МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ



**Звіт**

До розрахунково-графічної роботи

З дисципліни:

«Веб-технології та веб-дизайн»

Варіант №25

*Виконав:*

*студент групи КН-47*

Шандра Олег

*Прийняв:*

*доцент* Василюк А. С.

Львів 2018

**ЗМІСТ**

1. Завдання …………………………………………………………………………..3
2. Вступ……………………………………………………………………………....4
3. Розділ 1. Теоретична частина…………………………………………………....5
4. Розділ 2. Практична частина…………………………………………………….10
5. Висновки………………………………………………………………………….14
6. Список використаної літератури………………………………………………..15

**ЗАВДАННЯ**

В ході виконання розрахунково-графічної роботи потрібно в першому розділі викласти теоретичні положення про методи моніторингу WWW. В другому розділі за допомогою мови гіпертекстової розмітки HTML написати ієрархічний веб-сайт з трьох сторінок на довільну тему. Також в цьому розділі потрібно детально описати процес створення цього сайту, включно з скріншотами. Код сторінки потрібно додати до роботи.

**ВСТУП**

*Актуальність теми*. З огляду на те, що зараз активно розвивається напрямок веб-розробки, з кожним роком все більше нових веб-орієнтованих проектів відкривається. WWW на даний час найбільше всесвітнє багатомовне сховище інформації в електронному вигляді: десятки мільйонів пов'язаних між собою документів, що розташовані на комп'ютерах, розміщених на всій земній кулі. Вважається найпопулярнішою і найцікавішою службою мережі Інтернет, яка дозволяє отримувати доступ до інформації незалежно від місця її розташування.

*Мета і задачі роботи.* Метою роботи є надання коротких, проте повних теоретичних відомостей про методи моніторингу WWW написання власної веб-орієнтованої аплікації. Для цього потрібно виконати наступні задачі:

* Проаналізувати літературні джерела та віднайти корисну та актуальну інформацію про методи моніторингу WWW;
* Систематизувати теоретичні знання про методи моніторингу WWW та викласти основні факти про цю мову та її особливості;
* Вибрати набір технологій для написання аплікації.
* Вибрати тематику для написання аплікації;
* Провести етап написання коду;
* Провести етап тестування аплікації.

**РОЗДІЛ 1. ТЕОРЕТИЧНА ЧАСТИНА**

Всесві́тня мережа (англ. World Wide Web, скорочено: WWW; також: всемережжя, веб або тене́та) — найбільше всесвітнє багатомовне сховище інформації в електронному вигляді: десятки мільйонів пов'язаних між собою документів, що розташовані на комп'ютерах, розміщених на всій земній кулі. Вважається найпопулярнішою і найцікавішою службою мережі Інтернет, яка дозволяє отримувати доступ до інформації незалежно від місця її розташування.

WWW — інформаційна система, якій не можна дати конкретного визначення. Наведемо лише деякі з епітетів, якими вона може бути позначена: гіпертекстова, гіпермедійна, розподілена, інтегруюча, глобальна. Нижче буде показано, що слід розуміти під кожною з цих властивостей у контексті WWW.

Користувачі автоматично переходять від однієї бази даних (сайту) до іншої за допомогою гіперпосилань.

Кількість серверів WWW постійно зростає, а швидкість росту WWW навіть більша ніж у самої мережі Internet. WWW  — найрозвиненіша технологія Internet, вона вже стала масовою.

WWW працює за принципом клієнт-сервер: існує велика кількість серверів, які за запитом клієнта надають йому гіпермедійний документ. Такий документ складається із частин з різним представленням інформації (текст, графіка, звук, відео, тривимірні об'єкти тощо). В ньому кожен елемент може бути посиланням на інший документ чи його частину. Такі посилання в WWW організовані так, що кожний інформаційний ресурс в глобальній мережі Internet однозначно адресується, а надісланий сервером документ може посилатися на інші документи на цьому ж сервері, чи на документи на інших комп'ютерах Internet. При цьому користувач не помічає цього і працює з усім інформаційним простором Internet, як з єдиним цілим. Посилання WWW вказують не тільки на документи, специфічні для самої WWW, але й на інші сервіси і інформаційні ресурси Internet. Більш того, більшість програм клієнтів WWW (браузер) не просто розуміють такі посилання, а є додатково програмами-клієнтами відповідних сервісів: FTP, Gopher, новин мережі Usenet, електронної пошти і т. і. Таким чином, програмні засоби WWW — універсальні для різних сервісів Internet, а сама інформаційна система WWW грає інтегруючу роль.

