

ЗМІСТ

Скорочення та умовні позначки	5
1 Аналіз предметної області.....	7
1.1 Опис напрямку діяльності компанії	8
1.2 Вимоги до дизайну сайту	9
1.3 Вимоги до функціональності сайту	10
2 Обґрунтування вибору інструментальних засобів розробки	11
2.1 Мова розмітки гіпертекстових документів HTML5	11
2.2 Формальна мова опису зовнішнього вигляду документу CSS	13
2.3 Прототипно-орієнтована сценарна мова програмування	15
JavaScript.....	15
2.4 Бібліотека jQuery	17
2.5 Скриптова мова програмування PHP.....	18
2.6 Мова структурованих запитів SQL.....	19
2.7 Система керування базами даних MySQL	21
2.8 Підхід до створення web-застосунків AJAX	23
2.9 Фреймворк для створення веб-інтерфейсів Bootstrap.....	25
3 Опис етапів проектування сайту.....	28
3.1 Вибір системи керування контентом Webasyst	29
3.2 Проектування бази даних	30
3.3 Конфігураційні файли Webasyst	31
4 Дизайн інтернет-магазину	33
4.1 Проектування дизайну сайту	33
4.2 Реалізація дизайну сайту.....	35
5 Розробка додатку Shop	40
Висновки	48
Перелік джерел посилання	49
Додаток А Файлова структура Webasyst	51

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД	– база даних
ІС	– інформаційна система
СКБД	– система керування базами даних
ПЗ	– програмне забезпечення;
API	– Application Programming Interface (інтерфейс прикладного програмування)
CMS	– Content management system (Система управління контентом)
CSS	– Cascading Style Sheets (Каскадні таблиці стилів)
HTML	– HyperText Markup Language (Мова розмітки гіпертекстових документів)
MVC	– Model-View-Controller (Модель-вид-контролер)
SQL	– Structured Query Language (Мова структурованих запитів)
WYSIWYG	– What You See Is What You Get (Що бачиш, то і отримаєш)
UCS	– Universal Character Set (Універсальний набір символів)
UML	– Unified Modeling Language (Універсальна мова моделювання)
URL	– Uniform Resource Locator (Єдиний вказівник ресурсу)

ВСТУП

Купівля та продаж речей через інтернет-магазин вже не є чимось особливим, сьогодні це нормально для більшої половини жителів України. Згідно з публічною базою даних Google Adwords, від 01 травня 2019 року, в Україні понад 24 млн. 700 тисяч активних інтернет користувачів.

Можливість купувати і продавати через інтернет стало можливим для людства завдяки 2 речам: швидкому інтернету і сайтам-каталогам, або як їх прийнято називати, інтернет-магазини.

Коротко інтернет-магазин можна визначити як програмне забезпечення, спеціально розроблене для зручності покупок і продажів з веб-сайту. Власне, інтернет-магазин і є сайтом, тільки з особливою «начинкою» – набором програм, спеціально орієнтованих на електронну комерцію. Інтернет-магазин має ряд очевидних переваг у порівнянні зі звичайною точкою продажів.

Метою даної роботи є створення інформаційно-пошукової системи підприємства по продажу спортивних товарів Roliki, представленої у вигляді Інтернет-магазину на базі CMS Webasyst.

Така система надасть компанії Roliki нові можливості. Співробітникам компанії:

- розміщувати нові і приховувати старі товари, які вже недоступні для продажу;
- імпортувати і експортувати товари в форматах Microsoft Excel;
- опубліковувати маркетингові акції і спеціальні пропозиції;
- отримувати коментарі від клієнтів;
- вести складський облік;
- проводити аналітику продажів.

Клієнтам компанії:

- дізнатись інформацію про товар: прочитати опис, подивитися фото і відео, порівняти характеристики даного типу товару, зокрема конкретних позицій;
- побачити товар, що є у наявності, ціну, способи оплати і час доставки;
- фільтрувати і порівнювати товар за різними характеристиками;
- здійснювати пошук потрібного товару за кодом або назвою;
- мати можливість зробити замовлення і оплатити його в режимі онлайн, і через деякий час отримати його через зручний варіант доставки.

Для досягнення поставленої мети кваліфікаційної роботи потрібно реалізувати наступні завдання:

- провести аналіз предметної області;
- обґрунтувати вибір засобів розробки інтернет-магазину;
- розробити додаток Shop для CMS Webasyst, яке розширить функціонал CMS до повноцінного «Інтернет-магазину»;
- розробити додаткові плагіни, які доповнять програму "Веб-магазину" з необхідними зовнішніми сервісами оплати і доставки;
- розробка дизайн сайта і верстку макету.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Для ефективної реалізації інтернет-магазину необхідно заздалегідь продумати майбутню структуру магазину і необхідний функціонал. Виходячи з цього необхідно на початку проаналізувати існуючі можливості і рішення, після чого можна буде переходити до проектування та розробки.

В даному проекті предметною областю є каталог товарів компанії «Roliki», а об'єктами є таблиці бази даних, які зберігають і забезпечують пов-

ноцінну роботу магазину, в тому числі дозволяють здійснювати введення діяльності магазину за допомогою візуального інтерфейсу.

1.1 Опис напрямку діяльності компанії

Компанія «Roliki» – це приватна українська компанія, яка продає спортивні товари по всій Україні, за допомогою роздрібних точок і поштових відправлень по Україні. Компанія створена в Одесі в 2014 році, працює на підставі ліцензії, виданої в 2014 році.

Компанія «Roliki» – це магазин, ключовим об'єктом роботи якого є каталог товар. Виходячи з виконаного аналізу предметної області була створена структура ключових, товарних категорій компанії (рис.1.1).

Аналізуючи напрям діяльності компанії, так само були виявлені ключові об'єкти з якими регулярно взаємодіють її співробітники, які представлені на рис.1.2. Такі об'єкти обов'язково повинні мати відповідні розділи в інтернет-магазині.

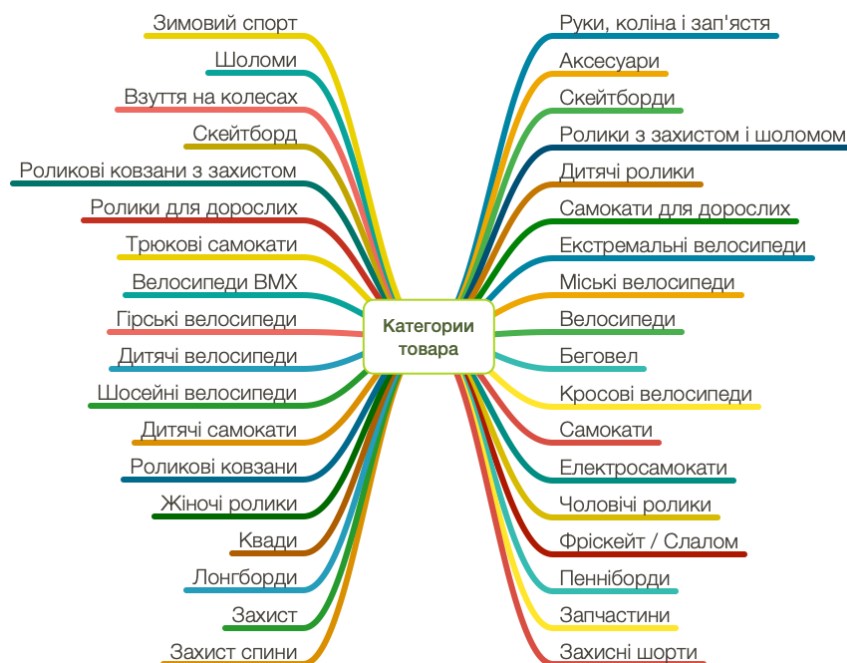


Рисунок 1.1 – Структура категорій товару

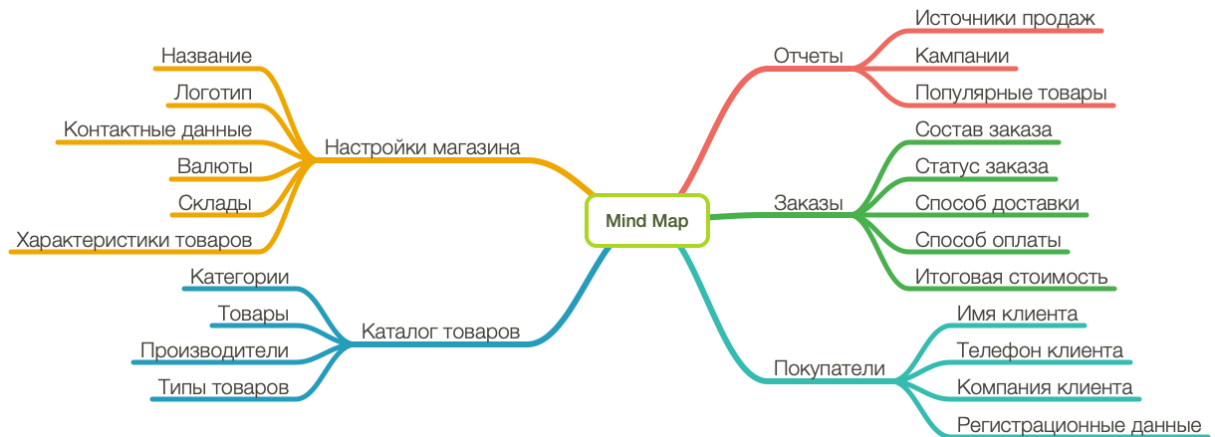


Рисунок 1.2 – Ключові об'єкти компанії

1.2 Вимоги до дизайну сайту

Якісний дизайн інтернет-магазину – це грамотне поєднання естетичної і функціональної складових інтернет-ресурсу. Інтуїтивне розуміння навігації, доступність інформації, приємне оформлення, яке затримає відвідувача на сайті і змусить купити товар, грає вирішальну роль при розробці інтернет-магазину.

У дизайні інтернет-магазину, і будь-якого сайту в цілому, існують наступні найважливіші завдання:

- зручність використання;
- привабливий зовнішній вигляд;
- запам'ятовуваність ресурсу.

Дизайн інтернет-магазину має прямий вплив на його продажі. Отже дизайн повинен:

- бути зручним у використанні;
- бути спокійним за колірною гамою;
- враховувати базові принципи побудови інтернет-магазинів в мережі інтернет, тому що все відвідувачі приходячи на сайт, вже мають минулий досвід взаємодії з іншими сайтами.

