ЗМІСТ

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ5

РОЗДІЛ 2.ВИКОРИСТАНІ ТЕХНОЛОГІЇ6

2.1. Front-end частина.HTML, CSS7

2.2. Back-end частина. Node.js (Express). JSON10

РОЗДІЛ 3. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ13

РОЗДІЛ4. ПРОГРАМНА РЕАЛІЗАЦІЯ22

ВИСНОВКИ26

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ27

ВСТУП

Значення інтернету та веб-сайтів важно переоцінити. Майже кожна компанія чи будь який бізнес-проект має своє представлення в інтернеті у вигляді веб-сайту.

Відразу ж після виходу на простори всесвітньої павутини, інтернет сайт стає доступним людям всього світу

В рамках веб-сайту існує можливість донесення інформації до мільйонів оглядачів (партнерів, клієнтів, акціонерів, інвесторів) про новітні продукти, події, послуги і документи.

Відвідуваність середнього корпоративного веб-сайту складає більш 100-і гостей в день, які здійснюють пошук всілякої інформації в мережі, або 2500 гостей за місяць, з врахуванням вихідних і урочистих днів.

Згідно статистики, продаж товарів за допомогою мережі інтернет збільшуються на 70% щороку!

Інтернет-сайт є інструментом, який володіє фактично необмеженим простором для розміщення інформаційних даних, також і необмеженими здібностями для маркетингу. На веб-сайті підприємства існує можливість представлення великого об’єму інформація про товари, послуги, новини і актуальні, на сьогоднішній день, пропозиції.

На сайті можна розташувати каталог продукції або послуг з цінами і детальним описом. Веб-сайт дає можливість провести презентацію новітніх продуктів і послуг. Також в мережі інтернет можуть бути розташовані різні рекламні матеріали, які клієнти можуть зберегти і надалі роздрукувати.

Професійний веб-сайт пропонує такі засоби спілкування, як форуми, чати, відео конференції. Обмін думками із співробітниками може зіграти величезну роль в підвищенні кваліфікації ваших працівників. В рамках веб-ресурсу можна організувати живу відео-конференцію співробітників, які знаходяться в різних частинах світу.

Ще однією перевагою веб-сайтів є дистанційне навчання. На веб-сайті можуть бути розташовані різні учбові матеріали. Знаходячись у відкритому доступі, вони істотно скоротять витрати на дистанційне навчання службовців організації. Портали по збільшенню кваліфікації нададуть вашому підприємству максимально повні і комфортні програми дистанційного навчання, створені спеціально для вашого бізнесу і режиму роботи службовців.

У інтернет мережі не буває перерв на обід або вихідних днів, створений сайт не потребує свят або карантину — він працездатний постійно, 365 днів в році. Це означає, що ви не впустите клієнта, якому будуть потрібні послуги вашої компанії або інформація про ваші пропозиції в третій годині ночі. За даними статистики, на ранковій, вечірній і нічний час доводиться близько 25% всіх гостей інтернету. Працюючи цілу добу, веб-сайт взмозі задовільнити запити кожного клієнта без виключення.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Завданням курсового проекту є створення веб-додатку, динамічної системи, яка б допомагала користовичам створювати власні веб-сайти, модифікувати вже створенні раніше. Від користивуча не вимагаються знання програмування.

До особливостей проекту можна віднести:

* зручність і функціональність інтерфейсу;
* відповідність дизайну сучасним вимогах;
* великий набір інструментів для користувача;
* можливість збереження модифікацій.

Основною частиною роботи є реалізація клієнтської частина, яка повинна бути гнучкою і швидкою при використанні, щоб забезпечити якісну роботу користувачу.

РОЗДІЛ 2. ВИКОРИСТАНІ ТЕХНОЛОГІЇ

**Веб-програмування** - галузь веб-розробки і різновид дизайну, в завдання якої входить проектування користувальницьких веб-інтерфейсів для сайтів або веб-додатків. Веб-дизайнери проектують логічну структуру веб-сторінок, продумують найбільш зручні рішення подачі інформації, а також займаються художнім оформленням веб-проекту. В результаті перетину двох галузей людської діяльності грамотний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і володіти відповідними художніми якостями.

Для розробки динамічної системи проектування веб-сторінок я використав

**користувач-сервер** архітектуру.

Архітектура клієнт-сервер є одним із [архітектурних шаблонів програмного забезпечення](http://uk.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%BD%D1%96_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F) та є домінуючою концепцією у створенні розподілених мережних [застосунків](http://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

* набір [серверів](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80), які надають інформацію або інші послуги програмам, які звертаються до них;
* набір [клієнтів](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), які використовують сервіси, що надаються серверами;
* [мережа](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0), яка забезпечує взаємодію між клієнтами та серверами.

