з декількох таблиць.[12]



Рис.2.2. Вигляд реляційної бази даних.

Нереляційні або NoSQL бази даних використовуються тоді, коли існує необхідність зберігати великі обсяги неструктурованих даних, що можуть відрізнятися типом між собою. Також використання NoSQL бази даних забезпечують більш швидку їх розробку.

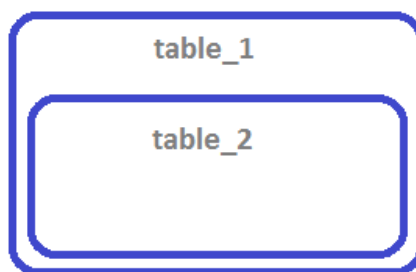


Рис.2.3. Вигляд нереляційної бази даних.

Серед систем управління реляційними базами даних найчастіше використовується MySQL. Мова запитів під назвою SQL була розроблена для полегшення виявлення, маніпулювання та переорганізації реляційних даних.

MySQL - це система управління реляційними базами даних з відкритим

кодом, що базується на використанні мови SQL. Завдяки своїй продуктивності, надійності та простоті у використанні MySQL найчастіше використовується для веб-додатків на таких ресурсах, як: Facebook, Twitter, YouTube. Крім того, це надзвичайно популярний вибір як вбудована база даних, що поширюється тисячами постачальників програмних засобів та оригінальних виробників.[13]

Застосовувати його і модифікувати може будь-хто. Таке ПЗ можна отримувати за допомогою Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

MySQL складається з двох частин: серверної і клієнтської. Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережеских засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

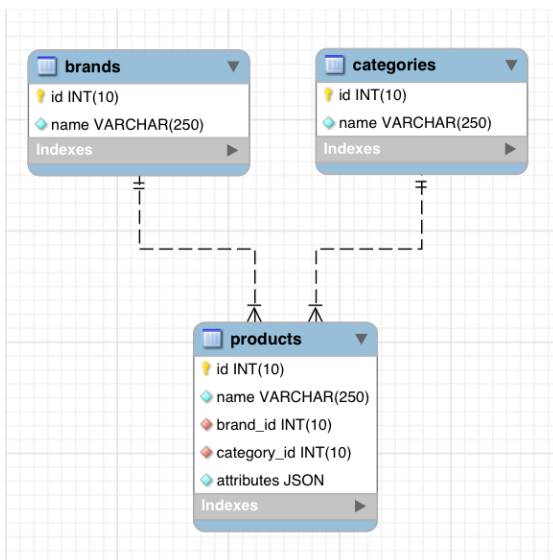


Рис.2.4. Приклад структури БД MySQL

Структура MySQL трирівнева: бази даних - таблиці - записи. Бази даних і таблиці MySQL фізично представляються файлами з розширеннями frm, MYD, MYI. Логічно таблиця являє собою сукупність записів. А записи - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. Один сервер MySQL може підтримувати одразу декілька баз даних, доступ до яких може розмежовуватись логіном і паролем.

Серед систем управління NoSQL-базами даних найбільшого поширення набула MongoDB. MongoDB це крос-платформна, гнучка, документо-орієнтована база даних з відкритим кодом, що легко масштабується.

MongoDB зберігає дані у вигляді JSON-документів, згрупованих в «колекції». Збережений в MongoDB JSON документ називається двійковим JSON або BSON і, як і будь-який інший документ цього формату є неструктурованим.[14]

Умовно структуру бази даних можна представити як безліч колекцій, що містить у собі безліч документів, котрі в свою чергу містять пари ключ-значення. Документ можна представити як об'єкт, що зберігає деяку інформацію. Документ має динамічну схему, що означає:

Документи в одній і тій же колекції не мають імені однакової кількості парного ключа.

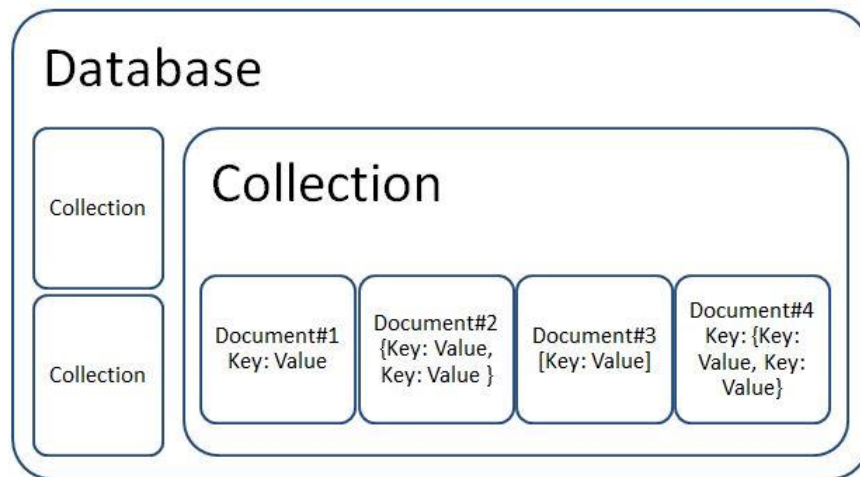


Рис.2.5. Вигляд структури БД MongoDB

Крім баз даних, широке поширення набули файли типу JSON.[15] JSON - це короткий термін для JavaScript Object Notation і є способом організованого, легкого доступу до інформації. JSON - це формат, який зберігає структуровану інформацію і в основному використовується для передачі даних між сервером і клієнтом.

Файл є в основному більш простою і легкою альтернативою XML (розширена мова розмітки), яка має подібні функції.

Розробники використовують JSON для роботи з AJAX. Ці формати добре працюють разом для досягнення асинхронного завантаження збережених даних, тобто веб-сайт може оновлювати свою інформацію, не оновлюючи сторінку.[16] При цьому перевагою JSON перед XML є те, що він дозволяє використовувати складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

```
{ "empinfo" :  
  {  
    "employees" : [  
      {  
        "name" : "James Kirk",  
        "age" : 40,  
      },  
      {  
        "name" : "Jean-Luc Picard",  
        "age" : 45,  
      },  
      {  
        "name" : "Wesley Crusher",  
        "age" : 27,  
      }  
    ]  
  }  
}
```

Рис.2.6. Приклад структури файлу формату JSON

Об'єкт JSON являє формат даних типу ключ-значення, який зазвичай формується в фігурних дужках. Цей формат виділяється двома фігурними дужками, які виглядають так {}, а дані в форматі ключ-значення вже знаходяться між ними.

Ключі в JSON знаходяться з лівого боку від двокрапки. Їх потрібно позначати лапками і це може бути будь-який рядок. У кожному об'єкті, ключі повинні бути унікальними.

JSON значення знаходяться з правого боку від двокрапки. Значення можуть являти собою один із шести типів даних: рядком, числом, об'єктом, масивом, булевим значенням або null. Кожен тип даних, який передається як значення в JSON підтримує свій власний синтаксис, так що рядки будуть в лапках, а цифри лапками не виділяються. Прийнято створювати JSON файли в багаторядковому форматі, що робить його більш читабельним, особливо файл містить в собі великий набір даних.

Найважливішою властивістю JSON є те, що він може легко передаватися між мовами програмування в форматі, який розуміють практично всі мови.[17]

## 2.4. Методи табличного верстання інтернет-сайтів

Таблична верстка сайту – це різновид верстки, при якій елементи сторінки сайту являють собою елементи таблиці. У минулому основний і найбільш популярний спосіб. Весь метод зав'язаний на тегу table і його дочірніх елементах, таких як рядки та клітинки. Вся структура сайту розмічається як велика таблиця.

Основні теги:

- Тег <table>, декларує таблицю;
- Тег <th>, заголовок таблиці;
- Тег <tr>, рядок таблиці;
- Тег <td>, одна комірка таблиці;
- Тег <col>, одна або декілька колонок.

