Міністерство освіти і науки україни
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРИКЛАДНОЇ ІНФОРМАТИКИ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ __Гамаюн В.П.
(підпис)
(ПІБ)
"_____"_____2021р.

ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ "БАКАЛАВР"

Тема: Веб-сайт дизайнера інтер'єру

Виконавець: _____ Нархов Євгеній Юрійович
(підпис) **(ПІБ)**

Керівник: _____ Сінько Юрій Іванович
(підпис) **(ПІБ)**

Нормоконтролер: _____ Боровик Володимир Миколайович
(підпис) **(ПІБ)**

Київ 2021

ВСТУП

У сучасному світі складно недооцінювати роль застосування електронних ресурсів та мережі інтернету.

У дипломному проекті розглянуті питання струткутури веб-сайту, методів його розробки, проектування, написання та тестування веб-сайту.

Інтернет сьогодні найбільш розвинена у світі інформаційна система, за допомогою якої здійснюється комунікація між мільйонами користувачами. За допомогою мережі Internet забезпечується доступ до більш як п'яти мільйонів інформаційних Web-сайтів. Якщо прийняти до уваги кількісні показники українського сегменту Internet, то вони налічують 200-300 тисяч користувачів, загальна кількість Web-серверів на сьогодні досягла позначки 4,5 тисяч.

З самого початку розвитку Інтернет, а особливо з появою Web-технологій, мережа орієнтована на інформаційне забезпечення своїх користувачів.

Одним з ланцюжків комп’ютеризації світу є веб-ресурси. Вони, в першу чергу, виконують інформативні та комунікативні функції , а також дозволяють запровадити проекти, що розробляються дистанційно і допомагають людям, не виходячи з дому, отримувати відповідні знання.

Веб-сайт це найкращий варіант за допомогою якого можна рекламувати свій товар,або послуги, давати повний список інформації про свою продукції, додавати контактні дані та багато іншого. Сама по собі розробка веб-сайту являє собою створення маркетингового інструменту, призначеного стимулювати попит на певні послуги чи продукцію, або інформаційного ресурсу, спрямованого на те, щоби донести до цільової аудиторії необхідну інформацію, або створення сервісного ресурсу, який би вирішував завдання з надання певних послуг зацікавленим відвідувачам. При цьому, варто пам'ятати, що розробка сайту, який може якісно виконувати всі ці функції, - процес досить складний і потребує високого професійного виконання.

Для для створення повноцінного сайту потрібно розглянути наступні питання:

– Проектування структури веб-сайту - розробити структуру, яка відповідатиме потребам замовника і буде мати високій рівень захищеності і

мати високу доступність.

- Написання програмного коду HTML – релізувати розроблену структуру та створити елементи веб-сайту;
- Розробка стильової частини – розробити звонішню стильову частину сайту, його вигляд, для того щоб він був зручним у використанні приємним на око та привабливим для клієнтів;
- Написання програмного коду CSS – реалізувати розроблену стильову та візуальну частину веб-сайту.

Необхідно розглянути загальну архітектуру веб-сайту, в якій визначені всі необхідні компоненти сайту, розглянути та проаналізувати основні методи створення сайтів та визначити основні вимоги до сайту.

Сьогодні більшість сторінок інтерактивні і надають сучасні інтерактивні послуги, такі як корзини Internet-магазинів, динамічна візуалізація та навіть складні соціальні мережі

Для забезпечення надійного функціонування веб-сайту, необхідно розглянути особливості методів розробки веб-сайтів та обрати інструменти створення, які надають користувачам нові можливості,.

Для взаємодії всіх сегментів сайту тобто front-end та back-end частин сайту необхідно виконати вибір взаємодічих інструментів створення та вивчення структури веб-сайт.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВЕБ-САЙТ ТА МЕТОДИ РОЗРОБКИ

1.1. Поняття про веб-сайт

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

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

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

1.2. Класифікація веб-сайтів

В даний час у всесвітній павутині розміщено декілька мільйонів веб-сайтів і їх число постійно росте. Це особисті сторінки, що містять інформацію про автора, його інтереси. Їх створюють для того, щоб знайти друзів по інтересах, розширити свій кругозір, свій світ.

За видовими ознаками існує наступна класифікація веб-ресурсів: комерційний корпоративний та тематичний.

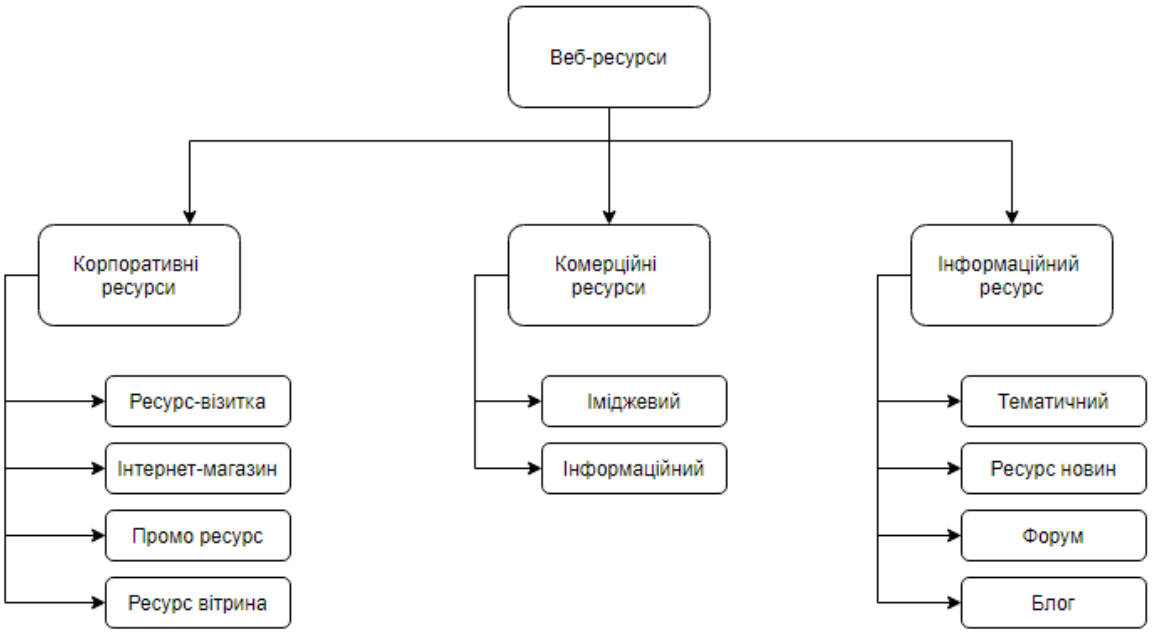

Рис 1.1. Класифікація видів веб-ресурсів

Ресурси комерційного типу є найбільш поширеними серед інших видів ресурсів. Основна мета їх створення – продавати товари користувачам мережі.

Комерційні ресурси поділяють на кілька основних видів :

– ресурс-візитка (створюється для однієї людини або невеликої фірми; метою створення є поширення професійної інформації про компанію або певного фахівця для великої кількості відвідувачів мережі Інтернет, що сприяє залученню нових клієнтів);

– інтернет-магазин (надає послуги в оформленні купівлі та доставки

різноманітних товарів у режимі реального часу (*on-line*); відвідувач може вибрати товар та зробити замовлення, обрати варіант оплати та спосіб доставки замовлення,а також одержати рахунок на оплату);

– промо-ресурси (призначені для активної реклами будь-якого продукту,

товару або послуги; вони зазвичай невеликі за об’ємом текстового наповнення, але повинні мати яскравий, незабутній та стильний дизайн, що поєднує велику кількість графіки і здатний ефектно проінформувати користувача про товар чи послугу);

– ресурс-вітрина (містить максимальну інформацію про власника та надає представницький каталог його товарів або послуг; зазвичай містить прайс-листи, описи товарів або послуг, фотогалереї).

Ресурси корпоративної спрямованості створюються для виконання одного з наступних завдань: підвищення іміджу компанії в очах існуючих і потенційних клієнтів або зміцнення зв'язку між відділеннями та філіями організації. Відповідно до цього корпоративні ресурси бувають :

– іміджеві ресурси;

– інформаційні ресурси.

Інформаційні ресурси призначені для донесення до користувачів найрізноманітнішої інформації. Це досить великі віртуальні масиви, що включають в себе безліч тематичних розділів меншого обсягу, або певну кількість самостійних проектів. Вони можуть бути для відвідувачів значним джерелом інформації, нагадуючи спеціалізований журнал або енциклопедію. Існує кілька різновидів ресурсів інформаційної спрямованості :

– тематичний ресурс;

– ресурс новин;

– форум (слугують для забезпечення спілкування всіх бажаючих у форумі);

– блог.

Залежно від мети створення веб-ресурси поділяються на :

– комерційні (основним призначення є розвиток бізнесу за допомогою засобів інтернет-технологій; вони сприяють отриманню доходів від продажу товарів, пошуку нових клієнтів, розміщенню реклами тощо);

– некомерційні (представництва урядових організацій, освітніх установ, політичних партій; основним завданням є повідомлення про своє існування, надання інформації про діяльність, створення засобів спілкування, надання безкоштовних послуг з певних питань).

