Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка адаптивної системи управління контентом для Web-сайту туристичної фірми»

Студента 4 курсу, 13 групи,
спеціальності
122 «Комп'ютерні науки» _____ Мантюка Кирила Юрійовича
підпис студента

Науковий керівник
кандидат технічних наук, доцент _____ Козлов Валерій Володимирович
підпис керівника

Гарант освітньої програми
кандидат технічних наук, доцент _____ Демідов Павло Георгійович
підпис керівника

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____Пурський О. І.
« » грудня 2020р.

Завдання
на випускний кваліфікаційний проект студента

Мантюка Кирила Юрійовича
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка адаптивної системи управління контентом для Web-сайту туристичної фірми»
Затверджена наказом ректора від *«04» грудня 2020 р. № 4111*
2. Строк здачі студентом закінченої роботи *29 травня 2021 року*
3. Цільова установка та вихідні дані до роботи
Мета роботи: обґрунтування та розробка CMS-системи управління контентом сайту туристичної фірми, з урахуванням сучасних тенденцій побудови організаційних та функціональних інформаційних структур підприємств.
Об'єкт дослідження: процес проектування CMS-системи управління контентом сайту туристичної фірми
Предмет дослідження: засоби створення CMS-системи управління контентом сайту туристичної фірми
4. Перелік графічного матеріалу_____
_____
_____

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

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

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

ВСТУП

РОЗДІЛ 1. Поняття та сутність CMS-систем

1.1 Веб-сайт. Класифікація веб-сайтів (статичні та динамічні)

1.2 Мета та функції системи управління контентом

1.3 Огляд ринку CMS-систем

Висновки до розділу

РОЗДІЛ 2. Організація розробки CMS-систем

2.1 Функціональні вимоги до системи

2.2 Загальна структура CMS-систем

2.3 Вибір програмних та апаратних засобів для розробки CMS-систем

Висновки до розділу

РОЗДІЛ 3 Програмна реалізація CMS-систем управління контентом

3.1. Розробка інтерфейсу користувача

3.2 Розробка бази даних системи

3 3 Результати реалізації

Висновки до розділу

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	*Вибір теми випускної кваліфікаційної роботи*	*01.10.2020*	*01.10.2020*
2	*Розробка та затвердження завдання на випускну кваліфікаційну роботу*	*15.12.2019*	*15.12.2020*
3	*Вступ*	*03.02.2021*	
4	*РОЗДІЛ 1. Поняття та сутність CMS-систем.*	*28.02.2021*	
5	*РОЗДІЛ 2. Організація розробки CMS-систем.*	*06.04.2021*	
6	*РОЗДІЛ 3 Програмна реалізація CMS-систем управління контентом.*	*12.05.2021*	
7	*Висновки*	*15.05.2021*	
8	*Здача випускної кваліфікаційної роботи на кафедру науковому керівнику*	*20.05.2021*	
9	*Попередній захист випускної кваліфікаційної роботи*	*26.05.2021*	
11	*Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи*	*27.05.2021*	
12	*Представлення готової зшитої випускної кваліфікаційної роботи на кафедру*	*29.05.2021*	
13	*Публічний захист випускної кваліфікаційної роботи*	*За розкладом роботи ЕК*	

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

9. Керівник випускної кваліфікаційної роботи (проекту)
Козлов В. В.
(прізвище, ініціали, підпис)

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

11. Завдання прийняв до виконання студент-дипломник
Мантюк К. Ю.
(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____
____

Керівник випускної кваліфікаційної роботи (проекту)

________________ ___ ___________2021 р.
(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента ____________________
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

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

Завідувач кафедри ______________________________________ Пурський О.І.
(підпис, прізвище, ініціали)
«_____»_________________2021 р.

ЗМІСТ

АНОТАЦІЯ

У випускній кваліфікаційній роботі здійснено розробку сайту для туристичної фірми. Представлена робота є прикладом роботи сайту-візитівки для підприємства. Теоретично обґрунтовано Content Management System (Система управління веб-вмістом).

Ключові слова: CMS, веб-сайт.

ANOTATION

In the final qualifying work, a site was developed for a travel company. The presented work is an example of a business card site for an enterprise. The Content Management System is theoretically substantiated.

Keywords: CMS, website.

ВСТУП

Актуальність: актуальність питання розробки персонального сайту за два десятиліття з моменту презентації технології World Wide Web ні скільки не потьмяніла, а набрала ще більшої популярності. Перший сайт info.cern.ch містив тільки заголовок, текст і посилання, але в повній мірі міг продемонструвати можливості протоколу передачі даних HTTP, системи присвоєння адрес URL і гіпертекстової розмітки HTML. З тих пір змінилася структура сайтів, їх наповнення і представленість в різних сферах діяльності людини. Активно використовується графіка, медіаконтент, інтерактивність і динамічність, повсюдно використовується персоніфікація і різні варіанти отримання зворотного зв'язку. Якщо розробники планували використання цієї технології тільки для обміну інформацією в сфері науки і техніки, то зараз складно уявити без інтернету будь-яку сферу економіки і особистому житті людини. Тому така пильна увага стала приділятися змістом, зовнішнім виглядом, структурою, оформлення, навігації і т. д. Безумовно, назвати певну дату змін, що відбулися неможливо, однак відомо, що всі вони відбувалися в результаті розвитку технічної бази інтернет-мереж: продуктивність комп'ютерів, пропускна здатність каналів зв'язку, поліпшення характеристик пристроїв відображення інформації та інше. Слідом за ними змінюються і вимоги до сучасного і якісному веб-ресурсу. Оскільки процес розвитку «всесвітньої павутини» незворотній і триває досі, то при створенні сайту слід використовувати технології, які в подальшому дозволять оновити його структуру і оформлення без руйнування раніше створеного. Все це призвело до того що з’явилися CMS системи.

Мета роботи: обґрунтування та розробка CMS-системи управління контентом сайту туристичної фірми, з урахуванням сучасних тенденцій побудови організаційних та функціональних інформаційних структур підприємств.

Об’єкт дослідження: процес проектування CMS-системи управління контентом сайту туристичної фірми.

Предмет дослідження: засоби створення CMS-системи управління контентом сайту туристичної фірми.

Задачі дослідження:

- визначити які бувають веб-сайти;

– отримати загальні відомості про CMS;

– проаналізувати ринок CMS;

– розробка веб-сайту.

РОЗДІЛ 1
ПОНЯТТЯ ТА СУТНІСТЬ CMS – СИСТЕМ

1.1 Веб-сайт. Класифікація веб-сайтів (статичні та динамічні)

Веб-сайт – це сукупність логічно зв'язаної гіпертекстової інформації, оформленої у вигляді окремих сторінок і доступної в мережі Інтернет.

Подібне визначення веб-сайту було правильним на початку існування Інтернету, коли Мережа і веб-сайти використовувалися в основному як розважальна система. До кінця 90-х років веб-сайти дійсно були в основному статичними сторінками. Для створення веб-сайту було потрібне лише знання мови гіпертекстової розмітки – HTML. Якщо ж сторінка надавала якісь програмні засоби – це були виключно засоби, що міг надати сервер, на якому розташований веб-сайт. Про зручність і красу тогочасних веб-сайтів взагалі особливо не доводилося говорити. Час спливає, розвиваються мови програмування, розширюються канали передачі інформації. Зараз Інтернет вже є самодостатньою галуззю економіки, а веб-сайти стали повноправними представництвами фірм в Інтернеті.

Веб-сайт – це сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. По суті ж веб-сайт – це віддзеркалення успішності фірми, її обличчя.

Веб-сайт виконує такі основні завдання:

- реклама продукції, послуг, ідей;
- продаж товарів, послуг, інформації;
- безкоштовне надання інформації або послуг;
- підтримка клієнтів.

Типи веб-сайтів

Рекламні веб-сайти: Веб-сайти можуть створюватися виключно в рекламно-промоутерських цілях. Такі сайти безпосередньо не займаються продажом, їх завдання полягає в донесенні до цільової аудиторії рекламної інформації, і створюються вони з розрахунку на певне коло товарів або послуг. Зазвичай такі сайти виконуються з використанням великої кількості

графіки. Для залучення клієнтів на сайт використовують ігрові й розважальні методи.

Веб-сайти-продавці: Для таких сайтів характерна наявність описового рекламного матеріалу для товарів або послуг, каталог даних товарів або послуг, інформації про фірму продавця, а також контактна інформація. Додаткові сервіси, такі, як корисна інформація, зручність замовлення через сайт у поєднанні із грамотною розкруткою, можуть зробити веб-сайт привабливим для сторонніх рекламодавців.

Веб-сайти-"альтруїсти": Інформаційні веб-сайти, або сайти, які надають деякі безкоштовні сервіси, теж потрібно обслуговувати, розвивати, а отже, вкладати в них кошти. Але проекти, які не приносять прибуток, довго не живуть, тому для таких веб-сайтів характерне заробляння грошей або на рекламі, або на зборі статистичних даних. На таких сайтах дуже часто пропонують зареєструватися, аби отримати маленький додатковий сервіс. І що в результаті про нас дізнаються подібні веб-сайти? Все: від нашого браузера й роздільна здатність екрану до місця роботи й рахунку в банку. Якщо ж ви не реєструєтеся на сайті, то все одно можна взнати ваші смаки, психологічний портрет, приблизний вік, ну й, нарешті, вам можна показати багато реклами.

Веб-сайти для підтримки: Останній тип веб-сайтів – це підтримка клієнтів. Зазвичай на таких сайтах розміщують оновлення для програмних продуктів, новини; якщо йдеться про сайт банку, це може бути система управління засобами клієнта. Ці інтернет-ресурси є рекламою фірми, товару та інше.

Уся величезна кількість існуючих сайтів може бути розбита на 2 основні групи: статичні сайти й динамічні сайти.

Статичним прийнято називати сайт, що складається з незмінних, тобто статичних, HTML-сторінок. HTML-сторінка являє собою сукупність тексту, графічних зображень і власне мови гіпертекстової розмітки HTML, відповідальної за представлення сторінки в браузері.

Статичні HTML-сторінки створюються вручну, після чого при кожному звертанні до сайту представляються користувачеві в незмінному вигляді. Щоб оновити інформацію на подібних сторінках, необхідно вручну вносити зміни безпосередньо в програмний код сторінки.

Статичні сайти мають як свої переваги так і недоліки. До переваг статичних сайтів відносять наступні:

- статичні сайти створюють мінімальне навантаження на сервер, а тому невимогливі до ресурсів хостинга;
- статичні сайти завантажуються швидко;
- розробка статичних сайтів обходиться дешевше;
- перенести статичні сайти на новий хостинг дуже просто.