Переваги:

- Такий підхід дозволяє достатньо легко створити багато колонок і рядків.
- Тег table і всі пов'язані з ним елементи підтримуються абсолютно однаково у всіх браузерах.

Недоліки:

- Для відображення таблиці необхідно безліч тегів, крім того, її важко редагувати;
- Якщо використовувати як метод верстки усього сайту – він буде важким: сторінки багато важитимуть, повільно завантажуватися;
- Табличні дані дуже складно адаптувати для перегляду на різних пристроях.[18]

```

<table>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
</table>

```

Рис.2.7. Приклад вигляду таблиці

## 2.5. Використання методу float для створення Web-додатку

Властивість float в CSS, дозволяє розробникові використовувати подібні таблиці стовпчики в розмітку HTML без використання таблиці.

Мета властивості float - притиснути елемент до лівого або правого краю батьківського елемента, а найближчий текст обходить елемент з інших сторін. Така поведінка тексту нагадує потік води, що оминає камінь, тому елементи з поплавком називають плаваючими або поплавками. Ця концепція аналогічна тій, яка представляється кожен день у друкованій літературі, де є фотографії та інші графічні елементи, вирівняні за будь-якою стороною, інформаційне наповнення (зазвичай текст), знаходиться навколо елемента, вирівняного навколо лівого або правого краю.[19]

Властивість Float може використовувати одне з 4 значень: вліво (left), вправо (right), без вирівнювання(none) та успадковане (inherit):

- Значення left у властивості float вирівнює елемент по лівому краю браузера, а всі інші елементи обтікають її по правій стороні.

- Значення `right` у властивості `float` вирівнює елемент по правому краю браузера, а всі інші елементи обтікають її по лівій стороні.
- Значення `none` у властивості `float` вказує на те, що обтікання елементу не задається.
- Значення `inherit` у властивості `float` вказує на спадковість властивості від свого батьківського елемента.

## 2.6. Модульна система верстки у веб-дизайні

Модульна система верстання веб-сторінок – це система верстання, при якій основою композиції смуг і шпальт стає модульна сітка з певним кроком (модулем), однаковим або різним по горизонталі і вертикалі. Модульна система спрощує і прискорює художнє конструювання і створює сприятливі умови для автоматизації верстки при використанні комп'ютерних настільно-видавничих систем.[20]

Веб-дизайнери відносно недавно почали застосовувати модульні сітки в проектуванні інтерфейсів веб-сайтів. Цей метод значно спрощує як проектування інтерфейсу, так і подальшу верстку макета. На відміну від друкарні, модулі в веб-дизайні можуть мати непостійну ширину і розтягуватися в залежності від ширини вікна браузера (дозволу екрану монітора).

Начастіше як шаблон сітки використовується CSS-фреймворк Bootstrap[21]. В сітці цього фреймворку 12 колонок, при кастомізації цього фреймворку можна виставити будь-яке інше значення, але користувача влаштовує 12 колонок. Вся сітка розміщується в загальному контейнері, з класом `container`, що має фіксований розмір 1170px. Це контейнер завжди розтягується на 100% ширини вікна, тобто сайт буде гнучким.

Всередині контейнера є блок із класом `row`, тобто 1 рядок сітки. В самому рядку розміщаються колонки. Колонка має клас `col-x-x`, де перший `x` –

визначення пристрою, а другий – кількість колонок от 1 до 12. Для найменших пристроїв, у яких ширина екрану менше 768 пікселів, префікс класу - col-xs- або якщо відкинути col, просто xs. Далі йде sm (small-devices, ширина від 768 до 991 пікселів), md (medium-devices, ширина від 992 до 1199 пікселів) і large-devices, з товщиною не менше 1200 пікселів.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Рис.2.8. Приклад сітки фреймворку Bootstrap 4

## РОЗДІЛ 3. ОПИС КОМПЛЕКСНОГО ВИРІШЕННЯ ЗАДАЧІ

### 3.1. Проектування БД

Весь зміст системи а також її основні дані, тобто дані про тести і опитування зберігається в базі даних.

База даних складається з наступних таблиць: «Користувач», «Респондент», «Питання опитування», «Результати опитування».

Кабінет користувача — це основне робоче місце користувача в системі. Він включає в себе інші підмодулі, які виконують основні функції системи, та є елементом компонування в системі.

Структура таблиці “Користувач” представлена наступним чином (таблиця 3.1):

Таблиця 3.1.

Структура таблиці «Користувач»

Ім'я поля	Тип і розмір поля	Опис поля
Id	int	Первинний ключ
Surname	varchar	Прізвище
Name	varchar	Ім'я
Patronymic	varchar	По батькові
UserName	varchar	Логін користувача
Password	varchar	Пароль
Status	int	Тип користувача

Підмодулі кабінету користувача:

До цих модулів належать наступні:

- модуль керування відповідями;
- модуль керування опитуваннями;

— модуль керування даними респондентів.

Модуль керування відповідями

Даний модуль дозволяє користувачу додатку керувати наборами тестів, а саме:

- створювати нові тести (для викладачів);
- проходити тестування (для студентів);
- переглядати результати тестів;
- видаляти застарілі дані про тести (для адміністраторів).

Модуль керування опитуваннями Даний модуль дозволяє користувачу додатку керувати наборами опитувань, а саме:

- створювати нові опитування (для розробників);
- проходити опитування (для респондентів);
- переглядати результати опитувань;
- видаляти застарілі дані про опитування (для адміністраторів).

Модуль керування даними респондентів

Даний модуль дозволяє користувачу додатку керувати власними даними, а саме:

- реєструвати у системі нових членів сім'ї (для адміністраторів);
- змінювати логін і пароль до облікового запису;
- змінювати основні дані респондентів (для адміністраторів);
- видаляти облікові записи респондентів (для адміністраторів).

### **3.2. Інтерфейс користувача**

Оскільки застосунок працює з використанням веб-технологій, він не потребує встановлення на пристрій користувача. Проте, для використання необхідний веб-браузер, який підтримує актуальні веб-стандарти. Також необхідно мати стабільний доступ до інтернету з швидкістю не менше ніж 50 мбіт

за секунду. Для використання можливостей програми з проведення і опитувань необхідно дати клієнтському додатку відповідний дозвіл.

При вході на клієнтський додаток, користувачеві необхідно авторизуватися в системі щоб почати працювати в системі. На рисунку 3.1 зображена форма авторизації.



The image shows a login form with a blue header bar containing the text "Вхід у систему". Below the header, there are two input fields. The first field is labeled "Логін" (Login) and has a person icon to its left. The second field is labeled "Пароль" (Password) and has a lock icon to its left. At the bottom right of the form, there is a button labeled "УВІЙТИ" (Login).

Рис. 3.1. — Форма авторизації

У випадку якщо користувач ще ніколи не користувався системою або хоче створити новий профіль, то йому необхідно зареєструватися в системі. На рисунку 3.2 зображена форма реєстрації.

The image shows a registration form titled "Реєстрація" (Registration) in a blue header. Below the header are three input fields: "Логін" (Login) with a person icon, "Пароль" (Password) with a lock icon, and "Підтвердіть пароль" (Confirm password) with a refresh icon. A grey button labeled "ЗАРЕЄСТРУВАТИСЯ" (REGISTER) is positioned to the right of the fields. The form is enclosed in a thin grey border.

Рисунок 3.2 — Форма реєстрації

Після того, як користувач авторизувався в системі, він отримує доступ до головного меню системи. За допомогою цього меню користувач має доступ до усіх сторінок системи, або має можливість вийти з свого профілю. Також він може перейти до свого основного робочого простору — кабінету користувача, та обрати функцію проходження опитування для національного перепису.

### **3.3. Алгоритмізація та реалізація комплексу задач автоматизації.**

Перед тим як створювати web-орієнтовану систему ПрАТ «Білоцерківський консервний завод» для підтримки клієнтури, необхідно створити макет системи та перевірити його на адекватність. Після цього необхідно перевірити її відповідність заданим вимогам замовника.

