

***Даний матеріал наводиться в демонстративних цілях.
Через це файл не відображає повний обсяг роботи. Для
отримання цілісної інформації по цій чи іншій роботі
звертайтеся до менеджерів сайту kursach.in.ua***

ТИТУЛЬНИЙ АРКУШ

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 ОГЛЯД ОСНОВ ВЕБ-ТЕХНОЛОГІЙ ТА ВЕБ-ДИЗАЙНУ	4
1.1 Основи мови розмітки гіпертексту HTML	4
1.2 Структура Web-сторінки	9
1.3 Правила веб-дизайну сторінок	9
РОЗДІЛ 2 ПРОЕКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СТОРІНКИ	16
2.1 Опис засобів розробки об'єкта проектування	16
2.2 Розробка програмного забезпечення.....	16
ВИСНОВКИ	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23

ВСТУП

Актуальність дослідження. На сучасному рівні розвитку інформаційних технологій використання комп'ютера для збереження будь-яких видів інформації стає єдиним засобом, що надає широкі можливості керування інформацією. Важливу роль у процесі отримання інформації відіграє мережа Інтернет. Сьогодні в Україні послугами Інтернет з різною періодичністю користуються близько 9 млн. жителів України.

Зараз дуже актуальною є тема створення web-сайтів. Різні підприємства, навчальні заклади, міністерства мають свою сторінку в Інтернеті. Це найкраща реклама, інформаційний портал для будь-якої організації, це важливий крок компанії до розширення кордонів власного бізнесу та здобуття нової аудиторії. Науковцю, аспіранту, фахівцю творчої професії також важливо мати свою власну WEB-сторінку в Internet, на якій розміщується резюме, звіт про творчі досягнення тощо.

Об'єкт дослідження. Веб-сторінка на тему «кіно».

Предмет дослідження. Створення веб-сторінки за допомогою HTML на тематику «кіно».

Мета дослідження. Розробити власну веб-сторінку за допомогою мови HTML.

Завдання дослідження.

- Розробити програмне забезпечення (власну веб-сторінку);
- Проаналізувати основи мови гіпертексту HTML, структуру веб-сторінки, правила веб-дизайну веб-сторінок;
- Описати засоби розробки об'єкта програмування.

Результатом проектування є набуття нових навичок із розроблення веб-сторінки мовою HTML з тематики кіно.

Курсова робота складається із 34 сторінок.

РОЗДІЛ 1 ОГЛЯД ОСНОВ ВЕБ-ТЕХНОЛОГІЙ ТА ВЕБ-ДИЗАЙНУ

1.1 Основи мови розмітки гіпертексту HTML

HyperText Markup Language (HTML) - мова розмітки гіпертексту - призначена для написання гіпертекстових документів, що публікуються в World Wide Web.

Гіпертекстовий документ - це текстовий файл, що має спеціальні мітки, так звані теги, які згодом розпізнаються браузером і використовуються нам для відображення вмісту файлу на екрані комп'ютера. За допомогою цих позначок можна виділяти заголовки документа, змінювати колір, розмір і накреслення літер, вставляти графічні зображення і таблиці. Але основною перевагою гіпертексту перед звичайним текстом є можливість додавання до вмісту документа гіперпосилань - спеціальних конструкцій мови HTML, які дозволяють клацанням миші перейти до перегляду іншого документа.

Існує два способи створення гіпертекстових документів. Можна скористатися одним з HTML-редакторів, для роботи з якими не потрібно спеціальних знань про внутрішню структуру створюваного документа. Цей спосіб дозволяє створювати документи для WWW без знання мови HTML. HTML-редактори автоматизують створення гіпертекстових документів, позбавляють від рутинної роботи. Проте їхні можливості обмежені, вони сильно збільшують розмір одержуваного файлу і не завжди отриманий з їх допомогою результат відповідає очікуванням розробника.

Альтернативою є створення і розмітка документа за допомогою звичайного редактора plain-тексту (таких, як emacs або NotePad). При цьому способі в текст вручну вставляються команди мови HTML. Створюючи документи таким способом, ви точно знаєте, що робите.

Як вже зазначалося, HTML-документ містить символічну інформацію. Одна

її частина - власне текст, тобто дані, що становлять зміст документа. Інша - теги (markup tags), так звані прапори розмітки, - спеціальні конструкції мови HTML, які використовуються для розмітки документа і керують його відображенням. Саме теги мови HTML визначають, у якому вигляді буде представлений текст, які його компоненти будуть виконувати роль гіпертекстових посилань, які графічні або мультимедійні об'єкти повинні бути включені в документ. Графічна та звукова інформація, включена в HTML-документ, зберігається в окремих файлах. Програми перегляду HTML-документів (браузери) інтерпретують прапори розмітки і відповідним чином відображають на екрані текст і графіку. Для файлів, що містять HTML-документи, прийняті розширення. Htm або. Html.

