

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

на тему: «Розробка WEB-застосунку для створення односторінкових сайтів»

Дін Ці

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
д.т.н., доцент Гончаренко Т.А.

«__» _____ 2024 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»

на тему: «Розробка WEB-застосунку для створення односторінкових сайтів»

Виконав: студент 2-го курсу, групи КНм-23

Спеціальності: 122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Дін Ці

(прізвище та ініціали)

Керівник: д.т.н. проф. Терентьев О.О.

(прізвище та ініціали)

Рецензент: к.т.н., доц Баліна О.І.

(прізвище та ініціали)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: магістр за ОП

Спеціальність: 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

д.т.н., доцент Гончаренко Т.А.

« ____ » _____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «МАГІСТР»**

Дін Ці

Тема роботи: Розробка WEB-застосунку для створення односторінкових сайтів
затверджена наказом ректора КНУБА № _____ від « ____ » _____ 2024 р.

2. Керівник роботи: Терент'єв Олександр Олександрович, д.т.н., професор
кафедри інформаційних технологій проектування і прикладної математики

3. Строк подання студентом роботи до захисту: грудень 2024 р.

4. Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та огляд існуючих проектів

P.2. Розробка web-застосунку

P.3. Практичне використання WEB-застосунку

P.4. Ергономіка інформаційних технологій

5. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
P. 1. Аналіз предметної області та огляд існуючих проектів	Вересень 2024р.
P. 2. Розробка web-застосунку	Вересень 2024р.
P. 3. Практичне використання WEB-застосунку	Жовтень 2024р.
P. 4. Ергономіка інформаційних технологій	Грудень 2024р.
Остаточне оформлення роботи, перевірка на плагіат	Грудень 2024р.
Попередній захист роботи на кафедрі	Грудень 2024р.

6. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	д.т.н., проф. Терентьев О.О.		
Прийом програмного продукту	д.т.н., проф. Бородавка Є.В.		

7. Дата видачі завдання: 5 серпня 2024 р.

Керівник

Терентьев О.О.

(підпис)

(прізвище та ініціали)

Магістр

Дін Ці

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

«Розробка WEB-застосунку для створення односторінкових сайтів».

Кваліфікаційна робота за спеціальністю: 122 «Комп'ютерні науки». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Результатом роботи є розробка, обслуговування та підтримка роботи діяльності сайту для генерації односторінкового лендінгу. Застосування цього WEB-застосунку дозволяє провести створення лендінгу для подальшого його впорядкування та користування.

Ключові слова: інформаційні технології, бази даних, інтернет провайдер, захист інформації.

SUMMARY

"Development of WEB-application for creating one-page sites".

Qualification work in the specialty: 122 "Computer science". - Kyiv National University of Construction and Architecture. - Kyiv, 2024.

The result is the development, maintenance and support of the site to generate one-page landing. The use of this WEB-application allows you to create a landing page for further organization and use.

Key words: information technologies, databases, internet provider, information protection.

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ ПРОЕКТІВ	9
1.1 Основні поняття cms.....	9
1.2 Критерії вибору cms.....	12
1.3 Поява та розвиток.....	13
1.4 Існуючі рішення CMS.....	15
1.5 Види cms.....	16
1.6 Рейтинг cms.....	20
1.7 CMS «1С-БИТРИКС».....	21
1.8 CMS «Joomla».....	22
1.9 CMS «Wordpress».....	24
2 РОЗРОБКА WEB-ЗАСТОСУНКУ	28
2.1 Робота розробленої спеціалізованої програмної системи керування контентом.....	28
2.2 Розробка інтерфейсу web-застосунку.....	31
2.3 Аналіз процесу впровадження.....	33
2.4 Ядро web-застосунку.....	38
2.5 Допрацювання ядра web-застосунку.....	42
2.6 Вибір хостинга.....	51
3 ПРАКТИЧНЕ ВИКОРИСТАННЯ WEB-ЗАСТОСУНКУ	55
3.1 Додавання та редагування елементів до проекту.....	55
3.2 Редагування SEO налаштувань.....	58
3.3 Експорт проекту.....	59
3.4 Передпоказ проекту.....	61
3.5 Редагування css елементу.....	62
3.6 Редагування контенту.....	63
4 ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	65
4.1 Розрахунок часу евакуації людей при пожежі в приміщенні.....	65
4.2 Ергономічні вимоги до організації і обладнання робочих місць з комп'ютерною технікою.....	73
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79

ВСТУП

На сьогоднішній день найактуальнішою темою є створення сайтів. Це з тим, що будь-яка сфера діяльності, незалежно від цього пов'язана вона з товарами чи послугами, потребує просування, основою якого є інформування.

Головне призначення сайту полягає в тому, щоб розповісти користувачам про цю компанію. В Інтернеті існують різні способи інформування, але, як правило, всі вони зводяться до одного - до веб-сайту.

Web-сайт є сукупністю електронних документів, об'єднаних однією адресою. Зі зростанням технологій web-сайти стають не просто сукупністю документів, але й представленням низки технологій, які використовуються як для привернення уваги, так і для додавання різних функцій, що дозволяють робити перебування користувача на сайті не лише захоплюючим, а й зручним.

Створення web-сайтів актуальне і спричинило створення ряду спеціальних систем, що дозволяють не тільки створювати сайти в досить короткі терміни, але ще й керувати вмістом цього сайту з використанням спеціальної панелі. Системи керування контентом - це програмне забезпечення, що дозволяє публікувати та змінювати інформацію на сайті самостійно, без залучення розробників.

Для більшості рекламодавців область підвищеного інтересу є користувачем мережі. Ні друкована продукція, ні радіо чи телебачення не можуть подати інформацію в такому зручному, презентабельному та переконливому вигляді, як це зробить веб-сайт. Вичерпна інформація, прайс-листи, каталог товарів, можливість здійснювати всілякі замовлення, рубрика "відповідь-питання", розміщені на сайті, позбавляють необхідності роз'яснювати одні й ті самі питання телефоном, дозволить звільнити робочий час співробітників фірми для ефективного вирішення нагальних завдань. Саме тому створення веб-сайту для комерційної організації продиктовано сучасною потребою.

Корпоративний сайт зараз це не просто частина будь-якої великої компанії, він є невід'ємною частиною практично кожної компанії, є складовою фірмового стилю, відображає особливості та напрямки діяльності компанії. Щоб бізнес був прибутковим та успішним наявність корпоративного сайту – це обов'язкова умова. Розвиток корпоративного сайту - це оптимальний та економічно вигідний засіб зміцнити позиції компанії, що дозволяє серйозно економити на інших видах реклами та зменшити штат співробітників.

Унікальний контент сайту, грамотно розбавлений ключовими словами, а також привабливий дизайн сторінок здатні принести будь-якій компанії відчутний економічний прибуток.

В дипломній роботі розробляється сервіс, створений для полегшення їх роботи, а також рішення відомих проблем, що трапляються в роботі веб-мастерів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ ПРОЕКТІВ

1.1 Основні поняття cms

Термін, CMS технології в даний час зустрічається досить часто. Одні називають CMS «движком сайту», інші інструментом для вирішення рутинних завдань, а для когось CMS це те, «на чому стоїть сайт».

CMS (від англійської Content Management System) – система керування вмістом (контентом) - комп'ютерна програма або інформаційна система, яка використовується для організації та забезпечення процесу спільного створення, управління та редагування вмісту сайту.

Основна мета будь-якої CMS - це наповнення, редагування та керування інформацією. Якщо говорити простою мовою, то це система, яка дає користувачеві можливість доповнювати та змінювати розділи, проводити редагування вмісту сайту, змінювати графіку та дизайн сайту. Сучасні CMS дозволяють вирішувати всілякі завдання управління сайтом користувачам, які не мають навичок програмування.

Залежно від характеру роботи шаблону, CMS диференціюють на три типи:

Автономні. Система самостійно опрацьовує дані. Такі CMS підходять для статичних сайтів, структура яких складається з набору незмінних блоків.

Інтерактивні. Застосовуються для керування динамічними сайтами.

Гібридні. Поєднують у собі функції динамічних та автономних систем.

Завдяки цій системі сам сайт неможливо представити у вигляді "книги" з вже надрукованими сторінками. Її просто не існує, а процеси верстки та видання починаються в той момент, коли користувач простягає руку до порожньої книжкової полиці, тобто надсилає запит. Сайт - це набір компонентів, які збираються на ходу в одне ціле при надходженні відповідного запиту. За "складання" сторінок і відповідає CMS. Чим швидше програма генерує у відповідь запит, тим вище оцінюється її робота. Звичайно, існують прості сайти, в яких можна обійтися і без CMS, але організувати роботу вручну для великих ресурсів просто неможливо.

CMS можна назвати незамінним помічником для власника сайту, який не має спеціальних навичок для роботи з ним. Система виступає в ролі посередника, який "перекладає" команди людини та надає їй лаконічний інструментарій для управління ресурсом.

Сайти, організовані за допомогою системи керування контентом, засновані на таких технологіях: веб-сервер, сховище даних (часто СУБД), веб-додаток для забезпечення роботи самої системи, візуальний (WYSIWYG) редактор сторінок, файловий менеджер з веб-інтерфейсом для управління файлами сайту, система управління правами користувачів та редакторів сайту.

Як основа веб-додатка, що реалізує роботу CMS, найбільш широкого поширення набули такі технологічні платформи як PHP, Perl, .NET.

Більшість сучасних систем керування контентом реалізовані як візуального (WYSIWYG) редактора. Це програма, що створює HTML-код із спеціальної спрощеної розмітки, яка дозволяє користувачеві простіше форматувати текст.