Серед недоліків статичних сайтів особливо виділяється складність оновлення сайту, внесення яких-небудь змін. Керування сайтом неможливе без знань і вмінь в області веб-програмування - це може викликати додаткові витрати при необхідності додавання нових матеріалів на сайт, нових розділів або категорій. А при розвитку сайту й збільшенні кількості сторінок взагалі стає важко підтримувати цілісність проекту, стежити за правильністю програмних кодів і та інше.

На відміну від статичних, динамічні сайти набагато більш гнучкі в керуванні. Динамічні сайти являють собою сукупність тексту й графіки, мови розмітки – точно так само, як і статичні сайти. Однак на додаток до цього динамічні сайти використовують також різні технології, що дозволяють «конструювати» веб-сторінки «на льоту».

Динамічні сайти можна розробляти «з нуля», вручну створюючи всі необхідні програмні коди, скрипти й т. д. Однак набагато частіше для створення динамічних сайтів використовуються спеціальні системи керування контентом – CMS. CMS дозволяють використовувати вже готові програмні модулі й компоненти, без необхідності щораз створювати їх «з нуля». На основі однієї CMS можна створити будь-яку кількість динамічних сайтів.

Динамічні сайти в браузері формуються з декількох частин або ж браузер заповнює інформацією вже готові шаблони сторінок. У динамічних сайтах реалізований поділ змісту й оформлення веб-сторінок - це дозволяє оперативно змінювати інформацію на сайтах без необхідності змінювати програмні коди сторінок.

Подібний підхід до формування веб-сторінок – одна з найголовніших переваг динамічних сайтів. Поділ контента й дизайну сайту дає можливість управляти сайтом будь-якому користувачеві, навіть без знання веб-програмування. В CMS для додавання й редагування матеріалів використовуються візуальні WYSIWYG-редактори (принцип «що бачу – те й одержую»).

Динамічні сайти можуть «підлаштовуватися» під своїх відвідувачів, реагуючи на їхні дії. Для цього використовуються технології серверних, клієнтських скриптів, за допомогою яких і створюються сценарії поводження сайту при певних діях користувачів.

Крім перерахованих переваг, динамічні сайти мають і ряд недоліків. У порівнянні зі статичними сайтами, динамічні більш «важкі», дають більше навантаження на сервер – отже, вони більше вимогливі до хостингу, ресурсів сервера.

Щоб динамічні сайти «працювали» потрібно додаткове програмне забезпечення, тоді як для відображення статичних сайтів досить одного лише браузера. Це робить розробку й підтримку динамічних сайтів більше дорогою у порівнянні зі статичними сайтами.

Однак зовсім необов'язково створювати складні динамічні сайти для розв’язання простих завдань, наприклад для реалізації сайтів-візиток з 3-5 сторінок. У цьому випадку на сайті практично не потрібне оновлення контента, не потрібна наявність інтерактивних функцій – сайт може бути статичним.

У свою чергу, статичні сайти не впораються зі складними інтерактивними завданнями. Наприклад, неможливо створити інтернет-

магазин, використовуючи лише статичні HTML сторінки, він просто не буде працювати.

Таким чином, залежно від поставлених завдань можуть використовуватися як статичні, так і динамічні сайти.

1.2 Мета та функції системи управління контентом

Content Management System (Система управління веб-вмістом) – програмний комплекс, що надає функції створення, редагування, контролю та організації веб-сторінок. CMS часто використовуються для створення блогів, особистих сторінок і інтернет-магазинів і націлені на користувачів, мало знайомих з програмуванням.

Використання CMS має цілий ряд переваг:

- Завдяки різноманітності CMS ви підберете відповідне програмне забезпечення. Така програма дозволить вам швидко і продуктивно вирішувати поставлені завдання.
- За допомогою CMS ви будете створювати, змінювати і видаляти розділи. Крім того, ви можете редагувати дані без стороннього втручання. Це є основною перевагою такої системи над статичними сайтами.
- У процесі роботи веб-ресурсу користувачі знаходять безліч помилок. Ця програма дозволяє швидко і ефективно усувати несправності. Сайт працює на сучасних і перевірених технічних рішень.
- Витрати на створення сайту істотно знижуються. Розробнику не потрібен час на вирішення технічних завдань.

Основні функції CMS:

- Створення - користувач отримує універсальний інструмент зі створення контенту.
- Управління - ви зможете обмежувати доступ до інформації, додавати, редагувати, видаляти і багато іншого.
- Публікація - стаття повністю адаптується до дизайну сайту, необхідно тільки внести її в потрібне поле.

• Подання – ви можете змінювати кольорову палітру, розташування і багато іншого будь-якого елементу на вашому сайті.

Переваги використання CMS:

• Зручність. Дружній інтерфейс і зрозуміле управління є основними плюсами цього програмного забезпечення.

• Економія. Вам не знадобиться вдаватися до послуг фахівців і платити за це додаткові гроші.

• Безпека. Ви завжди можете зробити відкат і будь-яку допущену помилки не будуть критичними. Також система протистоїть атакам хакерів. Ваша інформація буде в безпеці.

• Документація. Модулі мають help-файли. За допомогою цих документів ви розберетеся в функціях вашої CMS.

• Мультишаблонность. Більшість CMS підтримує численні шаблони. Крім того, розробники постійно оновлюють систему і надають на ваш вибір нові види.

• Функціональність. Кожен розділ або підрозділ має свої настройки і конфігурації. Залежно від ваших вимог ви зміните свій сайт аж до найдрібніших деталей.

• Комплексність. Можна створювати окремі вкладки і надати контроль над ними одному або декільком співробітникам (стрічка новин, блоги та ін.).

• Незалежність. Користувач програмного забезпечення не залежить від розробників. Він може змінювати конфігурації від свого профілю. Вам не буде потрібно дозвіл або згода розробника.

• Можливість розширення. Ви можете змінити свою систему і при цьому не втратити інформацію. В процесі перенесення ваш сайт буде функціонувати як і раніше.

• Привабливий зовнішній вигляд. Різна колірна гамма і зручна панель дозволить вам отримувати естетичне задоволення від роботи.

- Керованість. Система проста у використанні. За допомогою пару кліків ви можете змінити структуру, поміняти місцями пункти, розділи, кореневу папку, підняти рядок вище або нижче і багато іншого.

Кожна CMS незалежно від виробника створювалася з урахуванням всіх вимог користувача. Сучасний ринок представлений різним програмним забезпеченням. Деякі системи орієнтовані на вирішення конкретних завдань, а інші стали універсальними і практичними для будь-якого типу контенту. За допомогою програм ви будете коригувати, додавати, видаляти інформацію різного типу на вашому сайті. В деяких CMS є безліч функціональних розділів і підрозділів, інші, навпаки, складаються з єдиної системи.

Програмне забезпечення може бути як платним, так і безкоштовним. Крім того, виробники передбачили модулі з відкритим і закритим кодом. Тобто ви зможете вносити свої пропозиції в роботу CMS. Комп'ютерний світ не запропонував чіткої класифікації програм. Але на сучасному ринку з'явилися явні лідери.

1.3 Огляд ринку CMS-систем

В даний час на ринку є близько 300 різних CMS. При управлінні контентом з їх допомогою зовсім не потрібні навички програмування, а адміністрування зводитися до простих повторюваним функцій - наприклад, створення нових розділів і копіювання тексту з Word у вбудований редактор. При цьому використовується графічний користувальницький інтерфейс, який інтуїтивно зрозумілий більшості людей епохи смартфонів.

Саме програмне забезпечення являє собою інтерактивний ресурс, так званий веб-додаток. На практиці це працює таким чином, що співробітники заходять на свою сторінку входу, щоб потрапити на ту частину сайту, яка невидима для відвідувачів. Ця частина називається бекендом. Загальнодоступною частиною веб-сайту є інтерфейс.

Бекенд використовується для настройки ресурсу і управління контентом. Без CMS оператори веб-сайтів повинні були б редагувати сторінку при кожній зміні за допомогою редактора HTML, а потім

завантажувати її на сервер за допомогою програми FTP. З CMS це більше не потрібно.

Простота управління робить ці системи ідеальними для підтримки як великих, так і одно сторінкових сайтів. Деякі з CMS безкоштовні, деякі з відкритим вихідним кодом, що означає, що кожен власник може вносити правки під вимоги власних унікальних проектів.

Найважливіша відмінність різноманітних систем - в складності роботи з ними. Тобто, деякі підходять для новачків (WordPress), а для деяких потрібен мінімальний досвід користування (Drupal). За типом роботи з даними, всі системи поділяються на 3 класи. Класичні системи управління контентом з підключеною базою даних. Звичайні CMS, наприклад такі, як WordPress, Joomla, Drupal та TYPO3, працюють з підключеними базами даних. Це означає, що весь контент зберігається в окремій базі даних. В якості альтернативи звичайному CMS, які працюють з підключеними базами даних, існують так звані плоскі файлові системи. Вони зберігають вміст сайту у вигляді простих файлів, тому їм не потрібна власна база даних на сервері. До них відносяться такі продукти, як Grav, Pico і Kirby.

На додаток до систем управління контентом з базою даних або без неї на ринку з'являється все більше статичних генераторів веб-сайтів. Вони не заповнені контентом, як звичайні CMS. Кожен раз, коли вносяться зміни, система створює статичні HTML-файли і відновлює сторінку. Таким чином, сторінки статичні і мають високу продуктивність. Генератори більш підходять для професійних, технічно підкованих користувачів, але робота з ними стає все простіше завдяки додатковим сервісам і інтерфейсів адміністратора. Кращими представниками тут є Forestry.io, DatoCMS і Lektor.

За моделлю управління, всі системи поділяються на 4 типи:

- CMS Програмне забезпечення управління контентом незалежно від виду презентації даних.
- WPS Спрощена CMS для блогів.

• WCMS Система управління для веб-контенту, адаптована для роботи з мобільних пристроїв.

• ECMS Система управління контентом підприємств.

До найпопулярніших CMS можна віднести:

WordPress

WordPress починався як просте програмне забезпечення для створення блогу. З роками система ставала все більш популярною: з'являлися нові функції і шаблони. Такі рішення, як «5-хвилинна установка», безліч безкоштовних тем і проста інтеграція плагінів, безумовно, сприяли успіху цієї CMS.

Якщо вам потрібен сайт, але у вас практично немає бюджету і часу на нього, WordPress – це правильний вибір. Система встановлюється за кілька хвилин, потім користувачі можуть відразу ж почати публікувати контент. Вклавши трохи грошей в якісну тему, ви зможете швидко домогтися професійного зовнішнього вигляду. Якщо з часом виникають більш конкретні вимоги до проекту, ви зможете швидко знайти готове рішення у вигляді плагіна.