1.3 Вимоги до функціональності сайта

Так як покупки через інтернет вже є нормою для більшості жителів України, головним завданням є зробити його максимально популярним, чим популярніше він буде, тим більше продажів він зможе принести своїй компанії. Для створення популярного сайту необхідно враховувати наступні вимоги і побажання:

Адаптивний дизайн. Все більше користувачів використовують для покупок і пошуку інформації різні гаджети, які мають доступ в інтернет – смартфони, планшети, ноутбуки різної ширини екрану. Для того, щоб користувачам не доводилося збільшувати окремі елементи сайту, що вкрай незручно, необхідно адаптувати свій сайт під будь-який тип екрану.

Зручність сайту. Основне завдання будь-якого інтернет-магазину – це продажі. Тому питання юзабіліті сайту – це пріоритет для потенційних покупців. Зайшовши на сайт користувачеві має бути зрозуміло де знаходиться товар, що його цікавить, як зробити покупку, оплату і у які терміни буде здійснена доставка замовлення.

Повнота картки товару. Сторінка товару повинна містити вичерпну інформацію про товар або послугу. Покупцеві має бути зрозуміло, що це за товар, якими якостями він володіє, як купити і зробити оплату.

Оптимізованість під пошукові системи. Інтернет-магазин повинен бути повністю підготовлений і оптимізований для просування інтернет-магазину в пошукових системах (SEO). HTML-розмітка дизайну інтернет-магазину повинна бути побудована відповідно до стандартів мікроформатів опису даних Schema.org, які офіційно визнані провідними пошуковими системами.

Зрозуміла і логічна структура сайту. Інтернет-магазин повинен бути зрозумілим, передбачуваним і логічно правильним як для клієнтів, так і для співробітників компанії, ніщо так не втомлює при роботі з сайтом, як погана структура.

2 ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ РОЗРОБКИ

2.1 Мова розмітки гіпертекстових документів HTML5

HTML (HyperText Markup Language – мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML) [1]¹⁾. Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документу та ідеологію структурної розмітки тексту. HTML разом із каскадними таблицями стилів та вбудованими скриптами – це три основні технології побудови веб-сторінок. HTML впроваджує засоби для:

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

Документ HTML. Для поліпшення взаємодії, SGML вимагає аби кожна похідна мова (HTML у тому числі) визначала свою кодову таблицю для кожного документа, яка складається з репертуара (перелік різноманітних символів) та позиції символу (перелік цифрових посилань на символи з репертуара). Кожен документ HTML – це послідовність символів з репертуара.

HTML використовує найповнішу кодову таблицю UCS (Universal Character Set – Універсальний Набір Символів). Проте, однієї кодової таблиці недостатньо для того, щоб браузери могли правильно відтворювати документи

¹⁾ [1] HTML: HyperText Markup Language. URL:<http://uk.wikipedia.org/wiki/HTML> (дата звернення 18.05.2019)

HTML. Для цього браузерам потрібно «знати» специфічну кодову таблицю документа, яку автор має зазначати завжди в елементі `meta` із параметром `charset`. За замовчуванням використовується кодова таблиця ISO-8859-1, відома також як Latin-1.

Розмітка в HTML складається з чотирьох основних компонентів: елементів (та їхніх атрибутів), базових типів даних, символічних мнемонік та декларації типу документа.

Документ HTML 4.01 складається з трьох частин [2]¹⁾:

- декларація типу документу (Document type declaration, Doctype), на початку документа, в якій визначається тип документа (DTD);
- шапка документу (знаходиться в межах елемента `head`), в якій записані загальні технічні відомості або додаткова інформація про документ, яка не відтворюється безпосередньо в браузері;
- тіло документу (може знаходитися в елементах `body` або `frameset`), в якому міститься основна інформація документа.

Елементи являють собою базові компоненти розмітки HTML. Кожен елемент має дві основні властивості: атрибути та зміст (контент). Існують певні настанови щодо кожного атрибута та контенту елемента, які треба виконувати задля того, щоб HTML-документ був визнаний валідним.

У елемента є початковий тег, який має вигляд `<element-name>`, та кінцевий тег, який має вигляд `</element-name>`. Атрибути елемента записуються в початковому тегу одразу після назви елемента, контент елемента записується між його двома тегами. Наприклад: `<element-name element-attribute="attribute-value">контент елемента</element-name>`.

Деякі елементи, наприклад `br`, не містять контенту, тож і не мають кінцевого тега. Елемент може не мати початкового та кінцевого тега (наприклад, елемент `head`), проте він завжди буде представлений в документі.

¹⁾ [2] Комолова Н.В., Яковлева Е.С. HTML, XHTML и CSS. СПб.: Питер, 2012. 300 с.

Більшість з атрибутів елемента являє собою пару «назва-значення», розділених між собою знаком рівняння, та записаних у початковому тегу одразу після назви елемента. Значення атрибуту може бути окреслено лапками (подвійними або одинарними), також, якщо значення атрибуту складається з певних символів, його можна не виділяти лапками зліва. Проте, не виділення значення атрибутів в лапки вважається небезпечним кодом. На відміну від атрибутів виду «назва-значення», є певні атрибути, що впливають на елемент, назва яких лише з'явилась в початковому тегу (наприклад, атрибут `ismap` елемента `img`).

Оскільки HTML є похідною мовою від SGML, усі типи даних HTML ґрунтуються на базових типах даних SGML (наприклад, `PCDATA`, `CDATA`, `NAME`, `ID`, `NUMBER`).

Кожен елемент має дві властивості – атрибути і вміст, які мають певні значення. Всі можливі значення цих двох властивостей прописуються відповідно до визначених у DTD типів даних [3]¹⁾.

2.2 Формальна мова опису зовнішнього вигляду документу CSS

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг – можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS) [4]²⁾.

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати

¹⁾ [3] Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. СПб: Питер, 2014. 298 с.

²⁾ [4] CSS: Cascading Style Sheets. URL:<http://uk.wikipedia.org/wiki/CSS> (дата звернення 18.05.2019)

повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.)

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями. Переваги [5]¹⁾:

- інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
- різна інформація про стилі для різних типів користувачів: наприклад, великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
- сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;
- прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

CSS має порівняно простий синтаксис і використовує небагато англійських слів для найменування різних складових стилю. Стилі склада-

¹⁾ [5] Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов. М.: Эксмо, 2013. 480 с.

ються зі списку правил. Кожне правило має один або більше селектор (Selector) та блок визначення (Declaration block). Блок визначення складається із оточеного фігурними дужками списку властивостей [6]¹⁾.

2.3 Прототипно-орієнтована сценарна мова програмування

JavaScript

JavaScript (JS) – динамічна, об’єктно-орієнтована мова програмування [7]²⁾. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. Мова JavaScript також використовується для програмування на стороні серверу (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об’єктно-орієнтованої), мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам’яттю, наслідування, функції як об’єкти першого класу.

JavaScript є однією з найпопулярніших мов програмування в інтернеті. Але спочатку багато професіональних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-любителів. Поява

¹⁾ [6] Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web- сайтов. СПб.: БХВ-Петербург, 2011. 416 с

²⁾ [7] JavaScript. URL: <http://uk.wikipedia.org/wiki/JavaScript> (дата звернення 18.05.2019)

AJAX змінила ситуацію та повернула увагу професійної спільноти до мови. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, тестування та налагодження), створені бібліотеки та фреймворки, поширилося використання JavaScript поза браузером.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам,— функції як об'єкти першого класу, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має С-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності [8]¹⁾:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;
- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від С), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості та-

¹⁾ [8] Гаевский, А.Ю. Создание Web-страниц и Web-сайтов: HTML и JavaScript. М.: Триумф, 2008. 454 с.

кож не мають типів, а декларація функції може стояти в тексті програми після неї [9]¹⁾.

При використанні в рамках технології DHTML JavaScript код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги `<script></script>` з обов'язковим по специфікації HTML 4.01 атрибутом `type=»text/javascript»`, хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript.

2.4 Бібліотека jQuery

jQuery – популярна JavaScript-бібліотека з відкритим сирцевим кодом [10]²⁾. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресігом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день.

jQuery є вільним програмним забезпеченням під ліцензією MIT (до вересня 2012 було подвійне ліцензування під MIT та GNU General Public License другої версії) [11]³⁾.

Основне завдання jQuery – це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів. Також даний фреймворк надає інтерфейси для Ajax-застосунків, обробників подій і простої анімації.

Принцип роботи jQuery полягає в використанні класу (функції), який при звертанні до нього повертає сам себе. Таким чином, це дозволяє будувати послідовний ланцюг методів.

¹⁾ [9] JavaScript: Онлайн підручник. URL: <http://www.wisdomweb.ru/JS/javascript-first.php> (дата звернення 18.05.2019)

²⁾ [10] jQuery. URL: <http://uk.wikipedia.org/wiki/JQuery> (дата звернення 18.05.2019)

³⁾ [11] jQuery: Онлайн підручник. URL: <http://www.wisdomweb.ru/JQ/jquery-first.php> (дата звернення 18.05.2019)

2.5 Скриптова мова програмування PHP

PHP (Hypertext Preprocessor, гіпертекстовий препроцесор), попередня назва: Personal Home Page Tools – мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby) [12]¹⁾. PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів які виконуються вже на стороні клієнта.

PHP – мова, код якої можна вбудовувати безпосередньо в html-код сторінок, які, у свою чергу, будуть коректно оброблені PHP-інтерпретатором.

В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, Informix, InterBase, Sybase. Через стандарт відкритого інтерфейсу зв'язку з базами даних (Open Database Connectivity Standard – ODBC) можна підключатися до всіх баз даних, до яких існує драйвер [13]²⁾.