Затребуваність систем управління — найкращий показник ефективності їхньої роботи, але кожна програма має не тільки сильні, а й слабкі сторони. До очевидних переваг CMS можна віднести:

Доступність веб-розробки. Крім CMS із закритим вихідним кодом існує безліч безкоштовних систем управління. Їх функціонал може бути злегка урізаний, але це не означає, що з безкоштовними двигунами складно працювати.

Спрощення процесів роботи із сайтом. Щоб організувати функціонування власного веб-ресурсу, не потрібно бути сивоволосим гуру програмування. Система управління дає можливість користувачеві з будь-яким рівнем знань повноцінно працювати з ресурсом: керувати ним, редагувати інформацію, коригувати структуру (додавати та прибирати розділи).

Економія часу на створенні всього сайту повністю та його окремих компонентів. Раніше складання була процесом довгим та трудомістким. CMS автоматизувала більшу частину робочих процесів, заощадивши час людини.

Система управління дарує можливість попрацювати над дизайном ресурсу та покращити якість його вмісту.

Багатофункціональність. Сучасні системи управління мають настільки складну багатоблочну структуру, що схожі на конструктори. При роботі з такою універсальною гнучкою конструкцією будь-який програміст зможе скласти свій, неповторний сайт.

Контроль доступу. Система CMS має можливість розподілу ролей між користувачами сайту. Наприклад, відвідувач з особистим обліковим записом може додавати товари в кошик, але не має права зміни блоків меню і не здатний вплинути на інші елементи сайту.

Крім плюсів у CMS є і зворотний бік з недоліками:

Широке поширення систем управління: показник успіху та підвищення вразливості. Внутрішні структури найпоширеніших систем давним-давно вивчені хакерами, тому їх слабкі місця стали надбанням широкого загалу. Зламати таку систему для професіонала не важко.

Шаблонність. Хоча CMS відрізняються різним ступенем гнучкості, системи у спрощеному поданні – набір шаблонів. В одних CMS їх більше, а в інших – менше. Складно втілити свої креативні ідеї, якщо доводиться працювати із заздалегідь макетами.

Деякі модулі CMS доводиться доопрацьовувати окремо, оскільки у конструкції стандартних систем їх немає. У цьому плані CMS обмежують творчий потенціал користувача.

Більшість сучасних CMS зазвичай складаються з двох частин: backoffice - інфраструктурна система, що забезпечує функціональність і зберігання інформації та front-office – частина, яка забезпечує інтерфейс з користувачем. У більшості сучасних CMS back-office базується на будь-який СУБД, може включати сервера додатків та портальне рішення. Портальне рішення великий

клас програмних продуктів, які служать для оптимізації бізнес-процесів та управління інформаційними ресурсами.

У свою чергу front-office має веб-інтерфейс і допускає використання стандартних офісних пакетів. Редагування документів (текстові редактори, електронні таблиці, засоби створення презентацій, поштові системи тощо). При цьому вся функціональність, складність розробки та адміністрування зосереджені на back-office, а власні властивості - у front-office.

Завдяки дизайн-шаблонам, заготовленим заздалегідь у системах управління контентом, надається можливість конструювати зовнішній вигляд сторінок, просто вибравши шаблон, що сподобався. В цьому полягає великий плюс користувачів систем управління контентом, так як тепер вони менше залежать від розробників їх сайтів. Адміністратор системи управління контентом може розташувати різні інформаційні блоки в рамках заданої сторінки за допомогою однієї тільки миші, задати їх розміри, колір та інші атрибути і зберегти стан сторінки, щоб користувачі сайту бачили її у заданому вигляді.

Ще одна перевага, яку дає CMS, це можливість використання технології Drag&Drop під час управління змістом сайту. В сучасній CMS для того, щоб задати нове положення для документа в структурі або ж для запису в списку, потрібно лише «зачепити» цю позицію мишею та «перетягнути» на нове місце.

1.2 Критерії вибору cms

До питання вибору CMS для сайту потрібно поставитися з усією серйозністю, адже від цього залежить комфорт роботи з ним та потенціал подальшого розвитку. І треба сказати, що немає "поганих" або "хороших" двигунів, а є підходящі для ваших цілей, і не дуже.

Без чітко визначеного набору вимог, яким повинна відповідати відповідна вам система, зробити правильний вибір буде дуже складно. Найчастіше люди вибирають перевантажені функціоналом CMS, більшу частину функцій яких вони просто не будуть використовувати. І в цьому

проблема, адже безліч функцій призводить до того, що сайтом складно керувати, а для новачка, іноді взагалі неможливо.

Що найцікавіше - можна викласти кілька тисяч доларів за систему корпоративного рівня, в той же час, коли популярний open-source движок дозволяє вирішувати аналогічні завдання безкоштовно.

Двигун сайту – це не той випадок, коли «безкоштовно лише сир у мишоловці». Якщо безкоштовна CMS може виконувати завдання, використовуйте її. Тим більше що можна не орієнтуватися на початковий набір функцій, а розширювати функціонал спеціальними плагінами. До речі, плагіни можуть бути і платними.

У безкоштовних двигунах багато в чому доведеться розбиратися самостійно (читати форуми, ставити питання, виправляти помилки своїми руками). Або платити за це фахівцю. Якщо двигун платний, всі питання можна звернутися до підтримки фірми-виробника.

Необхідність використання CMS з'являється там, де процеси створення, управління та публікації є надто складними. Для того, щоб оцінити необхідність використання CMS, мають бути проаналізовані такі критерії:

- кількість контенту. Цей пункт включає не тільки кількість елементів, а й кількість компонентів, які система повинна керувати;
- кількість співробітників та їх функцій;
- Розмір зміни контенту - розмір контенту, який додається або видаляється із системи або кількість змін у конструкції, які система має підтримувати.
- кількість публікацій, які будуть створені. Складність публікації залежить від кількості різних публікацій, які мають бути створені, а також від ступеня персоналізації, яка повинна застосовуватися.

1.3 Поява та розвиток

Історія систем управління контентом така стара, як і історія контенту, яка збігається з історією людського знання. Згодом розвиваються інструменти та методи, які змінюють характер управління контентом людьми.

Поліпшується передача знань від покоління до покоління, що дозволяє сучасному світу бути настільки неймовірно продуктивним, порівняно з нашими предками.

У зв'язку з цим слід зазначити, винахід листа, створюючи портативні документи, засновані на папері (папірусі), угруповання декількох сторінок у книгах (свитки переплетених томів), колекції книг у бібліотеках (Олександрія), каталогізація книг (Перші рукописні списки, потім за окремими картки, розташовані в алфавітному порядку за назвою або автором) класифікації книг на категорії, та розташування книг про фізичні полиці для полегшення навігації подібних матеріалів. Тому можна сказати, що методологія управління контентом, з погляду сучасного оцифрування знань, дуже мало відрізняється від методології, яка використовується до появи комп'ютерних технологій.

Управління контентом може бути описано як вирішення важливої проблеми для виробника контенту - проблема доставки змісту до споживача, а також проблеми вилучення змісту з друкованої або електронної форми.

У той час як "Система керування контентом" може бути описана як ручний процес, тут термін використовується у сенсі комп'ютерних програмних засобів, які допомагають людям створювати, керувати, доставляти, переміщувати або переглядати контент.

Деякі великі досягнення в галузі комп'ютерного програмного забезпечення та апаратних засобів можуть бути використані для диференціації історичних моментів у розвитку CMS. Серед апаратних стадій:

- від зберігання інформації на перфокартах та магнітних стрічках, до зберігання на жорстких дисках та SSD пам'ять;
- загальний доступ до центральної ЕОМ через окремі термінали;
- РС;
- моделі клієнт/сервер;
- Комп'ютерні мережі.

Інноваційної діяльності у сфері програмного забезпечення включають:

- комп'ютерні операційні системи та файлові системи;
- бази даних (плоска, ієрархічна та реляційна);
- розподілені обчислювальні системи;
- мови розмітки (SGML, HTML);
- Набір протоколів (TCP/IP тощо), які підтримують веб-програми.

1.4 Існуючі рішення CMS

В даний час системи управління контентом розвинулися настільки, що є одним із найбільш конкурентних ринків додатків. На сьогоднішній день у світі існують десятки тисяч таких додатків. Класифікувати CMS можна на системи з відкритим вихідним кодом та системи із закритим вихідним кодом.

Розглядаючи системи з відкритим кодом, слід зазначити, що головною перевагою таких продуктів є їхня доступність, наявність вихідного коду, можливість локалізації. Таке ліцензування програм дає можливість не лише вільно отримувати, а й швидко оновлювати версії товарів до актуальних. Нові версії виходять регулярно, а виправлення знайдених помилок не виливається у тривале очікування «латок» або сервіспаків, як це зазвичай відбувається у випадку із комерційними рішеннями. У продукт із відкритим кодом можна внести зміни та з урахуванням своїх власних вимог, тим самим отримати максимальний результат.

Системи з відкритим кодом мають також свої недоліки. Використання таких систем може бути пов'язане з наступними проблемами:

- Відсутність техпідтримки – системи пропонуються переважно без технічної підтримки та всі проблеми, пов'язані з використанням таких систем розробник має вирішувати самостійно.
- Безпека – системи з відкритим кодом мають слабку захистом.

В даний час найбільш використовувані та відомі системи керування контентом, це системи з відкритим вихідним кодом.

