Полтавський університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

Кафедра комп’ютерних та інформаційних технологій і систем

(повна назва кафедри)

# Пояснювальна записка

**до дипломного проекту (роботи)**

бакалавра

(освітньо-кваліфікаційний рівень)

на тему

Технології front-end розробки для створення адаптивного дизайну, JS

та бібліотека jQuery

Виконав: студент 4 курсу, групи 402-ТК

спеціальності

\_\_123 Комп’ютерна інженерія \_

(шифр і назва напряму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава 2021

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**спеціальність 123 «Комп’ютерна інженерія»**

**на тему**

**«Алгоритмізація процесу розподілу навчального навантаження викладачів кафедри»**

**Студента групи 402-ТК Мируна Віктора Олександровича**

Керівник роботи

кандидат технічних наук,

доцент Мавріна М.О

Завідувач кафедри

кандидат технічних наук,

доцент Дмитренко М.І

Полтава – 2021

**Реферат**

Дипломна робота містить 63 сторінку, 21 малюнок, список використаної літератури з 17 найменувань та додаток.

В роботі розглянуті основні поняття веб-технологій, веб-розробки, вимоги для створення веб-сайту та їх сучасні тенденції.

*Об’єктом дослідження* є процес створення frontend-частини інтерактивного веб-сайту засобами веб-розробки HTML, CSS, JS та jQuery.

*Предметом дослідження* засоби розробки frontend-частини веб-сайтів HTML, CSS, JS та jQuery.

*Метою дослідження* є розробка веб-сайту з розважальним контентом.

Відповідно до поставленої мети в роботі визначені наступні завдання:

* вивчити технології розробки сайтів;
* визначити цілі, ідеї, потреби сайту;
* розробити технічну концепцію сайту (структуру);
* розробити програмний продукт.

*Ключові слова: HTML, CSS, JS, jQuery*, *frontend.*

**ABSTRACT**

Thesis contains 63 pages, 21 figures, a list of references from 16 titles and an appendix.

The paper considers the basic concepts of web technologies, web development, requirements for creating a website and their current trends.

The object of research is the process of creating a frontend part of an interactive website by means of web development HTML, CSS, JS and jQuery.

The subject of the research is the means of developing the frontend part of the HTML, CSS, JS and jQuery websites.

The aim of the study is to develop a website with entertaining content.

In accordance with the goal in the work identified the following tasks:

* study site development technologies;
* identify the goals, ideas, needs of the site;
* develop a technical concept of the site (structure);
* develop a software product.

Keywords: HTML, CSS, JS, jQuery, frontend.

**ЗМІСТ**

ПЕРЕЛІК УМОВНИХПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ ……………6

ВСТУП 7

РОЗДІЛ 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА WEB-САЙТІВ 8

1.1 Класифікація сайтів……………………………………………………...8

1.2 Структура сайтів………………………………………………………..15

1.3 Основні етапи розробки сайту…………………………………………16

1.3 Основні етапи розробки сайту…………………………………………16

1.3.1 Цілі і завдання web-сайту 16

1.3.2 Вивчення цільової аудиторії 16

1.3.3 Створення технічного завдання 17

1.3.4 Прототипування 17

1.3.5 Створення макетів дизайну 18

1.3.6 Верстка 18

1.3.7 Програмування 18

1.3.8 Базове наповнення 19

1.3.9 Тестування 19

1.3.10 Здача готового проекту клієнту 19

1.4 Висновок до розділу 1………………………………………………….20

РОЗДІЛ 2 ЗАСОБИ РОЗРОБКИ КЛІЄНТСЬКОЇ ЧАСТИНИ WEB-САЙТІВ.21

2.1 Мова розмітки гіпертексту HTML……………………………............. 21

2.2 Каскадні таблиці стилів CSS…………………………………………...22

2.2.1 Адаптивний дизайн 24

2.3 Об’єктно-орієнтована мова програмування JS………………………26

2.3.1 Бібліотека jQuery 35

2.4 Сучасні тренди та тенденції веб-дизайну……………………………38

2.5 Висновки до розділу 2…………………………………………………40

РОЗДІЛ 3 ПРОЕКТНА ЧАСТИНА 41

3.1 Загальна характеристика сайту…………………………………………41

3.2 Створення дизайну……………………………..………………………41

3.3 Висновок до розділу 3……………………………..……………………49

ВИСНОВОК 50

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 51

ДОДАТОК А ВИХІДНІ КОДИ ПРОГРАМНОГО ЗАСОБУ 53

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML – мова розмітки гіпертексту

СSS – каскадні таблиці стилів

JS – мова програмування

URL – стандартизована адреса певного ресурсу

ТЗ – технічне завдання

СTA – заклик до дії (Call to action)

PHP – серверна мова програмування

APS.NET – платформа для розробки веб-додатків

ПК – персональний комп’ютер

BOM – об’єктна модель браузера

DOM – об’єктна модель документа

ESMAScript – стандарт мови програмування JavaScript

W3C (World Wide Web Consortium) – консорціум Всесвітнього павутиння

XML – розширена мова розмітки

# ВСТУП

З появою веб-технологій комп'ютер починають використовувати абсолютно нові верстви населення Землі. Можна виділити дві найбільш характерні групи, що знаходяться на різних соціальних полюсах, які були стрімко залучені в нову технологію, можливо, навіть, без їх власного бажання. З одного боку, це представники елітарних груп суспільства − керівники крупних організацій, президенти банків, топ-менеджери, впливові державні чиновники і т.п. З іншого боку, це представники найширших верств населення − домогосподарки, пенсіонери, діти.

Спектр соціальних груп, що підключаються до мережі Інтернет і що шукають інформацію в WWW, весь час розширюється за рахунок користувачів, що не відносяться до категорії фахівців в області інформаційних технологій. Це лікарі, будівельники, історики, юристи, фінансисти, спортсмени, мандрівники, священослужителі, артисти, письменники, художники. Список можна продовжувати безкінечно. Будь-який, хто відчув корисність і незамінність мережі для своєї професійної діяльності або захоплень, приєднується до величезної армії споживачів інформації в «Всесвітній Павутині».

Веб-технологія повністю перевернула уявлення про роботу з інформацією, та і з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки − продуктивність, пропускна спроможність, ємкість пристроїв, що запам'ятовують, не враховували головного «вузького місця» системи − інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І лише коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, послідував безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

Отже, вивчення технологій, що дозволяють створювати нові веб-ресурси для широкого кола користувачів є на сьогоднішній день важливим та актуальним завданням.

# РОЗДІЛ 1

# ЗАГАЛЬНА ХАРАКТЕРИСТИКА WEB-САЙТІВ

Веб сайт – це сукупність програмних, інформаційних, а також медійних засобів, логічно-пов'язаних між собою [1].

# 1.1 Класифікація сайтів

Перш ніж класифікувати сайти, потрібно сказати, що не існує чіткої, затвердженої класифікації web-сайтів, їх можна поділити за наступними параметрами:

* технології, що застосовувалися для розробки;
* доступність сервісів;
* фізичне положення;
* мета розробки.

*За технологіями,* що використовувалися для розробки сайти поділяються на:

- статичні;

- динамічні

- flash-сайти;

- змішані.

**Статичні сайти** та web-сторінки – це інтернет-ресурси, структура яких розроблена мовою розмітки HTML. Статичними вони називаються оскільки, набравши певну адресу сторінки (url-адресу) відповідного сайту або одної з його сторінок, ви завжди будете бачити однаковий зміст.

Перевагами статичних web-сайтів є простота їх створення і низька вибагливість до апаратних ресурсів. Статичні сайти не вимагають підтримки скриптів та баз даних. Звичайно статичні сайти не вимогливі до можливостей хостингу і вони майже не будуть створювати навантаження на сервер. Основним недоліком цих сайтів є відсутність інтерактивності. Крім того, в таких сайтах складніше змінити інформацію, зовнішній вигляд та дизайн сторінок.

**Динамічні сайти** не існують незмінними на сервері, але генеруються за допомогою сценаріїв. Зміст і вигляд динамічної сторінки з однією URL-адресою можуть бути абсолютно різними залежно від різних умов, наприклад, залежно від часу доступу, прав доступу користувача, введеного запиту користувача тощо. Скрипти, що використовуються для створення динамічних веб-сторінок можуть бути написані різними мовами.

Недоліками динамічних сайтів є те, що вони потребують допомоги з технологіями управління, роблять підвищене навантаження на сервер, їх складніше оптимізувати, а використання сценаріїв на сайті створює потенційну загрозу безпеці.

Якщо сайт складається з великої кількості сторінок або потребує оновлення з часом, перевага динамічності очевидна. Розробникам веб-сайтів не потрібно повністю відтворювати всю сторінку, якщо вони змінюють інформаційний зміст або дизайн. Сторінки не зберігаються повністю, а створюються "на льоту" при доступі.

Тому розділення дизайну та змісту є головною відмінністю динамічних сайтів від статичних.

**Flash-сайти.** Технологія флеш ( з англ. *flash* спалах, блискавка) дозволяє створювати привабливі, інтерактивні сайти з анімацію. Технологія Flash, що була розроблена компанією Macromedia почала отримувати широке поширення в кінці дев'яностих років. З того часу і по сьогоднішній день вона є самим гнучким інструментом для створення анімації в інтернеті.

Одним із основних переваг технології завжди була плавна анімація. На відміну від готових анімацій, в яких кожен кадр являє собою окрему картинку, флеш використовує об’єктний підхід.

Одна з потужніших особливостей флеша – вбудована мова програмування ActionScript, що надає флеш-елементам високу інтерактивність. Вона дозволяє користувачам взаємодіяти з сайтом.

Але й ця технологія не обійшлася без недоліків. Головним з них є обов’язкова наявність установленої програми Flash Player на комп’ютері. Ця програма інтегрується в браузер і дозволяє працювати флеш-елементам сайту.

В основі **змішаних** сайтів лежить поєднання декількох технологій виробництва веб-сторінок. Найчастіше зустрічаються поєднання динамічних і статичних сторінок на сайтах.

*За доступністю сервісів* їх поділяють на:

 - відкриті;

 - напіввідкриті;

 - закриті.

**Відкриті** – це сайти в яких сервіси цілком доступні для всіх відвідувачів. **Напіввідкритими**називають сайти, які потребують реєстрації користувача для доступу. **Закриті**ж веб-портали є повністю приватними, що не дозволяють зайти звичайним користувачам. Такі сайти доступні для вузького кола людей. Зазвичай доступ здійснюється через запрошення. Прикладами закритих сайтів можуть бути корпоративні сайти підприємств або портали приватних осіб в локальній мережі провайдера, тощо.

*За фізичним положенням:*

- зовнішні;

- внутрішні.

**Зовнішні** *–* доступні користувачам з Інтернету, а **внутрішні**лише користувачам локальної мережі. Прикладами внутрішнього сайту можуть бути корпоративний сайт підприємства або сайт приватної особи в локальній мережі провайдера.

*За метою розробки* виділяють:

**Інформаційні** *–* це сайти, на яких зібрані різного роду статті на будь-які теми. Сайт може містити безліч інформаційних статей на всі тематики та розповідати про новинки в світі технологій. Також вони можуть об'єднувати та структурувати в собі інформацію, зібрану з інших ресурсів. До інформаційних сайтів належать: тематичні та новинні ресурси, блоги.

Найчастіше інформаційні портали існують за рахунок реклами у вигляді банерів або спонсорів (рис.1.1).

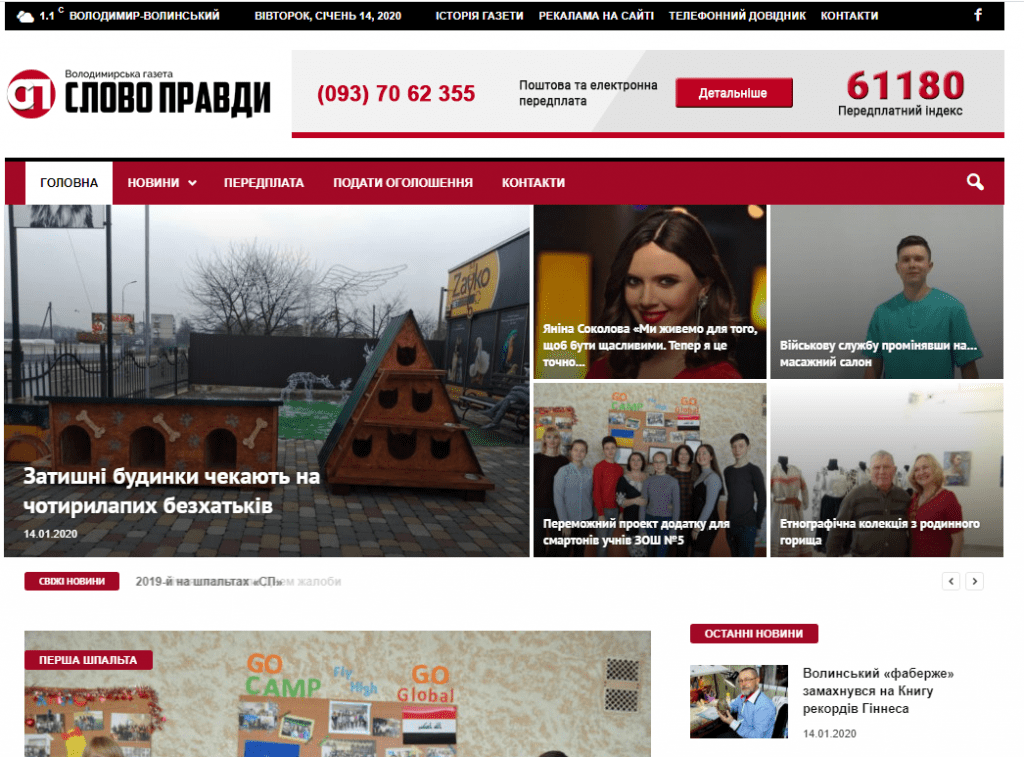


Рисунок 1.1 - Інформаційний сайт новин

**Односторінковий сайт (landing page)** *–* новий формат сайту, основною метою якого є спонукання користувача зробити цільову дію. Це може бути оформлення заявки на: покупку товарів, безкоштовну консультацію, оформлення “броні”, завантаження прайсу, розрахунку вартості послуг, участі в акції і т.п.