Перш за все необхідно визначитись з програмним забезпеченням, в середовищі якого буде розроблятися web-ресурс. Під час вибору належного програмного забезпечення потрібне проведення аналізу ефективності кожної з позицій. Тільки в цьому випадку можна очікувати результативності та ефективності роботи самого процесу розробки та виправданості бюджету. З

особистого досвіду та гарної обізнаності в програмі “Sublimetext 3” вибір було зроблено в його користь.

Дана програма є кодовим редактором з розширеними можливостями, автозаповненням, підключенням бібліотек та завантаженням коректора будь-якої необхідної для розробки мови програмування.

При визначенні цільової аудиторії сайту, слід врахувати, що загальні запити з'являються в системах пошуку часто, при цьому покупцями стають в основному ті користувачі, які приходять за запитом вужчого і більш конкретного характеру, так як саме цей користувач уже готовий до певного роду інформації, тому головною метою дизайну сайту є виникнення бажання у клієнта залишитись на сторінці та ознайомитись з наведеною інформацією, що в повинно призвести до інвестування або просто подальшого розповсюдження сайту серед власного оточення.

Відповідно до типів цільових аудиторій сформувалися і стратегії, орієнтовані на той дизайн та макет сторінки, який був би новим, простим у використанні та привабливим для користувача.

### **3.2.1. Розробка макету десктоп-версії web-ресурсу**

Наступним етапом буде створення дизайну та макету сайту. Так як створюваний web-ресурс необхідно створювати як односторінковий сайт, з цього випливає ряд важливих моментів, які необхідно враховувати до створення макету.

Оскільки процес створення макету буде поділений на створення десктопної версії сайту та мобільної, необхідно визначити ці поняття.

Десктопна версія – це та версія сайту, яку використовують користувачі, відкриваючи сторінки за допомогою ноутбуків або персональних комп'ютерів [9].

Перш за все, перший екран або ж верхівка сторінки повинна коротко але змістовно інформувати відвідувача про головне та давати можливість зручно пересуватись сайтом, полегшуючи його взаємодію та зацікавлюючи в переході на іншу сторінку зі збором інвестицій.

Аналізуючи отримані матеріали для розміщення, а саме: текстова інформація, зображення, гіф-анімація та відеоматеріали, було прийнято рішення, компонувати їх одне з іншим для досягнення різноманіття можливостей під час перегляду сторінки та альтернатив під час вибору способу перегляду матеріалів.

Необхідно створити мінімалістичний перший екран, котрий буде відображувати поточний прогрес в процесі збору коштів та пропонувати переглянути наведену нижче на сторінці інформацію.

Оскільки в розпорядженні наявні двоє готових та повністю завершених відеоматеріалів, було прийнято рішення використати один з них на першому екрані. Він повинен одразу приковувати до себе погляди відвідувачів, тому було запропоновано зробити його більше половини ширини екрана.

В іншій, вільній частині екрану було вирішено розмістити інформацію про кількість вже зібраних коштів та загальної необхідної для завершення інвестування суми, а також кнопки навігації, кожна з яких виконує гіперпосилання на свою інформацію. Перша з них з назвою «Button1» повинна при натисканні робити плавний перехід від першого екрану до матеріалів, котрі розміщуються нижче. Це буде реалізовано за допомогою посилань всередині одного документу, а точніше якорю, котрий буде встановлений на певному місці в документі. Головною ціллю існування якорів в документі є те, що вони

дозволяють атрибутам робити гіперпосилання не на сторонній документ або сторонню web-сторінку, а саме на окремо виділені місця в тому ж документі, дозволяючи робити навігацію користувачу по одній сторінці.

Найголовнішою частиною першого екрану було вирішено зробити саме кругову діаграму, що відображає поточний прогрес в процесі збору коштів. Її реалізація буде вимагати підключення сторонніх модулів, створення запитів та отримання відповідей від іншого серверу та використання API-технологій. Зовнішнє оформлення кругової діаграми буде реалізовано за допомогою CSS та її взаємодії з функціями мов JavaScript та PHP.

Наведений на Рис. 3.3. приклад гарно поєднує визначені вимоги до першого екрану, тому він буде використовуватись як макет першого екрану.

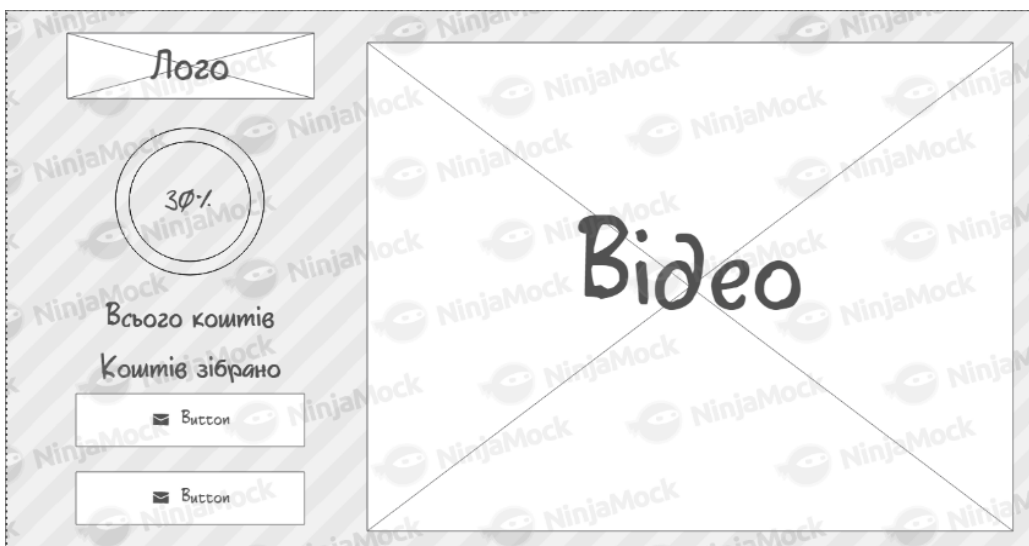


Рис. 3.3. Макет першого екрану сторінки

Матеріали будуть викладені послідовно не суміщаючи в одному ряді інформацію різного типу, адже враховуючи невеликий об'єм для розміщення

даних, необхідно повністю розкрити суть проекту та залучити користувача стати клієнтом.

З метою обмежити можливість користувача перейти одразу в кінець сторінки було прийняте рішення прибрати полосу прокрутки. Таким чином користувач зможе сприймати інформацію послідовно та краще її засвоювати. Для допомоги в навігації без полоси прокрутки допоміжним рушієм буде «Кнопка 2», котра ініціює перший перехід до нижченаведеної інформації.

Після розміщення всієї інформації, останніми елементами сторінки будуть саме зображення команди розробників, та футер з посиланнями на сторінки в соціальних мережах.

Футер – у web-програмуванні це нижній колонтитул web-сторінки, розділ, що розташований під основним текстом або останнім елементом. Зазвичай він використовується в якості місця для посилань на сайти партнери або на інші важливі для web-ресурсу сторінки.

Але в односторінкових сайтах зазвичай футер або містить інформацію про розробників, або просто контакти такі як телефон та e-mail, або ж посилання на сторінки в соціальних мережах чи на сторінку завантаження мобільного додатку.

Оскільки даний проект має створені та функціонуючі сторінки в таких соціальних мережах як Instagram та Facebook, в футері буде розміщено саме посилання на ці сторінки.

Отже, створивши повний макет сайту для десктопної версії, можна переходити до створення макету мобільної версії.

Розташування інших не згаданих елементів буде залишене в такому ж порядку та будуть збережені всі пропорції з першого макету.

Отже, в результаті проведеної роботи було створено два макета розроблюваного web-ресурсу: макет десктопної версії та макет мобільної версії сайту. За отриманими макетами будуть розроблені відповідно дві версії сайту та поєднані в одну остаточну, що буде постійно перевіряти розширення екрану користувача, тим самим розуміючи, яку з двох версій потрібно створювати в даний момент часу. Поєднання двох макетів в одну версію допоможе скоротити витрати ресурсів та часу на створення web-ресурсу.

