

РЕФЕРАТ

Пояснювальна записка: ___ с., ___ рис., ___ табл., ___ дод., ___ джерел.

Об'єкт розробки: веб-сайт для забезпечення роботи спортивного комплексу.

Мета кваліфікаційної роботи: розробка веб-орієнтованого програмного забезпечення для спортивного комплексу «Аквармарин», яке надаватиме різну інформацію про послуги комплексу та дозволить адмініструвати його роботу.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано платформу для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування програми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження програми, описана робота програми.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню програми та розраховано час на його створення.

Практичне значення полягає у створенні програмного додатка, що надає можливість зацікавленому користувачу отримати повну інформацію про спортивний комплекс та надавані ним послуги, можливість онлайн запису для відвідування секцій, а також спрощуватиме адміністрування роботи закладу.

Актуальність даного програмного продукту визначається великим попитом на подібні розробки, через те, що веб-сайти допомагають значно збільшити продажі, будучи головним інструментом для вирішення різних маркетингових завдань і значно знизити навантаження на офісних працівників, відповідаючи клієнтам на рутинні питання, такі як розташування, список послуг тощо. Також значно знижується документообіг за рахунок наявності на веб-сайті особистого кабінету для діючих клієнтів і форм зворотного зв'язку для онлайн заявок.

Список ключових слів: ВЕБ-ДОДАТОК, САЙТ, ПРОЕКТУВАННЯ, БАЗА ДАНИХ, ШАБЛОН, ПРОГРАМУВАННЯ, СПОРТИВНИЙ КОМПЛЕКС, РОЗКЛАД.

ABSTRACT

Explanatory note: ___ pp., ___ fig., ___ table, __ appendix, ___ sources.

Object of development: a website to ensure the operation of the sports complex.

The purpose of the qualification work: development of web-oriented software for the sports complex "Aquamarine", which will provide various information about the services of the complex and allow you to administer its work.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development is defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes existing solutions, selects a platform for development, designs and develops the program, describes the algorithm and structure of the program, defines input and output data, provides characteristics of the parameters of hardware, describes the call and download of the program, describes the program.

The economic section determines the complexity of the developed information system, calculates the cost of work to create a program and calculates the time for its creation.

The practical significance is to create a software application that allows the interested user to get complete information about the sports complex and its services, the ability to register online to visit the sections, as well as simplify the administration of the institution.

The relevance of this software product is determined by the high demand for such developments, due to the fact that websites help to significantly increase sales, being the main tool for various marketing tasks and significantly reduce the burden on office workers, answering customers routine questions such as location, list services, etc. There is also a significant reduction in paperwork due to the presence on the website of a personal account for current customers and feedback forms for online applications.

Keywords: WEB APP, SITE, DESIGN, DATABASE, TEMPLATE, PROGRAMMING, SPORTS COMPLEX, SCHEDULE.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІС – інформаційна система;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

НТТР – HyperText Transfer Protocol, протокол передачі гіпертексту;

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. Технології створення сайтів.....	10
1.1.2. Технічне супроводження веб-сайтів.....	15
1.2. Призначення розробки та галузь застосування.....	17
1.3. Підстава для розробки.....	17
1.4. Постановка завдання.....	18
1.5. Вимоги до програми або програмного виробу.....	18
1.5.1. Вимоги до функціональних характеристик.....	18
1.5.2. Вимоги до інформаційної безпеки.....	20
1.5.3. Вимоги до складу та параметрів технічних засобів.....	20
1.5.4. Вимоги до інформаційної та програмної сумісності	21
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	22
2.1. Функціональне призначення системи	22
2.2. Опис застосованих математичних методів.....	22
2.3. Опис використаних технологій та мов програмування.....	22
2.3.1. Розробка шаблону веб-сайту.....	23
2.3.2. Програмування на мові PHP.....	26
2.3.3. Засоби розробки бази даних.....	32
2.3.4. Вибір операційної системи.....	32
2.4. Опис структури програми та алгоритмів її функціонування ...	37

2.4.1. Розробка контенту веб-сторінки.....	37
2.4.2. Розробка бази даних системи.....	48
2.4.3. Опис фізичної структури проєкту.....	55
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	56
2.6. Опис розробленої системи	56
2.6.1. Використані технічні засоби.....	56
2.6.2. Використані програмні засоби.....	58
2.6.3. Виклик та завантаження програми.....	58
2.6.4. Опис інтерфейсу користувача.....	58
2.6.4.1.Опис веб-сайту.....	58
2.6.4.2.Тестування додатку.....	78
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....	81
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту.....	81
3.2. Розрахунок витрат на створення програми.....	84
ВИСНОВКИ.....	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
Додаток А. Код програми.....	89
Додаток Б. Відгук керівника економічного розділу.....	108
Додаток В. Перелік файлів на диску.....	109

ВСТУП

Інтернет, як інструмент комунікації, який не має територіальних кордонів, дозволяє обмінюватися різними видами інформації.

В останні роки Інтернет виконує величезний вплив на розвиток українських компаній, змінюючи способи їх подання перед потенційними клієнтами, а також дозволяє дистанційно надавати різноманітні послуги вже існуючим клієнтам. Кількість людей які використовують Інтернет, як найважливіший засіб для отримання необхідних відомостей про послуги, що надаються, значно зросла останнім часом. Інтернет допомагає в зростанні бізнесу як великим учасникам ринку так і маленьким підприємцям.

Щоб представляти свої послуги в мережі Інтернет компаніям необхідно обзавестися особистим веб-сайтом. Він буде як веб-візитка для отримання клієнтами контактних даних та подання, в якій сфері послуг працює дана компанія. Наявність власного веб-сайту позитивно позначається на іміджі компанії, клієнти ставляться до таких компаній з більшою довірою. Також веб-сайти допомагають значно збільшити продажі, будучи головним інструментом для вирішення різних маркетингових завдань і значно знизити навантаження на офісних працівників, відповідаючи клієнтам на рутинні питання, такі як розташування, список послуг тощо. Також значно знижується документообіг за рахунок наявності на веб-сайті особистого кабінету для діючих клієнтів і форм зворотного зв'язку для онлайн заявок.

З кожним роком посилюється роль Інтернету у житті кожної людини, від малої дитини до людей похилого віку. Сьогодні інтернет-користувач має необмежені можливості в "світовій павутині". Сайт-найсучасніший та достатньо ефективний спосіб комунікації з потенційним клієнтом. Тільки можливості Інтернет-сайту дозволяють повною мірою використовувати всі способи донесення інформаційного повідомлення до користувача. Текст, аудіо та відео інформація можуть бути подані в рамках одного сайту для ознайомлення споживача з ними.

Отже, сучасний інтернет-сайт це інформаційний центр будь-якої компанії, не виключенням є і спортивні комплекси, розвиток яких безпосередньо залежить від кількості залучених клієнтів.

Для розробки програмного забезпечення інтернет-магазину у наш час існують готові інструментально-програмні засоби, які дозволяють ефективно і у більш короткий термін створювати подібні проєкти, однак вони потребують значних зусиль для створення конкретної системи.

Метою кваліфікаційної роботи є розробка веб-орієнтованого програмного забезпечення для спортивного комплексу «Аквамарин», яке надаватиме різну інформацію про послуги комплексу та дозволить адмініструвати його роботу.

Для досягнення мети слід розв'язати такі задачі:

- поглиблено вивчити технологію створення веб-сайтів;
- проаналізувати сферу діяльності спортивних комплексів;
- порівняти наявні у мережі сайти-аналоги;
- висунути функціональні вимоги до власного програмного забезпечення;
- спроектувати й створити власний інформаційний продукт.

Результатом виконання даної роботи є повністю роботоспроможний додаток, який надаватиме зацікавленому користувачу повну інформацію про спортивний комплекс та надавані ним послуги, можливість онлайн запису для відвідування секцій, а також спрощуватиме адміністрування роботи закладу.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Технології створення сайтів

Одним з ключових моментів в розвитку всесвітньої павутини грає веб-розробка - процес створення веб-сайта або веб-додатку. Термін включає розробку додатків електронної комерції, веб-дизайн, програмування для веб на стороні клієнта і серверу, а також конфігурування веб-серверу. Основними етапами веб-розробки є:

- проектування сайту або веб-додатку;
- створення макетів сторінок;
- наповнення;
- обслуговування працюючого сайту або його програмної основи.
- подальше просування сайту в мережі та підняття його рейтингу

Існуючі технології для створення сайтів:

1. Ручний метод (HTML, CSS). HTML (англ. HyperText Markup Language - мова розмітки гіпертекстових документів) - стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді. У більшості випадків автор документа суворо визначає зовнішній вигляд документа. У разі HTML читач, ґрунтуючись на можливостях Web-браузера, може, певною мірою, керувати зовнішнім виглядом документа (але не його вмістом). HTML дозволяє відзначити, де в документі повинен бути заголовок або абзац за допомогою тега HTML, а потім надає Web-браузеру інтерпретувати ці теги. Наприклад, один Web-браузер може розпізнавати тег початку абзацу і представляти документ у потрібному вигляді, а інший не має такої можливості і надає документ в один рядок.

Користувачі деяких Web-браузерів мають, також, можливість

налаштовувати розмір і вид шрифту, колір та інші параметри, що впливають на відображення документа. [17] HTML - теги можуть бути умовно розділені на дві категорії:

- теги, що визначають, як буде відображатися Web-браузером тіло документа в цілому;
- теги, що описують загальні властивості документа, такі як заголовок чи автор документа. HTML-документи можуть бути створені за допомогою будь-якого текстового редактора або спеціалізованих HTML-редакторів і конвертерів.

Вибір редактора, який буде використовуватися для створення HTML-документів, залежить виключно від поняття зручності і особистих пристрастей кожного автора. Наприклад, HTML редактори, такі, як "Netscape Navigator Gold" компанії Netscape, дозволяють створювати документи графічно з використанням технології WYSIWYG (What You See Is What You Get). З іншого боку, більшість традиційних засобів для створення документів мають конвертери, що дозволяють перетворювати документи до формату HTML.

Основна перевага HTML полягає в тому, що ваш документ може бути переглянутий на Web-браузерах різних типів і на різних платформах.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) - спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. Таблицю стилів CSS можна вмонтувати прямо в HTML -сторінку - це внутрішня таблиця стилів. Або ж її можна створити в окремому файлі, і вже потім приєднати посилання на нього до потрібної HTML - сторінки - це зовнішня таблиця стилів.

Зовнішню таблицю необхідно підключити до основного HTML документу за допомогою спеціальних тегів: `< link rel="stylesheet" type="text/css" href="/style.css >`, де style.css - це ім'я файлу, що містить таблицю CSS. Завдяки цьому, стиль, описаний у зовнішній таблиці CSS, можна використовувати

повторно скільки завгодно разів. [6]. Роблячи висновки з попередньої інформації можна зробити висновки, що використання HTML та CSS є зручним та не складним у використанні, але має ряд недоліків таких як:

- обмежені можливості при створенні сайту;
- відсутність динамічних елементів;
- велика кількість коду.

2. Використання інструментальних систем FrontPage , Dreamweaver та інших. На сьогодні найпопулярнішою системою розробки Web-сайтів являється Dreamweaver - HTML-редактор. Розроблений і підтримувався компанією Macromedia у версіях до MX (6 версія, 2005 рік), а нині – Adobe. Багатий інструментарій, відвертість додатку для всіляких налаштувань, зручний інтерфейс і інші особливості зробили Dreamweaver одним з найпопулярніших HTML-редакторів в світі. Для деяких версій Dreamweaver виконувалися переклади довідкових систем на російську мову. Dreamweaver включає багато інструментів і засобів для редагування і створення професійного сайту: HTML, CSS, JavaScript, відладчик JavaScript, редактори коду (перегляд коду і інспектор коду), що дозволяє вам, редагувати JavaScript, XML і інші текстові документи, які підтримуються в Dreamweaver. Технологія Roundtrip HTML імпортує документи HTML без потреби переформатувати код і ви можете налаштувати Dreamweaver, щоб він наводив лад і переформатовував HTML як ви цього хочете.

Можливість візуального редагування в Dreamweaver також дозволяє вам швидко створювати або міняти дизайн проекту без написання коду. Ви можете розглянути всі ваші централізовані елементи і перетягнути їх із зручної панелі безпосередньо в документ. Microsoft FrontPage це редактор HTML, призначений для створення веб-сторінок і сайтів без необхідності знання користувачем мови HTML. Окрім цього, програму використовують для розташування сайтів на веб-сервері та їхньої наступної експлуатації.

Користувач має змогу створити окрему сторінку чи сайт двома способами: з нуля, або скориставшись шаблоном з колекції програми FrontPage. FrontPage дозволяє створювати веб-сторінки так само, як документ в Microsoft Word. Крім

того, він додає функції, такі як управління проектами. Основні функції: Підтримка XML і підтримки сценаріїв, повне управління веб-сайтом, розширена підтримка CSS, вбудований валідатор, підтримка баз даних, а також WYSIWYG і кольором редагування XHTML. Microsoft FrontPage має такі особливості:

- HTML-валідатор;
- текстовий редактор;
- редактор WYSIWYG;
- FTP і веб-сайт менеджер;
- розширений пошук і заміна;
- редагування декількох файлів;
- перевірка посилань;
- підтримка спеціальних символів.

В програмі виявлені наступні недоліки: надмірність і некоректність коду, орієнтація на технології IE.

HomeSite - редактор HTML який належить Adobe Systems (до цього він належав Macromedia). HomeSite це не WYSIWYG HTML редактор як FrontPage чи Dreamweaver, а який працює з кодом HTML. HomeSite призначений для редагування, або для «ручного кодування» на HTML, та на інших мовах вебсайтів. Він доступний для Windows платформ. Можливості:

- інтерфейс, що налаштовується та включає панель інструментів;
- підсвічування синтаксису коду ASP, CFML, CSS, HTML, Java, JavaScript, JSP, Perl, PHP, SQL, VBScript, VTML і XHTML ;
- керування проектами та вбудований FTP для керування та завантаження вебсайту;
- вбудований браузер.

3. Використання CMS. Також останнім часом стали дуже популярні системи CMS - система управління вмістом або система управління контентом (англ. Content management system, CMS) - комп'ютерна програма, що використовується для управління вмістом чого-небудь (звичайно цей вміст розглядається як неструктуровані дані наочної задачі в протилежність структурованим даним,

звичайно що знаходяться під управлінням СУБД (Система Управління Базами Даних)). Звичайно такі системи використовуються для зберігання і публікації великої кількості документів, зображень, музики або відео. [1]

Нижче в алфавітному порядку представлені приклади CMS. Наявність в цьому списку не указує на популярність системи:

- Drupal;
- Etomite CMS;
- eZ publish;
- Irokez;
- Joomla;
- Mambo;
- mojoPortal;
- WordPress.

CMS рішень багато, але при виборі варто спиратися на наступні можливості:

- призначений для користувача сервіс – наявність тих або інших функцій і модулів, зрозумілість і доступність користувачу;
- технологічність – використання технологій, що дозволяють підвищити пропускну спроможність і надійність системи;
- сумісність (апаратна і програмна) – можливість функціонування системи на різних платформах, сумісність з СУБД, можливість підключення додаткових модулів;
- масштабованість – можливість розвитку і нарощування системи.

4. Joomla - відкрита універсальна система керування вмістом для публікації інформації в інтернеті. Підходить для створення маленьких і великих корпоративних сайтів, інтернет порталів, онлайн-магазинів, сайтів спільнот і 22 персональних сторінок. З особливостей Joomla можна відзначити: гнучкі інструменти з управління обліковими записами, інтерфейс для управління медіа-файлами, підтримка створення багатомовних варіантів сторінок, система управління рекламними кампаніями, адресна книга користувачів, голосування,

вбудований пошук, функції категоризації посилань і обліку кліків, WYSIWYG-редактор, система шаблонів, підтримка меню, управління новинними потоками, XML-RPC API для інтеграції з іншими системами, підтримка кешування сторінок і великий набір готових доповнень.

Joomla написана на мові PHP з використанням архітектури MVC. Для збереження інформації використовується база даних MySQL, PostgreSQL чи MS SQL. CMS Joomla містить різні інструменти для розробки Веб-сайту. Особливістю системи є мінімальний набір інструментів при початковій установці, який розширюється в разі необхідності. Завдяки цьому знижується рівень навантаження на сервер і економиться місце на хостингу.

CMS «Joomla!» має такі особливості:

- багаторівневий доступ зареєстрованих користувачів, як до адміністративної частини так і до фронтальної частини сайту;
- понад 8 тисяч модулів і компонентів (останні новини, лічильник відвідувань, гостьова книга, форум тощо);
- можливість писати власні компоненти, модулі, плагіни і шаблони або редагувати вбудовані;
- наявність менеджера розсилки новин;
- редагування матеріалів за допомогою Візуального редактора TinyMCE (аналог текстових редакторів, наприклад Word);
- вбудована багатомовність, починаючи з версії 1.6;
- робота системи Linux, FreeBSD, WinNT, Win2K, MacOS X, Solaris, AIX, SCO та інших.

1.1.2. Технічне супроводження веб-сайтів

Технічне супроводження сайту – невід’ємна складова його стабільної та швидкої роботи. Технічна підтримка значно зменшує витрати на обслуговування Web-сайту або Web-порталу штатними спеціалістами, зменшує ризики простою через ураження вірусами, атаки зловмисників чи помилки людей, що працюють з

ним. Технічна підтримка передбачає проведення комплексу заходів по технічному супроводженню системи керування сайтом, резервному копіюванню, наданню консультацій та виконанню додаткових робіт, що стосуються нового чи вже вбудованого функціоналу сайту. Розглянемо функції замовника та розробників Web-сайту. Замовник є одною з найбільш важливих ланок розробки. Він, по-перше, вирішує задачу обґрунтування необхідності сайту, по-друге, повинен мати інформаційне наповнення сайту, пропозиції по його структурі та графічному оформленню, і що не менш важливо, так це фінансові ресурси на розробку та супроводження сайту.

Замовник це ланцюг розробки, який приймає чи повинен приймати участь майже в усіх етапах розробки сайту. Він визначає і назви розділів, і можливе скорочення назв, і першорядні і другорядні розділи та інше. Якість сайту теж суттєво залежить від замовника. Сайт не повноцінний, якщо є не повним інформаційне наповнення. Сайт вмирає, якщо він не супроводжується, а ця функція звичайно лягає на замовника та його спеціалістів. Фінансує і контролює роботу Webмайстра замовник.

Веб-майстер – людина, яка займається розробкою веб-сайтів. В залежності від розміру колективу розробників, він може об'єднати в собі функції автора, дизайнера, верстальника, програміста, Дані спеціалісти розробляють і підтримують невеликі сайти. Функції Web-майстра можна поділити на:

- визначення цілі та задачі проекту;
- розробка інтерфейсу веб-серверу;
- здійснення адміністрацію веб-сервера;
- забезпечення безпеку веб-сервера ;
- здійснення аналізу трафіку (статистику відвідування).

Обов'язки веб-адміна:

- сприяє розповсюдження сайту в інтернеті;
- розробляє і підтримує основну концепцію сайту, вносить пропозиції по доробці, по введенні нових сервісів для відвідувачів сайту;
- слідкує за текстовим наповненням сайту, постійним оновлення

інформації;

- здійснює контроль за виконанням правил користувачами;
- бере участь в оформленні інформації, що розміщується на сайт разом із веб-майстром.

1.2. Призначення розробки та галузь застосування

Метою кваліфікаційної роботи є розробка веб-орієнтованого програмного забезпечення для спортивного комплексу «Аквамарин», яке надаватиме різну інформацію про послуги комплексу та дозволитиме адмініструвати його роботу.

Веб-сайт призначений для того, щоб значно збільшити продажі, будучи головним інструментом для вирішення різних маркетингових завдань і значно знизити навантаження на офісних працівників, відповідаючи клієнтам на рутинні питання, такі як розташування, список послуг тощо. Також через впровадження даного програмного забезпечення значно знижується документообіг за рахунок наявності на веб-сайті особистого кабінету для діючих клієнтів і форм зворотного зв'язку для онлайн заявок.

Результатом виконання даної роботи є повністю роботоспроможний додаток, який надаватиме зацікавленому користувачу повну інформацію про спортивний комплекс та надавані ним послуги, можливість онлайн запису для відвідування секцій, а також спрощуватиме адміністрування роботи закладу.

1.3. Підстава для розробки

Підставою для розробки кваліфікаційної роботи бакалавра на тему «Розробка веб-орієнтованого програмного забезпечення для спортивного комплексу засобами мов Html, Css, JavaScript та PHP» є наказ по Національному технічному університету «Дніпровська політехніка» від __.__. 2021р. № ____-__.

1.4. Постановка завдання

Завдання даної кваліфікаційної роботи є розробка веб-орієнтованого програмного забезпечення для спортивного комплексу, застосувавши засоби мов програмування Html, Css, JavaScript та PHP. Для цього необхідно спроектувати і розробити веб-сторінку для спортивного комплексу «Аквамарин».

Розроблювана веб-сторінка повинна забезпечувати представлення різноманітної інформації про спортивний комплекс у мережі Інтернет клієнтам та звичайним користувачам а також забезпечувати адміністрування роботи закладу.

Для досягнення мети слід розв'язати такі задачі:

- поглиблено вивчити технологію створення веб-сайтів;
- проаналізувати сферу діяльності спортивних комплексів;
- порівняти наявні у мережі сайти-аналоги;
- висунути функціональні вимоги до власного програмного забезпечення;
- спроектувати й створити власний інформаційний продукт.

Результатом виконання даного проекту повинен бути повністю роботоспроможний веб-додаток для організації роботи спортивного закладу. Програмний додаток повинен бути протестований та перевірений на робото спроможність.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Сайт матиме дворівневий доступ: адміністратор та користувач, для кожного з яких передбачені відповідні функціональні можливості.

Рівень «Користувач» матиме змогу:

- переглядати всю інформацію яка розміщена на веб-сторінці спортивного комплексу;

- за допомогою додаткової форми задавати питання, які його цікавлять;
- за допомогою додаткової форми користувач може залишати відгуки;
- за допомогою додаткової форми користувач може записатися на заняття.

Рівень «Адміністратор» матиме наступні привілеї:

- переглядати всю інформацію, яка розміщена на веб-сторінці спортивного комплексу;
- давати відповіді на питання користувачів, а також відповідати на відгуки;
- зможе бачити та редагувати інформацію стосовно того, який користувач на яку секцію записався.

Розроблений інформаційний сайт повинен задовольняти наступним вимогам:

1. Верхній інформаційний рядок повинен містити такі елементи:

- логотип сайту: назва спортивного комплексу з відповідним зображенням;
- гіперпосилання на інші сторінки сайту;
- фото із зображенням відповідного до теми.

2. В центральній частині сайту розташовано:

- основна інформація про спортивний комплекс;
- слайдер, де зображено усіх тренерів які працюють у спортивному комплексі.

3. В нижній частині сайту повинні знаходитись наступні елементи:

- слайдер з тренажерними залами і повинно бути гіперпосилання на відповідну сторінку;
- слайдер із SPA залами і повинно бути гіперпосилання на відповідну сторінку .

На головній сторінці передбачити наступні вкладки:

- «Главная» - посилання на головну сторінку із загальною інформацією про спортивний комплекс;
- «Тренажерный зал» - міститиме основну інформацію про тренажерний зал, розклад, прайс, фотогалерея, а також форма для запису;
- «Фитнес» - міститиме основну інформацію про фітнес зал, розклад, прайс, фотогалерея, а також форма для запису;
- «СПА» - міститиме основну інформацію про SPA зал, розклад, прайс, фотогалерея, а також форма для запису;
- «Новости» - міститиме інформацію про актуальні новини з життя спортивного комплексу;
- «Контакты» -міститиме форму для того, щоб задати питання;
- «Карта» - місце знаходження спортивного комплексу
- форму для відгуків.