Наведемо роз'яснення деяких термінів, які використовуються в WWW.

HTML (hyper text markup language, мова розмітки гіпертексту). Це формат гіпермедійних документів, які використовують в WWW для представлення інформації. Цей формат описує вміст документу, його структуру, а також його зв'язки з іншими документами. Зовнішній вигляд документа на екрані користувача визначається навігатором — якщо користувач працює за графічним або текстовим терміналом, у кожному випадку документ на екрані матиме різний вигляд. HTML виконує інтегруючу роль для елементів гіпермедійного документа. Імена файлів у форматі html, як правило, закінчуються на html (або мають розширення htm у випадку, якщо сервер працює під Windows).

URL (uniform resource locator, універсальний вказівник на ресурс). Таку назву носять словесні посилання на будь-які інформаційні ресурси Internet.

До ресурсів Internet можна отримувати доступ і за IP-адресою певного комп'ютера.

HTTP (hypertext transfer protocol, протокол передачі гіпертексту). Таку назву носить протокол, за яким взаємодіють клієнт та сервер WWW для передавання гіпермедійного документа клієнту.

WWW — сервіс прямого доступу, який потребує повноцінного підімкнення до Internet. Він вимагає швидких ліній зв'язку для документів, що містять багато графічної або іншої нетекстової інформації. Коли швидкості нижчі, втрачається частина переваг, які зробили WWW таким популярним.

На цьому комп'ютері Тімом Бернерсом-Лі було запущено першу програму веб-сервера, створену ним же

Розробниками технології всесвітньої павутини вважаються Тім Бернерс-Лі і Роберт Кайо[1]. Тім Бернерс-Лі є автором технологій HTTP, URI/URL та HTML. У 1980—1981 роках він працював у Європейській організації по ядернх дослідженнях (фр. conseil européen pour la recherche nucléaire, CERN) консультантом з програмного забезпечення. Саме там він для власних потреб написав програму «Енквайр» (англ. «Enquire», можна вільно перекласти як «Дізнавач»), яка використовувала випадкові асоціації для зберігання даних та заклала концептуальну основу для Всесвітньої павутини.

У 1984 році він отримав стипендію в CERN і зайнявся там розробкою розподілених систем для збору наукових даних. У цей час він працював над системою «FASTBUS» і розробив свою систему RPC (англ. Remote Procedure Call — віддалений виклик процедури).

У 1989 році, працюючи в CERN над внутрішньою системою обміну документів ENQUIRE, Бернерс-Лі запропонував глобальний гіпертекстовий проект, нині відомий як Всесвітня павутина. Аналогічна пропозиція незалежно від Тіма Бернерса-Лі була висунута у тому ж році іншим працівником CERN Робертом Кайо У 1990 році це привело до спільної пропозиції цієї технології. Проект було затверджено і реалізовано. Проект мав на увазі публікацію гіпертекстових документів, зв'язаних між собою гіперпосиланнями, що полегшило б пошук і консолідацію інформації для вчених CERN.

З 1991 по 1993 рік Тім Бернерс-Лі продовжував роботу над Всесвітньою павутиною. Він збирав відгуки від користувачів і координував роботу Павутини. Тоді він разом з помічниками вперше запропонував для широкого обговорення свої перші специфікації: ідентифікатори URI, протокол HTTP і мову HTML, технології, без яких уже неможливо уявити сучасний Інтернет. Всё ж, офіційно роком народження Всесвітньої павутини вважається 1989 рік.