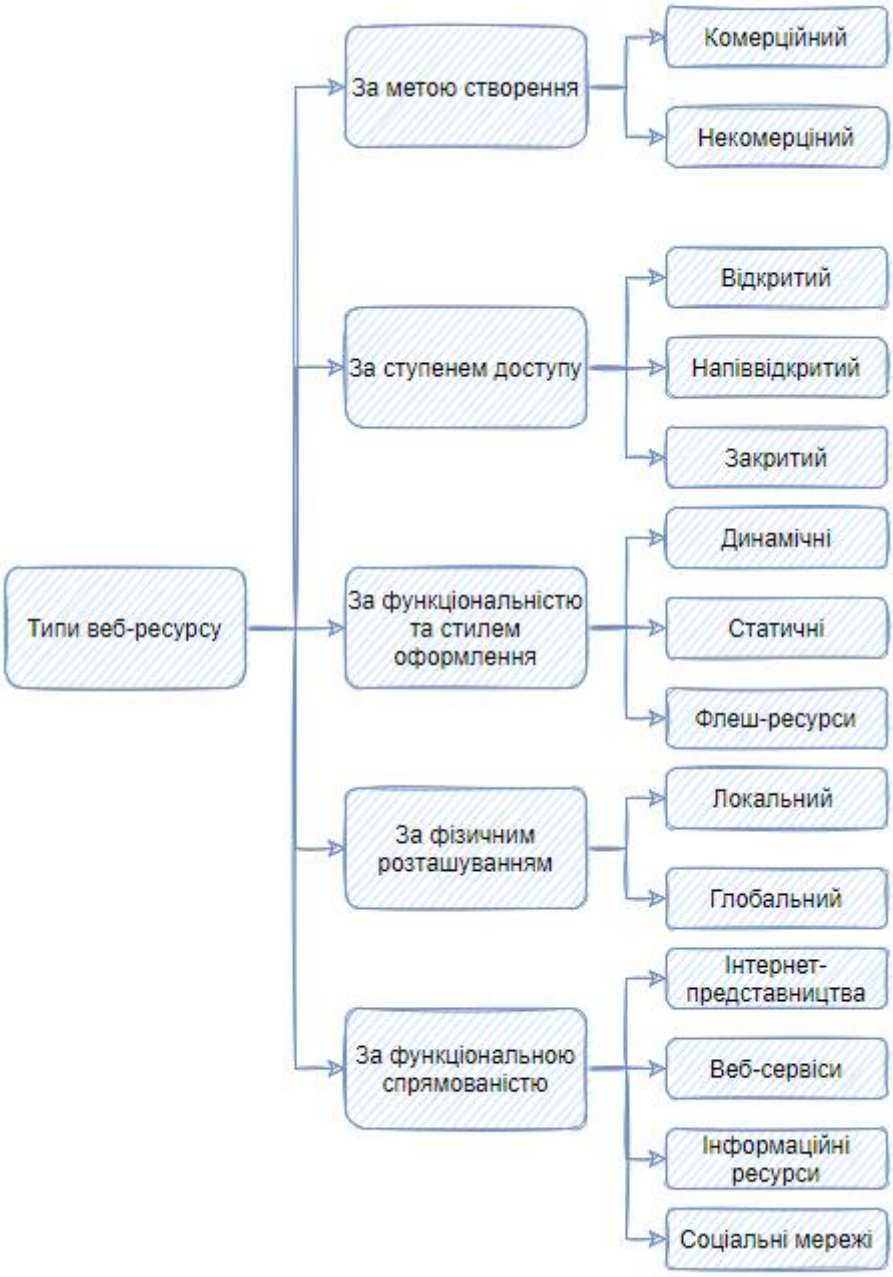

Рис 1.2. Класифікація типів веб-ресурсів

За ступенем доступу веб-ресурси бувають :

– відкриті (доступ має будь-який користувач);

– напіввідкриті (для доступу до ресурсу необхідна реєстрація);

– закриті (право доступу має обмежене коло осіб).

За функціональністю та стилем оформлення веб-ресурси розрізняються :

– динамічні (генерують свій вміст спеціальними програмами, що мають назву скрипти, і формують інформацію в залежності від дій користувача на підставі аналізу його попередніх дій);

– статичні (інформаційне наповнення залишається незмінним для всіх користувачів ресурсом);

– флеш-ресурси (конструюються за допомогою технології використання анімаційних зображень *Adobe Flash*, саме тому вони дуже яскраві та особливо привабливі).

За фізичним розташуванням веб-ресурси поділяються на :

– локальні (доступні тільки для конкретної локальної мережі);

– глобальні (надають доступ для будь-якого користувача мережі).

За функціональною спрямованістю веб-ресурси виокремлюють:

– інтернет-представництва різних організацій і компаній;

– веб-сервіси (слугують для надання тих чи інших послуг та завдань);

– інформаційні ресурси (включають в себе текстові, ілюстративні, мультимедійні матеріали або повідомлення інших будь-яких видів, що дозволяють розкрити тематичний зміст ресурсу або одного з його розділів);

– соціальні мережі (інтерактивний веб-ресурс, призначений для можливості надання засобів зв’язку для спілкування з іншими користувачами).

За функціональними можливостями веб-ресурси поділяються на п’ять основних різновидів: брошурний, електронної комерції, портал, *wiki*-ресурс, соціальні медіа .

1.3. Методи розробки веб-сайту

Розробка веб-сайту – це процес створення вебсайтів або вебдодатків. Основними етапами процесу є вебдизайн, верстка сторінок, програмування для веб на стороні клієнта і сервера, а також конфігурування вебсервера .

На сьогоднішній день існують кілька етапів розробки вебсайту:

- Проєктування сайту або вебдодатку (збір і аналіз вимог розробка Технічного завдання, проєктування Інтерфейс користувача);
- Розробка концепції сайту;
- Створення дизайнконцепції сайту;
- Створення макетів сторінок;
- Створення мультимедіа;
- Верстка сторінок і дизайнів;
- Програмування (розробка функціональних інструментів) або інтеграція в систему управління вмістом (CMS);
- Оптимізація та розміщення матеріалів сайту;
- Тестування та внесення коригувань;
- Відкриття проєкту на хостингу;
- Обслуговування сайту, що працює, або його програмної основи.

У залежності від поточного завдання деякі з етапів можуть бути відсутні або бути тісно пов'язані один з іншим.

1.3.1 Розробка за допомогою *HTML*

При появі стандарту *HTML*, цей метод був найпоширенішим. Основною програмою для розробки був Notepad. Але у цього методу є істотні недоліки. Цей спосіб досить трудомісткий. І до того ж зробити нормальний Web-сайт без CSS, JavaScript та інших мов програмування досить важко. Цей підхід Ви вивчали раніше .

HTML5 - це нова специфікація мови розмітки, що використовується в створенні веб-сторінок. На відміну від попередніх версій це не просто специфікація мови для гіпертекстової розмітки, а набір різнопланових модулів

- від *HTML*-елементів до відео-, аудіо-, векторної графіки SVG, растрової JavaScript-графіки Canvas, локальних баз даних і навіть різних API браузера. Весь цей список модулів дозволяє HTML5 успішно конкурувати з технологіями Flash і Silverlight. Причому успішна конкуренція з Flash можлива ще й тому, що *HTML5* потенційно набагато менше навантажує процесор комп'ютера, ніж Flash, не вимагає установки плагінів і оновлень, а значить, менш вразливий для хакерських атак.
Фактично саме поява в HTML5 нових тегів <video> та <audio> робить його потенційним конкурентом існуючих технологій від Adobe і Microsoft.

HTML 5 вводить кілька нових елементів і атрибутів. Деякі з них технічно є еквівалентами і , але мають своє семантичне значення, наприклад <nav> (навігаційна панель) і <footer>. Ці теги будуть полегшувати роботу пошукачам, а також обробку сайту з Корман персонального комп'ютера або читають програм. Інші елементи надають нову функціональність, такі як <audio> і <video>. Деякі застарілі елементи *HTML 4*, такі як <font> і , були видалені з *HTML 5*.

Обробка помилок
HTML 5-сумісні браузери дуже гнучкі при обробці помилок, на відміну від *XHTML*. *HTML 5* розроблений так, що не підтримують його браузери можуть спокійно ігнорувати елементи *HTML 5*. На відміну від четвертої, п'ята версія чітко прописує правила лексичного розбору, щоб різні браузери відображали один і той же результат в разі некоректного синтаксису.

1.3.2. Розробка за допомогою програмних засобів розробки сайтів

Існує багато готових рішень, для більш швидкої і зручної розробки сайтів. Вони надають можливість генерувати html код, розробляти сайт у візуальному режимі і мають багато інших можливостей.

Виділимо декілька інструментальних систем для розробки *HTML*:

• *програми, що мають у своєму складі візуальні редактори (design-based editor)* – засоби, які автоматично формують необхідний HTML-код, дозволяючи розробляти Web-сторінки в режимі WYSIWYG;

• *програми-редактори (code-based editors)*, які надають редактор і допоміжні засоби для автоматизації написання коду.

Розглянемо найбільш популярні design-based редактори:

• *Adobe DreamWeaver* – один з кращих візуальних редакторів, що генерують HTML код. Він дозволяє працювати в декількох режимах одночасно, з HTML кодом або у візуальному режимі. Але основним недоліком є те, що програма генерує занадто "важкий" код, додаючи багато зайвого. Але, якщо знайомі з HTML, тоді текст HTML можна відредагувати. Ця програмна система випускалася до 2005 року компанією Macromedia, після чого була придбана фірмою Abode.

• *Microsoft FrontPage* – це простий в засвоєнні і зручний Web-редактор для проектування, підготовки і публікації Web-сайтів. Завдяки інтеграції з сімейством продуктів MS Office, звичного інтерфейсу і великої кількості шаблонів програма дозволяє швидко засвоїти роботу навіть початківцям, які знайомі з основами роботи в MS Word. При цьому FrontPage не можна назвати рішенням для «чайників»: програма надає широкі функціональні можливості та різноманітні засоби оптимізації при колективній розробці. Вона дозволяє швидко створювати динамічні комплексні Web-вузли практично будь-якої складності.

Розглянемо популярні code-based редактори:

• *Adobe HomeSites* – це потужний пакет, до складу якого входить багато корисних функцій і підпрограм. Об'ємний дистрибутив редактора включає в себе, крім самого редактора, редактор TopStyle для редагування таблиць CSS, перевірку орфографії та багато іншого.

• *HotDog* – цілком професійний редактор. Вбудована підтримка дуже широкого набору інструментів, що використовуються в Web-дизайні: HTML, CSS, JavaSript, VBScript, ASP, а також DOM – об'єктну модель документа, що використовується при програмуванні на VBScript і JavaScript. При цьому

перевірка синтаксису цих інструментів може налаштовуватися в досить широких межах. Наприклад, HTML можна перевіряти на відповідність версії 3.2, 4, або на “перегляді” тільки в Internet Explorer та інше.

• *AceHTML* – основні функціональні можливості – подібно HomeSite і FirstPage. З цінних якостей AceHTML треба відзначити вмілу роботу з кодуваннями російської мови. Другий плюс – дуже непогана вбудована утиліта для перегляду графічних файлів у комп’ютері. У просторому вікні відображаються ескізи всіх картинок в директорії, а також їх параметри і розмір у пікселях .