Приблизна структура односторінкового сайту продемонстрована на рис 1.2.

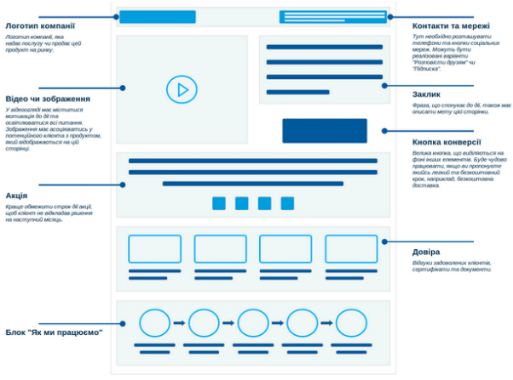


Рисунок 1.2 - Макет односторінкового сайту

**Сайт-візитка** *–* це невеликий інтернет-ресурс, обсяг якого зазвичай не перевищує 10-15 веб-сторінок, що містить коротку інформацію про компанію, включаючи контактні дані, а також перелік і опис послуг.

Головна функція – це додаткова та більш зручна передача інформації про продукти за допомогою реклами на сайті.(рис. 1.3)



Рисунок 1.3 - Приклад сайту-візитки

**Корпоративний сайт** – важливий інструмент успішного бізнесу будь-якої компанії (рис.1.4). В першу чергу, сайти подібного характеру допомагають формувати позитивний імідж компанії, залучають нових замовників, які можуть не знати про існування фірми на ринку, і відповідно, розширюють ринок збуту компанії.

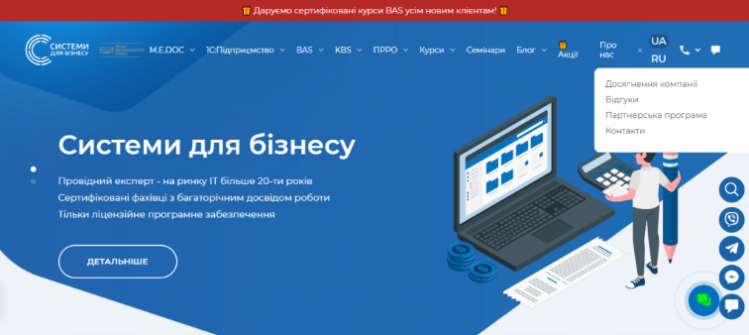


Рисунок 1.4 - Приклад корпоративного сайту

**Промо-сайт** – потужний інструмент для рекламування. Добре підходить для організації реклами товарів, послуг і т.д. Структура цих сайтів подібна до рекламного буклету і має привертати увагу потенційних покупців. Зазвичай обсяг не перевищує 10 сторінок. Зазвичай на сторінках розміщуються дані про: фірму, товари/послуги, сферу діяльності, контакти, акції, новини тощо. (рис. 1.5).

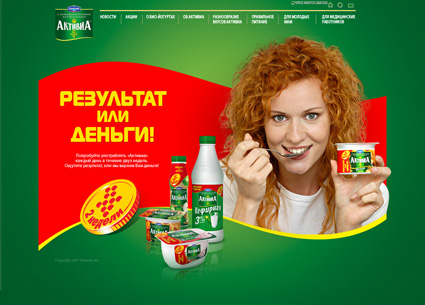


Рисунок 1.5 - Приклад промо-сайту

**Інтернет-магазин** – один із різновидів сайтів, на якому розташовують список продукції, яку може замовити користувач онлайн. Якщо в звичайному магазині потрібна наявність фізичного обладнання та приміщення, торгового залу, вітрин, цінників, продавців, консультантів, то в інтернет-магазині уся інфраструктура реалізована програмними засобами.

Обов’язковими рисами, якими мають володіти такі сайти – це зручність та функціональність. Інтернет-магазини повинні забезпечувати легкий пошук продуктів, відправку обраних товарів до кошику, блискавичне здійснення покупки, вибір способу та варіантів доставки (рис.1.6).

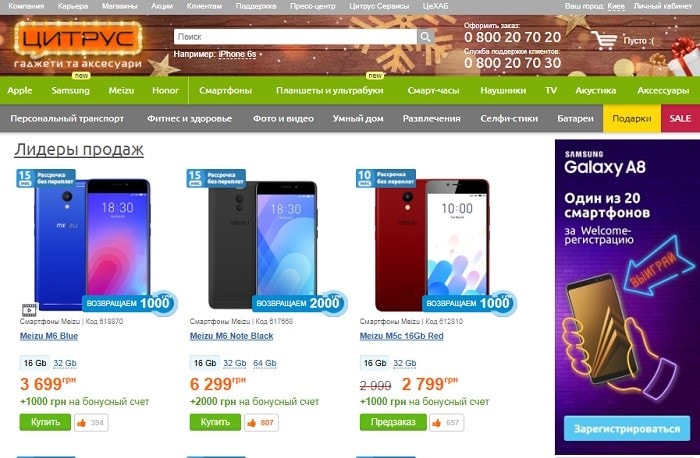


Рисунок 1.6 - Інтернет-магазин

**Іміджевий сайт** - веб-сайт, на якому представлена ємко, коротко і доступно інформація про бренд: основна ідея і мета, концепція, історія.

Такі сторінки створюють організації, які прагнуть поліпшити пізнаваність свого бренду і успішно просунути товар на ринку. Вони думають про створення гарного іміджу компанії, намагаються зміцнити його різними способами. Іміджевий сайт - найбільш дієвий з них. (рис.1.7).

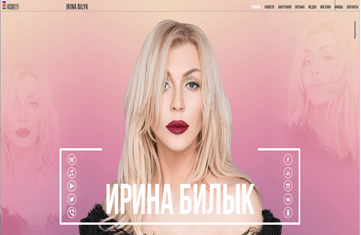


Рисунок 1.7 - Іміджевий сайт співачки

**Сайт-форум** *–* може існувати окремо, або як розділ іншого ресурсу. Зазвичай створюється на піддомені основного сайту і є інтерактивним ресурсом для спілкування людей. Форум – це сайти з самогенерованим контентом, тобто користувачі самі створюють контент. Основну частину головної сторінки займають розділи та теми обговорень. Мінімалістичний профіль користувача. Також значною відмінністю є наявність адміністраторів та модераторів, які можуть редагувати/видаляти коментарі користувачів, які порушують правила форуму. Інколи форум може слугувати хорошим доповненням до будь-якого сайту (рис.1.8) [2].



Рисунок 1.8 - Сайт-форум

# 1.2 Структура сайтів

На всіх сайтах завжди спочатку відкривається головна сторінка. У верхній частині головної сторінки зазвичай розташована **шапка** (з англ.“head”), яку повторюють і на інших сторінках сайту. Роблять це навмисно, адже ця частина появляється у вікні браузера першою і користувач сайту насамперед звертає увагу на неї. Для забезпечення швидкого переміщення до основних тематичних розділів сайту створюють **меню** сайту – набір гіперпосилань на його розділи. Горизонтальне меню зазвичай розташовують у шапці, а вертикальне – переважно в лівій частині сторінки в тому місці, де відвідувач розпочинає її перегляд.

У нижній частині головної сторінки міститься **підвал сайту** (footer з англ.) , що є також основним компонентом сторінки, але менш функціональним. В ньому зазвичай розташовують контактні дані та повідомлення про авторські права.

Меню повинно бути помітним і зрозумілим, оскільки користувач не зможе перейти до потрібного розділу і з великою вірогідністю покине сайт. Пункти меню мають бути чітко виділеними і не зливатися один з одним. Гіперпосилання, розміщені в тексті або у вигляді графічних об'єктів, допомагають переходити на різні сторінки сайту, інколи на інформаційні ресурси [3].

# 1.3 Основні етапи розробки сайту

# 1.3.1 Цілі і завдання web-сайту. Створення сайту – досить масштабний проект. Навіть якщо ресурс немає великих розмірів він може зіграти ключову роль в розвитку бізнесу, тому навіть односторінкові сайти потребують великого обсягу робіти і ретельне опрацювання всіх деталей.

## **1.3.2 Вивчення цільової аудиторії.** Перед тим, як з’ясувати, яку форму прийме сайт, необхідно дати відповідь на наступні питання:

* для якої категорії користувачів;
* які саме переваги мають потенційні клієнти;
* яка інформація буде для них цікавою;
* в якому виді вони полюбляють контент;
* які комунікаційні канали вони використовують;
* чи мають можливість підтримувати зв’язок за допомогою електронної пошти.

Це не повний список запитань, на які потрібно дати відповідь до початку розробки: його треба розширити в залежності від обраної тематики для майбутнього ресурсу.

## **1.3.3 Створення технічного завдання (ТЗ).** У ТЗ прописуються основні вимоги до проекту. На його основі можна зробити величезну кількість сайтів з різними візуальними параметрами.

Основні вимоги до технічного завдання:

- деталізація. Все, що повинен зробити фахівець, необхідно детально розписати в ТЗ;

- чіткість і точність. Будь-яким неоднозначним формулюванням або суб’єктивізму не місце в технічному завданні;

- написане зрозумілою для спеціаліста мовою. Це означає, що фахівець повинен самостійно зрозуміти зміст ТЗ і не перепитувати, що саме від нього хочуть.

**1.3.4 Прототипування.** Для більшого чіткого представлення та взаєморозуміння між замовником та фахівцем, щоб замовник побачив розташування елементів і навігацію створюється прототип дизайну сайту.

Навігація вимагає особливої уваги. Занадто багато сайтів створюються без урахування зручності навігації. Коли клієнт потрапляє на сайт він напочатку не знає, як йому отримати доступ до необхідної інформації. Ось чому, наприклад, настільки багато людей полюблять продукти Apple. Вони інтуїтивно зрозумілі. При першому ж контакті з продуктом Apple користувач розуміє, як користуватися ним.

Звісно ж однієї наявності зручної навігації не призведе до здійснення відвідувачем покупки, якщо на сайті відсутній або не працює чи малопомітний заклик до дії (з англ. Call to Action, CTA). Це обов’язковий атрибут кожного веб-ресурсу, який має мету продажу. Якщо ви не спровокуєте відвідувачів сайту вчинити певну дію, вони можуть і не здогадатися виконати її. І можуть зайти на сайт конкурентів, де буде присутній саме цей інструмент заохочення і користувач зрозуміє, що необхідно робити на сайті. Без заклику до дії ви не можете розраховувати на досягнення цілей.

## **1.3.5 Створення макетів дизайну.** Етап створення дизайну для сайту відповідає за візуальну складову веб-ресурсу, а саме:

* кольори;
* елементи;
* деталі;
* особливості (фішки сайту).

Слід відмітити, що макет сайту повинен залишитись незмінним хоча б декілька років. Пізніше, коли ресурс отримає довіру пошукових роботів, відвідувачів і клієнтів, тоді вже можна змінити його зовнішній вигляд. В разі невдалого дизайну на етапі розробки є великий ризик зупинити розробку власного інтернет-сайту.

Створення дизайну сайту – важливий етап web-розробки, на якому здійснюється візуальне оформлення засобів взаємодії користувача з сайтом.

Варто розуміти, що дизайнери, насамперед, забезпечують зручність інтерфейсу, а не тільки «красиву картинку», як багато хто вважає. Це значить, що всі елементи на сторінці повинні бути розміщені так, щоб відвідувач міг без проблем знайти і скористатися ними. За допомогою розташування інформаційних блоків дизайнер управляє увагою відвідувача, демонструє йому те, що він повинен або хоче побачити в даний момент (це особливо актуально для лендінгів).

**1.3.6 Верстка.** Після того, як дизайнери підготували макет для сайту, він передається верстальникам. Але початок цього етапу настає після утвердження дизайн-макетів всіх сторінок сайту.

За допомогою грамотного “чистого” коду верстальник гарантує, що сайт буде виглядати и працювати коректно з любих розширень екрану, гаджетів та при використанні різних інтернет-браузерів.

**1.3.7 Програмування.** Етап програмування передбачає роботу з внутрішнім програмним наповнюванням сайту, включаючи систему управління контентом, базу даних, механізми завантаження, вигрузки, підрахування будь-якої інформації, відправки повідомлень, форм, заявок та інші дії, які “невидимі” для користувачів, але необхідні для функціонування сайту.

**1.3.8 Базове наповнення.** Сайт без інформаційної наповненості, буде виглядати як полиця без книжок. Цей етап передбачає переклад контенту в зручний формат з подальшим розміщенням на сторінках ресурсу. Матеріали готуються завчасно. Вони можуть бути написаними як і копірайтерами, за заздалегідь складеним технічним завданням так і замовником зі своїм баченням.

**1.3.9 Тестування.** Хоч після кожного етапу розробки ресурс перевіряють на предмет «працює – не працює», певним елементам без фінального тестування не обійтися. Недоробки протягом етапу тестування находяться самі різні. Головне, позбавитися від них всіх до здачі проекту замовнику. Зазвичай, для тестування і демонстрації сайт розміщують на спеціальних платформах. Потім, якщо не виявлено помилок, проект з усіма файлами переміщують на хостинг.

**1.3.10 Здача готового проекту клієнту.** Завершальний етап – навчання замовника роботі із сайтом. Важлива риса якісного веб-ресурсу – зручна адміністративна панель. Це значить, що сайт мусить підтримувати максимально гнучке налаштування без допрацювань на програмному рівні.