Joomla

Як і WordPress і TYPO3, Joomla заснована на мові програмування PHP, який візуально перетворює вміст бази даних MySQL. Установка Joomla працює практично на кожному веб-хостингу і займає всього 30 секунд. Розробники можуть запрограмувати велика кількість індивідуальних розширень для системи. На відміну від інших CMS, Joomla пропонує тільки поле введення з підтримкою редактора для текстів для створення і форматування контенту. Це виглядає дуже просто на перший погляд, але викликає проблеми при складному форматуванні.

Drupal

Колись задуманий як соціальна платформа для обміну інформацією, Drupal перетворився в одну з найбільш широко використовуваних систем редагування з відкритим вихідним кодом. На додаток до основних функцій,

Drupal фокусується на розробці соціальних публікацій і порталів спільноти, щоб учасники могли створювати власний контент і взаємодіяти з іншими учасниками. В Drupal, як і в WordPress і Joomla, розділи управляються на основі об'єктів. Модульна структура CMS дозволяє реалізувати окремі і складні структури сторінок. Спочатку досить тонка, вона може бути адаптована до ваших власних бажань за допомогою різних додаткових розширень. В цілому, Drupal більше підходить для досвідчених веб-розробників, тому що, на відміну від WordPress, бажана конфігурація повинна бути спочатку зібрана в певному порядку.

TYPO3

TYPO3 є найбільш популярною системою для великих фірм і компаній. CMS доступна на більш ніж 50 мовах і має понад 5000 розширень. Велике співтовариство постійно і активно бере участь у розвитку системи. Навіть у базовій установці ця потужна CMS включає в себе безліч функцій, таких як підтримка декількох доменів і розширене управління правами для кількох адміністраторів і користувачів.

Для реалізації ж складних структур сторінок з багатомовним контентом, система редагування вимагає спеціальних знань. В цілому, TYPO3 – це неймовірно складна CMS, яка зазвичай може робити більше, ніж потрібно користувачам на їх сайті. У порівнянні з іншими системами, вона вимагає тривалого періоду навчання і порівняно великих адміністративних зусиль.

MODx

MODx Evolution є поєднанням CMS і CMF – з «Системи управління контентом» і «Структури управління контентом», тобто платформи. Система підходить як для невеликих, так і для великих веб-сайтів, тобто від веб-візитки до ресурсу міжнародної компанії.

Найбільша перешкода, яку ви повинні прийняти в MODx, це ви самі: MODx – це система для професіоналів. Для використання необхідно мінімальне знання HTML і CSS. У MODx особлива увага приділяється

зручності використання створених веб-сайтів і оптимальному поєднанню дизайну і контенту. Для початку знання програмування вам не будуть потрібні, однак, як тільки ви будете хотіти реалізувати спеціальні та індивідуальні рішення, потрібно буде застосувати CSS і HTML. MODx управляє вмістом і надає розмітку, яка повинна бути вказана до останньої деталі. Після успішної установки MODx надає порожню білу сторінку з мінімальним початковим кодом HTML. Так починається кожен проект MODx.

CMS так само різноманітні, як і веб-сайти, для яких вони використовуються. Drupal непростий для початківців, а WordPress для повноцінної роботи вимагає установки десятка плагінів. Що ж стосується MODx, то її частіше вибирають професійні веб-дизайнери. Як правильно вибрати CMS:

1. Визначення області. Якщо ви зібралися робити блог - вам потрібен WordPress, якщо сайт візитку або каталог краще зупинити вибір на Joomla. Якщо форум можна використовувати Drupal. Якщо корпоративний сайт – MODx або TYPO3. Для створення інтернет-магазину підійде будь-яка CMS, але WordPress і TYPO3 мають переваги в захисті транзакцій.

2. Визначення часу навчання. При повній відсутності досвіду, ви швидше за все освоїте WordPress, на другому місці по легкості варто Joomla, потім Drupal. MODx або TYPO3 більш складні в освоєнні.

3. Визначення часу зайнятості. Тут картина аналогічна. Найбільш просто адмініструвати WordPress, а найбільш складно – MODx.

4. Визначення ефективності SEO. В принципі, рівень якості SEO не так вже й сильно залежить від CMS, однак WordPress лідирує за кількістю плагінів для оптимізації.

Висновок до першого розділу:

З першого розділа можна зробити висновок що з часом веб-сторінки ставали все складніші в розробці і було все важче вийти на ринок веб-

додатків. Все це привело до того, що з’явилися CMS системи. Саме завдяки CMS люди без знань в програмуванні можуть створювати свої веб-сайти.

Також можна зрозуміти, що ринок CMS досить великий. Перед тим як починати розробку треба зрозуміти який потрібен веб-сайт і вже від цього обирати CMS систему.

РОЗДІЛ 2
ОРГАНІЗАЦІЯ РОЗРОБКИ CMS – СИСТЕМ

2.1 Функціональні вимоги до системи

Дотепер ще відсутні загальноприйняті терміни, якими користуються спеціалісти для опису вимог замовника до ПС, а саме, функціональних, системних, технологічних. У різних тематичних джерелах наведені різні визначення поняття вимог, виходячи з особистих поглядів замовників на програмний продукт, який потрібно побудувати. У ряді публікацій формування вимог до ПС розглядається як певна ділова гра, під час якої виявляються інтереси зацікавлених у розробці ПС осіб, правила реалізації цих інтересів у конкретному продукті. При цьому висловлюються різного роду претензії й обмеження на властивості і способи забезпечення вимог для отримання кінцевого програмного продукту. Отже, зрозуміло що нема формалізованих методів збирання й опису вимог, а також відсутнє загальноприйняте визначення самого поняття вимога.

Вимоги – це твердження про функції й обмеження системи.

Вимоги – це властивості, які повинен мати продукт, щоб бути цінним для своїх користувачів.

Вимоги – це специфікація того, що і як повинно бути реалізовано.

Відповідно до міжнародного глосарію з термінології комп’ютерної науки вимога містить у собі опис:

1) умов або можливостей, необхідних користувачеві для вирішення поставлених проблем або для досягнення цілей;

2) функцій і обмежень, які повинна мати система або системні компоненти, щоб виконати контракт замовника на розробку;

3) положень і регламенту, встановлених використаними стандартами і відображених у специфікаціях або інших формальних документах на систему;

4) умов, можливостей і обмежень середовища, необхідних для проектування і виконання системи.

Основні типи вимог:

Вимоги до продукту охоплюють умови користувачів щодо зовнішнього поводження системи і погляди розробників на деякі параметри системи. Термін користувач стосується осіб, зацікавлених у створенні системи.

Вимоги до ПЗ є такі: системні, функціональні і нефункціональні вимоги.

Вимоги користувачів (user requirements) задаються умовами досягнення цілей і задач, віддзеркалюють вимоги споживачів до спектра розв’язуваних майбутньою системою задач. Вони подаються як текстовий опис або сценарії, прецеденти, таблиці «подія-відгук» тощо.

Системні вимоги (system requirements) визначають зовнішні умови виконання системних функцій і обмежень на створення продукту, а також вимоги до опису програмно-апаратних підсистем. Системні вимоги накладають обмеження на архітектуру системи, засоби її візуального подання і функціонування. Для опису системних вимог використовують спеціальні шаблони і форми, що допомагають уявити вхідні і вихідні дані й автоматизувати ці вимоги.

Вимоги до атрибутів якості (quality attributes) – це деякі обмеження на властивості функцій або системи, важливі для користувачів або розробників. Наприклад, переносність, цілісність, стійкість системи до збоїв.

Функціональні вимоги – це перелік функцій або сервісів, які повинна надавати система, а також обмежень на дані і поводження системи при їхньому виконанні. Специфікація функціональних вимог (software requirements specification) – опис функцій та їхніх властивостей, які не містять у собі протиріч і виключень. Нефункціональні вимоги визначають умови виконання функцій (наприклад, захист інформації у БД, автентифікація доступу до ПС тощо) у середовищі, що безпосередньо не пов'язані з функціями, а відбивають потреби користувачів щодо їх виконання. Ці вимоги характеризують принципи взаємодії із середовищами або іншими системами, а також визначають показники часу роботи, захисту

даних і досягнення якості з урахуванням рекомендацій використовуваного стандарту.

Вони можуть встановлюватися як числові значення (наприклад, час чекання відповіді, кількість клієнтів, що обслуговуються і ін.) у різних одиницях виміру, включаючи, наприклад, ймовірність (значення ймовірності безвідмовної роботи системи – показника її надійності). Для більшості сучасних ПС вимоги складаються з умов й обмежень типу:

– конфіденційність, безпека і захист даних;

– відмово стійкість;

– одночасність доступу користувачів до системи;

– час чекання відповіді при звертанні до системи (продуктивність);

– склад виконуваних функцій системи (запуск, швидкість реакції й ін.);

– положення стандартів з виконання сформульованих вимог.

Дані вимоги визначаються і формалізуються аналітиками на процесі аналізу і проектування структури системи. Так, у випадку вимог з безпеки функціонування системи, в системі виділяються категорії користувачів, що мають право доступу до тих або інших функцій (програмних компонентів) або даних, та передбачаються додаткові функції системи з перевірки доступу (санкціонований доступ до них чи ні). Якщо потрібно обмежити доступ до конкретних даних (наприклад, до окремих записів, полів у таблиці), то в системі може передбачатися, наприклад, мандатний захист.

Для захисту всієї системи від несанкціонованого доступу користувачі реєструються і проходять автентифікацію для роботи із системою. Для відновлення і збереження резервних копій БД, архівів баз даних аналізуються можливості СУБД і способи забезпечення необхідного рівня безперебійної роботи системи, правил доступу авторизованих користувачів і заходів боротьби з різними загрозами, що надходять ззовні від користувачів, які не мають прав доступу до деяких або до всіх даних системи.

До вихідного продукту пред'являються нефункціональні вимоги до:

– застосування (якість інтерфейсу, продукту й ін.);

– продуктивності (пропускна здатність, час реакції й ін.);

– надійності виконання (без помилок і відмов);

– зовнішніх інтерфейсів, за якими виконується взаємодія з іншими компонентами або підсистемами.

Опис усіх видів вимог проводиться з урахуванням стандартів, наприклад, стандарту з якості ISO/IEC ДСТУ 9126 і стандартизованого понятійного і термінологічного довідника, що містить у собі загальноприйняті терміни щодо структури ПрО і призначення функцій системи. Специфікація вимог відображає принципи взаємодії проектованої системи з іншими середовищами, платформами і загальносистемним забезпеченням (БД, СКБД, мережі та ін.). Формування документа зі специфікаціями вимог завершується на процесі проектування архітектури, після чого він узгоджується з замовником системи і використовується як керування дій при виконанні задач розробки програмного продукту на процесах ЖЦ і отриманні готового продукту. При виявленні на них неузгоджених вимог, проводиться їхнє уточнення і, відповідно, вносяться зміни у деякі задачі процесу розроблення системи або характеристики продукту.