1.5.2. Вимоги до інформаційної безпеки

Для надійної роботи додатку необхідно:

- використовувати ліцензійне програмне забезпечення на сервері;
- здійснювати захист від вірусів на сервері;
- здійснювати захист від несанкціонованого доступу;
- застосовувати на сервері джерело безперебійного живлення для захисту від перепадів напруги або збоїв у живленні;
- здійснювати контроль даних, що вводяться користувачем.

1.5.3. Вимоги до складу та параметрів технічних засобів

Дане програмне забезпечення розроблюється для ПК, який матиме наступні мінімальні технічні характеристики:

1. Частота процесора: не менше 1 ГГц.
2. Оперативна пам'ять: не менше 1024 Мб.
3. Вільне місце на жорсткому диску: не менше 500 Мб.
4. Роздільна здатність екрану не менше 1280x1024.
5. Наявність засобів вводу (миша, клавіатура);
6. Мережева карта Ethernet BNC/UTP 10/100 Мбіт/с.
7. Доступ до Інтернету.

1.5.4. Вимоги до інформаційної та програмної сумісності

Розроблений ресурс повинний коректно працювати з усіма версіями операційної системи Microsoft Windows.

Для роботи програми необхідно дотриматися наступних програмних вимог:

1. Операційна система: Windows 7/8/8.1/10.
2. Встановлений фреймворк .NET 4.0 або вище.
3. Остання версія Adobe Flash Player.
4. Один з перерахованих веб-браузерів: IE 5.5 і вище, Firefox 0.8 і вище, Safari 1.2.4 і вище, Netscape 7.1 і вище, Mozilla 1.4 і вище, Opera 7 і вище.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Завдання даної кваліфікаційної роботи є розробка веб-орієнтованого програмного забезпечення для спортивного комплексу, застосувавши засоби мов програмування Html, Css, JavaScript та PHP. Для цього спроектовано і розроблено веб-сторінку для спортивного комплексу «Аквамарин».

Розроблювана система забезпечує представлення різноманітної інформації про спортивний комплекс у мережі Інтернет клієнтам та звичайним користувачам а також забезпечує адміністрування роботи закладу.

Веб-сайт має дворівневий доступ: адміністратор та користувач, для кожного з яких передбачені відповідні функціональні можливості.

Результатом виконання даного проекту є повністю роботоспроможний web-додаток для організації роботи спортивного закладу.

2.2. Опис застосованих математичних методів

Під час проєктування та розробки даної кваліфікаційної роботи ніякі математичні методи не застосовувалися.

2.3. Опис використаних технологій та мов програмування

Для реалізації даного завдання було обрано веб-сторінку написану на html, css , java script та мову PHP. Веб-розробка поділяється на роботу на стороні браузера (верстка, фронтенд) та на стороні сервера (бекенд + бази даних).

2.3.1. Розробка шаблону веб-сайту

Для створення веб-сайтів існує безліч різних програм. Одні з них є редакторами html, інші - редакторами серверних або клієнтських скриптів, а треті - редакторами таблиць css. Всі вони найчастіше візуалізують процес редагування.

Кожний веб-сайт складається з веб-сторінок, кожна з яких, у свою чергу, складається з коду мовою html, що відповідає за оформлення сторінки, і власне вмісту. Тому редактори html - програми, створення сайту без яких неможливо.

Для створення веб-сайту було обране середовище розробки Microsoft FrontPage[1].

Microsoft FrontPage - html-редактор з пакета MS Office. Microsoft FrontPage є візуальним редактором (WYSIWYG - What You See Is What You Get - що бачиш, то й одержиш), але в Microsoft FrontPage код можна також редагувати вручну. Інтерфейс Microsoft FrontPage схожий на інтерфейс Microsoft Word, звичний для більшості користувачів, тому дуже зручний для новачків у веб-дизайні. В Microsoft FrontPage для роботи доступні три режими: Normal, HTML і Preview.

До переваг цього середовища розробки відносить те, що є можливість відразу бачити, як буде виглядати веб-сторінка, наявність підказок і підсвічування тегів, можливість легкої й зручної вставки різних об'єктів у сторінку без знань мови html. Недоліком же візуальних редакторів є їхня вартість - гарні візуальні html-редактори платні. До того ж багато візуальних редакторів створюють «зайвий» код, що «поважчує» сторінку.

Hypertext Markup Language (HTML) (укр. мова розмітки гіпертекстових документів) стандартна мова розмітки для створення веб сторінок і веб-додатків. З Cascading Style Sheets (CSS) і JavaScript, вона утворює триаду основних технологій для World Wide Web.

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує

структуру веб-сторінки семантично і спочатку включає сигнали для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами , написаними з використанням кутових дужок . Теги, такі як і безпосередньо вводять вміст на сторінку. Інші теги, такі як ```<input />``<p>` оточують і надають інформацію про текст документа і можуть включати інші теги як під-елементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компонування вмісту. World Wide Web Consortium (W3C), які супроводжують як HTML і CSS стандартів, заохочує використання CSS над явним презентаційним HTML з 1997 року.

HTML впроваджує засоби для:

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

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) це спеціальна мова стилю сторінок, що використовується для опису їх зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки - розділення змісту сторінки (даних) та їхньої візуальної презентації.

JavaScript (JS)-динамічна, об'єктноорієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme[1].

2.3.2. Програмування на мові PHP

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

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

Мова PHP, завдяки своїм особливостям, широко використовується при розробці динамічних веб-сайтів. До основні властивості мови можна віднести:

- автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-серверу в зумовлені масиви;
- файлові функції успішно обробляють як локальні, так і віддалені файли;
- фвтоматична відправка HTTP-заголовків;
- робота з cookies і сесіями;
- обробка файлів, що завантажуються на сервер;
- робота з HTTP заголовками і HTTP авторизацією;
- робота з віддаленими файлами і сонетами.

Принцип роботи PHP полягає в наступному. Програма інтерпретується веб-сервером в HTML-код, Який передається на сторону клієнта. На відміну від таких скриптових мов програмування, як JavaScript, користувач не має доступу до PHP-коду, що є переваги з точки зору безпеки але значне погіршує

інтерактивність сторінок. Але ніщо не забороняє використовувати PHP для генерування і JavaScript-кодів, які виконуються вже на стороні клієнта (рис.2.1).

PHP - мова, яка може бути вбудована безпосередньо в html-код сторінок, які, в свою чергу коректно будуть оброблені PHP -інтерпретатором. Механізм PHP просто починає виконувати код після першої екрануючої послідовності (<?) і продовжує виконання до того моменту, коли він зустрине парну екрануючу послідовність (?>).

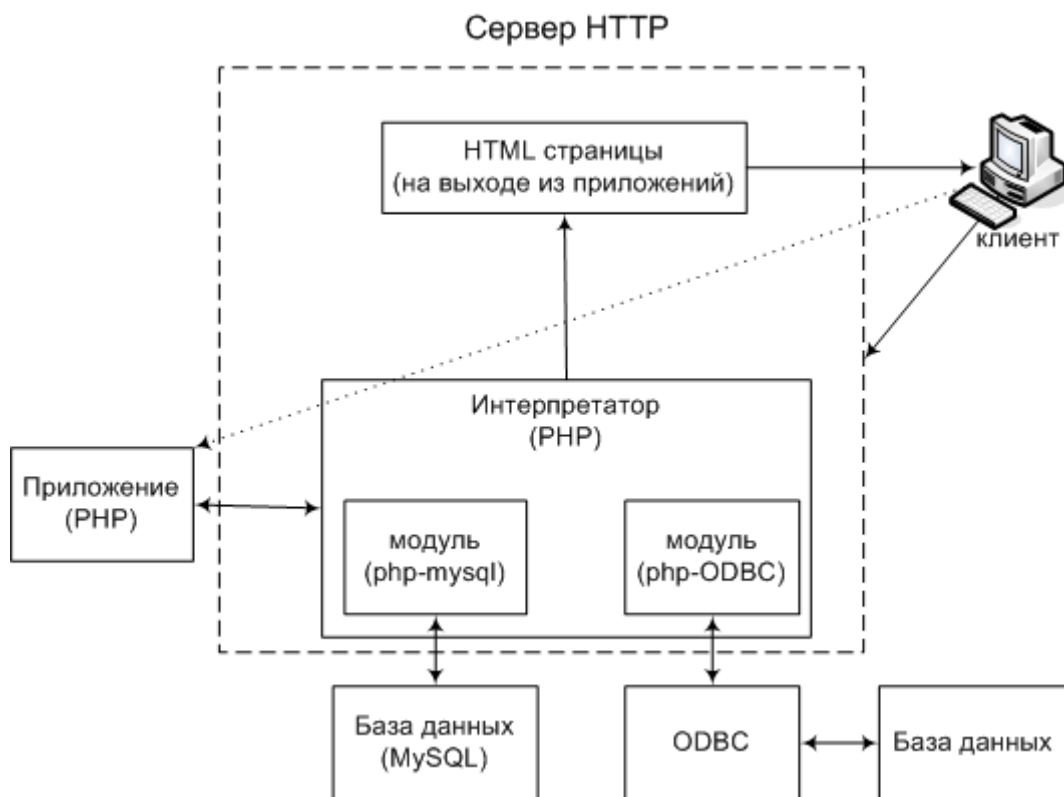


Рис. 2.1. Алгоритм обработки кода PHP

Синтаксис PHP подобный синтаксису мови Сі. Деякі елементи, такі як асоціативні масиви і цикл foreach, запозичені з Perl [14].

Для роботи програми не потрібно описувати будь-які змінні, які використовуються модулі і т. п. Будь-яка програма може починатися безпосередньо з оператора PHP.

Типова програма на PHP містить наступні складові частини:

```
<?php
```

```
echo 'Тест!';  
?>
```

PHP виконує код, що знаходиться всередині обмежувачів, таких як `<?Php?>`. Все, що знаходиться поза обмежувачів, виводиться без змін. В основному це використовується для вставки PHP-коду в HTML-документ, наприклад, так:

```
<html>  
<head>  
<title>Тест PHP</title>  
</head>  
<body>  
<?php echo 'Hello, world!'; ?>  
</body>  
</html>
```

Імена змінних починаються з символу `$`, тип змінної оголошувати не потрібно. На відміну від імен функцій і класів, імена змінних чутливі до регістру. Іменовані константи можуть бути оголошені як регістру, так і регістро незалежними. Змінні обробляються в рядках, укладених в апострофи або подвійні лапки, і heredoc-рядках (рядках, створених за допомогою оператора `<<<`).

Змінні в PHP представлені знаком долара з наступним ім'ям змінної. Ім'я змінної чутливе до регістру.

Імена змінних відповідають тим же правилам, що й інші найменування в PHP. Правильне ім'я змінної має починатися з букви або символу підкреслення з подальшими в будь-якій кількості літерами, цифрами або символами підкреслення. Це можна відобразити регулярним виразом: `'[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'`

Можливо використання символів: a-z, A-Z, и ASCII-символів з 127 по 255 (0x7f-0xff).

```
<?php
```

```
$var = "Інтернет_";  
$Var = "Магазин";  
echo "$var, $Var"; // виводить " Інтернет, Магазин "  
?>
```

PHP розглядає перехід на новий рядок як пробіл, так само як HTML та інші мови з вільним форматом. Інструкції розділяються за допомогою крапки з комою (;), за винятком деяких випадків.

PHP підтримує три типи коментарів: у стилі мови Cі (обмежені / * * /), C (що починаються з // і що йдуть до кінця рядки) та оболонки UNIX (з # до кінця рядка).

PHP є мовою програмування з динамічною типізацією, що не вимагає вказівки типу при оголошенні змінних, так само як і самого оголошення змінних. Перетворення між скалярними типами часто здійснюються неявно без додаткових зусиль (втім, PHP надає широкі можливості і для явного перетворення типів).

Скалярні типи даних:

- цілий тип (integer);
- речовинний тип даних (float, double);
- логічний тип (boolean);
- строковий тип (string);
- спеціальний тип NULL.

І нескалярні:

- «ресурс» (resource);
- масив (array);
- об'єкт (object);
- анонімна функція (closure) або псевдотип callback.

Посилання на зовнішні ресурси мають тип «ресурс» (resource). Змінні даного типу, як правило, представляють собою дескриптор, що дозволяє управляти зовнішніми об'єктами, такими як файли, динамічні зображення, результуючі таблиці бази даних тощо

Масиви (array) підтримують числові і рядкові ключі і є гетерогенними. Масиви можуть містити значення будь-яких типів, включаючи інші масиви. Порядок елементів і їх ключів зберігається. Не зовсім коректно називати php-масиви масивами, насправді це швидше упорядкований хеш.

Показчик на функцію в PHP може бути представлений замиканням або типом callback. Замикання доступне з версії 5.3 та в коді виглядає як просте визначення функції, в яку явно можна тягнути значення з контексту. Прим: `function ($ args .. $ argsN) use ($ ctxVar, $ ctxVar1) (definition;)`. callback тип може бути представлений: рядком (інтерпретується як назва функції); масивом де нульовий і перший елемент рядка (інтерпретується як назва статичної функції в класі); масивом де нульовий елемент об'єкт, а перший рядок (інтерпретується як метод у об'єкту). Для перевірки чи є значення викликуваним слід використовувати `is_callable ($ var)`

Як у будь-якій серйозній мові програмування в PHP є функції. Розглянемо приклад використання функції.

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Приклад функції.\n";
    return $retval;
}
?>
```

Всередині функції можна використовувати будь-який коректний PHP-код, в тому числі інші функції і навіть оголошення класів.

У PHP 3 функції повинні були бути визначені перш, ніж вони будуть використані. Починаючи з PHP 4 такого обмеження немає, виключаючи той випадок, коли функції визначаються умовно, як це показано в двох наступних прикладах.

У випадку, коли функція визначається в залежності від якого-небудь умови, наприклад, як це показано у двох наведених нижче прикладах, обробка опису функції повинна передувати її викликом.

Функції, залежні від умов:

```
<?php
$makefoo = true;

/* Функція foo() не може бути викликана в цьому місці, оскільки вона
ще не призначена, але ми можемо звернутись к bar() */
bar();
if ($makefoo) {
    function foo()
    {
        echo " Я не існую, поки виконання програми дійде до мене.\n";
    }
}
/* тепер ми можемо визвати foo(), оскільки $makefoo була
інтерпретована як true */
if ($makefoo) foo();
function bar()
{
    echo "Я існую одразу, як стартує програма. \n";
} ?>
```

Вкладені функції

```
<?php
function foo()
{
    function bar()
    { echo "Я не існую, поки foo() не визвана. \n"; } }
/* Ми поки не можемо звернутися до bar (), оскільки вона ще не
визначена. */
```