PHP 5 володіє прекрасним потенціалом реалізації об'єктного програмування. Окрім цього, PHP збагатився рядом цінних розширень для роботи з XML, різними джерелами даних, генерації графіки і інше. Серед інших українських доповнень в PHP 5 слід зазначити нову схему обробки виключень. Конструкція try/catch/throw дозволяє весь код обробки помилок локалізувати

¹⁾ [12] PHP: Hypertext Preprocessor. URL: <http://uk.wikipedia.org/wiki/PHP> (дата звернення 18.05.2019)

²⁾ [13] Справочник по PHP. URL: <http://www.php.su> (дата звернення 18.05.2019)

в одному місці сценарію. У PHP 5 також включені два нових модулі для роботи з протоколами – SimpleXML і SOAP. SimpleXML дозволяє значно спростити роботу з XML-даними, представляючи вміст XML-документа у вигляді PHP-об'єкта. Розширення SOAP дозволяє будувати на PHP сценарії, що обмінюються інформацією з іншими застосуваннями за допомогою XML-повідомлень поверх існуючих веб-протоколів, наприклад, HTTP. Новий модуль PHP 5 MySQLi (MySQL Improved) призначений для роботи з MySQL-сервером версій 4.1.2 і вище, реалізуючи не тільки процедурний, але і об'єктно-орієнтований інтерфейс до MySQL. Додаткові можливості цього модуля включають – SSL, контроль транзакцій, підтримка реплікації та ін. Очевидно, що, на цьому історія PHP не закінчується. Слід очікувати наступних версій мови із розширеними можливостями [14,15]¹⁾.

2.6 Мова структурованих запитів SQL

SQL (Structured query language – мова структурованих запитів) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції.

¹⁾ [14] Зандстра М. PHP: объекты, шаблоны и методики программирования. Издательский дом «Вильямс», 2011. 560 с.

[15] Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. СПб.: Питер, 2013. 496 с.

SQL також включає CLI (Call Level Interface) для доступу і управління базами даних дистанційно.

SQL складається з наступних елементів мови [16]¹⁾:

- положення, які є складовими компонентами заяв і запитів;
- предикати для тризначної логіки (3VL) (так / ні / невідомо) або логічних значень, які використовуються для обмеження наслідків заяв та *запитів або змінити хід виконання програми;
- запити для видачі скалярних величин або таблиць, що складаються із стовпців і рядків даних;
- запити вилучення даних на основі певних критеріїв;
- заяви впливу на схеми і даних, зв'язків;
- заяви управлінням транзакціями, ходом виконання програми, завдань та діагностики.

Нижче наведені переваги SQL.

Незалежність від конкретної СКБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СКБД в іншу. Існують системи, розробники яких спочатку закладалися на застосування щонайменше кількох СКБД (наприклад: система електронного документообігу Documentum може працювати як з Oracle, так і з Microsoft SQL Server та IBM DB2). Природно, що при застосуванні деяких специфічних для реалізації можливостей такої переносимості добитися вже дуже важко.

Наявність стандартів. Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах, наприклад, Core-частину стандарту SQL:2003 включає понад 1300 сторінок тексту.

¹⁾ [16] SQL: Structured Query Language. URL:<http://uk.wikipedia.org/wiki/SQL> (дата звернення 18.05.2019)

Декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СКБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип – програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СКБД розбиратиме текст його запиту. Особливо критичні такі моменти стають при роботі з великими базами даних і зі складними запитами – чим складніше сконструйований запит, тим більше він допускає варіантів написання, різних за швидкістю виконання, але тих самих за набором даних.

До недоліків SQL слід віднести наступне: невідповідність реляційної моделі даних. Творці реляційної моделі даних Едгар Кодд, Крістофер Дейт та їхні прихильники указують на те, що SQL не є істинно реляційною мовою. Зокрема вони указують на такі проблеми SQL:

- рядки, що повторюються;
- невизначені значення (null);
- явна вказівка порядку стовпчиків зліва направо;
- стовпці без імені та імена стовпчиків, що дублюються;
- відсутність підтримки властивості «=>»;
- використання покажчиків;
- висока надлишковість.

2.7 Система керування базами даних MySQL

MySQL – вільна система керування реляційними базами даних. MySQL була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних [17]¹⁾. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширюва-

¹⁾ [17] Справочное руководство по MySQL. URL:<http://www.mysql.ru/docs/man/> (дата звернення 18.05.2019)

лася і зараз MySQL – одна з найпоширеніших систем керування базами даних.

MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GPL [18]¹⁾.

Гілка MySQL 5.5 містить ряд значних поліпшень, пов'язаних з підвищенням масштабованості та швидкодії, серед яких:

- використання за замовчуванням рушія InnoDB;
- підтримка напівсинхронного (semi-synchronous) механізму реплікації, заснованого на патчах до InnoDB від компанії Google.
- поліпшення функцій з партіціювання даних. Розширений синтаксис для розбиття великих таблиць на кілька частин, розміщених в різних файлових системах (partitioning). Додані операції RANGE, LIST і метод оптимізації «partition pruning»;
- новий механізм оптимізації вкладених запитів та операцій JOIN.
- перероблена система внутрішніх блокувань;
- інтегровані патчі Google з оптимізацією роботи InnoDB на CPU з великою кількістю ядер.

MySQL – компактний багатонитевий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

¹⁾ [18] Б. Шварц. MySQL. Оптимизация производительности. СПб.: Символ-Плюс, 2010. 832 с.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатонитковості, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

2.8 Підхід до створення web-застосувань AJAX

AJAX (Asynchronous JavaScript And XML) – підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звітти довантажує потрібні користувачу дані [19]¹⁾. AJAX – один з компонентів концепції DHTML.

AJAX – це не самостійна технологія, а швидше концепція використання декількох суміжних технологій. AJAX підхід до розробки, який призначений для користувачів інтерфейсів, комбінує кілька основних методів і прийомів:

- використання DHTML для динамічної зміни змісту сторінки;
- використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю;

¹⁾ [19] AJAX: Asynchronous Javascript and XML. URL: <http://uk.wikipedia.org/wiki/AJAX> (дата звернення 18.05.2019)

- альтернативний метод – динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON);
- динамічне створення дочірніх фреймів.

AJAX – асинхронний, тому користувач може переглядати далі контент сайту, поки сервер все ще обробляє запит. Браузер не перезавантажує веб-сторінку і дані посилаються на сервер без візуального підтвердження (крім випадків, коли ми самі захочемо показати процес з'єднання з сервером).

Порівняння класичного підходу та AJAX. Класична модель веб-застосування:

- користувач заходить на веб-сторінку і натискає на який-небудь її елемент;
- браузер надсилає запит серверу;
- у відповідь сервер генерує повністю нову веб-сторінку і відправляє її браузеру і т.д. З боку сервера можлива генерація не всієї сторінки наново, а тільки деяких її частин, з подальшою передачею користувачу.

Модель AJAX:

- користувач заходить на веб-сторінку і натискає на який-небудь її елемент;
- браузер відправляє відповідний запит на сервер;
- сервер віддає тільки ту частину документа, яка змінилася.

Переваги:

- економія трафіку;
- зменшення навантаження на сервер;
- прискорення реакції інтерфейсу;
- майже безмежні можливості для інтерактивної обробки.

Недоліки:

- відсутність інтеграції зі стандартними інструментами браузера;

- зміни вмісту сторінки при постійному URL полягає в неможливості збереження закладки на бажаний матеріал;
- старі методи обліку статистики сайтів стають неактуальними. Багато сервісів статистики ведуть облік переглядів нових сторінок сайту. Для сайтів, сторінки яких широко використовують AJAX, така статистика втрачає актуальність;
- ускладнення проекту. Перерозподіляється логіка обробки даних – відбувається виділення і часткове перенесення на сторону клієнта процесів первинного форматування даних. Це ускладнює контроль цілісності форматів і типів;
- потрібно включений JavaScript в браузері;
- низька швидкість при грубому програмуванні;
- ризик фабрикації запитів іншими сайтами. Для захисту використовують POST-запит. Але GET вважається ідемпотентність і тому кеширується, POST – ні, тому Google вставляє в початок відповіді нескінченний цикл: AJAX може робити з відповіддю що завгодно, у тому числі прибрати цикл, а тег `<script>` підключить скрипт як є і зациклиться.

2.9 Фреймворк для створення веб-інтерфейсів Bootstrap

Bootstrap – фреймворк, набір HTML + CSS інструментів і шаблонів для верстки та більш ефективного і швидкого створення сайтів і веб-додатків.

Bootstrap – сучасний помічник, розробників інтерфейсів, дизайнерів і веб-майстрів, доступний для використання за відкритою ліцензією. Фреймворк дуже динамічний і регулярно оновлюваний, тому не всі його функції можуть коректно підтримуватися старими браузерами [20]¹⁾.

¹⁾ [20] Что такое Bootstrap. URL:<http://blogwork.ru/chto-takoebootstrap/> (дата звернення 21.05.2019)

У Bootstrap є наступні шаблони: шрифти, кнопки, форми, мітки, навігація, сітка, JavaScript-розширення.

Зараз найбільш популярною версією фреймворка Bootstrap є третя. У ній подальший розвиток отримала адаптивність і принцип mobilefirst.

Сучасний світ дуже динамічний, і веб-програмісти часто стикаються з вибором – робити швидкі прототипи і запускати їх або повільно і довго переробляти верстку, доводячи все до ідеального стану. Bootstrap прекрасно реалізує перший підхід.

Переваги фреймворка Bootstrap:

- економія часу – досягається за рахунок використання вже готових класів і дизайну. Це дозволяє направити заощаджені енергію і гроші на розробку додаткової функціональності;
- адаптивність (висока швидкість і оптимізація, стандартизація інтерфейсів) - динамічні макети якісно відображаються на самих різних пристроях без необхідності внесення змін до розмітку;
- дизайн – єдині шаблони і стильове оформлення елементів макета і всіх сторінок на сайті в цілому. І при цьому Bootstrap крос-браузерні і добре відображається у всіх браузерах Safari, Firefox, IE, EDGE і тих, що на основі Chromium (движок Blink на основі Webkit: яндекс.браузер, Opera, Google Chrome);
- простота і відкритість – використовувати Bootstrap настільки просто, що з ним справляються навіть початківці веб-розробники, а відкритий вихідний код дозволяє брати участь в розробці, модифікувати під ужди або просто користуватися хорошим безкоштовним рішенням.

При цьому код HTML, JavaScript і CSS в Bootstrap продуманий сотнями розробників з усього світу – все для того, щоб пересічні веб-майстри і верстальники могли легко і просто налаштувати сітку сайту або вбудувати необхідні елементи в інтерфейс.