2.2 Загальна структура CMS-систем

Сьогодні для створення і функціонування сайту в мережі існують зручні інструменти. Сучасні CMS дозволяють наповнити сайт необхідним функціоналом і зручно управляти його вмістом. Навіть безкоштовні рішення дозволяють новачкам без проблем наповнювати свій сайт інформацією, не маючи при цьому особливих знань. Але в будь-якій справі є винятки, які вимагають особливого підходу.

Часом сайт повинен мати особливий функціонал або відповідати певним вимогам, які ставить перед розробником замовник. В такому випадку доводитися розробляти додаткові модулі, і не завжди це зручно, а часом і не можливо зробити зі «стандартної» CMS. У подібних випадках для сайту розробляється унікальний «движок», хоча це трапляється вкрай рідко. У

більшості випадків розробники (студії) створюють фірмові CMS з інших причин, пропонуючи користувачеві додаткові зручності, функціонал та безпеку. Модульність, розширюваність і простота в управлінні - це основні вимоги до любого веб-проекту. Сама CMS повинна забезпечувати лише базовий функціонал (управління сторінками, структурою сайту і редагування інформації на ньому), який в міру розвитку спроможний розширюватися.

Основна вимога – це гнучка конфігурація сайту за допомогою функціональних модулів. Вони повинні розширювати функціонал сайту в будь-яких межах, від сайту-візитки, до Інтернет магазину.

Дуже важливо робити адміністрування сайту максимально простим і зрозумілим, щоб клієнт вже через півгодини самостійно міг додавати сторінки, редагувати інформацію, управляти розділами і меню на сайті. Тому, треба максимально спростити процес адміністрування, залишивши лише необхідні функції, які б знадобилися недосвідченому власнику сайту.

Список базових функцій (операцій) адмін-панелі CMS:

- Загальні налаштування сайту.
- Створення сторінок (WYSIWYG редактор).
- Управління сторінками (редагування властивостей і змісту, видалення).
- Управління розділами (додавання, редагування властивостей).
- Управління меню (додавання, редагування посилань).
- Редактор дизайну (візуальний редактор для шаблонів HTML).
- Робота з модулями (управління настройками модулів).

Цей функціонал повинен задовольнити більшість користувачів (адміністраторів), тому, варто сфокусуватися на зручному інтерфейсі і ергономіці, не навантажуючи її зайвими елементами. Розпочинати роботу слід з файлу index.php, потім з'являються необхідні каталоги, які поступово наповнюються скриптами. Змінюється їх структура, код переміщається з файлу у файл і попутно оптимізується. Функції об'єднуються в класи, а в базі

даних з'являються нові таблиці і колонки. Перевірка, налагодження та доопрацювання.

Компоненти веб-сайту які забезпечують його структуру та наповнення, можна умовно поділити на «логічний» і «фізичний» рівень:

- Таблиці баз даних визначають структуру сайту, і його наповнення. Ці дані і є «логічним» (інформаційним) рівнем.

- «Фізичний» (файловий) рівень містить файли шаблонів і контент.

Шаблон це текстовий файл (.html) з фрагментом коду HTML, який визначає дизайн певної частини сторінки і знаходитися в спеціальному каталозі. Сторінки сайту умовно поділено на шість зон (заголовок, ліва колонка, центр, права колонка, нижня лінія і підвал), які складають головний шаблон (каркас). Для кожної зони своя група шаблонів, умовний поділ якої визначає лише дизайн і назву файлу. Тобто, сторінка може бути побудована як мінімум з одного шаблону (наприклад заголовка), і як максимум з шести. Склад шаблонів зберігається в межах розділу, і обумовлює зовнішній вигляд в ньому. Різні розділи відповідно можуть містити різні шаблони і модулі, якщо звичайно в цьому є необхідність. Спеціальний каталог з скриптами php є модулем, і містить основні файли, що включаються (скрипти) для сайту і адмін-панелі (властивості і налаштування модуля). Складним питанням є механізм включення модулів в основній виконуваний файл. Серед різних варіантів реалізації цього процесу варто зупинитися на «напівавтоматичному». Для вставки модуля в тому чи іншому місці сторінки, необхідно прописувати спеціальні «мітки» (у вигляді спеціального HTML коментаря), які в процесі обробки замінюються на вміст індексного файлу модуля і вбудовуються в загальний виконуваний код. А вже в адмін-панелі CMS адміністратор визначає, який саме модуль, куди і в який розділ необхідно встановити.

Звичайно це не ідеальний спосіб, адже для включення модуля необхідно правити шаблони, але оскільки CMS розробляється тільки для

однієї студії і є «закритою», і всі маніпуляції буде здійснювати розробник, а не замовник, то цей механізм виявився цілком зручним.

Таблиця розділів містить такі основні колонки:

- ID розділу (унікальний ключ, і ключ прив'язки до сторінок).
- Префікс розділу (службове ім'я).
- Назва розділу (назва для зручної ідентифікації або виведення в заголовок).
- Опис розділу (розширена інформація для адміністратора).
- Колонки шаблонів (шість колонок для прив'язки шаблонів на каркас сайту).
- Колонки мод

2.3 Вибір програмних та апаратних засобів для розробки CMS-систем

Сучасні технології створення та підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно керувати інформаційним наповненням і даними, які надходять від відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP або використовують готові потужні засоби для створення корпоративних сайтів, що орієнтовані на впровадження вказаних технологій.

Створення веб-сторінок за фрагментами серверного коду є технологією ASP, ASP.NET (Active Server Pages). Характерною особливістю такої технології є можливість відокремлення функціональної частини розробки від процесів створення дизайну. ASP-сторінки можуть містити HTML-текст, змішаний зі сценаріями мов JavaScript і VBScript. У процесі обробки запиту нової сторінки його виконує сервер і динамічно генерує браузеру потік HTML-тексту, що відображується на екрані монітора. ASP-технологія Microsoft набула подальшого розвитку у технологіях JSP, PHP та ін. Технологія JSP (Java Server Pages) − це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є

альтернативою технології ASP корпорації Microsoft. Специфікація Sun за назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом та орієнтована на розробку серверних компонентів створення інтерфейсу. Однією з перших технологій створення веб застосувань, які виконуються сервером, була Common Gateway Interface (CGI) технологія. Вона дозволила розробку і виконання серверних застосувань, звернення до яких відбувається за допомогою зазначеного в URL імені (та параметрів).

Залежно від обраного протоколу вхідною інформацією таких веб-додатків вважають безпосередньо код HTTP-заголовка або запит пошукової системи. CGI- 60 застосування − це консольні додатки, які генерують HTML-код, переданий браузеру. Подібні застосування можуть являти собою код на скриптових мовах, який інтерпретується на сервері. Крім того, CGI-застосування презентують робочий файл, котрий можна створити за допомогою будь-якого засобу розробки, що генерує консольні застосування для операційної системи, під керуванням якої функціонує веб-сервер.

Серед інших популярних технологій, які реалізують створення веб-сторінок із фрагментами коду, виконуваного на сервері, виділимо некомерційну, вільно розповсюджувану технологію PHP (Personal Home Pages). Ця технологія заснована на використанні CGI-застосувань, що інтерпретують впроваджений у HTML-сторінку код на скриптовій мові. Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого й ефективного вирішення поставлених завдань. Вона вирізняється винятковою гнучкістю до потреб розробника. Хоча PHP традиційно рекомендують використовувати у поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови Інтернет-програмування.

Хоча PHP є досить молодою мовою, вона користується значною популярністю серед web-програмістів і сьогодні вважається мало чи не найпопулярнішою мовою для створення web-додатків (скриптів). Головними

перевагами PHP є практичність, легкість у застосуванні, ефективність, продуктивність та гнучкість.

Всі методи розробки сайтів можна умовно розділити на дві основні групи. Перша група методів розробки сайтів – це методи ручного написання сайтів на одній або декількох мовах веб-програмування. При цьому робота може здійснюватися як у простих, так і візуальних редакторах HTML та CSS.

У випадку створення статичного сайту цілком достатнім для ручного написання буде використання «зв'язування» HTML і CSS, з можливим включенням JavaScript. Для створення ж динамічного сайту не обійтися без серверних скриптів, таких як PHP, ASP.NET.

При використанні «ручних» методів розробки сайту дизайн сайту – графічне оформлення також створюється вручну. Для цих цілей застосовуються будь-які 61 графічні редактори за бажанням. Вручну можна відредагувати й уже готові шаблони дизайну, як платні так і безкоштовні. Друга група методів розробки сайтів містить у собі методи автоматизованого створення сайтів: за допомогою спеціальних конструкторів сайтів або ж систем керування контентом (CMS). Конструктори сайтів – це, як правило, онлайн-системи, що дозволяють із готового типового набору модулів і компонентів отримати сайт і відразу ж розмістити. Методи розробки сайтів з використанням CMS – одні з найбільш популярних на сьогоднішній день. CMS, умовно, являє собою певну готову візуальну й програмну оболонку, що користувач може заповнити необхідним контентом, а також за своїм бажанням змінити й настроїти. Методи ручної розробки сайтів досить складні, адже вони вимагають значних пізнань в області веб-програмування або дизайну сайтів. Однак вони мають незаперечну перевагу: створюючи сайт вручну, завжди можна одержати саме те, що хочеш. «Ручним» методам розробки зазвичай і надають перевагу саме тому. Широкі можливості для розробки сайтів будь-якої складності надають CMS. Саме цей метод розробки сайтів вважається одним з найбільш зручних і практичних. Гнучка система настроювань, можливість редагування самої CMS або ж окремих її

елементів, легкість додавання й зміни контенту – все це зробило розробку сайтів на базі CMS по-справжньому ефективною.

Висновки до другого розділу:

З другого розділу можна зрозуміти, що існує багато вимог до оформлення веб-сайту, але якщо використовувати CMS то зазвичай весь набір інструментів, функцій та скриптів представлений вже там і підлаштуватись під вимоги буду набагато простіше використовуючи CMS.

РОЗДІЛ 3
ПРОГРАМНА РЕАЛІЗАЦІЯ CMS – СИСТЕМ УПРАВЛІННЯ КОНТЕНТОМ

3.1 Розробка інтерфейсу користувача

Інтерфейс користувача (UI – англ. User interface – призначений для користувача інтерфейс) являє собою сукупність засобів і методів, за допомогою яких користувач взаємодіє з різними пристроями і апаратурою.

