**Р Е Ф Е Р А Т**

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 86 с., рис. - 20, бібліографічних джерел - 18, додатків - 2.

Об'єкт дослідження та розробки: веб-сайт підсистеми навчання.

Мета роботи: розробка підсистеми навчання основним мовам розробки веб-додатків.

В дипломній роботі розробляється підсистема навчання засобам розробки веб-додатків, таким як мова розмітки веб-сторінок HTML та каскадні таблиці стилів CSS. Підсистема повинна бути зручною для користувачів, простою в управлінні, інтуїтивно зрозумілою.

Практичне значення, галузь застосування роботи: електронне навчання.

**Ключові слова:** система навчання, веб-сайт, комп’ютерні мережі, Інтернет, веб-дизайн.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А,. м. Сєвєродонецьк, 93400.

**ЗМІСТ**

[ВСТУП](#_Toc479591433) 6

[1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА РОЗРОБКА ТЕХНІЧНОГО ЗАВДАННЯ](#_Toc479591434) 7

1.1 Веб-сайти та мережа Інтернет ……………………………………………7

1.2 Класифікація веб-сайтів …………………………………………………11

1.3Актуальність використання електронного навчання……………………14

1.4 Порівняння з аналогічними системами навчання……………………….18

1.4.1 Пояснювально-ілюстроване навчання……………………………..18

1.4.2 Проблемне навчання………………………………………………..18

1.4.3 Програмоване навчання…………………………………………….19

1.4.4 Комп’ютерне навчання…………………………………………….20

1.5 Технічне завдання…………………………………….…………………...21

[2 ОБГРУНТУВАННЯ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ ПІДСИСТЕМИ НАВЧАННЯ ЗАСОБАМ РОЗРОБКИ ВЕБ-ДОДАТКІВ](#_Toc479591436) 23

[2.1 Вибір середовища розробки](#_Toc479591437) 23

[2.1.1 Система розробки сайтів ASP.NET](#_Toc479591437) 23

[2.1.2 Мова серверного програмування PHP](#_Toc479591437) 28

[2.2 Вибір бази даних](#_Toc479591437) 30

[2.2.1 СКБД Microsoft SQL Server](#_Toc479591437) 30

[2.2.2 СКБД MySQL ………………………………………………………](#_Toc479591437) 31

[3 СТВОРЕННЯ ПІДСИСТЕМИ НАВЧАННЯ ЗАСОБАМ РОЗРОБКИ ВЕБ-ДОДАТКІВ](#_Toc479591440) 33

[3.1 Складові компоненти підсистеми](#_Toc479591441) 33

[3.2 Реалізація головної сторінки підсистеми навчання](#_Toc479591441) 36

3.3 [Навчальні сторінки для мови розмітки HTML](#_Toc479591441) 39

3.4 [Навчальні сторінки для мови розмітки HTML](#_Toc479591441) 41

[4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ](#_Toc479591451) 44

[4.1 Загальні питання з охорони праці](#_Toc479591452) 43

[4.1.1 Правові та організаційні основи охорони праці](#_Toc479591452) 45

[4.1.2 Організаційно-технічні заходи з безпеки праці](#_Toc479591452) 45

[4.2 Аналіз стану умов праці](#_Toc479591452) 46

[4.2.1 Вимоги до приміщень](#_Toc479591452) 46

[4.2.2 Вимоги до організації місця праці](#_Toc479591452) 46

[4.2.3 Навантаження та напруженість процесу праці](#_Toc479591452) 47

[4.3 Виробнича санітарія](#_Toc479591452) 48

[4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу](#_Toc479591452) 48

[4.3.2 Пожежна безпека](#_Toc479591452) 51

[4.3.3 Електробезпека](#_Toc479591452) 51

[4.4 Гігієнічні вимоги до параметрів виробничого середовища](#_Toc479591452) 52

[4.4.1 Мікроклімат](#_Toc479591452) 52

[4.4.2 Освітлення](#_Toc479591452) 53

[4.4.3 Шум та вібрація, електромагнітне випромінювання](#_Toc479591452) 55

[4.4.4 Вентилювання](#_Toc479591452) 55

[4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій](#_Toc479591452) 56

[ВИСНОВКИ](#_Toc479591458) 59

[ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ](#_Toc479591459) 60

[ДОДАТОК А](#_Toc479591460) 61

[ДОДАТОК Б](#_Toc479591460) 82

# ВСТУП

На сьогоднішній день в мережі Інтернет налічується більше 1 мільярда сайтів. Це дуже велике досягнення, так як всього десять років назад цей показник становив близько ста мільйонів сайтів. 67% населення землі користується глобальною павутиною. Мережа Інтернет складається з мільйонів сайтів багатьох різновидів: блогів, новинних порталів, соціальних мереж, Інтернет-магазинів й ще багатьох менш розповсюджених типів сайтів.

В наш час студенти здобувають додаткові знання шукаючи їх в Інтернеті. Це вже не відмінна особливість нашого часу. Один сайт вирішує більшу частину проблем, спрощуючи навчання. Більше не потрібно шукати різні статті серед не потрібної інформації щоб зрозуміти як користатися тими чи іншими засобами.

Досить зробити сайт з простим інтерфейсом у вигляді змісту книги, де кожен початковий студент зможе знайти корисний йому засіб з його описом та наглядним прикладом.

В цьому дипломному проекті проектується та розробляється підсистема навчання засобам створення веб-додатків.

**1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА рОЗРОБка технІчНого ЗАВДАННЯ**

**1.1 Веб-сайти та мережа Інтернет**

Веб-сайт (від англ. website: web - «павутина», «мережа» і site - «місце», букв. «місце, сегмент, частина в мережі»), або просто сайт, - у комп'ютерній мережі об'єднана під однією URL-адресою (доменним ім'ям або IP-адресою) сукупність електронних документів (файлів) приватної особи або організації. За замовчуванням мається на увазі, що сайт розташовується в мережі Інтернет.

Веб-сайти інакше називають інтернет-представництвом людини або організації. Сайти-візитки усе рідше знаходять застосування. Як коментар до посилання може бути сторінка-візитка на повнофункціональному сайті (порталі). Коли говорять «своя сторінка в Інтернет», то мають на увазі цілий веб-сайт або особисту сторінку в складі чужого сайту (портал). Крім веб-сайтів (порталів), у мережі Інтернет також доступні WAP-сайти для мобільних телефонів. Сайт-портал - багатокомпонентна розгалужена структура, скомпонована з функціонально самодостатніх сайтів самостійних організацій або підрозділів корпоративної структури. Наприклад, наявність філій, інститутів академії, журнали й книжковий відділ видавництва, факультети і інститути університету, відділення в країнах великої транснаціональної корпорації, і т. п. У простому уявленні портал може представляти кафедру або факультет, де розміщені сайти фахівців, сайт видань, конференцій, семінарів, новин, наукового підрозділу, бібліотеки й ін.

Споконвічно веб-сайти являли собою сукупність статичних документів, типу - сайт-візитка. У міру розвитку комунікацій, кількість внутрішніх і зовнішніх посилань збільшувалося. Сайт став виконувати не тільки роль довідки, анотації, але й функціонального офісу, новинного або медійного центру. У цей час більшості з них властива динамічність і інтерактивність. Для таких випадків фахівці використовують термін веб-додаток - готовий програмний комплекс для рішення завдань веб-сайту. Веб-додаток входить до складу веб-сайту, але веб-додаток без даних сайтом являється тільки технічно. Оболонку (форму, шаблон) потрібно наповнити й активізувати. Просування сайтів стало ємною індустрією в мережі.

У більшості випадків в Інтернеті одному веб-сайту відповідає одне доменне ім'я. Саме по доменних іменах сайти ідентифікуються в глобальній мережі. Можливі інші варіанти: один сайт на декількох доменах або кілька сайтів під одним доменом. Звичайно декілька доменів використовують великі сайти (веб-портали) щоб логічно відокремити різні види надаваних послуг (mail.google.com, news.google.com, maps.google.com). Нерідкі й випадки виділення окремих доменів для різних країн або мов. Наприклад, google.ru і google.fr логічно являються сайтом Google на різних мовах, але технічно це різні сайти. Об'єднання декількох сайтів під одним доменом характерно для безкоштовних хостингів. Іноді для ідентифікації сайтів в адресі після зазначення хосту стоїть тильда й ім'я сайту: example.com/~my-site-name/, а найчастіше використовується домен третього рівня: my-site-name.example.com.

Апаратні сервера для зберігання веб-сайтів називаються веб-серверами. Сама послуга зберігання називається веб-хостингом. Раніше кожний сайт зберігався на своєму власному сервері, але із збільшенням Інтернету, технологічним поліпшенням серверів на одному комп'ютері стало можливо розміщення безлічі сайтів (віртуальний хостинг). Зараз сервера для зберігання тільки одного сайту називаються виділеними (англ. dedicated).

Особливу роль виконують по розробці й обслуговуванню сайту (порталу) адміністратори (по-іншому - адміни, відповідно до інтернет-сленгу). Якщо виготовлення форми (оболонки) виконує група або дуже кваліфікований фахівець (програміст, Веб-дизайнер, системний адміністратор (відповідно до інтернет-сленгу - сисадмін), координатор, він же адміністратор проекту), то обслуговування й інформаційне наповнення підпорядковане стратегічним завданням і вимагає часто участі команди учасників проекту під управлінням адміна проекту (сайту, порталу). Зараз напрацьовано багато програм і технологій PHP, але це підвищило й вимоги до кваліфікації учасників проекту, у зв'язку із багатопрофільністю розв'язуваних задач.

**1.2 Класифікація веб-сайтів**

По доступності сервісів:

Відкриті - всі сервіси повністю доступні для будь-яких відвідувачів і користувачів.

Напіввідкриті - для доступу необхідно зареєструватися (зазвичай, безкоштовно).

Закриті - повністю закриті службові сайти організацій (у тому числі корпоративні сайти), особисті сайти приватних осіб. Такі сайти доступні для вузького кола людей. Доступ новим людям звичайно дається через т. зв. інвайти (запрошення).