Клієнти можуть функціонувати паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

2.1. Front-end частина.HTML, CSS

Front-end частина будь якого проекту – це те, що бачить користувач і з чим може взаємодіяти. Для розробки проекту я використав HTML, CSS а також, JavaScript, для того щоб додати більше динамічності.

HTML (від англ. Hypertext Markup Language - мова розмітки гіпертексту) - це стандартна мова розмітки документів у Всесвітній павутині. Всі веб-сторінки створюються за допомогою мови HTML (або XHTML). Мова HTML інтерпретується браузером і відображає у вигляді документа, зручному для людини. HTML є додатком SGML (стандартної узагальненої мови розмітки) і відповідає міжнародному стандарту ISO 8879. HTML-документ є текстовим файлом розмічений за допомогою спеціальних (природно, текстових) команд. Текстовий формат представлення веб-документів був вибраний виходячи з основних вимог до веб-документу: простота, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу. Мова розмітки гіпертекстових документів HTML дозволяє визначити різні типи елементів ( у оригіналі element ), що забезпечують функціональність документа: текстові фрагменти із заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання і т.д. Елементи HTML оголошуються за допомогою команд розмітки, званих тегами (від англійського tag - ярлик). Усі HTML-теги, що зустрічаються в тексті документа інтерпретуються браузером при відображенні.З 1994 року по теперішній час розробка проводиться під егідою наддержавної організації World Wide Web Consortium ( W3C ).

CSS використовується творцями веб-сторінок для задання кольорів, шрифтів, розташування і інших аспектів представлення документа. Основною метою розробки CSS було розділення вмісту (написаного на HTML або іншій мові розмітки) і представлення документа (написаного на CSS). Це розділення може збільшити доступність документа, надати велику гнучкість і можливість управління його уявленням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах висновку, таких як екранне уявлення, друк, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при висновку пристроями, що використовують Шрифт Брайля. Стандарт CSS визначає пріоритети, у порядку яких застосовуються правила стилів, якщо для якогось елементу підходять деякі правила одночасно. Це називається "каскадом", в якому для правил розраховуються пріоритети або "ваги", що робить результати передбаченими.Таблиця стилів складається з набору правил. Кожне правило, у свою чергу, складається з одного або декількох селекторів, розділенихкомами і блоку визначень. До появи CSS оформлення веб-сторінок здійснювалося безпосередньо усередині вмісту документа. Проте з появою CSS стало можливим принципове розділення змісту і представлення документа.За рахунок цього нововведення стало можливим легке застосування єдиного стилю оформлення для маси схожих документів, а такожшвидка зміна цього оформлення.

Переваги CSS розмітки:

1)Декілька дизайнів сторінки для різних пристроїв перегляду. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на КПК і стільниковому телефоні меню буде слід за вмістом.

2)Зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і кешується**.**

3) Простота подальшої зміни дизайну. Не потрібно правити кожну сторінку, а лише змінити CSS-файл.

4)Додаткові можливості оформлення. Наприклад, за допомогою CSS-розмітки можна зробити блок тексту, який решта тексту обтікатиме (наприклад для меню) або зробить так, щоб меню було завжди видно при скролінгу сторінки.

JavaScript (JS) -динамічна, [об'єктно-орієнтованa](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) [мова програмування](http://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Реалі- зація стандарту [ECMAScript](http://uk.wikipedia.org/wiki/ECMAScript). Найчастіше використовується як частина [браузера](http://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), що надає можливість коду на стороні [клієнта](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, [асинхронно](http://uk.wikipedia.org/wiki/AJAX) обмінюватися даними з [сервером](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0), змінювати [структуру](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0) та [зовнішній вигляд](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD) [веб-сторінки](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0). Мова JavaScript також використовується для [програму-вання](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)на на стороні серверу (подібно до таких мов програмування, як [Java](http://uk.wikipedia.org/wiki/Java) і [C#](http://uk.wikipedia.org/wiki/C_Sharp)), розробки [ігор](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%96_%D1%96%D0%B3%D1%80%D0%B8), стаціонарних та мобільних додатків, сценаріїв в прикладному [ПЗ](http://uk.wikipedia.org/wiki/%D0%9F%D0%97) (наприклад, в програмах зі складу [Adobe Creative Suite](http://uk.wikipedia.org/wiki/Adobe_Creative_Suite)), всередині [PDF](http://uk.wikipedia.org/wiki/PDF)-документів тощо.