Іншими словами, це той набір кнопок, посилань, форм, діалогових вікон, іконок, піктограм, банерів, повзунків і стрічок прокрутки, за допомогою якого користувач управляє продуктом. Інтерфейс – тільки половина у взаємодії з системою, інша половина – людина, користувач. Для хорошої роботи інтерфейсу потрібно точно знати, що саме в будь-який конкретний момент користувач сприймає в інтерфейсі, про що думає, чого хоче отримати. Інтерфейси є основою взаємодії всіх створених інформаційних систем. Якщо інтерфейс будь-якого об’єкта (персонального комп’ютера, програми, функції) не змінюється (стабільний, стандартизований), це дає можливість модифікувати сам об'єкт, який не перебудовує принципи його взаємодії з іншими об’єктами. Наприклад, навчившись працювати з однією програмою Microsoft Office, користувач з легкістю освоїть і інші, тому що вони мають однаковий інтерфейс.

Інтерфейс ділиться на наступні компоненти:

- Інтерфейс командного рядка – інструкції комп'ютера даються шляхом введення з клавіатури текстових рядків (команд). В цьому вигляді інтерфейсу людина подає «команди» комп'ютера, а комп’ютер їх виконує і видає результат людині. Командний інтерфейс реалізований у вигляді пакетної технології та технології командні рядки;
- Графічний інтерфейс користувача (або WIMP - інтерфейс: Window - вікно, Image - образ, Menu - меню, Pointer - покажчик) – програмні функції представляються графічними елементами

екрану. Характерною особливістю цього виду інтерфейсу є те, що діалог з користувачем ведеться не з допомогою команд, а за допомогою графічних образів - меню, вікон, інших елементів. Хоча і в цьому інтерфейсі подаються команди машині, але це робиться «опосередковано», через графічні образи. Даний вид інтерфейсу реалізований на двох рівнях технологій: простий графічний інтерфейс і «чистий» WIMP - інтерфейс. Відмінні риси простого графічного інтерфейсу: виділення областей екрану; перевизначення клавіш клавіатури в залежності від контексту; використання маніпуляторів і сірих клавіш клавіатури для управління курсором. Власне WIMP-інтерфейс характеризується наступними особливостями: вся робота з програмами, файлами і документами відбувається у вікнах; всі об'єкти представляються у вигляді значків; Всі дії з об’єктами здійснюються за допомогою меню; здійснюється широке використання маніпуляторів для вказівки на об'єкти;

- Природно-мовний інтерфейс (або SILK-інтерфейс: Speech – мова, Image – образ, Language – мова, Knowlege – знання) – користувач «розмовляє» з програмою на рідній йому мовою. Цей вид інтерфейсу найбільш наближений до звичайної, людській формі спілкування. В рамках цього інтерфейсу йде звичайна «Розмова» людини і комп’ютера. При цьому комп'ютер знаходить для себе команди, аналізуючи людську мову і знаходячи в ній ключові фрази. Результат виконання команд він також перетворює в зрозумілу людині форму. Цей вид інтерфейсу найбільш вимогливий до апаратних ресурсів комп’ютера, і тому його застосування почалося для військових цілей. Зараз цей вид інтерфейсу широко освоюється, особливо для мобільних додатків. Він використовує такі технології:

Мовна технологія. При використанні цієї технології команди подаються голосом шляхом проголошення спеціальних зарезервованих слів-команд. Інтерфейс часто ототожнюється з діалогом, який подібний до діалогу або взаємодії між двома людьми. Мовні технології розпізнають, аналізують і синтезують голос людини. Імітація мови, сприйняття сенсу фраз, конвертація мови в текст, робота з голосом як з біометричною характеристикою - все це різні типи мовних технологій.

Біометрична технологія. Тут людина постає як сукупність ознак поведінки. Картинка зчитується з цифрової відеокамери, а потім за допомогою спеціальних програм розпізнавання образів з цього зображення виділяються команди.

Семантична технологія. Про цю технологію відомо вкрай мало. Схоже, що вона тісно пов'язана з штучним інтелектом і подібна зі всіма підтипами SILK і іншими типами теж. Можливо, що в зв'язку з важливим військовим значенням цих розробок ці напрямки були засекречені.

Візуально привабливий і зручний призначений для користувача інтерфейс - ключовий показник якості сайту. В поєднанні з грамотної структурою і логічною навігацією по розділах ресурсу, він приваблює відвідувачів і покращує функціональність сайту. Головне завдання в такій роботі, як проектування web - інтерфейсів - максимально спростити життя користувачеві, зробити так, щоб він досягав бажаного результату, витрачаючи мінімум зусиль.

Переваги хорошого інтерфейсу користувача:

1) підвищення конкурентоспроможності;

2) зниження вартості розробки;

3) збільшення аудиторії продукту;

4) зменшення витрат на навчання і підтримку користувачів;

5) зменшення втрат продуктивності працівників при впровадженні системи і більш швидке відновлення втраченої продуктивності;

6) доступність функціональності системи для максимальної кількості користувачів;

7) зниження ризику помилок.

Відмінні риси якісного інтерфейсу:

1. Стильова гнучкість - можливість використовувати різні інтерфейси з одним і тим же додатком, на практиці реалізується у вигляді набору «skins», для web-інтерфейсів – за допомогою таблиці стилів, в тому числі можливість у виборі користувачем власних установок інтерфейсу користувача (колір, ікони, підказки та ін.).

2. Спільне нарощування функціональності – можливість розвивати додаток без руйнування існуючого інтерфейсу.

3. Масштабованість - можливість легко налаштовувати і розширювати як інтерфейс, так і сам додаток при збільшенні числа користувачів, робочих місць, обсягу і характеристик даних.

4. Адаптивність до дій користувача – додаток має допускати можливість введення даних і команд різних способів (клавіатура, миша, інші пристрої) і багато варіативного доступу до прикладних функцій (ікони, «Гарячі клавіші», меню і т. д.). Крім цього програма повинна враховувати можливість переходу і повернення від вікна до вікна, від режиму до режиму, і правильно обробляти такі ситуації.

5. Незалежність в ресурсах – для створення призначеного для користувача інтерфейсу повинні надаватися окремі ресурси, спрямовані на зберігання і обробку даних, необхідних для обслуговування клієнтів (словники, контекстно залежні списки, історії запитів та ін.).

6. Кросплатформеність – при переході на іншу апаратну (програмну) платформу повинен здійснюватися перенесення призначеного для користувача інтерфейсу.

7. Мультимедійність – сукупність всіх видів інформації (графічної, звукової, відео).

3.2 Розробка бази даних системи

Програмування сайтів виконується з метою створення деякого корисного функціоналу. Робота з базами даних – це одна з найважливіших складових програмування сайтів динамічного типу. Чи то формування сторінок «на льоту», чи реагування на дії відвідувачів сайтів – взаємодія з базами даних потрібна завжди.

Бази даних для сайтів (БД) використовуються з метою зберігання різної інформації і, спрощено, представляють собою певний набір взаємозалежних таблиць. Розміри таблиць в БД різні, їх кількість довільна. Саме в базах даних зберігається на сервері необхідна для роботи сайту інформація, наприклад, інформація про клієнтів, каталог товарів, статистичні дані і т. д.

Програмування сайтів динамічного типу виконується за допомогою різних скриптів, поділюваних зазвичай на серверні і клієнтські. Програмування сайтів за допомогою серверних скриптів дозволяє обробляти дані, введені відвідувачами сайтів у веб-форми, генерувати динамічні сторінки, відсилати й приймати cookies. Для отримання інформації, необхідної при виконанні подібних дій, серверні скрипти звертаються до баз даних. Звернення скрипта до БД називається запитом.

Для побудови запитів до баз даних широко застосовується SQL (Structured Query Language) – “мова структурованих запитів”. За допомогою SQL може здійснюватися додавання, видалення, редагування записів в таблицях баз даних, вибірка даних у відповідності з різними умовами, сортування даних і багато іншого. У програмуванні сайтів управління БД здійснюється за допомогою клієнт-серверних систем управління базами даних (СУБД), таких як Oracle, MS SQL Server, PostgreSQL, MySQL та ін. Клієнт-серверні СУБД обробляють запити централізовано, до їх переваг відносять забезпечення високої надійності баз даних, високої доступності і високої безпеки. СУБД MySQL – вільна система управління базами даних, одна з найбільш часто вживаних в програмуванні сайтів. СУБД MySQL підтримує велику кількість існуючих типів таблиць (InnoDB, MyISAM і т. д.),

а завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць. Управління базами даних за допомогою MySQL дуже зручне, що зробило дану систему затребуваною і популярною. Система управління реляційними базами даних Microsoft SQL Server поставляється компанією Microsoft на комерційній основі (за винятком безкоштовної редакції Express Edition). Ця СУБД використовує мову запитів Transact-SQL, підтримується операційними системами сімейства Windows Desktop / Server. У СУБД Microsoft SQL Server присутнє графічне ПЗ для конструювання та оптимізації запитів (SQL Management Studio і Studio Express). Об'єктно-реляційна система управління базами даних компанії Oracle - Oracle Database – працює на Windows, Unix, Linux, MacOS. Oracle Database, на відміну від MySQL, наприклад, має більш широку сферу застосування. СУБД Oracle має високу продуктивність, широкий функціонал, унікальні технології (RAC, RAT і т. д.). У програмуванні сайтів для невеликих і середніх компаній застосовується досить рідко через свою високу вартість. До того ж, досить складно знайти хостинг з підтримкою даної СУБД.

Вільна система управління базами даних PostgreSQL існує в редакціях для Linux, Solaris/OpenSolaris, Win32, Win x86-64, Mac OS X, FreeBSD, QNX 4.25, QNX 6. Базується на мові SQL. Серед переваг PostgreSQL виділяють підтримку БД практично необмеженого розміру, наявність надійних механізмів реплікації, легку розширюваність, підтримку великого набору вбудованих типів даних та багато іншого.

Програмування сайтів, що взаємодіють різним чином з базами даних, включає кілька основних етапів роботи з БД**:** побудова запитів до БД за допомогою мови SQL, програмування сценаріїв для обробки цих запитів і програмування модулів для відображення результатів обробки запитів.

Надмірне число звернень від сайтів до баз даних робить завантаження сайтів більш повільним, збільшує навантаження на сервер. У результаті можливі перебої в роботі сайтів, аж до повного припинення доступу.

Зменшення кількості запитів до БД дозволяє зменшити навантаження на сервер, а також зменшити час завантаження динамічних сторінок з сервера. Тому оптимізація взаємодії сайтів з базами даних – це одне із завдань професійного програмування сайтів.

3.3 Результати реалізації

Розробляємо статичний сайт візитку для туристичного агентства під назвою «Travel Life» за допомогою HTML, CSS та JavaScript. Також подібний сайт можна розробити за допомогою CMS.

Спочатку робимо головну сторінку та меню сайту (Рис. 1, Рис. 2).