По природі вмісту:

Статичні - весь зміст заздалегідь підготовлюється. Користувачу видаються файли в тім вигляді, у якому вони зберігаються на сервері.

Динамічні - уміст генерується спеціальними скриптами (програмами) на основі інших даних з будь-якого джерела.

По фізичному розташуванню:

Зовнішні сайти мережі Інтернет.

Локальні сайти - доступні тільки в межах локальної мережі. Це можуть бути як корпоративні сайти організацій, так і сайти приватних осіб у локальній мережі провайдера.

За схемою представлення інформації, її об'єму й категорії розв'язуваних задач можна виділити наступні типи веб-ресурсів:

* Інтернет-представництва власників бізнесу (торгівля й послуги, не завжди пов'язані прямо з Інтернетом):
* Сайт-візитка - містить самі загальні дані про власника сайту (організація або індивідуальний підприємець). Вид діяльності, історія, прайс-лист, контактні дані, реквізити, схема проїзду. Фахівці розміщають своє резюме. Тобто докладна візитна картка.
* Представницький сайт - так іноді називають сайт-візитку з розширеною функціональністю: докладний опис послуг, портфоліо, відгуки, форма зворотного зв'язку й т.д.
* Корпоративний сайт - містить повну інформацію про компанію-власника, послуги/продукції, подіях у житті компанії. Відрізняється від сайту-візитки й представницького сайту повнотою представленої інформації, найчастіше містить різні функціональні інструменти для роботи з контентом (пошук і фільтри, календарі подій, фотогалереї, корпоративні блоги, форуми). Може бути інтегрований із внутрішніми інформаційними системами компанії-власника (КІС, CRM, бухгалтерськими системами). Може містити закриті розділи для тих або інших груп користувачів - співробітників, дилерів, контрагентів та ін.
* Каталог продукції - у каталозі присутній докладний опис товарів/послуг, сертифікати, технічні й споживчі дані, відгуки експертів і т.д. На таких сайтах розміщається інформація про товари/послугах, що неможливо помістити в прайс-лист.
* Інтернет-магазин - веб-сайт із каталогом продукції, за допомогою якого клієнт може замовити потрібні йому товари. Використовуються різні системи розрахунків: від пересилання товарів післяплатою або автоматичним пересиланням рахунку по факсу до розрахунків за допомогою пластикових карт.
* Промо-сайт - сайт про конкретну торгову марку або продукт, на таких сайтах розміщається вичерпна інформація про бренди, різних рекламних акціях (конкурси, вікторини, ігри й т.п.).
* Сайт-квест - Інтернет-ресурс, на якому організоване змагання з розгадування послідовності взаємозалежних логічних загадок.

Інформаційні ресурси:

* Тематичний сайт - веб-сайт, що надає специфічну вузькотематичну інформацію про яку-небудь тему.
* Тематичний портал - це дуже великий веб-ресурс, що надає вичерпну інформацію з певної тематики. Портали схожі на тематичні сайти, але додатково містять засоби взаємодії з користувачами й дозволяють користувачам спілкуватися в рамках порталу (форуми, чати) - це середовище існування користувача.

Веб-сервіс - послуга створена для виконання будь-яких завдань у рамках мережі WORLD WIDE WEB:

* Дошка оголошень.
* Каталог сайтів - наприклад, Open Directory Project.
* Пошукові сервіси - наприклад, Yahoo!, Google.
* Поштовий сервіс.
* Веб-форуми.
* Блоговий сервіс.
* Файлообмінний Піринговий сервіс - наприклад, Bittorrent.
* Датахостинговий сервіc - наприклад, Skydrive.
* Датаедіторинговий сервіc - наприклад, Google Docs.
* Фотохостинг - наприклад, Picnik, ImageShack, Panoramio, Photobucket.
* Зберігання відео - наприклад, YouTube, Dailymotion.
* Соціальні Медіа - наприклад, Buzz.
* Комбіновані Веб сервіси (Соціальні мережі) - наприклад, Facebook, Twitter.
* Комбіновані Веб сервіси (Спеціалізовані соціальні мережі) - наприклад, MySpace, Flickr.

**1.2.1 Склад веб-сайтів**

Сторінки сайтів - це файли з текстом, розміченим мовою HTML. Ці файли, будучи завантаженими відвідувачем на його комп'ютер, обробляються браузером і виводяться на його засіб відображення (монітор, екран КПК, принтер або синтезатор мови). Мова HTML дозволяє форматувати текст, розрізняти в ньому функціональні елементи, створювати гіпертекстові посилання (гіперпосилання) і вставляти у відображувану сторінку зображення, звукозапису й інші мультимедійні елементи. Відображення сторінки можна змінити додаванням у неї таблиці стилів мовою CSS або сценаріїв мовою JavaScript.

Сторінки сайтів можуть бути простим статичним набором файлів або створюватися спеціальною комп'ютерною програмою на сервері - так званим движком сайту. Движок може бути або зроблений на замовлення для окремого сайту, або бути готовим продуктом, розрахованим на деякий клас сайтів. Деякі із движків можуть забезпечити власнику сайту можливість гнучкого настроювання структурування й виводу інформації на веб-сайті. Такі движки називаються системами керування вмістом (CMS).

**1.3 Актуальність використання електронного навчання**

Розвиток сучасної освіти в Україні характеризується широкомасштабним використанням комп’ютерної техніки, комп’ютерних комунікацій та інформаційних технологій у навчальному процесі. Перед освітою висувається завдання – формування особистості, конкурентоспроможною, успішною в електронному середовищі. Це зумовило необхідність створення нових моделей навчання, які будуються на використанні мережевих та інформаційно-телекомунікаційних технологій навчання.

Створення середовища, в якому студенти, або учні не були прикуті до стаціонарного комп’ютера, а мали змогу пересуватися з ним, стало можливим з появою портативних комп’ютерів.

Визначаючи інноваційність всіх цих засобів та можливості, які відкривають комп’ютерні технології в забезпеченні наочності й доступності навчального матеріалу, комп’ютерна лабораторія вбудовується у традиційну систему навчання, що орієнтована на трансляцію знань від викладача до студента (учня).

Створення середовища електронного навчання в навчальному закладі дозволяє кардинально змінити парадигму трансляції знань. Ця модель, що орієнтована на принципи «вчитися завжди і всюди», створює умови для реалізації принципів особистісно орієнтованої освіти.

Упровадження електронного навчання (Е – learning) в українську систему освіти є актуальним, дозволяє здійснювати освіту на відстані, в будь-який час, експортувати вітчизняні освітні послуги та ін.

Розвиток Е – learning висуває нову перспективну модель навчання, яка будується на використанні новітніх мультимедійних технологій, Інтернету з метою підвищення якості навчання, полегшення доступу до ресурсів, послуг, а також обміну та спільної роботи на відстані. На світовому ринку освітніх послуг електронне навчання прогресує та одержує розвиток в усіх країнах світу.

Електронне навчання можна використовувати з такими цілями:

- самостійна робота з електронними матеріалами, використовуючи комп’ютер, мобільний телефон і т. д;

- отримання консультацій, проведення нарад, оцінок віддаленого експерта (викладача), можливість дистанційної взаємодії;

- створення розподіленої спільноти користувачів, які ведуть спільну віртуальну навчальну діяльність;

- своєчасна неперервна доставка електронних навчальних матеріалів;

- стандартизація та сертифікація електронних навчальних матеріалів, технологій, дистанційних засобів навчання;

- формування та підвищення інформаційної культури всіх учасників навчального процесу;

- засвоєння, популяризація та передача інноваційних педагогічних технологій, підвищення ефективності діяльності педагогів;

- можливість розвивати навчальні Web-ресурси;

- можливість в будь-який час, з будь-якого місця одержувати сучасні знання;

- доступність одержання освіти особами з фізичними вадами;

Для організації та здійснення Е – learning необхідно визначитися з тим, для кого буде здійснено навчання:

- школярі;

- студенти;

- дорослі слухачі (аспіранти, докторанти, слухачі магістратури).

Відповідно до кожної з цих груп визначаються цілі та завдання навчання, будується графік навчального процесу під час якого кожний має дати відповідь на наступні питання: Що? Як? Навіщо? Чому?

Нові сервіси, які одержали назву соціальних сервісів “Веб 2.0” - друге покоління мережних сервісів в Інтернеті, які, на відміну від першого покоління, дозволяють користувачам працювати з сервісами спільно, обмінюватись інформацією, а також працювати з масовими публікаціями).

Соціальні сервіси Веб 2.0 – це сучасні засоби, мережне програмне забезпечення, яке підтримує групову взаємодію.

Ці групові взаємодії включають:

- персональні дії, думки учасників (ВікіВікі), розміщення медійних файлів (Флікр), фотографії, відеокліпи, радіопередачі;

- створення на основі геоінформаційної системи Google Maрs (карти Google) тематичних сервісів;

- комунікації учасників між собою.

Ураховуючи стрімкий розвиток мережних співтовариств, нині отримали розвиток у педагогічній практиці наступні технології: Делішес, Флікр, енциклопедії колективного авторства на базі технології ВікіВікі.

Так, сервіси Делішес, Флікр можуть бути використанні в педагогічній діяльності наступним чином:

- як джерела навчальних матеріалів;

- як сховище посилань на навчальні матеріали;

- як розв’язання класифікаційних задач;

- як карти знань;

- як засіб для спільної навчальної діяльності.

В процесі вибору програмного забезпечення для здійснення Е-learning необхідно враховувати та дотримуватися наступних характеристик:

- надійність в експлуатації;

- сумісність;

- зручність використання;

- модульність;

- забезпечення доступу.

Для забезпечення ефективності Е – learning ще необхідно враховувати правила використання новітніх ІТ, інтегрованих програм, використання мультимедійних технологій навчання.

Електронне навчання поєднує в собі традиційні та дистанційні методи навчання з можливостями інформаційних технологій, дозволяє значно зменшити витрати на навчання, підвищити ефективність навчального процесу.