У рамках проекту Бернерс-Лі написав перший у світі веб-сервер, що мав назву «httpd», і перший у світі гіпертекстовий веб-браузер, що мав назву «WorldWideWeb». Цей браузер був одночасно і WYSIWYG-редактором (скорочено від англ. what you see is what you get — що бачиш, те й отримаєш), його розроблення було розпочате у жовтні 1990 року, а завершене у грудні того ж року. Програма працювала у середовищі NeXTStep й почала поширюватись Інтернетом влітку 1991 року.

Роберт Кайлліау (англ. Robert Cailliau) досить детально описує як розгортались події у CERN, пов'язані із створенням Web[2]:

|  |  |  |
| --- | --- | --- |
| « | Майк Сендалл (Mike Sendall) купує у той час комп’ютер «NeXT cube» для того, щоб зрозуміти, у чому полягають особливості його архітектури, та віддає його згодом Тіму Бернерсу-Лі. Завдяки досконалості програмної системи «NeXT cube» Тім написав прототип, що ілюстрував основні положення проекту, за декілька місяців. Це був вражаючий результат: прототип пропонував користувачам, крім іншого, такі розвинені можливості, як WYSIWYG browsing/authoring!… Під час однієї із сесій спільних обговорень проекту в кафетерії ЦЕРНа ми з Тімом спробували підібрати «зачіпляючу» назву (англ. catching name) для системи яка створювалась. Єдине, на чому я наполягав, це щоб назва не була черговий раз видобута все з тієї ж грецької міфології. Тім запропонував «world wide web». Все у цій назві мені зразу сподобалось, лише важко вимовляється французькою | » |

Перший у світі веб-сайт був розміщений Бернерсом-Лі 6 серпня 1991 року на першому веб-сервері, що був доступний за адресою http://info.cern.ch/[3]. Ресурс визначав поняття «Всесвітньої павутини», містив інструкції із встановлення веб-сервера, використання браузера тощо. Цей сайт також був першим у світі інтернет-каталогом, тому що згодом Тім Бернерс-Лі розмістив і підтримував там список посилань на інші сайти.

На першому фото, опублікованому у Всесвітній павутині, був зображений пародійний філк-гурт «Les Horribles Cernettes» (у перекладі з французької — «Страшненькі дівчатка з CERN»)[4]. Тім Бернес-Лі попросив у лідера гурту відскановані світлини після музичного фестивалю «CERN hardronic festival».

І все ж теоретичні основи вебу були закладені значно раніше від Бернерса-Лі. Ще у 1945 році Веннівер Буш розробив концепцію Memex (поєднання англ. memory та англ.index)[5], гіпотетичного прототипу гіпертекстової системи, допоміжних механічних засобів «розширення людської пам'яті]». Memex — це пристрій, у якому людина зберігає усі свої книги й записи (а в ідеалі — і усі свої знання, що піддаються формальному опису) і, який видає потрібну інформацію з достатньою швидкістю та гнучкістю. Пристрій є розширенням і доповненням до пам'яті людини. Бушем було також передбачене всеосяжне індексування текстів та мультимедійних ресурсів з можливістю швидкого пошуку потрібної інформації. Наступним значним кроком на шляху до Всесвітньої павутини було створення гіпертексту (поняття уведене Тедом Нельсоном у 1965 році[6].

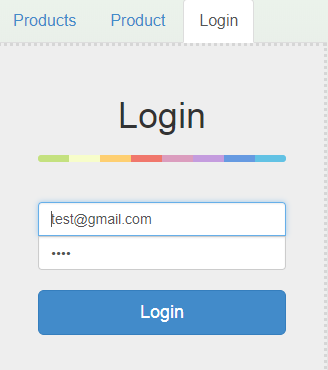
З 1994 року основну роботу з розвитку Всесвітньої павутини взяв на себе консорціум Всесвітньої павутини (англ. world wide web consortium, у скороченому запису W3C), заснований і очолюваний Тімом Бернерсом-Лі. Цей консорціум — організація, що розробляє і впроваджує технологічні стандарти для Інтернету і Всесвітньої павутини. Місія W3C: «Повністю розкрити потенціал Всесвітньої павутини шляхом створення протоколів і принципів, що гарантують довготривалий розвиток Мережі». Два інші завдання консорціуму — забезпечити повну «інтернаціоналізацію Мережі» і зробити Мережу доступною для людей з обмеженими можливостями.