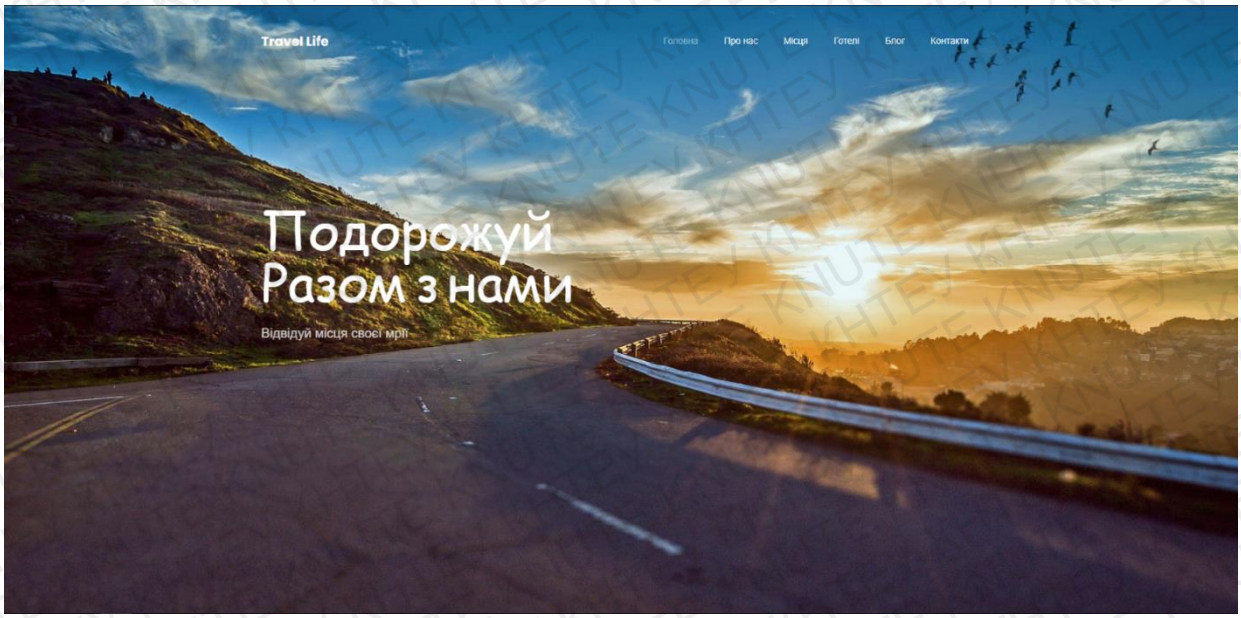

Рис. 1. Головна сторінка

Рис. 2. Меню сайту

Далі підписуємо основні принципи на яких базується компанія (Рис. 3)

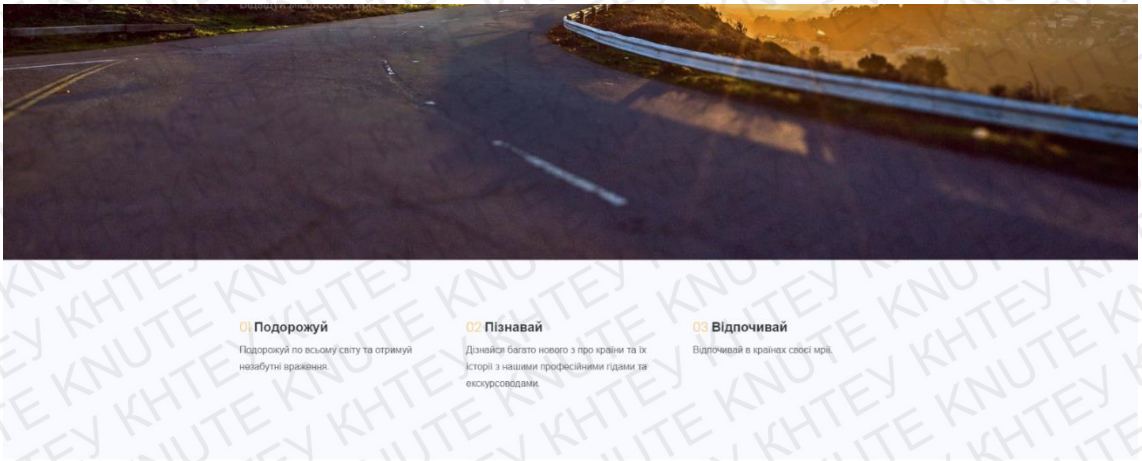

Рис. 3. Основні принципи

Починаємо заповнювати пункт меню «Про нас» (Рис. 4)

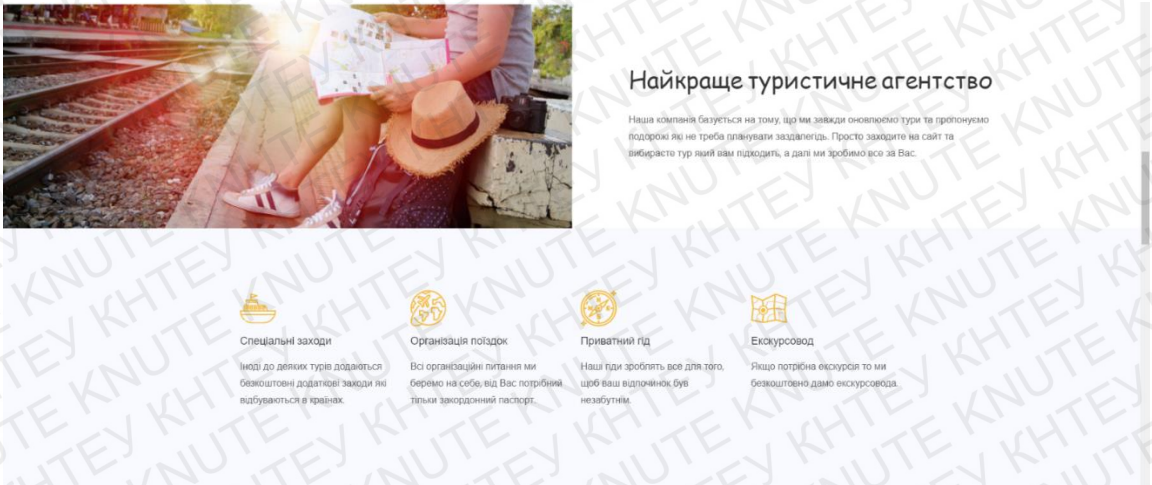

Рис. 4. пункт меню «Про нас»

Наступним кроком додаємо різні тури, їх короткий опис, ціну, кількість днів та осіб (рис. 5).

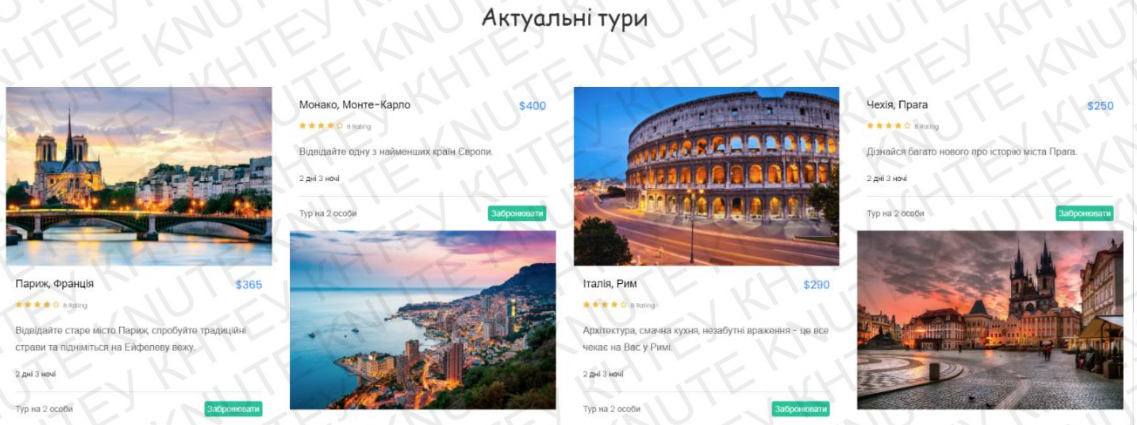

Рис. 5. Огляд турів

Далі заповнюємо пункт «Блог» де будуть актуальні новини та відгуки (рис. 6).

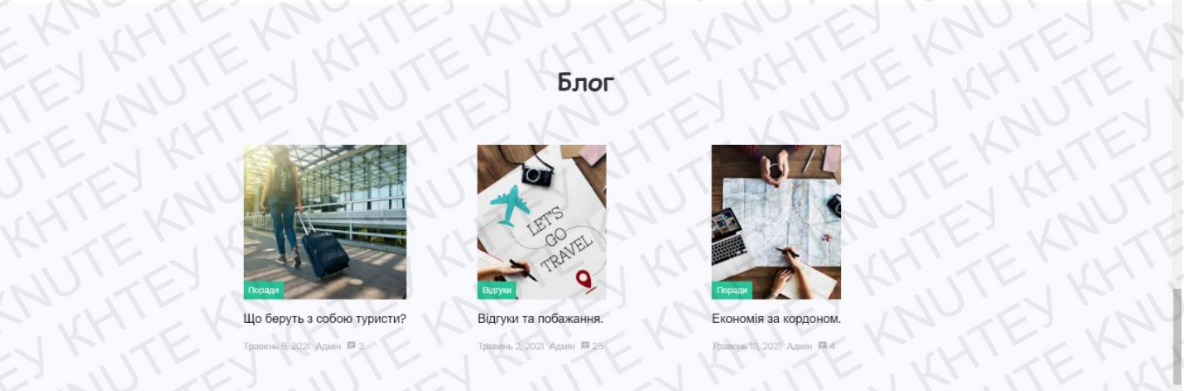

Рис. 6. Пункт «Блог»

І останнє робимо поле для розсилки актуальних новин на електронну пошту та пункт меню «контакти» для зворотного зв’язку (Рис.7).