Але ж не слід забувати, що електронне навчання – це не єдиний спосіб одержання знань в межах глобального навчального процесу, це лише одна з його складових.

**1.4 Порівняння з аналогічними системами навчання**

**1.4.1 Пояснювально-ілюстративне навчання**

Цей вид називають традиційним. Основними його методами є розповідь, пояснення в поєднанні з демонструванням наочності. Діяльність учнів зводиться до сприймання, запам'ятовування і відтворення навчальної інформації, яке є основним критерієм ефективності. Пояснювально-ілюстративне навчання економить час, зберігає сили вчителів та учнів, полегшує дітям розуміння складної навчальної інформації, забезпечує ефективне управління пізнавальною діяльністю школярів. Водночас воно має істотні недоліки: знання пропонують у готовому вигляді; учні "звільняються" від необхідності самостійно і продуктивно мислити; недостатні можливості індивідуалізації і диференціації навчального процесу та ін.

**1.4.2 Проблемне навчання**

Воно передбачає самостійне оволодіння знаннями у процесі вирішення пізнавальних проблем, розвиток самостійного мислення і пізнавальної активності учнів. Технологія проблемного навчання ґрунтується на чітких алгоритмах, що містять послідовність взаємопов'язаних етапів: створення проблемної ситуації, яка спричинює відчуття розумового Утруднення; аналіз проблемної ситуації, пошук нових елементів знань різними способами (висунення гіпотез); розв'язування проблеми і перевірка одержаних результатів, зіставлення їх з робочою гіпотезою; систематизація та узагальнення здобутих знань і вмінь. Проблемний вид навчання забезпечує самостійне здобуття знань, формування інтересу до пізнавальної діяльності, розвиток продуктивного мислення. Серед його недоліків - значні затрати часу, недостатня ефективність для формування практичних умінь і навичок, слабка ефективність при засвоєнні нового матеріалу, коли самостійний пошук недоступний для учнів.

Основними способами (прийомами) створення проблемних ситуацій є:

1) повідомлення учням інформації, яка містить у собі суперечність;

2) сприймання і осмислення різних тлумачень одного і того самого явища;

3) використання сукупності способів і прийомів, під час якого виникає проблемна ситуація;

4) невідповідність між системою знань, навичок та вмінь учнів і новим фактом, явищем.

**1.4.3 Програмоване навчання**

Основним його принципом е передавання змісту навчального матеріалу невеликими логічно завершеними частинами. Часто навчальну інформацію учні отримують не від вчителя, а з програмованого посібника або дисплея комп'ютера. Основна мета програмованого навчання - удосконалити управління навчальним процесом.

Застосування цього виду забезпечує контроль за кожним кроком учня на шляху пізнання, що уможливлює своєчасне надання допомоги, підтримання інтересу.

Програмоване навчання має такі особливості:

- навчальний матеріал поділено на окремі частини (дози);

- процес навчання передбачає послідовні кроки, які містять формування знання і мисленнєві дії для їх засвоєння;

- кожен крок завершується контролем (запитання, завдання та ін.);

- за умови правильного виконання контрольних завдань учень одержує нову дозу матеріалу і виконує наступний крок навчання;

- у разі неправильної відповіді учень одержує допомогу і додаткові пояснення;

- кожен учень працює самостійно і оволодіває навчальним матеріалом в індивідуальному темпі;

- результати виконання контрольних завдань фіксуються, вони стають відомими для учнів і педагогів;

- педагог є організатором і помічником, консультантом;

- широко застосовуються специфічні засоби навчання (програмовані посібники, тренажери, контролюючі та навчальні машини).

Позитивним у програмованому навчанні є виокремлення основного, істотного в навчальному матеріалі; забезпечення оперативного контролю за процесом засвоєння знань; логічна послідовність у засвоєнні знань; можливість працювати в оптимальному темпі і здійснювати самоконтроль; змога індивідуалізувати навчання. До вад програмованого навчання належать такі: алгоритмізація навчальної діяльності, що обмежує комунікацію між її учасниками; дроблення змісту навчального матеріалу, яке утруднює сприйняття і засвоєння його цілісності, зниження розвивального і виховного потенціалу навчання, що гальмує розвиток творчої активності учнів.

**1.4.4 Комп'ютерне навчання**

Можливості цього виду навчання великі, зокрема подача інформації, управління перебігом навчання, контроль і корекція результатів, виконання тренувальних вправ, накопичення даних про розвиток навчального процесу.

ЕОМ можна ефективно використовувати для підвищення успішності учнів; розвитку загальних когнітивних здібностей (самостійно мислити, володіти комунікативними навичками); автоматизованого тестування, оцінювання та управління педагогічним процесом.

Основою комп'ютерного навчання є створення навчальних програм, виокремлення алгоритмів, системи послідовних дій, які необхідні для повноцінного засвоєння знань і вмінь. Ефективність навчальних програм і комп'ютерного навчання повністю залежить від якості алгоритмів управління розумовою діяльністю. Якісних засобів для комп'ютерного навчання наразі істотно бракує.

**1.5 Технічне завдання**

В результаті виконання даного дипломного проекту повинна бути розроблена підсистема навчання засобам розробки веб-додатків, таким як мова розмітки веб-сторінок HTML та каскадні таблиці стилів CSS. Підсистема повинна бути зручною для користувачів, простою в управлінні, інтуїтивно зрозумілою.

При розробці підсистеми навчання мовним засобам розробки веб-додатків потрібно вирішити наступні завдання:

* дослідити задачі електронного навчання та процеси роботи з ним;
* виконати порівняння з аналогічними системами навчання;
* ознайомитись з існуючим інструментарієм веб-розробки;
* розробити дизайн сайту;
* розробити програми управління HTML-сторінками;
* розробити набір сторінок з навчальним матеріалом.

Головною метою розробки підсистеми повинно бути полегшення навчання та розширення можливостей наочного вивчення засобів веб-розробки. Розроблювана підсистема навчання засобам розробки веб-додатків повинна містити:

1. Головну сторінку, яка повинна бути простою у використанні, відвідавши яку, користувач відразу зрозумів би, чим цей навчальний сайт може бути йому корисним.
2. Зрозуміле і зручне меню навігації по сайту.
3. Засоби демонстрації можливостей мов розмітки та стилізації.

**2 ВИБІР ТА ОБҐРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ВЕБ-САЙТУ НАВЧАННЯ**

**2.1 Вибір середовища розробки**

**2.1.1 Система розробки сайтів ASP.NET[19]**

Microsoft .NET Framework - це платформа для створення, розгортання й запуску Web-сервісів і додатків. Вона надає високопродуктивне, засноване на стандартах, багатомовне середовище, що дозволяє інтегрувати існуючі додатки з додатками й сервісами наступного покоління, а також вирішувати задачі розгортання й використання інтернет-додатків. .NET Framework складається із трьох основних частин - загальномовного середовища виконання (common language runtime), ієрархічної множини уніфікованих бібліотек класів і компонентної версії ASP, так звану ASP.NET.

ASP.NET - це частина технології .NET, використовувана для написання потужних клієнт-серверних інтернет-додатків. Вона дозволяє створювати динамічні сторінки HTML. ASP.NET виникла в результаті об'єднання більш старої технології ASP (активні серверні сторінки) і .NET Framework. Вона містить безліч готових елементів управління, використовуючи які можна швидко створювати інтерактивні web-сайти. Ви також можете використовувати сервіси, що надаються іншими сайтами, прозоро для користувачів вашого сайту. Загалом, можливості ASP.NET обмежені тільки вашою уявою.

Але що, якщо ми бажаємо відобразити на сторінці поточний курс євро або прогноз погоди? Якщо ми написали сторінку HTML учора, сьогодні вона вже застаріє. Отже, ми повинні вміти створювати динамічні сторінки. Динамічне наповнення сторінки - це інформація, зміст якої визначається тим, кому вона призначена, і яка відрізняється від перегляду до перегляду. Воно дозволяє забезпечити двосторонній обмін інформацією - від клієнта до сервера й зворотно.

Динамічними прийнято називати web-сторінки, які перед відправленням клієнту проходять цикл обробки на сервері. У найпростішому випадку це може бути деяка програма, що модифікує запитувані клієнтом статичні сторінки, використовуючи параметри отриманого запиту й деяке сховище даних. Навіть при такій примітивній організації «нерозв'язна» задача з попереднього абзацу знаходить очевидне рішення: досить підготувати всього одну статичну сторінку - шаблон - і перед відправленням сторінки програмно підставляти в нього значення, отримане сьогодні з банку або метеобюро.

Існують мови, здатні динамічно змінювати вміст веб-сторінки. З одного боку, це мови скриптів, що виконуються безпосередньо в клієнта. Приклади скриптових мов - JavaScript і VBScript. Скрипти на цих мовах убудовані в код HTML, що сервер посилає браузеру. Сценарії, що виконуються на стороні клієнта, виділяються тегами. Браузер інтерпретує цей код і показує користувачу результат. Сам код можна переглянути через меню View 🡪 Source браузера. Природно, ці програми не можуть бути великими. Наприклад, якщо потрібно виконати пошук у базі даних, ми не може відправити користувачу весь її вміст. Але скрипти можуть перевірити правильність запиту, введеного у форму, тоді не прийдеться перезавантажувати сервер обробкою неправильних запитів. Деякі програмісти створюють на JavaScript анімаційні ефекти. Але виконуваних на стороні клієнта сценаріїв недостатньо для створення повноцінних динамічних сторінок. Навіть якщо на сторінці використовується JavaScript, анімовані картинки .gif, вона називається статичної.

Динамічна веб-сторінка повинна бути створена миттєво програмою, що виконується на інтернет-сервері. Широко застосовуються механізм шлюзів CGI(Common Gateway Interface). Спочатку користувач одержує статичну сторінку з формою. На сервері знаходяться файли програм, що виконуються, написані, наприклад на C/С++ або Дельфі, які по протоколу HTTP приймають дані із вхідного потоку або зі змінних оточення й записують у стандартний вихідний потік готову сторінку.