Також, в Bootstrap використовується динамічний мову стилів LESS, які розширює можливості CSS: розробники можуть керувати квітами, створювати вкладені колонки і змінні.

До мінусів Bootstrap можна віднести:

- одноманітність – Bootstrap часто критикують за те що сайти виглядають однаково. Всі сайти, які використовують Bootstrap, схожі один на одного, тобто відсутня унікальність. Подібні сайти-близнюки просто не запам'ятовуються;
- негнучкість – якщо потрібно створити що-небудь відмінне від інтерфейсу Bootstrap, то доводиться з найперших кроків боротися зі стилями за замовчуванням. На практиці виходить подвійна праця;
- надмірний код – те, що реально зробити двома вкладеними блоками, часто робиться п'ятьма. Будь-які зміни тягнуть за собою годинник мук;
- обмеження у використанні в старих браузерах – регулярне оновлення і доповнення фреймворка найсучаснішими можливостями HTML і CSS вносить деякі обмеження у використанні з IE7 і IE8.

Розвиток інтернету і вебсайтів показало, що в 2019 році половина користувачів інтернету відвідує сайти з мобільних пристроїв і невеликих екранів. Це означає, що сучасний веб-розробник повинен думати про функціонування сайту не тільки на ПК, але і на смартфонах і планшетах з тачскріном.

Створення окремої мобільної версії для сайту звичайно вихід, але тоді потрібно робити в два рази більше роботи на розробку і підтримку коду, а це не завжди економічно ефективно.

Концепція чуйного веб-дизайну, втіленого в фреймворку Bootstrap, вирішує саме цю проблему: сайт однаково відображає інформацію найбільш повним чином незалежно від типу екрану і розміру пристрою. Зміст і колірна гамма не змінюються, змінюється лише форма і спосіб згрупувати

інформаційні та навігаційні блоки сайту найбільш зручним для користувача способом.

Bootstrap це сучасний багатофункціональний фреймворк. Їм можна користуватися регулярно або навпаки лише в разі потреби, але кожен початківець веб-майстер, верстальник або фронтенд-розробник в обов'язковому порядку повинен мати в своєму арсеналі мінімальний набір навичок і знань для роботи з Bootstrap.

3 ОПИС ЕТАПІВ ПРОЕКТУВАННЯ САЙТУ

Проектування сайту – ключовий етап створення інтернет-магазину, в процесі якого визначаються цілі та завдання створення ресурсу, способи досягнення цілей і як сайт в підсумку буде виглядати і функціонувати.

Переваги проектування: висока гарантія досягнення результатів, економія часу і коштів, ефективний поділ роботи. виправити помилку на етапі проектування набагато дешевше, ніж на будь-який іншому етапі в майбутньому.

Основне призначення інтернет-магазину це набір інструментів, які спрямовані на поліпшення сервісу, залучення більшої кількості покупців, зменшення торговельних і амортизаційних витрат, економічне розміщення і демонстрацію товару, доступність найширшому контенту, гнучку рекламу продукції, що продається.

Метою створення інтернет-магазину для компанії Roliki є створення можливості вибору і покупки товару через інтернет і огляд усього каталогу товарів в різних форматах: фото товару, текстовий опис, характеристики, відео та ін. Так само важливо, щоб інтернет-магазин мав гнучкі настройки прав доступу для співробітників компанії.

3.1 Вибір системи керування контентом Webasyst

Для розробки інтернет-магазину є декілька найголовніших варіантів вибору системи управління сайтом. Розглянемо їх.

Хмарні рішення сторонніх компаній (наприклад, pro.ua) які дозволяють створити інтернет-магазин на шаблоні, буквально за день. Основний мінус такого рішення – обмежений функціонал і повна залежність цієї компанії.

Створення своєї системи управління сайтом з нуля. Це найкращий, але одночасно самий витратний по ресурсах шлях. Цей варіант завжди виграє при нестандартних завданнях, у випадку ж інтернет-магазину, це спроба ви-
найти велосипед.

Створення сайту на базі готової системи управління сайтом для інтернет-магазину є найкращим рішенням. Серед багатьох CMS, був обраний саме Webasyst по наступними причинами:

- Webasyst – це насправді фреймворк з відкритим вихідним кодом, а значить, можливості для кастомізації бізнес-логіки просто безмежні. Можна додати і переписати що завгодно;
- Webasyst надає гнучкі інструменти для настройки прав доступу до додатків (і їх даних) для користувачів і груп користувачів. Розробнику програми необхідно лише оголосити елемент настройки прав доступу і потім перевіряти його значення в своєму коді. Інтерфейси налаштування прав доступу вже реалізовано в меню «Контакти»;
- кожен користувач входить зі своїм логіном і паролем, бачить тільки ті додатки, до яких у нього відкритий доступ, і виконує всі дії від свого імені. Більше не доведеться створювати користувача admin і всім видавати однаковий пароль;
- разом з безкоштовним додатком «Сайт» фреймворк Webasyst перетворюється в CMS/CMF (Content Management System/Framework), яка підійде для створення навіть самих склад-

них сайтів, що працюють на основі декількох незалежних додатків, гнучко пов'язаних між собою загальним роутингом і темами дизайну;

- Webasyst надає набір готових CSS-класів, за допомогою яких легко формуються гарні інтерфейси, оптимізовані під браузері Firefox 3.6+, Google Chrome, Opera 11, Safari, Internet Explorer 8 і 9;
- Webasyst API дозволяє працювати з даними додатків Вебасіста віддалено, поза оточення веб-сервера, на якому працює фреймворк. Наприклад, з десктоп-клієнтів або додатків для смартфонів. Доступ до API проводиться через єдину точку доступу `api.php`.

Webasyst – це PHP-фреймворк (платформа) для розробки веб-додатків, у яких є бекенд і фронтенд. За допомогою фреймворка Webasyst можна створювати веб-проекти різної спрямованості, проте найкраще фреймворк підходить саме для веб-проектів з бекендом і фронтенда, розрахованих на багато користувачів доступом і гнучкою можливістю розмежування прав доступу [21]¹⁾.

3.2 Проектування бази даних

Робота з базою реалізує шар «модель» в термінології MVC. До складу фреймворка входить абстрактний клас `waModel` для опису моделі. Класи опису сутностей предметної області додатку успадковуються від `waModel`.

Файли класів моделі розміщуються в файловій структурі додатку в підкаталозі додатку `lib/ models/`. Файлова структура сайту наведена у додатку А.

В описі моделі є тільки одна обов'язкова поле – назва таблиці в базі даних в змінній `$ table`. Приклад класу опису моделі:

¹⁾ [21] Центр управління бизнесом – Webasysy. Офіційний сайт. URL: <https://www.webasyst.ru/> (дата звернення 22.05.2019)

```
<?php
class blogModel extends waModel
{
    protected $table = 'blog';
}
```

Модель ініціалізується в коді додатку наступним чином:

```
$model = new blogModel();
```

У даному прикладі `blog` – це приклад ідентифікатора додатка. Після ініціалізації модель отримує від бази даних опис зазначеної таблиці (за допомогою запиту `DESCRIBE $ table`) і кешує його. На цьому етапі визначається головний ключ (`primary key`) таблиці, визначаються поля та їх типи.

У коді контролерів і екшенів все взаємодія з базою даних відбувається виключно через класи моделей програми:

```
// Створюємо екземпляр моделі для для отримання даних з БД
$model = new guestbookModel();
// Отримуємо записи гостьової книги з БД
$records = $model->order('datetime DESC')->fetchAll();
```

Клас `waModel` містить ряд сервісних методів для автоматичної побудови SQL-запитів, що дозволяє досягти незалежності коду програми від конкретного виду використовуваної СКБД.

3.3 Конфігураційні файли Webasyst

Файли повинні зберігатися в піддиректорії додатку `lib/config /`.

Файл `app.php` – це головний конфігураційний файл програми. У ньому задаються назва програми, шлях до іконки, номер версії, ознака наявності фронтенда і інші параметри.

Розглянемо деякі параметри у файлі `app.php`:

- `name` – назва програми, обов'язковий параметр. Цю назву видно в головному меню вибору додатків в бекенде. Якщо додаток багатомовний, то в якості назви потрібно ключ локалізації для рядка, який буде автоматично переведений на кожен з підтримуваних мов.

- `img` – відносний шлях до файлу іконки програми, від кореня директорії додатку. Іконка видна поруч з назвою додатка в головному меню бекенда. Зображення іконки повинно бути розміром 48x48 пікселів. Параметр `img` необов'язковий. Якщо він не вказаний, то файл з іконкою повинен бути розміщений по стандартному шляху: `img / [app_id] .png`.
- `icon` – масив шляхів до файлів додаткових ікон різного розміру. Використовується замість параметра `img` в тих випадках, коли іконки декількох розмірів представлені окремими файлами. Іконки розмірів 24x24 і 16x16 використовуються для інтеграції з додатком «Сайт». Якщо окремих файлів іконок немає, то використовується стандартна іконка розміром 48x48, яка масштабується браузером.
- `prefix` – префікс назв таблиць додатка в базі даних. Необов'язковий параметр. Значення за замовчуванням – ідентифікатор додатки `app_id`.
- `rights` – прапор, що позначає, чи використовуються в додатку розширені настройки доступу. Необов'язковий параметр. Значення за замовчуванням: `false`. Детальніше про систему розмежування прав написано в статті «Права доступу».
- `frontend` – прапор, що позначає, чи є у додатку фронтенд. Необов'язковий параметр. Значення за замовчуванням: `false`.
- `version` – номер версії програми.
- `sash_color` – позначення кольору маркера додатки для записів в «Стрічці подій» Webasyst.
- `vendor` – ідентифікатор розробника Webasyst, обов'язковий для публікації додатка в магазині Webasyst.
- `routing_params` – масив додаткових параметрів, які застосовуються за замовчуванням при додаванні для правила маршрутизації додатки в структурі сайту. За замовчуванням правило створюється публічно доступним.

4 ДИЗАЙН ІНТЕРНЕТ-МАГАЗИНУ

4.1 Проектування дизайну сайту

Дизайн сайт потрібно розробити для ключових сторінок будь-якого інтернет-магазину, це: головна сторінка, категорія товару, сторінка товару і корзина. Розберемо вимоги для кожного типу сторінок, а так само наскрізні вимоги.