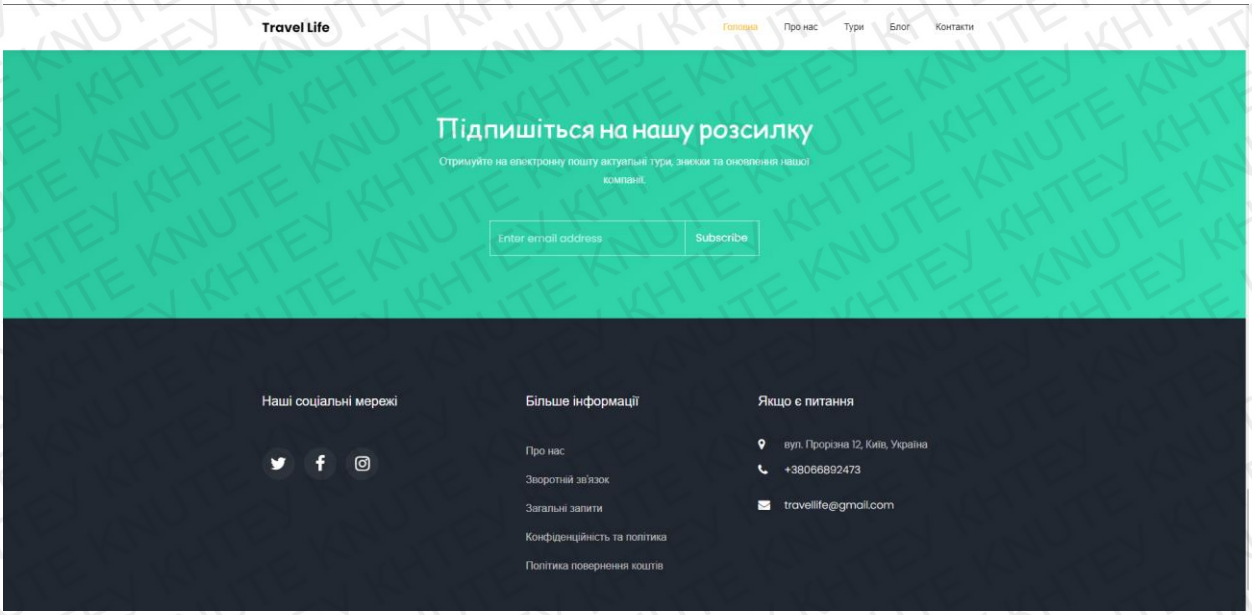

Рис. 7. Розсилка та контакти.

Висновки до третього розділу:

Взаємодія з веб-сайтом це основна мета користувача, якщо інтерфейс буде виконаний по всім сучасним стандартам то вірогідність того, що користувач знову повернеться на сайт досить велика.

ВИСНОВК

Якщо робити висновок, то можна зрозуміти що CMS системи – це дуже гарний варіант для людей яким потрібен веб-сайт, але не мають досвіду в сфері розробки. CMS пропонують різноманітні шаблони, плагіни, скрикти, які можуть бути як платні, так і безкоштовні. За допомогою CMS можна зробити майже будь які сайти, але зазвичай їх вигляди будуть упиратися в шаблони які доступні, і це на мій погляд не дуже добре.

Краще розробляти сайти з нуля. Таким чином не треба буде підлаштовуватися під певні обмеження CMS. Також інформація не буду проходити через сторонні ресурси. Зробити сайт з нуля це дійсно важче, проте ефективніше в перспективі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-сайт: визначення й застосування. URL: http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site

2. Статичні чи динамічні сайти: що обрати? URL: https://webstudio2u.net/ua/design-web/391-static-or-dynamic.html

3. Системи керування контентом. URL: https://www.victoria.lviv.ua/library/students/wd4/work10.html

4. Що таке CMS? Огляд кращих CMS. URL: https://www.ukraine.com.ua/blog/site-administration/chto-takoe-cms-obzor-luchshih-cms.html

5. Аналіз вимог до програмного забезпечення. Лекції 1-2. Визначення вимог до програмних систем. URL: http://baklaniv.at.ua/ANALIZ_VYMOG/lekcija_1-2.pdf

6.Практика розробки CMS. URL: https://www.victoria.lviv.ua/library/students/wp/lab2.html

7. Програмування сайтів. Бази даних для сайтів. URL: https://webstudio2u.net/ua/programming/494-site-programming.html

8. Проектування інтерфейсів користувача: посібник для студ. вузів. Брусенцова Т. П., Кішкурно Т.В.

ДОДАТОК

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <title>Дипломний проект</title>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

   <link href="https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,6
00,700" rel="stylesheet">
   <link href="https://fonts.googleapis.com/css?family=Abril+Fatface" rel="styles
heet">

   <link rel="stylesheet" href="css/open-iconic-bootstrap.min.css">
   <link rel="stylesheet" href="css/animate.css">

   <link rel="stylesheet" href="css/owl.carousel.min.css">
   <link rel="stylesheet" href="css/owl.theme.default.min.css">
   <link rel="stylesheet" href="css/magnific-popup.css">

   <link rel="stylesheet" href="css/aos.css">

   <link rel="stylesheet" href="css/ionicons.min.css">

   <link rel="stylesheet" href="css/bootstrap-datepicker.css">
   <link rel="stylesheet" href="css/jquery.timepicker.css">


   <link rel="stylesheet" href="css/flaticon.css">
   <link rel="stylesheet" href="css/icomoon.css">
   <link rel="stylesheet" href="css/style.css">
 </head>
 <body>

    <nav class="navbar navbar-expand-lg navbar-dark ftco_navbar bg-dark ftco-
navbar-light" id="ftco-navbar">
      <div class="container">
        <a class="navbar-brand" href="index.html">Travel Life</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#ftco-nav" aria-controls="ftco-nav" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="oi oi-menu"></span> Меню
```

```
        </button>

        <div class="collapse navbar-collapse" id="ftco-nav">
         <ul class="navbar-nav ml-auto">
           <li class="nav-item active"><a href="index.html" class="nav-
link">Головна</a></li>
           <li class="nav-item"><a href="about.html" class="nav-
link">Про нас</a></li>
           <li class="nav-item"><a href="places.html" class="nav-
link">Тури</a></li>
           <li class="nav-item"><a href="blog.html" class="nav-
link">Блог</a></li>
           <li class="nav-item"><a href="contact.html" class="nav-
link">Контакти</a></li>
         </ul>
        </div>
      </div>
    </nav>
  <!-- END nav -->

   <div class="hero-wrap js-fullheight" style="background-
image: url('images/bg_1.jpg');">
     <div class="overlay"></div>
     <div class="container">
       <div class="row no-gutters slider-text js-fullheight align-items-center justify-
content-start" data-scrollax-parent="true">
         <div class="col-md-9 ftco-animate mb-5 pb-5 text-center text-md-left" data-
scrollax=" properties: { translateY: '70%' }">
           <h1 class="mb-4" data-
scrollax="properties: { translateY: '30%', opacity: 1.6 }">Подорожуй <br>Разом
з нами</h1>
           <p data-
scrollax="properties: { translateY: '30%', opacity: 1.6 }">Відвідуй місця своєї мр
ії</p>
         </div>
       </div>
     </div>
   </div>



   <section class="ftco-section bg-light">
     <div class="container">
       <div class="row">
         <div class="col-md-4">
```

```
            <div class="intro ftco-animate">
              <h3><span>01</span> Подорожуй</h3>
              <p>Подорожуй по всьому світу та отримуй незабутні враження
.</p>
            </div>
          </div>
          <div class="col-md-4">
            <div class="intro ftco-animate">
              <h3><span>02</span> Пізнавай</h3>
              <p>Дізнайся багато нового з про країни та їх історії з нашими п
рофесійними гідами та екскурсоводами.</p>
            </div>
          </div>
          <div class="col-md-4">
            <div class="intro ftco-animate">
              <h3><span>03</span> Відпочивай</h3>
              <p>Відпочивай в країнах своєї мрії.</p>
            </div>
          </div>
        </div>
      </div>
    </section>

    <section class="ftco-section">
      <div class="container">
        <div class="row justify-content-center mb-5 pb-3">
       <div class="col-md-7 heading-section text-center ftco-animate">
        <h2 class="mb-4">Перегляньте наші останні ідеї відпусток</h2>
       </div>
      </div>
      <div class="row">
        <div class="col-md-4 ftco-animate">
          <a href="#" class="destination-entry img" style="background-
image: url(images/destination-1.jpg);">
            <div class="text text-center">
              <h3>Пляжні пейзажі</h3>
            </div>
          </a>
        </div>
        <div class="col-md-4 ftco-animate">
          <a href="#" class="destination-entry img" style="background-
image: url(images/destination-2-1.jpg);">
            <div class="text text-center">
              <h3>Груповий відпочинок</h3>
            </div>
```

```
            </a>
          </div>
          <div class="col-md-4 ftco-animate">
            <a href="#" class="destination-entry img" style="background-
image: url(images/destination-3.jpg);">
              <div class="text text-center">
                <h3>7 днів 5 країн</h3>
              </div>
            </a>
          </div>
        </div>
        </div>
    </section>

        <section class="ftco-about d-md-flex">
        <div class="one-half img" style="background-
image: url(images/about.jpg);"></div>
        <div class="one-half ftco-animate">
        <div class="heading-section ftco-animate ">
         <h2 class="mb-4">Найкраще туристичне агентство</h2>
        </div>
        <div>
              <p>Наша компанія базується на тому, що ми завжди оновлюємо тур
и та пропонуємо подорожі які не треба планувати заздалегідь. Просто заходи
те на сайт та вибираєте тур який вам підходить, а далі ми зробимо все за Вас.
</p>
          </div>
        </div>
    </section>

    <section class="ftco-section services-section bg-light">
      <div class="container">
        <div class="row d-flex">
          <div class="col-md-3 d-flex align-self-stretch ftco-animate">
            <div class="media block-6 services d-block">
              <div class="icon"><span class="flaticon-yatch"></span></div>
              <div class="media-body">
                <h3 class="heading mb-3">Спеціальні заходи</h3>
                <p>Іноді до деяких турів додаються безкоштовні додаткові заходи я
кі відбуваються в країнах.</p>
              </div>
            </div>
          </div>
          <div class="col-md-3 d-flex align-self-stretch ftco-animate">
            <div class="media block-6 services d-block">
```

```
            <div class="icon"><span class="flaticon-around"></span></div>
            <div class="media-body">
              <h3 class="heading mb-3">Організація поїздок</h3>
              <p>Всі організаційні питання ми беремо на себе, від Вас потрібний тільки закордонний паспорт.</p>
            </div>
          </div>
        </div>
        <div class="col-md-3 d-flex align-self-stretch ftco-animate">
          <div class="media block-6 services d-block">
            <div class="icon"><span class="flaticon-compass"></span></div>
            <div class="media-body">
              <h3 class="heading mb-3">Приватний гід</h3>
              <p>Наші гіди зроблять все для того, щоб ваш відпочинок був незабутнім.</p>
            </div>
          </div>
        </div>
        <div class="col-md-3 d-flex align-self-stretch ftco-animate">
          <div class="media block-6 services d-block">
            <div class="icon"><span class="flaticon-map-of-roads"></span></div>
            <div class="media-body">
              <h3 class="heading mb-3">Екскурсовод</h3>
              <p>Якщо потрібна екскурсія то ми безкоштовно дамо екскурсовода.</p>
            </div>
          </div>
        </div>
      </div>
    </section>

    <section class="ftco-section">
      <div class="container">
            <div class="row justify-content-center mb-5 pb-3">
        <div class="col-md-7 heading-section text-center ftco-animate">
          <h2 class="mb-4">Актуальні тури</h2>
        </div>
      </div>
      </div>
      <div class="container-fluid">
        <div class="row">
          <div class="col-sm col-md-6 col-lg ftco-animate">
            <div class="destination">
```