1.3.3. Розробка за допомогою інструментальних систем таких як CMS

Для створення динамічного сайту можливі два шляхи. По-перше, це написання власних програм, які відповідають за створення потрібних шаблонів і підтримують необхідні функції. При цьому створена система буде повністю відповідати потребам, проте можливо вимагатиме великих програмістських зусиль і часу.

Другий шлях - це скористатися вже існуючими системами, які і називаються системами управління Web-контентом. Перевагою цього шляху є зменшення витрат часу і сил. До його недоліків можна віднести зниження гнучкості, надання недостатнього або надмірного набору можливостей.

Другий шлях є основним на цей час для створення складних, сучасних сайтів, порталів, Веб-додатків. Це метод з використанням CMS. Вікіпедія дає наступне визначення. CMS це система керування вмістом (контентом) (англ. Content management system, CMS) — комп'ютерна програма чи система, що використовується для забезпечення і організації сумісного процеса створення, редагування та управління текстовими і мультимедійними документами (вмісту чи контента). Звичайно цей вміст розглядається як не структуровані данні предметної задачі в протилежність структурованим даним, що звичайно знаходяться під керуванням СУБД. Звичайно, що встановлення CMS робиться вже на вибраному хостінзі. При цьому як мінімум вимагається FTP доступ та дозвіл роботи MySQL.

Таким чином, відділення дизайну від контенту є головною відміною особливістю динамічних сайтів від статичних. На цій основі можливі подальші удосконалення структури сайту, такі як визначення різних призначених для користувача функцій і автоматизація бізнес-процесів, а саме головне, контроль контенту, що надходить на сайт.

1.3.4. Розробка за допомогою фреймворків.

Фреймворк це програмний продукт, який є основою для створення сайтів, але він не має готових рішень для побудови сайтів, не має рішень для виконання певних функцій. Це більш низький рівень ніж CMS. Розробники на фреймворках створюють і інтерфейсну частину, і базу даних, і алгортми та програмні рішення проблемно орієнтованої частини і скоріше не сайту , а Веб додатку. Створюючи також його адміністративний інтерфейс.

Фреймворк (в інформаційних системах)- це структура програмної системи, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. На відміну від бібліотек, які об'єднують набір підпрограм близької функціональності, фреймворк містить в собі велику кількість різних за призначенням бібліотек. Вживається також слово каркас, а деякі автори використовують його в якості основного, в тому числі не базуючись взагалі на англомовному аналогу. Можна також говорити про каркасний підход як про підхід до побудови програм, де будь-яка конфігурація програми будується з двох частин: перша, постійна частина - каркас, незалежний від конфігурації до конфігурації і несе в собі гнізда, в яких розміщується друга, змінна частина - змінні модулі (або точки розширення).

Для того щоб відповісти на питання що повинен мати фреймворк , щоб розглядатися як WEB-технологія розглянемо, які види фреймворків є:- Фреймворки програмної системи; Фреймворки додатків;
Фреймворки концептуальної моделі;

Фреймворк програмної системи - це каркас системи або підсистеми. Він може включати допоміжні програми, мови сценаріїв, все, що полегшує розробку і об'єднання різних компонентів. Від бібліотеки він відрізняється

виконанням коду, який написаний для нього, але не виконується сам. До цього виду фреймворків відносяться і фреймворки для WEB.

Фреймворк додатку має стандартну структуру. З ростом необхідності в графічних інтерфейсах користувача з'явилася і необхідність у фреймворках додатків. З їх допомогою простіше створювати засоби для створення графічних інтерфейсів автоматично. Для створення фреймворку додатків використовують об'єктно-орієнтоване програмування. Перший такий Фреймворк написала компанія Apple для Macintosh. Спочатку він був створений за допомогою Паскаль, потім же перероблений в C ++.

Фреймворк концептуальної моделі - це абстрактне поняття даної структури для визначення способів вирішення конкретної проблеми.

WEB фреймворки - це каркас, призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, розробку інтерфейсу, а також зменшують дублювання коду .

Є п'ять типів веб-фреймворків: Request-based, Component-based, Hybrid, Meta and RIA-based.

Request-based: фреймворки, які безпосередньо обробляють вхідні запити. Збереження стану відбувається за рахунок серверних сесій. Приклади: Django, Ruby на Rails, Struts, Grails.

Component-based: фреймворки, які абстрагують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Своєю поведінкою дані каркаси нагадують стандартні програмні графічні інтерфейси.. Приклади: JSF, Tapestry, Wicket.

Hybrid-based: фреймворки, які комбінують Request-based та Component-based фреймворки, беручи під свій контроль всі дані і логічний потік в заснованої на запиті моделі. Розробники мають повний контроль над URL, формами, параметрами, cookies і pathinfos. Однак замість того, щоб

відобразити дії і контролери безпосередньо до запиту, гібридні фреймворки забезпечують об'єктну модель компонентів, яка поводиться тотожно в багатьох різних ситуаціях, таких як окремі сторінки, перервані запити, подібні порталу фрагменти сторінок і інтегровані віджети. Компоненти можуть розподілятися окремо і ефективно інтегруватися в інші проекти. Приклади: RIFE.

Meta -based: у фреймворків є ряд базових інтерфейсів для загального обслуговування і основу яка легко розширюється, для інтегрування компонентів і служб. Приклад: Keel.

RIA-based: фреймворки для розробки Rich Internet Applications (RIA). Служать для розробки повноцінних додатків, що запускаються всередині браузера. Приклад: Flex.

Найбільш поширеними є Request-based і Component-based веб-фреймворки. Більшість WEB-фреймворків побудовані на архітектірі MVC.Model View Controller (MVC, «модель- представлення-контролер», «модель-вид- контролер») – схема використання декількох шаблонів проектування, за допомогою яких модель додатки, інтерфейс і взаємодія з користувачем розділені на три окремі компоненти таким чином, щоб модифікація одного з компонентів чинила мінімальний вплив на інші . Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметної області.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

Висновки до розділу

На основі аналізу інформації данної у трьох підрозділахможна дійти висновку про відсутність єдиного підходу у створенні веб сайтуЗа видовими ознаками веб-ресурси розподіляються на: комерційні, корпоративні та інформаційні ресурси. За типологічними ознаками веб- ресурси розподіляються: залежно від мети створення, за ступенем доступу, за функціональністю та стилем оформлення, за фізичним розташуванням та за функціональною спрямованістю.

Принципи проектування веб-ресурсів базуються на трьох взаємопов’язаних складових: дизайні, програмуванні та *SEO*-оптимізації. Дизайн ґрунтується на візуальному зверненні, що забезпечується дотриманням принципів: прозорості, ефективності, орієнтованості на користувача, естетичності, акцентування, рівноваги, гармонії, лаконічності та передбачуваності. Програмування забезпечує функціональні можливості за

допомогою принципів: інформаційної підтримки, доступності, реагування, багатомовності, сумісності, збереження, персоналізації, контрольованості, корекційності та адаптивності. *SEO*-оптимізація допомагає підвищити рейтинг веб-ресурсу серед пошукових систем за допомогою принципів: побудови посилань, прив’язки тексту, щільності ключових слів, альтернативного тексту, оптимальної *URL*-адреси, тегів заголовків та мета-описів

РОЗДІЛ 2
ПРОЕКТУВАННЯ ВЕБ-САЙТУ ТА РОЗРОБКА ІНТЕРФЕЙСУ

2.1. Принципи проектування веб-сайтів

Існує перелік принципів для проектування веб-ресурсів: прозорість, ефективність, доступність, орієнтованість на користувачів, реагування, багатомовність, сумісність, захист авторського права, збереженість.

Під принципом прозорості розуміється чітке визначення користувачем назви, характеру та призначення веб-ресурсу, його вміст та ідентифікація керуючої ланки, що відповідальна за його управління .

Основою принципу ефективності є вміст. Веб-ресурс високої якості повинен мати вміст, який належним чином підібраний і відповідно до тематики супроводжується коментарями та підтримуючою інформацією. Також важливим елементом ефективного веб-ресурсу є легкість, з якою користувачі можуть орієнтуватися серед представлених на ньому матеріалах .

Принцип доступності фокусується на необхідності обслуговувати всіх членів спільноти користувачів. До них відносяться сліпі та слабозорі користувачі, глухі та слабочуючі, особи з порушеннями рухових нервів, індивідуми з порушеннями читання або з труднощами в навчанні. Існує певний клас технологій, які виступають в якості інтерфейсу між Інтернетом та тими, хто має труднощі при використанні стандартної парадигми доступу клавіатури та миші. Вони включають пристрої користувальницького інтерфейсу для заміни тематичної та

клавіатурної форми, екрани та клавіатури Брайля, а також програми програмного забезпечення *Wellas*, які „читають” веб-сторінки .

Принцип орієнтованості на користувачів зосереджується на необхідності задоволення перш за все вимог кінцевого споживача. До основних критеріїв цього принципу відносяться: відповідність вмісту ресурсу, зручний користувальницький інтерфейс, комфортна навігація, наявність інтерактивних елементів

Реагування пов’язане з можливістю ресурсу та його власників відповідати на запитання та пропозиції користувачів. Реалізація даного принципу відбувається за допомогою відкритого форуму, на якому будь-які користувачі мають можливість залучатися до обговорення та покращення якості ресурсу. Ресурс, що оперативно реагує на дії відвідувачів, додає цінності та інтересу для кінцевих користувачів через його інтерактивну природу.

Головною метою використання принципу багатомовності є надання інформації максимальній кількості користувачів, адже, чим більша аудиторія, яку охоплює та обслуговує веб-ресурс, тим більш ефективно він виконує свої завдання.

Принцип сумісності розглядає, як веб-ресурс може взаємодіяти з іншими ресурсами. Основна увага приділяється стандартам, які включають в себе методики, технології, моделі даних та інтерфейси, що полегшують взаємодію між веб- ресурсами. Цей принцип передбачає також наявність чітких, автоматизованих пошукових систем, що реалізують віддалений пошук .

Основна проблема принципу захисту авторського права полягає у забезпеченні управління правами інтелектуальної власності та конфіденційності.