Користувачу у відповідь на запит посилає HTML код, що був спеціально сгенерований для нього. Це може бути, наприклад, результат пошуку в пошуковій системі. CGI-скрипти можуть бути написані на інтерпретованій мові (Perl) або навіть скрипті командного рядка. Вхідний і вихідний потоки перепризначуються. На вхід інтернет-сервер приймає дані, введені користувачем. Після обробки отриманих даних, користувачу вертається результуюча сторінка. При виконанні cgi-програма завантажується в пам’ять сервера, а при завершенні - видаляється. Коли 100 клієнтів одночасно звертаються до сервера, у пам'яті створюються 100 процесів, для розміщення коду кожного з яких потрібна пам'ять. Це негативно позначається на масштабованості. Масштабованість - це можливість плавного росту часу відповіді програмної системи на запит з ростом числа одночасно працюючих користувачів.

Для рішення цієї проблеми Microsoft була запропонована альтернатива - ISAPI(Internet Server Application Programming Interface) - розширення й фільтри. Замість файлів, що виконуються, використовуються DLL - бібліотеки. Код DLL знаходиться в пам'яті увесь час і для кожного запиту створює не процеси, а нитки виконання. Всі нитки використовують той самий програмний код. ISAPI-додаток виконується в процесі IIS-сервера. Це дозволяє підвищити продуктивність і масштабованість.

ISAPI-розширення можна створювати в Visual Studio C++, користуючись майстром.

В ISAPI теж є недоліки, що відносяться до розробки. Якщо ми міняємо вихідний код dll, ми повинні його відкомпілювати й помістити у директорію виконання сервера. Але тому що попередній варіант dll перебуває в пам'яті, необхідно зупинити сервер, щоб отримати доступ на зміну файлу. У цей час клієнти не зможуть отримати в сервера жоден документ, і, звичайно, будуть не задоволені.

Виконання коду asp підтримується ISAPI-розширенням сервера. У діалозі конфігурації сервера IIS обумовлюються способи обробки файлів з різними розширеннями. Для обробки URL-адреси з розширенням в установках сервера визначений файл asp.dll. Файли asp відправляються до нього на обробку. На вхід надходить asp, а на виході маємо потік HTML-коду.

В 2000 році на конференції розроблювачів як частину нової технології .NET Microsoft представила ASP+. З виходом .NET Framework 1.0 вона стала називатися ASP.NET.

ASP.NET - це не продовження ASP. Це концептуально нова технологія Microsoft, створена в рамках ідеології .NET. В ASP.NET закладено все, для того, щоб зробити весь цикл розробки веб-додатку більш швидким, а підтримку більш простішою. ASP.NET заснована на обєктно-орієнтованій технології, але зберегла модель розробки ASP: ви створюєте програму й поміщаєте її в директорію, виділену сервером, і вона буде працювати. В ASP.NET з'явилося багато нових функцій, а існуючі в asp значно вдосконалені.

В ASP.NET використовуються мови, які компілюються. Під час компіляції перевіряється синтаксична коректність вихідного тексту. Скомпільований у проміжну мову код виконується швидше, і він буде таким же незалежно від мови, що ми використовуємо. Мови, які компілюються, підтримують строгу типізацію.

Компіляція відбувається на сервері в момент першого звернення користувача до сторінки. Якщо програміст змінив текст сторінки, програма перекомпілюється автоматично. При написанні коду можна використовувати набір компонентів, що поставляють з .NET.

Платформа .NET Framework надає додаткам середовище виконання, сама безпосередньо взаємодіючи з операційною системою. Вище лежить інтерфейс ASP.NET додатків, на якому у свою чергу базуються веб-форми (ASP.NET сторінки) і веб-сервіси. Інтерфейс .NET Framework дозволяє стандартизувати звертання до системних викликів і надає середовище для більш швидкої й зручної розробки. CLR забезпечує єдиний набір сервісів для всіх мов.

ASP.NET використовує технологію доступу до даних ADO.NET, що забезпечує єдиний інтерфейс для доступу до баз даних SQL Server і файлам XML. Крім того, посилена модель безпеки дозволяє забезпечувати захист клієнта й сервера від несанкціонованого доступу.

Попередні версії Visual Studio для проектів ASP.NET вимагали наявності на машині розроблювача сервера IIS. Тепер сервер убудований у середовище розробки.

ASP.NET і Visual Studio 2015 надають інструменти для легкої побудови локалізуємих сайтів, які визначають мову, якій віддає перевагу користувач й посилають йому сторінки на його мові.

Можливість перекомпіляції дозволяє виявити помилки до завантаження сторінок на сервер. Можна не зберігати на сервері вихідні сторінки aspx, тим самим захищаючи свою інтелектуальну власність.

В ASP.NET Core 2.0 убудована технологія автоматичного відновлення кешування баз даних. Дані, отримані з бази, зберігаються на сервері й він не звертається до бази для обробки повторного запиту. При зміні бази даних кеш обновляє свій вміст.

ASP.NET - це технологія, а не мова, і дозволяє програмувати на різних мовах - С#, Visual Basic, J#. У платформі .NET всі мови рівні. Мова С# була спеціально створена для цієї платформи. Програмування C# дозволяє повною мірою використовувати концепції, методи й патерни об’єктно-орієнтованої розробки. Мова Visual Basic наділена майже тими ж можливостями.

**2.1.2 Мова серверного програмування PHP[20]**

PHP (рекурсивний акронім словосполучення PHP: Hypertext Preprocessor) - це поширена мова програмування загального призначення з відкритим вихідним кодом. PHP сконструйований спеціально для ведення Web-розробок і його код може впроваджуватися безпосередньо в HTML.

Замість рутинного виведення HTML-коду командами мови (як це відбувається, наприклад, в Perl або C), скрипт PHP містить HTML з вкрапленнями коду (в нашому випадку, це висновок тексту "Привіт, я - скрипт PHP!"). Код PHP відділяється спеціальними початковим і кінцевим тегами <? Php і?>, Які дозволяють "перемикатися" в "PHP-режим" і виходити з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері і генерують HTML, який надсилається клієнту. Якби у вас на сервері був розміщений скрипт, подібний вищенаведеного, клієнт отримав би тільки результат його виконання, але не зміг би з'ясувати, який саме код його справив. Ви навіть можете налаштувати свій сервер таким чином, щоб звичайні HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

PHP вкрай простий для освоєння, але разом з тим здатний задовольнити запити професійних програмістів. Не лякайтеся довгого списку можливостей PHP. Ви можете швидко почати, і вже протягом декількох годин зможете створювати прості PHP-скрипти. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості PHP

PHP може все. Головна область застосування PHP - написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або відсилати і приймати cookies. Але PHP здатний виконувати набагато більше.

Існують три основні області застосування PHP.

Створення скриптів для виконання на стороні сервера. PHP традиційно і найбільш широко використовується саме таким чином. Для цього вам будуть потрібні три речі. Інтерпретатор PHP (у вигляді програми CGI або серверного модуля), веб-сервер і браузер. Для того щоб можна було переглядати результати виконання PHP-скриптів в браузері, потрібен працюючий веб-сервер і встановлений PHP. Переглянути висновок PHP-програми можна в браузері, отримавши PHP-сторінку, що згенерувала сервером. У разі, якщо ви просто експериментуєте, ви цілком можете використовувати свій домашній комп'ютер замість сервера. За більш докладною інформацією зверніться до глави Поради по установці.

Створення скриптів для виконання в командному рядку. Ви можете створити PHP-скрипт, здатний запускатися без сервера або браузера. Все, що вам потрібно - парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах \* nix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в задачах простий обробки текстів. За додатковою інформацією звертайтеся до глави Використання PHP в середовищі командного рядка.

Створення віконних додатків, що виконуються на стороні клієнта. Можливо, PHP є не найкращим мовою для створення подібних додатків, але, якщо ви дуже добре знаєте PHP і хотіли б використовувати деякі його можливості в своїх клієнтських додатках, ви можете використовувати PHP-GTK для створення таких додатків. Подібним чином ви можете створювати і крос-платформні додатки. PHP-GTK є розширенням PHP і не поставляється разом з основним дистрибутивом PHP. Якщо ви зацікавлені, відвідайте »сайт PHP-GTK.

PHP доступний для більшості операційних систем, включаючи Linux, багато модифікації Unix (такі як HP-UX, Solaris і OpenBSD), Microsoft Windows, macOS, RISC OS, і багато інших. Також в PHP включена підтримка більшості сучасних веб-серверів, таких як Apache, IIS і багатьох інших. В принципі, підійде будь-який веб-сервер, здатний використовувати бінарний файл FastCGI PHP, наприклад, lighttpd або nginx. PHP може працювати в якості модуля або функціонувати як процесор CGI.

Таким чином, вибираючи PHP, ви отримуєте свободу вибору операційної системи і веб-сервера. Більш того, у вас з'являється вибір між використанням процедурного або об'єктно-орієнтованого програмування (ООП) або ж їх поєднання.

PHP здатний генерувати не тільки HTML. Доступно формування зображень, файлів PDF і навіть роликів Flash (з використанням libswf і Ming), створюваних «на льоту». PHP також здатний генерувати будь-які текстові дані, такі, як XHTML і інші XML-файли. PHP може здійснювати автоматичну генерацію таких файлів і зберігати їх в файлової системі вашого сервера замість того, щоб віддавати клієнту, організовуючи, таким чином, серверний кеш для вашого динамічного контенту.

Одним із значних переваг PHP є підтримка широкого кола баз даних. Створити скрипт, який використовує бази даних, - неймовірно просто. Можна скористатися розширенням, специфічним для окремої бази даних (таким як mysql) або використовувати рівень абстракції від бази даних, такий як PDO, або під'єднатися до будь-якої базі даних, що підтримує Відкритий Стандарт З'єднання Баз Даних (ODBC), за допомогою однойменного розширення ODBC. Для інших баз даних, таких як CouchDB, можна скористатися cURL або сокетами.