JavaScript класифікують як [прототипну](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) (підмножина [об'єктно-орієнтова-ної](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [скриптову мову](http://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування ([імперативну](http://uk.wikipedia.org/wiki/%D0%86%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) та частково [функціональну](http://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) і деякі відповідні архітектурні властивості, зокрема: [динамічна](http://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B0_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F) та [слабка](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%BB%D0%B0%D0%B1%D0%BA%D0%B0_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F&action=edit&redlink=1) [типізація](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%82%D0%B8%D0%BF%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%97), автоматичне керування пам'яттю, прототипне [наслідування](http://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D1%96%D0%B4%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), [функції](http://uk.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D1%96%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) як [об'єкти першого класу](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_%D0%BF%D0%B5%D1%80%D1%88%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%83).

Незважаючи на схожість назв, мови [Java](http://uk.wikipedia.org/wiki/Java) та JavaScript є двома різними мовами, що мають відмінну [семантику](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), хоча й мають схожі риси в [стандартних бібліотеках](http://uk.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0_%D0%B1%D1%96%D0%B1%D0%BB%D1%96%D0%BE%D1%82%D0%B5%D0%BA%D0%B0&action=edit&redlink=1) та правилах іменування. [Синтаксис](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_%D0%BC%D0%BE%D0%B2%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) обох мов отриманний «у спадок» від мови [С](http://uk.wikipedia.org/wiki/C_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)), але семантика та дизайн JavaScript є результатом впливу мов [Self](http://uk.wikipedia.org/w/index.php?title=Self_(%D0%BC%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)&action=edit&redlink=1) та [Scheme](http://uk.wikipedia.org/wiki/Scheme).

2.2. Back-end частина. Node.js (Express). JSON.

Node.js –платформа з [відкритим](http://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%96%D1%82%D0%B8%D0%BA%D0%B0_%D0%B2%D1%96%D0%B4%D0%BA%D1%80%D0%B8%D1%82%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D1%83) [сирцевим кодом](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%80%D1%86%D0%B5%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%B4) для написання [серверн-ої](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) частини[веб-реалізації](http://uk.wikipedia.org/w/index.php?title=%D0%92%D0%B5%D0%B1-%D1%80%D0%B5%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%96%D1%8F&action=edit&redlink=1) на [JavaScript](http://uk.wikipedia.org/wiki/JavaScript), автором якої є [Раян Дал](http://uk.wikipedia.org/w/index.php?title=%D0%A0%D0%B0%D1%8F%D0%BD_%D0%94%D0%B0%D0%BB&action=edit&redlink=1).

Node.js характеризується такими властивостями:

* асинхронна [однонитева](http://uk.wikipedia.org/wiki/%D0%9D%D0%B8%D1%82%D1%8C) модель виконання запитів
* неблокуючий ввід/вивід
* система модулів [CommonJS](http://uk.wikipedia.org/wiki/CommonJS)
* [рушій JavaScript](http://uk.wikipedia.org/wiki/%D0%A0%D1%83%D1%88%D1%96%D0%B9_JavaScript) [Google V8](http://uk.wikipedia.org/wiki/Google_V8)

Для управління модулями використовується пакетний менеджер [npm](http://uk.wikipedia.org/w/index.php?title=Npm&action=edit&redlink=1) (node package manager).

Node.js призначений для відокремленого виконання високопродуктивних мережних [застосунків](http://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для [веб](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1), платформа може використовуватися і для створення звичайних клієнтських і серверних мережевих програм. Для забезпечення виконання JavaScript-коду використовується розроблений компа-нією [Google](http://uk.wikipedia.org/wiki/Google) [рушій](http://uk.wikipedia.org/wiki/%D0%A0%D1%83%D1%88%D1%96%D0%B9_JavaScript) [V8](http://uk.wikipedia.org/wiki/V8_(%D1%80%D1%83%D1%88%D1%96%D0%B9_JavaScript)).