Прописні і малі літери при записі тегів не розрізняються. У більшості випадків теги використовуються парами. Пара складається з відкриваючого (Start tag) і закриваючого (end tag) тегів. Синтаксис відкриваючого тега:

```
<Імя_тега [атрибути]>
```

Прямі дужки, які використовуються в описі синтаксису, означають, що даний елемент може бути відсутнім. Ім'я закриваючого тега відрізняється від імені відкриваючого лише тим, що перед ним ставиться похила риска:

```
</Імя_тега>
```

Атрибути тега записуються в наступному форматі:

```
ім'я [= "значення"]
```

Лапки при заданні значення аргументу не обов'язкові і можуть бути опущені. Для деяких атрибутів значення може не вказуватися. У закриваючих тегів атрибутів не буває.

Дія будь-якого парного тега починається з того місця, де зустрівся відкриваючий тег і закінчується при зустрічі відповідного закриваючого тега. Часто пару, що складається з відкриваючого і закриваючого тегів, називають контейнером, а частина тексту, обмежену відкриваючим і закриваючим тегом, -

елементом.

Текст може складатися з пропусків, табуляцій, символів переходу на новий рядок, символів повернення каретки, букв, знаків пунктуації, цифр, і спеціальних символів (Наприклад, +, #, \$, @), за винятком наступних чотирьох символів, що мають у HTML спеціальний сенс: <(менше), >(більше), &(Амперсенд) і "(лапки). Якщо необхідно включити до тексту будь-якої з цих символів, то слід закодувати його особливою послідовністю символів.

До спеціальних символів можна віднести і нерозривний пробіл. Використання цього символу - один зі способів збільшити відстань між деякими словами в тексті. Звичайні пропуски використовувати для цих цілей не можна, так як група стоячих поспіль пробілів інтерпретується браузером як один.

Кожна з таких зарезервованих послідовностей починається символом амперсенда (&) і закінчується крапкою з комою (;).

Таблиця 1.1 – Демонстрація послідовності та символу

Послідовність	Символ
<	символ <
>	символ >
&	символ &
"	символ " (лапки)
 	нерозривний пробіл

HTML (Hypertext Markup Language — Мова гіпертекстової розмітки) — це мова опису структури сторінок документів, яка дозволяє звичайний текст формувати в абзаци, заголовки, списки та інші структури, створювати посилання на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну

інформацію. Теги повідомляють браузерам, як форматувати і представляти інформацію на екрані.

Мова гіпертекстової розмітки HTML була запропонована Тімом Бернерсом-Лі у 1989 як один з компонентів технології розробки розподіленої гіпертекстової системи World Wide Web. Ідея гіпертекстової інформаційної системи полягає у тому, що користувач має можливість переглядати документи (сторінки тексту) у найбільш зручному для себе порядку, а не послідовно, як це прийнято при читанні книг. Досягається це шляхом створення спеціального механізму пов'язування різних сторінок тексту за допомогою гіпертекстових посилань.

Мова HTML дозволяє визначити структуру електронного документа з поліграфічним рівнем оформлення. Результуючий документ може містити різноманітні елементи: ілюстрації, аудіо і відео фрагменти. Мова HTML включає розвинені засоби для визначення кількох рівнів заголовків, шрифтових виділень, різних груп об'єктів та багато інших можливостей.

Важливим чинником, який вплинув на розвиток мови HTML, став її вибір за основу для гіпертекстової бази даних звичайного текстового файлу, який можна створювати у будь-якому текстовому редакторі на будь-якій апаратній платформі у середовищі будь-якої операційної системи.

Таким чином, гіпертекстова база даних у концепції WWW – це набір текстових файлів, розмічених мовою HTML, яка визначає форму представлення інформації (розмітка) і структуру зв'язків цих файлів (гіпертекстові посилання).

За основу моделі розмітки документів у HTML прийнята тегова модель. Тегова модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами. Тобто документ HTML є не чим іншим, як звичайним ASCII-файлом з доданими до нього керуючими HTML-кодами (тегами).

Теги HTML-документів в основному є простими і зрозумілими для використання, оскільки вони створені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень.