PHP також підтримує «спілкування» з іншими сервісами через такі протоколи, як LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) і багатьох інших. Крім того, ви отримуєте можливість працювати з мережевими сокетами безпосередньо. PHP підтримує стандарт обміну складними структурами даних WDDX практично між всіма мовами веб-програмування. Звертаючи увагу на взаємодію між різними мовами, слід згадати про підтримку об'єктів Java і можливості їх використання в якості об'єктів PHP.

PHP має багато можливостей по обробці тексту, включаючи регулярні вирази Perl (PCRE) і багато інших розширень і інструментів для обробки і доступу до XML документів. У PHP обробка XML-документів стандартизована і відбувається на базі потужної бібліотеки libxml2, розширивши можливості обробки XML додаванням нових розширень SimpleXML, XMLReader і XMLWriter.

Є ще багато інших цікавих розширень, які можна переглянути як в алфавітному порядку, так і за категоріями. Є ще багато додаткових PECL розширень, які також можуть (а може і ні) бути задокументовані в цьому посібнику, такі як "XDebug.

**2.2 Вибір бази даних**

**2.2.1 Реплікація MySQL**

Реплікація MySQL, якщо її грамотно застосовувати, - інструмент дуже корисними. У той же час, вона може стати справжнім головним болем в разі збою або неправильної настройки. У цьому розділі ми розглянемо основи реплікації MySQL. Для початку візьмемо простеньку систему, а потім перейдемо до базових методів, які ви додасте в свій «арсенал». У цьому розділі розглянуто наступні сценарії використання:

Гарячий резерв. Якщо «впаде» сервер, то зупиниться все: чи не буде можливості обробляти транзакції (може бути, критично важливі), отримувати відомості про клієнтів або витягувати необхідні дані. подібного сценарію потрібно уникнути (майже) будь-яку ціну, так як це може серйозно підірвати бізнес. Найпростіший спосіб: налаштувати додатково вальний сервер, єдина мета якого - бути в гарячому резерві, готовим взяти на себе функції головного сервера, якщо той вийде з ладу.

Створення звітів. Створення звітів на основі зберігаються на сервері даних знизить продуктивність сервера, в деяких випадках, значикові. Якщо при створенні звітів виконується багато фонових завдань, то для цих цілей краще створити додатковий сервер. У заданий час зупиняється реплікація, і на сервері звітів створюється знімок БД, після чого «громіздкі» запити можна адресувати йому, а не головному сервера підприємства. Наприклад, якщо ви зупините реплікацію після завершення останньої транзакції дня, то отримаєте звіт за день, в той час як інша діяльність підприємства буде йти в звичайному ритмі.

Налагодження і аудит. Вивчення виконаних на сервері запитів допоможе вам розібратися в деяких питаннях, наприклад: чи не виникала нехватка продуктивності при виконанні окремих запитів, мала чи місце розсинхронізація сервера через помилку в запиті.

Реплікація - перші кроки.

У цьому розділі представлені складні прийоми, що дозволяють максимально підвищити ефективність і вигоду від реплікації, але почнемо ми з простого прикладу з одним екземпляром реплікації від головного сервера до підлеглого. Для цього не потрібні знання внутрішньої архітектури або механізму виконання реплікації (про це мова піде перед виконанням більш складних сценаріїв).

Налаштування простої реплікації виконується в три етапи:

1. Конфігурація одного з серверів в якості головного.

2. Конфігурація одного з серверів в якості підлеглого.

3. Підключення підлеглого сервера до головного.

Якщо реплікація не планувалася з самого початку, і потрібні параметри конфігурації не були включені до файли my.cnf, то для того щоб виконати кроки 1 і 2, буде потрібно перезавантажити обидва сервера.

**2.2.2 Мова SQL**

Мова SQL (Structured Query Language - структурована мова запитів) являє собою стандартний високорівнева мова опису даних і маніпулювання ними в системах управління базами даних (СКБД), побудованих на основі реляційної моделі даних [1].

Мова SQL був розроблений фірмою IBM в кінці 70-х років. Перший міжнародний стандарт мови був прийнятий міжнародної стандартизує організацією ISO в 1989 р [2], а новий (більш повний) - в 1992 р [3]. В даний час всі виробники реляційних СУБД підтримують з різним ступенем відповідності стандарт SQL92.

Єдиною структурою подання даних (як прикладних, так і системних) в реляційної базі даних (БД) є двовимірна таблиця. Будь-яка таблиця може розглядатися як одна з форм представлення теоретико-множинного поняття відношення (relation), звідси назва моделі даних - реляційна .

У реляційної моделі даних таблиця має такі основні властивості:

1. ідентифікується унікальним ім'ям;

2. має кінцеве (як правило, постійне) нульове кількість стовпців;

3. має кінцеве (можливо, нульовий) число рядків;

4. стовпці таблиці ідентифікуються своїми унікальними іменами і номерами;

5. вміст усіх комірок стовпчика належить одному типу даних (тобто стовпці однорідні), вмістом комірки стовпчика не може бути таблиця;

6. рядки таблиці не мають будь-якої впорядкованості та ідентифікуються тільки своїм вмістом (тобто поняття номер рядка не визначене);

7. в загальному випадку елементи таблиці можуть залишатися порожніми (тобто не містити жодного значення), таке їх стан позначається як NULL.

На вміст таблиць допустимо накладати обмеження у вигляді:

1. вимоги унікальності вмісту кожного осередку будь-якого стовпця і / або сукупності комірок в рядку, що відносяться до кількох стовпців;

2. заборони для будь-якого стовпчика (стовпців) мати порожні (NULL) осередки.

Основними операціями над таблицями є наступні.

1. Проекція - побудова нової таблиці з вихідної шляхом включення в неї обраних стовпців початкової таблиці.

2. Обмеження - побудова нової таблиці з вихідної шляхом включення в неї тих рядків вихідної таблиці, які відповідають певним критерієм у вигляді логічного умови (обмеження)

3. Об'єднання - побудова нової таблиці з 2-ух або більш вихідних шляхом включення в неї всіх рядків вихідних таблиць (за умови, звичайно, що вони подібні до).

4. Декартово твір - побудова нової таблиці з 2-ух або більш вихідних шляхом включення в неї рядків, утворених всіма можливими варіантами конкатенації (злиття) рядків вихідних таблиць. Кількість рядків нової таблиці визначається як добуток кількостей рядків всіх вихідних таблиць.

Перелічені вище 4 операції створюють базис, на основі якого може бути побудована більшість (але не всі) практично корисних запитів на отримання інформації з реляційної БД. Примітка. Набір операцій буде повним, якщо доповнити його операціями перетину і віднімання. Однак в даному посібнику реалізація цих операцій в мові SQL не розглядається.

Крім перелічених вище в мові SQL реалізовані операції модифікації вмісту рядків таблиці і поповнення таблиці новими рядками (що теоретично може розглядатися як операція об'єднання), а також операції управління таблицями.

Розглянуті вище операції над таблицями реляційної БД володіючи функціональної повнотою, будучи реалізовані на практиці в своєму чистому канонічному вигляді, як правило, вкрай неекономічні (в першу чергу це стосується комбінації операцій обмеження і декартового твори). Розробники реальних реляційних СУБД вдаються до всіляких прийомів і хитрощів для мінізаціі обчислювальних витрат (в першу чергу, машинного часу) при виконанні цих операцій. Загальним способом, що знайшли відображення в мові SQL, підвищення ефективності виконання запитів в СУБД є імпользованіе ключів індексів.

Індексом називається прихована від користувача допоміжна керуюча структура, що забезпечує прямий (або квазі прямий) метод доступу до рядків таблиці, що дозволяє виключити послідовний перегляд всіх рядків таблиці для виявлення відповідальних певним критерієм пошуку. Індекси неявним чином (приховано від користувача) автоматично створюються для всіх ключів таблиці.

В даний час найбільшого поширення набули реляційні SQL СУБД двох груп:

1. потужні великі комерційні СУБД, орієнтовані на зберігання величезних обсягів інформації (від гігабайт);

2. мобільні компактні вільно поширювані (в тому числі і в вихідних кодах) СУБД, використання яких виправдано і для БД об'ємом всього лише в десятки кілобайт.

Найбільш відомими СУБД першої групи є:

* Sybase SQLserver фірмы[Sybase, Inc.](http://www.sybase.com/);
* Oracle фірми [Oracle Corporation](http://www.oracle.com/);
* Ingres фірми [Computer Associates International](http://www.cai.com/);
* Informix фірми [Informix Corporation](http://www.informix.com/).

До найбільш популярних СУБД другої групи відносяться:

* PostgreSQL організації [PostgreSQL](http://www.postgresql.org/);
* microSQL фірми [Hughes Technologies Pty. Ltd.](http://www.hughes.com.au/);
* mySQL фірми [T.C.X DataKonsult AB](http://www.tcx.se/).

**3 СТВОРЕННЯ ПІДСИСТЕМИ НАВЧАННЯ ЗАСОБАМ РОЗРОБКИ ВЕБ-ДОДАТКІВ**

**3.1 Складові компоненти підсистеми**

Створена підсистема навчання засобам розробки веб-додатків в себе включає:

1. Просту в використанні головну сторінку, з допомогою якої користувач використовує всі можливості навчального сайту засобам веб-розробки.

2. Зрозуміле і зручне меню навігації по навчальному сайту.

Зовнішній вигляд сайту нагадує електронну книгу. Сайт використовує фрейми:

<frameset rows="22%, 78%">

<frame src="1.html">

<frameset cols="30%, 30%, 40%">

<frame src="2.html">

<frame name="a">

<frame name="b">

</frameset>

</frameset>

За допомогою їх користувачі постійно мають доступ до змісту сайту.

Головна сторінка поділена на чотири частини. Верхня частина, яка представлена рис 3.1, містить логотип сайту.