Для забезпечення обробки великої кількості паралельних запитів Node.js задіює асинхронну модель запуску коду, засновану на обробці подій в неблокуючому режимі і визначенні обробників зворотніх викликів (callback). В якості способів мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується [бібліотека](http://uk.wikipedia.org/wiki/%D0%91%D1%96%D0%B1%D0%BB%D1%96%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC) libev, для створення пулу [нитей](http://uk.wikipedia.org/wiki/%D0%9D%D0%B8%D1%82%D1%8C) (thread pool) задіяна бібліотека libeio, для виконання [DNS](http://uk.wikipedia.org/wiki/DNS)-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що викликають блокування, виконуються всередині пула нитей і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки [Perl](http://uk.wikipedia.org/wiki/Perl) [AnyEvent](http://uk.wikipedia.org/w/index.php?title=AnyEvent&action=edit&redlink=1), [Ruby](http://uk.wikipedia.org/wiki/Ruby) [Event Machine](http://uk.wikipedia.org/w/index.php?title=Event_Machine&action=edit&redlink=1) і [Python](http://uk.wikipedia.org/wiki/Python) [Twisted](http://uk.wikipedia.org/w/index.php?title=Twisted&action=edit&redlink=1), але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у [веб-застосунку](http://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA), що працює в [браузері](http://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80). При написанні програм для Node.js необхідно враховувати специфіку подієво-орієнтованого програмування, наприклад, замість виконання синхронного виклику з очікуванням завершення роботи і наступною обробкою результатів, в Node.js використовує принцип асинхронного виконання, тобто код трансформується в асинхронний, при якому управління миттєво перейде до подальшого коду, а результат запиту буде оброблений по мірі надходження даних. Ні одна функція в Node.js не повинна безпосередньо виконувати операції вводу/виводу — для отримання даних з диска, від іншого процесу або з мережі потрібна установка callback-обробника.

Для розширення функціональності застосунків на базі Node.js підготовлена ​​велика колекція модулів, в якій можна знайти модулі з реаліза-цією [HTTP](http://uk.wikipedia.org/wiki/HTTP), [SMTP](http://uk.wikipedia.org/wiki/SMTP), [XMPP](http://uk.wikipedia.org/wiki/XMPP), [DNS](http://uk.wikipedia.org/wiki/DNS), [FTP](http://uk.wikipedia.org/wiki/FTP), [IMAP](http://uk.wikipedia.org/wiki/IMAP), [POP3](http://uk.wikipedia.org/wiki/POP3) [серверів](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) і [клієнтів](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), модулі для інтеграції з різними веб-фреймворками, обробники [WebSocket](http://uk.wikipedia.org/wiki/WebSocket) і [AJAX](http://uk.wikipedia.org/wiki/AJAX), конектори до [СУБД](http://uk.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) ([MySQL](http://uk.wikipedia.org/wiki/MySQL), [PostgreSQL](http://uk.wikipedia.org/wiki/PostgreSQL), [SQLite](http://uk.wikipedia.org/wiki/SQLite), [MongoDB](http://uk.wikipedia.org/wiki/MongoDB)), шаблонізатори, [CSS](http://uk.wikipedia.org/wiki/CSS)-рушії, реалізації криптоалгоритмів і систем авторизації (наприклад, [OAuth](http://uk.wikipedia.org/wiki/OAuth)), [XML](http://uk.wikipedia.org/wiki/XML)-парсери.

JSON ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) JavaScript Object Notation, [укр.](http://uk.wikipedia.org/wiki/%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) об'єктний запис [JavaScript](http://uk.wikipedia.org/wiki/JavaScript), вимовляється джейсон) — це текстовий [формат](http://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82) обміну [даними](http://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D1%96) між [комп'ютерами](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80). JSON базується на тексті, і може бути з легкістю прочитаним людиною. Формат дозволяє описувати [об'єкти](http://uk.wikipedia.org/wiki/%D0%9E%D0%B1%27%D1%94%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) та інші [структури даних](http://uk.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85). Цей формат головним чином використовується для передачі структурованої інформації через [мережу](http://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0) (завдяки процесу, що називають [серіалізацією](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F)).

Розробив і популяризував формат [Дуглас Крокфорд](http://uk.wikipedia.org/w/index.php?title=%D0%94%D1%83%D0%B3%D0%BB%D0%B0%D1%81_%D0%9A%D1%80%D0%BE%D0%BA%D1%84%D0%BE%D1%80%D0%B4&action=edit&redlink=1).