Найбільш найпопулярнішими є системи: Joomla, WordPress, Drupal, MODX, Django. Системи керування контентом із закритим кодом мають ряд своїх переваг. Виробник платної CMS зацікавлений у притоці нових клієнтів

та продовженні співпраці зі старими. Тому такі системи мають високий рівень служби підтримки та супроводу. Комерційні системи, як правило, створюються великою командою розробників, до яких крім програміста входять архітектор, тестувальник, проектувальник, документатор та інші. У загальному випадку це призводить до того, що система комерційного плану зручніша у використанні.

Вона більш надійна у роботі та оптимізована за вихідним кодом. Такі системи краще документовані. У 2016 році найпоширенішими були наступні CMS із закритим вихідним кодом: 1С-Бітрікс, UMI.CMS, AMIRO.CMS, UralSoft.WEB та CMS S.Builder .

1.5 Види cms

Всі CMS-системи, що випускаються в даний час, можна поділити на 3 категорії:

- індивідуальні (студійні) CMS, розробку сайтів на
- яких може здійснювати лише їхній розробник;
- тиражні, що випускаються великими компаніями з метою
- отримання прибутку;
- безкоштовні, що вільно розповсюджуються в мережі інтернет.

Студійні CMS-системи (їх ще називають індивідуальними) – це такі системи, розробку сайтів на яких може здійснювати тільки їх розробник. Системи такого виду поділяються на платні та безкоштовні. Як правило, такі системи не користуються популярністю у користувачів. Проблема полягає в тому, що купуючи студійний двигун, або більше, SAAS рішення (технологія продажу та використання програмних продуктів, що передбачає, що замовникам надається доступ до ПЗ через всесвітню павутину), потрібно бути готовим до того, що при необхідності, розірвання відносин зі студією, можливо, виділяється в дуже складний процес. Також власнику такого сайту буде складно знайти веб-розробника, який буде готовий розбиратися з незнайомою йому системою CMS. Багато розробників кажуть, що у таких

випадках набагато простіше створити сайт з нуля, ніж вникати у тонкощі іншої студії.

Більшість студій намагається робити свої проекти на професійних двигунів. Можна сказати, що студійні системи керування контентом це пережиток минулого. Основні гравці ринку та відомі студії на Сьогодні практично не працюють на CMS-системах власної розробки.

Такі системи мають значні недоліки. Одним із Недоліками студійних систем є слабка система захисту. В наш час, зі збільшенням кількості хакерських атак, цей фактор є особливо важливим. Особливо якщо йдеться про сайт інтернет-магазину, де потрібний особливо ефективний захист даних користувачів.

Ще одним недоліком таких систем є обмежений доступ. Це означає, що на нашому сайті не буде можливості вносити корективи, додавати дані чи кардинально змінювати наш сайт.

Другий вид систем – це тиражні (комерційні) CMS-системи. Тиражні системи управління контентом – це продукти, розроблені спеціалізованими комерційними організаціями з метою отримання прибутку в результаті продажу ліцензійного продукту та наступного технічної підтримки.

Комерційні системи замовляють переважно великі компанії, які бажають отримати готовий продукт швидко та не хочуть витратити час на налаштування. Такі системи коштують від кількох десятків доларів до кілька тисяч доларів і зазвичай мають кілька редакцій. Кожна редакція має свої можливості та дозволяє комплектувати систему різними модулями в залежності від потреб конкретного покупця.

Можливості системи можна розширити будь-якої миті, звернувшись до розробнику товару. Вартість впровадження часто включена у вартість самої авторської CMS. Оплачуються найчастіше кожна окрема ліцензія для кожного створюваного сайту. Як правило, такі системи містять вичерпну документацію.

Останній вид систем - безкоштовні CMS, які є найбільш популярними серед користувачів систем керування контентом. Безкоштовні CMS дозволяють веб-майстру вносити в систему велике кількість змін, модулів, плагінів, тем оформлення, шаблонів та модулів, що суттєво розширює функціонал сайту, дозволяє встановлювати додаткові можливості, серед яких обробка, порівняння, аналіз прайсів та інші функції. Використання таких систем може бути використано для створення інтернет-ресурсу найрізноманітнішої спрямованості та складності. Таким ресурсом може бути інтернет магазин, форум, блог, сайт-візитка або просто звичайний сайт, корпоративний портал.

Головною перевагою таких систем є те, що вони є безкоштовними та при встановленні або оновленні не вимагають жодних вкладень у фінансовому плані. Для веб-майстра-початківця, який не має до цього практики створення сайтів та не має можливості придбання комерційного продукту, такі системи є найоптимальнішим варіантом.

Крім того, що ці системи безкоштовні, вони мають ще поряд наступних переваг.

1. Відкритий код. Безкоштовні системи мають відкритий код вдосконаленні цих систем бере участь багато фахівців з всьому світу. Завдяки цьому, безкоштовні системи розвиваються стрімко та динамічно.

2. Безкоштовні шаблони. Для безкоштовних систем у мережі інтернет можна знайти безліч безкоштовних шаблонів, доповнень, плагінів, модулів, тем оформлення. Все це спрощує роботу веб-розробника та розширює функціональні можливості самої системи.

3. Навчальний контент. По роботі з безкоштовними системами керування сайтом можна знайти безліч навчальних уроків, спільнот, які допоможуть веб-майстру розібратися в системі, вирішити проблеми і знайти відповіді на питання.

Але, крім своїх плюсів, безкоштовні системи мають і свої мінуси.

Основними мінусами таких систем є.

1. Відсутність техпідтримки. На відміну від комерційних CMS, вебмайстер, який використовує безкоштовну систему, не має гарантованої техпідтримки. В результаті пошук необхідної інформації може виявитися дуже довгим та стомлюючим. Звичайно, є форуми та спеціалізовані спільноти, у яких можуть допомогти із рішенням проблеми. Але в таких випадках відповіді, можливо, доведеться чекати дуже довго, на відміну від випадку з наявною техпідтримкою.

2. Наявність помилок. Помилки у безкоштовних CMS зустрічаються набагато частіше, ніж у платних. Причина цього проста. У випадку з комерційними продуктами, розробники не можуть ризикувати своєю репутацією, тому змушені постійно відстежувати та виправляти помилки. Творці безкоштовних CMS нічим не зобов'язані людям, що використовують їх розробки, і тому не завжди своєчасно займаються виправленням помилок, що виникають у системі.

3. Проблеми із безпекою. Програмісти, які займаються написанням платних систем керування контентом, дуже уважно належать до питань безпеки. У той час як розробники безкоштовні CMS не надають цій проблемі настільки великого значення.

Цю особливість безкоштовних CMS можна назвати найсерйознішою проблемою, тому що впоратися з цим мінусом самотійно вкрай важко. Навіть якщо веб-майстру вдасться внести в CMS зміни, які допоможуть покращити її безпеку, то подібні до маніпуляції. займуть дуже багато його робочого часу.

Як видно з вище сказаного, свої переваги мають як комерційні версії систем керування контентом, так і безкоштовні.

Перш ніж розпочати розробку сайту на одному з розглянутих вигляді CMS-систем, необхідно визначити його основні завдання та функціонал. У кожній системі є переваги та недоліки, які наполегливо рекомендується вивчити перед розробкою контенту. Тільки після того, як визначилися основні цілі та завдання, тоді можна приступати до вибору системи керування контентом. Якщо це невеликий, простий проект, ні коштів на комерційну

версію або просто немає потреби переплачувати, тоді можна зупинитись на безкоштовній системі. Якщо проект серйозний, то заощаджувати не потрібно і краще придбати комерційну версію системи.

1.6 Рейтинг cms

В даний час існує безліч систем управління контентом. Не тільки їх точну, але навіть зразкову кількість назвати неможливо, тому що кожен день випускаються нові продукти, та їх стає дедалі більше. Тому розглянемо ті системи, які є найпопулярнішими та відомими на цьому ринку.

Від вірного вибору CMS залежить успішність майбутнього сайту, а правильно підібраний функціонал движка забезпечить зручне та просте розвиток ресурсу. Тому вибору CMS-системи приділяється особливе увага. Спростити вибір системи іноді допомагає рейтинг. У 2016 році компанією iTrack був складений рейтинг CMS-систем. Усього було опитано близько 5 млн. доменів зони RU.

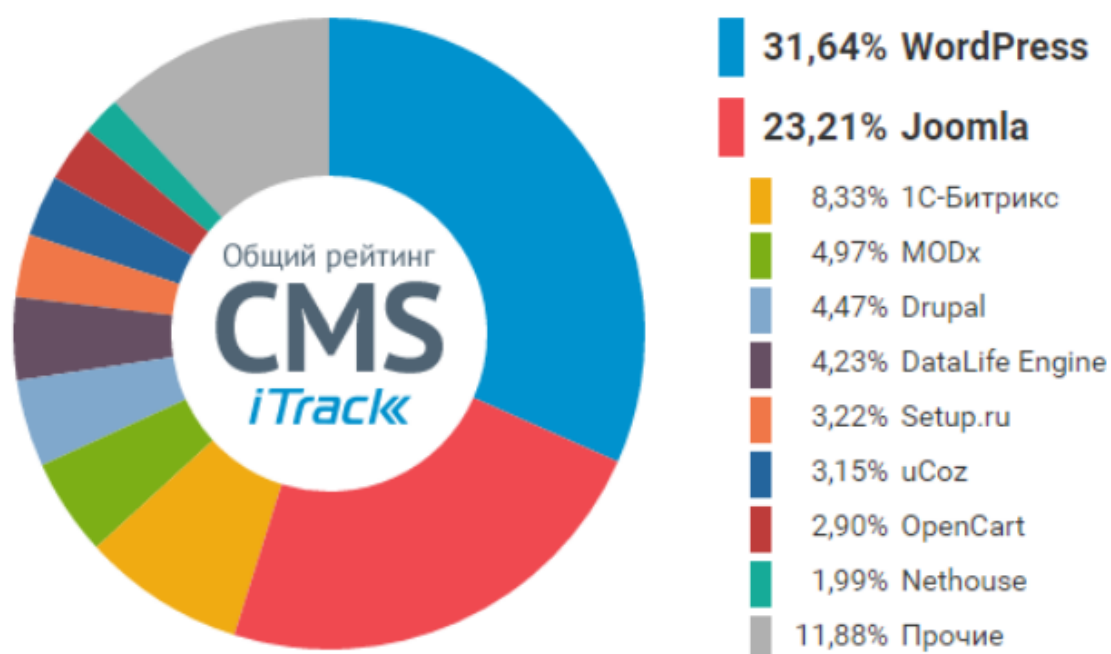


Рисунок 1.1- Загальний рейтинг CMS.