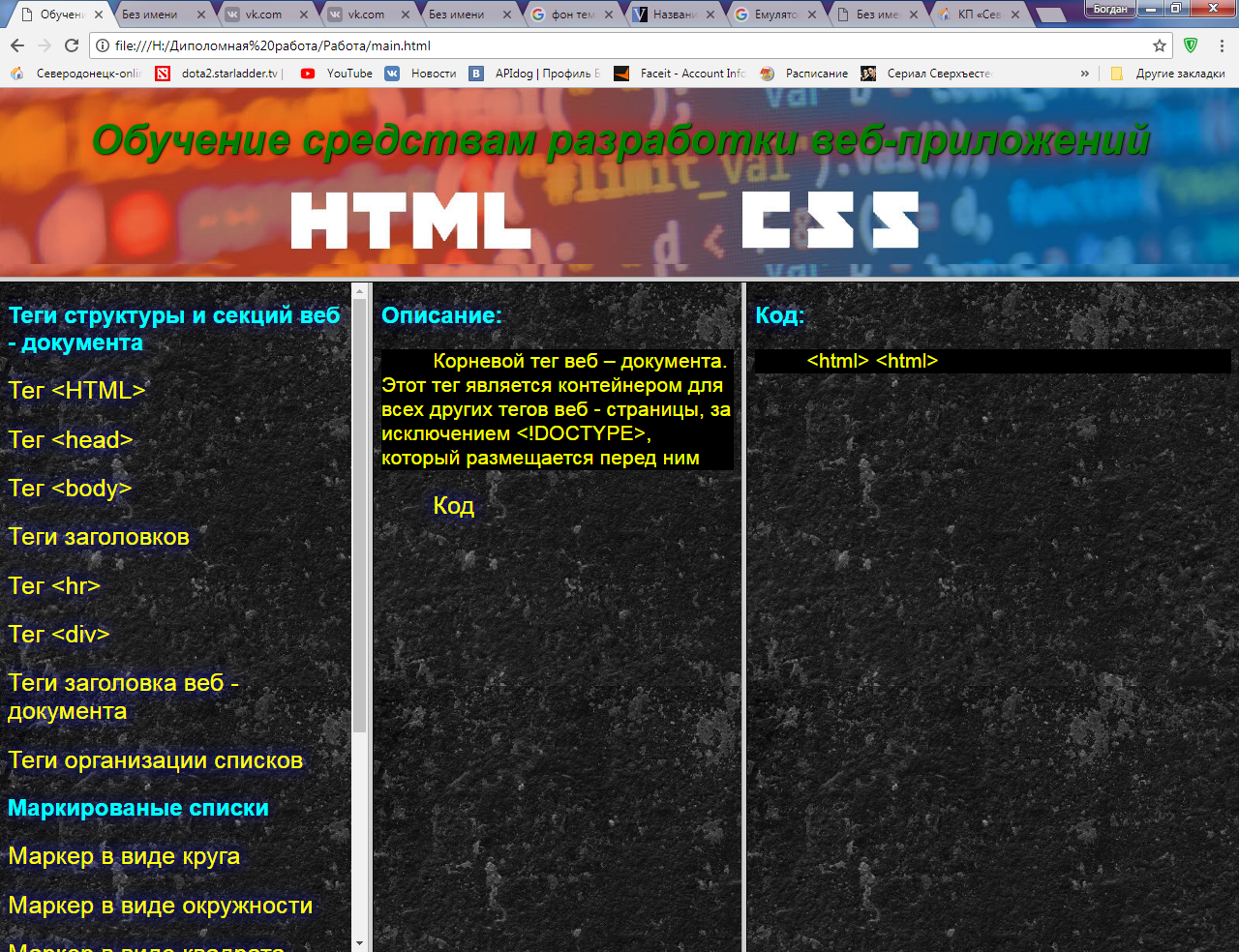


Рисунок 3.1 – Логотип сайту

Ліва частина головної сторінки, показаної на рис. 3.2, - це зміст сайту та навігаційне меню, з допомогою якого ми обираємо необхідний нам розділ навчання.

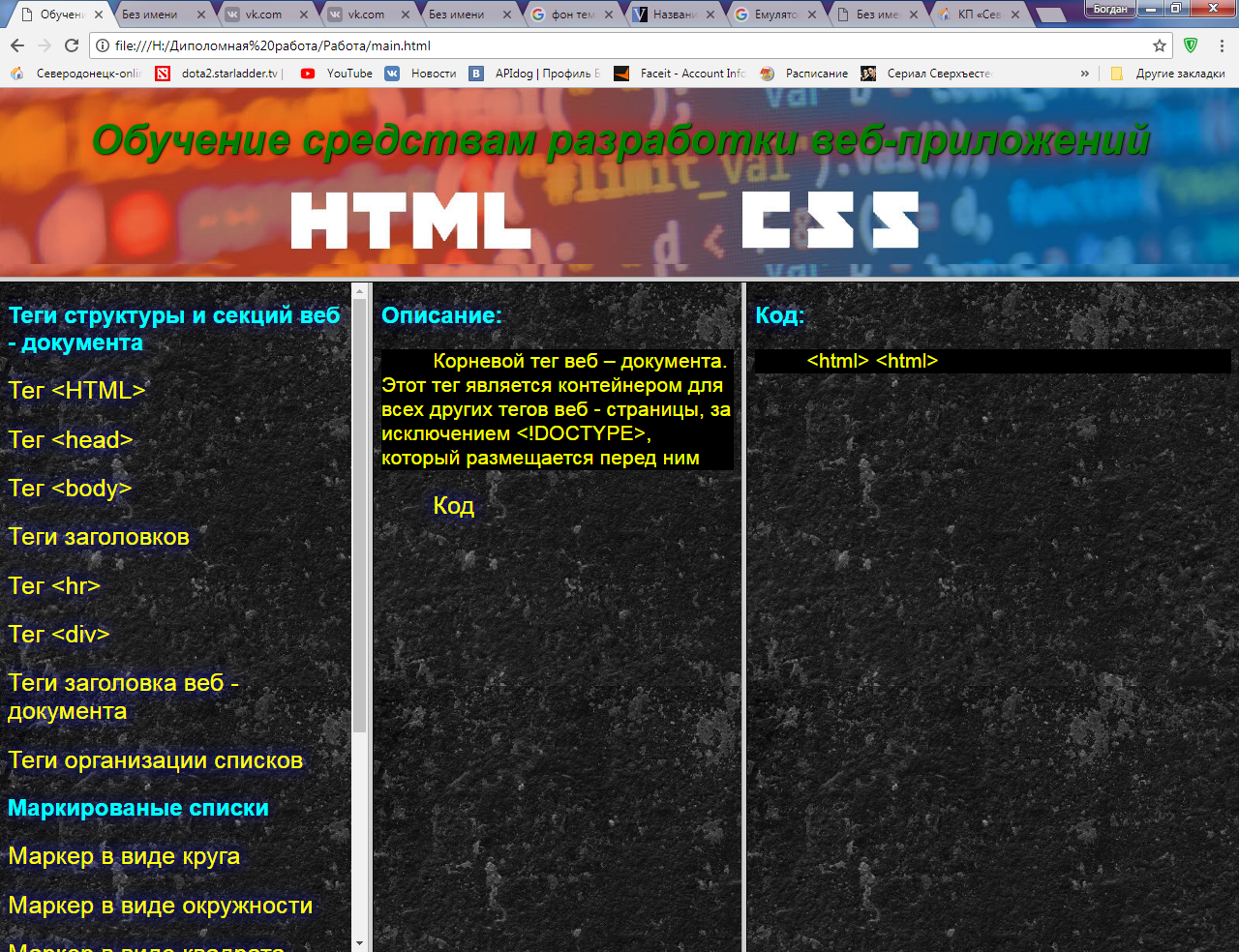


Рисунок 3.2 Навігаційне меню

Центральна частина сайту (рис. 3.3) - це відділ опису вивчення програмного засобу, обраного у навігаційному меню.