HTML-тег складається з імені, за яким може слідувати необов'язковий список атрибутів тегу. Текст тегу вміщується у кутові дужки (<I>). Найпростіший варіант тегу – ім'я, вміщене у кутові дужки, наприклад, <HEAD>. Для більш складних тегів характерна наявність різних атрибутів, які можуть мати конкретні значення, визначені для видозмінення функцій тегу.

Атрибути тегу слідують за ім'ям і відділяються один від одного одним або кількома пропусками. Порядок запису атрибутів у тегу не має значення. Значення атрибута слідує за знаком рівняння, який стоїть після імені атрибута. Якщо значення атрибута – одне слово або число, його можна вказати безпосередньо після знаку рівняння, не виділяючи додатково. Решту значень необхідно вміщувати у одинарні або подвійні лапки, особливо якщо вони містять декілька розділених пропусками слів.

Найчастіше HTML-теги складаються з початкового і кінцевого компонентів, між якими розміщуються текст та інші елементи документа. Ім'я кінцевого тега ідентичне імені початкового тегу, але перед ім'ям ставиться коса риска (/) (наприклад, для тегу заголовка <TITLE> закриваючою парою буде </TITLE>). Кінцеві теги не містять атрибутів.

При використанні вкладених тегів їх слід закривати, починаючи з останнього і рухаючись до першого.

Деякі HTML-теги не мають кінцевого компонента, оскільки є автономними елементами. Наприклад, тег зображення , призначений для вставки зображення у документ, не має кінцевого компонента.

Для створення HTML-документа можна застосувати редактор ASCII (зокрема, Блокнот системи Windows). Такі редактори дозволяють вводити HTML-теги, не додаючи до створеного нічого додатково. Створення документа у такому редакторі дозволяє паралельно переглядати результат у програмі-браузері. Інший тип редакторів – візуальні HTML-редактори, наприклад, Microsoft FrontPage. Їх інтерфейс побудований за тим же принципом, що і інтерфейс текстового процесора, такого, як, наприклад, Word. Для роботи з візуальним редактором

можна взагалі не володіти мовою HTML. Недоліком візуальних редакторів є те, що розмір створюваного ними HTML-документа у декілька разів більший, ніж документа, створеного звичайним Блокнотом системи Windows. В умовах низької пропускної здатності вітчизняних мереж цей недолік, який стосується швидкості завантаження сторінки (і, відповідно, вартості часу, який на це витрачається), є досить суттєвим недоліком (файл .htm, створений у WORD, в 4 - 9 разів більший, ніж файл аналогічного змісту, створений програмою Блокнот).

1.2 Структура Web-сторінки

1.3 Правила веб-дизайну сторінок

Британський журнал «Поведінка та інформаційні технології» (Behaviour & Information Technology) колись опублікував статтю канадських психологів, які експериментальним шляхом вивели ту саму заповітну цифру, яка дає нам, розробникам, зрозуміти скільки часу необхідно людині, щоб оцінити плід нашої праці, а точніше 1 сторінку сайту. Раніше передбачалося, що менше, ніж за 500 мс людина не здатна осмислити практично нічого з побаченого на сторінці.

Однак як показала практика, йому досить в 10 разів менше часу, щоб скласти ту чи іншу думку про те веб-ресурсі, на який він потрапив. Ну і само собою, якщо на людину за одну двадцяту секунди ваш сайт не справив приємного враження, швидше за все користувач піде в пошуках іншого ресурсу. Сумна статистика, чи не так?

Дизайн сайту в першу чергу покликаний вирішувати саме цю проблему. Першою і основною метою дизайнера є змусити людину затримати погляд і залишитися на сторінці для подальшого дослідження, а згодом й пошуку конкретної інформації. Люди дуже швидко можуть поставити оцінку чого завгодно, ґрунтуючись виключно на емоційній реакції. І якщо у дизайнера

виходить заманити користувача хоча б одним елементом на сторінці, то останній можливо пробачить інші недоліки, розглядаючи ту саму «родзинку».

Виходячи з практичних спостережень, тієї самої «родзинкою» може бути і відчуття сайту в цілому, його юзабіліті (комплексне розташування елементів призначеного для користувача інтерфейсу, що створює зручну, інтуїтивно-зрозумілу, комфортне середовище). Сучасні люди, які не мають навіть якогось конкретного освіти в області дизайну, прекрасно відчують атмосферу сайту, рівень спілкування з користувачем і загальний сприятливий вид. Виходячи з останніх тенденцій веб-дизайну ми майже завжди з упевненістю можемо сказати, що саме нам не подобається: зловживання так званими фотошопний ефектами минулого покоління «drop shadow» і «bevel emboss», заїжджені кліпарти, які в силу своєї «безкоштовності» і загальнодоступності зустрічаються чомусь аж на кожному білборді на дорозі або банері в інтернеті, перевантаженість інформацією на сторінці. Наповнення сайту повинно бути збалансованим, контент повинен бути оригінальним і унікальним, а візуальні ефекти помірні.