```
foo();  
/* Тепер ми можемо викликати функцію bar (), обробка foo () зробила її  
доступною* / bar(); ?>
```

2.3.3. Засоби розробки бази даних

MySQL вільна система керування реляційними базами даних. MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. 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 розширювалася і зараз MySQL одна найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

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 внутрішні 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 [1].

2.3.4. Вибір операційної системи

Операційна система, скорочено ОС (англ. operating system, OS) це базовий комплекс програм, що виконує управління апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем.

Операційна система звичайно складається з ядра операційної системи та базового набору прикладних програм.

Найпоширенішою ОС для ПК є WINDOWS, створена фірмою Microsoft. На сьогодні WINDOWS поряд з IBM-сумісними ПК є одною найпоширеніших технологічних платформ. WINDOWS у версіях WINDOWS-95, WINDOWS-98, WINDOWS-2000 - це 32 розрядна багатозадачна ОС[2]. цій ОС використовуються:

- робочі вікна, призначені для різних ПП;
- впливаючі вікна повідомлень програми і отримання реакції користувача;
- діалогові вікна, через які здійснюється основний ввід інформації в програму.

Яскравою зовнішньою ознакою WINDOWS є багатовіконний графічний інтерфейс. Вікно - це частина екрану монітора, яка є полем дії програми. Відображення вводу управляючих команд і інформації в програму, а також вивід зображень і даних з неї відбувається через вікно. Для позначення об'єктів, програм, операцій, опцій вибору використовуються піктограми. Піктограми - це графічні символи, які асоціативно нагадують те, що вони позначають.

Поряд з графічним інтерфейсом важливою рисою WINDOWS є багатозадачність. Одночасно можуть виконуватися декілька різних програм або копій однієї і тої ж програми. Наприклад, користувач може одночасно використовувати текстовий процесор, базу даних, програму для графічного представлення даних і електронні таблиці.

Сучасні версії Windows реалізують механізм ВП(віртуального простору). Його суть полягає в тому, що користувачу надається віртуальний (можливий, але реально не існуючий) адресний простір, що визначається 32-х розрядною довжиною адреси (4 Гбайти) розміщення в пам'яті комп'ютера. В той же час фізична ОП може бути значно меншою (одиниці - десятки Мбайт). Інформація, що не поміщається в фізичній ОП, знаходиться на дисковому запам'ятовуючому пристрої.

MacOS (до версії 10.7 включно - Mac OS X, до версії 10.11 включно - OS X) - POSIX-сумісна операційна система корпорації Apple. Є спадкоємицею Mac OS 9 - так званого остаточного релізу «класичної» Mac OS - основної операційної системи корпорації Apple 1984 року. OS X входить в сімейство операційних систем Apple OS X, до якого також належить ОС для мобільних пристроїв - Apple iOS. У macOS використовується ядро Darwin, засноване на мікроядрі Mach, що містить код, написаний самою компанією Apple та код, отриманий ОС NeXTSTEP та FreeBSD. Apple macOS випускається для комп'ютерів Macintosh (Макінтош) на базі процесорів PowerPC і Intel (починаючи з версії 10.6,) macOS підтримує тільки комп'ютери Mac на базі процесора Intel. Mac OS - друга за популярністю у світі операційна система. Її ринкова частка в червні 2010 року – склала 6,8%.

Браузер, також переглядач, вебпереглядач, вебоглядач, вебнавігатор (англ. browser) - програмне забезпечення для комп'ютера або іншого електронного пристрою, як правило, під'єданого до Інтернету, що дає можливість користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій вебсторінці. Тексти та малюнки можуть містити посилання на інші вебсторінки, розташовані на тому ж вебсайті або на інших вебсайтах. Вебпереглядач з допомогою гіперпосилань дозволяє користувачеві швидко та просто отримувати інформацію, розміщену на багатьох вебсторінках[3].

Найпершим вебпереглядачем був Mosaic, розроблений в Національному центрі застосування суперкомп'ютерів (NCSA) Іллінойського університету в Урбана-Шампейн.

Станом на квітень 2021 року існує низка програм для навігації в Інтернеті. Найпопулярніші з них:

- Google Chrome;
- Safari;
- Mozilla Firefox;
- Opera.

Кожний переглядач має свої відмінності в інтерпретації гіпертексту, що пов'язано з різною реалізацією рушіїв відображення та створенням пропрієтарних тегів для розширення можливостей стандартного HTML. Так, один текст може бути відтворено по-різному Internet Explorer та Safari. Багато Інтернет-сервісів оптимізують свої сторінки під один стандарт, віддаючи перевагу одному переглядач (приклад Flickr з Mozilla Firefox).

Google Chrome-це потужний безкоштовний і безпечний переглядач веб-сторінок, що поєднав у собі простоту та провідні новітні технології. Він використовує високопродуктивний рушій обробки для завантаження сторінок, характеризується стабільністю роботи та мінімалістичним інтерфейсом.

Інтерфейс браузера не перевантажено, він простий для розуміння та використання й має гнучку систему налаштувань. Серед основних елементів керування: назад, вперед, відновлення та зупинення завантаження, кнопка додавання поточної сторінки в закладки, перехід на головну сторінку. Основним компонентом інтерфейсу є панель вкладок. Від інших браузерів інтерфейс Chrome відрізняється наявністю сторінки швидкого доступу, зручним розміщенням багатьох елементів інтерфейсу та широким асортиментом тем і розширень.

Google Chrome запускається за лічені секунди. Його потужний механізм покликаний блискавично виконувати найскладніші операції. Універсальне вікно пошуку браузера, куди можна вводити пошукові запити та веб-адреси, показує пропозиції просто під час введення запитів. Функція автоматичного завершення дозволяє отримати потрібну інформацію, ввівши всього кілька символів, а спеціальний механізм візуалізації миттєво завантажує сторінки.

Браузер має вбудований захист від зловмисного ПЗ та фішингу. Автоматичне оновлення постійно вносить виправлення в систему безпеки, чим позбавляє користувача від зайвих клопотів.

З Google Chrome ви можете повністю контролювати особисту інформацію. Для цього можна перейти в режим анонімного перегляду. Всі куки, створені в цьому режимі, автоматично видаляються після закриття вікна.

Переваги Google Chrome: веб-переглядач безкоштовний; простий, швидкий і безпечний; багатомовний інтерфейс; функція автоматичного перекладу сторінок; конфіденційність використання; персоналізація даних; імпорт закладок та паролів з інших браузерів; вбудований переглядач PDF; після закриття браузера всі вкладки зберігаються; наявні інструменти для веб-майстрів.

Недоліки Google Chrome: навантажує операційну систему.

Firefox – це потужний, популярний і багатофункціональний веб-переглядач, що є гарантом безпечної, швидкої та результативної діяльності його користувачів на просторах Інтернету. Він безпечний, пропонує безліч можливостей для своїх користувачів, має зручне й стильне обличчя.

Перше з чим ми стикаємося – це інтерфейс. З часом розробники значно удосконалили його дизайн. Відчутно, що вони опиралися не лише на естетичний смак нового покоління, а й продумали максимально зручне та доступне розташування важливих програмних елементів.

Не менш вагомою перевагою є безпеки діяльності користувачів у світі спаму, реклами та вірусів. Firefox має кілька функцій, які блокують рекламне спостереження, видаляють куки, стирають історію перегляду, визначають наявність непомітних елементів збору інформації, приховують та систематизують ваші паролі.

Основний функціональний багаж браузера, орієнтований на користувачів, – це корисні додатки, розширення, теми, закладки, функція синхронізації роботи з іншими пристроями.

Переваги Firefox: швидке завантаження сторінок; підтримка вкладок, керування ними, реорганізація, збереження; зручна панель закладок, мітки сайтів, бібліотека; зручний менеджер паролів; сучасне інтуїтивне оформлення дизайну; можливість персоналізації панелі інструментів; конфіденційність діяльності користувача в інтернеті завдяки приватному перегляду; можливість очищення історії навігації, куків, даних веб-форм, паролів; інструмент створення скріншотів; величезна кількість доступних додатків; синхронізація Firefox на всіх пристроях користувача; адресний рядок з функцією автодоповнення; потужні

інструменти для веб-розробників; автоматичне оновлення браузера, тем, додатків, мовних пакетів; повноцінне масштабування веб-сторінок; максимальна підтримка новітніх веб-технологій.

Недоліки Firefox: час від часу, після оновлень перестають працювати деякі додатки.

2.4. Опис структури системи та алгоритмів її функціонування

2.4.1. Розробка контенту веб-сторінки

CMS (Content management system) – інформаційна система або комп'ютерна програма, яка використовується для забезпечення і організації спільного процесу створення; редагування і управління вмістом, інакше – контентом.

Основні функції CMS :

- надання інструментів для створення вмісту, організація спільної роботи над вмістом;
- обробка вмісту: зберігання, контроль версій, дотримання режиму доступу, управління потоком документів;
- публікація вмісту;
- подання інформації у вигляді, зручному для навігації, пошуку.

Прикладами CMS є WordPress, OpenCart, Joomla. При проектуванні сайту використовувалась CMS WordPress. Щоб потрапити до середовища адміністрування CMS WordPress, потрібно у адресному полі, після домену ввести /wp-admin/ та заповнити відповідні поля в вікні авторизації. При введенні коректних даних з'явиться головна сторінка панелі адміністратора, макет якої складається з потрібних для роботи елементів (рис. 2.2).

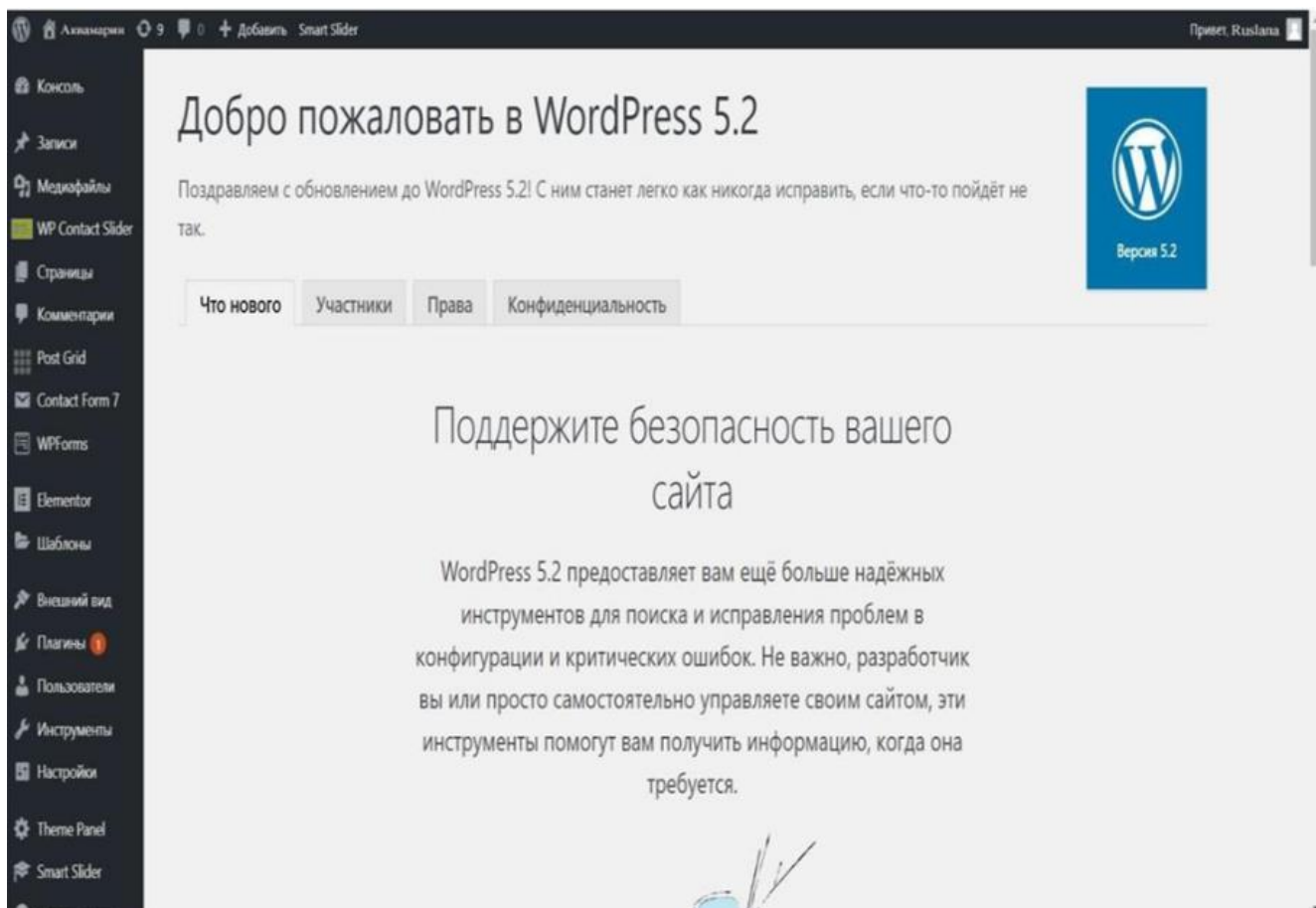


Рис. 2.2. Головна сторінка панелі розробника

Інтерфейс сайту – це «лице» закладу, перший етап взаємодії з користувачем. Цей етап дуже важливий, тому що саме на ньому користувач сайту (потенціальний клієнт клубу) вирішує, чи буде він користуватись послугами, чи піде в інший.

Інтерфейс сайту формується з трьох частин:

- шаблонів файлів (templates);
- налаштувань каскадної таблиці стилів CSS, яка зберігається у файлі style.css
- файлів зображень, які прописані у style.css.

Схема генерації інтерфейсу приведена на рис. 2.3.

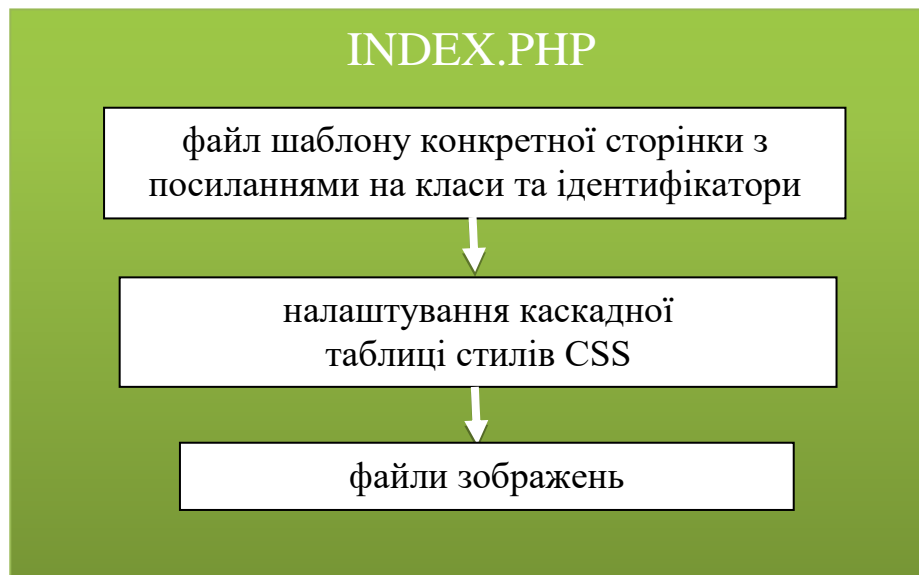


Рис. 2.3. Схема генерації інтерфейсу

При такій організації інтерфейс сайту зберігається в одному місці і не дублюється, що робить код простіше. При створенні інтерфейсу використано таблично-шаровий метод верстки, який складається з таблиць і шарів (DIV). Параметри таблиць та шарів описанні у файлах шаблонів та у style.css.

Структуру файлу index.php наведено на рис. 2.4. Оскільки контактна інформація вкрай важлива для сайту інтернет-магазину, то її необхідно вивести на примітне місце, тобто в шапку.

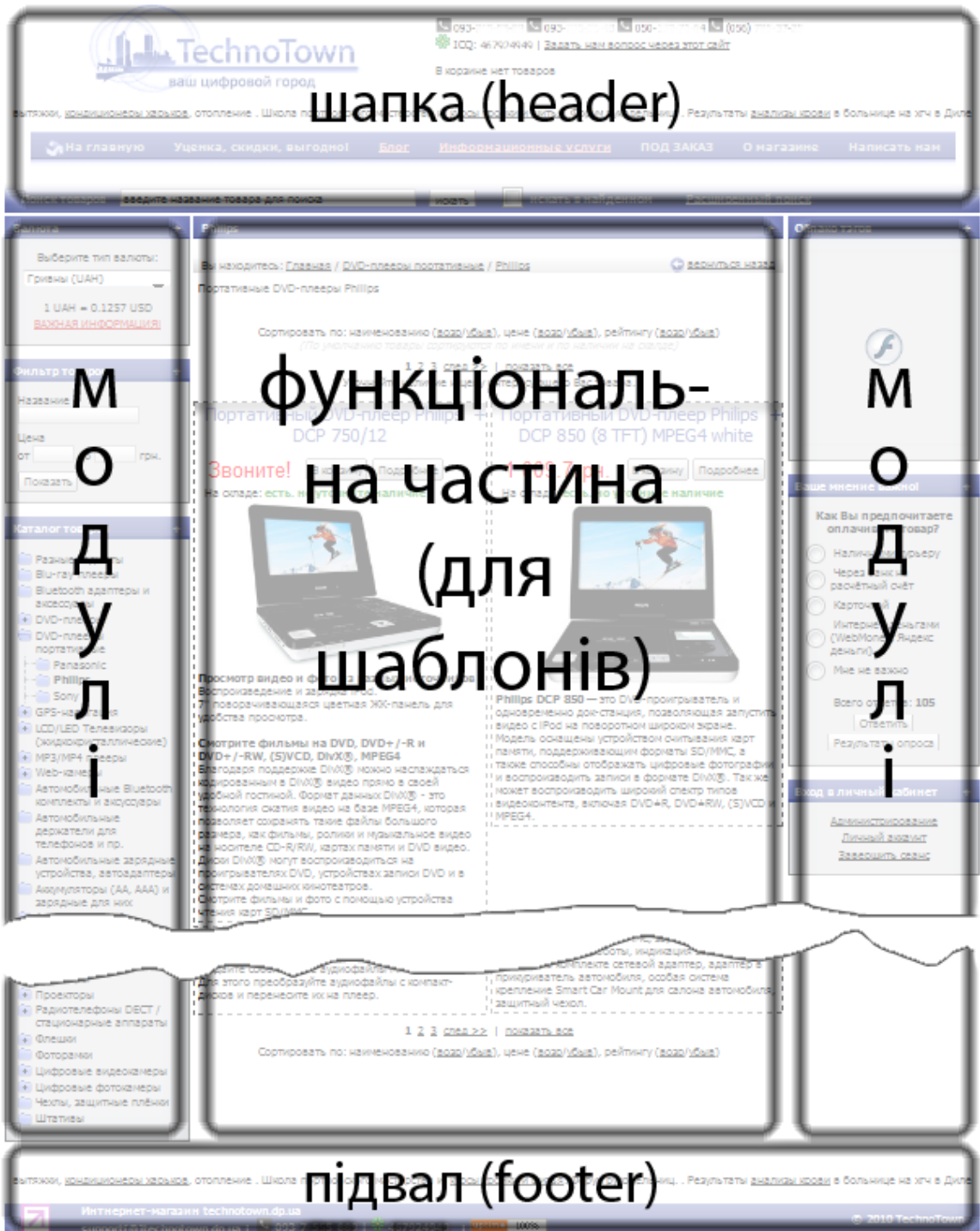


Рис. 2.4. Загальна структура файлу index.php

Основна частина складається з блоків виводу модулів зліва та зправа від функціональної частини, які виводить шаблон. Пишемо такий код:

```
<table cellspacing="0" cellpadding="0" width="100%">
  <tr>
    <td class="container">
      <table cellspacing="0" cellpadding="0"
width="100%"> <tr>
```

Модулі лівого блоку знаходяться у окремому файлу, нам треба його вивести:

```
{if $count_left_blocks > 0}<td id="left" valign="top" align="center"
class="print"><div class="fill white"></div>{include
file="left_blocks.tpl.html"}</td>{/if}
```

Шаблони основної функціональної частини також зберігаються у окремих файлах, тож посилаємось на них, щоб вивести:

```
<td class="center" align="left" valign="top">{include
file="$main_content_template"}</td>
```

Модулі лівого блоку також зберігаються у окремому файлі:

```
{if $count_right_blocks > 0}<td id="right" valign="top" align="center"
class="print"><div class="fill white"></div>{include
file="right_blocks.tpl.html"}</td>{/if}
```

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

Треба зауважити, що контактну інформацію ми виводимо за допомогою графічних файлів, тип самим ми унеможливуємо спам на наш інтернет-маєнджер та телефони. Адреса пошти виводиться з "[@]" посередині, що також не дасть потрапити адресі до спам-розсилки.

Для підключення файлів, які відповідають за вивід інтерфейсу у index.php ми пишемо такий код:

```
<link rel="stylesheet" href="design/user/default/images/tabber.css"
type="text/css" >
```

```
<link rel="stylesheet" href="design/user/default/images/style.css" type="text/css"
>
```

```
<link rel="stylesheet" href="design/user/default/images/styleprint.css"
type="text/css" >
```

```
<link rel="icon" href="design/user/default/images/favicon.ico" type="image/x-
icon">
```

Звернення до файлів стилів ми будемо використовувати на всіх сторінках сайту, шаблонах та модулях. CSS параметри вбудовуються в HTML код. Наприклад, для виводу головного меню сайту у файлі index.php пишемо такий код:

```
<table align="center" border="0" vspace="0" cellspacing="0" >
  <td class="menust" onClick="location.href='http://technotown.dp.ua'"><a
href="http://technotown.dp.ua" class="menulink">На главную</a></td>
  <td class="menust" onClick="location.href='/page_7.html'"><a
href="page_7.html" class="menulink">Уценка, скидки, выгодно!</a></td>
  <td class="menustred" onClick="location.href=' /blog'"><a href=" /blog"
class="menulinkred">Блог</a></td>
  <td class="menust" onClick="location.href=' /page_1.html'"><a href="
/page_1.html" class="menulink">ПОД ЗАКАЗ</a></td>
  <td class="menust" onClick="location.href=' /page_1.html'"><a href="
/page_1.html" class="menulink">О магазине</a></td>
  <td class="menust" onClick="location.href=' /feedback.html'"><a href="
/feedback.html" class="menulink">Написать нам</a></td>
</table>
```

Вигляд стовбців таблиці заданий класами menust, menulink, menustred, menulinkred. Це класи стилів прописані у файлі стилів. Для її оголошення прописуємо наступний код у файл style.css.

```
.menust {
background: url(blank.gif) repeat-x;
color:#DDDDDD;
cursor: pointer;
text-decoration:none;
FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
```



```
padding-left: 15px; padding-right:15px; padding-top: 7px; vertical-align:text-top;
font-weight:bold; font-size:13px; }
.menulink {
color:#DDDDDD;
cursor: pointer;
text-decoration:none;
FONT-FAMILY: verdana, arial, sans-serif;
font-weight:bold; font-size:11px; }
.menust:hover {
background: url(blank_over.gif) repeat-x;
background-color: 49589b;
cursor: pointer;
text-decoration:underline;
FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
padding-left: 15px; padding-right:15px; padding-top: 7px; vertical-align:text-top;
font-weight:bold; font-size:13px; }
.menulink:hover {
color: #fff;
cursor: pointer;
text-decoration:underline;
FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
font-weight:bold; font-size:11px; }
.menustred {
background: url(blank.gif) repeat-x;
background-color: 49589b;
color:#DDDDDD;
cursor: pointer;
text-decoration:none;
```

```

FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
padding-left: 15px; padding-right:15px; padding-top: 7px; vertical-align:text-top;
font-weight:bold; font-size:13px;}

.menustred:hover {
background: url(blank_overred.gif) repeat-x;
cursor: pointer;
text-decoration:underline;
FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
padding-left: 15px; padding-right:15px; padding-top: 7px; vertical-align:text-top;
font-weight:bold; font-size:13px; }

.menulinkred {
color:#F4C6C6;
cursor: pointer;
text-decoration: underline;
FONT-FAMILY: verdana, arial, sans-serif;
font-weight:bold; font-size:11px; }