W3C розробляє для Інтернету єдині принципи і стандарти (носять назву «рекомендації», англ. W3C recommendations), котрі у подальшому запроваджуються виробниками програм і обладнання. Так досягається сумісність між програмними продуктами та апаратурою різних компаній, що робить Всесвітню мережу досконалішою, універсальнішою і зручнішою. Усі рекомендації консорціуму Всесвітньої павутини є відкритими, тобто є не захищеними патентами і можуть впроваджуватись будь-ким без усяких фінансових зобов'язань перед консорціумом.

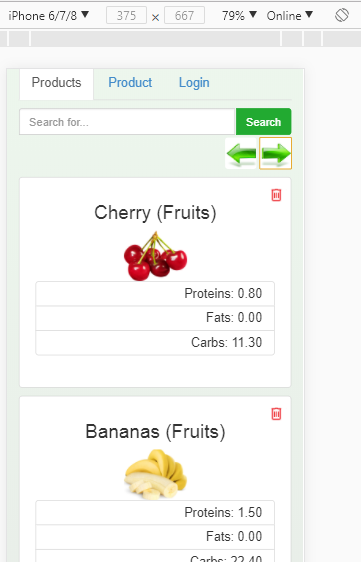
**РОЗДІЛ 2. ПРАКТИЧНА ЧАСТИНА**

У даній практичні частині було створено сайт «Книга рецептів» що має інформацію про продукти та рецепти користувача. Сайт написаний на технологіях HTML, CSS, JavaScript, JQuery, Bootstrap. Сайт створено

Першою сторінкою що бачить користува це сторінка логування в систему де користувач повинен ввести логін та пароль щоб мати доступ до системи (рис. 1).