Але саме собою для побудови сприятливого середовища необхідно керуватися певними основними принципами.

У підручника описані такі речі як послідовне розташування елементів по сітці, порядок, симетрія, золотий перетин і т.д. Цим правилам web дизайну дійсно варто слідувати, щоб створити зрозумілу, легко сприймається структуру. Але іноді потрібно і відійти відправив, трохи зламавши стереотипи. Не варто забувати про такі речі, як різноманітність, контраст, акцент.



Рисунок 1.1 Приклад дизайну веб-сторінки №1

Адже тут немає симетрії, але зате є збалансована неправильна форма, з акцентованими елементами. Очам є за що зачепитися.



Рисунок 1.2 - Приклад дизайну веб-сторінки №2

Асиметрія, високий контраст, акцент. Все в одному. Яскраво і привабливо.

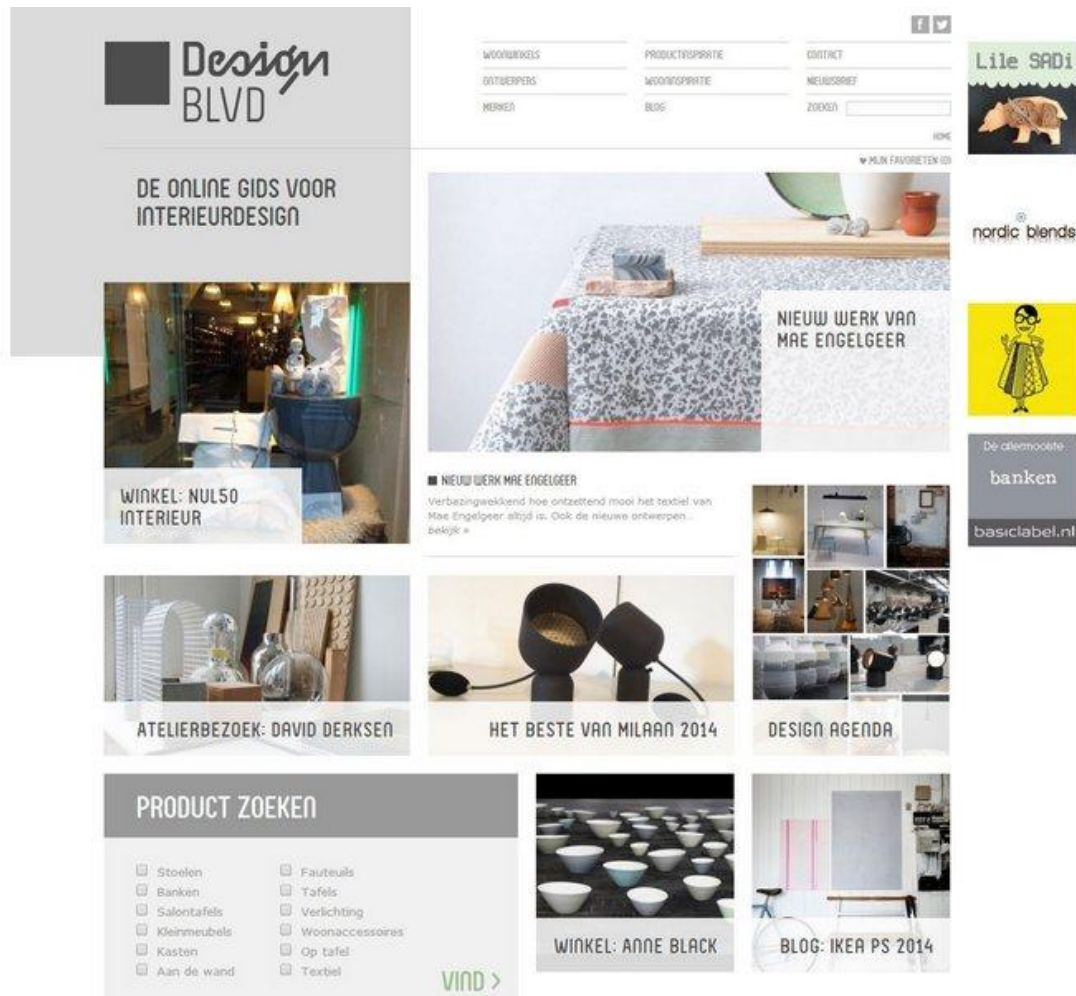


Рисунок 1.3 - Приклад дизайну веб-сторінки №3

На перший погляд трохи хаотично, але якщо придивитися, то і тут є певна схема і порядок. Саме через цю закладену «таємну» схему, сайт не лякає і не створює враження чогось розрізненого і незібраного. Навпаки, він має досить зрозумілу і легку для сприйняття структуру.