.menulinkred:hover {
color: #fff;
cursor: pointer;
text-decoration:underline;
FONT-FAMILY: verdana, arial, sans-serif;
height: 37px;
font-weight:bold; font-size:11px; }

```

Як результат, ми маємо меню сайту, яке зображено на рис. 2.5.



Рис. 2.5. Меню сайту

Шапка сайту містить логотип, який дозволяє повертатись на головну сторінку проекту та назви сторінок проекту (рис. 2.6).

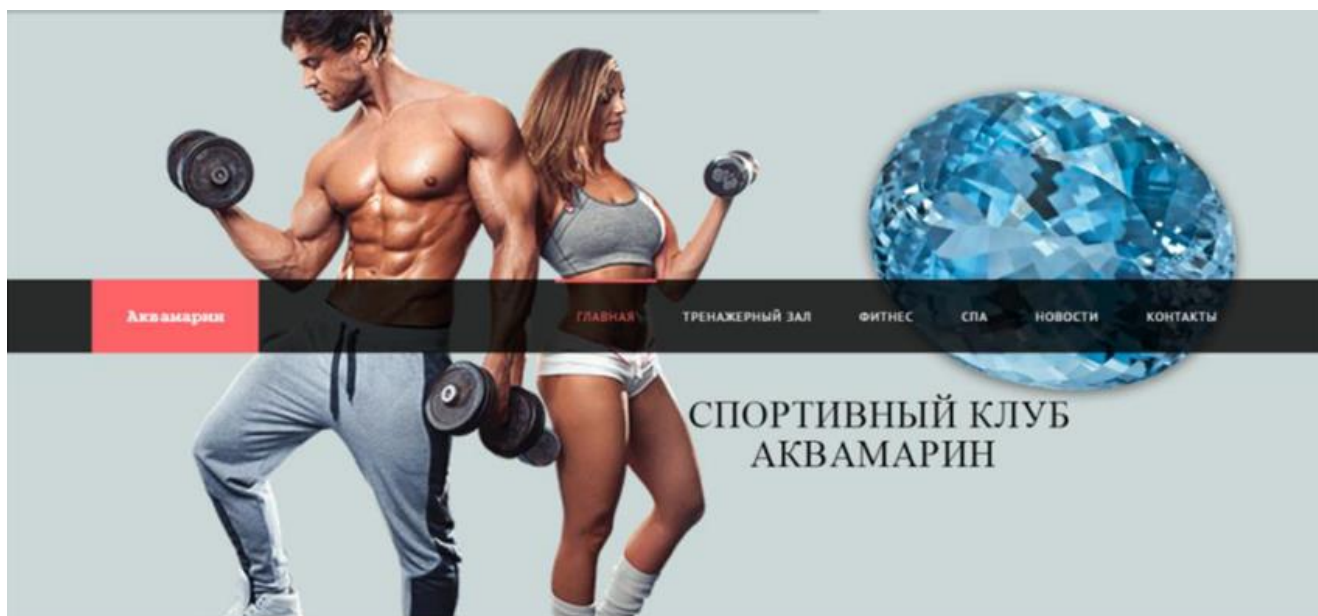


Рис. 2.6. Шапка сайту

Основний блок сайту містить наступні елементи:

- інформаційний блок про спортивний комплекс та час його роботи (рис. 2.7);
- інформаційний слайдер про тренерів, які працюють у спортивному комплексі (рис. 2.8);
- інформаційний слайдер про тренажерні зали (рис. 2.9);
- інформаційний слайдер про СПА (рис. 2.10).

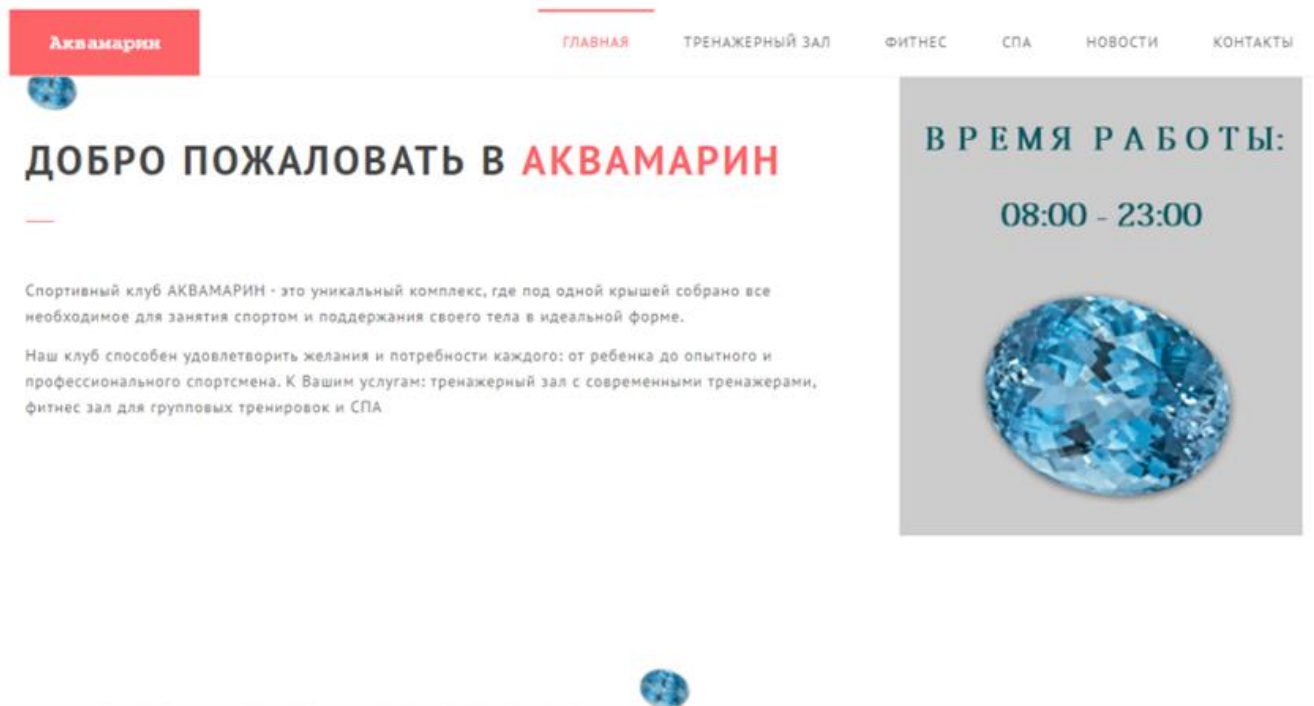


Рис. 2.7. Основний зміст сайту

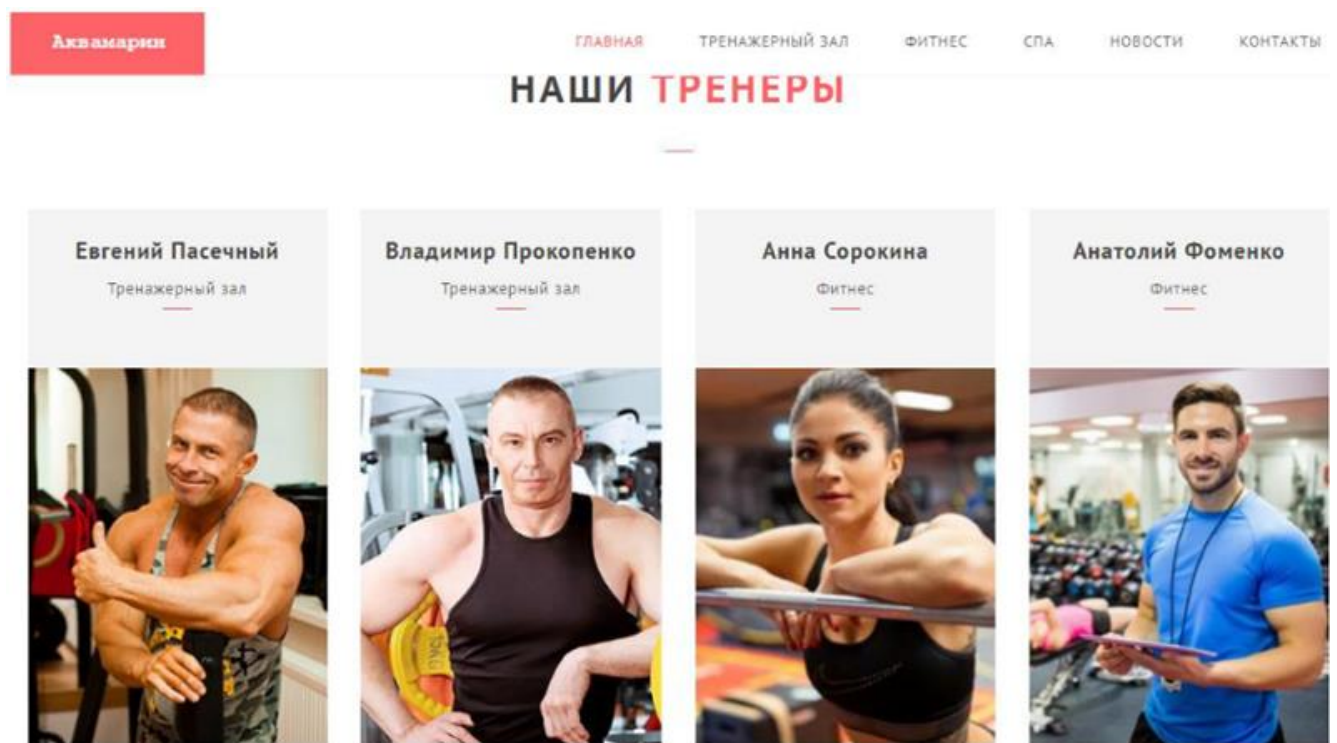


Рис. 2.8. Блок «Наши тренера»

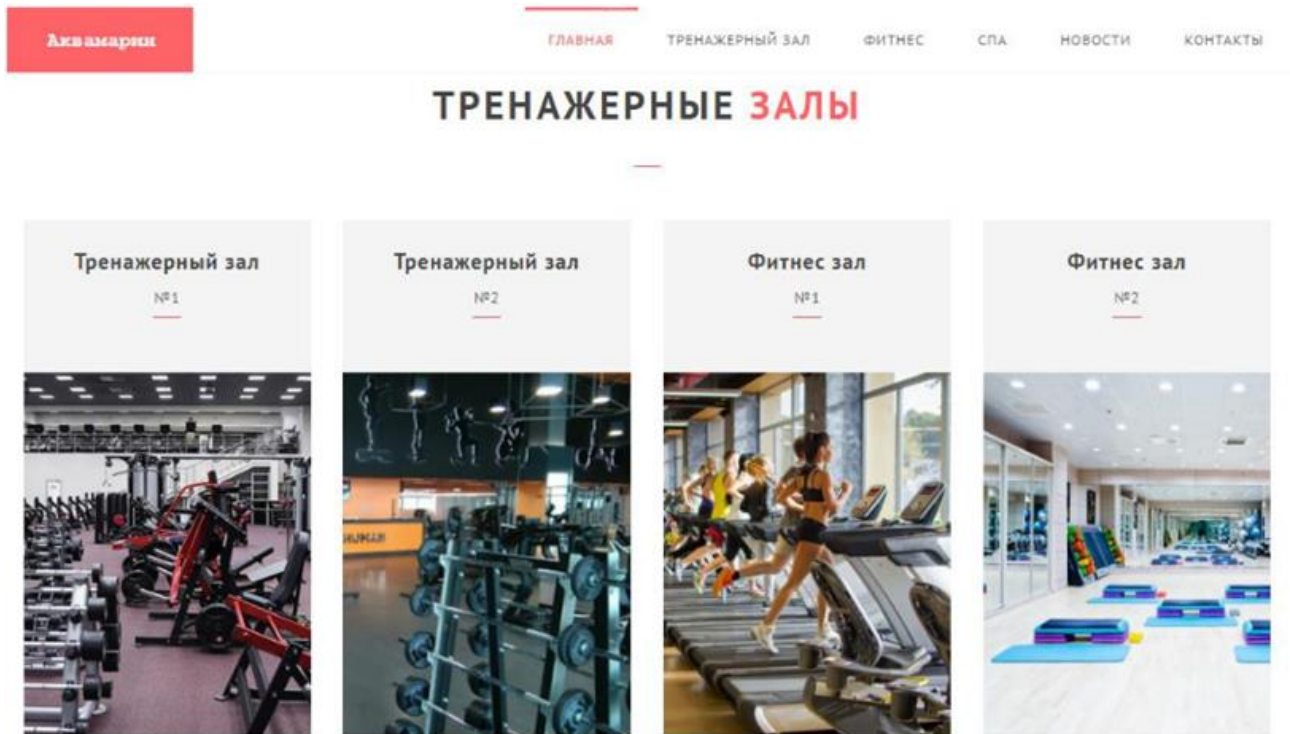


Рис. 2.9. Блок «Тренажерные залы»



Рис. 2.10. Блок «СПА салон»

На кожній сторінці сайту передбачена плаваюча кнопка, яка при натисканні на неї повертає користувача на початок сторінки сайту.

2.4.2. Розробка бази даних системи

Сайт матиме дворівневий доступ: адміністратор та користувач. Для кожного з яких передбачені відповідні функціональні можливості. Для розуміння виконуваних ними дій на сайті зображена діаграма (рис. 2.11), на якій проілюстровано взаємодію об'єктів у рамках відповідного процесу.

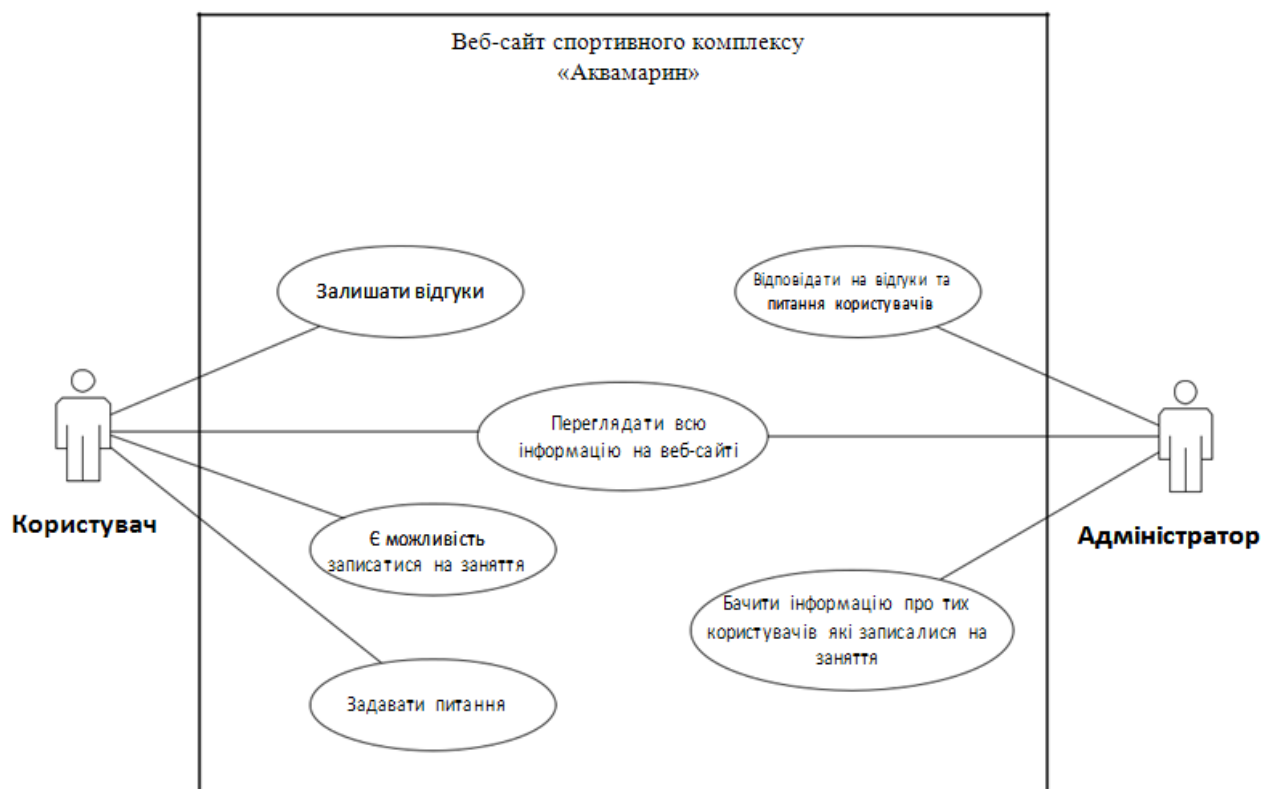


Рис. 2.11. Діаграма варіантів використання

Інформація з сайту обробляється за допомогою phpMyAdmin. phpMyAdmin – веб-додаток з відкритим кодом, написаний на мові PHP і представляє собою веб-інтерфейс для адміністрування СКБД MySQL. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера

MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних.

Ця програма користується великою популярністю у веб-розробників, оскільки дозволяє керувати базою даних MySQL без вводу SQL команд через дружній інтерфейс і з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення.

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL.

База даних сайту складається з 10 таблиць з потрібними відношеннями, атрибутами та їх властивостями (рис. 2.12).

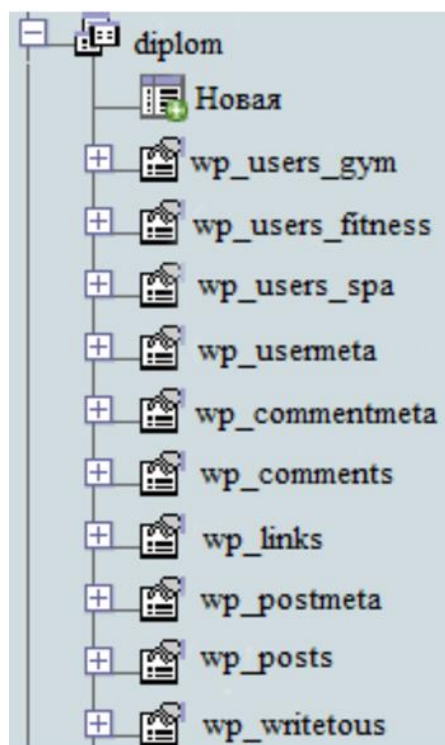


Рис. 2.11. Створені таблиць у базі даних

Для кожного відношення наведені атрибути з їх внутрішньою назвою, типом та довжиною:

– «Користувач» - «Відгуки»: тип зв'язку - один до багатьох; для зв'язування відношень в «Користувач» був розміщений зовнішній ключ «ID»;

– «Коментарії» - «Статті»: тип зв'язку - багато до одного; для зв'язування відношень в «Відгуки» був розміщений зовнішній ключ «comment_ID»;

– «Статті» - «Форми»: тип зв'язку один до одного; для зв'язування відношень в «Статті» був розміщений зовнішній ключ «ID»;

– «Форми» - «Елементи»: тип зв'язку - один до багатьох; для зв'язування відношень в «Форми» був розміщений зовнішній ключ «forms_ID».

Після нормалізації всієї бази даних відбувається налаштування зв'язків (рис. 2.12).

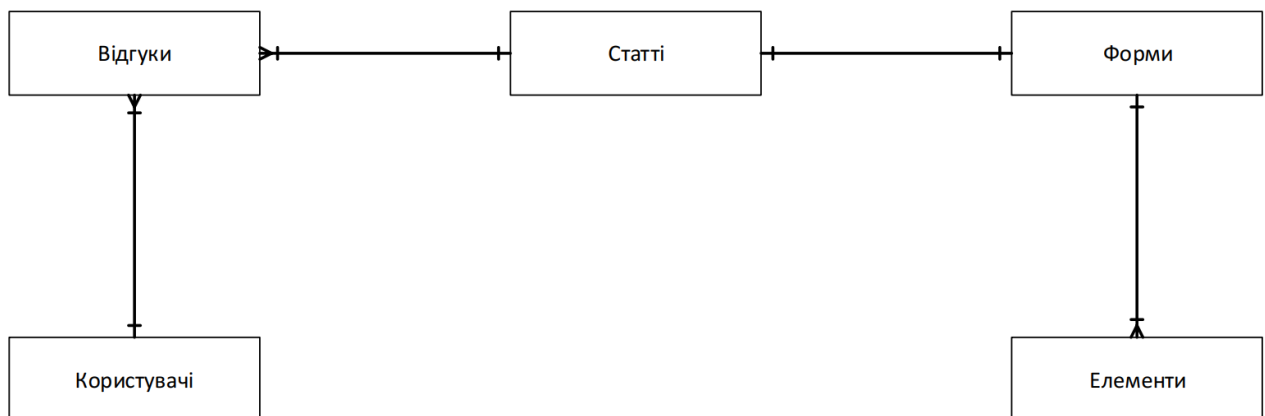


Рис. 2.12. Логічна схема зв'язків бази даних

Сутності та їх атрибути наведені в таблицях 2.1 - 2.10.

Таблиця 2.1

Схема відношення (wp_usersmeta)

Ім'я поля	Тип, довжина	Null	Примітки
umeta_id	bigint(20)	Ні	первинний ключ
user_id	bigint(20)	Ні	зовнішній ключ
meta_key	varchar(255)	Так	обов'язкове поле
meta_value	longtext	Так	обов'язкове поле

Таблиця 2.2

Схема відношення (wp_users_gym)

Ім'я поля	Тип, довжина	Null	Примітки
ID	bigint(20)	Hi	первинний ключ
user_name	varchar(50)	Hi	обов'язкове поле
user_time	datetime	Hi	обов'язкове поле
user_data	datetime	Hi	обов'язкове поле
user_text	varchar(400)	Hi	обов'язкове поле

Таблиця 2.3

Схема відношення (wp_comments)

Ім'я поля	Тип, довжина	Null	Примітки
comment_ID	bigint(20)	Hi	первинний ключ
comment_post_ID	bigint(20)	Hi	зовнішній ключ
comment_author	tinytext	Hi	обов'язкове поле
comment_author_email	varchar(100)	Hi	обов'язкове поле
comment_author_url	varchar(200)	Hi	обов'язкове поле
comment_author_IP	varchar(100)	Hi	обов'язкове поле
comment_date	datetime	Hi	обов'язкове поле
comment_date_gmt	datetime	Hi	обов'язкове поле
comment_content	text	Hi	обов'язкове поле
comment_karma	int(11)	Hi	обов'язкове поле
comment_approved	varchar(20)	Hi	обов'язкове поле
comment_agent	varchar(255)	Hi	обов'язкове поле
comment_type	varchar(20)	Hi	обов'язкове поле
comment_parent	bigint(20)	Hi	обов'язкове поле
user_id	bigint(20)	Hi	обов'язкове поле

Таблиця 2.4

Схема відношення (wp_users_fitness)

Ім'я поля	Тип, довжина	Null	Примітки
ID	bigint(20)	Ні	первинний ключ
user_name	varchar(50)	Ні	обов'язкове поле
user_time	datetime	Ні	обов'язкове поле
user_data	datetime	Ні	обов'язкове поле
user_text	varchar(400)	Ні	обов'язкове поле

Таблиця 2.5

Схема відношення (wp_links)

Ім'я поля	Тип, довжина	Null	Примітки
link_id	bigint(20)	Ні	первинний ключ
link_url	varchar(255)	Ні	обов'язкове поле
link_name	varchar(255)	Ні	обов'язкове поле
link_image	varchar(255)	Ні	обов'язкове поле
link_target	varchar(25)	Ні	обов'язкове поле
link_description	varchar(255)	Ні	обов'язкове поле
link_visible	varchar(20)	Ні	обов'язкове поле
link_owner	bigint(20)	Ні	обов'язкове поле
link_rating	int(11)	Ні	обов'язкове поле
link_updated	datetime	Ні	обов'язкове поле
link_rel	varchar(255)	Ні	обов'язкове поле
link_notes	mediumtext	Ні	обов'язкове поле
link_rss	varchar(255)	Ні	обов'язкове поле

Схема відношення (wp_posts)

Ім'я поля	Тип, довжина	Null	Примітки
ID	bigint(20)	Ні	первинний ключ
post_author	bigint(20)	Ні	обов'язкове поле
post_date	datetime	Ні	обов'язкове поле
post_date_gmt	datetime	Ні	обов'язкове поле
post_content	longtext	Ні	обов'язкове поле
post_title	text	Ні	обов'язкове поле
post_excerpt	text	Ні	обов'язкове поле
post_status	varchar(20)	Ні	обов'язкове поле
comment_status	varchar(20)	Ні	обов'язкове поле
ping_status	varchar(20)	Ні	обов'язкове поле
post_password	varchar(255)	Ні	обов'язкове поле
post_name	varchar(200)	Ні	обов'язкове поле
to_ping	text	Ні	обов'язкове поле
pinged	text	Ні	обов'язкове поле
post_modified	datetime	Ні	обов'язкове поле
post_modified_gmt	datetime	Ні	обов'язкове поле
post_content_filtered	longtext	Ні	обов'язкове поле
post_parent	bigint(20)	Ні	обов'язкове поле
guid	varchar(255)	Ні	обов'язкове поле
menu_order	int(11)	Ні	обов'язкове поле
post_type	varchar(20)	Ні	обов'язкове поле
post_mime_type	varchar(100)	Ні	обов'язкове поле
comment_count	bigint(20)	Ні	обов'язкове поле

Таблиця 2.7

Схема відношення (wp_commentmeta)

Ім'я поля	Тип, довжина	Null	Примітки
meta_id	bigint(20)	Ні	первинний ключ
comment_id	bigint(20)	Ні	зовнішній ключ
meta_key	varchar(255)	Так	обов'язкове поле
meta_value	longtext	Так	обов'язкове поле

Таблиця 2.8

Схема відношення (wp_users_spa)

Ім'я поля	Тип, довжина	Null	Примітки
ID	bigint(20)	Ні	первинний ключ
user_name	varchar(50)	Ні	обов'язкове поле
user_time	datetime	Ні	обов'язкове поле
user_data	datetime	Ні	обов'язкове поле
user_text	varchar(400)	Ні	обов'язкове поле

Таблиця 2.9

Схема відношення (wp_postmeta)

Ім'я поля	Тип, довжина	Null	Примітки
meta_id	bigint(20)	Ні	первинний ключ
post_id	bigint(20)	Ні	зовнішній ключ
meta_key	varchar(255)	Так	обов'язкове поле
meta_value	longtext	Так	обов'язкове поле

Схема відношення (wp_writetous)

Ім'я поля	Тип, довжина	Null	Примітки
ID	bigint(20)	Ні	первинний ключ
user_name	varchar(50)	Ні	обов'язкове поле
user_email	varchar(100)	Ні	обов'язкове поле
user_phone	varchar(14)	Ні	обов'язкове поле

2.4.3. Опис фізичної структури проєкту

Розроблене програмне забезпечення складається з 193 файлів розміром 9,01 МБ. До складу проєкту входять наступні файли: *.html, *.js, *.css, *.eot, *.svg, *.woff, *.otf, *.ttf, *.png.

Розроблений проєкт має кореневу структуру, представлену файлами та папками .

Головні файли та папки проєкту:

- index.html-головний файл сайту;
- comment.php-файл для створення відгуку;
- class-detail.html-файл сторінки сайту «Тренажерный зал»;
- contact.html- файл сторінки сайту «Контакты»;
- event-list.html- файл сторінки сайту «Новости»;
- fitnes-detail.html- файл сторінки сайту «Фитнес»;
- send.php-файл для створення питання;
- SPA-detail.html- файл сторінки сайту «СПА»;
- zapis.php-файл для створення запису;
- css-папка з усіма стилями сайту;
- fonts-папка з шрифтами сайту;
- images-папка з усіма зображеннями;
- js-папка з файлами JavaScripts.

Інші папки та файли генеруються CMS WordPress автоматично, та зазвичай не редагуються програмістом.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Дані системи зберігаються в базі даних, яка складається з 10 таблиць та містить наступну вхідну інформацію, що забезпечує роботу всієї програми: ПІБ користувача, е-мейл, номер телефону, час та назву заняття, на яке він реєструється, а також текст повідомлення, надісланого адміністратору сайту.

Вихідною інформацією є зміст відображених за запитом сторінок сайту та інша відповідна реакція на запити користувача.

2.6. Опис роботи розробленої системи

2.6.1. Використані технічні засоби

Комп'ютерна техніка сьогодні представлена величезною різноманітністю пристроїв з різними характеристиками, які знаходяться в широкому ціновому діапазоні. Залежно від вартості комп'ютери можна умовно розділити на кілька груп, кожна з яких представлена комп'ютерною технікою, здатної вирішувати певні завдання:

- бюджетні системні блоки. Група представлена комп'ютерами, вартість яких коливається близько 300 \$, здатні вирішувати відносно прості завдання: здійснювати роботу з текстом, простими офісними програмами та інтернет-ресурсами, переглядати відеоролики і т.п. Окремі моделі даної категорії комп'ютерів здатні навіть відтворювати гри сучасних творців, але тільки на найнижчих настройках, що не забезпечують комфортної гри;

- універсальні настільні ПК, вартістю до 800 \$. Категорія є, мабуть, найбільш затребуваною на сучасному ринку комп'ютерної техніки. Вона представлена пристроями, оснащеними досить

– потужними процесорами і продуктивними відеокартами, що зумовлюють можливість вирішення досить складних задач і використовувати програми досить високої ємності. Крім вирішення найскладніших офісних завдань, вони здатні запускати сучасні ігри на середніх (а деякі - і на найвищих налаштуваннях);

– ігрові комп'ютери. Пристрої високої потужності, вартість яких перевищує 800 \$. До цієї категорії відносяться високопродуктивні ПК, вони здатні впоратися з найскладнішими завданнями і відтворювати найважчі ігри на максимальних налаштуваннях, забезпечуючи комфорт ведення гри. Їх недоліком є потреба в ефективному охолодженні, тому деякі моделі характеризуються підвищеним шумом при роботі;

– комп'ютери спеціалізованого призначення. Цю категорію складають ПК, що збираються індивідуально під конкретні завдання. Як правило, вони оснащуються потужним процесором, продуктивною відеокартою і декількома платами оперативної пам'яті.

Особливості вибору комплектуючих для збірки ПК:

- корпус;
- блок живлення;
- процесор;
- материнська плата;
- відеокарта;
- оперативна пам'ять;
- жорсткий диск.

Для користування веб-додатком потрібен персональний комп'ютер, який повинен мати мінімальні системні вимоги, необхідні для роботи (представлені в табл. 2.10[4]).

Мінімальні системні вимоги

Обладнання	Значення
Відеоадаптер	Nvidia geforce gtx 1650
Вінчестер	400 Мб
Оперативна пам'ять	512 Мб
Мережева карта	D-Link DFE-538TX
Контролери	клавіатура, комп'ютера миша
Розмір монітору	800x600
Процесор	Intel Pentium 4

2.6.2. Використані програмні засоби

Для реалізації даного завдання було обрано веб-додаток, написаний на html, css , java script та мові PHP. При проектуванні сайту використовувалась CMS WordPress.

2.6.3. Виклик та завантаження програми

Для запуску проекту потрібно в пошуковій системі набрати назву спортивного комплексу «Аквамарин» та перейти на сайт за посиланням «www.akvamarin.ua».

2.6.4. Опис інтерфейсу користувача**2.6.4.1. Опис веб-сайту**

Головна сторінка сайту містить шапку сайту, в основній частині якій знаходиться інформація про спортивний комплекс, блок з інформацією про тренерів, про тренажерні зали та спа. Футер та містить меню для переходу між

його сторінками:

- «Главная» - гіперпосилання, яке надає головної сторінки сайту (рис. 2.13-2.17);
- «Тренажерный зал» - надає перехід до вікна з інформацією про тренажерні зали, фото залів, прайс абонементів та час їх роботи, а також можливість записатися на заняття (рис. 2.18-2.21);
- «Фитнес» - надає перехід до вікна з інформацією про фітнес, прайс, фотогалерея, розклад занять, а також можливість записатися на заняття (рис. 2.22-2.23);
- «СПА» - надає перехід до вікна з інформацією про СПА, прайс, фотогалерея, розклад часу, а також можливість записатися на заняття (рис. 2.24-2.26);
- «Новости» - надає перехід до вікна з інформацією про новини та останні події спортивного комплексу (рис. 2.27-2.28);
- «Контакты» - надає перехід до вікна, де містить форму «Напиши нам», а також містить карту розташування спортивного комплексу в місті Дніпро, відгуки і можливість його залишити (рис. 2.29-2.31).

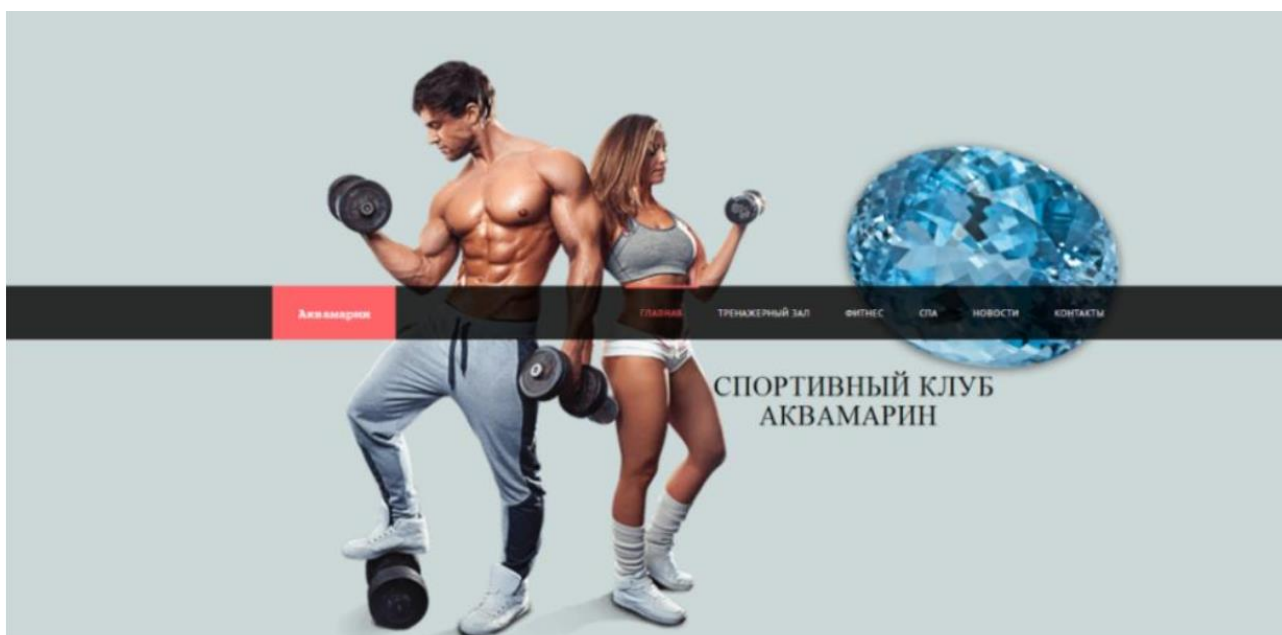
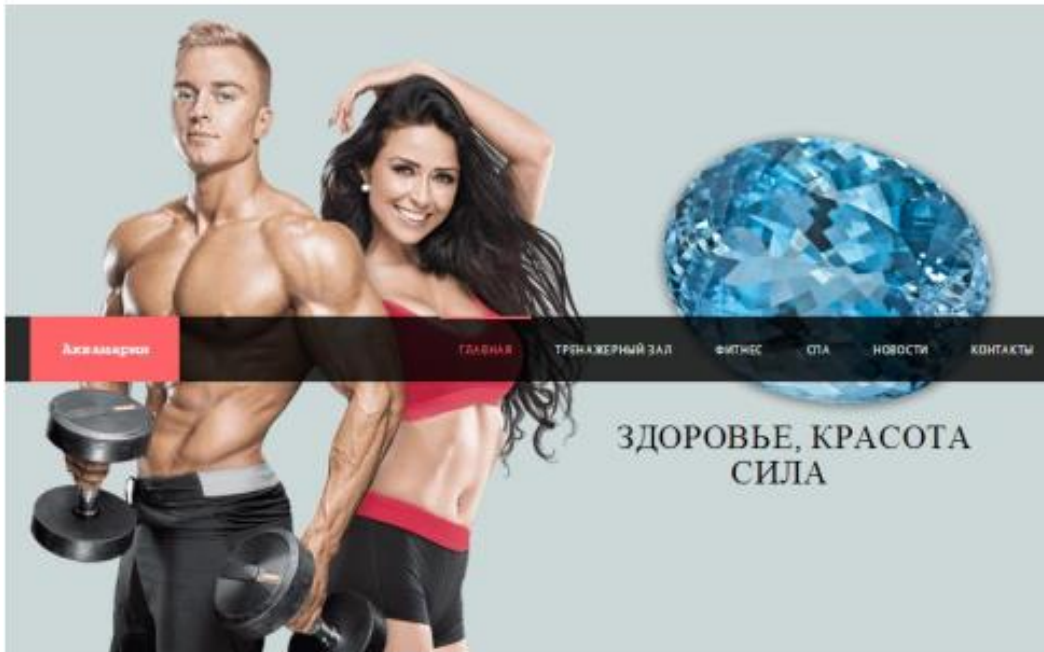


Рис. 2.13. Сторінка «Головна»(частина 1)



ДОБРО ПОЖАЛОВАТЬ В АКВАМАРИН



Спортивный клуб АКВАМАРИН – это уникальный комплекс, где под одной крышей собрано все необходимое для занятия спортом и поддержания своего тела в идеальной форме.

Наш клуб способен удовлетворить желания и потребности каждого: от ребенка до опытного и профессионального спортсмена. К Вашим услугам: тренажерный зал с современными тренажерами, фитнес зал для групповых тренировок и СПА.

ВРЕМЯ РАБОТЫ:

08:00 - 23:00



НАШИ ТРЕНЕРЫ

Евгений Пасечный

Тренажерный зал



Владимир Прокопенко

Тренажерный зал



Анна Сорокина

Фитнес



Анатолий Фоменко

Фитнес



Рис. 2.14. Страница «Главная»(часть 2)

Якщо навести на тренера курсор миші, то можемо побачити стаж роботи тренера (рис. 2.15).

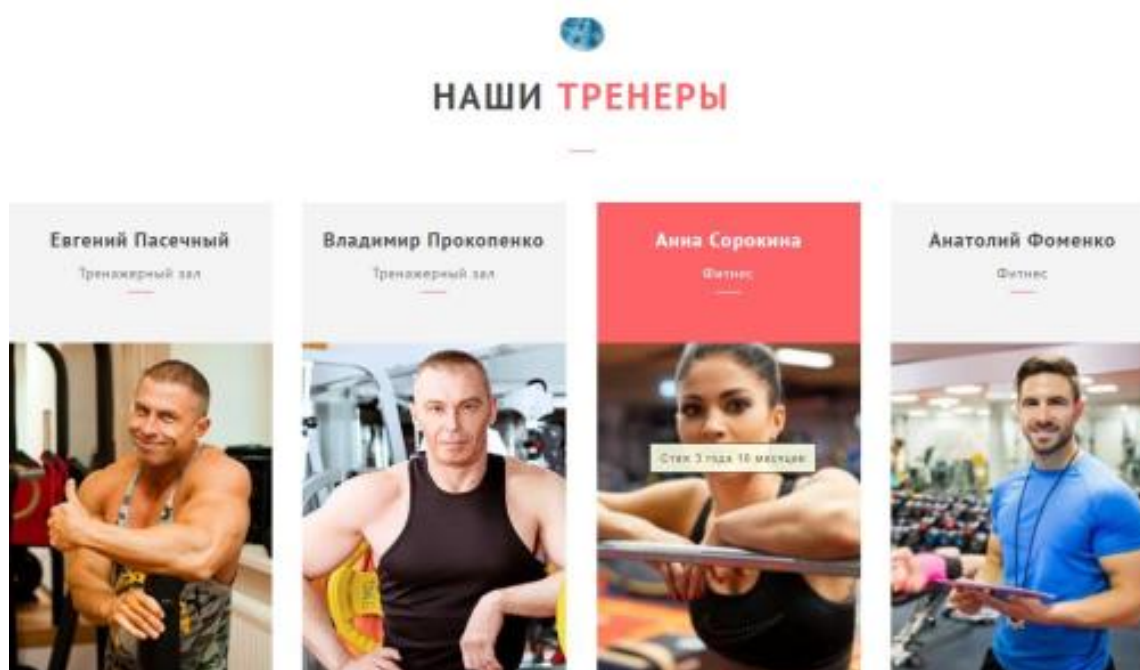


Рис. 2.15. Сторінка «Головна»(частина 3)

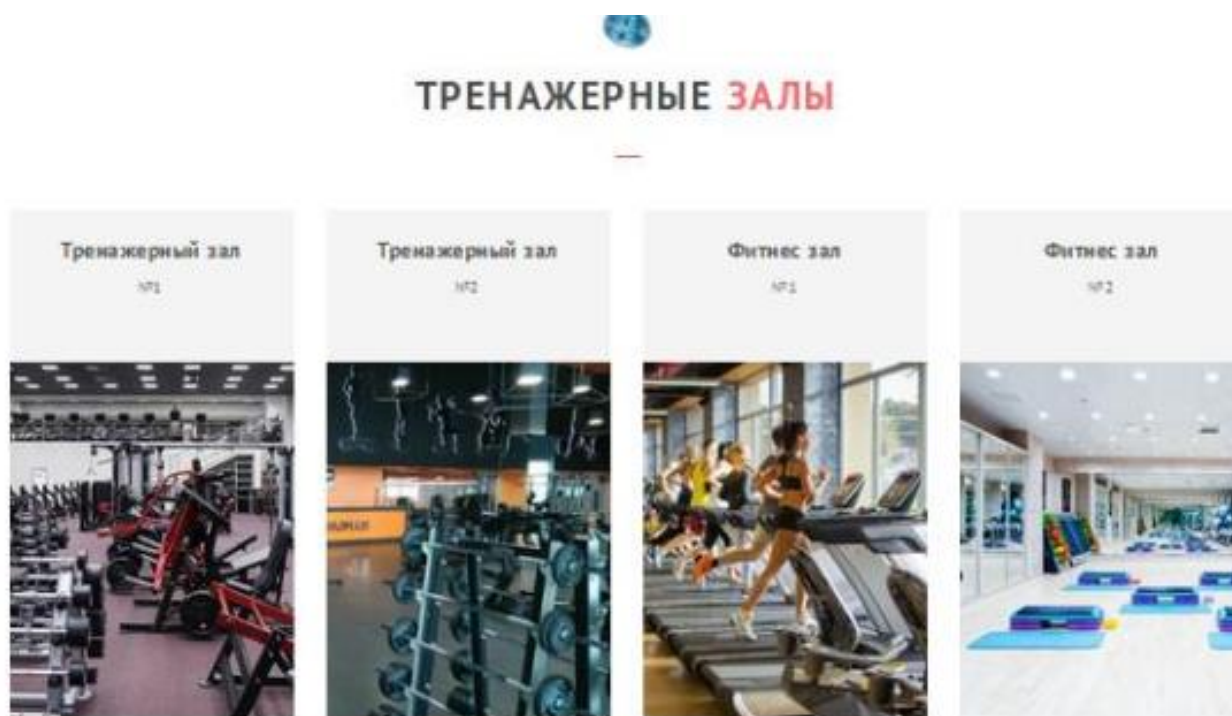


Рис. 2.16. Сторінка «Головна»(частина 4)



Рис. 2.17. Сторінка «Головна» (частина 5)

При виборі користувачем тренажерного залу, він автоматично переходить на відповідну сторінку веб-сайту, де користувач може побачити інформаційний блок, прайс, фотогалерею та записатися на заняття. Аналогічно відбувається і з вибором спа залу (рис. 2.18-2.20).

ТРЕНАЖЕРНЫЙ ЗАЛ

Главная > Тренажерный зал

ОСНОВНАЯ ИНФОРМАЦИЯ



Общая площадь тренажерного зала — 1000 м². Зал разделен на 5 зон: кардиотеатр, свободные веса, грузоблочные тренажеры, зону функционального тренинга и зона CrossTraining. В зале предусмотрено все для эффективных тренировок: широкий ассортимент кардиотренажеров, популярные тренажеры дублируются, полная линейка свободных весов.

Тренажерный зал

Рис. 2.18. Сторінка «Тренажерний зал» (частина 1)

Тренажеры в тренажерном зале – это то ради чего многие приходят в спортивный клуб. Сложные упражнения в тренажерном зале не только укрепляют мускулатуру и делают организм более выносливым, но и развивают способность быстрее восстанавливаться после нагрузок повседневной жизни, помогают предотвратить возрастные изменения.

Зал для силовых тренировок в СК «Аквамарин» оснащен самыми современными тренажерами от известных производителей спортивного оборудования «VASIL» и «VADZAAR». Здесь есть все, что необходимо для того, чтобы основательно задействовать все группы мышц. Помимо разнообразия силовых тренажеров присутствует кардио-зона, которая оборудована велотренажерами Johnson AeroFit, велотренажерами, эллипсоидами и степлерами Johnson.

Новичкам, а также тем, кто желает добиться максимального результата, стоит заниматься в тренажерном зале с тренером. Опытный специалист подберет для Вас подходящий комплекс упражнений, проследит за правильностью их выполнения и предоставит консультации и рекомендации. Для того, чтобы тренировки дали ожидаемый эффект, стоит учитывать множество нюансов, таких как цель тренировки, особенности телосложения, степень физической подготовки. Необходимо правильно распределять нагрузку и оценивать состояние организма, чтобы не возникло травм, растяжений и прочих неприятных последствий. Также важно уделять внимание питанию, ведь для набора мышечной массы и для избавления от лишнего веса оно будет отличаться. Тут тоже личный тренер сможет помочь и дать необходимые рекомендации.

Кроме непосредственно зала для занятий спортом здесь есть мужская и женская раздевалки, оборудованные персональными шкафчиками и душевыми кабинками с холодной и горячей водой. Также Вы можете посетить наш спортивный бар.

Дневная программа:

- ✓ Разовый _____ 100 грн.
- ✓ Абонемент СТУДЕНТ (ДНЕВНОЙ) _____ 400 грн.
- ✓ Абонемент «Выходного дня» _____ 400 грн.
- ✓ Месяц _____ 600 грн.
- ✓ Год _____ 4 200 грн.

В стоимость входит: Вход в клуб до 15:00 без ограничений по дням недели; Участие в клубных мероприятиях; 1 индивидуальное занятие с тренером в тренажерном зале.



Полная программа:

- ✓ Разовый _____ 100 грн.
- ✓ Месяц _____ 600 грн.
- ✓ Год _____ 4 800 грн.

В стоимость входит: Вход в клуб без ограничений времени и по дням недели; Участие в клубных мероприятиях; 1 индивидуальное занятие с тренером в тренажерном зале; Для этого пакета предусматривается возможность приостановки срока действия Клубной карты.



ФОТОГАЛЕРЕЯ



РАСПИСАНИЕ

Тренажерный зал	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
08:00-09:00	+	+	+	+	+		
09:00-10:00	+	+	+	+	+		
10:00-11:00	+	+	+	+	+	+	+
11:00-12:00	+	+	+	+	+	+	+
12:00-13:00	+	+	+	+	+	+	+
13:00-14:00	+	+	+	+	+	+	+
14:00-15:00	+	+	+	+	+	+	+
15:00-16:00	+	+	+	+	+	+	+
16:00-17:00	+	+	+	+	+	+	+
17:00-18:00	+	+	+	+	+	+	+
18:00-19:00	+	+	+	+	+	+	+
19:00-20:00	+	+	+	+	+		
20:00-21:00	+	+	+	+	+		
21:00-22:00	+	+	+	+	+		
22:00-23:00	+	+	+	+	+		

Рис. 2.19. Страница «Тренажерный зал»(часть 2)

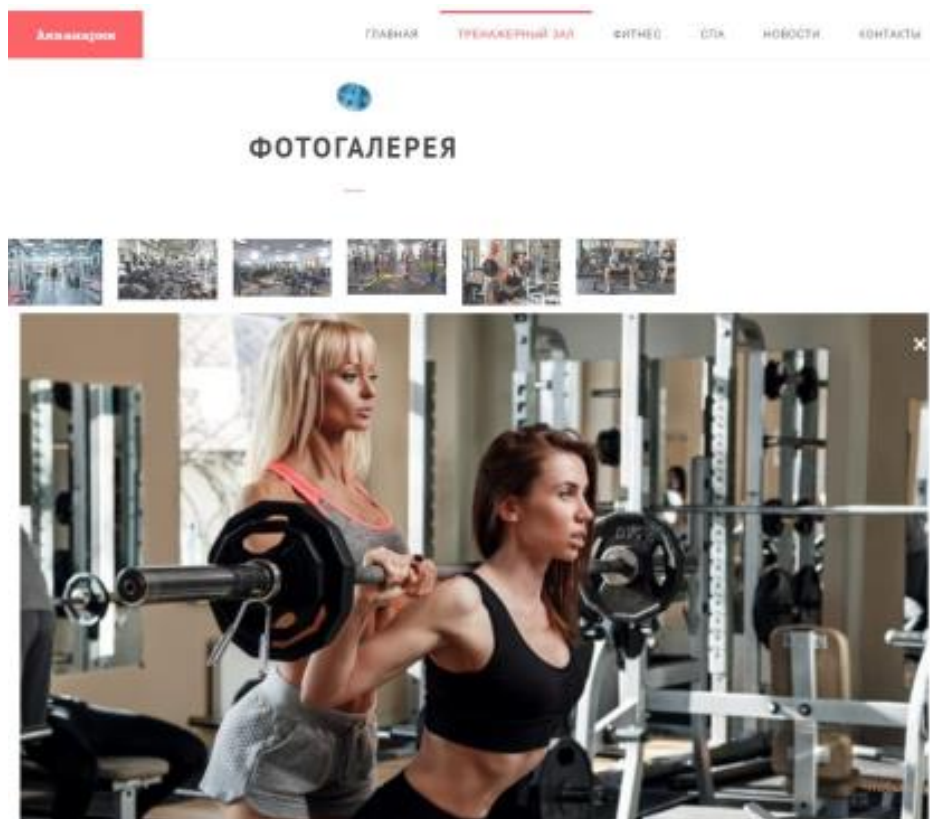


Рис. 2.20. Сторінка «Тренажерный зал»(частина 3)

На сторінці сайту «Тренажерный зал», наявна «Форма зворотного зв'язку», яка знаходиться в нижній її частині (рис. 2.21). Вона представляє собою форму з полями , де користувач вводить своє ПІБ, дату, час та назву заняття на яке хоче потрапити. Аналогічна «Форма зворотнього зв'язку»,наявна і на сторінках «Фитнес», «СПА».

ЗАПИШИСЬ СЕЙЧАС

Савченко Руслана Сергеевна	06.06.2020	18:00
----------------------------	------------	-------

Добрый день ,хотела бы записаться на пробное занятие к тренеру Владимиру.

ОТПРАВИТЬ

Рис. 2.21. Сторінка «Тренажерный зал» форма зворотного зв'язку

ФИТНЕС

Главная > Фитнес

ОСНОВНАЯ ИНФОРМАЦИЯ



Фитнес – лучшие программы и групповые тренировки от заслуженных мастеров и чемпионов приведут вас в идеальную физическую форму!

Фитнес

Кардио фитнес-программы, комплекс тренировок, направленный на улучшение работы сердца и кровеносной системы. Аэробика является необходимым элементом работы над телом, поскольку готовит сердце к тяжелой нагрузке.

Силовые фитнес-программы. Во время силовых тренировок эффективно прорабатываются все мышечные группы, повышается тонус мышц и развивается выносливость. Тренировки идеальны для сжигания калорий, приобретения упругого тела и отличной физической формы.

Индивидуальные занятия. Индивидуальные занятия с персональным тренером станут лучшим способом быстро и надежно прийти в спортивную форму. Тренер не только назначит программу упражнений, но и будет полностью сопровождать вас на пути к идеальному телу.

Групповые фитнес-программы. Уникальные групповые фитнес-программы с профессиональными тренерами станут залогом быстрого достижения результата. Тренировки направлены как на общий фитнес, так и на проработку конкретных зон.

Фитнес прайс:

- ✓ Разовый _____ 200 грн.
- ✓ Разовый (Индивидуальный) _____ 400 грн.
- ✓ Месяц _____ 900 грн.
- ✓ Год _____ 8 800 грн.