Як видно з рейтингу, наразі лідерами ринку CMS є безкоштовні рішення WordPress і Joomla.

1.7 CMS «1С-БИТРИКС»

1С-Бітрікс: Управління сайтом» - це професійна система управління веб-проектами, універсальний програмний продукт для створення, підтримки та успішного розвитку:

- корпоративних сайтів
- інтернет-магазинів
- інформаційних порталів
- сайтів спільнот
- соціальних мереж та інших веб-проектів

Технологія «Композитний сайт» поєднує всі переваги статичного сайту з динамічними можливостями. Статичний сайт – це готова HTML-сторінка, яка не містить програмного коду. Зазвичай дуже швидко завантажується для користувача, але в сучасному сайтобудуванні використовується рідко і не застосовується для інтернет-торгівлі.

Динамічні сайти використовують програмний код, тому швидкість завантаження сайту у користувачів значно повільніша. При цьому динамічні сайти використовуються більшість сучасних сайтів, включаючи інтернет-магазини. «Композитний сайт» забезпечує максимально високу швидкість відображення динамічного сайту за рахунок відображення кешованих копій сторінок сайту та відображення оновлених даних у місцях, де вони зазнали змін з моменту кешування.

CMS 1С-Бітрікс містить велику кількість модулів та готових рішень, які дозволяють створювати на ньому сайти різної тематики та складності, від ефективного односторінника до великого інтернет-магазину з тисячами товарів. Функціональність системи залежить від редакції: що дорожча ліцензія, то більше можливостей отримує веб-майстер. Наприклад, при покупці редакції «Бізнес» та будівництві інтернет-магазину доступні такі модулі та інструменти:

Модуль управління продажами. У ньому налаштовуються платіжні системи, задаються параметри магазину.

Каталог з наочним представленням товарів, а також управлінням вартістю та знижками.

Двосторонній зв'язок з 1С, що забезпечує постійний обмін даними.

Інтеграція з CRM-системою "Бітрікс24".

Складський облік для управління товарами з усіма документами, звітами, постачальниками та приходом за штрих-кодом.

Необмежену кількість інтернет-кас.

Налаштування валют з автоматичним імпортом курсів із Центрального Банку.

Конструктор звітів із шаблонами.

1.8 CMS «Joomla»

CMS Joomla – система управління контентом (Content management system), написана за допомогою мов PHP та JavaScript, що застосовує в як сховища бази даних MySQL. Система відноситься до вільного ПЗ та поширюється під ліцензією GNU GPL.

Joomla – це проект, що розробляє програмне забезпечення з відкритим вихідним кодом, і, як і всякий подібний проект, він постійно у русі.

Перша версія платформи з'явилася в 2005 році, успадкувавши проект Mambo. На основі цього продукту розробники створили нову CMS із відкритим кодом. У розробці Joomla беруть участь програмісти з усього світу, яких налічується вже понад 500 000 чоловік.

На сьогоднішній день існує два типи версій Joomla, які відрізняються один від одного періодичністю виходу: зі стандартною підтримкою (STS) та з довгостроковою підтримкою (LTS). Версії зі стандартною підтримкою

придатні до застосування протягом 7 місяців, з довгостроковою - протягом 22 місяців. Кожні 6 місяців виходить нова STS. Версії x.5 включають LTS (наприклад, такі як Joomla 2.5, Joomla 3.5 і т.д.).

На CMS Joomla можна побудувати будь-які динамічні сайти. Такі сайти складаються зі сторінок, що містять контент, тобто текст, зображення, аудіо та відео. Сторінки таких сайтів часто автоматично змінюються або підлаштовуються, за певним зразком.

За допомогою даної системи керування контентом можна побудувати такі сайти, як блоги, сайти для створення та просування бренду, сайтивізитки, сайти спільнот, інтернет-магазини, форуми, онлайн-галереї, новинні чи особисті веб-сайти.

Особливістю системи є те, що за початкової установці вона має мінімальний набір інструментів, але при необхідності його можна доповнювати самостійно. Завдяки цьому істотно зменшується загроза адміністративної панелі зайвими елементами, а також зменшується навантаження на сервер та значно економиться місце на хостингу. У CMS «Joomla» є можливість показати інтерфейс фронтальної та адміністративної частини усіма мовами.

Велика кількість мовних пакетів включено до каталогу розширень. Їх можна легко встановлюватися базовими засобами адміністрування.

Як і всі сучасні CMS, Joomla дозволяє працювати з двома інтерфейсами – користувальницьким та адміністративним. Саме адміністрування сайту зазвичай не викликає великих труднощів навіть у початківця.

CMS «Joomla» має велику кількість платних і безкоштовних розширень. На сьогоднішній день налічується близько 8000 таких розширень, завдяки яким можна надати будь-якому сайту функціональність. І з кожним днем кількість цих розширень лише зростає.

Функціонал CMS Joomla влаштований так, що для того, щоб створити сайт, практично не потрібно мати знання спеціальних мов веб-програмування. Це необхідно найчастіше лише для створення своїх розширень. Для того щоб

створити сайт за допомогою даної системи, достатньо її встановити, по необхідності вибрати розширення, а потім, на панелі адміністратора, використовуючи спеціальні елементи управління, виконати налаштування свого майбутнього сайту та наповнити його необхідним контентом.

На базі Joomla зверстані сайти таких компаній як Peugeot та Mitsubishi, сайт Гарвардського університету, а також деякі з відомих споживчих майданчиків.

Якщо порівнювати CMS Joomla з іншими системами, то можна зробити висновок, що ця система більш пристосована саме для веб-майстра-початківця. Кількість матеріалів, уроків, різних спільнот допоможуть швидше освоїтися у цій системі. Також варто згадати величезну кількість різних доповнень, розширень, плагінів, модулів та шаблонів. У розглянутих раніше системах таких доповнень набагато менше.

1.9 CMS «WordPress»

CMS "WordPress" на даний момент є лідером цього ринку.

CMS WordPress – сучасна система управління вмістом сайту з відкритим вихідним кодом, яка є безкоштовною поширенням (поширюється під універсальною загальнодоступною ліцензією GNU). Спочатку розроблялася як система для зручного створення та ведення інтернет-щоденників. Написано систему на PHP. Як і інші системи, написані на PHP, WordPress є автономним у тому сенсі, що завдання, пов'язані з установкою, конфігурацією, управлінням та адмініструванням містяться у модулях PHP. Як база даних, як і більшості безкоштовних CMS, використовується MySQL. За допомогою даної системи можна створювати сайти будь-яких типів - від блогів до різних ресурсів новин або інтернет-магазинів. Вбудована система "тем" та "плагінів" разом з вдалою архітектурою дозволяє конструювати практично будь-які проекти.

Основною перевагою WordPress є те, що він надзвичайно гнучкий у використанні. Ідеологія WordPress полягає в легкому та максимально швидкому програмному ядрі, що дозволяє підключати до нього необмежену кількість різних додаткових модулів та плагінів залежно від завдань, що

ставляться розробником. окрім цього WordPress, як і Joomla має величезний набір різних готових шаблонів, які дозволяють зробити веб-сайт посправжньому унікальним. Якщо при цьому веб-майстер володіє основами HTML та CSS, PHP та MySQL, то завдяки цьому можна досягти великої гнучкості у функціоналі сайту та його оформлення. Але навіть якщо користувач не має всіх цих знань, він все одно без особливих зусиль зможе зробити свій власний сайт. Також, не маючи знань у програмуванні, користувач може керувати рубриками, створювати нові сторінки та записи, вставляти у них картинки, таблиці, відео, додавати або видаляти віджети.

Однією з головних особливостей Wordpress є те, як організована структура бази даних. Завдяки гнучким та функціональним зв'язкам можна створювати та виводити на сторінку матеріал будь-якого виду з будь-якими параметрами. Вбудована система "тегування" створює додаткові зв'язки для матеріалів сайту, що, дозволяє оперувати всіма записами, що відповідають певним умов.

У CMS "Wordpress" пропонується гнучка схема організації структури сайту на основі таксономії Таксономія – механізм, який дозволяє створювати довільну кількість зв'язків між об'єктами вмісту сайту та асоціювати їх з "Ключами записів", "Категоріями записів" або "Категоріями посилань".