1 правило. Максимально використовуйте іконки і зображення для візуального спілкування

Існує ряд іконок, які сприймаються користувачем однозначно. Не варто ними нехтувати: лупа (пошук), будиночок (додому або домашня сторінка), дискета (збереження). Ці образи вже прошиті в мозок практично кожної людини на землі і

цим потрібно користуватися, не намагаючись «винаходити велосипед». Це «дорожні знаки», які читаються і розпізнаються моментально, не замислюючись.

2 правило. Використовуйте тільки унікальний і якісний контент на сайті

Ніколи не використовуйте заїжджені зображення, зображення низької якості, а так само непрофесійно зроблені фотографії. Зображення - це обличчя вашого сайту і перше, що звертає на себе увагу.

3 правило. Колір - не прикраса, а елемент дизайну

Колір - основний елемент, який є ключовим правилом web дизайну і розповідає користувачеві з екранів моніторів, планшетів і телефонів саме те, що ви хотіли йому передати. Колір потрібно підбирати так, щоб він підтримував контент, а не прикрашав його. Чи використовуєте велике фото на сайті? Беріть колірну гамму саме з цього фото, щоб оформити підтримку контенту для створення загальної єдиної атмосфери.

4 правило. Вибирайте шрифти, які вдало підтримають інформацію, яку ви подаєте

Ніколи не варто використовувати більше трьох шрифтів одночасно. Виберіть один, який ідеально підійде для великих заголовків, один для зручного читання тексту і один для закликів, цитат та інших моментів, які повинні свідомо виділятися. Найбільш часто цей набір являє собою шрифт без зарубок для великих обсягів тексту і цікавіший із зарубками або без для оформлення заголовків.

5 правило. Використовуйте тільки унікальний і якісний контент на сайті

Ніколи не використовуйте заїжджені зображення, зображення низької якості, а так само непрофесійно зроблені фотографії. Зображення - це обличчя вашого сайту і перше, що звертає на себе увагу.

Кольорове оформлення документа.

Визначення кольору складових частин документа - один з перших кроків у його створення. Якщо цього не зробити, то будуть використовуватися кольори за замовчуванням, які визначаються установками браузера. Не існує будь-яких правил створення добре збалансованої кольорової палітри документа. Потрібно

лише піклуватися про те, щоб можна було прочитати текст, не відчуваючи при цьому незручностей. При виборі колірної палітри намагайтеся підтримувати високу контрастність тексту і фону та уникайте сусідства областей з близькими кольорами.

Колір може бути заданий назвою (наприклад, green) або шістнадцятковий числом, що визначає колір в моделі RGB. Ця колірна модель базується на визначенні кольору як композиції трьох основних відтінків кольору: червоного (Red), зеленого (Green) і синього (Blue). Кожна компонента задається двозначним шістнадцятковий числом (тобто змінюється від 00 до FF). Потім ці значення об'єднуються в одне число, перед яким ставиться символ # (Більшість сучасних браузерів може розпізнати колір і без вказівки символу #).

Слід також відзначити, що в записі шістнадцяткового значення кольору можна використовувати як великі, так і маленькі латинські літери, наприклад, запис #00FF00 рівнозначна запису #00ff00.

Нижче представлена таблиця 16 стандартних кольорів разом з їх шістнадцятковий кодами.

Колір	Код	Колір	Код
black (чорний)	# 000000	silver (срібний)	# C0C0C0
maroon (темно-бордовий)	# 800000	red (червоний)	# FF0000
green (зелений)	# 008000	lime (вапно)	# 00FF00
olive (оливковий)	# 808000	yellow (жовтий)	# FFFF00
navy (темно-синій)	# 000080	blue (синій)	# 0000FF
purple (фіолетовий)	# 800080	fuchsia (фуксія)	# FF00FF
teal (синьо-зелений)	# 008080	aqua (аква)	# 00FFFF
gray (сірий)	# 808080	white (білий)	# FFFFFFFF

Приклад

Щоб встановити синій колір фону документа потрібно для синьої складової кольору вказати максимальне значення, а інші зробити рівними нулю: `<BODY BGCOLOR="#0000FF">`.