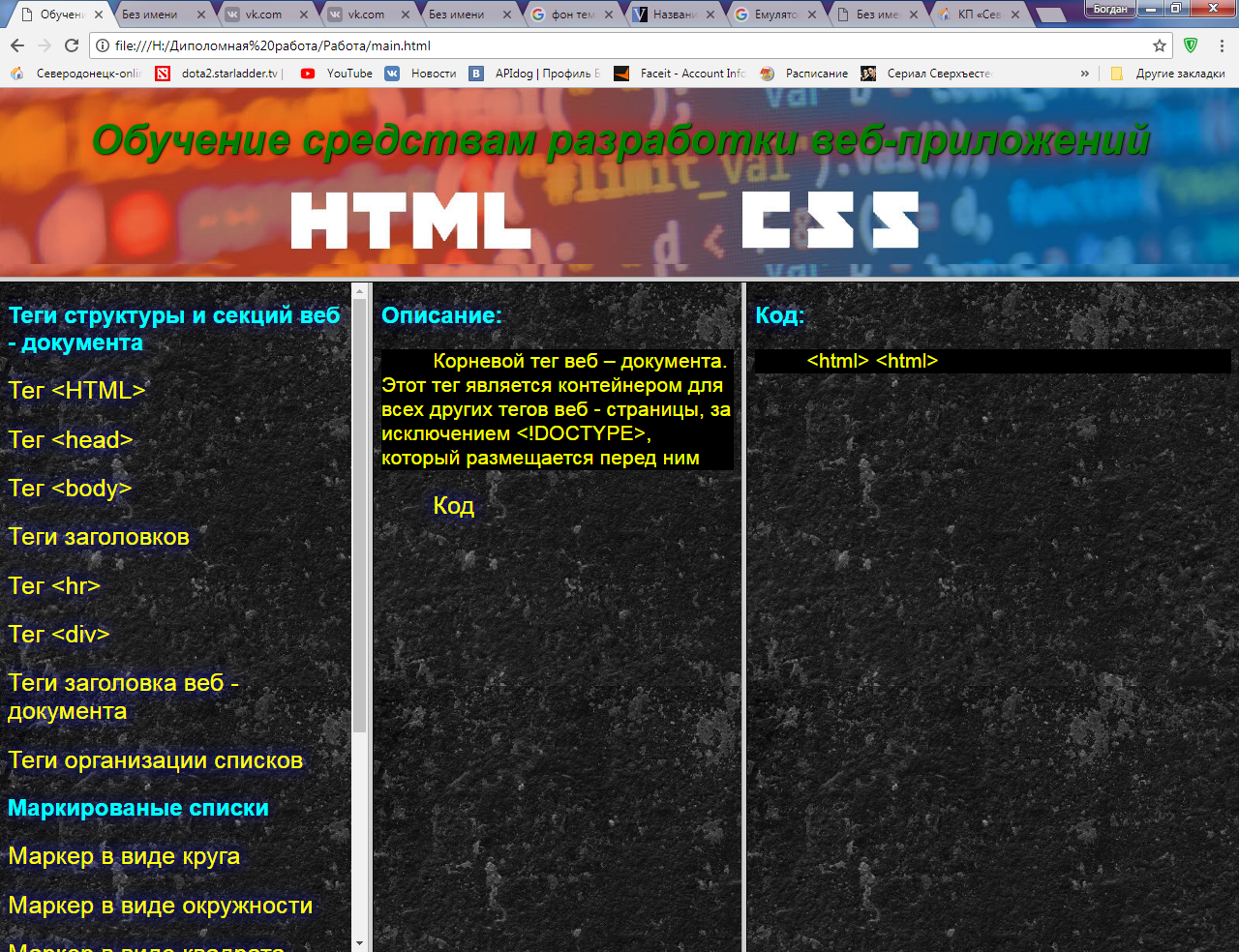


Рисунок 3.3 Відділ опису обраного засобу

Остання частина, яка розміщена справа, створена для демонстрації роботи обраного засобу (рис 3.4), або якщо продемонструвати неможливо деякі засоби, тоді цей відділ відображає код обраного засобу (рис 3.5).

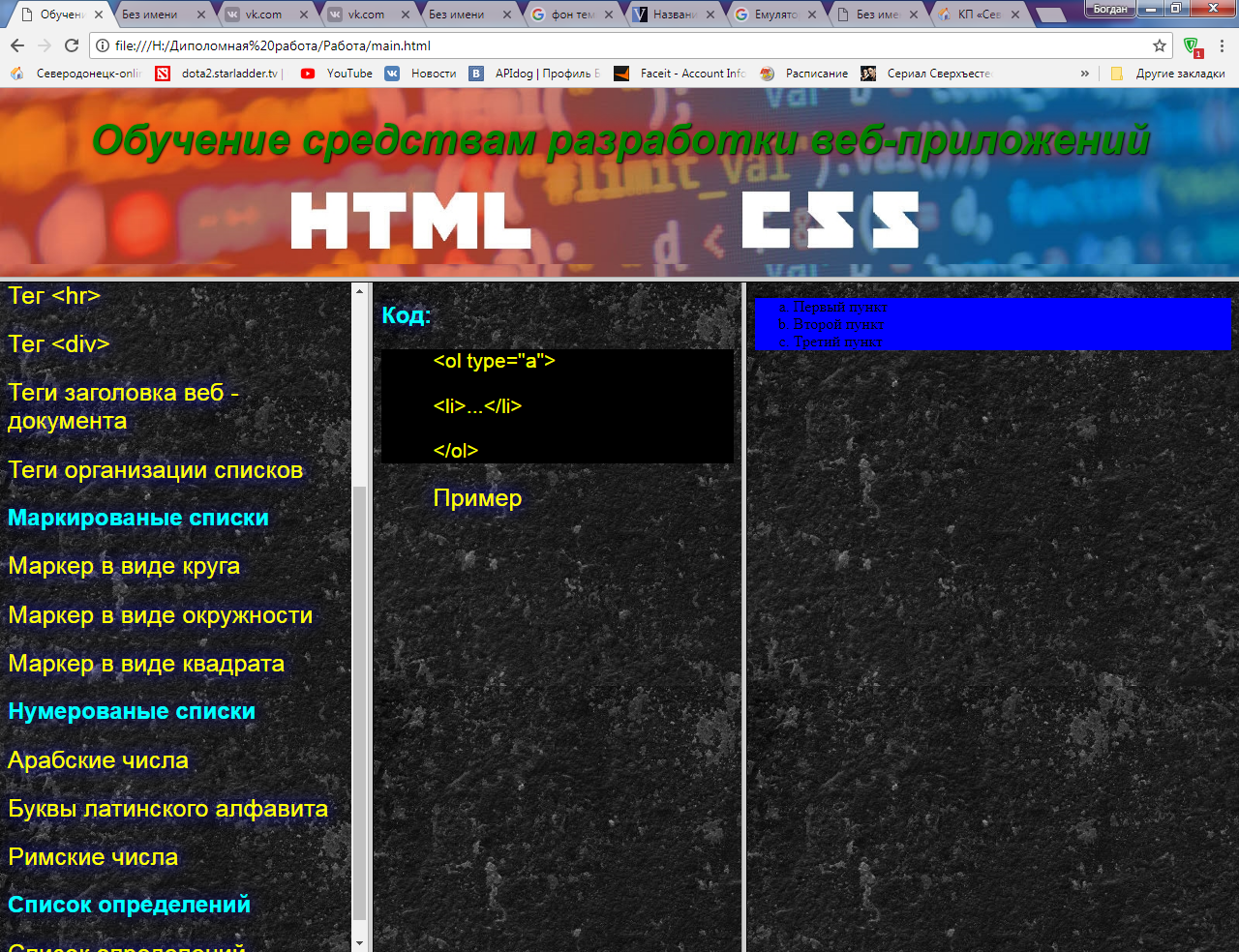


Рисунок 3.4 Демонстрація можливостей коду

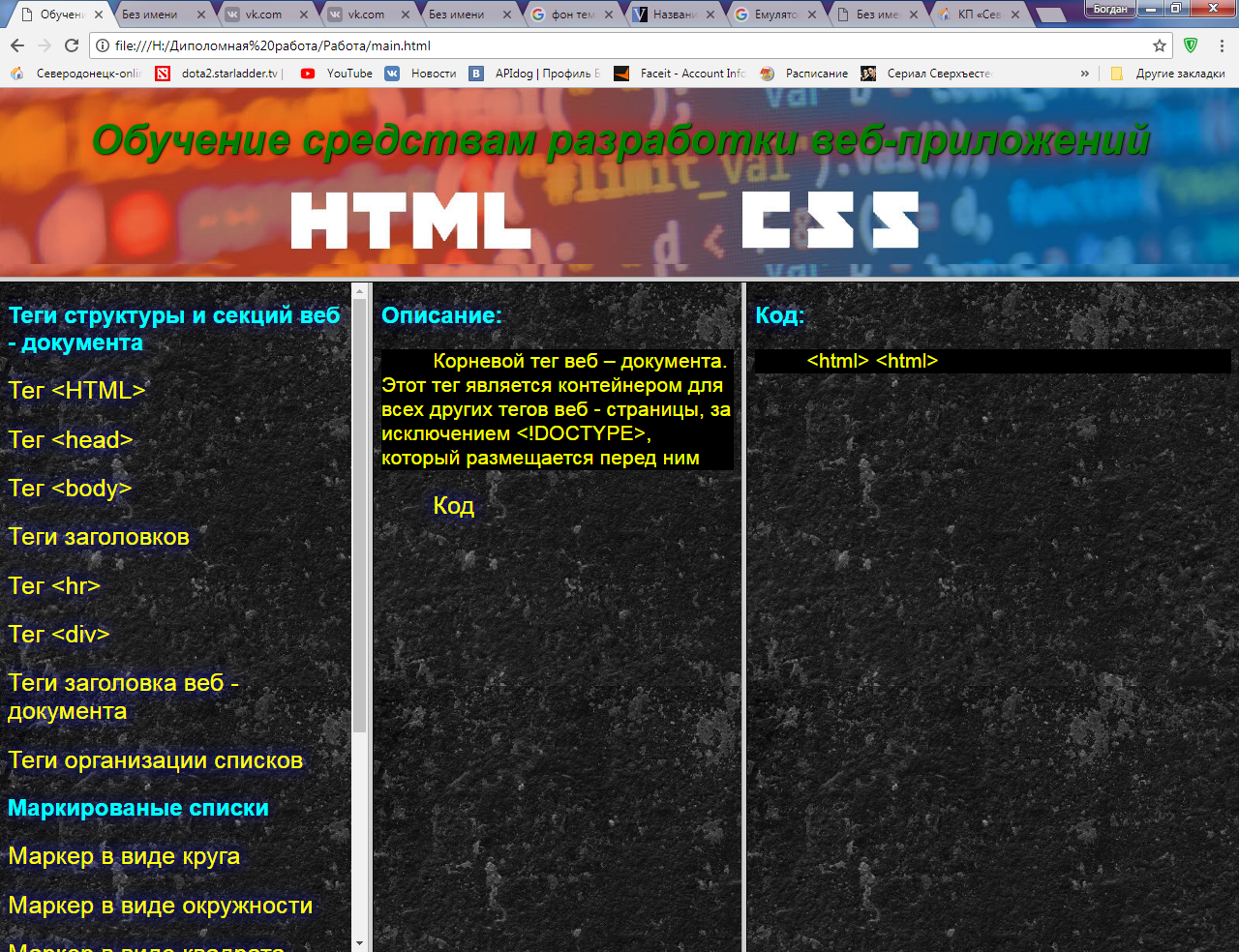


Рисунок 3.5 Код обраного засобу

* 1. **Реалізація головної сторінки підсистеми навчання**

Нижче наведений лістинг головної серверної програми-диспетчера управління формуванням головної та навчальними сторінками Router.php:

**//** Файл Router.php

<?php

/\*\*

\* Класс Router

\* Компонент для работы с ответами

\*/

class Router

{

/\*\*

\* Свойство для хранения таблиц с ответами

\* @var array

\*/

private $routes;

/\*\*

\* Конструктор

\*/

public function \_\_construct()

{

// Путь к файлу с ответами

$routesPath = ROOT . '/config/routes.php';

// Получаем ответы из файла

$this->routes = include($routesPath);

}

/\*\*

\* Возвращает строку запроса

\*/

private function getURI()

{

if (!empty($\_SERVER['REQUEST\_URI'])) {

return trim($\_SERVER['REQUEST\_URI'], '/');

}

}

/\*\*

\* Метод для обработки запроса

\*/

public function run()

{

// Получаем строку запроса

$uri = $this->getURI();

// Проверяем наличие такого запроса в массиве ответов (routes.php)

foreach ($this->routes as $uriPattern => $path) {

// Сравниваем $uriPattern и $uri

if (preg\_match("~$uriPattern~", $uri)) {

// Получаем внутренний путь из внешнего согласно правилу.