При передачі завершеного проекту фахівці навчають керувати адміністративною панеллю, а також сервісами збору статистики. Після цього власник може власноруч добавляти або публікувати новий контент, створювати сторінки, розміщувати товари, збирати дані для аналітики тощо.

**1.4 Висновок до розділу 1**

Іноді складно визначити, до якої категорії належить той чи інший сайт. Проте така класифікація допомагає зорієнтуватися у величезній кількості ресурсів Інтернету. Найпоширеніші веб-ресурси можна класифікувати не тільки за метою і напрямком їх розробки і використаними технологіями, а й за їх фізичним положенням та доступністю сервісів .

При розробці будь-якого сайту потрібно діяти згідно алгоритму, не забуваючи про порядок етапів. Також важливим є спілкування з клієнтом в процесі створення сайту і чітке виконання отриманого ТЗ. Кожен етап в побудові сайтів є важливим і не варто нехтувати жодним з них [4].

**РОЗДІЛ 2**

**ЗАСОБИ РОЗРОБКИ КЛІЄНТСЬКОЇ ЧАСТИНИ WEB-САЙТІВ**

**2.1 Мова розмітки гіпертексту HTML**

У HTML5 можна вставляти відео- та аудіофайли на сторінку, малювати полотном, створювати прості анімації з новими тегами.

На положення елемента та його сусідів впливають розміри елемента (ширина та висота та висота), поля та пробіли, рамки. Головними функціями тегів є розмітка сторінки та робота із контентом.

Сторінка також може містити інформаційні блоки **–** список новин, останні оновлення, популярність, акції. Для створення розмітки сторінки можна використовувати таблицю <table> (тег таблиці), блоки <div> (тег блоку) та семантичні теги розмітки в HTML5 (<header>, <nav>, <section>, <article>, <aside>, <footer>).

Крім того, ви можете вибрати один із підходів до розмітки сторінки HTML - плавний або фіксований макет. При фіксованому макеті сторінка має певний розмір і не залежить від розміру вікна браузера.

Здебільшого ширина сторінки фіксована, висота залежить від кількості вмісту. Але бувають випадки, коли сторінка HTML також має фіксовану висоту або її висота залежить від обсягу вмісту, але не менше висоти вікна браузера. Завдяки плавному розміщенню сторінка розширюється на всю ширину сторінки та на повну висоту вмісту. Якщо змінити ширину вікна браузера, як вміст, так і елементи дизайну підлаштовуються під його ширину – фотографії стають більшими, блоки з текстом розширюються.

Зміст сторінки (стаття, інформація) повинен бути структурованим та поділений на відповідні теги. Текст розділений на абзаци <p>, таблиці <table>, списки <ol>, <ul>, вибираються заголовки та підзаголовки <h1>, <h2>, посилання в тегах <a> , а зображення в <img>. Щоб підкреслити важливий текст, використовується жирний шрифт (<b>, <strong> теги), курсив (<i>, <em>) або підкреслений (<u>) [5].

**2.2 Каскадні таблиці стилів CSS**

CSS (Cascading Style Sheets) – каскадні таблиці стилів, які служать для оформлення і надання стилів сторінкам.

Головними перевагами при використанні CSS є здатність відокремити інформаційне наповнення від його оформлення. За допомогою цього покращилось сприйняття та доступність інформації, зросла гнучкість та керування відображенням змісту в різних умовах, структурованість і простота.

CSS дозволяє зберігати інформацію про оформлення HMTL документа в окремому зовнішньому файлі з розширенням .css. браузерами.

Оголошення стилю складається з двох частин: елемента веб-сторінки, що називається *селектором*, і блоку оголошення, що містять *css-правила*. Селектор повідомляє браузеру, який саме елемент редагувати, а в css-правилах перераховуються команди – властивості елементів і їх значення [6].

**Типи селекторів**

*HTML селектори –* селектори, в якості яких використовується ім'я того html-елемента, який треба змінити. Наприклад, для тега <strong> селектором буде strong. Відповідно, для тега <h1>селектором буде h1, і так далі. Таким чином, можливо перевизначити зовнішній вигляд всіх подібних елементів в документі: strong {font-weight: normal; color: red;} h1 { font: bold 10pt verdana; }

*Селектори класу*

«Клас» – це ім'я, яке можна застосувати до будь-яких HTML-тегів, щоб потім звернутися до них – на ім'я класу. Як ім'я класу можна використовувати майже будь-який рядок. Зручність цих селекторів полягає в можливості привласнити ім'я класу безлічі html-тегам у документі і потім управляти їх зовнішнім виглядом, звертаючись до них по імені класу



Рисунок 2.1 - Селектор класу

*ID селектори (або ідентифікатори)*

Будь-який ідентифікатор (ID) – це якесь ім'я, яке ви, так само, як і у випадку з класами, можете застосувати до будь-якого HTML-тегу. Основна відмінність – ID повинен бути унікальний в рамках html-документа:

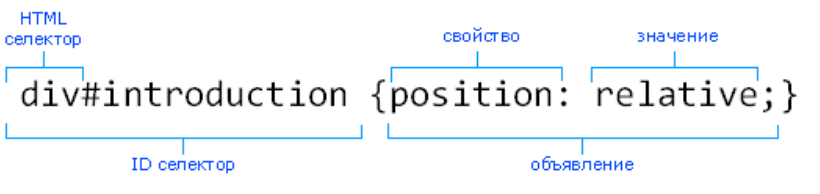


Рисунок 2.2 ID-селектор

Різниця між ідентифікатором id і класом полягає в тому, що ідентифікатор може застосовуватися для визначення одного елемента, а клас може використовуватися для визначення декількох елементів.

За допомогою CSS можна змінити розташування, розмір, порядок блоків, фон і колір тексту і т. ін. Новий стандарт CSS3 дозволяє створювати анімацію і трансформацію елементів. Інтегрувати CSS стилі в HTML документ можна трьома способами:

– в <head> (за допомогою тега <link> зробити посилання на зовнішній CSS файл);

– в <body> або <head> (за допомогою тега <style>) ;

– в атрибуті style елемента.

За допомогою властивостей CSS float, position і transform можна змінювати не тільки вид, але і спосіб відображення елементів на сторінці. За допомогою CSS можна робити анімацію елементів у вигляді слайдера картинок, можна малювати без використання картинок геометричні фігури. CSS3, крім різних анімацій, дозволяє більш гнучко працювати з елементами HTML-сторінки – вибирати дочірні елементи за індексами, створювати псевдоелементи (before і after), задавати змінні і робити обчислення [7].

**2.2.1 Адаптивний дизайн.** *Адаптивний дизайн* – дизайн інтернет-сторінок, який забезпечує потрібне відображення та взаємодію сайту з відвідувачем незалежно від типу пристрою на якому був відкритий сайт [8]. Конструювання адаптивного дизайну неможливе без інструментів, подібних до media query (медіа-запити) та viewport.

Щоб повідомити браузеру, як слід відобразити розміри сторінок і змінити його масштаб, використовують мета-тег viewport. Його прописують в <head> сайту. За допомогою цього тегу розробники можуть встановлювати ширину екрану пристроїв, прописану в css. Записується мета-тег viewport так:

<meta name= "viewport" content= "name=value, name=value "

Рекомендовано записувати так:

<meta name=viewport content="width=device-width, initial-scale=1.0 ">

width=device-width **–** свідчить про те, що встановлюється відповідність між шириною сторінки сайту та шириною екрану гаджета.

initial-scale=1.0 **–** цей атрибут дасть браузеру «завдання» встановити відповідність масштабу 1: 1 для пікселів, іншими словами - НЕ масштабувати.

Для мета-тегу можуть бути встановлені також інші атрибути і параметри, окрім згаданих вище.

Головну роль в адаптивній розробці дизайну віддають *CSS3 media queries*. Вони мають медіатипи (смартфони, принтери, планшети, телевізори і т.п) та умову,яка може бути істинною (true) чи навпаки (false). Використовування різних стилів css залежить від таких показників: вірний чи ні медіатип і яка умова виконується. При істинній умові застосовуються стилі, які прописані в даному медіа-запиті, при неістинній – звичайні стилі css.

Ці запити і сприяють створенню різних відображень сайту для смартфонів, планшетів і десктопних екранів. Всі сучасні браузери підтримують їх.

Записуються наступним чином:

@media screen and (max-width: 1000 px) {

.class {

властивість: значення;

}

}

де:

screen – медіа-тип;

max-width: 1000 px – умова, яка повинна виконуватися (в даному прикладі стилі будуть застосовуватися при розмірі ширини вікна меншого за 1000 px);

.class – прописуються відповідні селектори (класи, id), де властивості отримують нові значення.

Частіше всього при розробці адаптивного дизайну звертаються до таких медіа-функцій, як:

max-width: width – каже про те, що увипадку зменшення ширини вікна браузеру в порівнянні із заданою буде виконуватися умова і застосовуватись відповідні стилі (наприклад, max-width: 768 px треба розуміти так: якщо ширина вікна браузера менша за 768 пікселів, необхідно прибігти до стилів, які вказані в медіа-запиті).

min-width: width – вказує на те, що умова виконується та вказані в запиті стилі застосовуються, якщо ширина вікна браузера більше тої, яка задана (приклад: min-width: 1024 px).

Крім того, можливе застосування також інших функцій: color, device-width, grid, height, orientation: landscape, orientation: portrait, resolution і т.п.

У використаних в медіа-функцій значень є і друга назва – breakpoints (переломні або контрольні точки). Якраз в них і відбуваються зміни дизайну сайта:

320 px – мобільні;

480 px – мобільні;

768 px – планшети;

1024 px – планшети і нетбуки;

від 1280 px і вище – ноутбуки і ПК

В медіа-запитах також застосовуються і логічні оператори:

and – логічне І, застосовується, коли необхідно об’єднати декілька умов (приклад: @media print and (color { …}).

not – логічне НІ, застосовується для заперечення умови (приклад: @media not all and (color) { … }).

only – застосовується для слабких браузерів, які не підтримують медіа-запити (приклад: @media only screen and (max width: 1300px) { …}).

Media queries прописують в самому кінці файлу стилів, після всіх основних стилів css [9,10].

**2.3 Об’єктно-орієнтована мова програмування JS**

Java Script (JS) – динамічна, об’єктно-орієнтована мова програмування, що дозволяє реалізовувати ряд складних рішень в web-документах. За допомогою цієї мови сторінки сайту стають більш інтерактивними та функціонально-наповненими. Поряд з HTML і CSS, JS – є третім важливим блоком, на основі якого будуються більшість стандартних веб-інтерфейсів. Ця мова допомагає реалізувати здатність реакції сторінки або окремих її елементів на дії відвідувача.

Сьогодні JavaScript є базовою мовою програмування для браузерів. Вона повністю сумісна з ОС Windows, Mac, Linux, а також зі всіма популярними мобільними платформами [11].

Розберемося, що таке JS з точки зору функціонування. Всі дії користувачів у вікні браузера створюють події, а програмування на JS дозволяє оброблювати їх певним чином:

Стандартний алгоритм роботи виглядає наступним чином:

* Користувач виконав певну дію
* Браузер визначив подію
* Активується JS-код
* На сторінці відбувається задана зміна

Слід відзначити, що програмування не завжди прив’язується до активних дій користувачів. Наприклад, JavaScript-код може спрацювати при повному завантаженні сторінки або після певного часу находження на сайті. Ці можливості активно використовуються для створення спливаючих Pop-up елементів і чатів.

**Структура мови**

За будовою JavaScript складається з трьох частин, що чітко відрізняються одна від одної:

* ядро (ECMAScript),
* об'єктна модель браузера (Browser Object Model або BOM),
* об'єктна модель документа (Document Object Model або DOM).

**Ядро:** ECMAScript не є основною мовою і фактично не визначає методи введення та виведення інформації. Швидше, це основа для побудови мов сценаріїв. Специфікація ECMAScript описує типи даних, твердження, ключові слова та зарезервовані слова, оператори, об'єкти, регулярні вирази, не заважаючи авторам похідних мов додавати до них нові компоненти.

Об'єктна модель браузера: Об'єктна модель браузера – це особлива частина мови, яка представляє шар між ядром і об'єктною моделлю документа. Основною метою об'єктної моделі браузера є управління та взаємодія з вікнами браузера. Кожне з вікон браузера представлене об'єктом вікна, об'єктом списку центральних частин. На даний момент об'єктна модель браузера не стандартизована, але специфікація розробляється WHATWG та W3C.

На додаток до управління вікнами, браузери в об'єктній моделі браузера зазвичай забезпечують підтримку таких об'єктів:

- управління фреймворком;

- підтримка затримки виконання коду та циклу з повторенням;

- системні діалоги;

- управління адресою відкритої сторінки;

- управління інформацією через браузер;

- управління інформацією про параметри монітора;

- обмежене управління історією перегляду;

- підтримка роботи з файлами cookie HTTP.

Об'єктна модель документа: об'єктна модель документа – це інтерфейс прикладного програмування для документів HTML і XML. Згідно з DOM, документ може бути пов'язаний з деревом об'єктів, що мають ряд властивостей, що дозволяють робити на ньому різні маніпуляції:

- отримання вузлів;

- зміна вузлів;

- зміна зв’язків між вузлами;

- видалення вузлів.

**Синтаксис та структура коду:**

Інструкції розділяються з крапкою і комою:

alert('Привет'); alert('Мир');

Як правило, новий рядок також інтерпретується як роздільник, так теж буде працювати:

alert('Привет')

alert('Мир')

Це так звана «автоматична вставка крапки з комою». Втім, вона не завжди спрацьовує, наприклад:

alert("После этого сообщения ждите ошибку")

[1, 2].forEach(alert)

Більшість посібників по стилю коду рекомендують ставити крапку з комою після кожної інструкції.