Головна сторінка це основа для всіх інших сторінок інтернет-магазину. Важлива проста і зрозуміла навігація: меню, розділи, категорії та підкатегорії. Основні продукти компанії потрібно виділити в окремий розділ «Хіти продажу», «Краща ціна» і «Новинка» і публікувати їх окремими блоками на головній. Людям завжди цікаво, що купують інші, що сьогодні купити вигідно, і що з'явилося нового.

Категорія товару. На даній сторінці важливо зробити все для зручності вибору товару клієнтом. Обов'язково потрібно щоб у клієнта була можливість: сортувати товар по різних полях (ціна, назва, популярність та ін), фільтрувати товар за різними характеристиками (фірмі, розміром, кольором та ін), і порівнювати товар між собою.

Сторінка товару. На сторінці товару впершу чергу важливо приділити увагу фотографії. Показати товар у всіляких ракурсах, зробити великі (не менше 500px по більшій стороні), і світлі фотографії (на яких видно всі деталі). Так само важливо зробити розгорнутий блок з текстовою інформацією. Сюди включаємо: ціну, опис і характеристики, варіанти покупки (колір, розмір), схожі товари та ін. Всі ці елементи повинні грамотно поєднуватися один з одним. Не можна не відзначити важливість кнопки «Додати в кошик», кожен відвідувач повинен її помітити, щоб в разі готовності купити товар, покупець знав що робити на наступному етапі.

Сторінка кошика. Найголовніше для кошика це її доступність з усіх сторінок сайту, щоб відвідувач завжди знав де знайти корзину своїх покупок.

Звичне місце для кошика – верхній правий кут. Товари в кошику мають бути показані не просто назвою і описом, а з фотографією і посиланням на картку товару. Це потрібно для того, щоб користувач міг у будь-який момент подивитися, що ж він вибрав, і можливо, змінив кількість потрібного товару.

Наскрізні блоки (використовуються на всіх сторінках сайту):

- логотип компанії;
- навігація по товарних категоріях;
- навігація по інформаційних сторінках;
- пошук товару;
- телефон і графік роботи.

Прототип шапки сайту «Header» показано на рис. 4.1. Прототип основного контенту «Body» наведений на рис. 4.2, а прототип підвала сайту «Footer» – на рис. 4.3.



Рисунок 4.1 – Прототип шапки сайту «Header»

4.2 Реалізація дизайну сайту

Дизайн був розроблений з урахуванням всіх вище перчислених вимог і по аналогічній структурі, що була перерахованих в розділі проектування: основні сторінки і наскрізні блоки.

Дизайн шапки сайту «Header» показаний на рис. 4.4. Дизайн був виконаний в рамках проектування, винятком є лише блок з реєстрацією та авторизацією, який був доданий в процесі розробки дизайну.



Рисунок 4.2 – Прототип основного контента «Body»

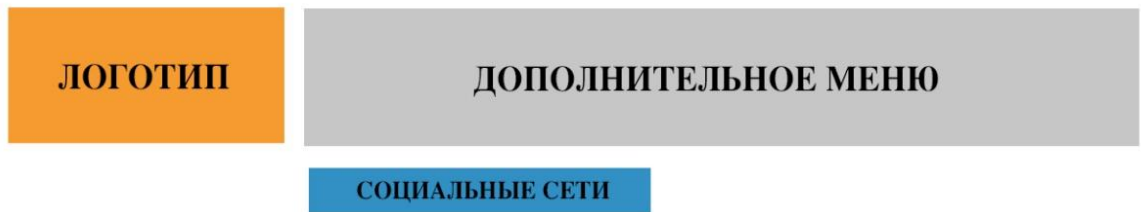


Рисунок 4.3 – Прототип подвала сайта «Footer»

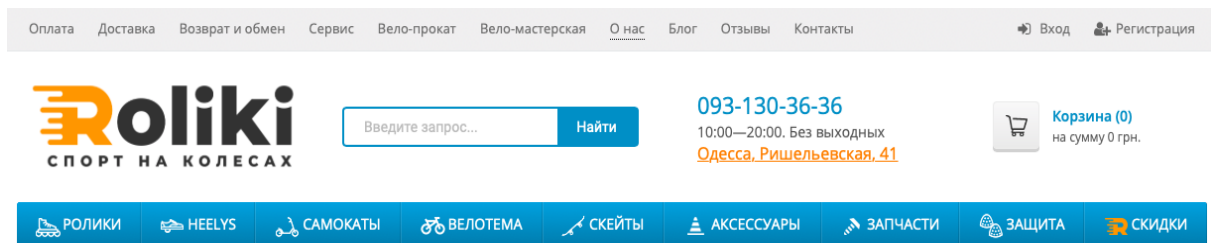


Рисунок 4.4 – Дизайн шапки сайту «Header»

Дизайн підвалу сайту «Footer» представлений на рис. 4.5. Це фінальний блок будь-якої сторінки, тут розміщена найменш важлива інформація.

Дизайн «Головную сторінки», блок популярних категорій, представлений на рис. 4.6. Даний блок розкриває найпопулярніші, ключові категорії товарів, який користуються великою популярністю серед відвідувачів.

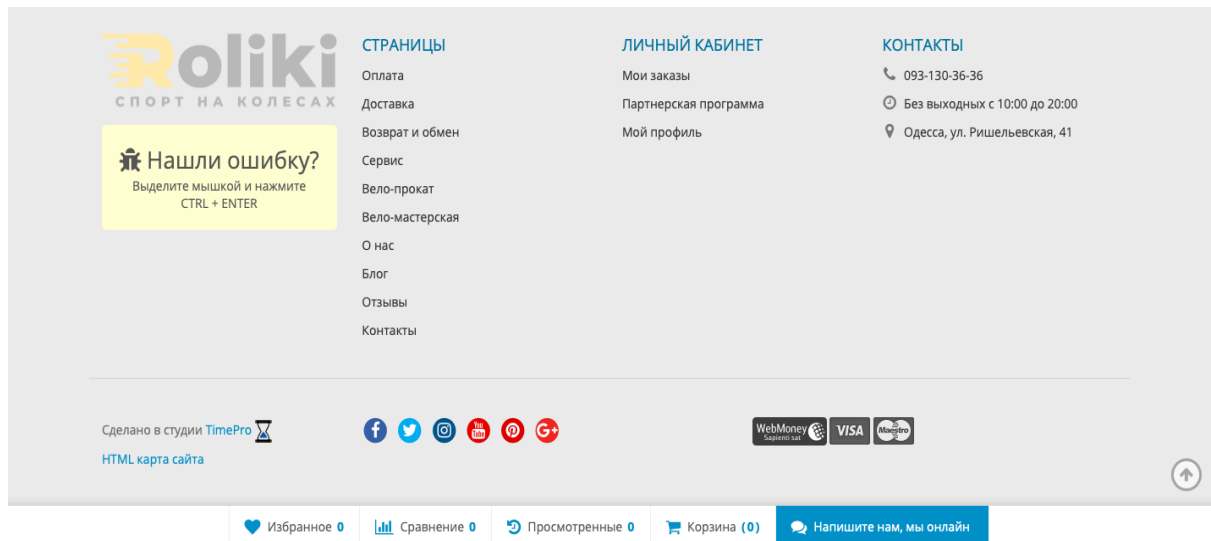


Рисунок 4.5 – Дизайн підвалу сайту «Footer»

ПОПУЛЯРНЫЕ КАТЕГОРИИ



Рисунок 4.6 – Блок популярних категорій

Був виконаний дизайн «Головную сторінки»: блок переваг і популярних товарів представлений. Дизайн «Категорії товарів»: блок з фільтром і пі-

дкатегоріями представлений на рис. 4.7. Це ключовий блок сторінки, він дає можливість перейти до вибору товару з категорії, або перейти до підкатегорії. Дизайн «Категорії товарів»: блок з самим товаром і функцією сортування товару представлений на рис. 4.8.