Тому створені раніше макети є невід'ємною частиною розробки web-орієнтованої інформаційної системи для підтримки дрібноорієнтованих клієнтів в покупців товарів харчування ПрАТ «Білоцерківський консервний завод».

#### **1.4. Технічне та системне забезпечення розробки.**

##### **1.4.1. Налаштування серверу**

Проведені в попередніх розділах дії дозволили визначити нам спосіб створення сучасного та конкурентоспроможного web-сайту інформаційної підтримки проведення рекламних акцій.

Першим, під час розробки web-ресурсу, необхідно визначити та налаштувати апаратну частину. Це може бути як безкоштовний хостинг, так і приватний сервер. Оскільки розроблюваний продукт виготовляється для конкретної компанії «LLC InbicoEurope» та вони мають власний орендований приватний сервер, на котрому бажають розмістити web-сайт.

Створивши папку ПрАТ «Білоцерківський консервний завод», та виставивши йому в налаштуваннях рівень доступу root, можна переходити до створення та подальшого завантаження необхідних матеріалів на сам сервер.

Обов'язковим етапом створення web-сайту є створення декількох файлів, таких як: *sitemap.xml* та *robots.txt*. Web-ресурс звісно може існувати без цих двох

файлів, проте витративши небагато часу на їх створення, адміністратор може забезпечити сайту кращі умови існування.

*Sitemap* або ж XML-карта сайту –це файл з інформацією для пошукових систем про сторінки, які необхідно проіндексувати. Іншими словами, карта сайту - список всіх сторінок в форматі XML, доступних для сканування пошуковим роботом.

Оскільки наш сайт буде створено односторінковим, цей етап можна пропустити, переходячи одразу до створення наступного файлу – robots.txt.

Файл *robots.txt* або індексний файл - звичайний текстовий документ в кодуванні UTF-8, діє для протоколів HTTP, HTTPS, а також FTP. Файл дає пошуковим роботам рекомендації: що варто сканувати. Якщо файл буде містити символи не в UTF-8, а в іншому кодуванні, пошукові роботи можуть неправильно їх обробити. Правила, перераховані у файлі robots.txt, дійсні лише щодо того хоста, протоколу і номера порту, де розміщений файл.

Файл повинен розташовуватися в кореневому каталозі в вигляді звичайного текстового документа.

Аналогічно попередньому файлу, так як наявна лише одна сторінка то прописати в цьому файлі необхідно саме те, що повинні знаходити пошукові роботи при запиті користувача в пошуковій системі.

Гарним тоном для будь-якого розробника буде створення окремих каталогів в папці сайту на сервері для правильного місцезнаходження файлів та документів, котрі будуть використовуватись. Тому було створено наступні необхідні для цього директорії.

Отже, створивши необхідні для існування web-ресурсу каталоги, отримання необхідних прав доступу та створення файлу *robots.txt*, етап налаштування серверу вважається завершеним.

### **3.4.2. Розробка та верстка сайту**

#### **3.4.2.1. Створення розмітки**

Оскільки головним інструментом розробки був обраний «SublimeText 3», тому майже весь процес створення та написання коду буде відбуватись саме там.

Маючи макет сайту, можна переходити до розмітки сторінки.

Розмітка сторінки, як впливає з самого терміну – це розстановка міток, в нашому випадку в кодї HTML документа, тобто web-сторінки. Мітками тут виступають теги, що дозволяють визначити межі дії розмітки або створити елемент HTML документа.

В свою чергу перший блок буде складатись з горизонтальних підблоків, кожен з яких буде містити відповідно один елемент. Вони будуть також мати відносні розміри у відсотках, що допоможе забезпечити адаптивність сайту під різні розширення екранів користувачів. На Рис.3.8. зображена попередня розмітка першого екрану.

На ній можна побачити анімаційно орієнтовну колонку зміну зображень головної сторінки сайту, яку в народі іноді називають «каруселею». Серед текстового представлення – заголовок сайту, а під ним – привітання від імені підприємства-власника сайту.

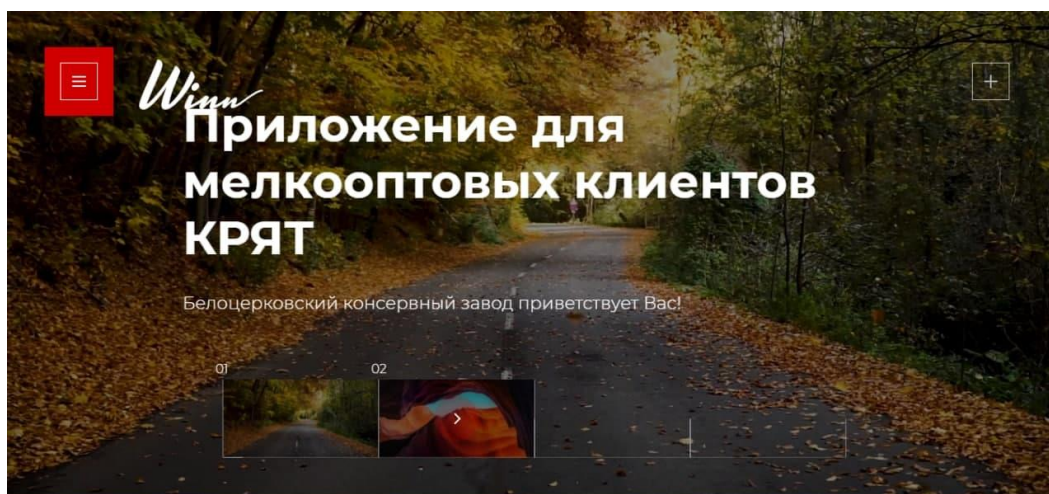


Рис. 3.4. Розмітка першого екрану сайту

Наступним етапом, є розмітка іншої частини сайту, котра буде містити всю інформацію, що необхідно донести до користувача. Ще з макету впливало те, що ц частина сторінки буде мати певні поля, для кращої читабельності інформації та візуальної централізації елементів.

Тому вся інша сторінка буде поділена на 3 колонки на які відповідно буде виділено 15% на першу, 70% на центральну та 15% на останню. Перша та остання не будуть містити жодної інформації, вони будуть забезпечувати однакові поля для центральної, котра і буде в свою чергу містити всю інформацію.

Розмітка центральної частини буде максимально простою, що іноді є корисним, адже чим менше непотрібних елементів на сторінці, тим швидше відбувається її завантаження в браузері. Кожен наступний елемент буде розміщуватись за попереднім. Для всіх блоків будуть використовуватись теги `<div>` в середині яких вже будуть розміщені відповідні елементи такі як: текст, зображення, анімація або ж відео.

Під цим великим блоком, найнижчим та останнім блоком сторінки буде окремий блок для футера, що буде містити посилання на сторінки в соціальних

мережах. Він буде мати ширину 100% від ширини сторінки та відносно невелику висоту.

Останнім в розмітці сторінки необхідно ще раз повторно передивитись всю щойно створену структуру. Якщо ніде нічого не було залишене з відкритим тегом або ж просто без необхідного блоку, то можна переходити до наступного етапу.

### **3.4.2.2. Наповнення сторінки вмістом**

Після створення розмітки сторінки, можна переходити до її наповнення необхідною інформацією.

Отримані матеріали від розробника проекту одразу необхідно поділити за категоріями. Всі зображення та гіф-анімація будуть зберігатись на сервері в окремій папці «*images*», в той час як відеоматеріали будуть знаходитись в теці «*video*». Ці прості розмежування файлів, що використовуються як елементи сторінки, дозволяють розробникам краще підтримувати структуру web-сторінок. Якщо буде якась необхідність швидко змінити якийсь елемент, то навіть адміністратор, котрий не розробляв сайт, зможе знайти місцезнаходження шуканого файлу.