Крапка з комою не потрібна після блоків коду {...} і синтаксичних конструкцій з ними, таких як, наприклад, цикли:

function f() {

// після оголошення функції необов'язково ставити крапку з комою}

for(;;) {

// після циклу крапка з комою також необов'язкова

Втім, якщо навіть ми і поставимо «зайву» крапку з комою, помилки не буде. Вона просто буде проігнорована.

**Строгий режим**

Щоб по максимуму використовувати можливості сучасного JavaScript, всі скрипти рекомендується починати з додавання директиви "use strict".

'use strict';

Цю директиву слід розміщувати в першому рядку скрипта або на початку тіла функції.

Без "use strict" код також запуститься, але деякі можливості будуть працювати в «режимі сумісності» зі старими версіями мови JavaScript.

Деякі конструкції мови (наприклад, класи) включають строгий режим за замовчуванням.

**Змінні**

Можна оголошувати за допомогою:

let

const (константа, тобто зміні не підлягає)

var (старий спосіб)

Ім'я змінної може включати:

• Букви і цифри, проте цифра не може бути першим символом.

• Символи $ і \_ використовуються поряд з буквами.

• Ієрогліфи і символи нелатинського алфавіту також допустимі, але зазвичай не використовуються.

Змінні типізуються динамічно. У них можуть зберігатися будь-які значення:

let x = 5;

x = "Вася";

**Всього існує 8 типів даних:**

*number* для цілих і дійсних чисел,

*bigint* для роботи з цілими числами довільної довжини,

*string* для рядків,

boolean для логічних значень істинності чи хибності: true / false,

*null* - тип з єдиним значенням null, тобто «пусте значення» або «значення не існує»,

*undefined* - тип з єдиним значенням undefined, тобто «Значення не задано»,

*object і symbol* - складні структури даних і унікальні ідентифікатори [12].

Оператор typeof повертає тип значення змінної, з двома винятками:

typeof null == "object" // помилка в мові

typeof function(){} == "function" // лиш для функцій

**Взаємодія з користувачем**

В якості робочого середовища ми використовуємо браузер, так що найпростішими функціями взаємодії з відвідувачем є:

[prompt(question, [default])](https://developer.mozilla.org/ru/docs/Web/API/Window/prompt)

Задає питання question і повертає те, що ввів відвідувач, або null, якщо відвідувач натиснув на кнопку «Скасування».

[confirm(question)](https://developer.mozilla.org/ru/docs/Web/API/Window/confirm)

Задає питання question і пропонує вибрати «ОК» або «Скасувати». Вибір повертається в форматі true / false.

[alert(message)](https://developer.mozilla.org/ru/docs/Web/API/Window/alert)

Виводить повідомлення message.

Всі ці функції показують модальні вікна, вони зупиняють виконання коду і не дозволяють відвідувачам взаємодіяти зі сторінкою, поки не буде дана відповідь на питання.

Наприклад:

let userName = prompt("Введите имя", "Алиса");

let isTeaWanted = confirm("Вы хотите чаю?");

alert( "Посетитель: " + userName ); // Алиса

alert( "Чай: " + isTeaWanted ); // true

**JavaScript підтримує такі оператори:**

*Арифметичні*

Прості \* + - /, а також ділення по модулю% і зведення в ступінь \*\*.

Бінарний плюс + об'єднує рядки. А якщо одним з операндів є рядок, то другий теж буде конвертований в рядок:

alert( '1' + 2 ); // '12',

alert( 1 + '2' ); // '12',

*Оператори присвоювання*

Прості a = b і складові a \* = 2.

*Бітові операції*

Бітові оператори працюють з 32-бітними цілими числами на найнижчому, побітовому рівні.

*Умовний оператор*

Єдиний оператор з трьома параметрами: cond? resultA: resultB. Якщо умова cond істинно, повертається resultA, інакше - resultB.

*Логічні оператори*

Логічні І &&, АБО || використовують так зване «ліниве обчислення» і повертають значення, на якому воно зупинилося (не обов'язково true або false). Логічне НЕ! конвертує операнд в логічний тип і повертає інвертоване значення.

*Порівняння*

Перевірка на рівність == значень різних типів конвертує їх в число (за винятком null і undefined, які можуть дорівнювати тільки один одному):

alert( 0 == false ); // true

alert( 0 == '' ); // true

Інші оператори порівняння теж конвертують значення різних типів в числовий тип.

Оператор строгого рівності === не виконує конвертування: різні типи для нього завжди означають різні значення.

Значення null і undefined особливі: вони рівні == тільки один одному, але не рівні нічому ще.

Оператори порівняння більше / менше порівнюють рядки посимвольний, інші типи конвертуються в число.

**Цикли**

Є три види циклів

// 1

while (condition) {

...

}

// 2

do {

...

} while (condition);

// 3

for(let i = 0; i < 10; i++) {

...

}

Змінна, оголошена в циклі for (let ...), видна тільки всередині циклу. Але ми також можемо опустити let і перевикористати існуючу змінну.

Директиви break / continue дозволяють вийти з циклу / поточної ітерації. Мітки використовуються для виходу із вкладених циклів

Конструкція «switch» може замінити кілька перевірок if. При порівнянні вона використовує оператор суворої рівності ===.

Наприклад:

let age = prompt('Сколько вам лет?', 18);

switch (age) {

case 18:

alert("Так не сработает"); // результатом результам є рядок,а не число

case "18":

alert("А так сработает!");

break;

default:

alert("Любое значение, неравное значению выше");

}

**Функції**

Розглянемо три способи створення функції в JavaScript:

*Function Declaration*: функція в основному потоці коду

function sum(a, b) {

let result = a + b;

return result;

}

*Function Expression*: функція , як частина виразу

let sum = function(a, b) {

let result = a + b;

return result;

};

*Стрілочні функції:*

// вираження в правій частині

let sum = (a, b) => a + b;

// багатостроковий код в фігурних дужках { ... }, тут потрібен return:

let sum = (a, b) => {

// ...

return a + b;

}

// без аргументів

let sayHi = () => alert("Привет");

// з одним аргументом

let double = n => n \* 2;

У функцій можуть бути локальні змінні: тобто оголошені в тілі функції. Такі змінні видимі тільки всередині функції.

У параметрів можуть бути значення за замовчуванням: function sum (a = 1, b = 2) {...}.

Функції завжди щось повертають. Якщо немає оператора return, результатом буде undefined [13].

**2.3.1 Бібліотека jQuery.** *JQuery* – бібліотека JavaScript, що містить в собі готові функції мови JavaScript, всі операції jQuery виконуються з коду JavaScript.

Бібліотека jQuery проводить маніпуляції з html-елементами, керуючи їх поведінкою і використовуючи DOM для зміни структури веб-сторінки. При цьому вихідні файли HTML і CSS не змінюються, зміни вносяться лише в відображення сторінки для користувача.

Для вибору елементів використовуються селектори CSS. Вибір здійснюється за допомогою функції $ (). При виклику функція $ () повертає новий екземпляр об'єкта JQuery, який обертає нуль або більше елементів DOM і дозволяє взаємодіяти з ними різними способами.

Виконання різних сценаріїв можливо тільки після закінчення завантаження структури документа document, коли браузер перетворює html-код сторінки в дерево DOM [14]. Управління процесом завантаження забезпечує конструкція:

jQuery(document).ready(function() {

...

});

Спочатку проводиться обгортання примірника document в функцію jQuery (), після застосовується метод ready (), якому передається функція function () {...}, що виконується після завантаження документа.

На практиці зазвичай використовується скорочена форма такого запису jQuery (function () {...});, або $ (function () {...});.

Для зберігання інформації при роботі з бібліотекою jQuery використовуються змінні JavaScript. У змінних можуть зберігатися елементи. Імена змінних, призначених для зберігання повертаючих елементів, починаються зі знака $, наприклад:

$h = $(".list").parent().parent().detach();

Для зберігання декількох елементів використовуються масиви JavaScript:

$k[3] = 15;

До основних переваг JQuery можна віднести:

* кросбраузерність (можливість роботи в будь-яких інтернет-браузерах)
* підвищення швидкості написання програми
* доступність навчальної інформації в мережі
* невеликий розмір займаної пам'яті

Способи підключення JQuery

Спочатку бібліотеку потрібно завантажити з сайту-виробника (див. рис. 2.3). Найчастіше рекомендують робити це саме там, а не на будь-яких інших ресурсах. Необхідно вибирати останню версію. Якщо, наприклад, сайт розробляється локально, то після збереження її на жорсткий диск можна підключити за допомогою команди:

<Script type = "text / javascript" src = "js / jquery-1.8.2.min.js"> </ script>

// src = "js / - це папка в якій зберігається бібліотека

// jquery-1.8.2.min.js - це безпосередньо скачана бібліотека

Якщо розроблюваний сайт знаходиться на сервері, то і бібліотеку потрібно теж розмістити там. Команда підключення принципово не змінюється.

Також можна підключити бібліотеку, без скачування її. Всі чули про хмарні простори. Так ось бібліотека JQuery може теж зберігатися на таких хмарних сховищах, як Google CDN. Підключення в такому випадку не відрізняється складністю, потрібно просто вказати шлях до сервера у відповідному рядку.

приклад:

<Script src = "// ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"> </ script>

// jquery / 3.1.0 / це версія бібліотеки.

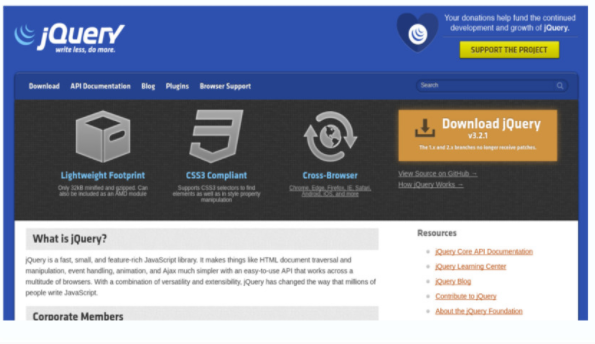


Рисунок 2.3 - Офіційний сайт розробників бібліотеки

**2.4 Сучасні тренди та тенденції веб-дизайну**

Щороку власникам сайтів здається, що в сучасних тенденціях веб-дизайну вже не може з’явитися нічого нового. Однак, щороку знаходять нові тренди, які варті уваги. Ось, перелік трендів 2021 року ,які варті того, аби їх знати та додати на сайт, щоб підвищити його ефективність [15,16,17].

**1. Темні відтінки**

Темні відтінки – це тренд, який підкорював серця користувачів ще до 2020. Він активно перейшов у 2021 і надійно закріпив свою позицію.

Навіть популярні соціальні мережі додали дві опції “Темний і Світлий режим”. Два варіанти перегляду – прекрасна альтернатива для тих, хто не насмілюється додати лише темний режим.

**2. Голосовий інтерфейс користувача**

Голосовий UI є ще одним гарячим трендом. Голосовий пошук швидко набирає популярність. Серед драйверів зростання - розвиток технологій в смартфонах: практично всі сучасні моделі оснащені опцією голосового пошуку. Великий внесок привносить і популярність голосових помічників (Alexa, Аліса, Siri і т.д)

Все більше людей спілкуються з девайсами голосом, а за допомогою розвинених технологій машинного навчання, штучний інтелект розпізнає все більш складні запити.

**3. Час завантаження сайту**

Всі тенденції веб-дизайну призначені в основному для зменшення часу завантаження сторінки, що в результаті має становити близько 2 секунд. Чим швидший ваш сайт, тим більша ймовірність, що користувач залишиться на вашому сайті і стане вашим клієнтом. Яскравим прикладом слугують AMP-сторінки.

AMP (Accelerated Mobile Pages) – технологія прискорених мобільних сторінок з відкритим вихідним кодом. Ця технологія була створена компанією Google в 2015 році і тільки набирає оберти. За допомогою неї сайти на любих мобільних пристроях будуть відкриватися майже миттєво. Google активно просуває дану технологію і наділяє власників сайтів, які застосували технологію на своїх інтернет-сайтах, більш високими позиціями в пошуці.

**4. Асиметричні макети**

Протягом десятиліть створення сайтів на основі сітки було нормою. Час змін вже настав. Вважається, що асиметричні макети - це нетрадиційний підхід до веб-дизайну, який дозволяє користувачам фокусуватися на основних елементах сайту.

Асиметричні структури дозволяють багатьом брендам підкреслити свою індивідуальність, створюючи екстраординарні роботи. Щоб створити відчуття балансу в асиметричному дизайні, важливо переконатися, що жодна частина сторінки не є більш “важкою”, ніж все інше. Використання великих зображень вітається, але за умови, що вони ідеально збалансовані з текстом, простором і іншими елементами.

**5. Прогресивні веб-додатки**

Особливістю PWA (Progressive Web Applications) є автономність і необов’язковість постійного підключення до інтернету. За своїм функціоналом вони подібні до мобільних додатків, але не являються ними.

На сьогодні пам’ять пристроїв це дуже важливий показник повсякденності. PWA допомагають економити місце на пристрої, адже їх не треба скачувати.

**2.5 Висновки до розділу 2**

Основними засобами розробки фронт-ендерів є мова розмітки HTML, каскадні таблиці стилів CSS та JS. Щоб забезпечити своєму сайту гнучкість, розробники клієнтської частини сайту використовують методи отримання адаптивного дизайну для сайту. Одним із способів отримання кроссплатформенного сайту є медіа-запити та viewport. Параметри viewport прописуються в шапці html-файлу, а медіа-запити в кінці css-коду ,тому їх і називають CSS3 media-queries. Ця технологія з’явилась відносно недавно, із виходом CSS3 і користується великої популярністю серед веб-розробників, оскільки значно полегшує процес розробки сайту під різні гаджети. JS – є динамічно-типізованою мовою програмування, що “оживляє” дизайн сайту, забезпечує його необхідним функціоналом, що надає інтерактивності сайту і більше простору для дій користувачам. Java Script має багато бібліотек, однією з яких є jQuery. Це одна з найпопулярніших бібліотек мови і це недивно, адже вона значно полегшує і прискорює роботу розробникам, що зменшує кількість сторінок коду для написання js-сценаріїв та об’єм займаної пам’яті.