Від інших CMS, WordPress відрізняється своїм гранично простим інтерфейс. Для програміста це, можливо, не буде гарантією всієї повноти зручностей, але адміністратору веб-ресурсу це значно полегшить роботу. Публікації створюються миттєво, сама платформа встановлюється швидко. Зазвичай час установки займає трохи більше п'яти хвилин.

На WordPress працюють такі ресурси, як електронна бібліотека WebReading та російський блог «Лабораторії Касперського». Великі портали, такі як eBay, New York Times і CNN, використовують CMS для блог директорії та стрічки новин.

WordPress має ряд своїх можливостей та відмінних особливостей, таких як:

- простота в установці CMS та початковому налаштуванні сайту;
- підтримка трансляції RSS-каналів, за допомогою яких надається короткий опис нової інформації, що з'являється на сайті;
- можливість реєстрації відвідувачів сайту, а також підтримка розрахованого на багато користувачів режиму, завдяки якому кожен відвідувач має можливість вести свій власний блог на сайті;
- підтримка «тегів», наявність яких є стандартом будь-якого сучасного сайту;
- користувачі можуть коментувати матеріали сайту, а адміністратори мають можливість фільтрації цих
- коментарів від різного спаму;
- підтримка завантаження на сайт фалів різних медіаформатів;
- поштова передплата для відвідувачів сайту;
- підтримка додаткових модулів, бібліотека яких постійно поповнюється;
- можливість доопрацювання вже наявних модулів або створення власних.

Це далеко не повний перелік можливостей, що є в системі керування контентом WordPress. Оскільки система є самою популярною у світі, то відповідно, має ряд переваг, через яких користувачі вибирають саме цю систему.

Головною відмінністю CMS WordPress і CMS Joomla є їх орієнтованість. WordPress більше підходить для простих сайтів, таких як блоги, "сайти-візитки". На Joomla створюється більше звичайних інформаційних веб-сайтів. У проектній частині диплома створюватиметься саме інформаційний сайт, тому для його створення буде обрано CMS "Joomla". Головною відмінністю CMS WordPress і CMS Joomla є їх орієнтованість. WordPress більше підходить для простих сайтів, таких як блоги, "сайти-візитки". На Joomla створюється більше звичайних інформаційних веб-сайтів. У проектній частині диплома

створюватиметься саме інформаційний сайт, тому для його створення буде обрано CMS "Joomla".

Назва	Переваги	Недоліки
Joomla!	Система шаблонів дозволяє легко змінювати зовнішній вигляд Всі компоненти, шаблони плагіни можна написати самому Відредагувати розширення на свій розсуд	Потребує БД MySQL Неможливо запустити сайт на безкоштовному хостингу Надмірно велика кількість налаштувань, опцій та можливостей
WordPress	Є безкоштовним Багато плагінів та шаблонів Вимоги до хостингу мінімальні Зрозуміла та зручна консоль адміністрування	Потребує бази даних MySQL Надмірно велика кількість налаштувань, опцій та можливостей

Таблиця 1.1 – Порівняльняння аналізованих CMS

2 РОЗРОБКА WEB-ЗАСТОСУНКУ

2.1 Робота розробленої спеціалізованої програмної системи керування контентом

Спеціалізована система керування контентом веб-сайтів – це система розширеної функціональності для організації інформаційних ресурсів у мережі, архітектура якої має три рівні ієрархії

Будь-який сайт – це, перш за все, згрупований певним чином контент. Всі ці меню, блоки, графічні оформлення служать для того, щоб зручно організувати інформацію та підкреслити найцікавіші моменти. Виходячи з цього, дамо визначення CMS (Content Management System) – це сімейство платформ, завдяки яким можна доволі зручно створити сайт, тобто належним чином керувати контентом. Системи управління контентом на сьогоднішній день займають міцну позицію серед різноманітних сайтбілдерів.

CMS також різноманітні, як і веб-сайти, для яких вони використовуються. Drupal непростий для початківців, а WordPress для повноцінної роботи потребує встановлення десятка плагінів. Що ж до MODx, то її частіше вибирають професійні веб-дизайнери. Нижче ми наведемо кілька практичних порад щодо вибору відповідної системи

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

Визначення часу навчання. При повній відсутності досвіду, ви найшвидше освоїте WordPress, на другому місці по легкості стоїть Joomla, потім Drupal. MODx або TYPO3 складніші в освоєнні.

Визначення часу зайнятості. Тут картина аналогічна. Найпростіше адмініструвати WordPress, а найскладніше – MODx.

Визначення ефективності SEO. В принципі, рівень якості SEO не так і сильно залежить від CMS, проте WordPress лідирує за кількістю плагінів для оптимізації.

Система управління контентом – це спеціальна програма (сайтовий движок), яка встановлюється на хостинг (віддалений сервер, під'єднаний до Інтернету), і покликана виконувати 2 функції:

Показувати сторінки сайту користувачам, динамічно формуючи їх вміст із заздалегідь визначених шаблонів оформлення контенту (тобто текстів, малюнків, таблиць та інших матеріалів, що знаходяться у базі даних). Важливо розуміти, що при такій схемі сайту як набору сторінок не існує. Окремо існує дизайн (шаблон) і набір різноманітного типу матеріалів (файли, текст і т. п.) CMS будує сторінку користувачу у момент його запиту. Тобто движок генерує сторінки в залежності від внесених до матеріалів змін, а також індивідуального статусу користувача (наприклад, вміст кошика у випадку магазину). При цьому CMS намагається обробити якнайбільше запитів за одиницю часу, а також заважає спамерам засмічувати базу даних, слідкує за безпекою та виконує у фоновому режимі безліч інших дій, життєво необхідних для нормальної роботи сайту.

Допомогти користувачу, який не має навичок програмування, просто і зручно керувати сайтом: публікувати сторінки, новини, викладати відео, робити посилання та багато іншого. Коротше кажучи, CMS забезпечує можливість створення сайту для більшості необізнаних у цьому питанні людей.

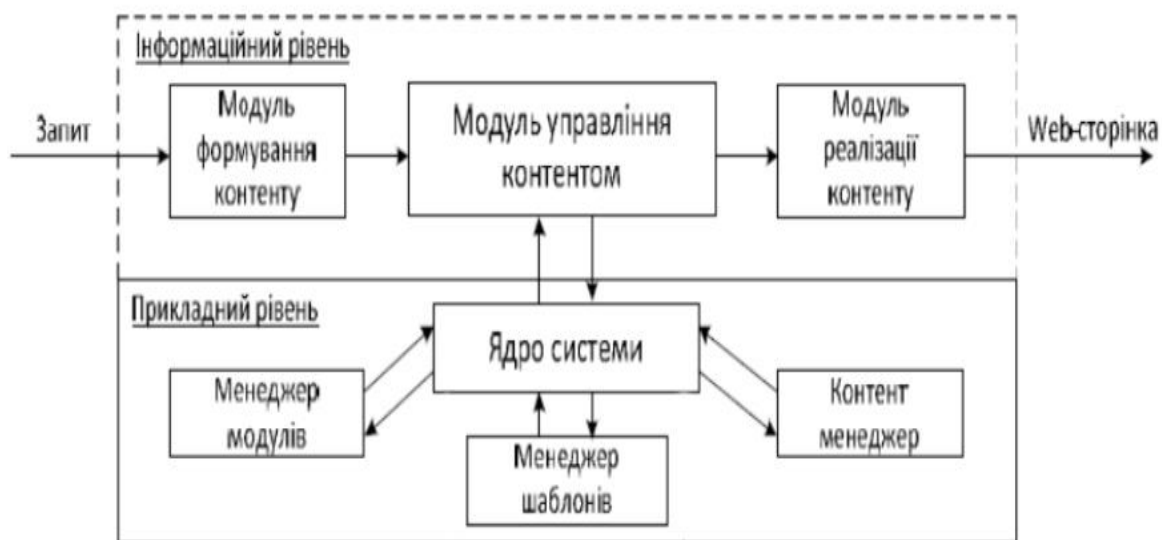


Рисунок 2.1 Архітектура спеціалізованої CMS системи

Архітектура розробленої спеціальної програмної системи складається з прикладного, інформаційного й логічного рівня.



Рисунок 2.2 Архітектура спеціалізованої програмної системи на прикладному рівні

Це надає можливість впроваджувати незалежність збереженої інформації від додатків, що їх використовують.



Рисунок 2.3 Діаграма процесу подання контенту

2.2 Розробка інтерфейсу web-застосунку

Інтерфейс користувача – засіб зручної взаємодії користувача з інформаційної системи Сукупність засобів обробки та відображення інформації, що максимально пристосована для зручності користувача; в графічних системах інтерфейс користувача реалізується багатовіконним режимом, зміною кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їх розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих їх елементів (файли, папки, ярлики, шрифти і т.п.), доступністю розрахованих на багато користувачів налаштувань.

Графічний інтерфейс користувача – тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні

зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

Для забезпечення зручної роботи користувачів, інтерфейс повинен включати наступні елементи:

- блок ідентифікації сайту - частина графічного інтерфейсу, що дозволяє відвідувачеві однозначно визначити, що являє ресурс, яким він користується. Являє собою логотип (знак) сайту, його назва та, можливо, короткий опис.
- гіперпосилання - частина документа, що посилається на інший елемент (текст, заголовок, примітка, зображення) в самому документі, на інший об'єкт (файл, директорія, додаток), розташований на локальному комп'ютері або в комп'ютерній мережі, або на елементи цього об'єкта.
- навігація - елемент інтерфейсу, що дозволяє виконувати перехід користувача до певних структурних частин (сторінок) сайту. Складається з головного і додаткового меню і являє собою набір гіперпосилань.
- контент - власне сам зміст сайту є сукупністю текстової та графічної інформації.