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології [AJAX](http://uk.wikipedia.org/wiki/AJAX). JSON виступає як заміна [XML](http://uk.wikipedia.org/wiki/XML) під час [асинхронної](http://uk.wikipedia.org/w/index.php?title=%D0%90%D1%81%D0%B8%D0%BD%D1%85%D1%80%D0%BE%D0%BD%D0%BD%D1%96%D1%81%D1%82%D1%8C&action=edit&redlink=1)передачі структурованої інформації між [клієнтом](http://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%94%D0%BD%D1%82_(%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) та [сервером](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80). При цьому перевагою JSON перед XML є те, що він дозволяє складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою [JavaScript](http://uk.wikipedia.org/wiki/JavaScript) в об'єкти.

За рахунок своєї лаконічності в порівнянні з [XML](http://uk.wikipedia.org/wiki/XML), формат JSON може бути більш придатним для [серіалізації](http://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8F) складних структур.

Якщо говорити про веб-застосунки, в такому ключі він доречний в задачах обміну даними як між браузером і сервером ([AJAX](http://uk.wikipedia.org/wiki/AJAX)), так і між самими серверами (програмні HTTP-інтерфейси). Формат JSON так само добре підходить для зберігання складних динамічних структур в реляційних базах даних або файловому кеші.

JSON будується на двох структурах:

* Набір пар ім'я/значення. У різних мовах це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список з ключем або асоціативним масивом.
* Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список, або послідовність.

Це універсальні структури даних. Теоретично всі сучасні мови програмування підтримують їх у тій чи іншій формі.

**Express** – швидкий серверний веб-фрейморк, який працює на базі Node.js.

Основні можливості express:

1. Гнучка система маршрутизації запитів.
2. Перенаправлення.
3. Продуктивність.
4. Налаштування уявлень на рівні додатків.

РОЗДІЛ 3. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ СИСТЕМИ

Головна сторінка проекту має наступний вигляд:

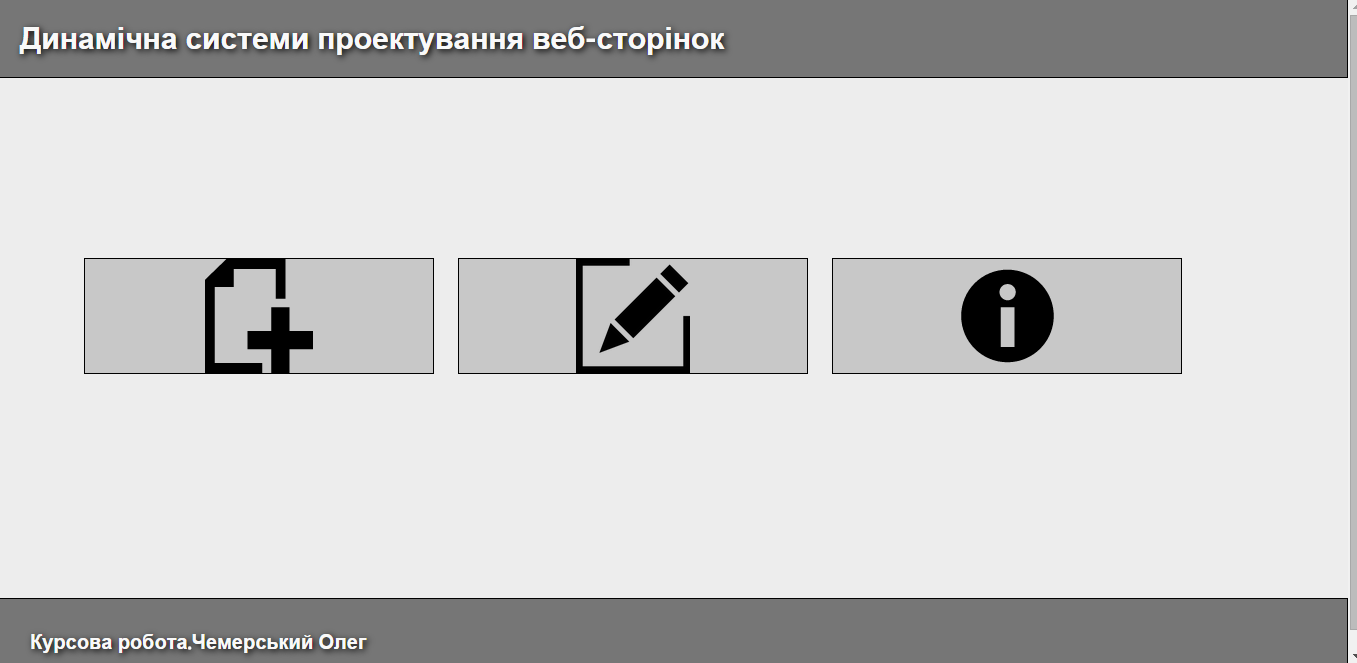


Рис. 3.1. Вигляд головного меню

Як ми бачимо, головними функціональними об’єктами є три клопки, які відповідають за навігацію по сайті. Після кліку на першу кнопку, ми будемо мати змогу почати створювати макет користувацького сайту з нуля.