В стоимость входит: Вход в клуб без ограничений времени и по дням недели; Участие в клубных мероприятиях; Для этого пакета предусматривается возможность приостановки срока действия Клубной карты.



Рис. 2.22. Страница «Фитнес»(часть 1)

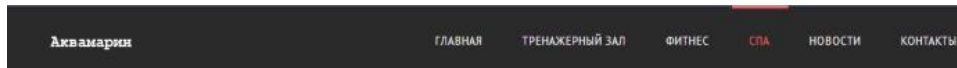
ФОТОГАЛЕРЕЯ



РАСПИСАНИЕ

Фитнес	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
08:00-09:00							
09:00-10:00							
10:00-11:00	Yoga			Yoga		Pilates	Pilates
11:00-12:00						TRX training	TRX training
12:00-13:00		Yoga			Yoga	Zumba	Zumba
13:00-14:00						Yoga	Yoga
14:00-15:00						TRX training	TRX training
15:00-16:00							
16:00-17:00	Pilates	Zumba	Pilates		Zumba	Pilates	Pilates
17:00-18:00		TRX training	Pilates	TRX training	Pilates	Yoga	Yoga
18:00-19:00	Yoga	Zumba	Yoga	Zumba		Zumba	Zumba
19:00-20:00	Yoga	Pilates	Yoga	Pilates			
20:00-21:00	Pilates	TRX training	Pilates	TRX training			
21:00-22:00							
22:00-23:00							

Рис. 2.23. Сторінка «Фітнес»(частина 2)



СПА

[Главная](#) > [СПА](#)

ОСНОВНАЯ ИНФОРМАЦИЯ



СПА (СПА) – это оздоровительный комплекс процедур с использованием воды, морских водорослей, целебных растений и лечебных грязей.

СПА

Давно известно, что физические нагрузки, дополненные спа-процедурами, дают потрясающий результат без излишней нагрузки на организм. Благодаря подобному сочетанию выводятся токсины и излишняя жидкость, мышцы, напряженные после интенсивной тренировки расслабляются и быстрее восстанавливаются. Быстрое снижение веса и тепло расслабляет

Рис. 2.24. Сторінка «СПА»(частина 1)

стройным и подтянутым, а настроение заметно улучшается.

СПА-зона в клубе Аквamarin оснащена финской сауной, которая делится на два помещения: на 2-3 человека и на 5-6 человек для небольшой дружеской компании. Помимо непосредственно парной, каждое оборудовано телевизором и комнатой отдыха. Сауна – отличный способ избавиться от токсинов и лишней жидкости, сохранить молодость кожи и улучшить состояние внутренних органов.

В СК «Аквamarin» можно записаться на сеанс или полный курс гидромассажа. Процедура проводится на высококачественном профессиональном оборудовании, а прием ведет специалист с большой буквы – настоящий мастер своего дела с двадцатилетним опытом работы. Гидромассаж – это не просто релакс, это еще и оздоровление организма и действенное средство против многих недугов. Особенно эффективен он в борьбе со стрессами, расстройствами сна, заболеваниями опорно-двигательного аппарата, последствиями травм.

Вертикальный солярий итальянского производства «J-MED 46 EXCELLENCE» позволяет получить равномерный бронзовый загар, подчеркивающий стройность фигуры без вреда для кожи.

СПА прайс:

- ✓ Финская сауна (1 час) _____ 200 грн.
- ✓ Гидромассаж (40 минут) _____ 300 грн.
- ✓ Инфракрасная кабина (30 минут) _____ 200 грн.
- ✓ Солярий (1 час) _____ 500 грн.



Спа-уход позволяет не только усилить эффект от занятий в тренажерном зале, но и практически остановить время, сохраняя молодость, красоту и отличную физическую форму на долгие годы.



ФОТОГАЛЕРЕЯ



Рис. 2.25. Сторінка «СПА»(частина 2)

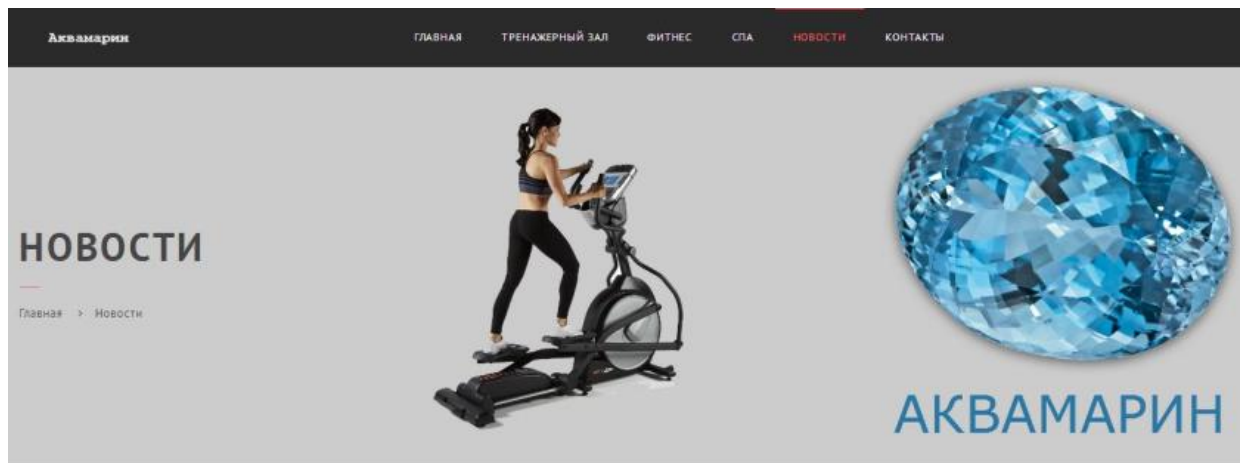


РАСПИСАНИЕ

SPA	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
08:00-09:00							
09:00-10:00						Солярий	Солярий
10:00-11:00						Гидромассаж	Гидромассаж
11:00-12:00						Солярий	Солярий
12:00-13:00	Солярий	Солярий	Солярий	Солярий	Солярий	Инфракрасная кабина	Инфракрасная кабина
13:00-14:00	Гидромассаж	Гидромассаж	Гидромассаж	Гидромассаж	Гидромассаж	Солярий	Солярий
14:00-15:00	Инфракрасная кабина		Инфракрасная кабина		Инфракрасная кабина	Солярий	Солярий
15:00-16:00						Гидромассаж	Гидромассаж
16:00-17:00						Солярий	Солярий
17:00-18:00						Солярий	Солярий
18:00-19:00	Солярий	Солярий	Солярий	Солярий	Солярий	Гидромассаж	Гидромассаж
19:00-20:00	Финская сауна	Финская сауна	Финская сауна	Финская сауна	Финская сауна	Солярий	Солярий
20:00-21:00	Инфракрасная кабина	Финская сауна	Инфракрасная кабина	Финская сауна	Инфракрасная кабина	Финская сауна	Финская сауна
21:00-22:00	Гидромассаж	Гидромассаж	Гидромассаж	Гидромассаж	Гидромассаж	Финская сауна	Финская сауна
22:00-23:00							

Рис. 2.26. Страница «СПА»(часть 3)

На следующей странице пользователь может посмотреть последние события и новости спортивного комплекса (рис. 2.27-2.28):



Акция!!!
14 ФЕВРАЛЯ. СЕАНС МАССАЖА В ПОДАРОК



Акция!!!
6 И 7 ЯНВАРЯ. ПОДАРКИ НА РОЖДЕСТВО + СЮРПРИЗ



Скидки!!!
С 1 ПО 7 ЯНВАРЯ. СКИДКИ НА ВСЕ УСЛУГИ СК "АКВАМАРИН"

Рис. 2.27. Страница «Новости»(часть 1)

Для клиентов с абонементом на год подарок ко Дню влюбленных - бесплатный массаж для 10 клиентов, которые первые запишутся.

7:30

01 февраля, 2021



Акция!!!/а>

С 1 ПО 15 ДЕКАБРЯ. СКИДКИ НА ГИДРОМАССАЖ ОТ 15%

Только с 1 по 15 декабря скидка на гидромассаж 15%! Приобретай подарочный сертификат на 2 сеанса и получи скидку 20%.

9:30

20 ноября, 2020

6 и 7 января скидки на солярий от 10% + приятный сюрприз! Успеете записаться порадовать себя на рождество!

10:00

01 января, 2021



Скидка!!!

7 И 8 ДЕКАБРЯ. ФИНСКАЯ БАНЯ СО СКИДКОЙ 10%!

7 и 8 декабря скидка на финскую баню 10%! Проведи выходные в нашем спортивном клубе и получи подарок!

8:45

17 ноября, 2020

С 1 по 7 января на все услуги СК "Акварин" скидки 25%. Купи абонемент для себя или в качестве подарка + подарок!

8:00

15 декабря, 2020



Акция!!!

20 НОЯБРЯ. СКИДКИ НА СОЛЯРИЙ ОТ 10%!

Только 20 ноября скидки на солярий от 10 до 40%! Порадуй себя красивым загаром!

9:30

15 ноября, 2020



Рис. 2.28. Сторінка «Новини»(частина 2)

На сторінці сайту «Контакты» наявна «Форма зворотного зв'язку», яка знаходиться на початку сторінки (рис. 2.29, 2.30) . Вона представляє форму з полями, де користувач додає ПІБ, електронну пошту, контактний телефон та своє запитання. Також дана сторінка має відгуки та «Форму зворотного зв'язку», яка знаходиться наприкінці сторінки. Вона представляє форму з полями для відгуків користувача(рис. 2.31). Відгук на сторінку сайту додається тільки тоді, коли адміністратор підтвердить даний відгук.

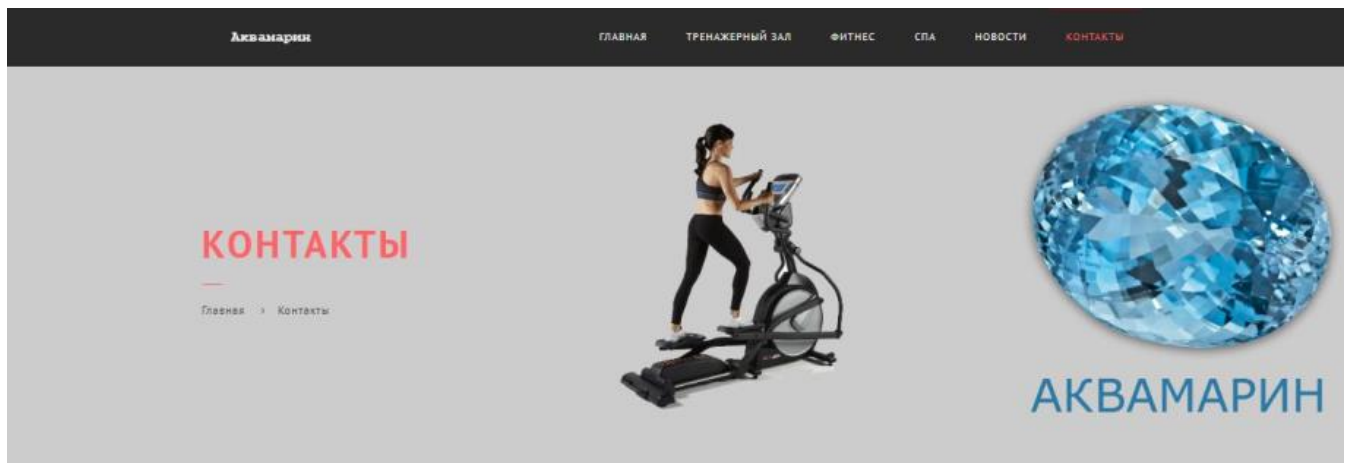


Рис. 2.29. Сторінка «Контакты»(частина 1)

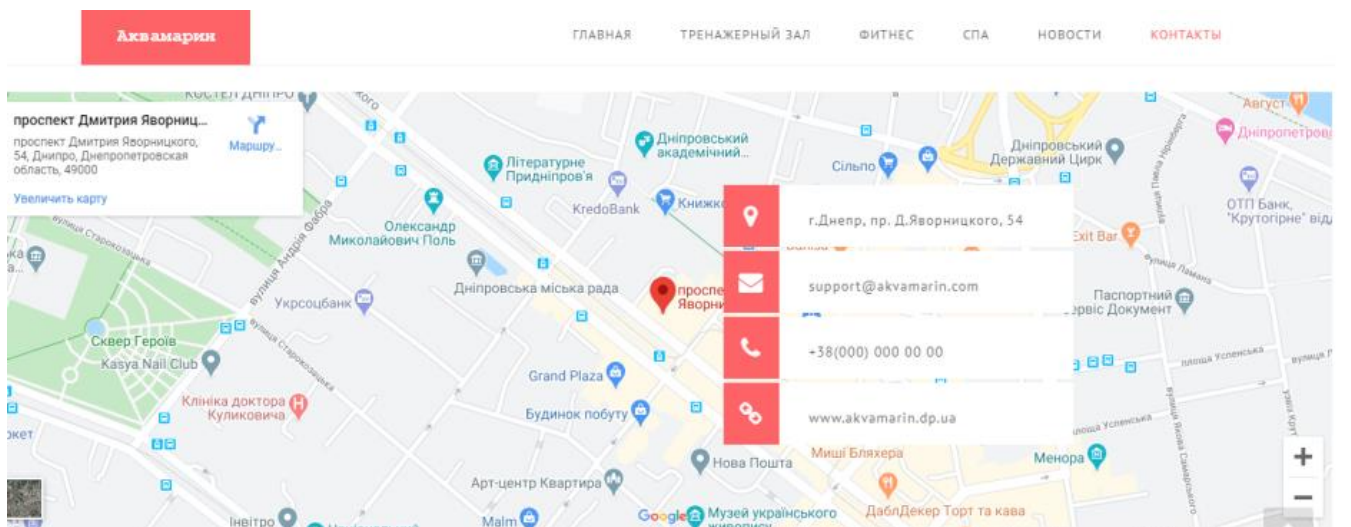


Рис. 2.30. Сторінка «Контакты»(частина 2)

ОТЗЫВЫ



Юлия 5 минут назад

Очень хороший зал, профессиональные тренеры

[← Ответить](#)

Админ 5 минут назад

Спасибо за отзыв

[← Ответить](#)

Олег 15 минут назад

Все понравилось! Спасибо за профессионализм!

[← Ответить](#)

ОСТАВЬТЕ СВОЙ ОТЗЫВ

Максим

max@gmail.com

Очень хороший зал, профессиональные тренеры|

СОХРАНИТЬ

Рис. 2.31. Страница «Контакты»(часть 3)

Для перехода на страницу административной части сайта необходимо в окне авторизации (рис. 2.32) ввести логин и пароль администратора, после чего сверху в браузере появится соответствующая консоль для перехода к административной части. Вход в консоль администратора осуществляется через кнопку в виде циферблата или текста рядом с ней (рис. 2.33).



Рис. 2.32. Вікно авторизації адміністратора

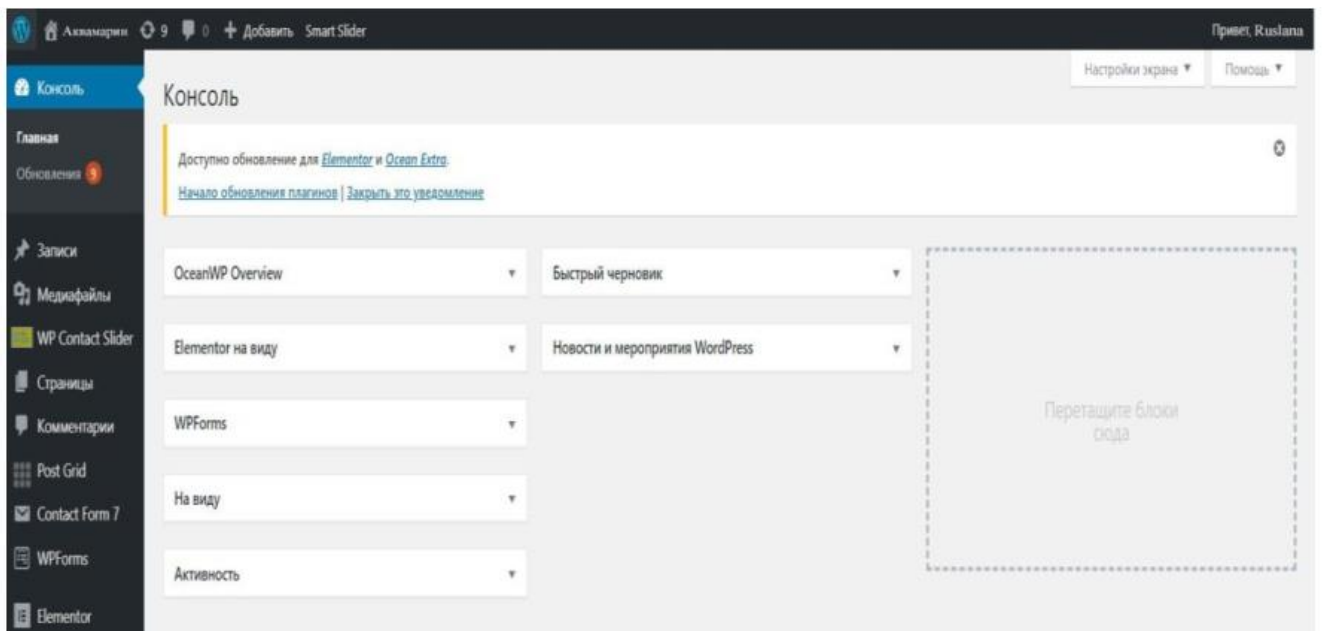


Рис. 2.33. Вікно адміністрування сайтом

Для адміністрування сайту використовуються такі сторінки як:

- «Записи» – необхідне для створення, редагування та видалення нових записів на сторінці (рис. 2.34);
- «Шаблони» – необхідне для створення, редагування, видалення шаблонів (рис. 2.35);
- «Страницы» - для представлення існуючих сторінок та створення нових (рис. 2.36);
- «Зовнішній вигляд» – додавання або видалення готових шаблонів сайту або створення своїх шаблонів, які були зроблені власноруч (рис. 2.37);
- «Налаштування» – налаштування панелі адміністрування або самого сайту (рис. 2.38).

Для додавання, редагування або видалення новин, потрібно зайти до панелі адміністрування та у блоці «Записи» натиснути на потрібний запис або створити новий. Створення нового запису наведено на рис. 2.35.



Рис. 2.34. Вкладка «Записи»

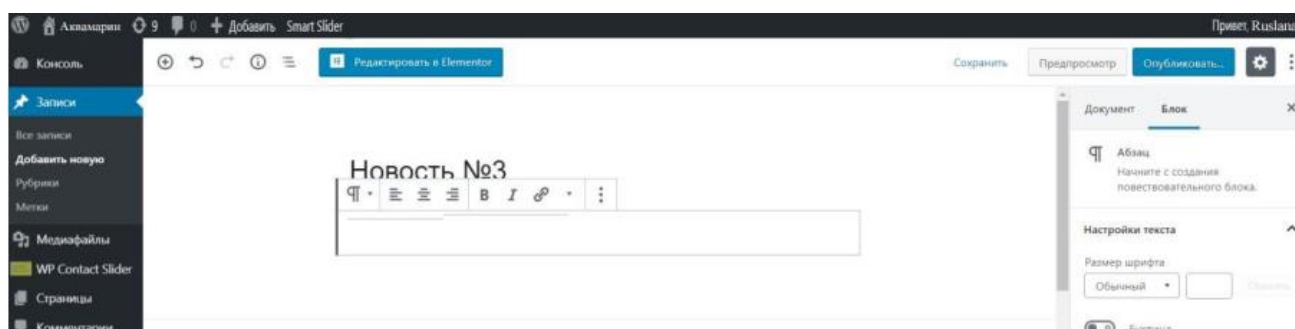


Рис. 2.35. Форма додавання запису «Новости»

Для створення сторінок веб-застосування потрібно перейти в консолі через вкладку «Страницы» на основне меню сторінок, де можна переглянути таблицю з усіма існуючими сторінками та скористатись потрібними функціональними кнопками (рис. 2.35). Для створення нової таблиці потрібно використати відповідну кнопку «Добавить новую», після чого в редакторі додати для неї необхідні властивості (рис. 2.36).

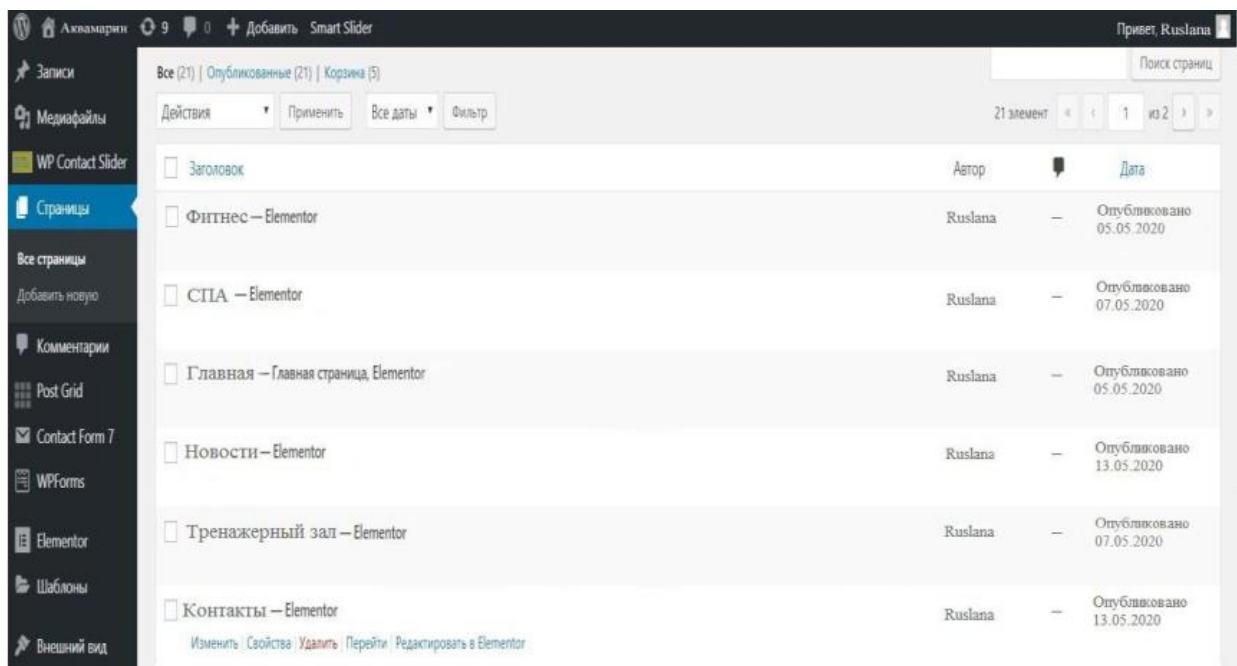


Рис. 2.36. Вкладка «Сторінки»

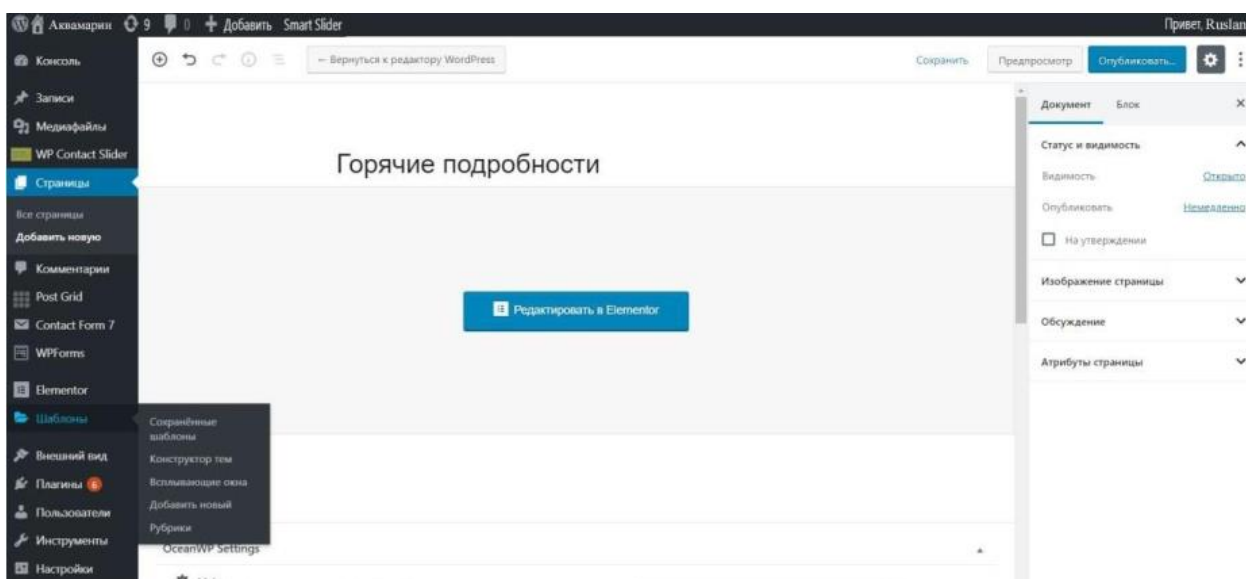


Рис. 2.37. Вкладка «Сторінки», створення нової сторінки

Для створення нового шаблону потрібно перейти в консолі через вкладку «Шаблони» на основне меню шаблонів, де можна переглянути таблицю з усіма існуючими шаблонами та скористатись потрібними функціональними кнопками (рис. 2.38). Для створення нового шаблону потрібно використати відповідну кнопку «Добавить новый», після чого в редакторі додати потрібні властивості та елементи (рис. 2.39).

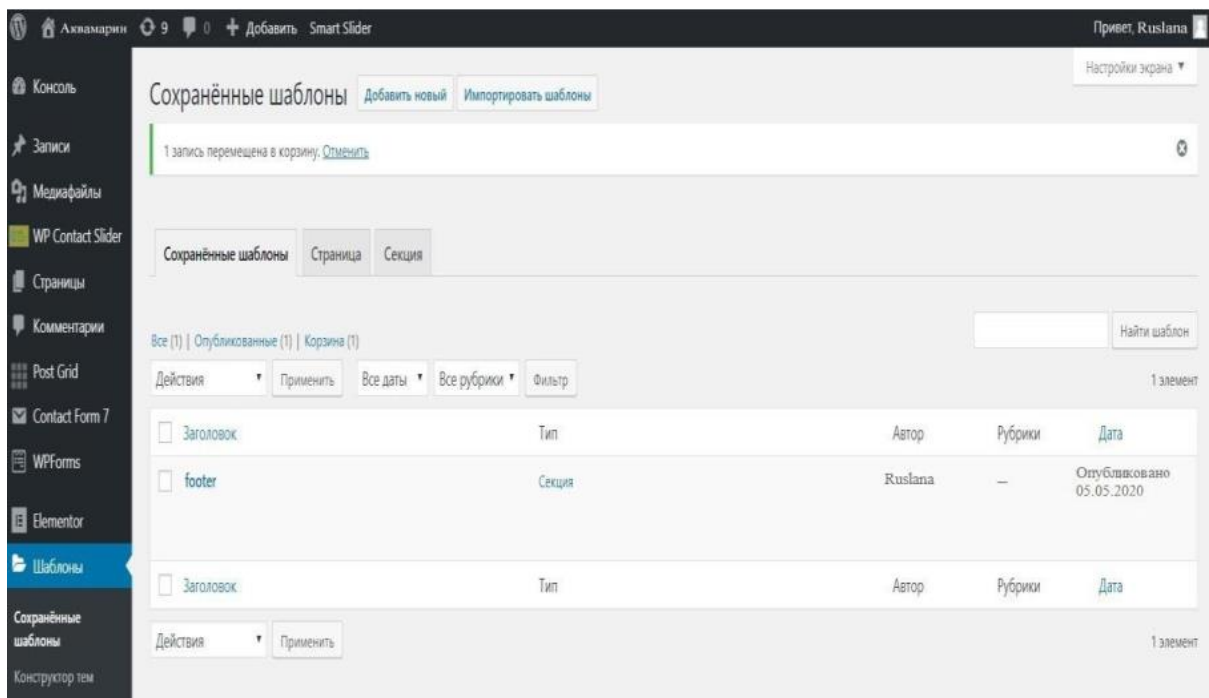


Рис. 2.38. Вкладка «Шаблони»

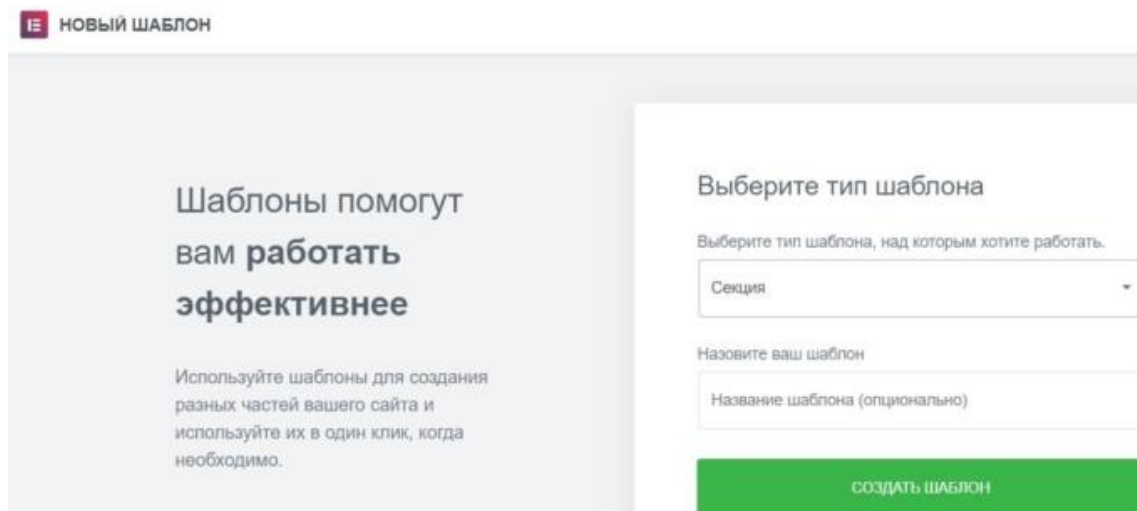


Рис. 2.39. Вкладка «Шаблони», створення нового шаблону

Додавання теми може бути реалізовано двома способами. Можна додати із сховища або завантажити теми з комп'ютера. Для завантаження теми з сховища потрібно перейти на вкладку «Внешний вид» (рис. 2.40) та натиснути кнопку «Добавить новую». Після виконання цих кроків перед користувачем відкриється вікно з різноманітними темами. Переглянувши декілька тем, можна обрати одну з них, завантажити її та застосувати потрібні дії (рис. 2.41).

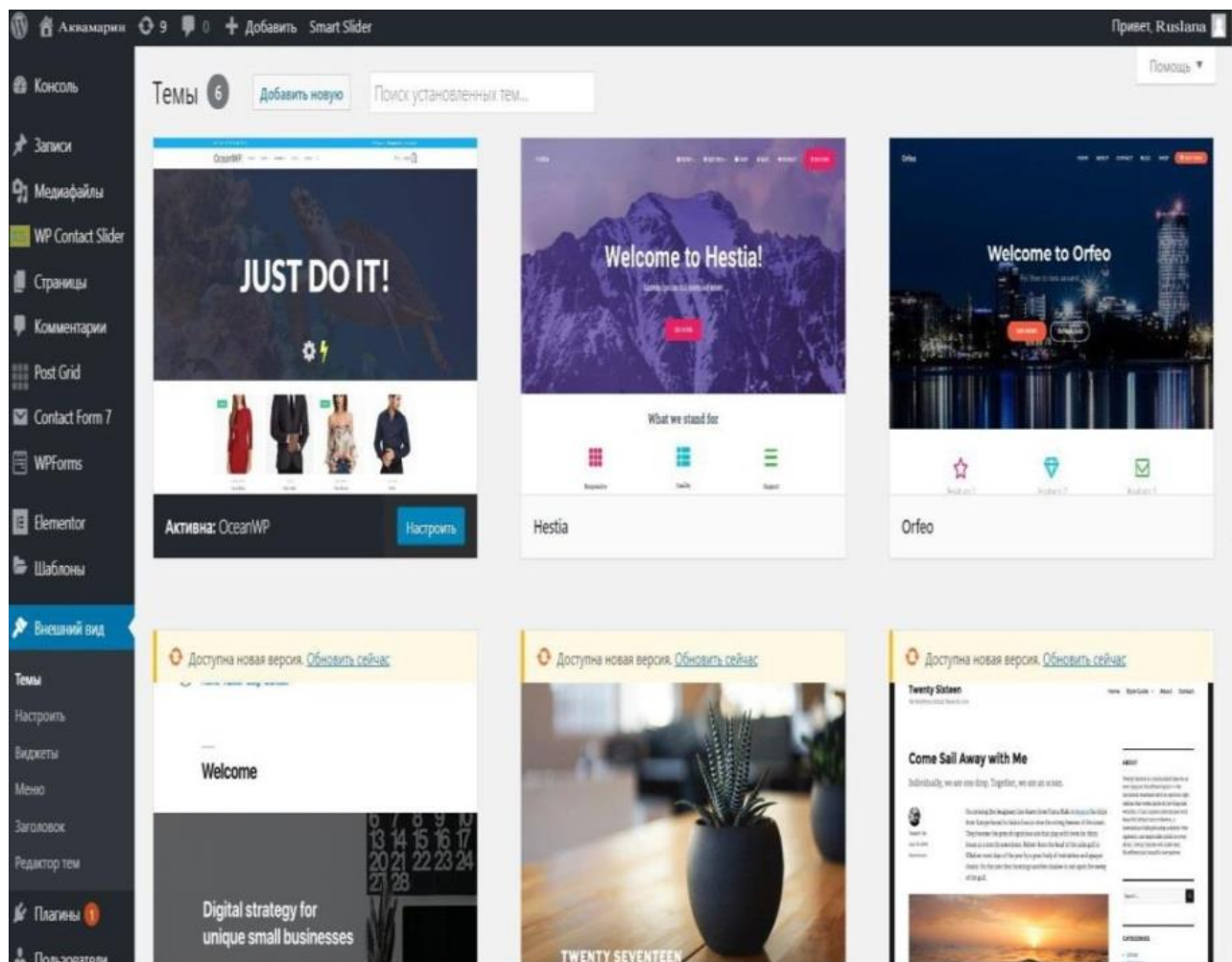


Рис. 2.40. Вкладка «Зовнішній вигляд»

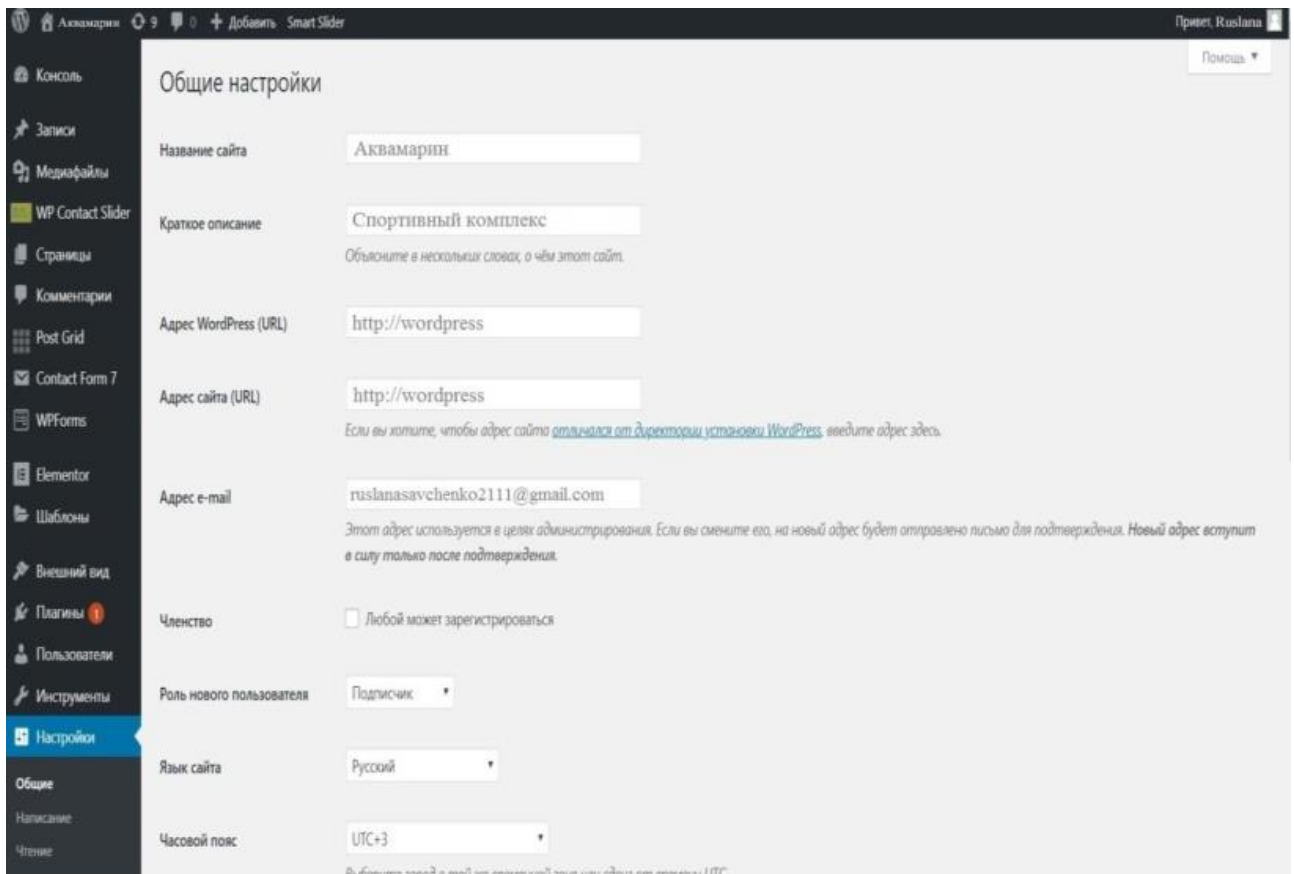


Рис. 2.41. Вкладка «Налаштування»

2.6.4.2. Тестування додатку

Після реалізації розробленого сайту було проведено повну перевірку функціонування сайту на роботу усіх стилів, функцій та зображень.

На головній сторінці перевірили функціонал JavaScript за допомогою допоміжного вікна браузера «DevTools», у вкладці «Network» обравши розділ «JS» можна сказати, що усі функції працюють (рис. 2.42).

Також було перевірено роботу CSS стилів та завантаженість усіх зображень (рис. 2.43 - 2.44).

Так як на усіх сторінках виконуються фактично однакові функції та використовуються більшість стилів, як на сторінці «Головна», то відповіді з тестування на усіх інших сторінках однакові.

Після перевірки помилок сайту на test-case було зроблено висновок, що сайт працює справно.

Name	▲	Status	Type	Initiator
bootstrap.min.js		Finished	script	index.html
hoverIntent-r7.min.js		Finished	script	index.html
imagesloaded.pkgd.min.js		Finished	script	index.html
isotope-2.0.0.min.js		Finished	script	index.html
jquery-migrate.min.js		Finished	script	index.html
jquery.cookie.js		Finished	script	index.html
jquery.js		Finished	script	index.html
jquery.themepunch.revolu...		Finished	script	index.html
jquery.themepunch.tools....		Finished	script	index.html
jquery.touchSwipe.js		Finished	script	index.html
main.js		Finished	script	index.html
modernizr-2.7.1.min.js		Finished	script	index.html
superfish-1.7.4.min.js		Finished	script	index.html

Рис. 2.42. Перевірка працездатності JS сторінки «Головна»

Name	▲	Status	Type	Initiator
css?family=PT+Sans:100,3...		200	styles...	css
css?family=Quicksand:300...		200	styles...	css
default.css		Finished	styles...	index.html
font-awesome.min.css		Finished	styles...	index.html
main.css		Finished	styles...	index.html
settings.css		Finished	styles...	index.html

Рис. 2.43. Перевірка працездатності CSS сторінки «Головна»
















Name	Status	Type	Initiator
 logo.png	Finished	png	index.html
 logo-retina.png	Finished	png	index.html
 transparent.png	Finished	png	index.html
 slideshow1_img.png	Finished	png	index.html
 logo-slideshow1.png	Finished	png	index.html
 slideshow1_img_2.png	Finished	png	index.html
 time.png	Finished	png	index.html
 1.png	Finished	png	index.html
 2.png	Finished	png	index.html
 3.png	Finished	png	index.html
 4.png	Finished	png	index.html
 zal1.png	Finished	png	index.html
 zal2.png	Finished	png	index.html
 zal3.png	Finished	png	index.html
 zal4.png	Finished	png	index.html

Рис. 2.44. Завантаженість зображень на сторінці «Головна»

РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

- передбачуване число операторів - 1500;
- коефіцієнт складності програми - 2;
- коефіцієнт корекції програми в ході її розробки - 0,08;
- годинна заробітна плата програміста, грн./год - 30.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (50),

t_i – витрати праці на дослідження алгоритму рішення задачі,

t_a – витрати праці на розробку блок-схеми алгоритму,

t_n – витрати праці на програмування по готовій блок-схемі,

t_{otl} – витрати праці на налагодження програми на ЕОМ,

t_d – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \times C \times (1 + p), \text{ людино-годин,} \quad (3.2)$$

де q - передбачуване число операторів,

C - коефіцієнт складності програми;

p - коефіцієнт кореляції програми в ході її розробки.

$$Q = 1500 \cdot 2 \cdot (1 + 0,08) = 3240 \text{ людино-годин.} \quad (3.3)$$

Витрати праці на вивчення опису задачі ти визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{QB}{(75 \dots 85)K}, \text{ людино-годин,} \quad (3.4)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі; $B=1.2 \dots 1.5$,

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Витрати праці на розробку алгоритму рішення задачі:

$$t_u = \frac{3240 \cdot 1,3}{80 \cdot 1,2} = 44, \text{ людино-годин.} \quad (3.5)$$

Витрати на складання програми по готовій блок-схемі:

$$t_a = \frac{Q}{(20 \dots 25)K} \text{ людино-годин.} \quad (3.6)$$

$$t_a = \frac{3240}{22 \cdot 1,2} = 123 \text{ людино-годин.} \quad (3.7)$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25)K} \quad \text{людино-годин.} \quad (3.8)$$

$$t_n = \frac{3240}{23 \cdot 1,2} = 117 \quad \text{людино-годин.} \quad (3.9)$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{\text{отл}} = \frac{Q}{(4...5)K} \quad \text{людино-годин.} \quad (3.10)$$

$$t_{\text{отл}} = \frac{3240}{5 \cdot 1,2} = 540 \quad \text{людино-годин.} \quad (3.11)$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \quad \text{людино-годин,} \quad (3.12)$$

де $t_{\partial p}$ - трудомісткість підготовки матеріалів і рукопису.

$$t_{\partial p} = \frac{Q}{(15...20)K}, \quad \text{людино-годин.} \quad (3.13)$$

$$t_{\partial p} = \frac{3240}{18 \cdot 1,2} = 150 \quad \text{людино-годин,} \quad (3.14)$$

де $t_{\partial o}$ - трудомісткість редагування, печатки й оформлення документації.

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \quad \text{людино-годин.} \quad (3.15)$$

$$t_{\partial o} = 150 + 73 = 223 \quad \text{людино-годин.} \quad (3.16)$$

Отримуємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 44 + 123 + 117 + 540 + 223 = 1097 \quad \text{людино-годин.} \quad (3.17)$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми $Z_{зп}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{по} = Z_{зп} + Z_{мв}, \text{ грн.} \quad (3.18)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{зп} = t \cdot C_{спр}, \text{ грн,} \quad (3.19)$$

де t - загальна трудомісткість, людино-годин,

$C_{спр}$ - середня годинна заробітна плата програміста, грн/година.

$$Z_{зп} = 1097 \cdot 30 = 32910 \text{ грн.} \quad (3.20)$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{мв} = t_{отл} \times C_{м}, \text{ грн,} \quad (3.21)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год.,

$C_{м}$ - вартість машино-години ЕОМ, грн/год.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

$$Z_{мв} = 540 \times 5 = 2700 \text{ грн.}, \quad (3.22)$$

$$K_{по} = 2700 + 32910 = 35610 \text{ грн.} \quad (3.23)$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.}, \quad (3.24)$$

де B_k - число виконавців,

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

$$T = \frac{1097}{1 \cdot 176} = 6.23 \text{ міс.} \quad (3.25)$$

Проведено розрахунок вартості роботи по створенню програми (39610 грн) та розраховано час на його створення (6,23 міс).

ВИСНОВКИ

Завданням даної кваліфікаційної роботи є розробка веб-орієнтованого програмного забезпечення для спортивного комплексу, застосувавши засоби мов програмування Html, Css, JavaScript та PHP. Для цього спроектовано і розроблено веб-сторінку для спортивного комплексу «Аквамарин».

В роботі було досліджено та проаналізовано найбільш важливу і актуальну інформацію щодо розробки веб-ресурсів а саме основні принципи створення веб-ресурсів, їх структуру і функціональність, взаємодію основних компонентів. Також були розглянуті новітні та найбільш перспективні веб-технології, які з успіхом вже використовуються користувачами по всьому світі.

Розроблювана система забезпечує представлення різноманітної інформації про спортивний комплекс у мережі Інтернет клієнтам та звичайним користувачам а також забезпечує адміністрування роботи закладу.

Веб-сайт має дворівневий доступ: адміністратор та користувач, для кожного з яких передбачені відповідні функціональні можливості.

Результатом виконання даної роботи є повністю роботоспроможний веб-додаток для організації роботи спортивного закладу.

В економічному розділі проведено розрахунок вартості роботи по створенню програми (39610 грн.) та розраховано час на його створення (6,23 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Відомість про мови програмування, URL: <http://webstudio2u.net>.
2. Відомість про операційні системи, URL: [//operatsijna-sistema.webnode.com.ua](http://operatsijna-sistema.webnode.com.ua). дата звернення 14.04.2021
3. Відомість про браузері, URL: <https://biblprog.org.ua>. дата звернення 14.04.2021
4. Відомість про склад технічних систем, URL: <https://pbb.lvi.v.ua>. дата звернення 14.04.2021
5. Э. Фримен, Э. Фримен Изучаем HTML, XHTML и CSS. Head First HTML with CSS & XHTML. - П.: «Питер», 2010. - 656 с.
6. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. HTML, XHTML & CSS For Dummies, 7th Edition. - М.: «Диалектика», 2011. - 400 с.
7. Питер Лабберс, Брайан Олберс, Фрэнк Салим. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. Pro HTML5 Programming: Powerful APIs for Richer
8. Internet Application Development. - М.: «Вильямс», 2011. - 272 с.
9. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание. HTML, XHTML, and CSS Bible, 5th Edition. - М.: «Диалектика», 2010. - 656 с.
10. Кроудер, Дэвид. Создание веб-сайта для чайников, 3-е издание. Москва: издательство “Диалектика”, 2009.
11. Структура Web-приложений – URL :<https://www.intuit.ru/studies/courses/1139/250/lecture/6422> дата звернення 14.04.2021
12. Верстка веб-приложений- URL: <https://semantica.in/blog/verstka-veb-sajtov.htm>. дата звернення 14.04.2021
13. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення : ДСТУ 3008-95. – Чинний від 1996–01–01. – К. : Держстандарт України, 1996. – 39 с.

14. ДСТУ 2394-94 Інформація та документація. Комплектування фонду, бібліографічний опис, аналіз документів. Терміни та визначення. – Чинний від 01.01.1995. - Київ : Держстандарт України, 1994. – 88 с.

15. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Комп’ютерні системи ” / Укладачі О.Г. Вагонова, Нікітіна О.Б. Н.Н. Романюк – Дніпропетровськ: Національний гірничий університет. – 2013. – 23с.

16. Методичні рекомендації до виконання кваліфікаційних робіт бакалаврів напряму підготовки 122 «Комп’ютерні науки» галузі знань 12 Інформаційні технології/, Л.М. Коротенко , О.С. Шевцова; Нац. гірн. ун-т. – Д : ДВНЗ НГУ, 2019. – 65 с.

17. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. / В.Дронов. – СПб.: БХВ-Петербург, 2011. – 416 с.

18. HTML и CSS. Путь к совершенству. / Б. Хеник – СПб.: Питер, 2011. – 336 с.

19. HTML и XHTML. Подробное руководство. Шестое издание. / Ч. Муссиано, Б. Кеннеди. – Издательство: Символ-Плюс, 2011. – 752 с.

20. JavaScript: Подробное руководство (Definitive Guide). / D. Flanagan. - Издательство: Символ-Плюс, 2008. – 992 с.

КОД ПРОГРАМИ