Той факт, що розробник Web-сторінки нічого не знає про комп'ютер, на якому документ буде

проглядатися, накладає додаткові обмеження на використання RGB-моделі. На деяких моніторах неможливо відобразити все розмаїття відтінків. Браузери в цьому випадку скорочують використовуване кількість кольорів, перепризначаючи їх під власні палітри. Використання розробниками гіпердокументів **Web-палітри** є в деякому роді гарантією того, що документ буде виглядати однаково на різних дисплеях.

Web-палітра розпізнає по шість відтінків червоного, синього і зеленого кольору, що в результаті дає 216 можливих значень кольорів ($6 \times 6 \times 6 = 216$). Тому її іноді називають "куб $6 \times 6 \times 6$ ". У таблиці наведено числові значення кольорів з Web-палітри.

Шістнадцяткові	Десяткові	Процентні
00 (самий темний)	0	0%
33	51	20%
66	102	40%
99	153	60%
CC	204	80%
FF (самий світлий)	255	100%

Кольори, RGB-компоненти яких входять в Web-палітру, називають **Web-надійними** кольорами. Серед всіх кольорів, що мають імена, тільки 10 кольорів входять в Web-палітру: aqua, black, blue, cyan, fuchsia, lime, magenta, red, white і yellow.

РОЗДІЛ 2 ПРОЕКТУВАННЯ І ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СТОРІНКИ

2.1 Опис засобів розробки об'єкта проектування

Завдання:

Створити документ HTML. Для цього спочатку створити директорію, у якій ви будете його зберігати. Потім в цій директорії створити простий текстовий файл з розширенням .html. Наприклад, kino.html. Це можна зробити за допомогою будь-якого текстового редактора, що дозволяє створювати і редагувати текстові файли. Таким редактором, що поставляється разом з Windows, є текстовий редактор NotePad (Блокнот). Можна використовувати будь-який інший редактор текстів або спеціалізований редактор.

2.2 Розробка програмного забезпечення

За допомогою наведених нижче команд ми можемо створити сайт з п'яти сторінок. Далше ми створимо горизонтальне меню і сторінки, з'єднані з кожним посиланням. Для того щоб було простіше визначити зони веб-сайту, ми розмалюємо таблиці.

Меню HOME буде посиланням на документ kino.html

Фільми films.html

Скоро skoro.html

Трейлери trailers.html

Новини news.html


```

kino — Блокнот
Файл Правка Формат Вид Справка
<!DOCTYPE HTML PUBLIC «-//W3C//DTD HTML 4.01 Transitional//EN» «http://www.w3.org/TR/html4/loose.dtd»>
<html>

<head>

<meta http-equiv= «Content-Type» content= «text/html; charset=windows-1251 » >

<title> Назва сайту, або заголовок</title>

</head>

<body>

<table width= «900» border= 1 align= center>

<tr>

<td align= center>

<table width= 100% height= 60 border = 1 >

<tr>

<td width= 800 bgcolor= «#cc3333» align = "center" > КІНОАФІША МІСТА ЛУЦЬК </td>

<td width= 100 bgcolor= «#66ff99" > час і дата</td>

</tr>

</table>

</td>

</tr>

<tr>

```

Рисунок 2.1 – Початок коду сайту

Збережемо наявний вище код під іменами:

films.html skoro.html trailers.html kino.html news.html Всі документи HTML повинні бути розміщені в одній директорії. Web-сайт щоб був оформлений належним чином повинен мати в своєму складі з'єднані між собою інтернет-сторінки.

Наприклад kino.html містить посилання на films.html skoro.html trailers.html kino.html news.html

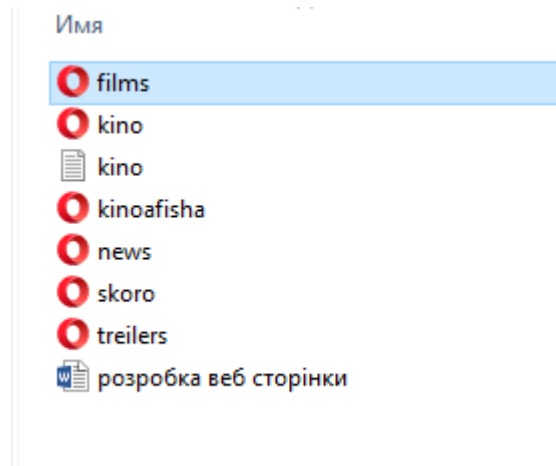


Рисунок 2.2 – Файли в папці після створення

Далі модифікуємо кожен файл нашого сайту, реалізуємо за допомогою тегу `strong` для сторінки `kino`.