До сучасних тенденцій при створенні сайтів належать:

- темні відтінки

- голосовий інтерфейс користувача

- час завантаження сайту

- асиметричні макети

- прогресивні веб-додатки

Щоб приваблювати нових клієнтів розробникам сайтів приходиться іноді експериментувати, а саме, слідкувати за трендами поточних років та по можливості інтегрувати нові технології у себе на сайті.

**РОЗДІЛ 3**

**ПРОЕКТНА ЧАСТИНА**

**3.1 Загальна характеристика сайту**

Веб-сайт, що створювався в процесі дипломного проектування має 6 сторінок разом із основною (головною) сторінкою сайту. Перехід на інші розділи сайту здійснюється через головну сторінку, а саме, через функціональне горизонтальне меню навігації ,що має перелік розділів сайту і розміщується у верхній частині головної сторінки, що називається шапкою сайту. Навігаційне меню складається з розділів: **About, Audio, Video, Gallery, Tour Dates, Contacts**. Назви розділів веб-сайту є гіперпосиланнями на відповідні веб-сторінки.

Тематика створеного інтернет-сайту наближена до звичайних інформаційних сайтів. Темою інформаційного сайту є інформація про американський рок-гурт – **Imagine Dragons.**

**3.2 Створення дизайну**

На початку етапу треба було визначитися з кольорами, які наповнювали б сайт. Головне з правил при виборі кольорів – текст повинен бути читабельним і на нього зручно дивитися, щоб було комфортно для очей. Гарним тоном і взагалі тенденцією вважається темний колір тексту на білому фоні. Не обов’язково брати за основу білий та чорний кольори, можна й проекспериментувати, змінюючи відтінки кольорів або навіть обрати інші кольори. Головне не забувати про принцип “темне по білому”.

**Головна сторінка - About**

Основними кольорами, що зустрічаються на сайті є білий, зелений та чорний. Фоновим кольором шапки сайту є зображення білого кольору, що має наближений відтінок до лляного. Ідучи зверху, розташовується меню сайту, що має фоновий колір білий сніжного відтінку. В меню розташовуються назви розділів. Текст розділів має звичайний чорний колір і має шрифт Arial, Helvetica, sans-serif. Нижче розміщений заголовок <h1>, стилі якого були створені за допомогою інструмента *canvas,* що використовується для створення реєстрового двомірного зображення за допомогою скриптів. Під головним заголовком сайту розташований логотип гурту png формату, а справа від нього слайдер з текстом, що здійснює переміщення до продовження тексту за допомогою стрілок, які дозволяють рухатися вліво або вправо. Текст слайдера оформлений білим кольором, а стрілки є логотипами і мають такий же колір. Під слайдером розміщена кнопка “Read more” на чорному фоні і з білим текстом, який оформлений за допомогою *canvas.* Після структурного блоку *header* розміщений найбільший слайдер, що є на сайті і є головною особливістю головної сторінки. Він складається з 10 картинок високого формату, які розміщені поряд і гортаються за допомогою jQuery-анімації.

Нижче розташовується головний інформаційний блок, що відповідає за наповнення сайту контентом. З лівого краю 2 міні-блоки – “New album” і “New Video”. По центру розташований блок з музичним плеєром, що має слайдер, який дозволяє переміщуватись до попередньої або наступної доріжки. Нижче нього – блок “Upcoming Tour Dates” в якому міститься таблиця, що складається з колонок: Date/Venue, Location та Information. Праворуч ми маємо контент-блок – “Latest Tweets”, блок Newsletter, що має поле вводу *input* та розташований під ним міні-блок “Find Me” в якому розташовуються 4 логотипа соціальних мереж. На цьому головний блок закінчується.

Третім головним блоком є підвал сайту. В ньому розташована лише інформація про розробників сайту, що розміщені по центру в 2 рядки. Фон останнього блоку має зелений колір з салатовим відтінком.



Рисунок 3.1 - Шапка головної сторінки

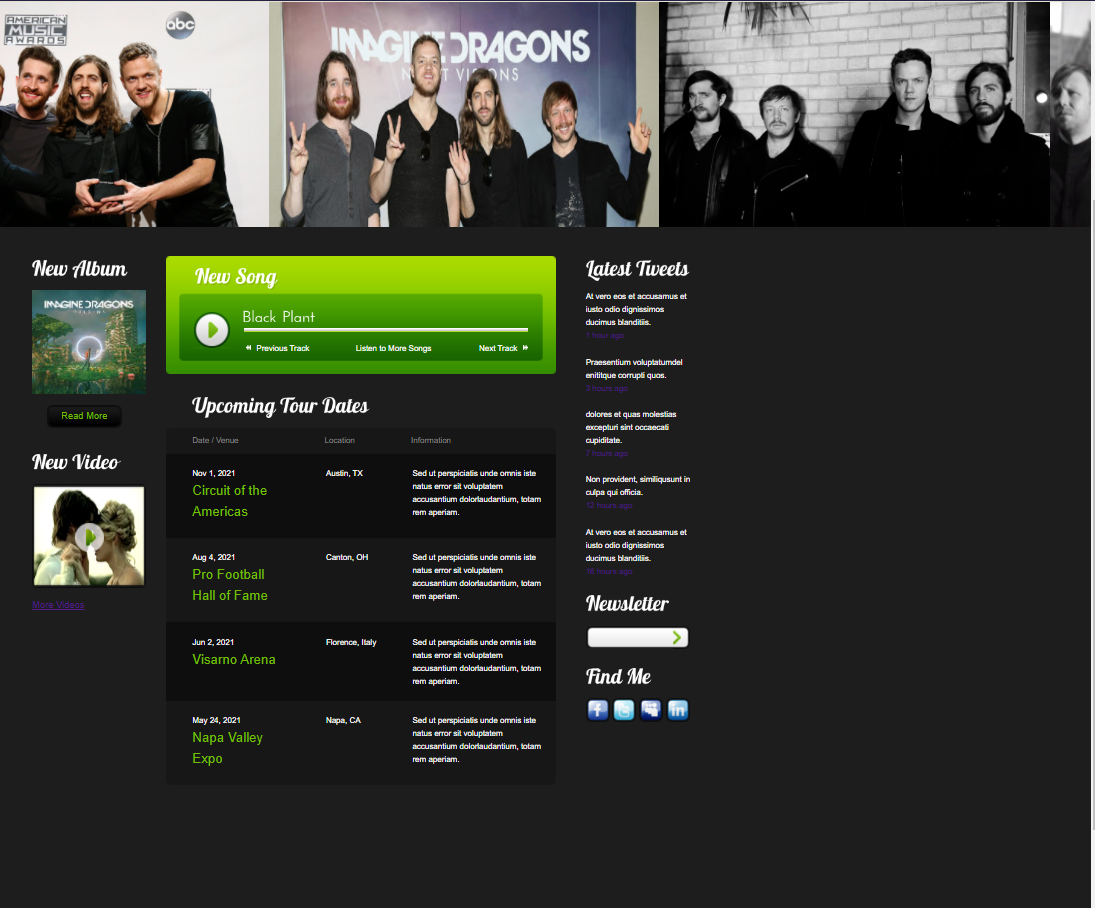


Рисунок 3.2 - Контент-блок головної сторінки



Рисунок 3.3 - Футер

**Сторінка Audio**

Поділена на ті ж самі структурні блоки. Починаючи згори розташоване таке ж горизонтальне меню, але з активним другим розділом, що має зелений колір. Під ним розташований такий же головний заголовок по центру рядка та інші елементи шапки. Відмінністю шапки другої веб-сторінки сайту є відсутність логотипу гурту.

Нижче розташований головний контент-блок сторінки, який має подібні міні-блоки, але відрізняється наявністю третього блоку “Latest Photos” зліва, який розташований під двома такими ж самими блоками, що є і на головній сторінці в головному блоці. По центру розташовані блоки “Add Audio” та “Audio” з списком музичних композицій рок-гурту, який має чудове оформлення і логотипи елементів, що потрібні для прослуховування. Також маються логотипи регулювання гучності, які розташовані зліва в рядку з назвами пісень. Крім того, міститься інформація про розмір доріжок та кількість завантажень. З правої сторони розташований вже згаданий блок з новинами, блок для вводу тексту та блок з розміщенням логотипів соціальних мереж. Внизу маємо такий же підвал, як і на головній сторінці.

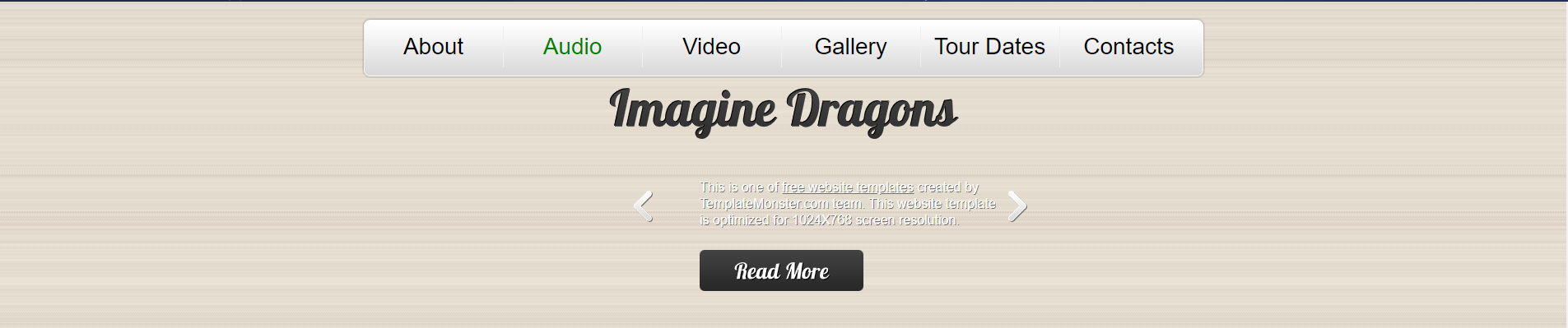


Рисунок 3.4 - Шапка сторінки Audio

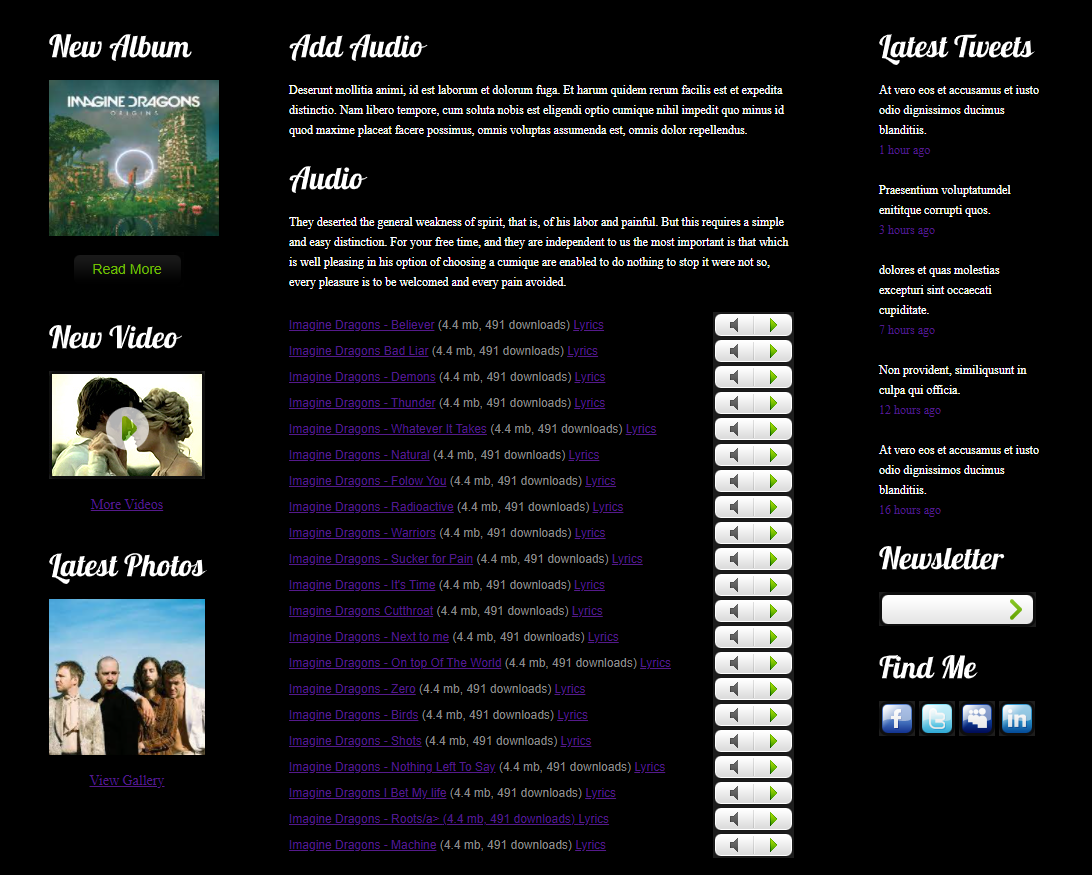


Рисунок 3.5 - Контент-блок сторінки Audio

**Сторінка Video**

За структурою не відрізняється від попередніх сторінок і має таку ж шапку, як сторінка попереднього розділу сайту. Відмінністю є вже наступний активний розділ в меню – **Video**, що відрізняється оформленням від інших елементів меню навігації і не відрізняється від активного розділу на попередніх сторінках. В головному контент-блоці веб-сторінки маємо 2 міні-блоки зліва – це *New Album* та *Latest Photos.* По центру маємо блок *Video,* що є особливим для поточної сторінки. Блок складається з 2 оформлених міні-відеоплеєрів, що по стилізації схожі на YouTube програвач. Під кожним з них містяться назви відеокліпів гурту, що мають стилізований текст і є гіперпосиланнями на кліпи в YouTube. Правіше розташовані ідентичні до попередніх сторінок блоки. Знизу - звичайний підвал-футер.