```
Index. html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<head>
<meta charset="utf-8">
  <title>Вход - Аквамарин - спортивный комплекс</title>
  <meta name="robots" content="index, follow">
  <link href="http://sk-aquamarin.vn.ua/theme/sk-aquamarin/style.css"
rel="stylesheet">
  <script src="http://sk-aquamarin.vn.ua/theme/sk-aquamarin/flash/swfobject.js"
type="text/javascript"></script>
  <script type='text/javascript'>
var flashvars={m:""};
var params={wmode:"opaque", scale:"scale", allowScriptAccess:"always",
bgcolor:"#0098d1"}; //transparent opaque
var attributes={};
swfobject.embedSWF("http://sk-aquamarin.vn.ua/theme/sk-aquamarin/flash/vhod.swf",
"idFlash", "1100px", "750px", "9.0.0", "http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/flash/expressInstall.swf", flashvars, params, attributes);
</script>
  <meta name="description" content="" />
  <meta name="generator" content="GetSimple" />
  <link rel="canonical" href="http://sk-aquamarin.vn.ua/" />
</head>
<body>
<div style="position:absolute; display:none;">
<!--LiveInternet counter--><script type="text/javascript"><!--
new Image().src = "//counter.yadro.ru/hit?r"+
escape(document.referrer)+((typeof(screen)=="undefined")?"":
";s"+screen.width+"*"+screen.height+"*"+(screen.colorDepth?
screen.colorDepth:screen.pixelDepth))+";u"+escape(document.URL)+
";"+Math.random();//--></script><!--/LiveInternet-->
```