Застосовується в сферах: захисту прав власника ресурса від неадекватного поводження з кінцевим користувачем, захист власника ресурса від судових

переслідувань з боку контентного власника, захист конфіденційності кінцевого користувача .

Принцип збереження забезпечує довготривале зберігання інформації, оскільки неможливо з надійністю передбачити технології та підходи, які будуть використовуватися для доступу до інформації в майбутньому .

2.2. Постановка завдання

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

· гнучкістю, зручною для адміністраторів системою управління структурою;

· веб-сайт повинен підтримувати використання графічних вставок, анімації, які повинні підсилювати емоційно-ціннісний компонент змісту, формувати мотивацію;

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

2.3. Вибір програмних засобів для створення веб-сайту

Враховуючи поставлені вище завдання, автор вирішив, що найбільш оптимальним засобом для розробки такого роду електронної комерції – HTML+CSS+SCSS.

HTML Обмін інформацією в Інтернет здійснюється за допомогою протоколів прикладного рівня, що реалізують той або інший прикладний сервіс (пересилку файлів, гіпертекстової інформації, пошта і так далі). Одним з найбільш молодих і популярних сервісів Інтернет, розвиток якого і привело до сплеску популярності самої Інтернет, стала World Wide Web (WWW), заснована на протоколі HTTP (Hyper Text Transfer Protocol - протокол передачі гіпертекстовій інформації). Гіпертекстові документи, представлені в WWW, мають одну принципову відмінність від традиційних гіпертекстових документів - зв'язки, в них що використовуються, не обмежені одним

документом, і більш того, не обмежені одним комп'ютером. Для підготовки гіпертекстових документів використовується мова HTML (Hyper Text Markup Language – мова розмітки гіпертекстових документів), що надає широкі можливості по форматуванню і структурній розмітці документів, організації зв'язків між різними документами, засоби включення графічної і мультимедійної інформації. HTML-документи є видимими за допомогою спеціальної програми - браузера. Найбільшого поширення в даний час набули браузери MozillaFirefox і Internet Explorer компанії Microsoft (MSIE). Реалізації MozillaFirefoxдоступні практично для всіх сучасних програмних і апаратних платформ, реалізації MSIE доступні для всіх Windows платформ, Macintosh і деяких комерційних Unix-систем. HTML-документ складається з тексту, що є змістом документа, і *тег і в* , що визначають його структуру і зовнішній вигляд при відображенні браузером. Простий html-документ виглядає таким чином:

<html >

<head >

<title>Назва</title>

</head>

<body >

<p>Тіло документа

</body>

</html>

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

Текст всього документа полягає в теги *<html >* , сам документ розбивається на дві частини - заголовок і тіло. Заголовок описується тегами

<head>, в яких можуть бути включені назва документа (за допомогою тегов <title>) і інші параметри, що використовуються браузером при відображенні документа. Тіло документа поміщене в теги <body> і містить власне інформацію, яку бачить користувач. За відсутності тегів форматування весь текст виводиться у вікно браузера суцільним потоком, переклади рядків, пропуски і табуляції розглядаються як пробільні символи, декілька пробільних символів, що йдуть підряд, замінюються на один. Для форматування використовуються наступні основні теги:

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

<p align=right>;

 - переклад рядка в межах поточного абзацу;

<u></u> - виділення тексту підкресленням

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

CSS — це спеціальна мова стилю сторінок, що використовується для опису їхнього 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 має різні *рівні* та *профілі*. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

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

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

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерах та ін.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного **CSS**. Стилі для відображення сторінки можуть бути:

- Стилі автора (інформація надана автором сторінки):
    - зовнішні таблиці стилів (англ. *stylesheet*), найчастіше окремий файл або файли **.css**
    - внутрішні таблиці стилів, включені як частина документу або блоку
    - стилі для окремого елементу
- Стилі користувача

- локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Opera)

- Стилі переглядача (браузера)
  - стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип *каскадності*, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

SCSS- SCSS - "діалект" мови SASS. А що таке SASS? SASS це мова схожий на HAML (вельми лаконічний шаблонизатор), але призначений для спрощення створення CSS-коду. Простіше кажучи, SASS це така мова, код якого спеціальної ruby-програмою транслюється в звичайний CSS код.

Синтаксис цієї мови дуже гнучкий, він враховує безліч дрібниць, які так бажані в CSS. Більш того, в ньому є навіть логіка (@if, each), математика (можна складати як числа, рядки, так і кольори). Можливо, деякі можливості SCSS здадуться вам надмірними, але, на мій погляд, зайвими вони не будуть, залишаться "про запас".

Відмінність SCSS від SASS полягає в тому, що SCSS більше схожий на звичайний CSS код.

В якості середовища розробки буде розглянута программа Sublime Text 3.

Sublime Text 3 - це гарний крос-платформний редактор. Він швидкий і багатий функціоналом, для практично кожної мови програмування. Підтримує декілька виділень, згортання коду, макроси, проекти та інше. Також можливо повноекранне редагування, яке виглядає чудово на великих моніторах. Запускається на Linux, Windows і OSX. Цей редактор надається з необмеженим тестовим періодом .

Концептуальні переваги редактора:

- Швидка навігація (Goto Anything)
- Множинні виділення (Multiple Selections)
- Розподілене редагування (Split Editing)
- Перемикання між проектами на льоту (Instant Project Switch).
- Збірка програм (Build System)
- Можливість розширення функціональності через Plugin API
- Крос-платформенность.

Особлива цінність програми - в можливості підключення плагінів і пакетів. Через модуль управління пакетами і плагінами Package Control можна знайти і додати (а також без проблем видалити, оновити, активувати або деактивувати) безліч необхідного в роботі - і перевірку орфографії, і розширення спектра даних, що відображаються, і елементи, що дозволяють більше "заточити" програму під розробку, і підгонку середовища під звичні стилі роботи (наприклад, робота з Emacs) і під звичні користувачу набори гарячих клавіш, додавання фрагментів для роботи з HTML5 або грамотної роботи з HTML-тегами.

2.4. Принципи розробки інтерфейсу

Хоча привернення уваги є важливою складовою будь-якого веб-ресурсу, але важливо намагатися подати інформацію максимально доступно та зрозуміло. Цьому сприяє використання принципу лаконічності, який полягає в використанні лише необхідної кількості текстового та ілюстративного матеріалу. Зайва графіка може заважати сприйняттю, а надмірна кількість шрифтів або кольорів може відволікати.

Для того, щоб зображення та текст було легко ідентифікувати користувачу застосовується принцип зрозумілості, який полягає в

опрацюванні інформації та зведення її до тривіального викладу, уникаючи наукової або сленгової мови.

В теперішній час все більше користувачів використовують для доступу до веб- ресурсів різноманітні мобільні пристрої, такі як телефони та планшети, тому при розробці ресурсів важливо враховувати даний факт.

У зв'язку з цим Д. Гоув (*Jenny Gove*) сформулювала основні принципи створення веб-ресурсів для мобільних пристроїв, що були розподілені на три категорії: головна сторінка та засоби навігації веб-ресурсом, пошук по веб-ресурсу, заповнення форм.

Як правило, на головній сторінці звичайної версії ресурсу розміщується інформація про власника, область обміну повідомленнями та рекламні матеріали.

Оскільки головна сторінка на мобільному девайсі займає менше простору, то є потреба у відповідних місцях розташувати блоки для того, щоб користувач мав змогу комфортно взаємодіяти з веб-ресурсом (рис. 1.3).

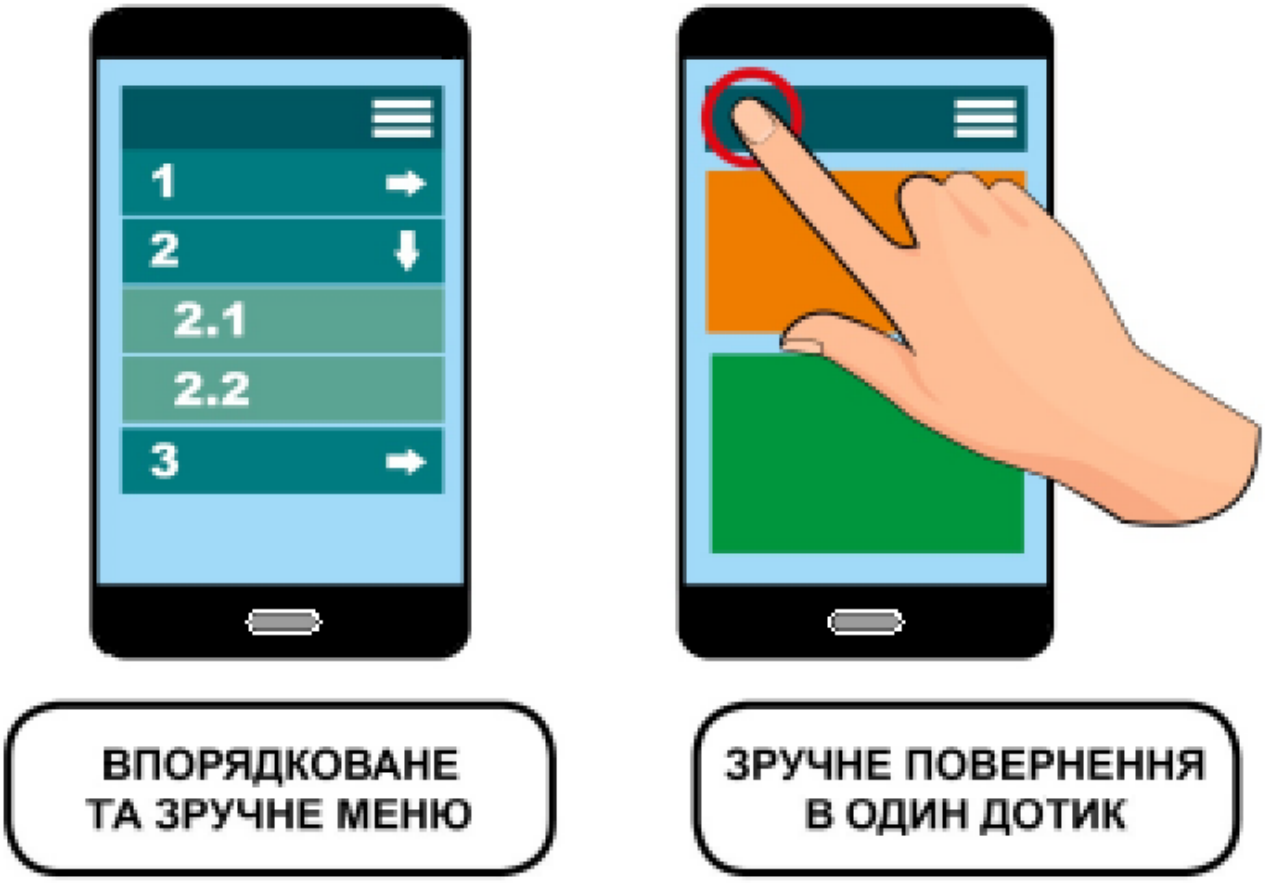

Рис 2.1 Принципи навігації для основної сторінки

Для реалізації даної категорії застосовуються наступні принципи:

– впорядковане та зручне меню (оптимальним варіантом є меню з мінімальною кількістю пунктів, які включають підпункти, що можна відкрити за необхідністю);

– адаптивність сторінок - користувачу на екрані будь-якого розміру не потрібно змінювати масштаб для перегляду інформації (рис. 1.4);

– зручне повернення на головну сторінку в один дотик.

