

Київський національний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

**«Розробка сайту студентського самоврядування
університету»**

студента 4 курсу, 13 групи,
спеціальності
122 «Комп'ютерні науки»

підпис студента

Лопуга Владислав
Віталійович

Науковий керівник
доктор технічних наук, професор

підпис керівника

Краскевич Валерій
Євгенович

Гарант освітньої програми
кандидат технічних наук, доцент

підпис керівника

Демідов Павло
Георгійович

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра комп'ютерних наук та інформаційних систем

Спеціальність 122 «Комп'ютерні науки»

Затверджую

Зав. кафедри _____ Пурський О.І.
(підпис, дата)

Завдання

на випускний кваліфікаційний проект студенту

Лопузі Владиславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту.

«Розробка сайту студентського самоврядування університету»

Затверджена наказом ректора від *«15» грудня 2020 р. № 3780*

2. Строк здачі студентом закінченого проекту *29 травня 2021 року*

3. Цільова установка та вихідні дані до проекту

Мета проекту: розробити сайт, який конкретно і структурно ознайомлює аудиторію з певною організацією, її командою, цілями, відділами, ідеєю та ідеологією діяльності

Об'єкт дослідження: механізм функціонування сайту наукової та громадської організації.

Предмет дослідження: засоби розроблення, постійної підтримки і редакції сайтів на базі WordPress.

4. Перелік графічного матеріалу _____

5. Консультанти по проекту із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Краскевич В. Є.	15.12.2020 р.	15.12.2020 р.
2	Краскевич В. Є.	15.12.2020 р.	15.12.2020 р.
3	Краскевич В. Є..	15.12.2020 р.	15.12.2020 р.

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ІСНУЮЧОЇ БАЗИ З ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Структура студентського самоврядування

1.2. Задачі та функції

1.3. Постановка задачі

1.4. Системний аналіз засобів вирішення задачі

РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ СТУДЕНТСЬКОГО САМОВРЯДУВАННЯ

2.1. Поняття веб-сайта та їх класифікація

2.2. Етапи розробки веб-сайту

2.3. Огляд засобів розроблення сайтів на базі WordPress

РОЗДІЛ 3. РОЗРОБКА САЙТУ СТУДЕНТСЬКОГО САМОВРЯДУВАННЯ

3.1. Розробка макету сайту

3.2. Технічне та системне забезпечення розробки.

3.3. Розробка сайту

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання проекту

№ Пор.	Назва етапів кваліфікаційного проекту	Строк виконання етапів проект	
1	2	3	4
1	Вибір теми випускного кваліфікаційного проекту	01.10.2020	
2	Розробка та затвердження завдання на випускний кваліфікаційний проект	15.12.2020	
3	Вступ	03.02.2021	
4	Розділ 1. Системний Аналіз Існуючої Базы 3 Вирішення Поставленої Задачі	26.02.2021	
5	Розділ 2. Проектування Сайту Студентського Самоврядування	06.04.2021	
6	Розділ 3. Розробка Сайту Студентського Самоврядування	12.05.2021	
7	Висновки	15.05.2021	
8	Здача випускного кваліфікаційного проекту на кафедру науковому керівнику	20.05.2021	
9	Попередній захист випускного кваліфікаційного проекту	02.06.2021	
11	Виправлення зауважень, зовнішнє рецензування випускного кваліфікаційного проекту	07.06.2021	
12	Представлення готової зшитої випускного кваліфікаційного проекту на кафедру	10.06.2021	
13	Публічний захист випускного кваліфікаційного проекту	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2020 р.».