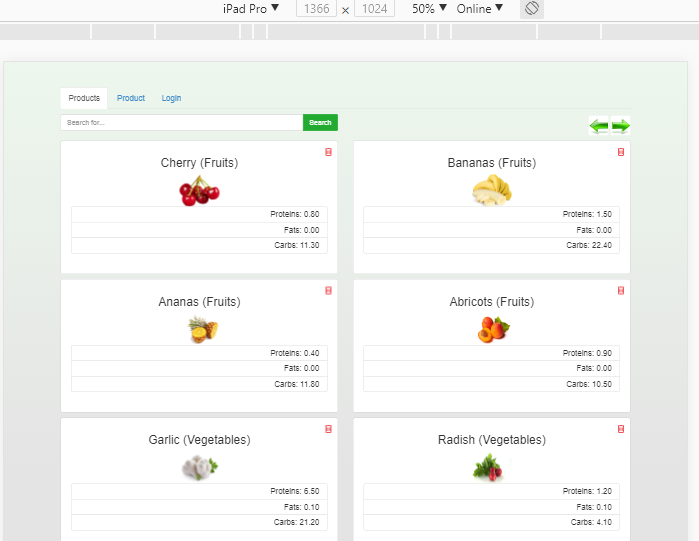


*Рис. 1. Форма логування.*

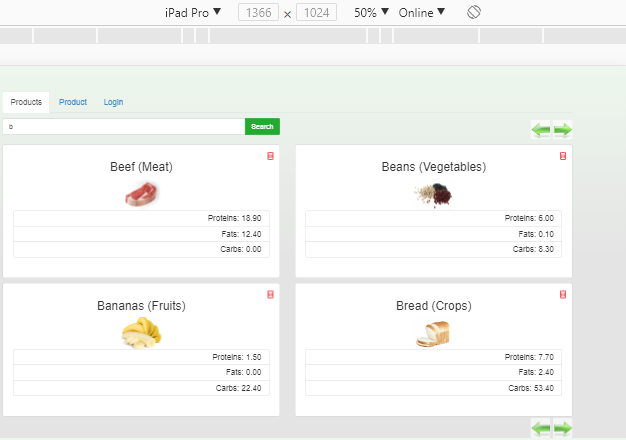


*Рис. 2.Вогляд на iphone*

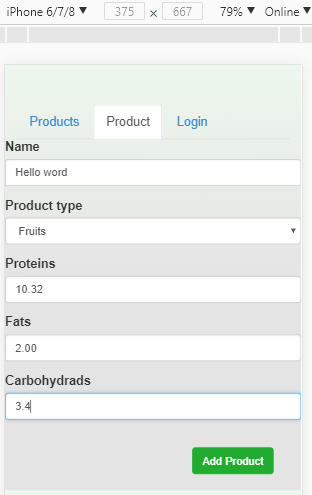
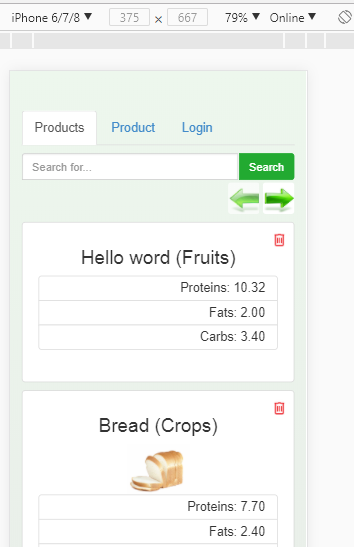
Наступною є сторінкою є сторінка зі списком продуктів (рис. 2-4). Також користувач має змогу додати новий продукт (рис. 5).



*Рис. 3.Вогляд на ipad pro*



*Рис. 4.Пошук продуктів за буквою’b’*

*Рис. 5. Додавання нового продукту.*

**ВИСНОВКИ**

Виконання даної розрахунково-графічної роботи дало змогу ознайомитися з методами моніторингу WWW та розробити власний сайт з переходами по різних сторінках.

Впродовж практики було здійснено ознайомлення з різними видами прикладного програмного забезпечення, вдосконалено свої навички.

Окрім загальних завдань роботи було виконано індивідуальне завдання та набуто практичні навички роботи у написанні сайтів.

Під час виконання роботи підібрано необхідну інформацію для придбання навичок практичної діяльності у розробці веб-сайтів.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Документація з використання ASP.NET Web API 2 (C#) [Електронний ресурс]. Режим доступу URL: https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api.
2. Офіційна документація Bootstrap. [Електронний ресурс]. Режим доступу URL: https://getbootstrap.com/.
3. Форум програмістів Електронний ресурс]. Режим доступу URL: https://stackoverflow.com/.
4. ГОСТ 34.201-90. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы**.** Стадии создания. – М.: Изд-во стандартов, 1991. – 16 с.
5. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – К.: Держстандарт України, 1995. – 36 с.
6. Сторінка в ресурсі Вікіпедія. [Електронний ресурс]. Режим доступу URL: https://uk.wikipedia.org/wiki/PHP

Додаток А

function bindProductPageControls() {

$("." + ProductListClasses.PagingLeft).unbind('click', goToProductsLeftPage).click(goToProductsLeftPage);

$("." + ProductListClasses.PagingRight).unbind('click', goToProductsRightPage).click(goToProductsRightPage);

$("#" + ProductListIds.BtnSearchByName).unbind('click', searchByName).click(searchByName);

}

function deleteProduct(event) {

var productId = $(event.target).parents().eq(1).find("." + ProductListClasses.ProductId).text();

$.post(RequestURLs.DeleteProduct, { Id: productId }, function () {

loadProducts();

});

}

function initProductSearchControls(currentPageProductsCount) {

$("." + ProductListClasses.PagingLeft).prop("disabled", searchedProductName !== "" || currentPageNumber <= 1);

$("." + ProductListClasses.PagingRight).prop("disabled", searchedProductName !== "" || currentPageProductsCount === 0);

}

function searchByName() {

searchedProductName = $("#" + ProductListIds.SearchByName).val();

loadProducts();

}

function goToProductsLeftPage() {

currentPageNumber--;

loadProducts();