```
                <a href="#" class="img img-2 d-flex justify-content-center align-items-center" style="background-image: url(images/paris-cathedrale.jpg);">
                  <div class="icon d-flex justify-content-center align-items-center">
                    <span class="icon-link"></span>
                  </div>
                </a>
                <div class="text p-3">
                  <div class="d-flex">
                    <div class="one">
                      <h3><a href="#">Париж, Франція</a></h3>
                      <p class="rate">
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star-o"></i>
                        <span>8 Rating</span>
                      </p>
                    </div>
                    <div class="two">
                      <span class="price">$365</span>
                    </div>
                  </div>
                  <p>Відвідайте старе місто Париж, спробуйте традиційні стра ви та підніміться на Ейфелеву вежу. </p>
                  <p class="days"><span>2 дні 3 ночі</span></p>
                  <hr>
                  <p class="bottom-area d-flex">
                    <span></i>Тур на 2 особи</span>
                    <span class="ml-auto"><a href="#">Забронювати</a></span>
                  </p>
                </div>
              </div>
            </div>
            <div class="col-sm col-md-6 col-lg ftco-animate">
              <div class="destination d-md-flex flex-column-reverse">
                <a href="#" class="img img-2 d-flex justify-content-center align-items-center" style="background-image: url(images/Mo1.jpg);">
                  <div class="icon d-flex justify-content-center align-items-center">
                    <span class="icon-link"></span>
                  </div>
                </a>
```

```
                <div class="text p-3">
                  <div class="d-flex">
                    <div class="one">
                      <h3><a href="#">Монако, Монте-Карло</a></h3>
                      <p class="rate">
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star-o"></i>
                        <span>8 Rating</span>
                      </p>
                    </div>
                    <div class="two">
                      <span class="price">$400</span>
                    </div>
                  </div>
                  <p>Відвідайте одну з найменших країн Європи.</p>
                  <p class="days"><span>2 дні 3 ночі</span></p>
                  <hr>
                  <p class="bottom-area d-flex">
                    <span></i>Тур на 2 особи</span>
                    <span class="ml-
auto"><a href="#">Забронювати</a></span>
                  </p>
                </div>
              </div>
            </div>
            <div class="col-sm col-md-6 col-lg ftco-animate">
              <div class="destination">
                <a href="#" class="img img-2 d-flex justify-content-center align-
items-center" style="background-image: url(images/Colosseum-Rome.jpg);">
                  <div class="icon d-flex justify-content-center align-items-
center">
                    <span class="icon-link"></span>
                  </div>
                </a>
                <div class="text p-3">
                  <div class="d-flex">
                    <div class="one">
                      <h3><a href="#">Італія, Рим</a></h3>
                      <p class="rate">
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
```

```
                        <i class="icon-star"></i>
                        <i class="icon-star-o"></i>
                        <span>8 Rating</span>
                      </p>
                    </div>
                    <div class="two">
                      <span class="price">$290</span>
                    </div>
                  </div>
                  <p>Архітектура, смачна кухня, незабутні враження -
 це все чекає на Вас у Римі.</p>
                  <p class="days"><span>2 дні 3 ночі</span></p>
                  <hr>
                  <p class="bottom-area d-flex">
                    <span></i>Тур на 2 особи</span>
                    <span class="ml-
auto"><a href="#">Забронювати</a></span>
                  </p>
                </div>
              </div>
            </div>
            <div class="col-sm col-md-6 col-lg ftco-animate">
              <div class="destination d-md-flex flex-column-reverse">
                <a href="#" class="img img-2 d-flex justify-content-center align-
items-center" style="background-image: url(images/Prague.jpg);">
                  <div class="icon d-flex justify-content-center align-items-
center">
                    <span class="icon-link"></span>
                  </div>
                </a>
                <div class="text p-3">
                  <div class="d-flex">
                    <div class="one">
                      <h3><a href="#">Чехія, Прага</a></h3>
                      <p class="rate">
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star"></i>
                        <i class="icon-star-o"></i>
                        <span>8 Rating</span>
                      </p>
                    </div>
                    <div class="two">
                      <span class="price">$250</span>
```

```
                    </div>
                  </div>
                  <p>Дізнайся багато нового про історію міста Прага.</p>
                  <p class="days"><span>2 дні 3 ночі</span></p>
                  <hr>
                  <p class="bottom-area d-flex">
                    <span></i>Тур на 2 особи</span>
                    <span class="ml-
auto"><a href="#">Забронювати</a></span>
                  </p>
                </div>
              </div>
            </div>
          </div>
        </section>


        <section class="ftco-section bg-light">
          <div class="container">
            <div class="row justify-content-center mb-5 pb-3">
              <div class="col-md-7 heading-section text-center ftco-animate">
                <h2><strong>Блог</strong>
              </div>
            </div>
            <div class="row d-flex">
              <div class="col-md-4 d-flex ftco-animate">
                <div class="blog-entry align-self-stretch">
                  <a href="blog-single.html" class="block-20" style="background-
image: url('images/image_1.jpg');">
                  </a>
                  <div class="text">
                    <span class="tag">Поради</span>
                    <h3 class="heading mt-
3"><a href="#">Що беруть з собою туристи?</a></h3>
                    <div class="meta mb-3">
                      <div><a href="#">Травень 5, 2021</a></div>
                      <div><a href="#">Адмін</a></div>
                      <div><a href="#" class="meta-chat"><span class="icon-
chat"></span> 3</a></div>
                    </div>
                  </div>
                </div>
              </div>
              <div class="col-md-4 d-flex ftco-animate">
```

```
          <div class="blog-entry align-self-stretch">
           <a href="blog-single.html" class="block-20" style="background-
image: url('images/image_2.jpg');">
           </a>
           <div class="text">
            <span class="tag">Відгуки</span>
            <h3 class="heading mt-
3"><a href="#">Відгуки та побажання.</a></h3>
            <div class="meta mb-3">
             <div><a href="#">Травень 2, 2021</a></div>
             <div><a href="#">Адмін</a></div>
             <div><a href="#" class="meta-chat"><span class="icon-
chat"></span> 25</a></div>
            </div>
           </div>
          </div>
          <div class="col-md-4 d-flex ftco-animate">
          <div class="blog-entry align-self-stretch">
           <a href="blog-single.html" class="block-20" style="background-
image: url('images/image_3.jpg');">
           </a>
           <div class="text">
            <span class="tag">Поради</span>
            <h3 class="heading mt-
3"><a href="#">Економія за кордоном.</a></h3>
            <div class="meta mb-3">
             <div><a href="#">Травень 10, 2021</a></div>
             <div><a href="#">Адмін</a></div>
             <div><a href="#" class="meta-chat"><span class="icon-
chat"></span> 4</a></div>
            </div>
           </div>
          </div>
         </div>
        </div>
       </section>

        <section class="ftco-section-parallax">
       <div class="parallax-img d-flex align-items-center">
        <div class="container">
         <div class="row d-flex justify-content-center">
          <div class="col-md-7 text-center heading-section heading-section-
white ftco-animate">
```

```
          <h2>Підпишіться на нашу розсилку</h2>
          <p>Отримуйте на електронну пошту актуальні тури, знижки та оновл
ення нашої компанії.</p>
          <div class="row d-flex justify-content-center mt-5">
            <div class="col-md-8">
              <form action="#" class="subscribe-form">
                <div class="form-group d-flex">
                  <input type="text" class="form-
control" placeholder="Enter email address">
                  <input type="submit" value="Subscribe" class="submit px-3">
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </section>

    <footer class="ftco-footer ftco-bg-dark ftco-section">
      <div class="container">
        <div class="row mb-5">
          <div class="col-md">
            <div class="ftco-footer-widget mb-4">
              <h2 class="ftco-heading-2">Наші соціальні мережі</h2>

              <ul class="ftco-footer-social list-unstyled float-md-left float-lft mt-3">
                <li class="ftco-animate"><a href="#"><span class="icon-
twitter"></span></a></li>
                <li class="ftco-animate"><a href="#"><span class="icon-
facebook"></span></a></li>
                <li class="ftco-animate"><a href="#"><span class="icon-
instagram"></span></a></li>
              </ul>
            </div>
          </div>
          <div class="col-md">
            <div class="ftco-footer-widget mb-4 ml-md-4">
              <h2 class="ftco-heading-2">Більше інформації</h2>
              <ul class="list-unstyled">
                <li><a href="#" class="py-2 d-block">Про нас</a></li>
                <li><a href="#" class="py-2 d-block">Зворотній зв'язок</a></li>
                <li><a href="#" class="py-2 d-block">Загальні запити</a></li>
```

```
            <li><a href="#" class="py-2 d-
block">Конфіденційність та політика</a></li>
            <li><a href="#" class="py-2 d-
block">Політика повернення коштів</a></li>
          </ul>
        </div>
      </div>

      <div class="col-md">
       <div class="ftco-footer-widget mb-4">
          <h2 class="ftco-heading-2">Якщо є питання</h2>
          <div class="block-23 mb-3">
           <ul>
            <li><span class="icon icon-map-
marker"></span><span class="text">вул. Прорізна 12, Київ, Україна</span></li
>
            <li><a href="#"><span class="icon icon-
phone"></span><span class="text">+38066892473</span></a></li>
            <li><a href="#"><span class="icon icon-
envelope"></span><span class="text">travellife@gmail.com</span></a></li>
           </ul>
          </div>
        </div>
       </div>
      </div>
      <div class="row">
       <div class="col-md-12 text-center">




  <!-- loader -->
  <div id="ftco-
loader" class="show fullscreen"><svg class="circular" width="48px" height="48p
x"><circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-
width="4" stroke="#eeeeee"/><circle class="path" cx="24" cy="24" r="22" fill="n
one" stroke-width="4" stroke-miterlimit="10" stroke="#F96D00"/></svg></div>


  <script src="js/jquery.min.js"></script>
  <script src="js/jquery-migrate-3.0.1.min.js"></script>
  <script src="js/popper.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
  <script src="js/jquery.easing.1.3.js"></script>
  <script src="js/jquery.waypoints.min.js"></script>
```

```
  <script src="js/jquery.stellar.min.js"></script>
  <script src="js/owl.carousel.min.js"></script>
  <script src="js/jquery.magnific-popup.min.js"></script>
  <script src="js/aos.js"></script>
  <script src="js/jquery.animateNumber.min.js"></script>
  <script src="js/bootstrap-datepicker.js"></script>
  <script src="js/jquery.timepicker.min.js"></script>
  <script src="js/scrollax.min.js"></script>
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBVWaKrjvy3
MaE7SQ74_uJiULgl1JY0H2s&sensor=false"></script>
  <script src="js/google-map.js"></script>
  <script src="js/main.js"></script>

  </body>
</html>
```