9. Керівник випускного кваліфікаційного проекту

Краскевич В. Є.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П. Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Лопуга В. В.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(nidnuc, dama)

Випускний кваліфікаційний проект студента Лопуга В. В.
(підпис, прізвище, ініціали)

Гарант освітньої програми _____ Демідов П. Г.
(підпис, прізвище, ініціали)

Завідувач кафедри _____ Пурський О. І.
(підпис, прізвище, ініціали)

5

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ІСНУЮЧОЇ БАЗИ З ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ	11
1.1. Структура студентського самоврядування	11
1.2. Задачі та функції	12
1.3. Постановка задачі	12
1.4. Системний аналіз засобів вирішення задачі	13
РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ СТУДЕНТСЬКОГО САМОВРЯДУВАННЯ.....	21
2.1. Поняття веб-сайта та їх класифікація	21
2.2. Етапи розробки веб-сайту	22
2.3. Огляд засобів структурування даних та їх використання в інформаційній системі	24
2.4. Методи табличного верстання інтернет-сайтів	28
2.5. Модульна система верстки у веб-дизайні	29
РОЗДІЛ 3. РОЗРОБКА САЙТУ СТУДЕНТСЬКОГО САМОВРЯДУВАННЯ	31
3.1. Розробка макету сайту	31
3.2. Технічне та системне забезпечення розробки	34
3.3. Розробка сайту	36
ВИСНОВКИ	45
БІБЛІОГРАФІЧНИЙ СПИСОК	46

АНОТАЦІЯ

Кваліфікаційна робота «Розробка сайту студентського самоврядування університету» розроблена Лопугою В. В., складається з 48 сторінок, 11 рисунків, 35 літературних джерел.

У кваліфікаційній роботі наведені характеристика функціонування web-орієнтованих інформаційних систем та запропоновано створення сайту для студентської організації, що займається громадською та науковою діяльністю.

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 візуалізації 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 – відобразити характер роботи нашої організації.

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

ABSTRACT

Qualification work "Development of the site of student self-government of the university" is developed by Lopuga V. V., consists of 48 pages, 11 figures, 35 literary sources.

The qualification work presents the characteristics of the functioning of web-oriented information systems and proposes the creation of a site for a student organization engaged in public and scientific activities.

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 - to reflect the nature of our organization.

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

ВСТУП

Актуальність дослідження полягає в тому, завжди неабияк важливим у діяльності організації, підприємства, компанії та будь-якої іншої установи буде вміння правильно презентувати ці самі організації і зробити це якомога ефективніше – при менших затратах охопити якомога більше аудиторії.

Саме тому важливим лишаються дослідження і аналітика питань стосовно ефективності вибору засобі презентації організації та яке місце у цій аналітиці буде залишене головному предмету дослідження цієї теми – засобам створення web-сайтів.

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: розробити сайт, який конкретно і структурно ознайомлює аудиторію з певною організацією, її командою, цілями, відділами, ідеєю та ідеологією діяльності.

Об'єкт дослідження: механізм функціонування сайту наукової та громадської організації.

Предмет дослідження: засоби розроблення, постійної підтримки і редакції сайтів на базі WordPress.

Методи дослідження: аналіз методів створення web-ресурсів, використання принципів інформаційного моделювання, застосування системного підходу та структурно-функціонального методу.

Основні завдання дослідження:

1. Аналіз роботи та важливих напрямів у діяльності студентського самоврядування.
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 ІСНУЮЧОЇ БАЗИ З ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Структура студентського самоврядування

Студентське самоврядування – це можливість самостійно і незалежно займатись громадською діяльністю, в межах юридично сили конкретного вишу, при цьому реалізуючи певні управлінські функції цього самого вишу, які мають відповідати меті та ідеї діяльності органу студентського самоврядування.

Для того, аби сила з якою здатне функціонувати самоврядування студентів а можливості для молоді, які воно відкриває та права, які захищає, були реальними, то необхідно сформувати орган, який би протокольно слідував и своє маніфесту, плану дій та моральному закону і в межах своєї юридичної сили реалізовував би свої повноваження.

В нашому університеті – Київському національному торговельно-економічному університеті (далі – КНТЕУ), цей орган зветься Рада Студентського Самоврядування (далі – РСС) і процес функціонування та виконання обов’язків РСС відбувається з врахуванням наступних чинників:

- Положення про діяльність РСС та Положення про студентське самоврядування;
- План заходів;
- Моральний кодекс;
- Організаційної структури РСС.

Варто відміти, що неабияк важливою складовою діяльності РСС є момент, де поєднується як громадська активність (як у представників студентського самоврядування), так і наукова діяльність (як у студентів наукової установи) і саме такою частиною в РСС є Наукове Товариство КНТЕУ.