Оперуючи цими поняттями, можна описати зовнішній вигляд системи, що відповідає вимогам зручності використання. Власне проектування графічного інтерфейсу для сайту являє собою організацію всіх необхідних елементів на сторінці, а так само розробку їх зовнішнього вигляду.

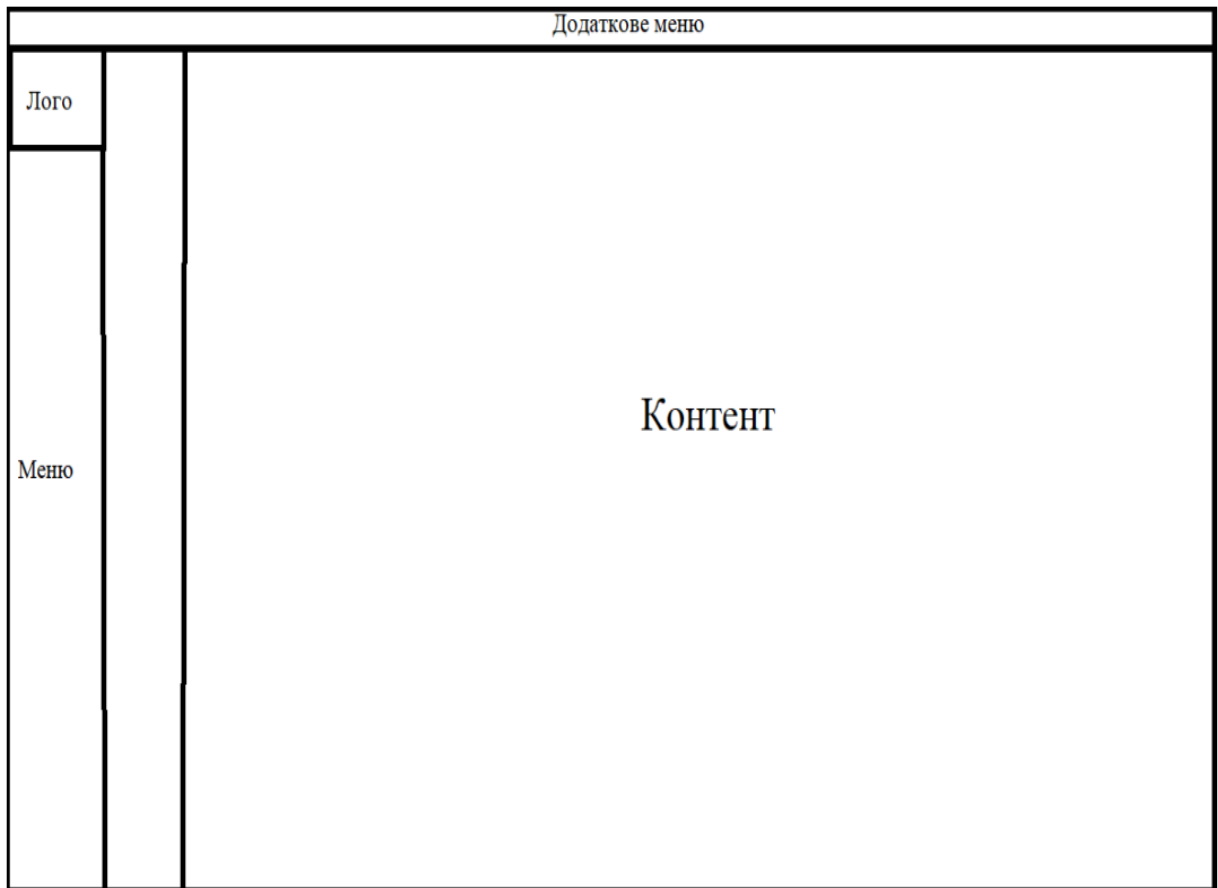


Рисунок 2.4 структура інтерфейсу

Користувацький інтерфейс веб-додатку складається з двох основних модулів:

- Головне меню
- Робоча область

2.3 Аналіз процесу впровадження

Розробка системи – це багатоетапний творчий процес, у якому беруть участь дизайнери, програмісти, фахівці з верстки та менеджменту проектів.

Створення індивідуальної системи управління контенту сайту – важливий крок, який дозволить розширити бізнес та отримати нових клієнтів.

Найкращі веб-сайти – це ті сайти, які мають власну індивідуальну систему керування контентом. Розробка структури є першочерговим кроком розробки програмного забезпечення продукту. Модель програмного продукту є певним каркасом, заготівлю, яка згодом наповнюється графічними та інформаційними зображеннями матеріалами.

Модель системи має визначатися ще на перших етапах створення розробки на початок розробки дизайну. І в майбутньому, у міру зростання сайту, веб-майстри повинні суворо її дотримуватися. В цьому випадку можна бути впевненим, що розробка буде більше схожа на доглянутий парк із прокладеними всюди стежками, а не на лісову хащу, в якій відвідувачі змушені будуть пробиратися до мети по справжньому бурілому.

Однак для цього необхідно розібратися, які структури систем існують і для яких проєктів вони найбільше підходять. Система, що розробляється, призначена для розробки, підтримки, наповнення сайту та подальшого управління ім. Вона передбачає забезпечення безпеки, тобто авторизацію та аутентифікацію користувачів, можливість керувати динамічними та статичними ресурсами, взаємодіяти з пошуковими системами та іншими системами.

Система складається з великої кількості складних, автономних компонентів, кожен з яких може значною мірою впливати на стабільність роботи системи в цілому.

Для розробки необхідно пройти такі етапи:

- розділити систему на окремі відносно автономні частини;
- для кожної з них виділити головні компоненти, які будуть реалізовані в тому чи іншому модулі;
- вибрати, які компоненти мають подібні характеристики, та об'єднати їх у базові компоненти;
- Визначити, в яких взаємозв'язках компоненти знаходяться один з одним;
- Побудувати моделі інших модулів;
- Визначити зв'язки між різними модулями;
- Скомпонувати загальну структуру системи

Розробка якісного web-застосунку досить нелегка, для цього будуть потрібні не тільки навички дизайнера для красивого та стильного оформлення. Складність та обсяг програм, необхідних для створення того чи іншого сайту,

залежить від багатьох факторів: структура, мета, необхідна функціональність, забезпечення надійності тощо.

Шаблон – це текстовий файл (.html) зі шматком коду HTML, який визначає дизайн певної частини сторінки та перебувати у спеціальному каталозі. Сторінки сайту умовно поділені на шість зон (заголовок, ліва колонка, центр, права колонка, нижня лінія та підвал), що становлять головний шаблон (каркас). Для кожної зони своя група шаблонів умовно розділена, якою визначає лише дизайн та назва файлу.

Тобто сторінка може бути побудована як мінімум з одного шаблону (наприклад заголовка), і як максимум із шести. Склад шаблонів зберігається в межах розділу та обумовлює зовнішній вигляд у ньому. Різні розділи можуть включати відповідно різні шаблони і модулі, якщо у цьому є потреба.

Спеціальний каталог зі скриптами php є модулем і містить основні файли включаються (скрипти) для сайту та адмін-панелі (властивості та настройки модуля).

Складним питанням став механізм включення модулів до основного виконуваний файл. Розглядалися різні варіанти реалізації цього процесу, але довелося зупинитися на «напівавтоматичному». Для вставки модуля в тому іншому місці сторінки, необхідно прописувати спеціальні «мітки» (у вигляді спеціального HTML коментаря), які в процесі обробки (2) замінювалися змістом індексного файлу модуля і вбудовувалися в загальний код, що виконується. А вже в адмін-панелі CMS адміністратор визначає, який модуль, куди і в який розділ необхідно встановити.

Тому для розробки було використано такі ресурси, як: HTML, CSS, JavaScript, BOOTSTRAP, , VebPage v.1.1.

HTML – мова є базовою в галузі технологій створення сайтів, оскільки відносно легка в освоєнні. Але надмірна простота є його недоліком. HTML чудово відповідав вимогам раннього періоду розвитку технологій створення сайтів, але з подальшим його розвитком виникли суттєві проблеми.

Мова HTML призначена для формування та оформлення сторінок тексту, яка призначена для перегляду в мережі Інтернет за допомогою спеціальних програм – браузерів.

Для формування файлу HTML немає необхідності використання потужного текстового процесора, зручніше використання, наприклад, редактора Notepad++. Вміст HTML-сторінок представлений у гіпертекстовому вигляді, що передбачає наявність у документі, що відображається на екрані, виділених певним чином місць, клацнувши мишею по яких можна переміститися в іншу частину цієї ж сторінки або на іншу сторінку.

Все що необхідно, щоб прочитати HTML-документ - це web-браузер, який інтерпретує HTML-теги і відтворює на екрані документ у вигляді, який йому надає автор. HTML дозволяє відзначити, де в документі має бути заголовок або абзац за допомогою тега, а потім надає web-браузер інтерпретувати ці теги. Основна перевага HTML у тому, що документ може бути переглянутий на веб-браузерах різних типів та на різних платформах [15].

CSS (Cascading Style Sheets) – каскадні таблиці стилів – спроба відокремити деталі дизайну сторінки від її структури та змісту. У класичному HTML структура та дизайн були упереміш: поруч із текстовим абзацом його колір, розмір шрифту, тип шрифту тощо. Якщо потрібно було змінити такий простий параметр, як розмір шрифту на своєму сайті, доводилося переписувати всі сторінки. Команди CSS дозволяють призначити всім об'єктам стиль, опис якого може зберігатися взагалі окремому файлі. Використовуючи декодер CSS, можна змінити розмір шрифту у всіх сторінках сайту, виправивши лише один файл із описом стилів.