Як бачимо, сама сторінка створення макету складається з двох функціональних об’єктів – “меню вибору” та кнопки “cтворити”.

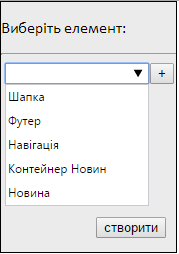


Рис. 3.2. Сторінка створення нових макетів

В “меню вибору” присутні наступні варіанти:

* Шапка (‘Header’)
* Навігація (‘Navigation’)
* Контейнер новин (‘Containter’)
* Новина (‘News’)
* Підвал (“Footer”)

При виборі опції “Шапка” зявиться блок із налаштуваннями “шапки” сайту:

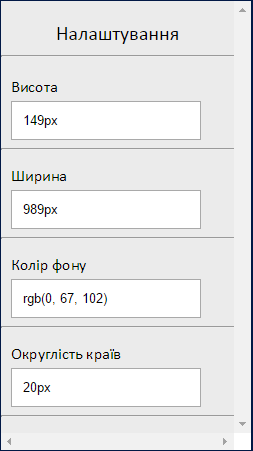


Рис. 3.3. Можливі налаштування “шапки” сайту

Можна змінювати наступні характеристики блоку “Шапка”.

* Ширину
* Висоту
* Колір заднього фону
* Заокругленість країв

**Ширину** та **висоту** можна задавати у **пікселях** або **відсотках**.

**Наприклад**, 200px або ж 30%.

Колір заднього фони можна задавати у трьох форматах:

* **hex**- значення – задається колір у шіснадцятковому вигляді (#00000-#FFFFF)
* **RGB** – задається трьома значеннями. Перше відповідає за червоний колір, наступні за зелений і синій. Наприклад, RGB(255, 0, 0) – червоний колір, RGB(0, 0, 255) – синій.
* **RGBA** – аналогічно до RGB, тільки додається ще альфа-канал, за допомогою якого можна задати коефіцієнт прозорості від 0 до 1 (0 – повністю прозорий, 1 – повністю видимий).

**Заукругленість країв** задається також в пікселях.

При виборі “Футер” зявиться блок із налаштуваннями “підвалу” сайту.

Налаштовувати, аналогічно до “Шапки” можна **ширину**, **висоту**, **колір заднього фону**, **заокругленість країв** а також змінювати **текстову інформацію**, яка переважно знаходиться в підвалі сторінки.Також можна задавати **колір шрифту**.

При виборі “Нагівації” можна створювати навігацію по користувацькому сайті. При натисканні кнопки “створити” зявиться меню налаштувань, в якому можна задати:

* Висоту
* Ширину
* Колір фону
* Округлість країв
* Вміст навігації (елементи навігації ). За допомогою цього налаштування можна змінювати вміст навігації, а саме, видаляти існуючі підпункти меню і додавати нові, як показано на рисунку 3.5.

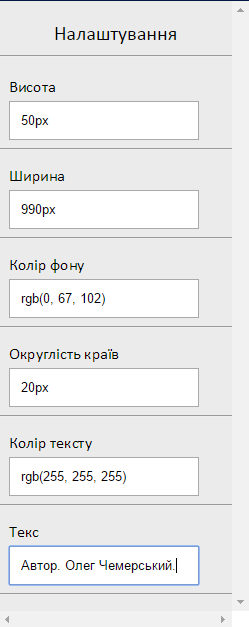


Рис. 3.4. Можливі налаштування “підвалу” сайту

При виборі “Контейнер новин” зявиться блок із налаштуваннями головногконтейнеру сайту, де зберігаються новини або ж будь яка інша інформація.

Аналогічно, як і для “Шапки” та ”Футеру” можна налаштовати ширину блоку, висоту та колір заднього фону та задавати заокругленість країв.