Текстову інформацію одразу необхідно розміщувати прямо у відповідних тегах. Зображення та гіф-анімація також розміщуються просто в тегах за допомогою власного тегу `<img>`, ціль існування якого і є в завантаженні зображень на сторінку.

Але існує і другий, менш поширений та менш зручний спосіб вбудовування відео на сторінку. Використання HTML5 плеєру. Цей плеєр є більш зручним для змінення його розмірів, пропорцій та положення на сторінці, проте він вимагає завантаження всього відео на сервер. Якщо сервер має слабку потужність, то він буде погано завантажувати відео в програвачі. Але на відміну від програвача

YouTube в даному плеєрі все залежить від серверу сайту, тому він є повністю незалежним.

І хоча перший спосіб є набагато простішим у використанні, проте для виконання вимог даного проекту використовуватись буде саме другий спосіб з плеєром від HTML5. Оскільки в проєкті будуть присутні два відео, вони обоє будуть завантажені даним методом.

Для внесення новизни та індивідуальності в сторінку, було вирішено змінити можливості користувача у взаємодії з медіа плеєром. А саме прибрати у відео програвача полосу прокрутки. Таким чином користувач матиме можливість продивитись відео лише повністю від початку до кінця.

Реалізація цього дуже проста, вона вимагає відключення одного з атрибутів тегу `<video>`, а саме написання наступного коду в файлі стилів CSS:

```
video: { display: none !important; }
```

Для кращого оформлення головного відео з першого екрану, було вирішено створити модальне вікно, в якому буде завантажуватись відео.

Модальне вікно відкривається при натисканні на фрейм першого відео. На ньому розміщується централізований блок, що містить в собі програвач відео.

Для виокремлення цього блоку від сторінки на фоні використовувалось заповнення всього фону напівпрозорим елементом, котрий містив в собі функцію, що закривала модальне вікно при натисканні, тим самим забезпечивши можливість користувача закрити відео та просуватись й далі по сторінці.

Проте головною проблемою використання даного прийому є те, що після цього користувач втрачає можливість запуску та паузи відео. Тому головним пріоритетом стала розробка функції, що дозволяє запускати та зупиняти програвач при натисканні на фрейм з відео.

Для створення даної функції використовувалась мова JavaScript та її бібліотека jQuery.

Реалізація цієї ідеї була можлива за допомогою функцій бібліотеки jQuery, а саме:

1) перш за все, необхідно створити функцію, котра буде запускатись після натискання користувачем в рамках фрейму з відео;

2) після цього йде перевірка умови: *if (radio == true)*, якщо значення змінної *true*, тоді виконується пункт №3 з запуском відео, якщо ж значення змінної інше, то виконується пункт №4.

3) при виконанні умови (*radio == true*) запускається наступна функція: *\$('#vidplay1').get(0).play()*, котра виділяє елемент з *id=«vidplay1»*, отримує доступ до керування відео через функцію *get(0)* та запускає відео через вбудовану функцію *play()*. Після чого обов'язковою є заміна значення змінної *radio* на *false*, адже при наступному натисканні, необхідно буде уже зупиняти відео.

4) В разі невиконання умови (*radio == true*), а оскільки змінна типу *bool* може приймати лише два значення, то при (*radio == false*) запускається наступна функція: *\$('#vidplay1').get(0).pause()*, котра виділяє елемент з *id=«vidplay1»*, також отримує доступ до керування відео через функцію *get(0)* та зупиняє відео через вбудовану функцію *pause()*. Після цього також обов'язковою є заміна значення змінної на протилежну.

Дана функція копіюється та дещо видозмінюється для наступного відео, проте її функціональність та спосіб реалізації залишаються такими ж. Таким чином, була розроблена функція, що дозволяє керувати відео без полоси прокрутки програвача.

Щоб користувач зрозумів, що по відео необхідно натиснути, була використана можливість, надана атрибутом *poster*, котрий дозволяє встановлювати будь яке зображення як заставку на відео до його першого запуску, тоді *poster* пропадає і відео йде у звичайному режимі. Заставка навмисно була створена таким чином, щоб вона нагадувала вікно програвача та давала візуальні підказку користувачеві в запуску відео.

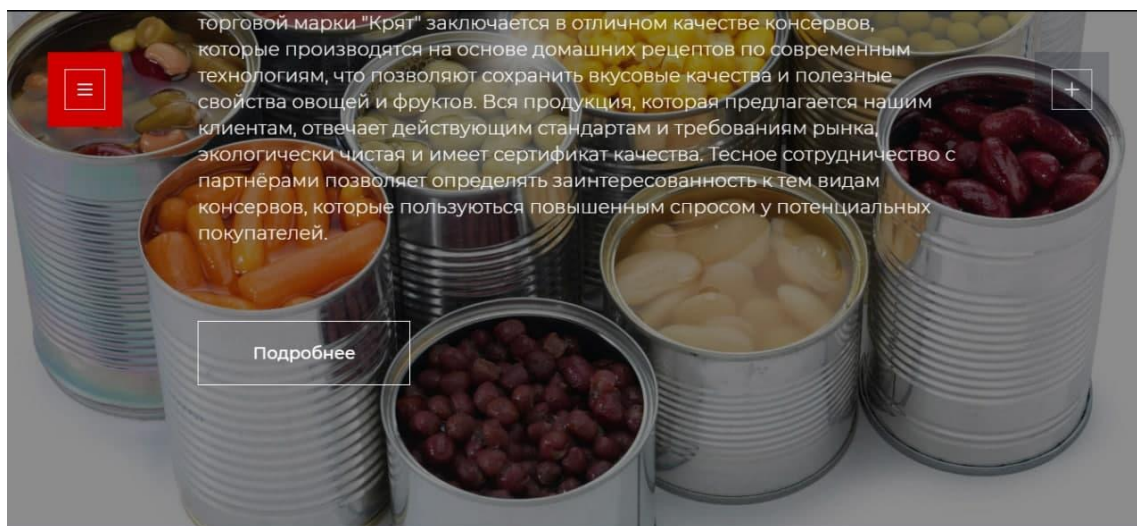


Рис. 3.5. Наповнення сторінки вмістом

### 3.4.2.3. Створення стилів web-орієнтовної інформаційної системи

Створення дизайну сайту – найголовніший етап в створенні сторінки. Саме від дизайну, грамотного поєднання кольорів, розстановки матеріалів, шрифтів, зображень, поєднання їх в одну композицію і робить сторінку привабливою для користувача. Саме дизайн показує рівень розробника та цінність компанії. Тому до цього питання завжди слід підходити ретельно [13].

Оскільки макет та розмітка сторінки вже створені, необхідно було підібрати необхідні стилі, які б задовольнили вимоги замовника та перетворили звичайний каркас розмітку на візуально завершений продукт.

Перш за все, для забезпечення повної ширини сайту та неможливості користувача пропустити важливу інформацію, було вирішено прибрати бічну полосу прокрутки. Реалізується це просто, прописавши в стилях наступний код: *html: overflow hidden*.

Наступним важливим кроком є підбір шрифтів, котрі будуть використовуватись на сторінці. Завдяки можливості обирати серед більш ніж восьмисот шрифтів на сайті [fonts.google.com](https://fonts.google.com), було відібрано три головних шрифти, котрі будуть застосовані в різних частинах сайту.

Перший шрифт буде використаний на першій сторінці відображаючи дані по збору коштів, другий буде застосований до заголовків та написів на кнопках, а третім буде відформатований весь текстовий матеріал з блоків.

Прописавши елементам з однаковим оформленням класи та ідентифікатори можна переходити в файл *style.css*, де зберігаються та визначаються стилі сторінки. Там необхідно прописати всі належні стилі, вирівнювання тексту, положення елементів, відносні та абсолютні розміри елементів, центрування існуючих блоків, заміну кольорів тексту та фонових елементів. Таким чином відбувається процес стилізації сайту та його перетворення з макету на готовий кінцевий продукт.