📄 kino — Блокнот

Файл Правка Формат Вид Справка

```
<td height= 30 bgcolor= «#cccc99” align = "center">
<a href = «index.html» ><strong> HOME </strong></a>
<a href = «films.html» > ФІЛЬМИ</A>
<a href = «skoro.html» >СКОРО</a>
<a href = «trailers.html» >ТРЕЙЛЕРИ</a>
<a href= «news.html» >НОВИНИ </a>
</td>
</tr>
<tr>
<td valign= top bgcolor= «green” >зона реклами</td>
</tr>
<tr>
<td height= 30 bgcolor = «ff6666” > Додати до обраних </td>
</tr>
</table>
```

Рисунок 2.3 – Продовження коду сайту

```
<a href = “kino.html” ><strong> HOME </strong></a>
<a href = “films.html” > ФІЛЬМИ</A>
<a href = “skoro.html” >СКОРО</a>
<a href = “trailers.html” >ТРЕЙЛЕРИ</a>
<a href= “news.html” >НОВИНИ </a>
для сторінки films.html
<a href = “kino.html” > HOME </a>
<a href = “films.html” > <strong>ФІЛЬМИ</strong></A>
```

СКОРО

ТРЕЙЛЕРИ

НОВИНИ

Надалі ми додаємо розклад сеансів кінофільмів за допомогою коду і гіперпосилання.

```

kino — Блокнот
Файл Правка Формат Вид Справка

<td height= 30 bgcolor = «ff6666" > Додати до обраних </td>

</tr>

</table>
<div class="session-cont">

<table cellpadding="0" cellspacing="0" border="0">
<tbody><tr>
<td class="cinema-room" colspan="2">
</td>
</tr>
<div class="js-cont-cinema-sessions">
    <i class="tip">Для купівлі натисніть на час сеансу</i>

<ul class="times">
<li>
<li>
<h4><a href="/films/neprikasaeme">1+1: Новая история </a></h4>
<p>
<i>20:40</i><a href="https://ticketsbox.com/checkout/12276/" onclick="newbrand(1006);tag_buy();" target="_blank" class="activeEvent" rel="nofollow">21:30</a>
</li>
<li>
<h4><a href="/films/akvamen">Аквамен <i class="ico-3d"></i></a></h4>
<p>
<i>20:45</i><a href="https://ticketsbox.com/checkout/12378/" onclick="newbrand(1006);tag_buy();" target="_blank" class="activeEvent" rel="nofollow">23:30</a>
</li>
<li>
<h4><a href="/films/bamblbi">Бамблби <i class="ico-3d"></i></a></h4>
<p>
    <i>19:50</i>          </p>
</li>
</ul>
</div>
</body>
</html>

```

Рисунок 2.4 - Розклад сеансів кінофільмів за допомогою коду і гіперпосилання.

Останнім кроком буде додавання зображення для нашого сайту на тематику кіно за допомогою команди вставки одного із зображень.

```



```

У результаті ми отримаємо:

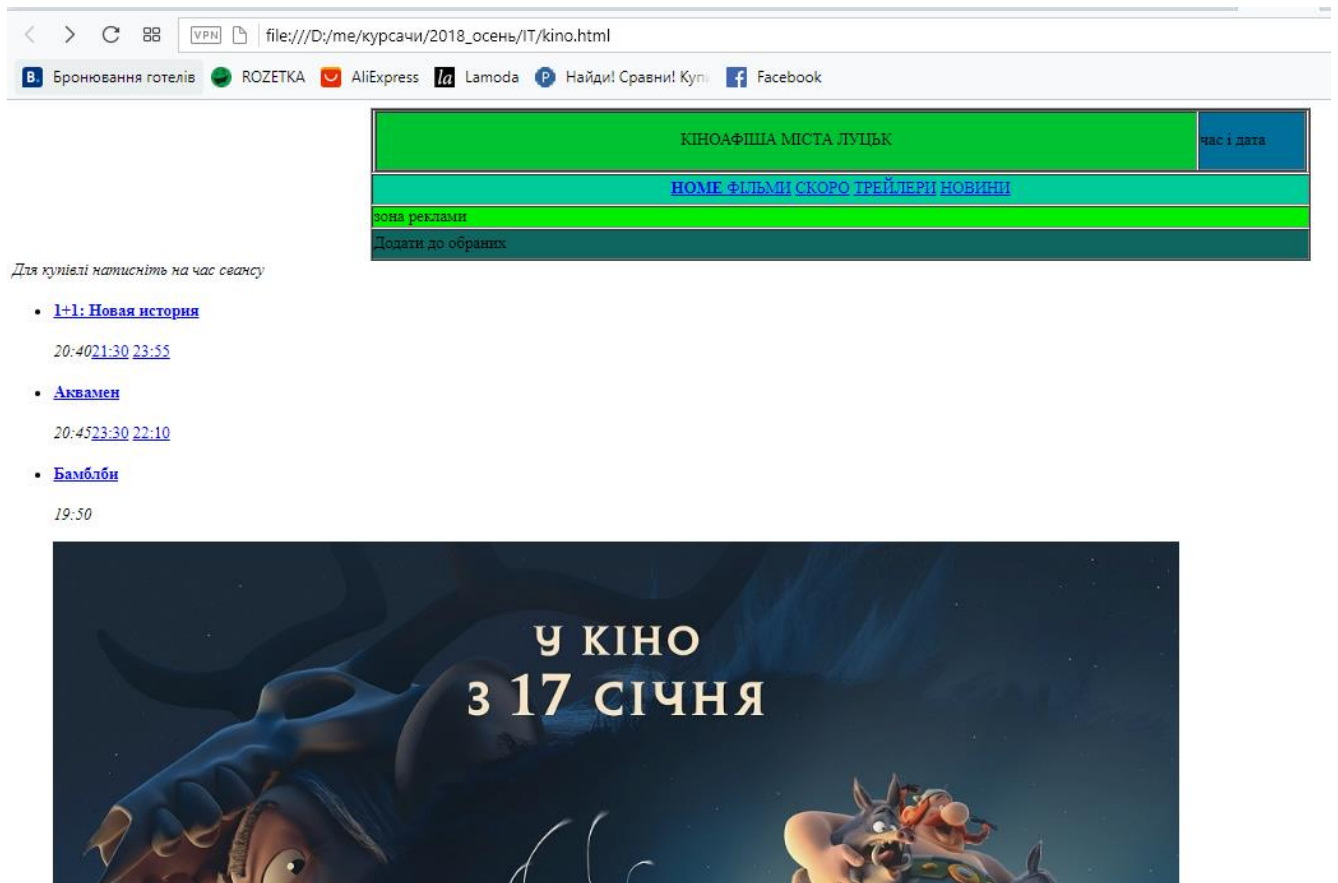


Рисунок 2.5 – Створений приклад сайту на тематику кіно

ВИСНОВКИ

Розроблено програмне забезпечення(власну веб-сторінку) у вигляді сайту-кіноафіши як прикладу.

Проаналізовано основи мови гіпертексту HTML, структуру веб-сторінки, правила веб-дизайну веб-сторінок;

Описано засоби розробки об'єкта програмування. Зокрема у нашому випадку було застосовано засіб HTML.

Сторінку створено в програмі Блокнот(формат *.txt) мовою HTML і збережено у форматі сторінки *.html для відкриття в браузері.

Hyper Text Markup Language (HTML) є стандартною мовою, призначеною для створення гіпертекстових документів в середовищі WEB. HTML- документи можуть бути видимими різними типами WEB-браузеров. Коли документ створений з використанням HTML, WEB-браузер може інтерпретувати HTML для виділення різних елементів документа і первинної їх обробки. Використання HTML дозволяє формувати документи для їх уявлення з використанням шрифтів, ліній і інших графічних елементів на будь-якій системі, що їх переглядає. Основна перевага HTML полягає в тому, що ваш документ може бути проглянутий на WEB - браузерах різних типів і на різних платформах.

Всі теги HTML починаються з "<" (лівої кутової дужки) і закінчуються символом ">" (правої кутової дужки). Як правило, існує стартовий тег і завершуючий тег. Для прикладу приведемо теги заголовка, що визначають текст, що знаходиться усередині стартового і завершуючого тега і заголовок документа:

```
<TITLE> Заголовок документа </TITLE>
```

Завершуючий тег виглядає також, як стартовий, і відрізняється від нього прямим слешем перед текстом усередині кутових дужок. У даному прикладі тег <TITLE> говорить WEB-браузеру про використання формату заголовка, а тег </TITLE> - про завершення тексту заголовка.