}

function goToProductsRightPage() {

currentPageNumber++;

loadProducts();

}

function loadProducts() {

var layout = $("." + ProductListClasses.CardWitProductItemLayout).eq(0).clone();

var productsListTab = $("#" + ProductListClasses.ProductsCards);

var productsListTabClone = productsListTab.clone();

productsListTabClone.empty();

var callback = function (products) {

var i = 1;

var previousCard = null;

initProductSearchControls(products.length);

$.each(products, function (index, value) {

var currentCard = layout.clone();

currentCard.removeClass(ProductListClasses.CardWitProductItemLayout).show();

productsListTabClone.append(currentCard);

currentCard.find("." + ProductListClasses.ProductId).text(value.Id);

currentCard.find("." + ProductListClasses.ProductName).text(value.Name);

currentCard.find("." + ProductListClasses.ProductType).text(value.ProductTypeModel.Name);

currentCard.find("." + ProductListClasses.ProductImage).attr('src', value.SmallPhotoLink);

currentCard.find("." + ProductListClasses.ProductProteins).text(value.Proteins.toFixed(2));

currentCard.find("." + ProductListClasses.ProductFats).text(value.Fats.toFixed(2));

currentCard.find("." + ProductListClasses.ProductCarbs).text(value.Carbohydrates.toFixed(2));

if (i % 2 === 0) {

var tempRow = $("<div class='row'></div>");

tempRow.append(previousCard);

tempRow.append(currentCard);

productsListTabClone.append(tempRow);

}

previousCard = currentCard;

i++;

});

productsListTab.empty().replaceWith(productsListTabClone);

$("." + ProductListClasses.DeleteProductBtn).click(deleteProduct);

}

if (searchedProductName !== "") {

$.get(RequestURLs.SearchProductByNameUrl, { name: searchedProductName }, function (products) {

callback(products);

});

}

else {

var param = {

count: 10,

pageNumber: currentPageNumber,

sortColumnName: null,

sortOrder: null,

filterProductTypeId: null

};

$.get(RequestURLs.GetProductsUrl, param, function (products) {

callback(products);

});

}

}

<form id="new-product-form">

<div class="form-group row">

<label for="product-name" class="col-2 col-form-label">Name</label>

<div class="col-10">

<input class="form-control" name="Name" type="text" value="" id="product-name">

</div>

</div>

<div class="form-group row">

<label for="product-type" class="col-2 col-form-label">Product type</label>

<div class="col-10">

<select class="form-control" name="ProductTypeModel.Id" id="product-type"></select>

</div>

</div>

<div class="form-group row">

<label for="product-proteins" class="col-2 col-form-label">Proteins</label>

<div class="col-10">

<input class="form-control" name="Proteins" type="number" value="0" min="0" max="100" id="product-proteins">

</div>

</div>

<div class="form-group row">

<label for="product-fats" class="col-2 col-form-label">Fats</label>

<div class="col-10">

<input class="form-control" name="Fats" type="number" value="0" min="0" max="100" id="product-fats">

</div>

</div>

<div class="form-group row">

<label for="product-carbohydrads" class="col-2 col-form-label">Carbohydrads</label>

<div class="col-10">

<input class="form-control" name="Carbohydrates" type="number" value="0" min="0" max="100" id="product-carbohydrads">

</div>

</div>

<div class="modal-footer">

<button class="btn btn-info" id="submit-new-product">Add Product</button>

</div>

</form>

function bindNewProductPageControls() {

$("#" + ProductFormIds.SubmitNewProduct).unbind("click")

.click(function () {

event.preventDefault();

var form = $("#" + ProductFormIds.NewProductForm);

$.post(RequestURLs.AddProduct, form.serialize(), function () {

$("#" + ProductFormIds.ProductsTabSelector).click();

});

});

}

function loadProductTypes() {

$.get(RequestURLs.GetAllProductTypesUrl, { }, function (productTypes) {

$.each(productTypes, function (i, type) {

$("#" + ProductFormIds.ProductType).append($('<option>', {

value: type.Id,

text: type.Name

}));

});

});