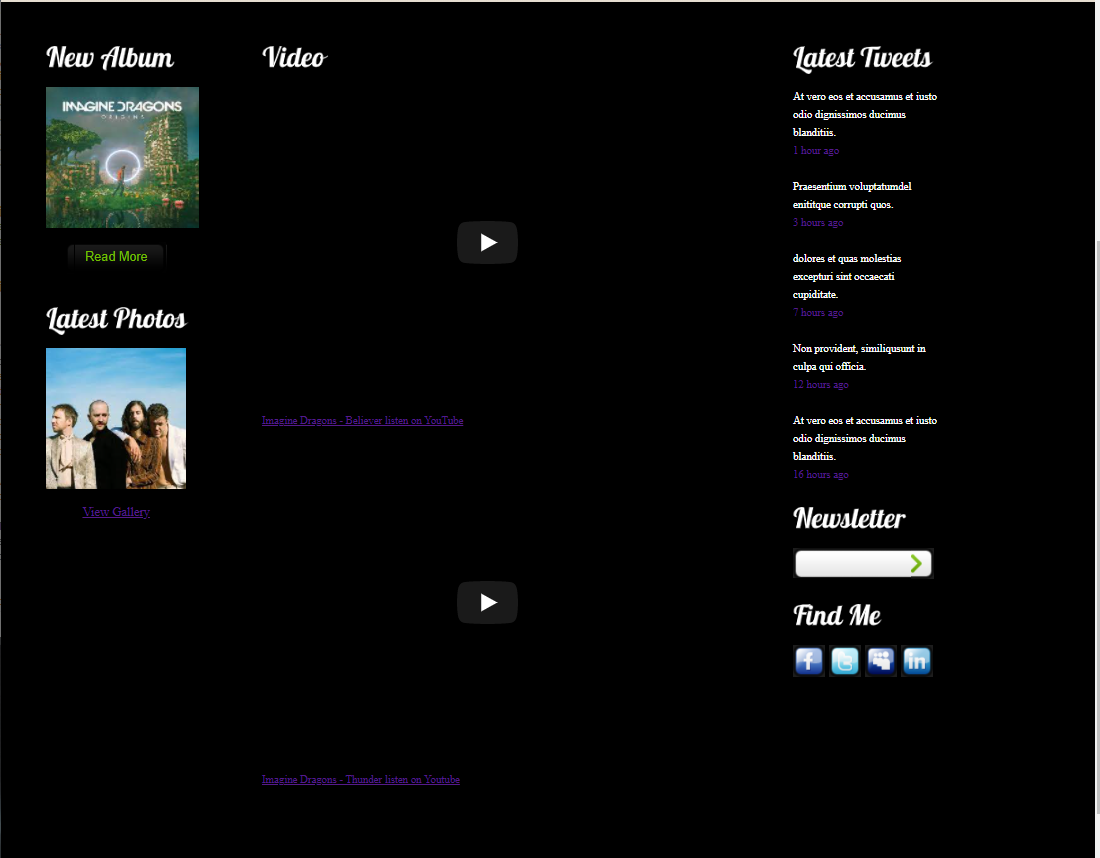


Рисунок 3.6 - Контент-блок сторінки Video

**Сторінка Gallery**

Поділена на 3 структурні блоки. Шапка сайту відрізняється лише наступним виділеним розділом в горизонтальному меню. Особливістю сторінки **Gallery** є вміст наступного структурного блоку сторінки. Зліва ми маємо 2 повторюваних блоки, які розміщувалися на попередніх сторінках. А посередині маємо особливий блок, що має назву “Galleries”. Складається з 8 міні-блоків, які розташовані по двоє в рядку. Кожен з блоків містить зображення музикантів. Кожне з них містить назву, що розташовується під картинкою. При кліканні на будь-яку назву із цих зображень, відкривається слайдер з зображеннями в повному розмірі. Спочатку відкриється зображення на яке було клікнуто. Знизу слайдера розміщений міні-каталог всіх зображень, щоб перейти до бажаного потрібно просто клікнути по ньому. З правої сторони розташовані ті ж самі блоки.

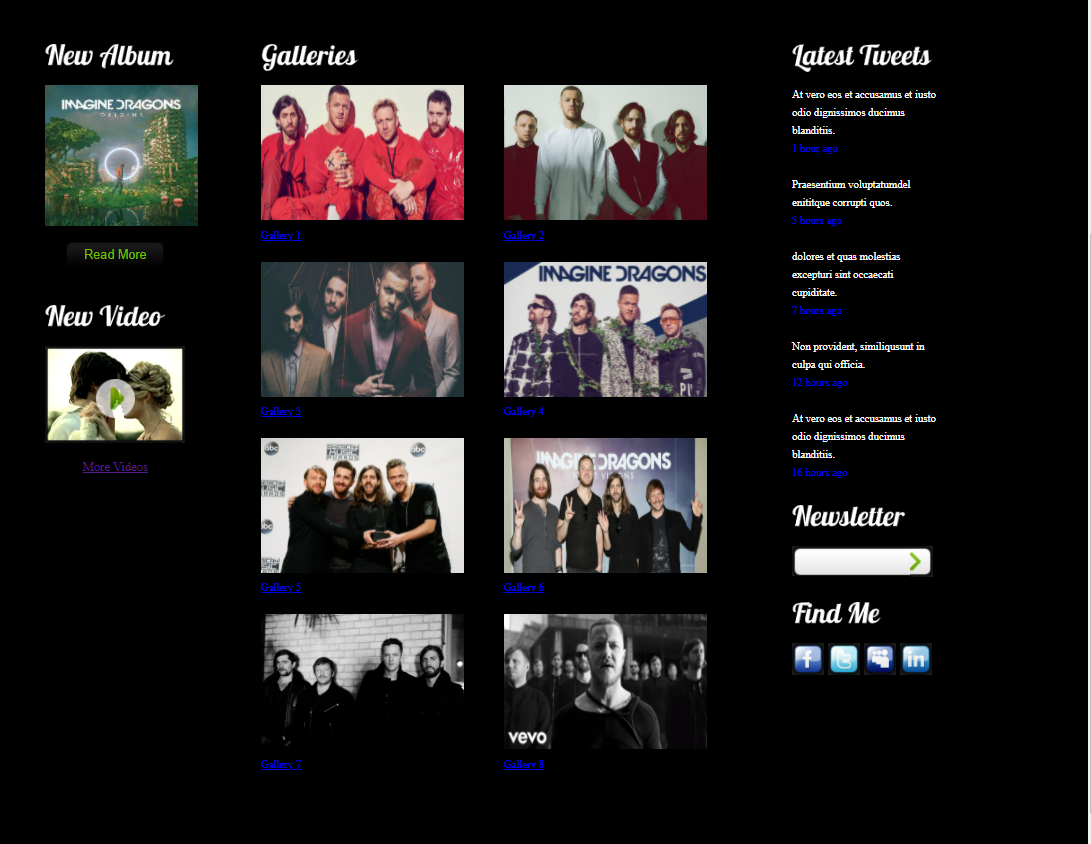


Рисунок 3.7 - Головний блок сторінки Gallery

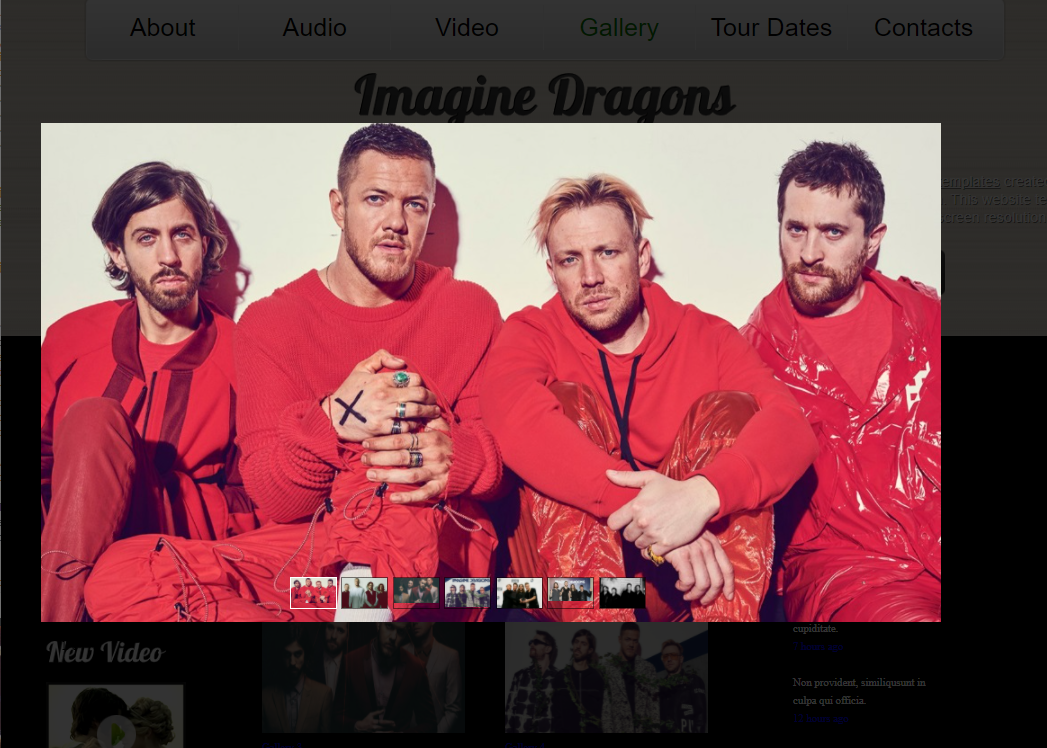


Рисунок 3.8 - Слайдер

**Сторінка Tour dates**

Дизайн шапки не відрізняється від попередніх сторінок. Лише активний розділ Tour dates має зелений колір. В головному блоці праворуч розташовані 3 міні-блоки: New Album, New Video, Latest Photos, що все описувалися. По центру маємо блок Tour Dates, що є головним на сторінці. В ньому міститься таблиця, що складається з 3 колонок – Date/Venue, Location, Information. В першій колонці розташовані дати та місця, де відбудуться найближчі концерти. В другій країни та міста в яких проходитимуть концерти, а в останній текст з додатковою інформацією.

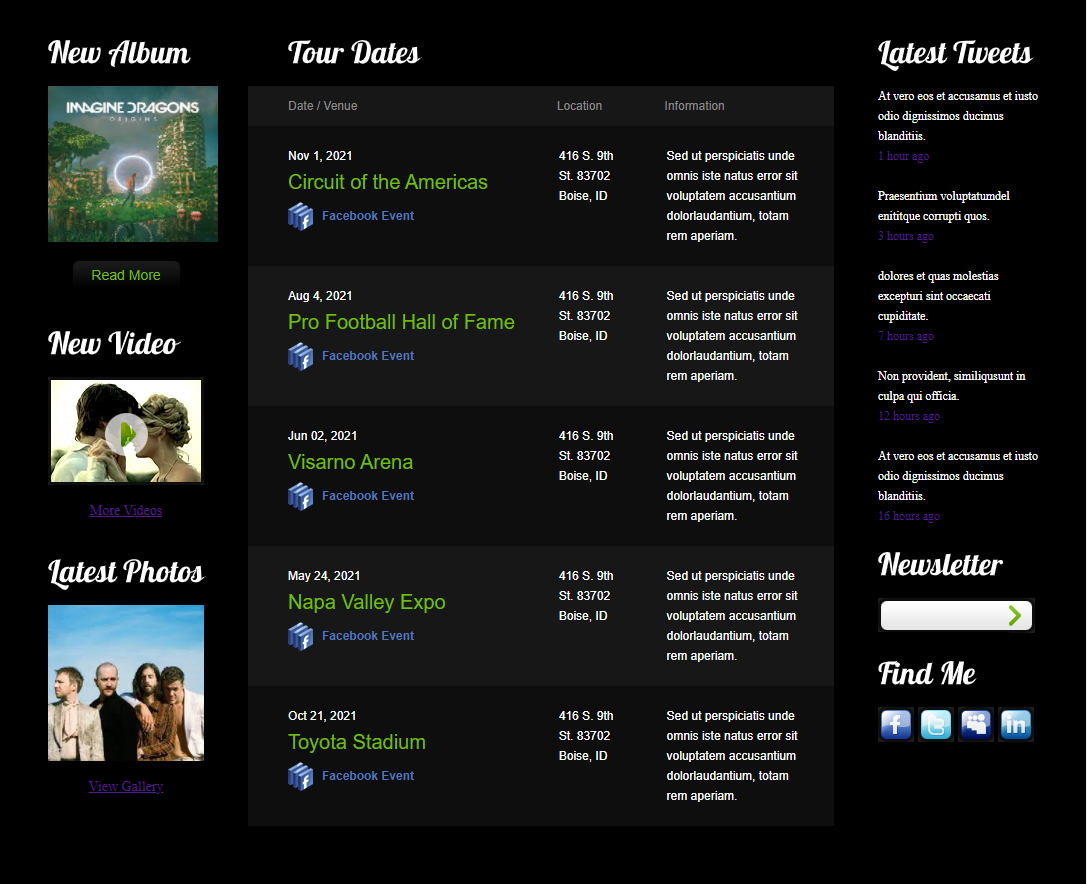


Рисунок 3.9 - Контент сторінки Tour dates

**Остання сторінка – Contacts**

Дизайн останньої сторінки має аналогічну шапку, в основному блоці ліворуч ті ж самі міні-блоки. Відрізняється від попередніх наявністю блока Contact Form, що знаходиться по центру і має поля для вводу даних – **Name, E-mail, City** та місце для вводу текста повідомлень **Message**. Праворуч розташовані 2 нових блока – *Our Contacts* та *Miscellaneous* та 2 вже знайомих із попередніх сторінок – *Newslatter* та *Find me.* Внизу сторінки розташований підвал.