До його складу входять відділу з організації діяльності Товариства: науковий, організаційних, редакційно-видавничий, робота з молодими вченими, втілення інновацій тощо; наукові організації: наукові клуби, що сформовані на

кафедрах по напрямку виховання кафедрою студентів, організовані студенти наукові організації, об'єднані ідеєю.

1.2. Задачі та функції

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.

1.3. Постановка задачі

З розвитком технологій варто відмітити і той факт, що якими б складними вони не здавались, у використанні та застосуванні технології стають тільки простішими. Так з'явилися конструктори сайтів, які дозволяють людям, що не володіють мовами програмування, проте мають бажання і відчуття прекрасного та того, яким повинен бути сайт, відчувати і навіть стати творцем сайту.

Таким чином в світі web-дизайну з'явився термін Content Management System (CMS) – програмне забезпечення для створення і організації web-сайтів або інших інформаційних ресурсів в інтернеті чи іншій комп'ютерній мережі.

Ефективних, функціональних CMS, які якась компанія чи організація може обрати для реалізації свого сайту з відтворенням якихось навіть унікальних ідей та функцій, існує десятки, а то і сотні. Одну і передових в контексті як і популярності, так і функціоналу позицій серед CMS займає WordPress, адже займає близько 35% усіх існуючих сайтів в інтернеті і певно що не спроста.

Дана CMS має ряд переваг: вона використовує PHP та MySQL, які підтримуються фактично усіма хостинг-провайдерами і при цьому тарифні плани хостингу саме для WordPress можуть мати кращі показники в порівнянні з іншими: покращені показники швидкості, продуктивності та надійності сайту.

Крім того, платформа WordPress досить універсальна, що дозволяє використовувати її для різних цілей: це і створення інтернет-магазину, і ведення блогу, публікація новин тощо.

Варто відмітити, що інтерфейс для роботи із CMS WordPress є інтуїтивно зрозумілим, що послуговує черговою причиною для вибору його як засобу реалізації завдання.

Найголовніше – платформа має відкритий код і є безкоштовною абсолютно для всіх.

Варто розуміти різницю між Wordpress.org і WordPress.com, де саме перший ресурс і є за собою розробки та публікації під доменом сайту, а друга – хостинговою компанією, яка використовує систему управління контентом на всі сайти, розміщені у них.

1.4. Системний аналіз засобів вирішення задачі

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 перехоплене і оброблене 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 (eXtensible 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.

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 реєстр 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 трирівневих 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 трирівневих 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 трирівневих 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++.

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 коі8-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 koі8, 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).

РОЗДІЛ 2. ПРОЕКТУВАННЯ САЙТУ СТУДЕНТСЬКОГО САМОВРЯУВАННЯ

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 браузером. 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 відображатиметься 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-сайту

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;

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



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]

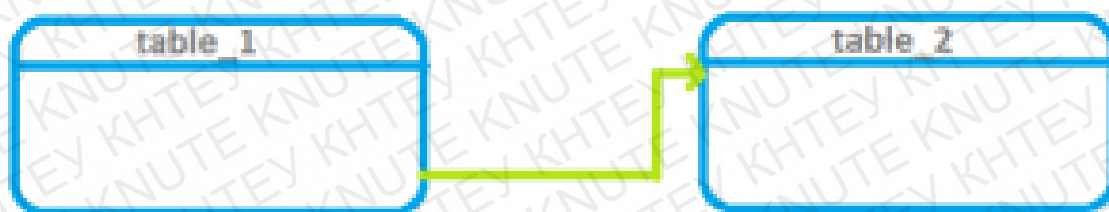


Рис.2.2. Вигляд реляційної бази даних.

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.

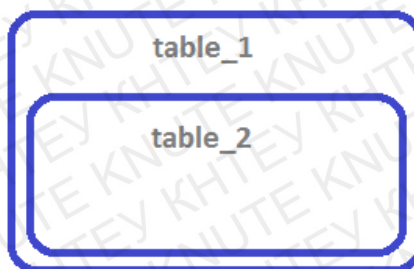


Рис.2.3. Вигляд нереляційної бази даних.