Рис 2.2 Адаптивність для веб-сторінки

Ефективний і функціональний пошук має вирішальне значення для допомоги користувачам веб-ресурсів знайти необхідну інформацію.

Для цього використовують наступні принципи:

– інструмент пошуку необхідно розміщувати на помітному місці;

– наявність функціональних фільтрів для оптимізації пошукових запитів.

Оптимізація пошукових систем (*SEO*) стосується методів, які допомагають веб-ресурсу стати більш помітним для користувачів, які шукають необхідну інформацію за допомогою пошукових систем.

Принцип побудови посилань завжди є найскладнішою частиною *SEO*, для цього використовуються два види посилань: зворотні та природні .

Зворотні посилання – це посилання з інших веб-ресурс на створений веб-

ресурс. Кількість зворотних посилань, релевантність сторінок, що посилаються, та ключові слова, які використовуються у зворотних посиланнях (текст прив'язки),впливають на позицію ресурсу в результатах пошуку. Посилання з одного домену враховуються лише один раз.

Природні посилання – це посилання, які надаються веб-ресурсами, що вирішили посилатися на створений ресурс. Розміщуються користувачами або власниками ресурсів, коли вони вважають ресурс або сторінку цікавими і корисними. Використання прив'язки тексту ставить на меті покращити *SEO*-показники. Текстом прив'язки вважають текст до якого застосовано гіперпосилання, що надає можливість переходити в інші місця веб-ресурсу. Текст прив'язки повинен відповідати сторінці, на яку автор посилається, оскільки слова, що наявні в тексті прив'язки, допомагають підняти рейтинг, який отримає сторінка.

Щільність ключових слів – це відсоток випадків, коли ключове слово чи фраза з'являється на веб-сторінці порівняно із загальною кількістю слів на сторінці. У контексті оптимізації пошукової системи щільність ключових слів може бути використана для визначення того, чи відповідає веб-сторінка певному ключовому слову чи фразі ключових слів.

Застосування альтернативного тексту за допомогою тегу *alt* є найважливішим фактором оптимізації зображення. Текст у тегу *alt* відображається замість зображення, якщо зображення не вдається відобразити або завантажується занадто довго.

Оптимально складена *URL*-адреса дає пошуковим системам просте розуміння з вказівками на те, що буде вміщувати веб-сторінка. Хоча використання *URL*-адреси, що використовує ключові слова, може покращити ефективність пошуку на веб-ресурсі, самі *URL*-адреси, як правило, істотно не впливають на здатність сторінки підвищувати рейтинг.

Існують наступні основні правила оптимізації *URL*-адреси :

– мають бути максимально простими, релевантними та влучними;

– повинні бути стислими і не перевищувати 2048 символів;

– для розділення слів слід користуватися тільки дефісами;

– для написання адреси слід використовувати тільки прописні літери, щоб уникнути проблеми з повторюваними сторінками.

Теги заголовків мають важливе значення, оскільки титульний елемент сторінки призначений для точного та стислого опису вмісту сторінки. Це важливо як для роботи користувачів, так і для оптимізації пошукових систем. Завжди слід важливі ключові слова робити максимально помітними та стежити за тим, щоб теги заголовків були описовими та читабельними.

Тег мета-опису існує для стислого опису вмісту сторінки. Пошукові системи не задіюють ключові слова або фрази в цьому тезі для ранжування, але мета-описи є основним джерелом для фрагменту тексту, що відображається під списком в результатах. Мета-описи можуть вміщувати в себе будь-яку кількість знаків, проте пошукові системи, зазвичай, вирізають фрагменти довжиною не більше 160 символів, тому бажано залишатися в цьому діапазоні.

2.5 Проектування сайту та інтерфейсу

Сайт буде містити в собі 6 веб-сторінок: Головна, Проекти, Послуги, Про мене, Новини та Контакти.

Головна сторінка - повинна максимально інформативно і в стислому об'ємі відображати необхідну користувачеві інформацію про сайт. На головній сторінці необхідно помістити основну інформацію сайту, основне меню сайту та інформативні дані.

Проекти – сторінка містить фотографії проектів з описами для ознайомлення та огляду.

Послуги – сторінка представляю доступні для клієнтів можливості, які вони можуть замовити у дизайнера.

Про мене – сторінка допомагає клієнтам прочитати про дизайнера та дізнатись хто він такий.

Новини – сторінка інформує клієнтів про новини в світі дизайну та новини сайту.

Контакти – сторінка існує для зв’язку клієнта з дизайнером, вона містить пошту, номер телефона та посилання на соціальні мережі.

Інтерфейс буде адаптивним, тобто буде адаптуватися до будь-яких пристроїв буде складатися з активних кнопок та посилань, меню, інтерактивних презентацій, графічних зображень, інформаційно-текстове наповнення та активних форм для заповнення.

Висновки до розділу

В даному розділі було розглянуто основні принципи проектування веб-ресурсів та розробки інтерфейсу. Проведений варіантний аналіз мов програмування та середовищ розробки а також програмне забезпечення, що використовуються при розробці веб-ресурсів.

Також було визначено з завданням завдання, яке перед нами поставлено.

Після проведення аналізу було вирішено розробити веб-ресурс за допомогою веб-технологій HTML – в якості базової розмітки, CSS та SCSS– для оформлення веб-ресурсу і його адаптивності на різних пристроях.

В якості середовища розробки було обрано Sublime Text 3, оскільки він швидкий і простий у використанні а також підтримує велику кількість плагінів , які значно спрощують та поліпшують процес розробки.

РОЗДІЛ 3
СТВОРЕННЯ САЙТУ ТА ЙОГО ТЕСТУВАННЯ

3.1 Розробка структури веб-ресурсу

Створення сайту починається з розробки його логічної структури. Структура сайту – це організація даних, ієрархія матеріалів, необхідна для представлення їх інтернет-аудиторії. Процес створення структури сайту можна розділити на два етапи:

1) структуризація інформації;

2) візуальне представлення структури.

Структуризація інформації заключається в необхідності класифікації і групування розрізнених матеріалів, об’єднанні їх в категорії або рубрики, присвоюванні їм зрозумілих для користувачів назв. При цьому слід враховувати логіку більшості користувачів.

Візуальне представлення структури сайту дозволяє розмістити елементи структури відносно один одного таким чином, щоб користувач з першого погляду міг зорієнтуватись на сайті і зрозуміти, де знайти потрібну йому інформацію. При цьому слід враховувати правило Міллера, згідно якого кількість категорій сайту повинно дорівнювати числу 7+2. Це засновано на властивості короткочасної пам’яті людини: одночасно людина може тримати в голові 5-9 слів (назви категорій сайту.

Після структуризації даних про предметну галузь при великій кількості інформації як текстової, так і графічної, її слід розбити на окремі сторінки. В залежності від способу зв’язування сторінок наш сайт буде мати Гратчасту структур.

Гратчаста структкура (рис.3.1) - заснована на побудові системи навігації сайту, коли присутній зв’язок між вертикальними і горизонтальними елементами сайту (сторінками), тобто є можливість швидкого переходу від однієї до іншої сторінки без відвідування проміжних сторінок

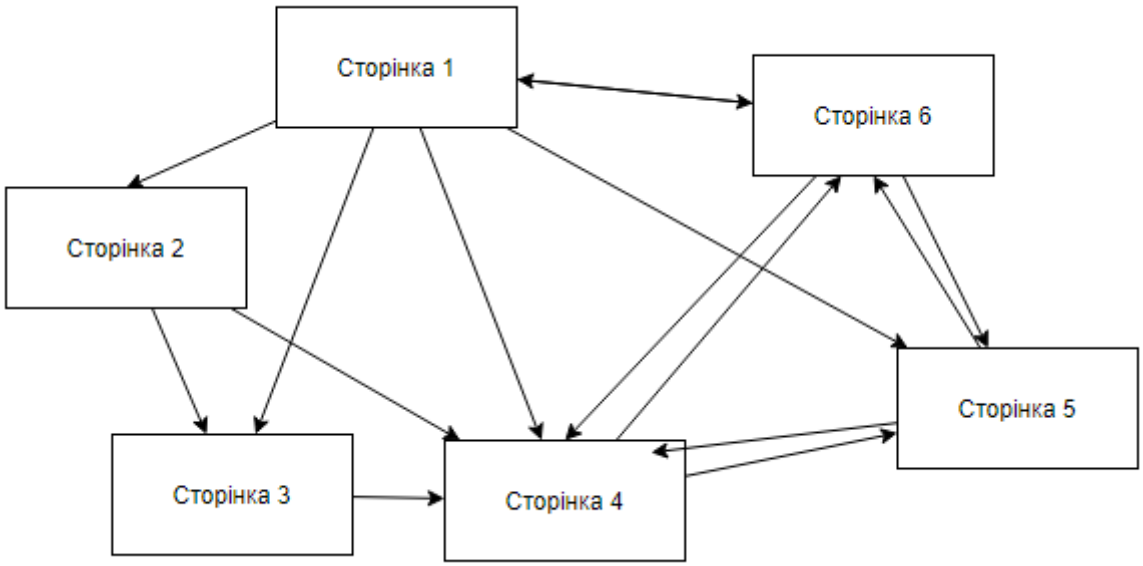

Рис 3.1. Гратчаста структура

Оскільки користувачами веб-ресурсу є люди будь-якого віку та статусу структура має бути простою і універсальною, тому найкращим варіантом буде ґратчаста структура.
Після розробки внутрішньої структури сайту, необхідно визначити місце розташування основних структурних елементів:
1. В верхнійчастині сторінки (шапка, хедер, header) містяться: меню навігації (по сайту), інтро панель.