Чим більше різних стилів використано, тим цікавіше змінювати дизайн. Найголовніше, що не доведеться змінювати готові HTML документи, що лежать на сервері. Браузер користувача сам звернеться до файлу (за посиланням) зі стилями і додасть сторінці потрібного вигляду [16].

JavaScript – скриптова мова програмування. Зазвичай використовується як вбудована мова для програмного доступу до об'єктів програм. Найбільш

широке застосування знаходить у браузерях як мову сценаріїв для надання інтерактивності web-сторінкам. Для написання програми можна використовувати NotePad++ - вільний текстовий редактор з відкритим вихідним кодом для Windows, з підсвічуванням синтаксису великої кількості мов програмування та розмітки. Базова функціональність програми може бути розширена як за рахунок плагінів, так і сторонніх модулів, таких як компілятори та препроцесори.

JavaScript та Java схожі в одних відносинах, але фундаментально відмінні в інших. Javascript нагадує Java, але не має статичної типізації та суворої перевірки типів, якими володіє Java. JavaScript слід здебільшого синтаксису Java у виразах, угодах про іменування та основного потоку управління конструкціями, тому він був перейменований з LiveScript на JavaScript.

На відміну від системи скомпілованих класів Java, побудованої на оголошеннях, JavaScript підтримує систему виконання, засновану на невеликій кількості типів даних, що представляють числові, логічні та строкові значення. JavaScript має об'єктну модель на основі прототипів замість більш загальної моделі на основі класів. Модель об'єктів з урахуванням прототипів надає динамічне успадкування, тобто. те, що успадковується, може відрізнитися окремих об'єктів. JavaScript також підтримує функції без будь-яких спеціальних декларативних вимог. Функції можуть бути властивостями об'єктів, які виконуються як вільно-типізовані методи.

У порівнянні з Java, JavaScript є мовою з дуже вільною формою. Вам не треба оголошувати змінні, класи та методи. Вам не треба турбуватися про те, чи є методи публічними (public), приватними (private) або захищеними (protected), а також вам не треба реалізовувати інтерфейси. Змінні, параметри та функціями, що повертаються, типи не є явно типізованими.

Java — це мова програмування, заснована на класах і призначена для швидкого виконання та безпеки типів. Безпека типів означає, наприклад, що ви не можете привести тип integer до типу object reference або отримати доступ

до приватної пам'яті, змінюючи байт-код Java. Орієнтована на класи модель Java означає, що програми складаються виключно з класів та їх методів. Успадкування класів і сувора типізація Java зазвичай вимагають тісно пов'язані ієрархії об'єктів. Ці вимоги роблять програмування Java більш комплексним, ніж програмування JavaScript.

За духом JavaScript походить від невеликих, динамічно типізованих мов, таких як HyperTalk та dBASE. Ці скриптові мови пропонують інструменти програмування для більш широкої аудиторії завдяки більш простому синтаксису, спеціалізованій вбудованій функціональності та мінімальним вимогам для створення об'єктів.

Майже всі браузерери в даний час підтримують CSS та багато інших програм. Щоб створити CSS, не потрібно нічого, крім текстового редактора, але є багато інструментів, які ще більше спрощують цей процес. В даний час більшість веб-інструментів для розробки забезпечують деяку підтримку таблиць стилів CSS.

2.4 Ядро web-застосунку

В основі web-застосунку покладено opensource проект VebPage v1.1.

Багато хостерів як додаткову послугу пропонують клієнтам Sitebuilder (якщо дослівно перекласти на російську, вийде сайтобудівник). Це дуже зручна програма, яка допомагає самотійно створити веб-ресурс, модернізувати, модифікувати його, а також керувати ним. Платний хостинг пропонує таку послугу зовсім недорого, зате переваг вона має масу хоча б перед тим самим SMS.

VebPage дозволяє видаляти, додавати нові сторінки чи інтерактивні модулі, розширювати функціональність сайту. Бракує на ресурсі гостьової книги, стрічки новин чи якоїсь смуги? Із сайтобудівником, який надає веб-хостинг, це не проблема. Буквально за кілька хвилин шляхом кількох натискань на кнопку мишки ваш сайт буде доповнено необхідними модулями.

Це ядро дає багато можливостей, а саме:

- Мегашвидке створення сайту – лише за 5 хвилин.

- Простота в управлінні - не потрібно знання мов програмування, адміністрування, а також інших новітніх комп'ютерних технологій.
- Найчастіше доступні готові шаблони для технічного та дизайнерського оформлення ресурсу.
- Доступні різні види вбудованих послуг (статистика та інші).
- Можливість редагування у режимі WYSIWYG, зрозумілий візуальний інтерфейс.
- Доступні готові модулі функціональності.

Будь-який дизайн легко змінюється шляхом встановлення своїх кольорів, заміни зображень та додавання необхідних блоків. Вибраний дизайн можна змінити невідомо, забезпечивши сайту унікальний зовнішній вигляд.

Процес створення сайту у візуальному редакторі дуже простий та нагадує роботу з текстовим редактором. Користувачу не потрібні технічні фахівці та спеціальні навички. Достатньо перетягнути необхідні блоки на сторінку сайту та заповнити їх своєю інформацією.

Програмне забезпечення, що вільно розповсюджується, з відкритим вихідним кодом (open source) — це програми, чий код доступний усім. Їх можна завантажити та/або використовувати на будь-якій кількості пристроїв. Ви можете взяти код програми і робити з ним все, що захочете, а потім поширити його серед ваших знайомих. Це так, тому що такі програми поширюються на вільні ліцензії, наприклад, ліцензії MIT.

Його переваги в тому, що:

- Зручний інтерфейс (Drag-and-drop);
- Розрахований на багато користувачів, клієнти реєструються і створюють безлімітну кількість сайтів;

- Редагування. Можна редагувати як блок цілком і його окремі деталі (меню, текст, картинки, відео тощо.), змінювати стилі, кольори, розміри.
- Готові шаблони сайтів.

Для клієнтської сторони сайту, тобто розмітки, використовувався фреймворк bootstrap - безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків.

Фреймворки - це програмні продукти, які спрощують створення та підтримку технічно складних чи навантажених проєктів. Фреймворк, зазвичай, містить лише базові програмні модулі, проте специфічні для проєкту компоненти реалізуються розробником з їхньої основи. Тим самим досягається не лише висока швидкість розробки, а й велика продуктивність та надійність рішень.

Веб-фреймворк – це платформа для створення сайтів та веб-застосунків, що полегшує розробку та об'єднання різних компонентів великого програмного проєкту. За рахунок широких можливостей у реалізації бізнес-логіки та високої продуктивності ця платформа особливо добре підходить для створення складних сайтів, бізнес-додатків та веб-сервісів.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері. Репозиторій із цим фреймворком є одним із найпопулярніших на GitHub.

Основні інструменти Bootstrap:

Сітки — заздалегідь задані розміри колонок, які можна відразу використовувати, наприклад, ширина колонки 140 px відноситься до класу .span2 (.col-md-2 в третій версії фреймворку), який можна використовувати в CSS-описанні документа.

Шаблони – фіксований або гумовий шаблон документа.

Типографіка - опис шрифтів, визначення деяких класів для шрифтів, таких як код, цитати тощо.

Медіа — надає певне керування зображеннями та відео.

Таблиці — засоби оформлення таблиць, до додавання функціональності сортування.

Форми - класи для оформлення форм та деяких подій, що відбуваються з ними.

Навігація – класи оформлення для панелей, вкладок, переходу по сторінках, меню та панелі інструментів.

Алерти - оформлення діалогових вікон, підказок та спливаючих вікон.

Для написання серверної частини сайту використано мову PHP – скриптована мова програмування, яка була створена для генерації HTML-сторінок на стороні веб-сервера. Вона інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Це мова, у код якої можна вбудовувати безпосередньо html-код сторінок, які, у свою чергу, коректно оброблюватимуться PHP-інтерпретатором.

PHP – скриптова мова загального призначення, що інтенсивно застосовується для розробки сайтів та веб-додатків. Специфіка PHP дозволяє поєднувати його та HTML розмітку сторінки. Завдяки простому синтаксису, він є лідером на ринку рішень для розробки веб-додатків.

Однак, PHP має ряд серйозних недоліків. Одним із головних недоліків є синтаксична нагромадженість. При порівняно простому синтаксисі, в екосистемі мови присутня велика кількість функцій, які виконують одну і ту ж дію, при цьому маючи різні назви та різний порядок параметрів, що передаються.

2.5 Допрацювання ядра web-застосунку

В допрацюванні ядра під свої потреби стояли такі задачі, як:

- Експорт сайту. Збереження сайту після збирання на комп'ютері;
- Розмітка сайту для адаптації під мобільні пристрої;
- Додавання контенту для SEO допомоги;
- Реалізація зміни елементів готових шаблонів;

Для реалізації веб-застосунку було вирішено використовувати зовнішній файл скриптів. JavaScript застосовується для створення різних анімацій, меню, що випадає або слайдера, суть яких полягає в тому, щоб в певному обмеженому просторі відбувалося чергування різних об'єктів.