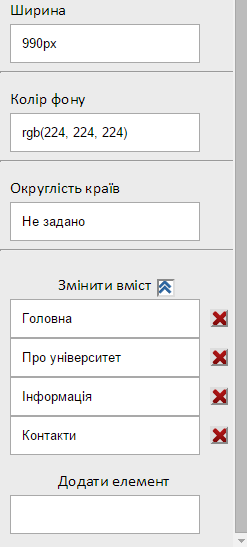


Рис. 3.5. Налаштування навігації сайту

Також можна налаштувати макет новин (елементів контейнеру новин):

* Видаляти внутрішній вміст новин (видаляти абзаци із інформацією, заголовки, зображення).
* Добавляти нові елементи.

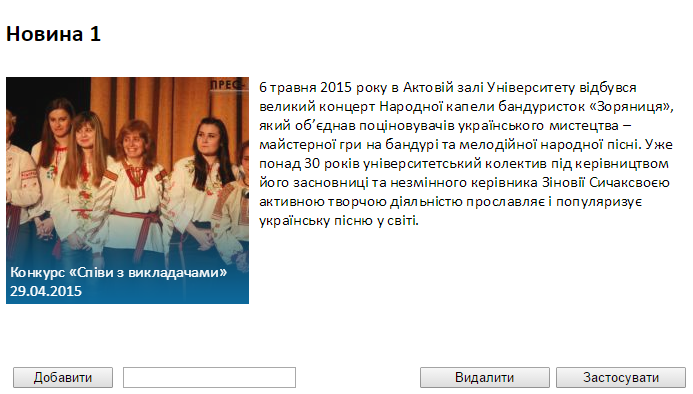


Рис. 3.6. Зміна вигляду новин

Ще один елемент, який можна додати на сторінку є “Другорядний інформаційний блок”.

В ньому також можна налаштовувати висоту, ширину, кольори та інші характеристики.

При подвійному кліку по елементу блока “Новина” можна змінити його внутрішній контент. Також можна добавляти нові елементи, такі як абзац із текстом або зображення.

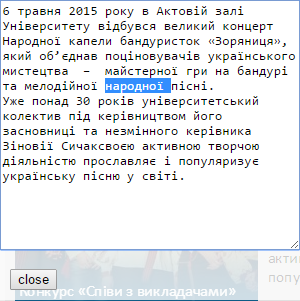


Рис. 3.7. Зміна внутрішнього вмісту абзацу новин

Також, як показано на малюнку, можна змінювати зображення на користувацькі.



Рис. 3.8. Зміна зображення

Ще до особливостей проекту можна віднести гнучкість позицінування елементів на сторінці. Користувач має змогу переміщати будь який елемент, як показано на рисунку 3.9.

Користувач може змінювати розміри блоків сторінки, виключно, за допомогою миші.



Рис. 3.9. Приклад роботи користувача

РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ

Найцікавішим та найскладнішим завданням було реалізувати **переміщення** та **змінну розмірів** елементів користувачем.

Для цього кожному блоку було прикріплений прослухувач подій, який викликався при затисненні лівої кнопки миші:

function onMouseDown (currentElement) ;

currentElement – даний елемент, на якому відбулася подія onMouseDown;

При переміщенні миші по сторнці, генерується подія mousemove. Прикріплюємо наступний прослухував подій:

addEventListener("mousemove", moving);

function moving(e) {

var x = e.clientX-startX;

var y = e.clientY-startY;

currentElement.style.top = startTop + y + "px";

currentElement.style.left = startLeft + x + "px";

}

x, y – зберігають зміщення, яке додаємо до поточної позиції елемента.

Для змінни розмірів використовується наступний алгоритм:

1.Для змінни висоти елементу за допомогою вверхньої межі:

var x = e.clientX-startX;

var y = e.clientY-startY;

currentElement.style.height = currentElementHeight + y + "px";

2.Для змінни ширини за допомогою правої межі:

var x = e.clientX-startX;

var y = e.clientY-startY;

currentElement.style.width = currentElementHeight + x + "px";

3. Для змінни ширини за допомогою лівої межі:

var x = e.clientX-startX;

var y = e.clientY-startY;

currentElement.style.left = startLeft + x + "px";

currentElement.style.width = currentElementHeight + x + "px";

Побудова елементів налаштування відбувалася наступним чином:

Властивості, які можна змінювати зберігалися, як об’єкт типу ім’я – значення;

var targetProperties = {

"userheader" : ["img", "height", "width", "background-color",

"borderRadius"],

"userfooter" : ["height", "width", "background-color", "borderRadius",

"innerHTML"],

"usermain" : ["height", "width", "background-color", "borderRadius"],

"usernavigation" : ["height", "width", "background-color", "borderRadius"],

"userheadernews" : ["height", "width", "background-color",

"borderRadius", "color", "padding", "innerHTML"],

"news" : ["height", "width","background-color", "borderRadius"],

"headermenu" : ["height", "width","background-color", "borderRadius",

"innerHTML"],

"usernavitem" : ["height", "width","background-color", "borderRadius",

"innerHTML"],

"newscontainer" : ["height", "width","background-color", "borderRadius"]

};

Тобто кожному елементу з лівої частини пари відповідає масив елементів, які можна редагувати. Ці елементи є простими HTML атрибутами.