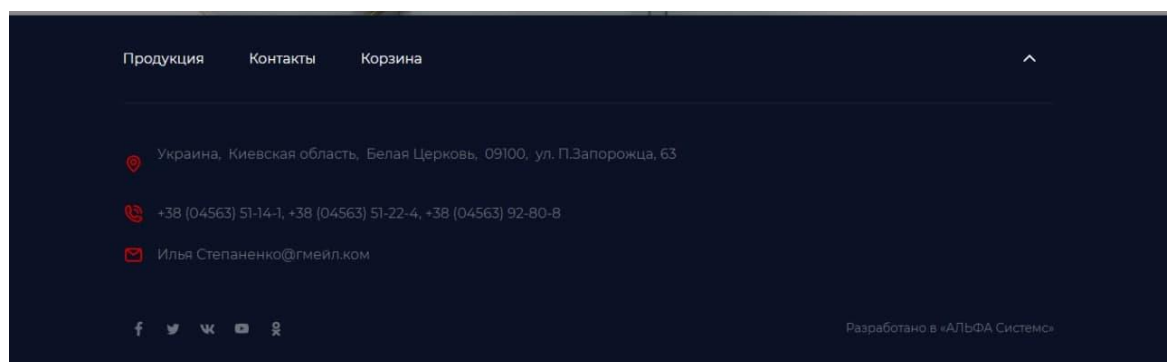


Рис. 3.6. Футер з гіперпосиланнями

#### 3.4.2.4. API-технології та їх значення у web-орієнтовній ІС

Завершивши верстку сайту, тобто створення візуальної частини сайту або *front-end*, можна переходити до створення технічної частини розробки. Для використання АРІ-технологій необхідна певна підготовка.

Перш за все необхідно, щоб головний файл сайту мав розширення не *.html*, як у звичайних сторінок, а *.php*, щоб мати можливість використовувати функції мови РНР.

Необхідно перевірити, чи є в розробника всі належні права доступу. В разі їх відсутності, необхідно перевірити та задати на сервері повторне привласнення прав між користувачами та адміністраторами серверу.

Після цього необхідно створити папку, котра буде містити код, котрий буде робити запит, отримувати та розкодовувати відповідь та передавати її в головний файл для використання в виведенні інформації на екран.

Для реалізації запиту була використана розроблена раніше технологія для отримання відкритої інформації з web-сайтів.

Для того, щоб вона могла виконувати свої функції, їй необхідно мати посилання на конкретну сторінку, котру необхідно прописати в правильному місці в коді головної сторінки. Тоді відбувається звернення до вказаного сайту, де з допомогою прописаних функцій відбувається пошук потрібного елемента серед всіх елементів сайту. Після знаходження відповідного елемента, функція переходить до того, що виокремлює текстову частину, що знаходиться в даному блоці та копіює її в свій буфер обміну, після чого надсилає собі ж назад на сервер[18].

Отримавши дані, починається процес їх обробки. Він поділяється на декілька етапів:

1. Перевірка отриманих даних. Після отримання даних необхідно одразу перевірити їх на коректність та відповідність цифровому типу. Якщо серед вивантажених з буферу обміну даних є не лише числова інформація, тоді відбувається перехід до етапу №2.

2. Якщо серед числової інформації є текстові, або інші недопустимі символи, тоді отримані дані перетворюються з числового типу даних *int* в текстовий тип *string*.

3. Наступним кроком є виокремлення виключно числової інформації та видалення всього зайвого з допомогою вбудованих функцій та використання циклу *for()*. Він проходить по кожному символу змінної *string* та визначає які символи є числами.

4. Після виділення позицій виключно числових даних, відбувається видалення зайвих символів, та конвертація отриманого результату до числового формату *int*.

5. Створення копії даної змінної з отриманою інформацією, для забезпечення надійності в подальшій роботі з даними.

Таким чином, отримана інформація перетворюється на коректні дані, що будуть використовуватись в побудові діаграми.

Після отримання оброблених коректних даних, можна переходити до їх вставлення на сторінку.

Для виведення наявної інформації на екран буде використовуватись самий звичайний метод мови PHP *echo()*, котрий просто виводить на екран інформацію. Для правильного оформлення, використання цієї функції буде одразу прописане в тег, де повинна буде знаходитись інформація, а за допомогою стилів, котрі будуть надані блоку в якому це відбуватиметься, можна буде легко форматувати

та змінювати текст для відповідності всьому стилю сайту. Нижче наведено приклад виведення інформація на сайт

```
<?php echo $project -> percentFunded >
```

За допомогою функцій мови PHP стала можлива передача змінних між різними файлами, що знаходяться на сервері. Саме завдяки цьому передача одразу багатьох змінних не викликає жодних проблем та навіть не знижує швидкість завантаження сайту, адже вся передача даних відбувається в асинхронному потоці, котрий проходить одразу паралельно до завантаження сторінки.

### **3.5. Розробка форми замовлення продукту**

Наступним етапом, можливо навіть найважливішим етапом, є розробка форми для замовлення продукту з розроблюваного web-ресурсу. Звісно, продукція, представлена на створюваному сайті, буде доступна до придбання з полиць магазинів та споживання в відповідних кафетеріях “Prostokava”, проте на даному етапі збору коштів на їх побудову, продукція буде доступна до придбання на даному web-ресурсі.

Перш за все для створення належної форми замовлення, спочатку необхідно створити тег <form> та прописати всі поля вводу текстової інформації. Для цього використовувалось поєднання тегів <input> та <label>.

Після створення та належного оформлення, необхідно було створити та стилізувати кнопки, котрі б відповідали за додання та видалення вкладень до замовлення, а також кнопку відправки форми. Оскільки призначення останньої кнопки значно відрізняється від всіх інших, наявних на сторінці, вона була створена за допомогою тегу <input type=”submit”>та стилізовано під всі інші кнопки, котрі були створені тегом <button>.

Кнопки «Add documents» та «Reset documents» створені відповідно для додавання вкладених файлів та для видалення помилково вкладених файлів. Реалізація їх взаємодії з користувачем була розроблена з допомогою поєднання мови програмування JavaScript та PHP.

Наступним кроком розроблення форми було додавання власне товарів на сторінку. Зважаючи на їх відносно невелику кількість, для кращого візуального оформлення було обрано використовувати повноцінні зображення прототипів продукції. Тому в правій частині екрану було створено блок, котрий містить зображення трьох товарів.

Під кожним з товарів було створено лічильник їх кількості. Натискаючи на стрілки вліво та вправо, користувач може змінювати кількість товару для замовлення.

Після створення, розміщення та стилізації всіх елементів форми, або ж після завершення front-end частини, прийшов час для роботи над back-end частиною. Функції, котрі працюють з елементами вводу інформації, оброблюють його та відправляють на сервер, створювались за допомогою мови PHP.



Рис. 3.7. Форма замовлення продукту

Процедура створення та відправлення форми виглядає наступним чином:

- 1) користувач заповнює текстові поля та обирає необхідну продукцію;
- 2) перший етап завершується лише в момент натискання користувачем кнопки «SEND», котра і є початком роботи PHP функцій на сервері;
- 3) після натискання кнопки «SEND» функції PHP присвоюють новим пустим змінним значення відповідних заповнених користувачем полів.
- 4) використовуючи раніше створений шаблон замовлення, нові змінні із введеними даними вносяться в відповідні місця шаблону, заповнюючи пробіли та формуючи лист замовлення для його відправлення користувачеві на вказану електронну адресу та збереження його копії на сервері для подальшого опрацювання.
- 5) на головній сторінці, після натискання кнопки «SEND» з'являється подяка за замовлення користувачеві та сторінка оновлюється очищаючи всі поля вводу для наступного замовлення.

## РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Згідно статті 8 Конституції України - основним правовим документом України є Конституція України. Конституція України має найвищу юридичну силу. Закони і інші нормативно-правові акти приймаються на підставі Конституції України і повинні відповідати їй. На підставі Конституції України прийнятий Закон України "про охорону праці".