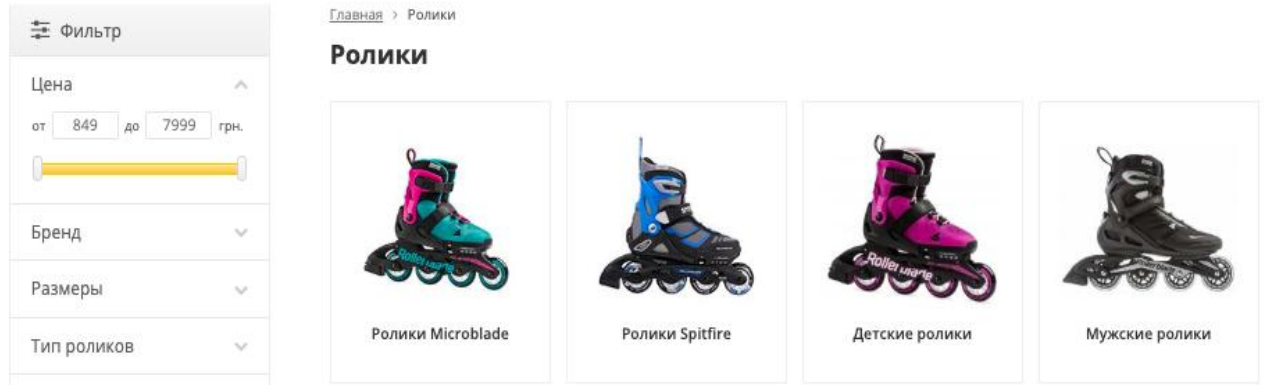


Рисунок 4.7 – Блок з фільтром і підкатегоріями

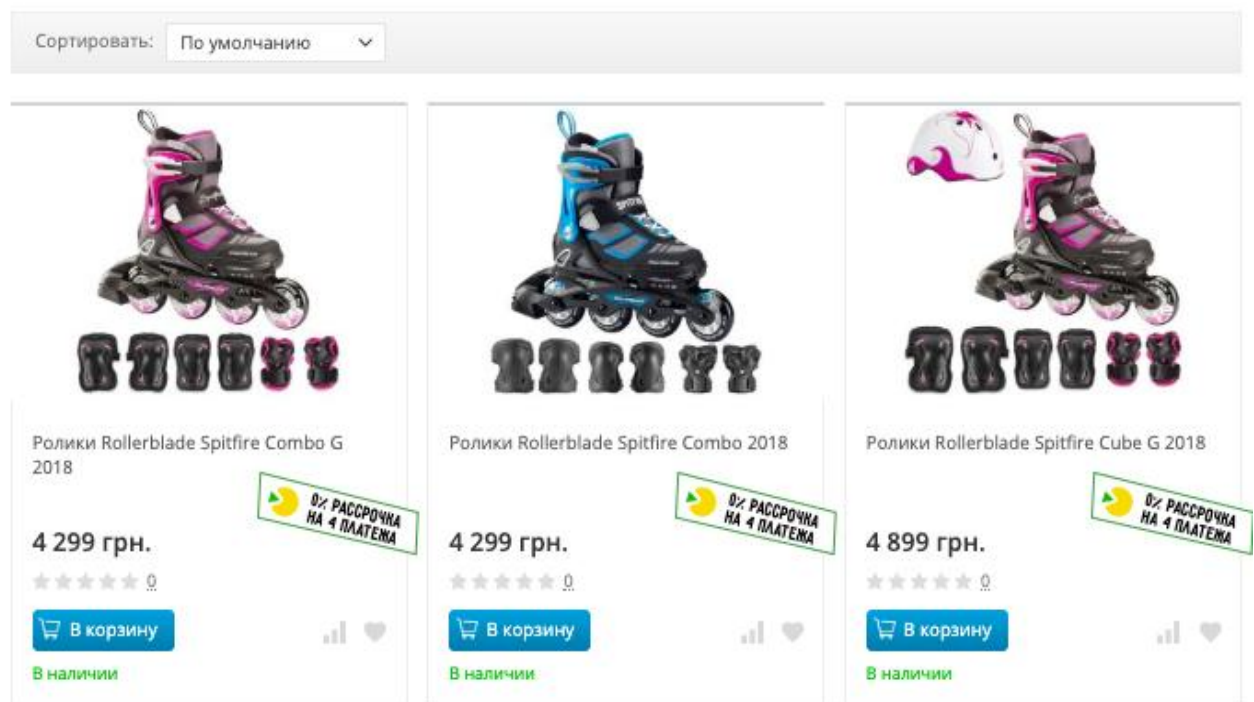


Рисунок 4.8 – Блок з самим товаром і функцією сортування товару

Дизайн «Сторінка товару» перший товарний блок представлений на рис.4.9.

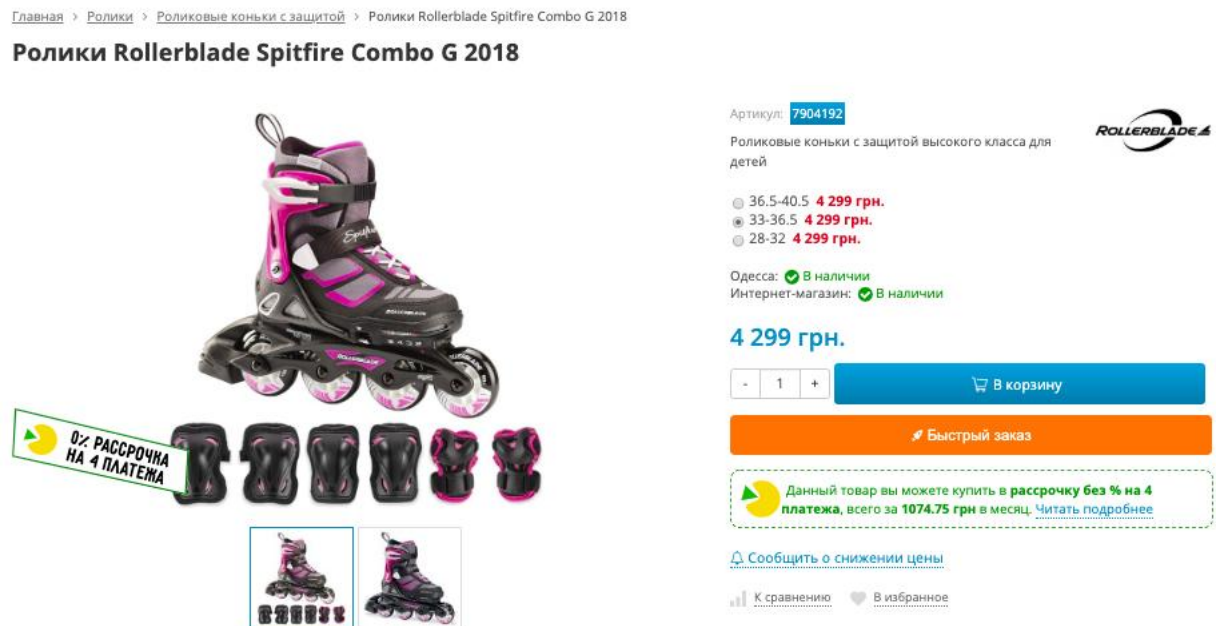


Рисунок 4.9 – Дизайн «Сторінка товару»: перший товарний блок

В даному блоці відображені найбільш пріоритетні дані для відвідувачів. Зверху повна назва товару. Зліва відображено основне фото, і додаткові мініатюри. Справа інформаційний блок в якому можна прочитати короткий опис, побачити ціну і варіанти покупки даного товару, артикул, статус товару, а так само знайти кнопки для покупки або додавання в корзину.

Дизайн «Сторінка товару» другий товарний блок представлений на рис. 4.10. Тут відображена вся додаткова інформація про товар: характеристики, детальний текстовий опис і т.д. Блок повинен бути зручним для наповнення за рахунок вкладок.

Дизайн «Сторінка товару» третій товарний блок «Альтернативні товари» представлений на рис.4.11. Цей блок задуманий для зручності покупців, якщо товар, який вони дивляться, викликає сумніву, в даному блоці можна подивитися альтернативні варіанти.

Обзор | Характеристики | Отзывы (0) | Оплати частями | Размерная сетка | 9 причин купить у нас

Новая модель 2018 года для девочки с комплектом защиты высокого качества.

Ролики раздвигаются на 4 полных размера. Очень хорошо фиксируют ногу. Для первых роликов, это одна из лучших моделей.

Они легкие, прочные. Очень хорошо удерживают ногу. Ребенку в таких роликах легко себя удерживать и набирать скорость. Колеса и подшипники тихие и отлично поддерживают скорость. В базовой комплектации колеса 72мм (не очень большие и мягкие), но можно оставить и больше при необходимости.

Ролики Rollerblade по праву можно считать лучшими детскими роликами всех времен и народов. Именно в детском сегменте они действительно сильны. Хотя и взрослая фитнес-линейка у них тоже очень неплохая.

По размеру, если нога 16см и более, а рост 95см и выше смело можно покупать эту модель. Для детей в возрасте 7-8 лет выбирайте размер (33-36.5) Если старше 10 лет, рекомендую размер (36.5 -40.5)

Рисунок 4.10 – Дизайн «Сторінка товару»: другой товарный блок

РЕКОМЕНДУЕМ ПОСМОТРЕТЬ

Назва товару	Ціна	Статус
Ролики Rollerblade Spitfire Combo 2018	4 299 грн.	В наличии
Ролики детские Rollerblade Microblade Combo G 2019	4 299 грн.	В наличии
Ролики детские Rollerblade Microblade Combo 2019	4 299 грн.	В наличии
Ролики Rollerblade Spitfire Cube G 2018	4 899 грн.	В наличии

Рисунок 4.11 – Дизайн «Сторінка товару»: третій товарний блок «Альтернативні товари»

Дизайн «Сторінки кошика» блок контактних даних представлений на рис.4.12. Всі зазначені поля клієнт заповнює для того, щоб в майбутньому співробітник компанії зміг з легкістю оформити замовлення, і відправити його замовнику, і в разі чого зв'язатися з ним. В цьому же блоці користувач має можливість ввести при наявності купони на надання знижки.

Дизайн «Сторінки кошика» блок доставки і оплати представлений на рис.4.13. Відвідувач вибирає тип доставки, і спосіб оплати, після чого здійс-

нює оплату і через деякий час забирає замовлений товар за вказаними даними. Є можливість підключення різних платіжних систем.

Фамилия *	Молодецкий	Имя *	Вячеслав
Телефон *	+38 (063) 525-36-58	<input type="radio"/> Не звоните мне!	Email *
			hiti22@mail.ru
Город	Киев	Купон	

Рисунок 4.12 – Дизайн «Сторінки кошика»: блок контактних даних

Доставка

<input checked="" type="radio"/>	Самовывоз из магазина (Одесса, Ришельевская, 41)	Бесплатно
<input type="radio"/>	Нова Пошта (На склад)	Менеджер просчитает стоимость доставки
<input type="radio"/>	Нова Пошта (Адресная доставка)	Менеджер просчитает стоимость доставки

Оплата

<input checked="" type="radio"/>	Наличными	<input type="radio"/>	POS-терминал
<input type="radio"/>	Visa, Maestro	<input type="radio"/>	Приват24
<input type="radio"/>	Оплата частями		

Рисунок 4.13 – Дизайн «Сторінки кошика»: блок доставки і оплати

5 РОЗРОБКА ДОДАТКУ SHOP

Тепер, коли дизайн інтернет-магазину реалізовано можна перейти до написання програмного коду. У наступному розділі наведемо докладний опис етапів розробки додатку Shop, що був створений спеціального для даного інтернет-магазину.

Перед початком розробки потрібно включити режим розробки `debug_mode` (включення цього режиму відключає всі механізми кешування у фреймворку). Режим розробки можна включити або в налаштуваннях програми «Інсталлер», або в файлі `wa-config / config.php`.

В якості ідентифікатора додатка (`APP_ID`) візьмемо `shop` і створимо в каталозі `wa-apps / підкаталог з такою ж назвою - wa-apps / shop /`. Це буде основним місцем зберігання файлів програми: PHP-файлів, шаблонів, стилів, скриптів JavaScript і т. П. Важливо, щоб назва цього підкаталогу збігалася з ідентифікатором додатка.

Бекенд (адміністративна частина, `backend`) додатка буде відображати список всіх об'єктів інтернет-магазину в хронологічному порядку з можливістю видалення, а фронтенд (загальнодоступна частина, `frontend`) – список категорій, товарів та інформаційних сторінок.