При кліку на елемент інтерфейсу будується відповідне вікно налаштувань на основі елементів з масиву. Варто зазначити, що такий підхід має багато перевах, адже можна швидко і легко розширювати функціональність.

Процес формування налаштувань відбувається наступним чином:

for( var i = 0; i < availibleProperties.length; ++i) {

if(getPropertyValue(availibleProperties[i]) == null)

{

if(availibleProperties[i] == "innerHTML") {

content[availibleProperties[i]] = target.innerHTML;

} else { content[availibleProperties[i]] = "Не задано";}

else{

content[availibleProperties[i]] =

Values.getPropertyValue(availibleProperties[i]);

}

}

Також цікаво відбувається процес зміни текту абзаців та заголовків всі рівнів:

var edit = document.createElement("textarea");

edit.style.width = "300px";

edit.style.height = "250px";

edit.style.position = "fixed";

edit.value = el.innerHTML;

edit.addEventListener("change", function(){

el.innerHTML = edit.value;

});

Процес завантаження картинки відбувається за допомогою можливостей Node платформи, а також, додаткової бібліотеки **connect-multiparty**, яка полегшує процес завантаження зображення.

app.post('/editor', multipartMiddleware, function(req, resp) {

var tempPath = req.files.file.path,

if (path.extname(req.files.file.name).toLowerCase() === '.png') {

fs.readFile(req.files.file.path, function (err, data) {

var newPath = \_\_dirname + "/public/images/newFile.png";

fs.writeFile(newPath, data, function (err) {});

});

newPath – шлях до папки, в якій зберігаються завантажені користувачем зображення.

fs – модуль, який дає змогу доступатися до файлової частини серверу.

writeFile(path, data, callback) – функція, яка зберігає data в заданий path, після закінчення збереження, викликається callback.

ВИСНОВКИ

Дуже важливо надати людям, які не мають навиків програмування, змогу реалізовувати свої замисли та проекти.

Ще донедавна створювати сайти могли лише люди, які мали спеціальну освіту для цього, їх навчали спеціальним мовам програмування.

За допомогою мого проекту стало набагато простіше створити свій сайт за кілька хвилин і зробити це може будь-яка особа яка просто користується комп'ютером хоча б на рівні користувача і має поняття що таке інтернет. Зручний конструктор інтернет сторінок, дає змогу формувати сайт не мовою програмування а візуально, тобто безпосередньо клікнути на новій сторінці і писати текст чи додавати картинки які будуть відображатися на цьому сайті. А для того, щоб сайт мав гарне оформлення я розробив декілька шаблонів які зможуть задовільнити певні вподобання. Можна як підібрати шаблон для стилю, так і створити свій унікальний дизайн за допомогою наявного інструменту.

Сайти створюють з метою заявити про себе, показати свої фото друзям, чи просто сайт для однокласників, однокурсників, для себе чи своє хоббі, інтернет магазини тощо.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Дэвид Хэррон - Node.js Разработка серверных веб-приложений на JavaScript – 2012 с.50 - 120 .
2. Brown E. - Web Development with Node and Express – 2014 с. 45 – 67.
3. Marijn Haverbeke - Eloquent JavaScript, 2nd Edition – 2015 с. 240 – 347.
4. Nicolas Bevacqua - JavaScript Application Design. A Build First Approach – 2015 с.40 – 223.
5. Lynda.com - Creating Apps With Angular, Node, and Token Authentication 2014

Уроки 1 – 12.

1. Документація Node.js : Офіційний інтернет сайт: [сайт]: - режим доступу: [http://www.nodejs.org/api](https://nodejs.org/api/) (дата звернення: 17.03.2015).
2. Api Express: [сайт]: - режим доступу:  [http://expressjs.com/api.html](%20http://expressjs.com/api.html) (дата звернення: 22.04.2015).
3. W3school: [сайт]: - режим доступу:  [http://www.w3schools.com/](%20http://www.w3schools.com/) (дата звернення: 01.04.2015).