Охорона праці - це система правових, соціально-економічних, організаційно-технічних заходів, а так само санітарно-гігієнічних і лікувально-профілактичних засобів, направлених на збереження здоров'я і працездатності людини в процесі праці. (згідно закону України "про охорону праці" ст.1)

Згідно ст.2. закону «про охорону праці» дія Закона "про охорону праці" розповсюджується на всі підприємства, установи і організації не залежно від форми власності і видів їх діяльності, на всіх громадян, які працюють, а також повернуті до праці на цих підприємствах.

Згідно ст 4.Закона України «про охороні праці» державна політика в області охорони праці визначається відповідно Конституції України Верховною Радою України і направлена на створення належних, безпечних і здорових умов праці, запобігання нещасним випадкам і професійним захворюванням.

За порушення законів і інших нормативно-правових актів про охорону праці, створення перешкод в діяльності посадовців органів державного нагляду за охороною праці, а також представників профспілок, їх організацій і об'єднань винні особи притягуються до дисциплінарної, адміністративної, матеріальної, кримінальної відповідальності згідно закону (ст. 44 Закону "про охорону праці").

При роботі з устаткуванням для зварювання варто виконувати приписання по експлуатації і безпечному обслуговуванню зварювальних установок, наявні в інструкціях заводів виготовлювачів.

При електричній напрузі джерела живлення зварювальних пристроїв вище 12 В для сирих приміщень й 36 В для сухих умов можлива поразка електричним струмом. Зазначена поразка викликає хворобливе відчуття при силі струму 0,002...0,05 А, важкий результат при силі струму 0,05...0,1 А.

Відповідальність за організацію навчання і перевірку знань з охорони праці на підприємстві покладається на його керівника, а в структурних підрозділах — на керівників цих підрозділів. Контроль за навчанням і періодичністю перевірки знань з питань охорони праці здійснює служба охорони праці або працівники, на яких керівником підприємства покладені ці обов'язки.

За характером і часом проведення інструктажі з питань охорони праці поділяються на вступний, первинний, повторний, позаплановий та цільовий.

Вступний інструктаж: з питань охорони праці провадиться з усіма працівниками, які щойно прийняті на постійну чи тимчасову роботу, незалежно від їх освіти, стажу роботи за цією професією або посади; з працівниками, які перебувають у відрядженні на підприємстві і беруть безпосередню участь у виробничому процесі; з водіями транспортних засобів, які вперше в'їжджають на територію підприємства; з учнями, вихованцями та студентами, які прибули на підприємство для проходження виробничої практики; з учнями, вихованцями та студентами в навчально-виховних закладах перед початком трудового і професійного навчання в лабораторіях, майстернях, на полігонах тощо.

Вступний інструктаж проводить спеціаліст з охорони праці або особа, на яку наказом по підприємству покладено ці обов'язки, а з учнями в навчально-виховних закладах — особа, компетентна в питаннях охорони праці, на яку

покладено ці обов'язки. На великих підприємствах окремі питання вступного інструктажу можуть висвітлювати відповідні фахівці.

Інструктаж провадиться в кабінеті охорони праці або приміщенні, що спеціально для цього обладнано, з використанням сучасних технічних засобів навчання та наочних посібників за програмою, що розроблена службою охорони праці з урахуванням особливостей виробництва. Програма і тривалість інструктажу затверджуються керівником підприємства чи навчально-виховного закладу. Орієнтовний перелік для складання програми міститься в Типовому положенні.

Запис про проведення вступного інструктажу робиться в спеціальному журналі, а також у документі про прийняття працівника на роботу.

Первинний інструктаж провадиться на робочому місці до початку роботи з працівником, новоприйнятим на підприємство постійно чи тимчасово; з працівником, який переводиться з одного цеху виробництва до іншого; з працівником, який виконуватиме нову для нього роботу; з відрядженим працівником, який бере безпосередню участь у виробничому процесі на підприємстві; зі студентом, учнем чи вихованцем, який прибув на виробничу практику, перед виконанням ним нових видів робіт, перед вивченням кожної нової теми під час проведення трудового і професійного навчання в навчальних лабораторіях, класах, майстернях, на дільницях, під час проведення позашкільного навчання в гуртках та секціях тощо.

Первинний інструктаж провадиться індивідуально або з групою осіб спеціального фаху за програмою, складеною з урахуванням вимог відповідних актів про охорону праці, технічної документації і орієнтовного переліку питань первинного інструктажу.

Програма первинного інструктажу розробляється керівником цеху, дільниці, узгоджується із службою охорони праці і затверджується керівником підприємства, навчального закладу або їх відповідного структурного підрозділу.

Усі робітники, у тому числі випускники професійних навчальних закладів, після первинного інструктажу на робочому місці мають протягом 2-15 змін, залежно від характеру роботи та кваліфікації працівника, пройти стажування під керівництвом досвідчених, кваліфікованих робітників або спеціалістів, які призначаються наказом по підприємству. Керівник підприємства або структурного підрозділу має право звільнити від проходження стажування робітника, який має стаж роботи за своєю професією не менше трьох років, переміщується з одного цеху до іншого, де характер його роботи та тип обладнання, на якому він працюватиме, не змінюються.

Повторний інструктаж: провадиться на робочому місці з усіма працівниками: на роботах з підвищеною небезпекою — один раз у квартал, на інших роботах — один раз на півріччя. Він провадиться індивідуально або з групою працівників, які виконують однотипні роботи, за програмою первинного інструктажу в повному обсязі.

Позаплановий інструктаж провадиться з працівниками на робочому місці або в кабінеті охорони праці при введенні в дію нових або переглянутих нормативних актів про охорону праці, а також при внесенні змін та доповнень до них; при зміні технологічного процесу, зміні або модернізації устаткування, приладів та інструменту, вихідної сировини, матеріалів та інших чинників, що впливають на охорону праці; при порушенні працівником, студентом, учнем або вихованцем нормативних актів про охорону праці, що може призвести або призвело до травми, аварії чи отруєння; на вимогу працівників органу державного нагляду за охороною праці, вищої господарської організації або

державної виконавчої влади у випадку, якщо виявлено незнання працівником, студентом або учнем безпечних методів, прийомів праці чи нормативних актів про охорону праці; при перерві в роботі виконаних робіт більше як на 30 календарних днів — для робіт з підвищеною небезпекою, а для решти робіт — не більше 60 днів.

Позаплановий інструктаж провадиться індивідуально або з групою працівників спільного фаху. Обсяг і зміст інструктажу визначаються в кожному окремому випадку залежно від причин і обставин, що спричинили необхідність його проведення.

Цільовий інструктаж провадиться з працівниками при виконанні разових робіт, не пов'язаних з їх безпосередніми обов'язками за фахом (навантаження, розвантаження, разові роботи за межами підприємства тощо); ліквідації аварії, стихійного лиха; проведенні робіт, на які оформляються наряд-допуск, дозвіл та інші документи; екскурсіях на підприємства; організації масових заходів з учнями та вихованцями (екскурсії, походи, спортивні змагання тощо). Проведення інструктажу фіксується нарядом-допуском або іншою документацією, що дозволяє проведення робіт.

Первинний, повторний, позаплановий і цільовий інструктажі проводить безпосередньо керівник робіт. Інструктажі завершуються перевіркою знань усним опитуванням за допомогою технічних засобів навчання, а також перевіркою набутих навичок безпечних методів праці. Знання перевіряє особа, яка проводила інструктаж.

Виконуючи функції нагляду, зокрема за точним виконанням законодавства про охорону праці, прокурор має право безперешкодно входити у державні чи громадські установи, підприємства, міністерства, відомства; мати доступ до документів і матеріалів, необхідних для проведення перевірки; вимагати для

перевірки накази, розпорядження, інструкції та інші акти; одержувати інформацію про стан законності та заходи щодо її забезпечення тощо.