Створюємо основний файл з описом програми: `wa-apps / shop / lib / config / app.php` (назва файлу має бути саме таким). Додамо в нього наступний код:

```
<?php
return array(
    'name' => Shop, // назва додатка
    'img' => 'img/shop.svg' // відносний шлях до файлу іконки
                        додатку
);
```

Назва програми та шлях до файлу іконки – це обов'язкові параметри файлу опису додатка поряд з його `APP_ID`. Крім них, файл опису може містити і інші параметри.

Тепер все готово для того, щоб додаток з'явилося в головному меню бекенда. Для цього необхідно «включити» його в системному файлі конфігурації фреймворка `wa-config / apps.php`

Після додавання рядка зберігаємо файл і відкриваємо бекенд фреймворка в браузері. Інтерфейс програми тепер доступний за адресою `http: // {адрес_установкі_вебасіста} / webasyst / shop /`. Якщо є повні адміністративні

права, то можна відразу побачити іконку програми в верхньому меню Вебасіста, представлену на рис.5.1.

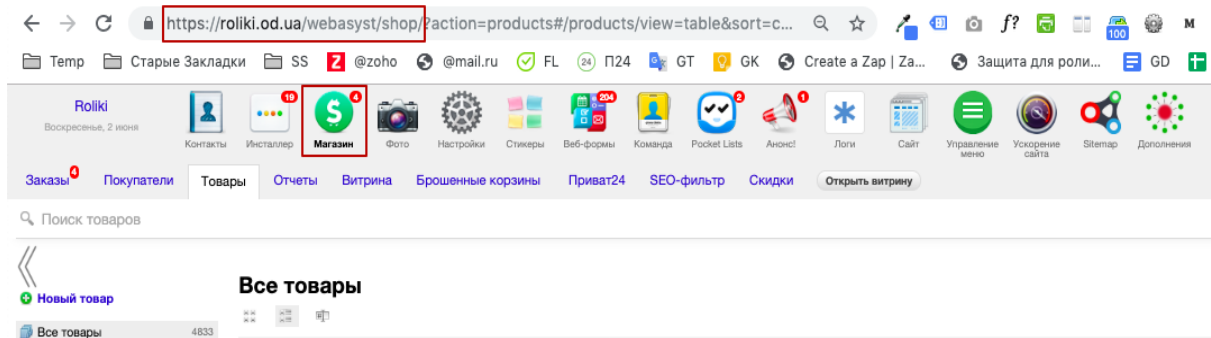


Рисунок 5.1 – Іконка програми в верхньому меню Вебасіста

Для початку створимо базовий функціонал для створення, видалення та редагування ключових об'єктів інтернет-магазину.

Для цього необхідно написати ряд екшенів (action). Екшен – це структурний елемент деякого дії, виконуваного додатком. Приклади таких дій: показати окрему сторінку бекенда, показати екран фронтенда, зберегти дані з форми реєстрації, видалити запис, отримати нову порцію даних за допомогою AJAX і т.п. Екшен є PHP-код, оформлений у вигляді окремого методу класу або цілого класу. Фреймворк запускає той чи інший екшен в залежності від вхідних даних (вмісту GET- або POST-запиту). Вибір конкретного екшену виконується контролером.

Логічно об'єднані контролери і екшени називаються модулями. Наприклад, модуль роботи з замовленнями в деякому додатку може надавати наступний набір екшенів: показати список усіх замовлень, видалити замовлення, додати замовлення, оновити інформацію про замовника, змінити статус замовлення і т. д. Модулі є структурними «цеглинками» додатка, їх вихідний код розташовується в каталогу `ewa-apps / {APP_ID} / lib / actions /`.

Кожен модуль – це або окремий PHP-файл з ім'ям виду `{APP_ID}{MODULE}.actions.php`, що містить код класу з ім'ям виду `{APP_ID}`

{MODULE} Actions, успадкованого від класу waViewActions і реалізує окремі екшени, або цілий підкаталог з набором PHP-файлів, що містять логіку виконання екшенів.

Загальна схема роботи Вебасіст-додатка виглядає наступним чином:

Системний клас фреймворка FrontController знаходить відповідний модуль відповідно до вхідними даними (отриманими з GET- / POST-запитів). Назва модуля зазвичай передається у вигляді окремого параметра, наприклад: /? Module = orders & action = add (в бекенде) або за правилами маршрутизації (у фронтенді). Якщо модуль не вказано явно, то фреймворк автоматично шукає модуль backend для бекенда або frontend – для фронтенда.

Контролер модуля визначає (також на підставі вхідних параметрів), який екшен потрібно запустити, і передає йому управління. Якщо в параметрах запиту екшен не вказано, то запускається екшен за замовчуванням – default.

Після готовності програмної частини інтернет-магазину, переходимо до інтеграції її в розроблений раніше дизайн.

Файли шаблонів для екшенів повинні знаходитися в каталозі wa-apps / {APP_ID} / templates / actions / {MODULE} / і мати ім'я, сформований за правилом {MODULE} {ACTION} .html. Файл, названий згідно з цим правилом, фреймворк автоматично підключаємо після виконання коду екшену. У нашому випадку необхідно створити файл wa-apps / shop / templates / actions / backend / BackendDefault.html.

Цей приклад містить каркас HTML-коду бекенда, який можна використувати в якості основи інтерфейсу більшості додатків Вебасіст. У шаблоні доступний хелпер \$ wa, що надає наступні методи:

- \$ wa-> appName () повертає назву програми;
- \$ wa-> accountName () повертає назву облікового запису, зазвичай відображається в лівому верхньому кутку бекенда;
- \$ wa-> css () додає HTML-код для підключення загальних стилів Вебасіста (докладніше про інтерфейс Вебасіста);

- \$ wa-> js () додає HTML-код для підключення JavaScript-файлів, необхідних для роботи Вебасіста;
- \$ wa-> url () повертає URL, за яким встановлено Вебасіст;
- \$ wa-> header () додає HTML-код головного меню зі списком встановлених додатків.

Головна функція магазину це створення, редагування і видалення ключових об'єктів каталогу таких як: категорія товару, товар, інформаційна сторінка. На рис. 5.2 показана форма створення нових категорій магазину. Щоб створити категорію товару, достатньо вказати назву і URI (майбутню посилання категорії).

Створення товару, одна з найбільших форм створення об'єктів з 7 вкладками, які ділять контент на типи: основні дані про товар (тут вказуються базові, необхідності дані); текстовий опис про товар; фото і відео про товар; Характеристики товару; рекомендовані товари (альтернативні варіанти). А також додаткові сторінки для розширеного опису чого завгодно.

The image shows a web form with the title "Новая категория" (New category). It contains two input fields: "Название" (Name) and "Ссылка" (Link). The "Ссылка" field is pre-filled with the URL "https://rollki.od.ua/".

Рисунок 5.2 – Форма створення нових категорій магазину

На рис. 5.3 представлена форма заповнення основних даних, без яких товар створити неможливо. А так же форма додавання зображень товару – рис.5.4.

Основные данные [🔗] Описание и SEO [🔗] Фото и видео ⁵ [🔗] Характеристики [🔗] Услуги ⁰ [🔗] Рекомендуемые товары [🔗] Подстраницы ⁰ [🔗]

Наименование [🔗] **Роликовые кроссовки Heelys Split Plus HES10121**

✓ Опубликован на сайте [🔗] <https://rolki.od.ua/sneakers-skates/krossovki/rolikovye-krossovki-heelys-split/> [🔗] редактировать Сгенерировать ЧПУ <http://140.82.39.198/sneakers-skates/krossovki/rolikovye-krossovki-heelys-split/>

Тип товаров [🔗] **Обувь** [Изменить тип](#)

Облагается налогом [🔗] **Нет** [🔗]

Категории [🔗] **-Роликовые кроссовки** [🔗] ✕

[Дополнительная категория](#)

Списки [🔗]

На проверку GMC (PPC) (na-proverku-gmc-v) [🔗] ✕

Без обязательных х-к (spisok-bez-obyzate) [🔗] ✕

Список кеш (spisok_kesh) [🔗] ✕

[Дополнительный список](#)


Рисунок 5.3 – Форма заполнения основных данных товара

Основные данные [🔗] Описание и SEO [🔗] **Фото и видео** ⁵ [🔗] Характеристики [🔗] Услуги ⁰ [🔗] Рекомендуемые товары [🔗] Подстраницы ⁰ [🔗]


Если вы указали адрес видео на YouTube или Vimeo, но не видите этого видео на страницах товаров на витрине, вам необходимо добавить в шаблон дизайна product.html код для отображения видео (код инструкция: <https://www.shop-script.ru/help/12167/add-new-features-7/>)

Адрес видео на YouTube или Vimeo


[Выберите изображения](#) или перетащите их сюда, чтобы начать загрузку




Роликовые кроссовки Heelys Split Plus HES10121




Роликовые кроссовки Heelys Split Plus HES10121



Роликовые кроссовки Heelys Split Plus HES10121



Роликовые кроссовки Heelys Split Plus HES10121



Роликовые кроссовки Heelys Split Plus HES10121

Рисунок 5.4 – Форма добавления изображений товару

Список товара в бекенде має формат таблиці, це дуже зручно, тому що відразу видно всі необхідні, базові дані (рис.5.5).

Аналогічно виведення списку замовлень, формат таблиці відображає всі необхідні, базові дані (рис.5.6).