```

<table cellpadding="0" cellspacing="0" border="0" width="88" height="31" style="line-
height:0;width:88px;"><tr style="height:10px;">
    <td style="padding:0;width:38px;height:10px;"><a href="http://www.24log.de"
target="_blank"></a></td>
    <td style="padding:0;width:50px;height:10px;"><a href="http://www.russianwoman.ca"
target="_blank"></a></td></tr>
    <tr style="height:21px;"><td style="padding:0;width:38px;height:21px"><a
href="http://www.24log.ru" target="_blank"></a></td>
    <script type='text/javascript' language='javascript'>
        document.write('<td style="padding:0px;width:50px;height:21px;"><a
href="http://www.24log.ru/rating/rating.php?c=13" target="_blank"></a></td>');
    </script></tr></table><NOSCRIPT><a href="http://www.1russianbrides.com">Dating
single russian women</a></NOSCRIPT>
</div>
<div align="center"><!-- выравнивание всего сайта по центру -->
<table align="center" style="width:100%; height:100%;" cellpadding="0" cellspacing="0">
    <tr align="center" style="width:100%;" valign="middle">
        <td align="center" style="width:100%;" valign="middle">
            <table style="height:100%; width:100%; max-width:2000px;" cellpadding="0"
cellspacing="0">
                <tr style="width:100%;">
                    <td style="background: url(http://sk-aquamarin.vn.ua/theme/sk-aquamarin/img/logo-
up.jpg) bottom no-repeat; width:100%;" colspan="3"></td>
                </tr>

```