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 мережевих 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).

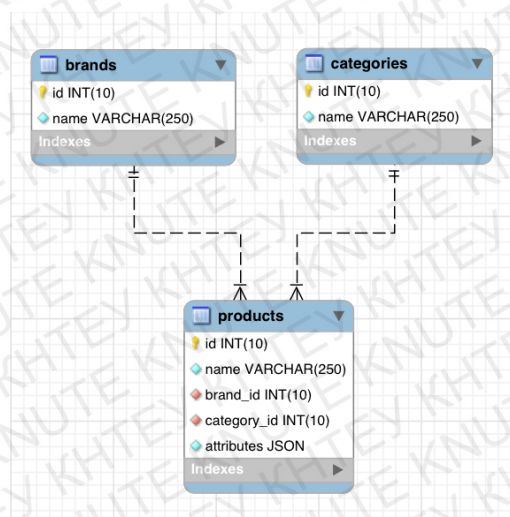


Рис.2.4. Приклад структури БД MySQL

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.

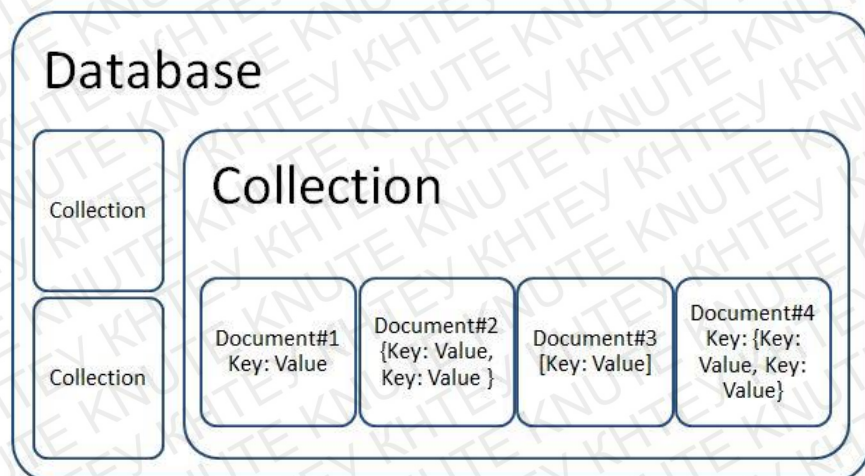


Рис.2.5. Вигляд структури БД MongoDB

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. Модульна система верстки у веб-дизайні

Властивість float в CSS, дозволяє розробникам 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.

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.

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.

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.

РОЗДІЛ 3. РОЗРОБКА САЙТУ СТУДЕНТСЬКОГО САМОВРЯДУВАННЯ

3.1. Розробка макету сайту

Перед тим як створювати web-орієнтовану систему студентського самоврядування, 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 розробляться 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.

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, компоувати 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.

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.

3.2. Технічне та системне забезпечення розробки.

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 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.



Рис. 3.1. — Форма авторизації

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. Розробка сайту

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 коді 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.