<input type="checkbox"/>	Код артикула	Название	Цена	В наличии	Количество изо	Бренд	Цвет	категория_тов	Приоритет
<input type="checkbox"/>	7634701	Велосипед Outleap BMX Revolt Blackg	4 999 грн.	101	1	Outleap	Черный	1026	500
<input type="checkbox"/>	5633971	Велосипед Mongoose Bmx Legion L20	6 584 грн.	54	6	Mongoose	Черный	1026	500
<input type="checkbox"/>	1600271	Велосипед Outleap BMX Clash Bronze	3 990 грн.	99	1	Outleap	Бронзовый	1026	500
<input type="checkbox"/>	1291502	Велосипед KINK BMX Launch	11 816 грн.	∞	3	Kink	Коричневый, Оранжевый, Черный	1026	500

Рисунок 5.5 – Таблица списка товаров

	#1009781	31 мая 2019 16:00	Екатерина Хоменко	4 284,90 грн.	Нова Пошта (На склад)	Приват24
	#1009789	31 мая 2019 16:00	Анастасия Цеховле	10 грн.	Нова Пошта (На склад)	Наличными (Наложный платеж в отделе)
	#1009747	30 мая 2019 14:00	Sergey_20782	6 840 грн.	не указан	не указан
	#1009714	29 мая 2019 17:00	Ирина Скляр	1 215 грн.	Самовывоз из магазина (Одесса, Ришельевская, 41) (Точка самовывоза в Одессе: ул. Ришельевс	Наличными (Наложный платеж в отделе)
	#1009673	27 мая 2019 20:00	Светлана Зубок	1 300 грн.	не указан	не указан
	#1009655	27 мая 2019 15:00	Берлинская Ирина	4 469 грн.	Нова Пошта (Адресная доставка)	Наличными (Наложный платеж в отделе)
	#1009650	26 мая 2019 22:00	Гринів Таня	4 469 грн.	Нова Пошта (На склад)	Наличными (Наложный платеж в отделе)
	#1009624	26 мая 2019 00:00	Евгений Годовани	3 999 грн.	Нова Пошта (На склад)	Приват24

Рисунок 5.6 – Таблица списку замовлень

Вибираючи замовлення можна побачити повний зміст замовлення: обраний товар, контакти клієнтів, способи доставки та оплати, статус замовлення, підсумкову суму (рис.5.7).

Заказ: **#1009781** Оплачен 31 мая 2019 14:33

Менедж Филтры сей Путинин

Ожидает отправки Отправлен Возврат **Выполнен, но в рассрочку** Выполнен

Екатерина Хоменко Новый покупатель
 furrlochka@gmail.com
 +38 (063) 730-26-93

Перезвоните мне: Да. Перезвоните мне для подтверждения заказа.

Доставка — **Нова Пошта (На склад)**
 Черноморск
 Отделение №3 (до 30 кг): ул. Спортивная (ран. Гайдара), 5
 Украина
 Город: Черноморск
 Адрес: Отделение №3 (до 30 кг): ул. Спортивная (ран. Гайдара), 5

Оплата — **Приват24**

Товар	Кол-во	Итого
Рюкзак Deuter Trans Alpine 30 (Чёрно-серый black-graphite) 7639023 Генерала Петрова	3 942 грн. × 1	3 942 грн.
Велосумка Deuter Bike bag Race I (Чёрный black) 3822181 Генерала Петрова	342,90 грн. × 1	342,90 грн.
Подытог		4 284,90 грн.
Скидка		- 0 грн.

Версия для печати
 Редактировать заказ
 Написать клиенту
 Изменить параметры доставки
 Создать задачу
Выставить счет р/с

Генерация ТТН
 Генерация ТТН СА
 Печать

Заказ создан: 31 мая 2019 16:50
 Референт: android.
 app://com.google.android.am
 Канал продаж: Витрина
 Витрина: rollid.od.ua
 IP-адрес: 97.73.182.25

Рисунок 5.7 – Зміст замовлення

Таким чином, в кваліфікаційній роботі представлено обґрунтування вибору засобів розробки інтернет-магазину і системи керування базами даних MySQL. Докладно описано етапи розробки дизайну сайту і верстку макету. Наведена структура бази даних інформаційної системи. Крім того, в роботі надано опис основних функціональних можливостей інтернет-магазину. Отже, якщо розглядати доцільність написання додатку «інтернет-магазин» з точки зору електронної комерції сміло можна робити висновок, що такі проекти є цілком виправданими. Такий додаток має широке розповсюдження в мережі, оскільки створення власного інтернет-магазину це спосіб для фізичної кампанії збільшити обсяг продажів, за рахунок масового розповсюдження мережі інтернет.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була розроблена інформаційно-пошукова система комерційного підприємства по продажу спортивних товарів «Roliki», яка представлена у вигляді Інтернет-магазину на базі CMS Webasyst.

В роботі було надано обґрунтування вибору засобів розробки інтернет-магазину і системи керування базами даних MySQL. Докладно описано етапи розробки дизайну сайту і верстку макету. Наведена структура бази даних інформаційної системи. Крім того, в роботі надано опис основних функціональних можливостей інтернет-магазину: реєстрація та авторизація користувачів, додавання та видалення товарів у/з кошику, виконання замовлення, можливості пошуку товарів та навігації по сайту.

Якщо зовсім просто описувати інтернет додаток, то це набір запитів до бази даних, що оброблюється та виконується завдяки серверній мові програмування PHP. Однак у ньому було б мало сенсу, якби робилось це статично. Налаштування та стилізація оболонки, а також написання сценаріїв із можливістю приймати дані від користувача, що впершу чергу спрощує його стосунки із БД – це все комплекс заходів для створення гарного додатку, який може бути корисним. А безперечна користь додатку свідчить про доцільність його створення.

Отже, якщо розглядати доцільність написання додатку «інтернет-магазин» з точки зору електронної комерції сміло можна робити висновок, що такі проекти є цілком виправданими. Такий додаток має широке розповсюдження в мережі, оскільки створення власного інтернет-магазину це спосіб для фізичної компанії збільшити обсяг продажів, за рахунок масового розповсюдження мережі інтернет.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. HTML: HyperText Markup Language. URL: <http://uk.wikipedia.org/wiki/HTML> (дата звернення 18.05.2019)
2. Комолова Н.В., Яковлева Е.С. HTML, XHTML и CSS. СПб.: Питер, 2012. 300 с.
3. Фрэйлн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. СПб.: Питер, 2014. 298 с.
4. CSS: Cascading Style Sheets. URL: <http://uk.wikipedia.org/wiki/CSS> (дата звернення 18.05.2019)
5. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. М.: Эксмо, 2013. 480 с.
6. Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web- сайтов. Спб.: БХВ-Петербург, 2011. 416 с
7. JavaScript. URL: <http://uk.wikipedia.org/wiki/JavaScript> (дата звернення 18.05.2019)
8. Гаевский А.Ю. Создание Web-страниц и Web-сайтов: HTML и JavaScript. М.: Триумф, 2008. 454 с.
9. JavaScript: Онлайн підручник. URL: <http://www.wisdomweb.ru/JS/javascript-first.php> (дата звернення 18.05.2019)
10. jQuery. URL: <http://uk.wikipedia.org/wiki/JQuery> (дата звернення 18.05.2019)
11. jQuery: Онлайн підручник URL: <http://www.wisdomweb.ru/JQ/jquery-first.php> (дата звернення 18.05.2019)
12. PHP: Hypertext Preprocessor. URL: <http://uk.wikipedia.org/wiki/PHP> (дата звернення 18.05.2019)
13. Справочник по PHP. URL: <http://www.php.su> (дата звернення 18.05.2019)
14. Зандстра М. PHP: объекты, шаблоны и методики программирования. Издательский дом «Вильямс», 2011. 560 с.

15. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.: Питер, 2013. 496 с.
16. SQL: Structured Query Language. URL:<http://uk.wikipedia.org/wiki/SQL> (дата звернення 18.05.2019)
17. Справочное руководство по MySQL. URL:<http://www.mysql.ru/docs/man/> (дата звернення 18.05.2019)
18. Шварц Б. MySQL. Оптимизация производительности. СПб.: Символ-Плюс, 2010. 832 с.
19. AJAX: Asynchronous Javascript and XML. URL: <http://uk.wikipedia.org/wiki/AJAX> (дата звернення 18.05.2019)
20. Что такое Bootstrap. URL:<http://blogwork.ru/chto-takoebootstrap/> (дата звернення 21.05.2019)
21. Центр управления бизнесом – Webasysy. Офіційний сайт. URL: <https://www.webasyst.ru/> (дата звернення 22.05.2019)

ДОДАТОК А

Файловая структура Webasyst

- wa-apps
 - исходный код приложений
 - *APP_ID*
 - api
публичные API-методы приложения (необязательно)
 - css
CSS-стили приложения
 - img
файлы изображений
 - js
файлы с JavaScript-кодом
 - lib
файлы с основным PHP кодом приложения
 - locale
файлы локализации приложения (gettext)
 - plugins
плагины приложения (необязательно)
 - *PLUGIN_ID*
 - css
CSS-стили плагина
 - img
файлы изображений
 - js
файлы с JavaScript-кодом
 - lib
файлы с основным PHP-кодом плагина
 - locale
файлы локализации плагина (gettext)
 - templates
файлы HTML-шаблонов плагина
 - templates
файлы HTML-шаблонов бекенда приложения
 - themes
темы дизайна для фронтенда приложения
 - *THEME_ID*
 - theme.xml
файл-манифест темы дизайна
 - index.html
основной шаблон
 - ...
 - другие шаблоны темы дизайна
 - widgets
виджеты приложения (необязательно)
 - *WIDGET_ID*
 - css
CSS-стили виджета

- img
файлы изображений
 - js
файлы с JavaScript-кодом
 - lib
файлы с основным PHP-кодом плагина
 - locale
файлы локализации виджета (gettext)
 - templates
файлы HTML-шаблонов виджета
- wa-cache
файлы кеша: SQL-запросы, скомпилированные шаблоны веб-страниц и т. п.
- wa-config
системные конфигурационные файлы
 - apps.php
список установленных приложений
 - db.php
параметры подключения к базе данных
 - routing.php
правила маршрутизации HTTP-запросов по URL
- wa-content
общее содержимое, подключаемое на страницах разных приложений: библиотеки JavaScript, CSS-стили, изображения и т. п.
- wa-data
файлы, хранящие данные, необходимые для работы приложений
 - public
файлы, доступные напрямую по HTTP
 - protected
защищенные файлы, доступные только с помощью PHP-скриптов приложений
- wa-installer
установочные скрипты фреймворка
- wa-log
лог-файлы
- wa-plugins
общесистемные плагины, например, для подключения к платежным системам, плагины расчета стоимости доставки, модули интеграции с SMS-шлюзами
- wa-system
файлы с исходным кодом ядра фреймворка
- wa-widgets
файлы с исходным кодом системных виджетов
- api.php
точка доступа к публичным API
- cli.php
обработчик запросов через командную строку (CLI)
- index.php
центральный диспетчер HTTP-запросов
- install.php
скрипт установки фреймворка (автоматически удаляется после завершения установки)

- wa.php
скрипт для выполнения служебных команд разработчиками