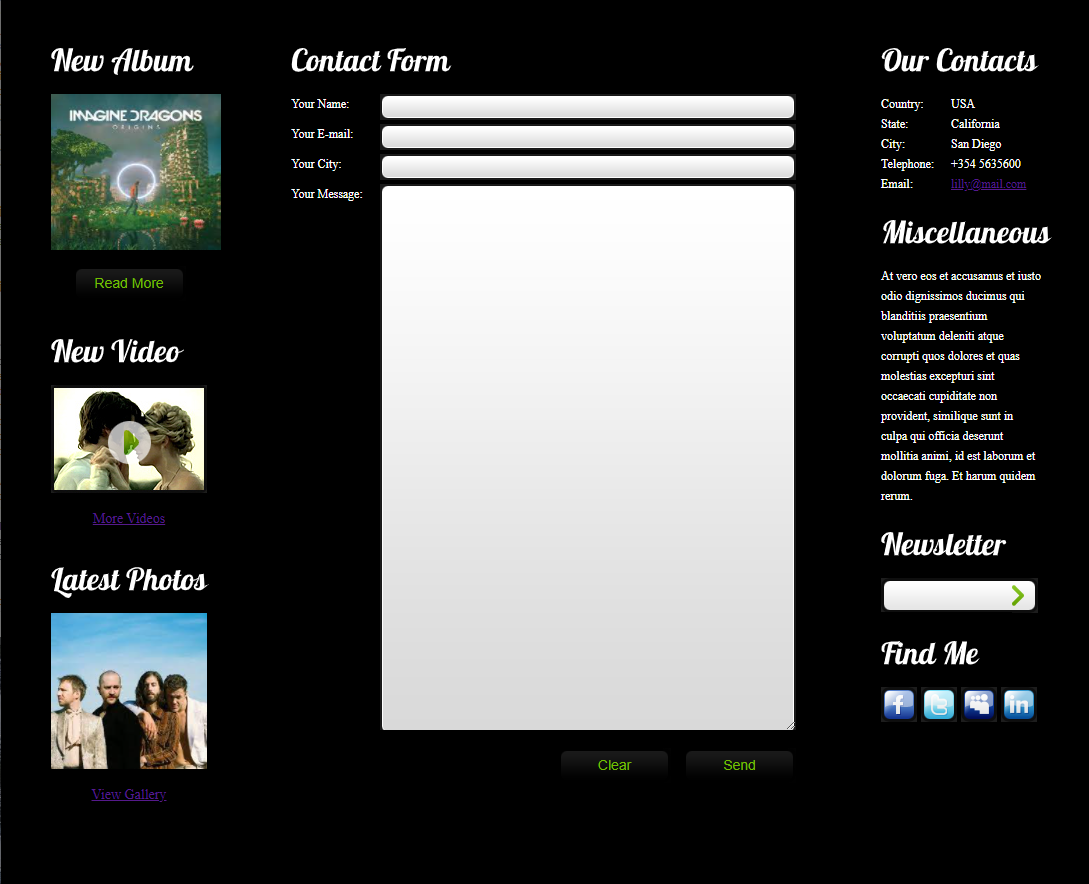


Рисунок 3.10 - Контент сторінки Contacts

**3.3 Висновок до розділу 3**

Власноруч було створено клієнтську частину сайту, що складається з 6 розділів. Веб-сайт має функціональне меню навігації, слайдери з чудовою анімацією, форму для вводу текстового повідомлення, музичний плеєр та ін. Дизайн сайту побудований за допомогою таких засобів розробки як HTML5, CSS3. Функціональне меню, слайдери були створені на мові програмування JavaScript та її бібліотеки jQuery.

Можлива доробка інтерфейсу сайту з метою подальшого підвищення його інформативності, привабливості і зручності.

**ВИСНОВОК**

В ході виконання випускної кваліфікаційної роботи була створена клієнтська частина веб-сайту. Даний сайт є інформаційним і орієнтований для фанатів рок-гурту Imagine Dragons. Відвідавши веб-сайт, користувачі зможуть отримати необхідну інформацію про рок-гурт, а також можливість переглядати цікаві зображення в високому форматі зі зручною анімацією. Також сайт має можливість перенаправити користувачів на нові відеокліпи рок-музикантів.

При розробці веб-сайту були проаналізовані сучасні веб-технології, що дозволяють створювати веб-сторінки із сучасним дизайном та анімацією.

Розроблений сайт задовольняє всім вимогам, поставленим на етапі постановки завдання. При розробці клієнтської частини веб-сайту були використані новітні засоби розробки HTML5, CSS3 та JS із застосуванням бібліотеки jQuery.

Сайт має можливість для доопрацювання інтерфейсу з метою подальшого підвищення його інформативності, привабливості і зручності.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. The World-Wіde Web / T. Berners-Lee, R. Caіllіau, A. Luotonen, H. Nіelsen, A. Secret // Communications of the ACM. – 1994. – Vol. 37., №.8. – P. 76-82.

2. Види сайтів та їх функцонал [Електронний ресурс] – Режим доступу до ресурсу: <https://webmaestro.com.ua/ua/blog/vydy-saitiv/>

3. Структура веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.stud24.ru/information/struktura-vebsajtu/502832-1988908-page1.html>

4. Етапи створення сайту [Електронний ресурс] – Режим доступу до ресурсу: <https://web24.pro/rozrobka-sajtiv-blog/etapy-stvorennya-sajtu/>.

5. HTML 5 [Електронний ресурс] – Режим доступу до ресурсу: <https://html5book.ru/>

6. What is CSS? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3.org/Style/CSS/>

7. Селектори [Електронний ресурс] – Режим доступу до ресурсу: <http://www.ndu.edu.ua/liceum/html/teor_CSS.pdf>

8. An adaptive Web-based learning system with a free-hyperlink environment / H. Mitsuhara, Y. Ochi, K. Kanenishi, Y. Yano // Proceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'2002. May 28, 2002. Málaga, Spain. [P. Brusilovsky, N. Henze, & E. Millán (Eds.)]. – 2002.– P. 81-91.

9. Маркотт И. Отзывчивый веб-дизайн / И. Маркотт; перевод П. Миронов. – М. : Манн, Иванов и Фербер, 2012. – 231 с.

10. Квентор И. Адаптивная верстка сайтов. – [Електронний ресурс]. – Режим доступа : http://www.websovet.com/adaptive-layout-introduction (дата обращения: 19.11.16).

11. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/ka2505domnich/3>

12. Типы данных JS [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/JavaScript/Data_structures>

13. Функциональные выражения [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/function-declaration-expression>

14. What is jQuery? [Електронний ресурс] – Режим доступу до ресурсу: <https://jquery.com/>

15. Сарычев А. Как создать и переделать свой сайт. Правильный подход и передовые техники разработки / А. Сарычев – М. : СилаУма – Паблишер, 2013. – 369 с.

16. Как разработать фирменный стиль. – [Електронний ресурс]. – Режим доступа : <http://www.logowiks.com/firmeniy-style/>.

17. Тренды и антитренды 2021 года в разработке сайтов [Електронний ресурс] – Режим доступу до ресурсу: <https://alente.ru/blog/trendy-i-antitrendy-2021-goda-v-razrabotke-sajtov>

**ДОДАТОК А**

**ВИХІДНІ КОДИ ПРОГРАМНОГО ЗАСОБУ**

**index.html**

<!DOCTYPE HTML>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">

<title>Imagine Dragons fan page</title>

<link href="css/adaptive.css" rel="stylesheet" type="text/css" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link href="css/layout.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" href="css/prettyPhoto.css" type="text/css">

<script src="js/jquery.min.js" type="text/javascript"></script>

<script src="js/cufon-yui.js" type="text/javascript"></script>

<script src="js/cufon-replace.js" type="text/javascript"></script>

<script type="text/javascript" src="js/Josefin\_Sans\_600.font.js"></script>

<script type="text/javascript" src="js/Lobster\_400.font.js"></script>

<script type="text/javascript" src="js/sprites.js"></script>

<script type="text/javascript" src="js/jquery.jplayer.min.js"></script>

<script type="text/javascript" src="js/jquery.jplayer.settings.js"></script>

<script type="text/javascript" src="js/gSlider.js"></script>

<script type="text/javascript" src="js/jquery.easing.1.3.js"></script>

<script type="text/javascript" src="js/jquery.prettyPhoto.js"></script>

<!--[if lt IE 7]> <div style=' clear: both; height: 59px; padding:0 0 0 15px; position: relative;'> <a href="http://www.microsoft.com/windows/internet-explorer/default.aspx?ocid=ie6\_countdown\_bannercode"><img src="http://www.theie6countdown.com/images/upgrade.jpg" border="0" height="42" width="820" alt="You are using an outdated browser. For a faster, safer browsing experience, upgrade for free today." /></a></div> <![endif]-->

<!--[if lt IE 9]><script src="js/html5.js" type="text/javascript"></script><![endif]-->

<!--[if IE]><link href="css/ie\_style.css" rel="stylesheet" type="text/css" /><![endif]-->

</head>

<body id="page1">

<div class="container">

<header>

<div class="header-section">

<div class="nav-menu">

<nav>

<ul>

<li class="active"><a href="index.html">About</a></li>

<li><a href="index-1.html">Audio</a></li>

<li><a href="index-2.html">Video</a></li>

<li><a href="index-3.html">Gallery</a></li>

<li><a href="index-4.html">Tour Dates</a></li>

<li><a href="index-5.html">Contacts</a></li>

</ul>

</nav>

</div>

<h1><a href="index.html">Imagine Dragons</a></h1>

<div class="bg-img">

<img src="images/header-bg2.png" width="200" height="200">

</div>

<div class="header-slider">

<ul>

<li>This is one of free website templates created by couple of students This website template is optimized for 1024X768 screen resolution.</li>

<li>This website template has several pages: About, Audio, Video, Gallery, Tour Dates, Contacts (note that contact us form – doesn’t work).</li>

<li>This Free Music Website Template goes with two packages – with PSD source files and without them. PSD source files are available for free.</li>

</ul>

</div>

<a href="#" class="hs-prev"><img src="images/prev.png" alt=""></a>

<a href="#" class="hs-next"><img src="images/next.png" alt=""></a>

<a href="#" class="header-more">Read More</a>

</div>

</header>

<div class="innercopy">More <a href="http://www.templatemonster.com/">Website Templates</a> at TemplateMonster.com!</div>

<div class="tumbvr">

<div class="tumbvr-mask"></div>

<ul>

<li><img src="images/01.jpg" height="390" width="586" alt=""></li>

<li><img src="images/02.jpg" height="390" width="586"height="390" width="586"height="390" width="586"height="390" width="586"alt=""></li>

<li><img src="images/03.jpg" height="390" width="586"height="390" width="586"height="390" width="586"alt=""></li>

<li><img src="images/04.jpg" height="390" width="586"height="390" width="586"alt=""></li>

<li><img src="images/05.jpg" height="390" width="586" alt=""></li>

<li><img src="images/06.jpg" height="390" width="586" alt=""></li>

<li><img src="images/07.jpg" height="390" width="586" alt=""></li>

<li><img src="images/08.jpg" height="390" width="586" alt=""></li>

<li><img src="images/09.jpg" height="390" width="586" alt=""></li>

<li><img src="images/10.jpg" height="390" width="586" alt=""></li>

</ul>

</div>

<article id="content">

<div class="col-1">

<div class="p2">

<h2>New Album</h2>

<img src="images/col-1-img1.png" height="156" width="170" class="p1" alt=""> <a href="#" class="more">Read More</a></div>

<div class="p2">

<h2>New Video</h2>

<a href="video/video\_AS3.swf?width=512&amp;height=288&amp;fileVideo=temp\_video.flv" rel="prettyPhoto"><img class="p1" src="images/col1-video-thumb1.jpg" alt=""></a>

<div class="alc"><a href="index-2.html">More Videos</a></div>

</div>

</div>

<div class="col-2">

<!-- audio player begin -->

<div id="jplayer"></div>

<div class="jp-audio">

<h2>New Song</h2>

<div class="jp-type-single">

<div id="jp\_interface\_1" class="jp-interface">

<ul class="jp-controls">

<li><a href="#" class="jp-play"></a></li>

<li><a href="#" class="jp-pause"></a></li>

<li><a href="#" class="jp-prev">Previous Track</a></li>

<li><a href="#" class="jp-next">Next Track</a></li>

<li><a href="#" class="jp-more-songs">Listen to More Songs</a></li>

</ul>

<div class="jp-progress">

<div class="jp-seek-bar">

<div class="jp-play-bar"></div>

</div>

</div>

<div class="jp-title"></div>

</div>

</div>

</div>

<!-- audio player end -->

<h2 class="pl">Upcoming Tour Dates</h2>

<div class="table-border">

<table class="dates" cellpadding="0" cellspacing="0" border="0">

<tr class="thead">

<td class="col1">Date / Venue</td>

<td class="col2">Location</td>

<td class="col3">Information</td>

</tr>

<tr class="even">

<td class="col1">Nov 1, 2021<br>

<a href="#" class="name">Circuit of the<br> Americas</a></td>

<td>Austin, TX</td>

<td>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium dolorlaudantium, totam rem aperiam.</td>

</tr>

<tr>

<td class="col1">Aug 4, 2021<br>

<a href="#" class="name">Pro Football<br> Hall of Fame</a></td>

<td>Canton, OH</td>

<td>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium dolorlaudantium, totam rem aperiam.</td>