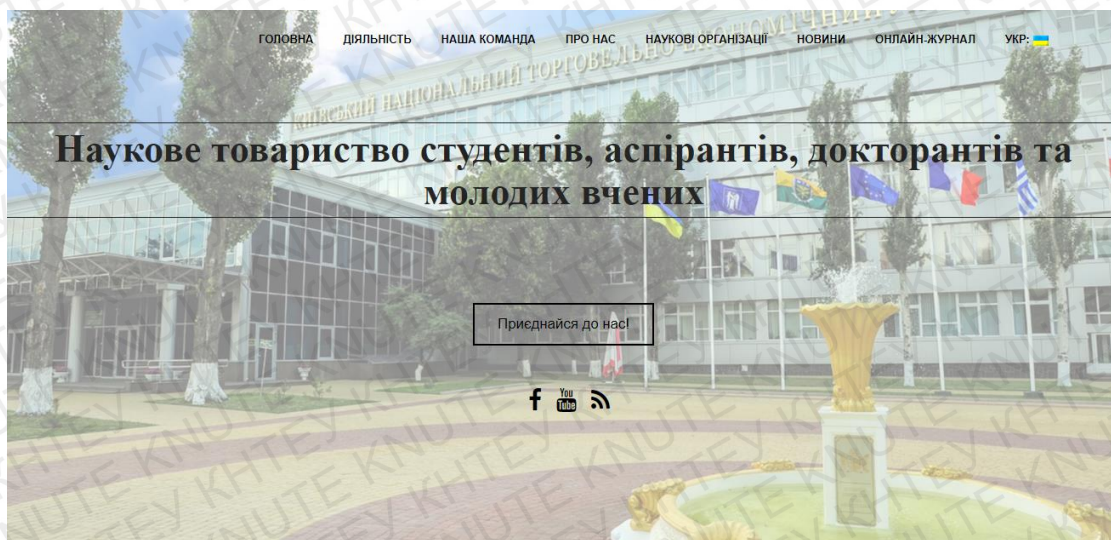


Рис. 3.2. Розмітка першого екрану сайту

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.

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 програвач 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.

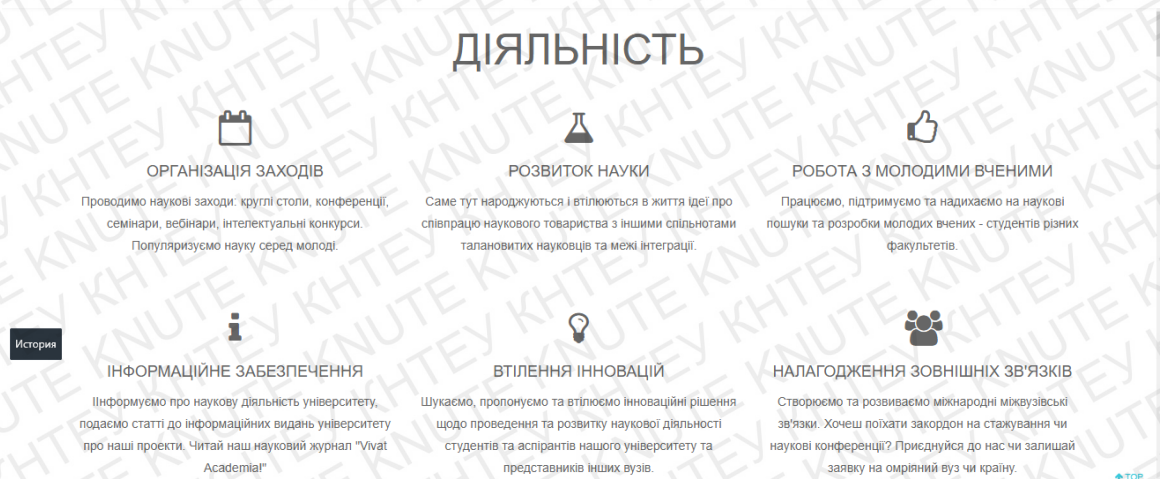


Рис. 3.3. Наповнення сторінки вмістом

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, 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. 3.4. Використання Шорткоду

Завершивши верстку сайту, тобто створення візуальної частини сайту або *front-end*, можна переходити до створення технічної частини розробки. Для використання API-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 PHP.

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 елементу серед всіх елементів сайту. Після знаходження відповідного елементу, 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.

ВИСНОВКИ

Питання презентації організації для популяризації її діяльності є неабияк актуально зараз, а засоби для створення сайту, маючи свій широкий потенціал, з цим завданням справляються краще всього. Варто відмітити тут і засоби для конструювання сайту, які дозволяють розширити кількість людей, що здатні володіти процесом створення сайту, завдяки інтуїтивно зрозумілому інтерфейсу.

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.

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 організацію та основні напрями її діяльності. Завдяки розробленим функціям та можливостям сайту, користувач зможе перейти на сайт збору коштів для подальшого інвестування.

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
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 программирования/ 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.
29. Гизберт Дамашке PHP и MySQL; HT 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.