```

        <tr align="center" style="height:750px; width:100%;">
            <td style="width:34%; background: url(http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/img/logo-left.jpg) right no-repeat;"></td>
                <td style="width:1100px;">
                    <!-- Контейнер СТАРТ ..... -->
                    <!-- enter START -->
                    <div style="position:absolute; z-index:1000;">
                        <a id="enter" href="main.html" title="Нажмите для входа на сайт">
                            
                        </a>
                    </div>
                    <!-- enter END -->
                    <div id="idFlash">
                        <div style="width:1100px; height:750px; background:url(http://sk-
aquamarin.vn.ua/theme/sk-aquamarin/img/logo-center.jpg) top center no-repeat;">
                            </div>
                        </div>

                    <!-- Контейнер ФИНИШ ..... -->
                    </td>
                <td style="width:34%; background: url(http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/img/logo-right.jpg) left no-repeat;"></td>
            </tr>
            <tr style="width:100%;">
                <td style="background: url(http://sk-aquamarin.vn.ua/theme/sk-aquamarin/img/logo-
down.jpg) top no-repeat; width:100%;" colspan="3"></td>
            </tr>
        </table>
    </td>
</tr>
</table>
</div><!-- выравнивание всего сайта по центру -->
</body>
</html>

```

Main.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta charset="utf-8">
  <title>Главная - Аквамарин - спортивный комплекс</title>
  <meta name="robots" content="index, follow">
  <link href="http://sk-aquamarin.vn.ua/theme/sk-aquamarin/style.css"
rel="stylesheet">

  <script type="text/javascript" src="highslide/highslide-full.js"></script>
  <script type="text/javascript" src="highslide/highslide.config.js" charset="utf-8"></script>
  <link rel="stylesheet" type="text/css" href="highslide/highslide.css" />
  <!--[if lt IE 7]>
  <link rel="stylesheet" type="text/css" href="highslide/highslide-ie6.css" />
  <![endif]-->
  <script src="http://sk-aquamarin.vn.ua/theme/sk-aquamarin/flash/swfobject.js"
type="text/javascript"></script>
  <script type='text/javascript'>
    var flashvars={m:""};
    var params={wmode:"opaque", scale:"scale", allowScriptAccess:"always",
bgcolor:"#34471C"}; //transparent opaque
    var attributes={};
    swfobject.embedSWF("http://sk-aquamarin.vn.ua/theme/sk-aquamarin/flash/main.swf",
"idFlash", "1100px", "750px", "9.0.0", "http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/flash/expressInstall.swf", flashvars, params, attributes);
  </script>
  <meta name="description" content="" />
  <meta name="generator" content="GetSimple" />
  <link rel="canonical" href="http://sk-aquamarin.vn.ua/main.html" />
</head>
<body>
<div style="position:absolute; display:none;">
  <!--LiveInternet counter--><script type="text/javascript"><!--
```

```

new Image().src = "//counter.yadro.ru/hit?r"+
escape(document.referrer)+((typeof(screen)=="undefined")?"":
";s"+screen.width+"*"+screen.height+"*"+(screen.colorDepth?
screen.colorDepth:screen.pixelDepth))+";u"+escape(document.URL)+
";"+Math.random();!--></script><!--/LiveInternet-->
<table cellpadding="0" cellspacing="0" border="0" width="88" height="31" style="line-
height:0;width:88px;"><tr style="height:10px;">
<td style="padding:0;width:38px;height:10px;"><a href="http://www.24log.de"
target="_blank"></a></td>
<td style="padding:0;width:50px;height:10px;"><a href="http://www.russianwoman.ca"
target="_blank"></a></td></tr>
<tr style="height:21px;"><td style="padding:0;width:38px;height:21px"><a
href="http://www.24log.ru" target="_blank"></a></td>
<script type='text/javascript' language='javascript'>
document.write('<td style="padding:0px;width:50px;height:21px;"><a
href="http://www.24log.ru/rating/rating.php?c=13" target="_blank"></a></td>');
</script></tr></table><NOSCRIPT><a href="http://www.1russianbrides.com">Dating
single russian women</a></NOSCRIPT>
</div>
<div align="center"><!-- выравнивание всего сайта по центру -->
<table align="center" style="width:100%; height:100%;" cellpadding="0" cellspacing="0">
<tr align="center" style="width:100%;" valign="middle">
<td align="center" style="width:100%;" valign="middle">

```

```

<table style="height:100%; width:100%; max-width:2000px;" cellpadding="0"
cellspacing="0">
  <tr style="width:100%;">
    <td style="background: url(http://sk-aquamarin.vn.ua/theme/sk-aquamarin/img/main-
up.jpg) bottom no-repeat; width:100%;" colspan="3"></td>
  </tr>
  <tr align="center" style="height:750px; width:100%;">
    <td style="width:34%; background: url(http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/img/main-left.jpg) right no-repeat;"></td>
    <td style="width:1100px; height:750px;">
      <!-- Контейнер СТАРТ ..... -->
      <div style="position:absolute; top:50%"></div>
      <!-- Flash Меню СТАРТ ..... -->
      <a style="position:absolute; left:50%; top:50%; margin-left:-350px; margin-top:-50px;
visibility:hidden;" id="info" href="info.html" onclick="return hs.htmlExpand(this, { objectType:
'ajax', width: 210} )" title="Инфо">Инфо</a>
      <a style="position:absolute; left:50%; top:50%; margin-left:-380px; margin-top:10px;
visibility:hidden;" id="akcii" href="akcii.html" onclick="return hs.htmlExpand(this, { objectType:
'ajax', width: 310} )" title="Акции">Акции</a>
      <a style="position:absolute; left:50%; top:50%; margin-left:-420px; margin-top:76px;
visibility:hidden;" id="gb" href="gb/" onClick="parent.location.href='gb/'"
title="Отзывы">Отзывы</a>

      <a style="position:absolute; left:50%; top:50%; margin-left:-420px; margin-top:140px;
visibility:hidden;" id="contact" href="contact.html" onclick="return hs.htmlExpand(this, {
objectType: 'ajax', width: 270} )" title="Контакты">Контакты</a>
      <a style="position:absolute; left:50%; top:50%; margin-left:-380px; margin-top:200px;
visibility:hidden;" id="raspisanie" href="raspisanie.html" onclick="return hs.htmlExpand(this, {
objectType: 'ajax', width: 900} )" title="Расписание">Расписание</a>
      <a style="position:absolute; left:50%; top:50%; margin-left:-240px; margin-top:250px;
visibility:hidden;" id="3d_pano" href="3d_pano.html"
onClick="parent.location.href='3d_pano.html'" title="3D Панорамы">3D Панорамы</a>
      <a style="position:absolute; left:50%; top:50%; margin-left:-40px; margin-top:-270px;
visibility:hidden;" id="silovye" href="silovye.html" onclick="return hs.htmlExpand(this, {
objectType: 'ajax', width: 250} )" title="Силовые направления">Силовые направления</a>

```

```
<a style="position:absolute; left:50%; top:50%; margin-left:210px; margin-top:-250px;
visibility:hidden;" id="fitnes" href="fitnes.html" onclick="return hs.htmlExpand(this, { objectType:
'ajax', width: 276} )" title="Фитнес направления">Фитнес направления</a>
```

```
<a style="position:absolute; left:50%; top:50%; margin-left:270px; margin-top:-100px;
visibility:hidden;" id="tancevalnye" href="tancevalnye.html" onclick="return hs.htmlExpand(this, {
objectType: 'ajax', width: 290} )" title="Танцевальные направления">Танцевальные
направления</a>
```

```
<a style="position:absolute; left:50%; top:50%; margin-left:250px; margin-top:60px;
visibility:hidden;" id="drugie" href="drugie.html" onclick="return hs.htmlExpand(this, {
objectType: 'ajax', width: 250} )" title="Другие услуги">Другие услуги</a>
```

```
<!-- Меню ФИНИШ ..... -->
```

```
<div align="center" style="width:1100px; height:750px;" id="idFlash">
  <p style="padding-top:200px;">
    <a href="http://www.adobe.com/go/getflashplayer"></a>
```

```
  </p>
  <div style="margin-top:50px;">
    <a href="info.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width: 250} )"
title="Инфо">Инфо</a><br>
```

```
    <a href="akcii.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width: 310}
)" title="Акции">Акции</a><br>
```

```
    <a href="gb/" title="Отзывы">Отзывы</a><br>
```

```
    <a href="contact.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width:
270} )" title="Контакты">Контакты</a><br>
```

```
    <a href="raspisanie.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width:
900} )" title="Расписание">Расписание</a><br>
```

```
    <a href="3d_pano.html" onClick="parent.location.href='3d_pano.html'" title="3D
Панорамы">3D Панорамы</a><br>
```

```
    <hr noshade="noshade" size="1" width="200" color="#4E6817">
```

```
    <a href="silovye.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width:
250} )" title="Силовые направления">Силовые направления</a><br>
```

```
    <a href="fitnes.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width: 276}
)" title="Фитнес направления">Фитнес направления</a><br>
```

```

    <a href="tancevalnye.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax', width:
276} )" title="Танцевальные направления">Танцевальные направления</a><br>
    <a id="drugie" href="drugie.html" onclick="return hs.htmlExpand(this, { objectType: 'ajax',
width: 250} )" title="Другие услуги">Другие услуги</a>
</div>
    <p>
    </p>
</div>
<!-- Контейнер ФИНИШ ..... -->
</td>
    <td style="width:34%; background: url(http://sk-aquamarin.vn.ua/theme/sk-
aquamarin/img/main-right.jpg) left no-repeat;"></td>
</tr>
<tr style="width:100%;">
    <td style="background: url(http://sk-aquamarin.vn.ua/theme/sk-aquamarin/img/main-
down.jpg) top no-repeat; width:100%;" colspan="3"></td>
</tr>
</table>
</td>
</tr>
</table>
</div><!-- выравнивание всего сайта по центру -->
</body>
</html>

```

Treni.html

```

(function(define, require, requireNative, requireAsync, exports, console, private,$Array,
$Function, $JSON, $Object, $RegExp, $String, $Error) {'use strict';// Copyright 2014 The
Chromium Authors. All rights reserved.

```

```

// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

```

```

// chrome.runtime.messaging API implementation.

```

```

// TODO(kalman): factor requiring chrome out of here.

```

```

var chrome = requireNative('chrome').GetChrome();

```

```

var Event = require('event_bindings').Event;

```

```

var lastError = require('lastError');
var logActivity = requireNative('activityLogger');
var logging = requireNative('logging');
var messagingNatives = requireNative('messaging_natives');
var processNatives = requireNative('process');
var utils = require('utils');
var messagingUtils = require('messaging_utils');
// The reserved channel name for the sendRequest/send(Native)Message APIs.
// Note: sendRequest is deprecated.
var kRequestChannel = "chrome.extension.sendRequest";
var kMessageChannel = "chrome.runtime.sendMessage";
var kNativeMessageChannel = "chrome.runtime.sendNativeMessage";
// Map of port IDs to port object.
var ports = {};
// Change even to odd and vice versa, to get the other side of a given
// channel.
function getOppositePortId(portId) { return portId ^ 1; }
// Port object. Represents a connection to another script context through
// which messages can be passed.
function PortImpl(portId, opt_name) {
  this.portId_ = portId;
  this.name = opt_name;
  var portSchema = {name: 'port', $ref: 'runtime.Port'};
  var options = {unmanaged: true};
  this.onDisconnect = new Event(null, [portSchema], options);
  this.onMessage = new Event(
    null,
    [{name: 'message', type: 'any', optional: true}, portSchema],
    options);
  this.onDestroy_ = null;
}
// Sends a message asynchronously to the context on the other end of this
// port.
PortImpl.prototype.postMessage = function(msg) {
  // JSON.stringify doesn't support a root object which is undefined.

```

```

if (msg === undefined)
  msg = null;
msg = $JSON.stringify(msg);
if (msg === undefined) {
  // JSON.stringify can fail with unserializable objects. Log an error and
  // drop the message.
  //
  // TODO(kalman/mpcomplete): it would be better to do the same validation
  // here that we do for runtime.sendMessage (and variants), i.e. throw an
  // schema validation Error, but just maintain the old behaviour until
  // there's a good reason not to (http://crbug.com/263077).
  console.error('Illegal argument to Port.postMessage');
  return;
}
messagingNatives.PostMessage(this.portId_, msg);
};

// Disconnects the port from the other end.
PortImpl.prototype.disconnect = function() {
  messagingNatives.CloseChannel(this.portId_, true);
  this.destroy_();
};

PortImpl.prototype.destroy_ = function() {
  if (this.onDestroy_)
    this.onDestroy_();
  privates(this.onDisconnect).impl.destroy_();
  privates(this.onMessage).impl.destroy_();
  // TODO(robwu): Remove port lifetime management because it is completely
  // handled in the browser. The renderer's only roles are
  // 1) rejecting ports so that the browser knows that the renderer is not
  // interested in the port (this is merely an optimization)
  // 2) acknowledging port creations, so that the browser knows that the port
  // was successfully created (from the perspective of the extension), but
  // then closed for some non-fatal reason.
  // 3) notifying the browser of explicit port closure via .disconnect().
  // In other cases (navigations), the browser automatically cleans up the

```



```

// port.
messagingNatives.PortRelease(this.portId_);
delete ports[this.portId_];
};
// Returns true if the specified port id is in this context. This is used by
// the C++ to avoid creating the javascript message for all the contexts that
// don't care about a particular message.
function hasPort(portId) {
    return portId in ports;
};
// Hidden port creation function. We don't want to expose an API that lets
// people add arbitrary port IDs to the port list.
function createPort(portId, opt_name) {
    if (ports[portId])
        throw new Error("Port " + portId + " already exists.");
    var port = new Port(portId, opt_name);
    ports[portId] = port;
    messagingNatives.PortAddRef(portId);
    return port;
};
// Helper function for dispatchOnRequest.
function handleSendRequestError(isSendMessage,
                                responseCallbackPreserved,
                                sourceExtensionId,
                                targetExtensionId,
                                sourceUrl) {
    var errorMsg = [];
    var eventName = isSendMessage ? "runtime.onMessage" : "extension.onRequest";
    if (isSendMessage && !responseCallbackPreserved) {
        $Array.push(errorMsg,
            "The chrome." + eventName + " listener must return true if you " +
            "want to send a response after the listener returns");
    } else {
        $Array.push(errorMsg,
            "Cannot send a response more than once per chrome." + eventName +

```

```

    " listener per document");
}
$Array.push(errorMsg, "(message was sent by extension" + sourceExtensionId);
if (sourceExtensionId != "" && sourceExtensionId != targetExtensionId)
    $Array.push(errorMsg, "for extension " + targetExtensionId);
if (sourceUrl != "")
    $Array.push(errorMsg, "for URL " + sourceUrl);
lastError.set(eventName, errorMsg.join(" ") + ". ", null, chrome);
}
// Helper function for dispatchOnConnect
function dispatchOnRequest(portId, channelName, sender,
    sourceExtensionId, targetExtensionId, sourceUrl,
    isExternal) {
    var isSendMessage = channelName == kMessageChannel;
    var requestEvent = null;
    if (isSendMessage) {
        if (chrome.runtime) {
            requestEvent = isExternal ? chrome.runtime.onMessageExternal
                : chrome.runtime.onMessage;
        }
    } else {
        if (chrome.extension) {
            requestEvent = isExternal ? chrome.extension.onRequestExternal
                : chrome.extension.onRequest;
        }
    }
    if (!requestEvent)
        return false;
    if (!requestEvent.hasListeners())
        return false;
    var port = createPort(portId, channelName);
    function messageListener(request) {
        var responseCallbackPreserved = false;
        var responseCallback = function(response) {
            if (port) {

```

```

port.postMessage(response);
privates(port).impl.destroy_();
port = null;
} else {
  // We nulled out port when sending the response, and now the page
  // is trying to send another response for the same request.
  handleSendRequestError(isSendMessage, responseCallbackPreserved,
    sourceExtensionId, targetExtensionId);
}
};
// In case the extension never invokes the responseCallback, and also
// doesn't keep a reference to it, we need to clean up the port. Do
// so by attaching to the garbage collection of the responseCallback
// using some native hackery.
//
// If the context is destroyed before this has a chance to execute,
// BindToGC knows to release |portId| (important for updating C++ state
// both in this renderer and on the other end). We don't need to clear
// any JavaScript state, as calling destroy_() would usually do - but
// the context has been destroyed, so there isn't any JS state to clear.
messagingNatives.BindToGC(responseCallback, function() {
  if (port) {
    privates(port).impl.destroy_();
    port = null;
  }
}, portId);
var rv = requestEvent.dispatch(request, sender, responseCallback);
if (isSendMessage) {
  responseCallbackPreserved =
    rv && rv.results && $Array.indexOf(rv.results, true) > -1;
  if (!responseCallbackPreserved && port) {
    // If they didn't access the response callback, they're not
    // going to send a response, so clean up the port immediately.
    privates(port).impl.destroy_();
    port = null;

```

```

    }
  }
}
privates(port).impl.onDestroy_ = function() {
  port.onMessage.removeListener(messageListener);
};
port.onMessage.addListener(messageListener);
var eventName = isSendMessage ? "runtime.onMessage" : "extension.onRequest";
if (isExternal)
  eventName += "External";
logActivity.LogEvent(targetExtensionId,
                    eventName,
                    [sourceExtensionId, sourceUrl]);
return true;
}
// Called by native code when a channel has been opened to this context.
function dispatchOnConnect(portId,
                          channelName,
                          sourceTab,
                          sourceFrameId,
                          guestProcessId,
                          guestRenderFrameRoutingId,
                          sourceExtensionId,
                          targetExtensionId,
                          sourceUrl,
                          tlsChannelId) {
  // Only create a new Port if someone is actually listening for a connection.
  // In addition to being an optimization, this also fixes a bug where if 2
  // channels were opened to and from the same process, closing one would
  // close both.
  var extensionId = processNatives.GetExtensionId();

  // messaging_bindings.cc should ensure that this method only gets called for
  // the right extension.
  logging.CHECK(targetExtensionId == extensionId);

```

```

if (ports[getOppositePortId(portId)])
    return false; // this channel was opened by us, so ignore it

// Determine whether this is coming from another extension, so we can use
// the right event.
va isExternal = sourceExtensionId != extensionId;
var sender = {};
if (sourceExtensionId != "")
    sender.id = sourceExtensionId;
if (sourceUrl)
    sender.url = sourceUrl;
if (sourceTab)
    sender.tab = sourceTab;
if (sourceFrameId >= 0)
    sender.frameId = sourceFrameId;
if (typeof guestProcessId !== 'undefined' &&
    typeof guestRenderFrameRoutingId !== 'undefined') {
    // Note that |guestProcessId| and |guestRenderFrameRoutingId| are not
    // standard fields on MessageSender and should not be exposed to drive-by
    // extensions; it is only exposed to component extensions.
    logging.CHECK(processNatives.IsComponentExtension(),
        "GuestProcessId can only be exposed to component extensions.");
    sender.guestProcessId = guestProcessId;
    sender.guestRenderFrameRoutingId = guestRenderFrameRoutingId;
}
if (typeof tlsChannelId !== 'undefined')
    sender.tlsChannelId = tlsChannelId;
// Special case for sendRequest/onRequest and sendMessage/onMessage.
if (channelName == kRequestChannel || channelName == kMessageChannel) {
    return dispatchOnRequest(portId, channelName, sender,
        sourceExtensionId, targetExtensionId, sourceUrl,
        isExternal);
}
var connectEvent = null;

```

```

if (chrome.runtime) {
    connectEvent = isExternal ? chrome.runtime.onConnectExternal
        : chrome.runtime.onConnect;
}
if (!connectEvent)
    return false;
if (!connectEvent.hasListeners())
    return false;
var port = createPort(portId, channelName);
port.sender = sender;
if (processNatives.manifestVersion < 2)
    port.tab = port.sender.tab;
var eventName = (isExternal ?
    "runtime.onConnectExternal" : "runtime.onConnect");
connectEvent.dispatch(port);
logActivity.LogEvent(targetExtensionId,
    eventName,
    [sourceExtensionId]);
return true;
};
// Called by native code when a channel has been closed.
function dispatchOnDisconnect(portId, errorMessage) {
    var port = ports[portId];
    if (port) {
        // Update the renderer's port bookkeeping, without notifying the browser.
        messagingNatives.CloseChannel(portId, false);
        if (errorMessage)
            lastError.set('Port', errorMessage, null, chrome);
        try {
            port.onDisconnect.dispatch(port);
        } finally {
            privates(port).impl.destroy_();
            lastError.clear(chrome);
        }
    }
}

```

```

};
// Called by native code when a message has been sent to the given port.
function dispatchOnMessage(msg, portId) {
  var port = ports[portId];
  if (port) {
    if (msg)
      msg = $JSON.parse(msg);
    port.onMessage.dispatch(msg, port);
  }
};
// Shared implementation used by tabs.sendMessage and runtime.sendMessage.
function sendMessageImpl(port, request, responseCallback) {
  if (port.name !== kNativeMessageChannel)
    port.postMessage(request);
  if (port.name === kMessageChannel && !responseCallback) {
    // TODO(mpcomplete): Do this for the old sendRequest API too, after
    // verifying it doesn't break anything.
    // Go ahead and disconnect immediately if the sender is not expecting
    // a response.
    port.disconnect();
    return;
  }
  function sendResponseAndClearCallback(response) {
    // Save a reference so that we don't re-entrantly call responseCallback.
    var sendResponse = responseCallback;
    responseCallback = null;
    sendResponse(response);
  }
  // Note: make sure to manually remove the onMessage/onDisconnect listeners
  // that we added before destroying the Port, a workaround to a bug in Port
  // where any onMessage/onDisconnect listeners added but not removed will
  // be leaked when the Port is destroyed.
  // http://crbug.com/320723 tracks a sustainable fix.
  function disconnectListener() {
    if (!responseCallback)

```

```

    return;
  if (lastError.hasError(chrome)) {
    sendResponseAndClearCallback();
  } else {
    lastError.set(
      port.name, 'The message port closed before a reponse was received.',
      null, chrome);
    try {
      sendResponseAndClearCallback();
    } finally {
      lastError.clear(chrome);
    }
  }
}

function messageListener(response) {
  try {
    if (responseCallback)
      sendResponseAndClearCallback(response);
  } finally {
    port.disconnect();
  }
}

privates(port).impl.onDestroy_ = function() {
  port.onDisconnect.removeListener(disconnectListener);
  port.onMessage.removeListener(messageListener);
};

port.onDisconnect.addListener(disconnectListener);
port.onMessage.addListener(messageListener);
};

function sendMessageUpdateArguments(functionName, hasOptionsArgument) {
  // skip functionName and hasOptionsArgument
  var args = $Array.slice(arguments, 2);
  var alignedArgs = messagingUtils.alignSendMessageArguments(args,
    hasOptionsArgument);
  if (!alignedArgs)

```



```

    throw new Error('Invalid arguments to ' + functionName + '.');
    return alignedArgs;
  }
var Port = utils.expose('Port', PortImpl, { functions: [
  'disconnect',
  'postMessage'
],
properties: [
  'name',
  'onDisconnect',
  'onMessage'
] });
exports.$set('kRequestChannel', kRequestChannel);
exports.$set('kMessageChannel', kMessageChannel);
exports.$set('kNativeMessageChannel', kNativeMessageChannel);
exports.$set('Port', Port);
exports.$set('createPort', createPort);
exports.$set('sendMessageImpl', sendMessageImpl);
exports.$set('sendMessageUpdateArguments', sendMessageUpdateArguments);
exports.$set('hasPort', hasPort);
exports.$set('dispatchOnConnect', dispatchOnConnect);
exports.$set('dispatchOnDisconnect', dispatchOnDisconnect);
exports.$set('dispatchOnMessage', dispatchOnMessage);
})

```

ДОДАТОК Б

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_ doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Program.rar	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація_.ppt	Презентація кваліфікаційної роботи