## ВИСНОВКИ

В кваліфікаційній роботі досліджено діяльність ПрАТ «Білоцерківський консервний завод» і розроблено АРМ для підтримки його діяльності. Основною метою розроблення АРМ є організація накопичення та зберігання інформації, її оперативна обробка в єдиній базі даних, забезпечення швидкого доступу до даних.

В процесі розробки здійснено системний аналіз діяльності головного технолога хлібозаводу і розроблено інформаційну систему у вигляді web-ресурсу, що задовільняє виконання поставленої задачі в роботі.

Функціональний аналіз дозволив виявити проблеми та задачі автоматизації, на основі яких було розроблено технічне завдання на проектування АРМ, в якому сформульовані основні вимоги до структури, функцій системи, а також функціонування комп'ютерної мережі.

Розробка АРМ реалізована, системою управління базою даних є MS SQL Server, засоби розробки web-сторінок, такі як HTML, JavaScript, CSS, PHP, jQuery та конструкторами для створення web-ресурсів (Зокрема, Winn).

Були досліджені питання, що стосуються web-орієнтованих інформаційних систем, були розглянуті основні способи проектування та створення web-ресурсу, основні підходи, переваги та недоліки створення web-орієнтованої інформаційної системи підприємства, орієнтуючись на засоби конструювання сайтів.

Було проведено доцільне обґрунтування та комплексна розробка web-сайту, з урахуванням сучасних світових тенденцій побудови організаційних та функціональних інформаційних структур підприємств.

В третьому розділі було проведене повне створення, розробка та впровадження всіх необхідних для реалізації технологій з їх подальшим налаштуванням. Під час створення web-сайту була виконана наступна робота:

Проведено аналіз наявних web-ресурсів зі схожою тематикою.

Проаналізовані першоджерела та визначені основні напрямки виконання роботи.

Проаналізована робота та використання засобів конструювання сайту.

Створені нові функції для альтернативних варіантів взаємодії користувача з відеоматеріалами.

Створено форму зворотного зв'язку для можливості онлайн замовлення користувачем представленого харчового продукту.

Обґрунтовується вибір програмних засобів реалізації системи, розроблено структуру сайту, описано частини системи і моменти її реалізації. Розроблено та реалізовано інформаційну систему.

Створена web-орієнтована інформаційна система буде презентувати новий харчовий продукт та виводити його на Інтернет ринок. Завдяки розробленим функціям та можливостям сайту, користувач зможе перейти на сайт збору коштів для подальшого інвестування, замовити продукт онлайн з допомогою форми зворотного зв'язку, повністю ознайомитись зі всіма етапами створення та розробки презентованого продукту та заохотить потенційного клієнта до популяризації даного ресурсу.

## БІБЛІОГРАФІЧНИЙ СПИСОК

1. Колисниченко Д.Н. PHP и SQL. Разработка Web-приложений. / Д.Н. Колисниченко. – СПб.: БХВ-Петербург, 2013. – 543 с.
2. Леки-Томпсон Э. PHP 5 для профессионалов / Э. Леки-Томпсон, А. Коув, С. Новицки, Х. Айде-Гудман. – М.: Диалектика, 2006. – 114 с.
3. Ленгсторф Дж. PHP и jQuery для профессионалов / Дж. Ленгсторф. — М.: «Вильямс», 2010. — 352 с.
4. Столлингс В. Передача данных/ В. Столлингс. - СПб.: Питер, 2004. – 750 с.
5. Інформаційні системи та їх види. [Електронний ресурс]. – Режим доступу: <http://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/95/95.html>
6. Types of Information Systems - Components and Classification of Information Systems [Електронний ресурс]. – Режим доступу: <https://www.managementstudyguide.com/types-of-information-systems.htm>
7. Муссель К.М. Предоставление и биллинг услуг связи. Системная интеграция. / К.М. Муссель.— М.: ИТЦ «Эко-Трендз», 2003.
8. Системи управління базами даних (СУБД), їх основні можливості та функції. [Електронний ресурс]. – Режим доступу: <https://vseosvita.ua/library/sistemi-upravlinna-bazami-danih-subd-ih-osnovni-mozlivosti-ta-funkcii-25461.html>
9. What is a Database Management System? - Purpose and Function [Електронний ресурс]. – Режим доступу: <https://study.com/academy/lesson/what-is-a-database-management-system-purpose-and-function.html>
10. Структура інформаційної системи. [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5064248/page:9/>
11. Components Of Information System. [Електронний ресурс]. – Режим доступу: <https://www.geeksforgeeks.org/components-of-information-system/>

- 12.Сучасні методи веб-програмування. [Електронний ресурс]. – Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1222.ukr.html>
- 13.Oracle MySQL Database Service. [Електронний ресурс]. – Режим доступу: <https://www.oracle.com/MySQL/>
- 14.Семенюк Р. А. ПЕРЕВАГИ ТА НЕДОЛІКИ ВИКОРИСТАННЯ СУБД MONGODB. [Електронний ресурс]. – Режим доступу: [http://eprints.zu.edu.ua/24821/1/Semenyuk\\_APSI2017.pdf](http://eprints.zu.edu.ua/24821/1/Semenyuk_APSI2017.pdf)
- 15.Introducing JSON. [Електронний ресурс]. – Режим доступу: <https://www.json.org/json-en.html>
- 16.What is JSON? [Електронний ресурс]. – Режим доступу: <https://www.hostinger.com/tutorials/what-is-json>
- 17.JSON. [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/uk/docs/Glossary/JSON>
- 18.Методи верстки сайтів. [Електронний ресурс]. – Режим доступу: <http://programer.org.ua/metodi-verstki-sajtiv/>
- 19.Розбираємо властивість float в CSS. [Електронний ресурс]. – Режим доступу:<http://programer.org.ua/rozbirayemo-vlastivist-float-v-css/>
- 20.Modular Grid: Complete Beginner’s Guide. [Електронний ресурс]. – Режим доступу: <https://icons8.com/articles/modular-grid-complete-beginners-guide/>
- 21.Уроки з Bootstrap v3. [Електронний ресурс]. – Режим доступу: <https://tokar.ua/read/6901>
- 22.Брайан Хоган. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – СПб.: Питер, 2012 – 272 с.: ил
- 23.Фельке-Моррис Т. Большая книга веб-дизайна/пер. с англ. Н.А.Райтмана. – М.: Эксмо, 2012 - 608с.

- 24.Макфарланд Д. Новая большая книга CSS. – СПб.: Питер, 2016 – 720 с.: ил. – (Серия “Бестселлеры O`Reilly”)
- 25.Шмитт К. HTML5.Рецепты прогаммирования/ К.Шмитт, К. Симпсон. – СПб.: Питер, 2018 -288 стр.
- 26.Бардзелл Джеффри Macromedia Dreamweaver MX 2004 с ASP, ColdFusion и PHP. Из первых рук (+ CD-ROM); Эком - М., 2016. - 560 с.
- 27.Бенкен Елена PHP, MySQL, XML. Программирование для Интернета; БХВ-Петербург - М., 2017. - 336 с.
- 28.Основы CSS. [Электронный ресурс]. – Режим доступа: [https://developer.mozilla.org/uk/docs/Learn/Getting\\_started\\_with\\_the\\_web/CS\\_S\\_basic](https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/CS_S_basic).
- 29.Гизберт Дамашке PHP и MySQL; ИТ Пресс - М., 2016. - 569 с.
- 30.Дронов В. PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов; БХВ-Петербург - М., 2016. - 480 с.
- 31.Дунаев В.В. HTML, скрипты и стили; БХВ-Петербург - М., 2017. - 527 с.
- 32.Жадаев Александр PHP для начинающих; Питер - М., 2016. - 768 с.
- 33.MySQL руководство администратора. - М.: Вильямс, 2005. - 621 с.
- 34.Аткинсон MySQL. Библиотека профессионала / Аткинсон, Леон. - М.: Вильямс, 2014. - 624 с.
- 35.Конверс PHP 5 и MySQL. Библия пользователя / Конверс, др. Т. и. - М.: Вильямс, 2006. - 426 с