2. В середній (основній) частині сторінки: привітання, основна інформація, зображення.

3. В нижній частині сторінки (футер, footer): копірайти, адреси, телефони, соціальні мережі і банери.

Враховуючи потреби користувачів було розроблено наступну зовнішню структуру веб-ресурсу (рис 3.4)

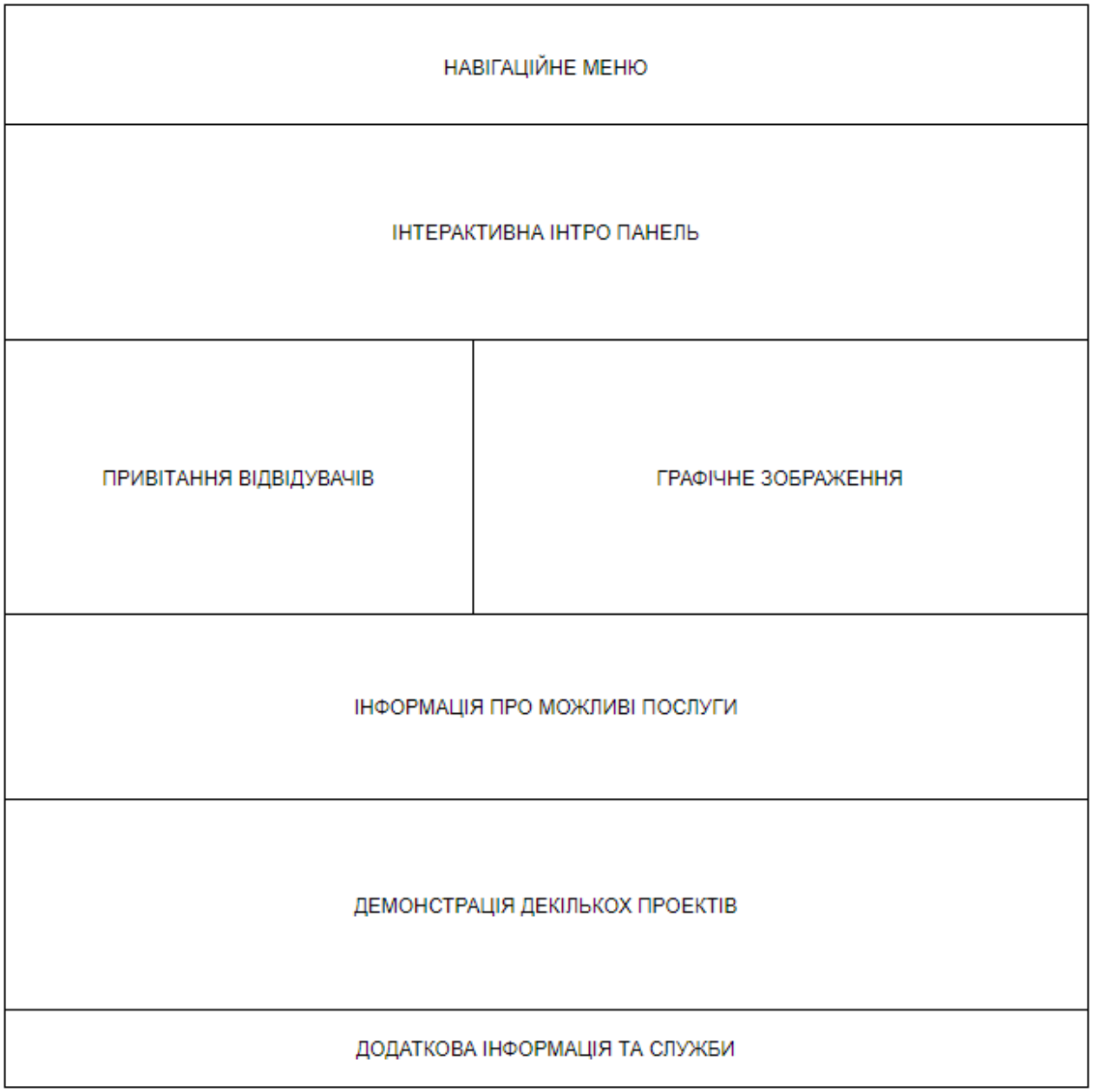

Рис 3.2. Зовнішня структура головної сторінки веб-ресурсу

3.2 Розробка інформаційного наповнення

Контент (content, вміст) — це інформаційне наповнення сайту, та інформація, яку розробник складає самостійно або копіює з дотриманням відповідних законностей.

Контент містить:

- Тексти.
- Графіку: картинки, фото, креслення, елементи інтерфейсу.
- Мультимедіа: аудіо та відео файли.
- Файли у форматі: Adobe, Excel, Word, Power Point, exe, rar.

Весь контент охороняється законом про авторське право, оскільки він є продуктом інтелектуальної праці і має своїх авторів і власників. Окрім якості контенту одним з важливих критеріїв є його доступність. Особливу важливість для користувача має актуальність контенту, його значущість на даний час і достовірність наданих даних, а також відповідність контенту до поставлених цілей.

Унікальний контент (ексклюзивний контент) - це інформація, яка не має аналогів на ресурсах схожої тематики або розміщена на веб-сайті з дозволу правовласника, така, що є результатом інтелектуальної праці та охороняється законом про авторське право. Найчастіше цей термін застосовують до текстового наповнення сайтів (текстовий контент).

3.3 Вибір кольорової гамми

Перше враження про сайт у користувача формується протягом перших 9 секунд відвідування. Необхідно, щоб колірна схема відповідала контенту сайту та його цільовій аудиторії. Колірна схема повинна підсилювати виразність сайту. Колір може здійснювати психо-емоційний вплив на людину. При виборі кольорів необхідно, щоб вони гармонійно поєднувались між собою, а також із контентом сайту. Якщо обрати дуже насичені яскраві кольори, то вони, по-перше, будуть відволікати користувача від контенту, по-друге будуть створювати психологічний тиск на користувача, він швидко втомиться і покине сайт, не знайшовши необхідну інформацію.

Зазвичай кольорова гамма не вигадується самим розробником веб-сайту, йде спільна робота разом із замовником та узгодження кольорів у якому замовник хоче бачити свій сайт. На сайті буде використано: білий, сірий та темно-зелений кольори.

3.4 Реалізація спроектованого інтерфейсу

Завдяки гарної кількості знань і галузі створення веб-сайтів ми створюємо файли с розширеннями: «*.scss» та «*.css» прописуємо в них всі змінні, класи та дані які будем використовувати при написанні сайту. Потім створюємо для кожної сторінки окремі HTML файли, підключаємо створені раніше файли до кожного з них та прописуємо кожну сторінку окремо

Таким чином вносячи зміни до коду одного файлу ми можемо змінювати вигляд елементів кожної сторінки окремо

При цьому завантаження стартової сторінки одразу змушує браузер завантажити і список стилів, що пришвидшує завантаження усіх інших сторінок. Вигляд розробленого інтерфейсу можна побачити на рисунку 3.3

Код розміщується у додатку А

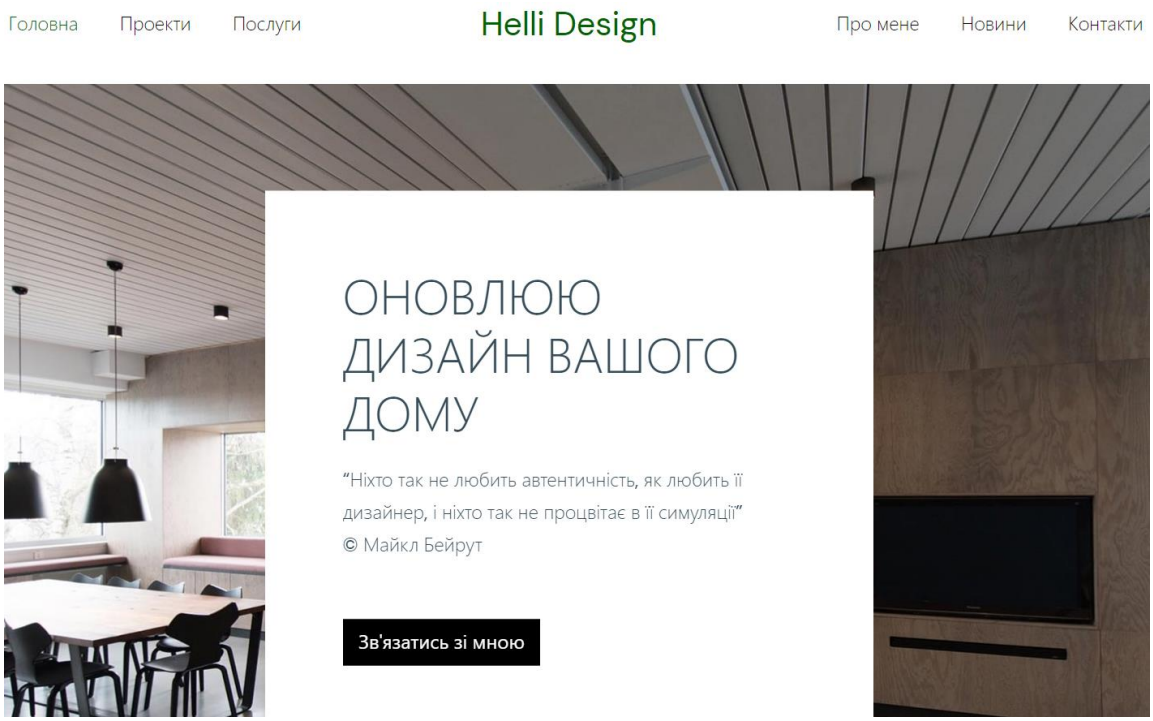

Рис 3.3. Початок головної сторінки

3.5 Тестування готового сайту

Тестування сайту проводиться з метою контролю адаптивності та якості відображення сайту, як в різних браузерах так і на різних пристроях, щоб сайт був пристосований для всіх типів користувачів і не орієнтувався на якусь конкретну аудиторію