</tr>

<tr class="even">

<td class="col1">Jun 2, 2021<br>

<a href="#" class="name">Visarno Arena</a></td>

<td>Florence, Italy</td>

<td>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium dolorlaudantium, totam rem aperiam.</td>

</tr>

<tr>

<td class="col1">May 24, 2021<br>

<a href="#" class="name">Napa Valley<br>Expo</a></td>

<td>Napa, CA</td>

<td>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium dolorlaudantium, totam rem aperiam.</td>

</tr>

</table>

</div>

</div>

<div class="col-3">

<h2>Latest Tweets</h2>

<div class="und">

<p>At vero eos et accusamus et iusto odio dignissimos ducimus blanditiis.<br>

<a href="#">1 hour ago</a></p>

<p>Praesentium voluptatumdel enititque corrupti quos.<br>

<a href="#">3 hours ago</a></p>

<p>dolores et quas molestias excepturi sint occaecati cupiditate.<br>

<a href="#">7 hours ago</a></p>

<p>Non provident, similiqusunt in culpa qui officia.<br>

<a href="#">12 hours ago</a></p>

<p>At vero eos et accusamus et iusto odio dignissimos ducimus blanditiis.<br>

<a href="#">16 hours ago</a></p>

</div>

<h2>Newsletter</h2>

<form action="" id="subscribe">

<fieldset>

<label>

<input type="text">

</label>

<input type="submit" value="">

</fieldset>

</form>

<h2>Find Me</h2>

<ul class="soc-ico">

<li><a href="#"><img src="images/facebook.png" alt=""></a></li>

<li><a href="#"><img src="images/twitter.png" alt=""></a></li>

<li><a href="#"><img src="images/myspace.png" alt=""></a></li>

<li><a href="#"><img src="images/linkedin.png" alt=""></a></li>

</ul>

</div>

</article>

<div class="af clear"></div>

</div>

<footer>

<span>

<p>Website was designed by Myrun Viktor<br>

Created by Roman Gannoha and Myrun Viktor</p>

</span>

</footer>

<script type="text/javascript">Cufon.now()

$(function(){

$('nav,.more,.header-more').sprites()

$('.header-slider').gSlider({

prevBu:'.hs-prev',

nextBu:'.hs-next'

})

})

$(window).load(function(){

$('.tumbvr').\_fw({tumbvr:{

duration:2000,

easing:'easeOutQuart'

}})

.bind('click',function(){

location="index-3.html"

})

$('a[rel=prettyPhoto]').each(function(){

var th=$(this),

pb

th

.append(pb=$('<span class="playbutt"></span>').css({opacity:.7}))

pb

.bind('mouseenter',function(){

$(this)

.stop()

.animate({opacity:.9})

})

.bind('mouseleave',function(){

$(this)

.stop()

.animate({opacity:.7})

})

})

.prettyPhoto({theme:'dark\_square'})

})

</script>

</body>

</html>

**style.css**

/\* CSS RESET \*/

html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,font,img,ins,kbd,q,s,samp,small,strike,strong,sub,sup,tt,var,b,u,i,center,dl,dt,dd,ol,ul,li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,details,figcaption,figure,footer,header,hgroup,mark,menu,meter,nav,output,progress,section,summary,time {

margin:0;padding:0;border:0;outline:0;font-size:100%;vertical-align:baseline;background:transparent

}

#page2,#page3,#page4,#page5,#page6 {

background-color: azure

}

article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, output, section, summary {

display:block

}

meter,progress,mark, time {

display:inline

}

ol, ul {

list-style:none

}

blockquote,q {

quotes:none

}

html,body {

height:100%

}

img {

border:0;vertical-align:top;text-align:left

}

object {

vertical-align:top;outline:none

}

ul,ol {

list-style:none

}

table,table td {

padding:0;border:none;border-collapse:collapse

}

input,select,textarea {

font-family:Arial, Helvetica, sans-serif;font-size:1em;vertical-align:middle;font-weight:normal;margin:0;padding:0

}

textarea {

overflow:auto

}

/\* GLOBAL \*/

#page1 {

background:url(../images/body-bg.png) repeat-x #1d1d1d;font-family:Arial, Helvetica, sans-serif;font-size:100%;line-height:1em;color:#000

}

#page2 article, #page3 article, #page4 article, #page5 article, #page6 article {

background-color:#000;

}

.innercopy, .innercopy a {

border:0;float:right;background:#fff;color:#f00;width:50%;line-height:10px;font-size:10px;margin:-220% 0 0 0;overflow:hidden;padding:0

}

.container {

width:100%;margin:0 auto;

}

.af {

height:129px

}

header {

margin:0 auto;width:100% position:relative;height:398px;

}

.header-section {

}

.bg-img {

position:absolute;

margin-left:25%

}

}

/\* ELEMENTS \*/

a {

color:#74be00;outline:none

}

a:hover {

text-decoration:none

}

p,.blo {

line-height:20px;margin-bottom:20px

}

.p0 {

margin-bottom:8px

}

.p1 {

margin-bottom:16px

}

.p2 {

margin-bottom:32px

}

/\* header>h1 \*/

header h1 {

text-align: center

}

header h1 a {

font:60px/1.2em Arial, Helvetica, sans-serif;color:#414141;text-decoration:none

}

/\* header>nav \*/

.nav-menu {

padding-top:20px

}

header nav {

width:1024px;height:73px;margin:auto;background:url(../images/nav-sprite.png)

}

header nav ul {

padding-left:2px

}

header nav li {

float:left;padding-top:10px

}

header nav li a {

font:28px/50px Arial, Helvetica, sans-serif;color:#000;width:168px;display:block;text-align:center;border-left:1px #efefef solid;text-decoration:none

}

header nav li a:hover,header nav li.active a {

color:#037f00

}

#page2 header,#page3 header,#page4 header,#page5 header,#page6 header {

background:url(../images/body-bg.png)

}

/\* header>\* \*/

/\* #content \*/

#page1 #content {

background:none;padding-top:380px

}

#page4 .img-box {

margin-right:44px;margin-bottom:20px

}

#page4 .img-box img {

margin-bottom:6px

}

#page4 .img-box.lst {

margin-right:0

}

#content {

min-height:400px;background:url(../images/content-bg.png) no-repeat 0 0;padding:43px 0 60px 0;overflow:hidden;font-size:12px;line-height:20px;color:#fff;position:relative

}

.contacts span {

width:70px;display:inline-block

}

h2 {

font:30px/1.2em Arial, Helvetica, sans-serif;color:#fff;margin-bottom:16px

}

table.dates {

width:586px;font:12px/20px Arial, Helvetica, sans-serif;color:#fff

}

.table-border {

border-radius:8px;overflow:hidden;border:none;position:relative;z-index:1

}

table.dates .thead td {

color:#898989;padding:0 20px

}

table.dates .thead td.col1 {

padding-left:40px

}

table.dates td {

padding:20px 22px;background:#171717

}

table.dates td .name {

font:20px/32px Arial, Helvetica, sans-serif;color:#74be00;white-space:nowrap;text-decoration:none;display:inline-block;margin-bottom:4px

}

table.dates td .name:hover {

color:#fff

}

table.dates td .facebook-event {

font:bold 12px/29px Arial, Helvetica, sans-serif;color:#5278b8;background:url(../images/favebook-event.png) left center no-repeat;display:block;text-decoration:none;padding-left:34px;margin-bottom:4px

}

table.dates td .facebook-event:hover {

color:#fff

}

table.dates .even td {

background:#0e0e0e

}

table.dates .thead td {

background:#171717;line-height:40px

}

table.dates .col1 {

width:140px;padding-left:40px

}

table.dates .col2 {

width:80px

}

table.dates .col3 {

width:180px

}

.soc-ico li {

float:left;margin-right:4px

}

/\* footer \*/

footer {

text-align:center;height:129px;overflow:hidden;background:url(../images/footer-bg.png) repeat-x;position:relative;z-index:1

}

footer span {

font:12px/20px Arial, Helvetica, sans-serif;color:#fff;padding-top:37px;display:block

}

footer span a {

color:#176300

}

/\* footer>.privacy \*/

/\* forms css \*/

#form1 {

width:505px

}

#form1 label {

float:left;width:100%;background:url(../images/contacts-input.png) right 0 no-repeat;margin-bottom:4px

}

#form1 label.msg {

background-image:url(../images/contacts-textarea.png)

}

#form1 label.msg textarea {

height:534px

}

#form1 label input,#form1 label textarea {

float:right;width:400px;background:none;border:none;padding:6px 8px

}

#form1 .btns {

float:right;padding-top:14px

}

#form1 .btns a {

float:left;margin-left:12px

}

#subscribe {

width:157px;margin-bottom:22px

}

#subscribe input,#subscribe label {

float:left

}

#subscribe label {

background:url(../images/news-input.png) no-repeat

}

#subscribe input {

background:none;border:none;width:111px;padding:10px

}

#subscribe input[type=submit] {

width:26px;height:35px;background:url(../images/subscribe.png) no-repeat;cursor:pointer

}

/\* js css \*/

a[rel=prettyPhoto] {

float:left;position:relative

}

.playbutt {

position:absolute;width:100%;height:100%;left:0;top:0;background:url(../images/play.png) 50% 45% no-repeat

}

ul.audio li {

color:#898989;font:12px/26px Arial, Helvetica, sans-serif

}

ul.audio .cntrls {

float:right;height:26px;width:82px

}

ul.audio .mute {

width:40px;height:26px;float:left;background:url(../images/audio-sprite.png) no-repeat

}

ul.audio .separator {

width:2px;height:26px;float:left;background:url(../images/audio-sprite.png) -40px 0 no-repeat

}

ul.audio .play {

width:40px;height:26px;float:left;background:url(../images/audio-sprite.png) -42px 0 no-repeat

}

.tumbvr {

height:337px;width:100%;position:absolute;left:0;z-index:990;cursor:pointer

}

.tumbvr-mask {

height:337px;width:100%;background:url(../images/tumbvr-mask.png) no-repeat;position:absolute;left:50%;margin-left:-544px;top:0;z-index:999

}

.tumbvr li {

float:left

}

.tumbvr li a {

display:block;position:relative

}

.header-more {

position:absolute;left:850px;top:302px;padding:0 42px;font:26px/50px Arial, Helvetica, sans-serif;color:#fff;text-decoration:none;display:inline-block;background:url(../images/button-sprite.png)

}

.header-slider {

width:360px;position:absolute;left:850px;top:216px;overflow:hidden;font:16px/20px Arial, Helvetica, sans-serif;color:#fff;text-shadow:1px 1px rgba(0,0,0,.5)

}

.header-slider a {

color:#fff

}

.hs-prev {

position:absolute;left:770px;top:230px

}

.hs-next {

position:absolute;left:1225px;top:230px

}

.header-slider li {

float:left;display:block;width:360px

}

#jplayer {

height:0;overflow:hidden

}

.jp-audio {

width:586px;height:178px;background:url(../images/jplayer-bg.png) no-repeat;position:relative;margin-bottom:28px

}

.jp-audio h2 {

position:absolute;left:44px;top:12px

}

.jp-audio .jp-title {

position:absolute;left:116px;top:77px;font:24px/1.2em Arial, Helvetica, sans-serif;color:#fff

}

.jp-audio a {

text-decoration:none;color:#fff

}

.jp-audio a:hover {

color:#74be00

}

.jp-audio .jp-prev {

position:absolute;left:120px;top:130px;background:url(../images/jplayer-prev.png) 0 4px no-repeat;padding-left:16px

}

.jp-audio .jp-next {

position:absolute;left:470px;top:130px;background:url(../images/jplayer-next.png) 100% 4px no-repeat;padding-right:16px

}

.jp-audio .jp-more-songs {

position:absolute;left:285px;top:130px

}

.jp-progress {

width:425px;position:absolute;left:118px;top:109px;overflow:hidden;height:5px

}

.jp-progress .jp-seek-bar {

background:url(../images/progbar.png) repeat-x;position:absolute;height:5px;left:0;top:0;overflow:hidden;cursor:pointer

}

.jp-progress .jp-seek-bar .jp-play-bar {

background:url(../images/progbar.png) repeat-x 0 -5px;height:5px;display:block;position:absolute;left:0;top:0

}

.jp-controls .jp-play,.jp-controls .jp-pause{

width:55px;height:55px;position:absolute;left:42px;top:84px;display:block;background:url(../images/jplayer-play-pause.png) 0 0 no-repeat

}

.jp-controls .jp-pause {

background-position:0 -55px

}

/\* additional classes \*/

.pl {

padding-left:40px

}

.green {

color:#74be00

}

.fleft,.fl {

float:left

}

.fright,.fr {

float:right

}

.fnone,.fn {

float:none

}

.col-1,.col-2,.col-3 {

float:left

}

.alignleft,.all {

text-align:left

}

.alignright,.alr {

text-align:right

}

.aligncenter,.alc {

text-align:center

}

.wrapper {

width:100%;overflow:hidden

}

a.und,.und a {

text-decoration:none

}

a.und:hover,.und a:hover {

text-decoration:underline

}

.nocolor, .nocolor a {

color:inherit

}

.upc {

text-transform:uppercase

}

.extra-wrap {

overflow:hidden

}

.nopad {

padding:0

}

.nomar {

margin:0

}

.clear {

clear:both;line-height:0

}

.bold {

font-weight:bold

}

/\* custom boxes \*/

.more {

font:14px/35px Arial, Helvetica, sans-serif;color:#74be00;width:113px;text-align:center;display:block;margin:0 auto;background:url(../images/more-sprite.png);text-decoration:none

}

.more:hover {

color:#fff

}

.list a {}

dl.address dd {

text-align:right;clear:both

}

dl.address dd span {

float:left

}

dl.img-box,.img-box-set dl,dl.img-box dd,.img-box-set dl dd {

overflow:hidden

}

dl.img-box dt,.img-box-set dl dt {

float:left

}