$internalRoute = preg\_replace("~$uriPattern~", $path, $uri);

// Определить контроллер, action, параметры

$segments = explode('/', $internalRoute);

$controllerName = array\_shift($segments) . 'Controller';

$controllerName = ucfirst($controllerName);

$actionName = 'action' . ucfirst(array\_shift($segments));

$parameters = $segments;

// Подключить файл класса-контроллера

$controllerFile = ROOT . '/controllers/' .

$controllerName . '.php';

if (file\_exists($controllerFile)) {

include\_once($controllerFile);

}

// Создать объект, вызвать метод (т.е. action)

$controllerObject = new $controllerName;

/\* Вызываем необходимый метод ($actionName) у определенного

\* класса ($controllerObject) с заданными ($parameters) параметрами

\*/

$result = call\_user\_func\_array(array($controllerObject, $actionName),

$parameters);

// Если метод контроллера успешно вызван, завершаем работу

// роутера

if ($result != null) {

break;

} else {

//404

// echo '<script

type="text/javascript">window.location.href="/";</script>';

echo '<h1>404</h1>';

echo "<a href='/'>Main page</a>";

}

}

}

}

}

Головна сторінка сайту, яка формується серверними програмами, написаними на мові PHP, містить в собі описані раніше шапку з логотипом, навігаційне меню зі змістом сайту, відділ опису обраного засобу та відділ демонстрації можливостей обраного засобу (рис. 3.6):

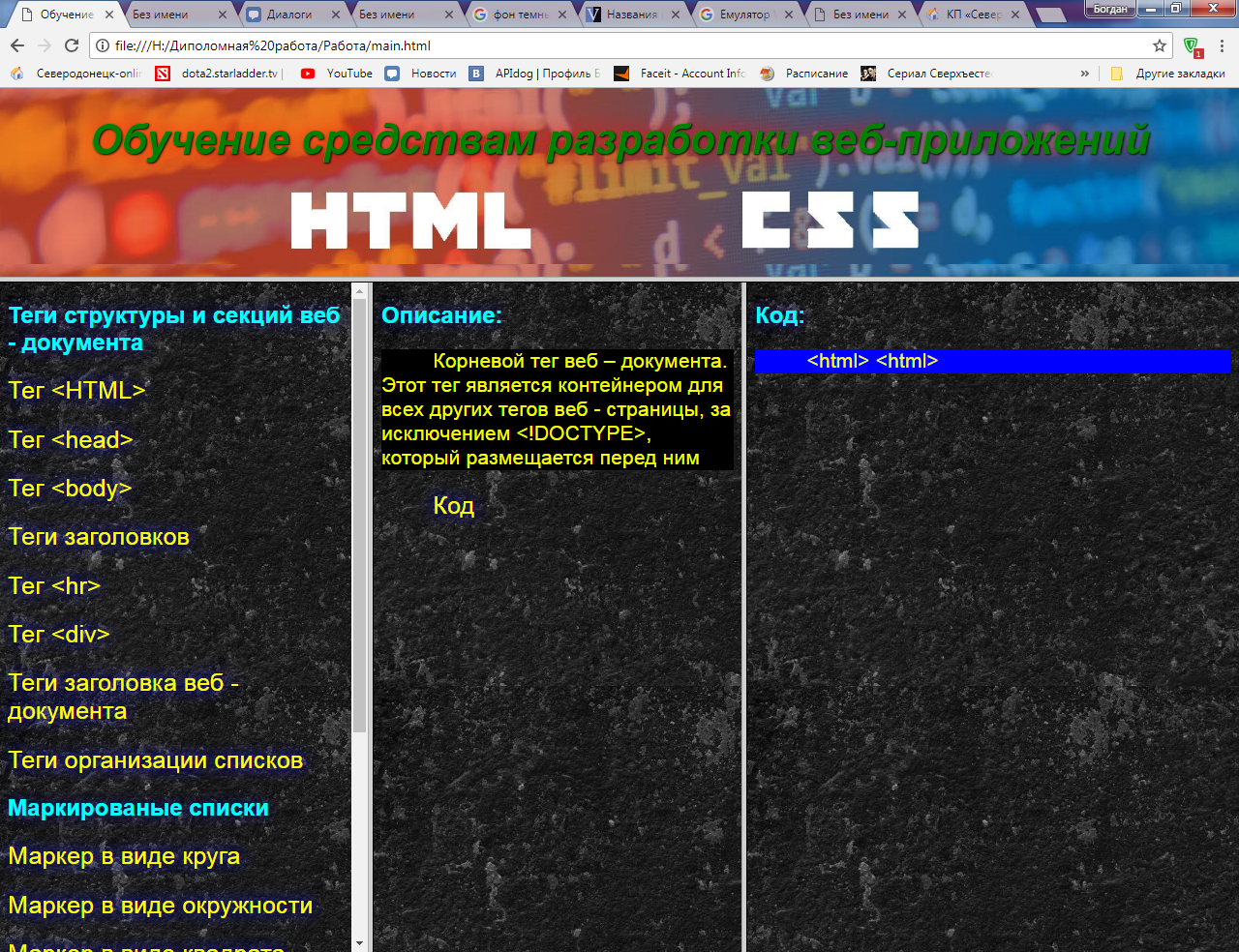


Рисунок 3.6 – Приклад використання головної сторінки підсистеми навчання

На даній сторінці розташовані три активні колонки, у першій відображається зміст навчального курсу який надає можливість постійно пересуватись до необхідної інформації, у другій колонці при першому відвідуванні сайту відсутня інформація, а після вибору користувачем потрібного йому посилання з’являється опис засобу програмування, третя колонка виконує роль демонстрації вигляду програмного коду, також є можливість наглядно побачити роботу коду.

**3.3 Навчальні сторінки для мови розмітки HTML**

Нижче на рис. 3.7 представлений базовий веб-документ, що містить основні теги мови розмітки HTML.

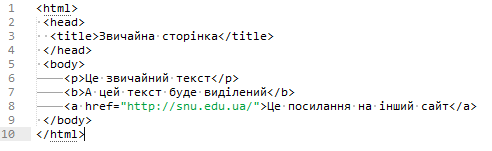


Рисунок 3.7 – Основні теги HTML-сторінки

В якості прикладу, нижче наведений лістинг навчальної сторінки для вивчення тегу <body>;

// Файл body.html

<!DOCTYPE html>

<html>

<head>

<title>Тіло документа</title>

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

</head>

<body>

<h2>Опис: </h2>

<div>

<p> Тіло документа. Обмежує початок і кінець вмісту (контенту) документа.</p>

</div>

<p><a href="body1.html" target="b">Код</a></p>

</body>

</html>

Лістинги інших навчальних сторінок для вивчення мови розмітки HTML містяться в додатку А.

* 1. **Навчальні сторінки для каскадних таблиць стилів CSS**

Коли браузер відображає документ, він повинен об'єднати інформацію про стилі з вмістом документу. Документ обробляється в два етапи:

1. HTML і CSS перетворюється браузером в Document Object Model. DOM представляє документ в пам'яті комп'ютера. Він поєднує стиль документу з його вмістом.
2. Відображається у браузері вміст DOM.

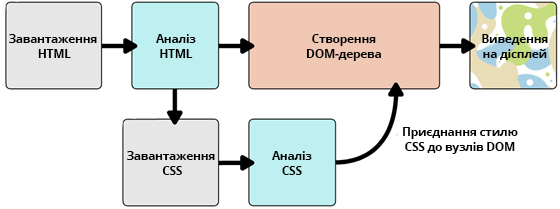


Рисунок 3.8 – Взаємодія таблиць стилів CSS елементами гіпертекстової розмітки

Document Object Model структуру має деревоподібну. В структурі дерева DOM стає вузлом кожен елемент, атрибут і фрагмент тексту в мові розмітки. Деякі елементи є батьками дочірніх вузлів, а дочірні вузли - брати і сестри. Вузли визначаються їх відношенням до інших вузлів DOM.

Розуміння DOM допомагає розробляти, налагоджувати і підтримувати CSS, тому що DOM - це місце, де зібрані таблиці стилів CSS і сам вміст документа.

Нижче наведений лістинг частини файлу стилів, використовуваного при формуванні навчальних сторінок підсистеми навчання:

// Файл style.css

body {

background-image: url(img/1.jpg);

}

p {

font-family: arial;

color: #F6FF00;

font-size: 16pt;

text-indent:2.5em;

}

h1,h2,h3,h4,h5,h6 {

font-family: Helvetica;

color: #00FFFF;

text-shadow: 0 0 1em #000080;

}

a {

font-family: arial;

color: #FFFF00;

font-size: 19pt;

text-shadow: 0 0 1em #0000FF;

text-decoration: none;

}

a:hover {

text-shadow: 0 0 15px #ff7900, 0 0 15px #ff7900, 0 0 15px #ff7900;

}

**ВИСНОВКИ**

В результаті виконання даного дипломного проекту розроблена підсистема навчання мовам розробки веб-додатків. Підсистема є зручною для користувачів, простою в управлінні, інтуїтивно зрозумілою.

При розробці підсистеми навчання засобам розробки веб-додатків були вирішити наступні завдання:

* досліджені задачі електронного навчання та процеси роботи з ним;
* виконане порівняння з аналогічними системами навчання;
* розроблений дизайн веб-сайту навчання;
* розроблені програми управління HTML-сторінками;
* розроблений набір сторінок з навчальним матеріалом.