Відображення у бразуері Google Chrome:

Рис 3.4. Вітання на головній сторінці

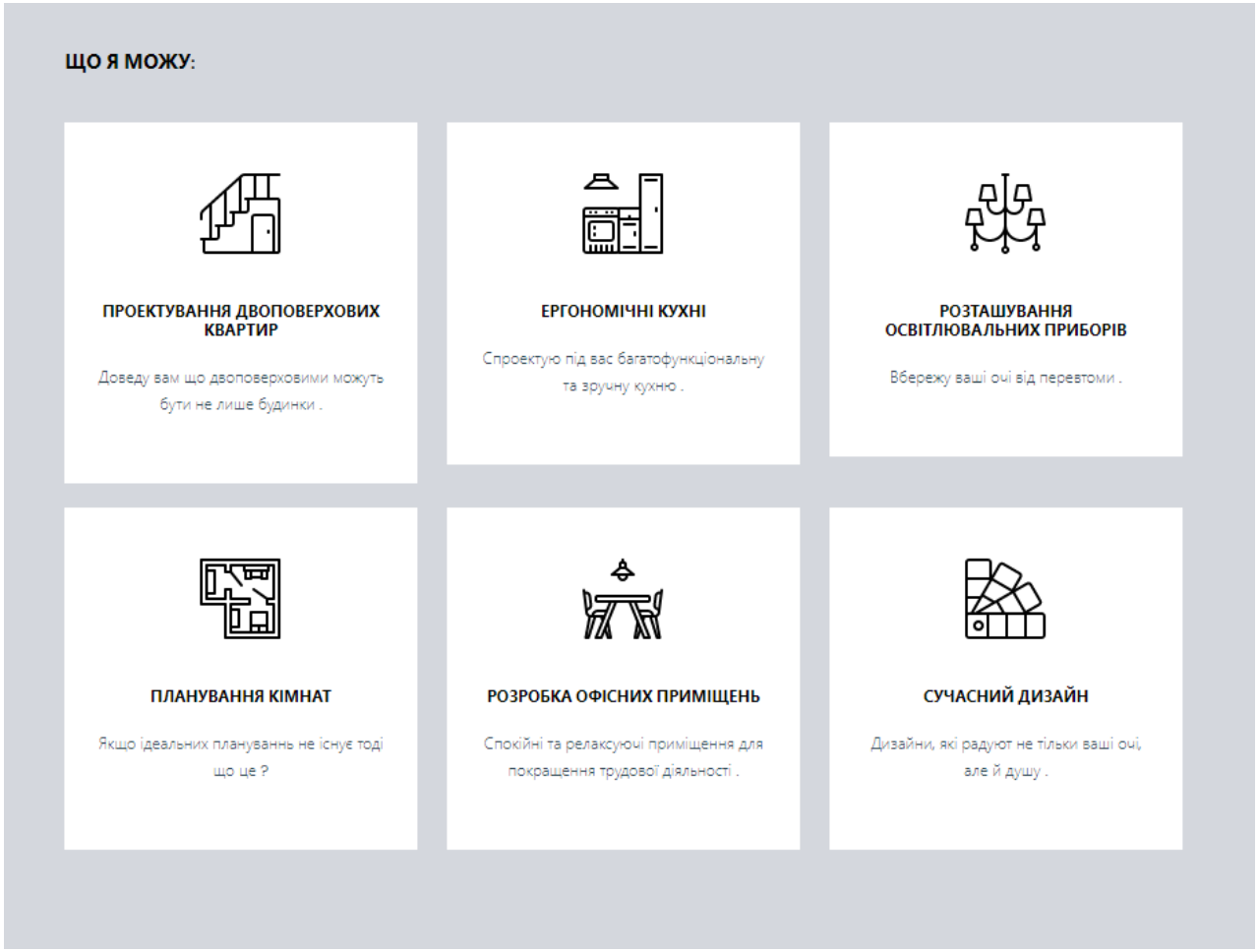

Рис 3.5. Перелік можливостей дизайнера

Рис 3.6. Відображення сторінки проектів

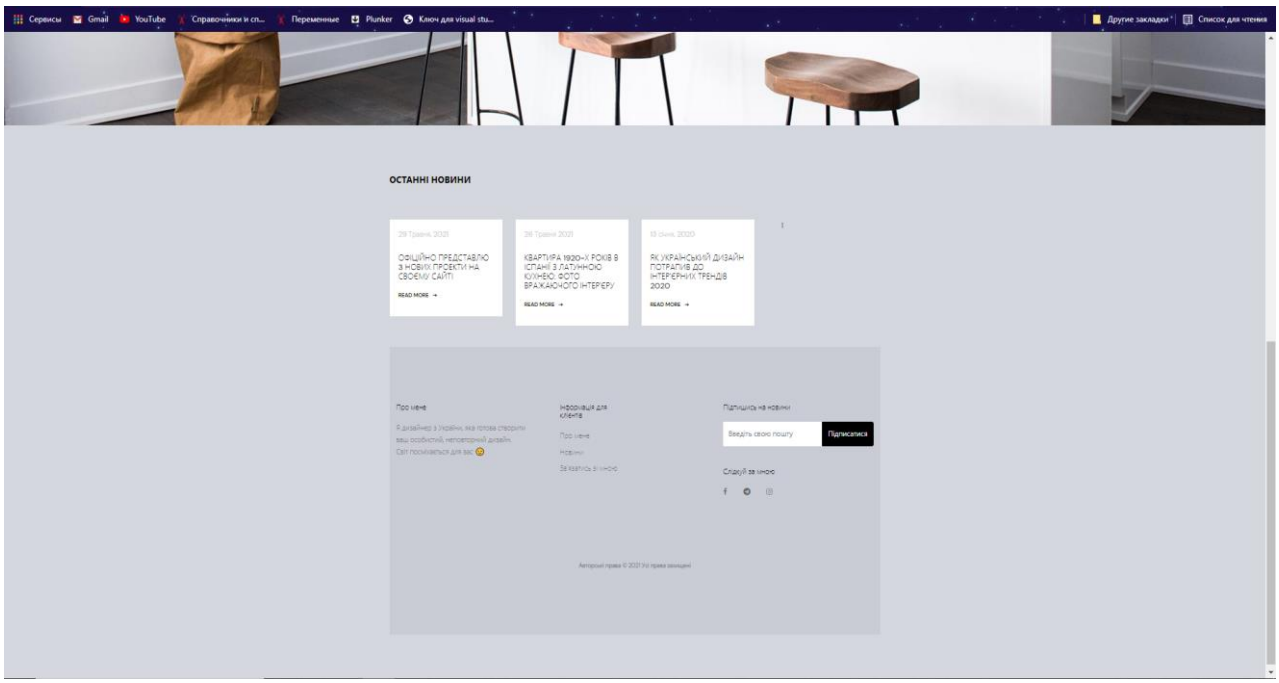

Рис 3.7. Відображення сторінки новини разом з додатковою інформацією

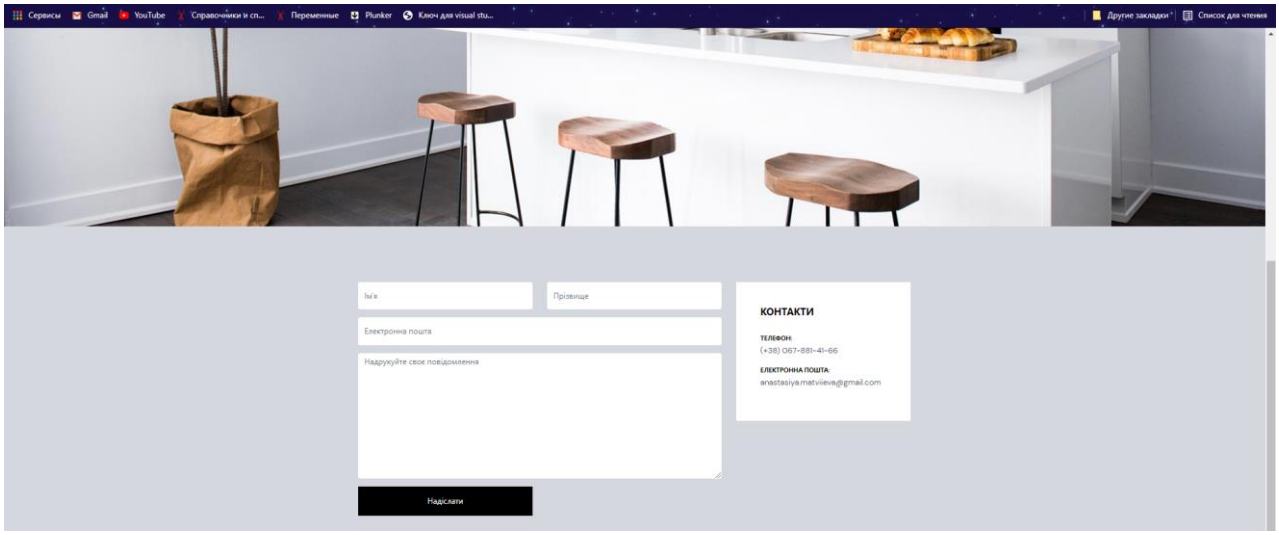

Рис 3.8. Відображення сторінки контактів

Відображення сайту в браузері Microsoft Edge:

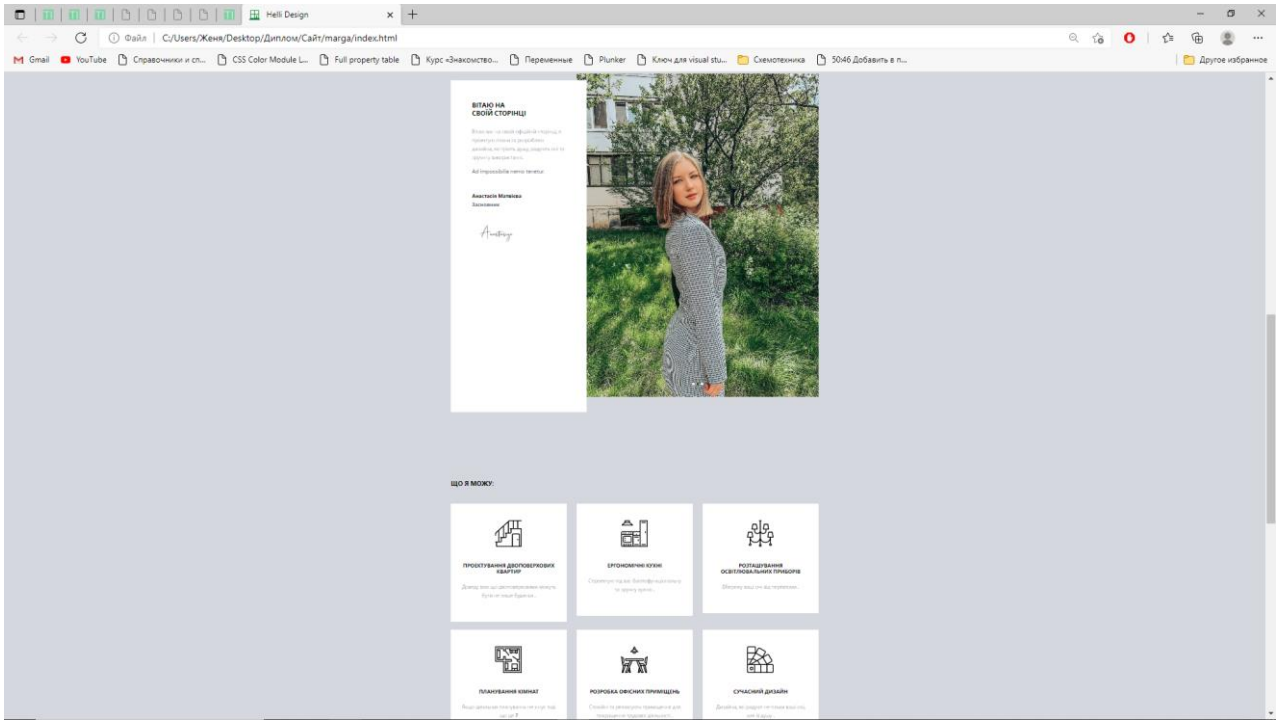

Рис 3.9. Відображення головної сторінки

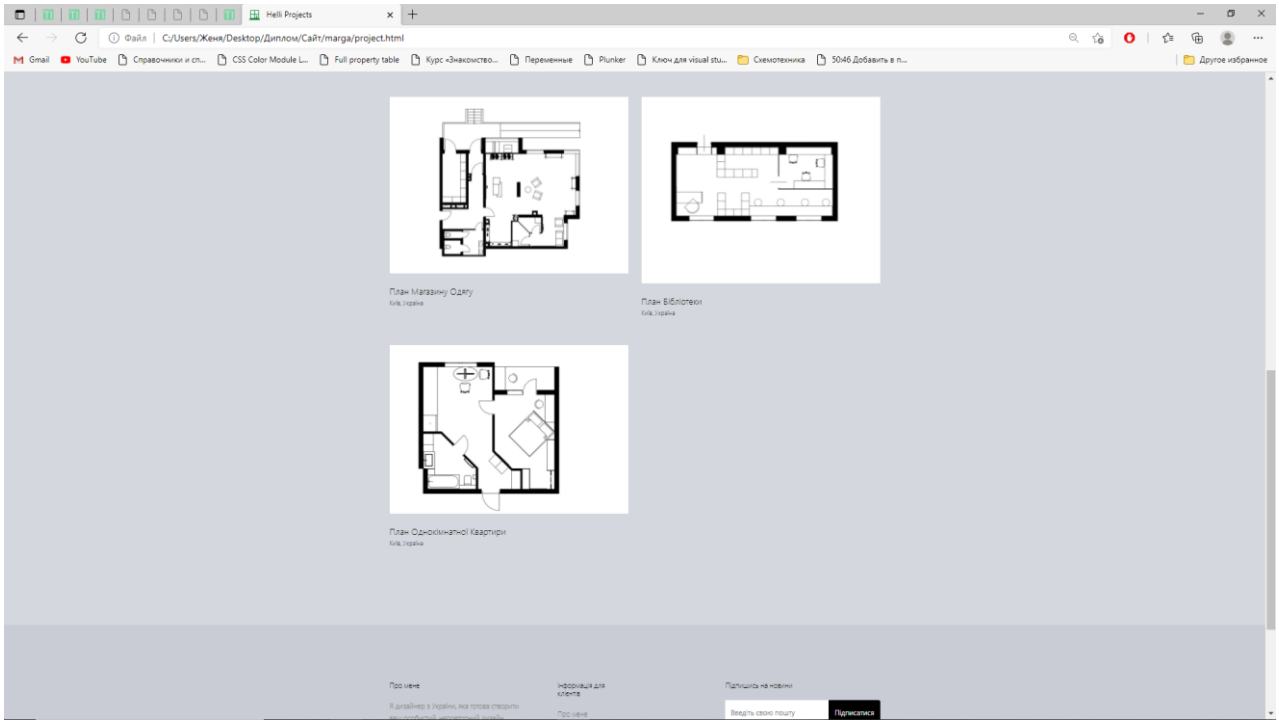

Рис 3.10. Відображення сторінки проектів

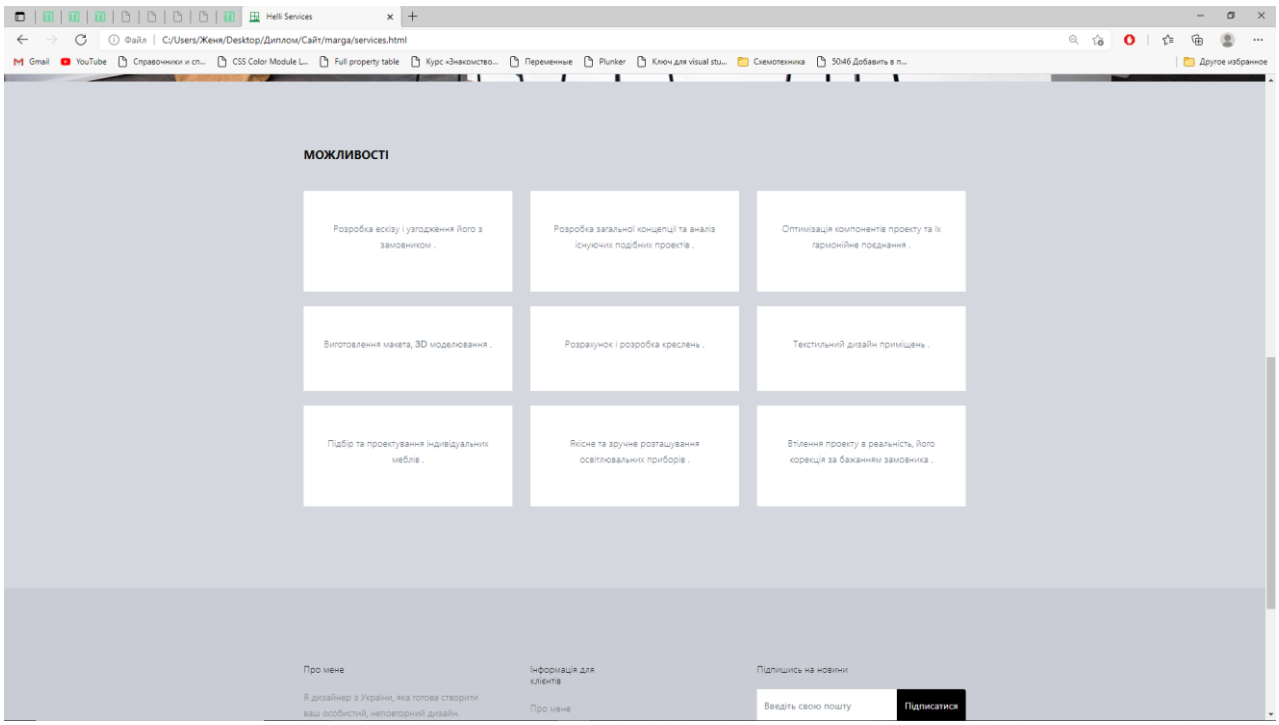

Рис 3.11. Відображення сторінки послуг

Висновок до розділу

В даному розділі було проаналізовано принципи реалізації сайту.

Для реалізації сайту було обрано ґратчасту структуру, оскільки вона є простою і універсальною.

Було розроблено внутрішню структуру сайту та вибрано кольорову гамму. Проведено реалізацію веб-сайту, а також проведено його тестування в різних браузерах.

Тестування показало:

- Сайт працює справно
- Коректно відображається в різних браузерах
- Всі елементи інтерфейсу працюють справно
- Кожна сторінка сайту адаптується окремо під пристрій
- Сторінка сайту адаптується під масштабування в браузері

ВИСНОВКИ

Метою даного бакалаврського дипломного проекту розробка методів і засобів для покращення візуального сприйняття та для поліпшення інтерфейсу для клієнтів та користувачів. В дипломному проекті проведено аналіз предметної області, існуючих варіантів розв’язання досліджуваної задачі. Проведено порівнальний аналіз аналогічних веб-ресурсів. Порівняльна характеристигка аналогів занесена до таблиці. Був проведений аналіз методів розв’язання задачі.

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. Після проведення аналізу було вирішено розробити веб-ресурс за допомогою веб-технологій HTML+CSS+SCSS

В якості середовища розробки було обрано Sublime Text 3, оскільки він швидкий і простий у використанні а також підтримує велику кількість плагінів, які значно спрощують та поліпшують процес розробки.

Таким чином, отримані результати доводять доцільність розробки веб-ресурсу.

Даний сайт повністю готовий для використання та може бути переданий на використання до осіб або компаній, які займаються дизайном. Та його шаблон може бути використаний для розвитку майбутніх новітніх дизайнів та створення нових сайтів.