Часто JavaScript використовується і для первинної перевірки даних, які користувач вводить в форми. Існує велика кількість бібліотек та фреймворків. Щоб спростити код ми будемо використовувати фреймворк Angular 10. Одне з головних переваг бібліотек та фреймворків це крос-браузерні інтерфейс до методів DOM. До того ж фреймворки позбавляють веб-розробників від необхідності вивчати в деталях сам JavaScript та TypeScript, надаючи ряд зручних у використанні інструментів, що дозволяють з легкістю управляти об'єктною моделлю документа.

JavaScript - це сценарій клієнтської мови, яка використовується для створення веб-сторінок. Це окрема мова, розроблена в Netscape. Він використовується, коли веб-сторінку потрібно зробити динамічною та додати спеціальні ефекти на такі сторінки, як перекидання, розгортання та багато видів графіки. Він здебільшого використовується всіма веб-сайтами з метою перевірки. На додаток до валідацій, він підтримує зовнішні програми, такі як PDF-документи, запущені віджети, підтримує флеш-програми тощо. Він також може завантажувати вміст у документ, коли користувач цього вимагає, навіть не перезавантажуючи всю сторінку.

Завдяки технологіям браузерів та персональних комп'ютерів вдосконалено настільки, що для створення надійних веб-додатків потрібна мова. Коли користувач досліджує карту на Картах Google, користувачеві потрібно просто натиснути та перетягнути мишу. Весь детальний вид видно лише одним клацанням миші. Це можливо завдяки JavaScript. Він взаємодіє з браузером, не надсилаючи повідомлення туди-сюди на сервери. JavaScript використовує інтерфейси програмування програм (API), які надають додаткові повноваження коду.

JavaScript сам по собі досить компактний, але дуже гнучкий. Розробниками написано велику кількість інструментів поверх основної мови JavaScript, які розблокують величезну кількість додаткових функцій із дуже невеликим зусиллям. До них відносяться:

Програмні інтерфейси програми (API), вбудовані в браузери, що забезпечують різні функціональні можливості, такі як динамічне створення HTML та встановлення CSS стилів, захоплення та маніпуляція відеопотоком, робота з веб-камерою користувача або генерація 3D графіки та аудіо семплів.

Сторонні API дозволяють розробникам впроваджувати функціональність у свої сайти від інших розробників, таких як Twitter або Facebook.

Також ви можете застосувати до вашого HTML сторонні фреймворки та бібліотеки, що дозволить вам прискорити створення сайтів та програм.

Верстка сайту є описом програмним кодом візуальної частини веб-сайту. Сайт повинен виглядати та працювати коректно на будь-якому моніторі, незалежно від його дозволу, і в будь-якому браузері.

Існує кілька підходів до верстки сайту:

- фіксована верстка – при зміні розміру вікна браузера блоки не змінюють свого розміру, що може призвести до появи смуги прокручування;
- гумова верстка – розмір блоків змінюється при зміні розміру вікна браузера;

- адаптивна верстка – створюється на основі різних скриптів, визначено для заданого дозволу, наприклад 320, 768, 1024, тому зміна розмірів відбувається ривками, після досягнення певного дозволу;
- чуйна верстка - злиття гумової та адаптивної верстки.

Для створюваного проекту застосуємо гумову верстку, тобто, розміри основних блоків будуть залежати від розмірів вікна браузера. Це дозволить коректно відображати інформацію сайту на моніторах, що мають різний дозвіл. Для цього розміри блоків будемо розраховувати у відсотках, щодо розмірів вікна браузера.

Для розмітки web-застосунку під свої потреби, а саме розмітки сайту для адаптації його на мобільних пристроях був вибраний фреймворк bootstrap. Bootstrap розроблявся як mobile first, тобто, його налаштування насамперед оптимізовано під мобільні пристрої, а вже потім за допомогою медіа-запитів підгоняється масштаб компонентів, як нам необхідно на інших пристроях.

Для більшої простоти масштабування CSS ми змінюємо глобальне значення box-sizing з content-box на border-box. Це гарантує, що padding не впливатиме на кінцеву обчислену ширину елемента; однак може створити проблеми для сторонніх програм, таких як Google Maps або пошуковий движок Google. Цей фреймворк чудово забезпечує уся необхідні функції, які необхідні були для розгортання проекту.

Уся розмітка сайту була виконана для одній сторінці, для полегшення роботи користувача з нею.

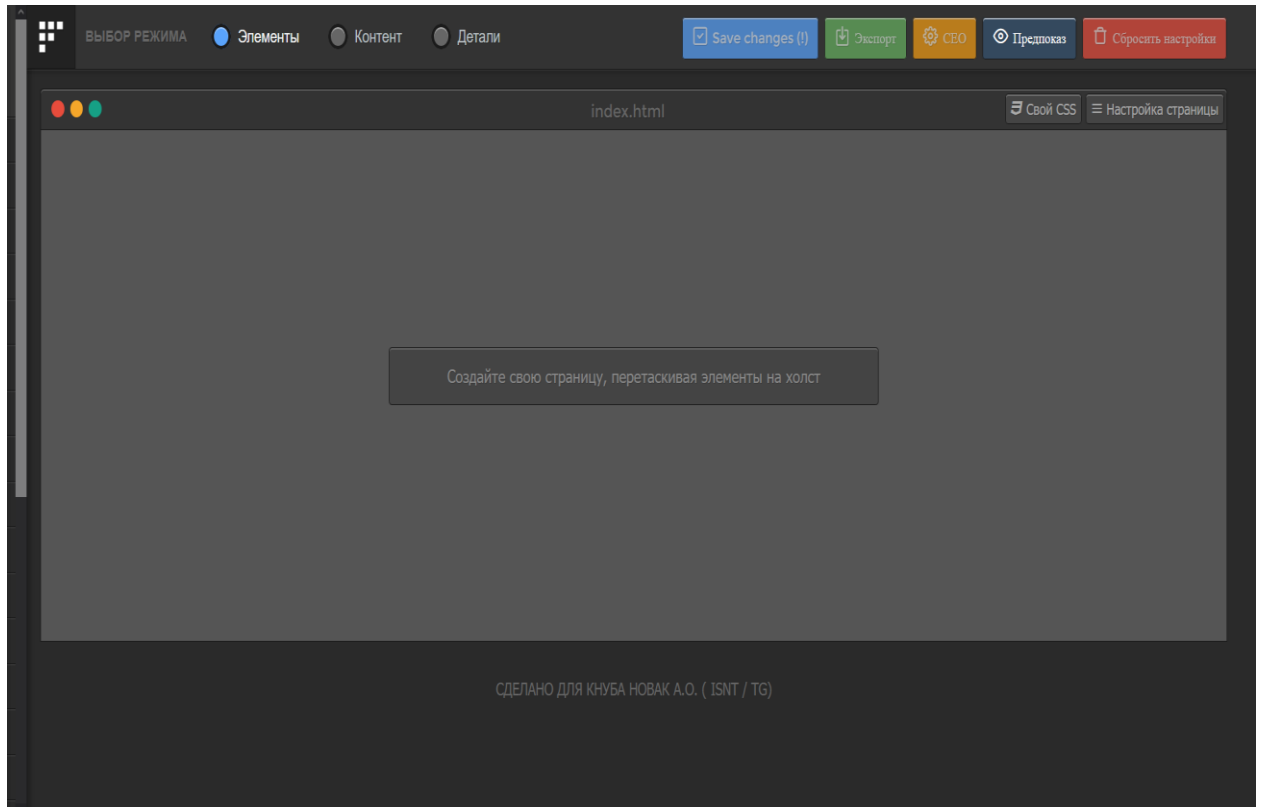


Рисунок 2.5 Головна сторінка сайту

Для забезпечення експорту проекту, було вибрана мова програмування PHP. Код експорту наведений в додатку А1. Експорт проекту користувача відбувається досить просто. В результаті експорту у клієнта буде готовий проект в .zip файлі, який він може переслати на свій FTP сервер та працювати з ним.

FTP або File Transfer Protocol — це один із базових протоколів передачі файлів, призначений для передачі файлів у мережі між комп'ютерами. З його допомогою можна підключатися до FTP-серверів, переглядати вміст їх каталогів та завантажувати файли з сервера чи сервер.

FTP, або File Transfer Protocol, у перекладі з англійської мови означає "протокол передачі файлів" і призначений саме для цієї процедури - передачі файлів у комп'ютерних мережах. З його допомогою можна підключитися до FTP-серверів для перегляду вмісту в їх каталогах і завантажувати файли на сервер та сервер. Така передача набагато швидше, ніж з'єднання через протокол http, тому її вигідно використовувати для завантаження файлів

великих розмірів, наприклад, ігор або фільмів. Веб-майстри, завантажуючи сторінки сайту на сервер, активно використовують FTP. Робота з ним схожа на роботу зі звичайними файловими менеджерами, наприклад, популярним Total Commander.

FTP зазвичай асоціюється зі скачуванням музики, відео чи картинок. Насправді його можна використовувати для різних типів файлів як невеликих текстових документів, так і спеціальних захищених програм. Для отримання доступу до FTP потрібна спеціальна програма-клієнт, яка є частиною практично всіх сучасних браузерів. Все, що потрібно зробити – це набрати в рядку адреси необхідну URL-адресу, яка починається, як правило, з ftp:// і закінчується потрібним доменним ім'ям.

Недолік такого способу підключення полягає в тому, що у разі урвища зв'язку відновити закачування буде неможливо. Отже, доведеться почати все спочатку, а це проблематично при роботі з великим обсягом інформації. Тому набагато вигідніше користуватися окремими, просунутішими версіями FTP-клієнта, які дозволяють завантажити, завантажити або поставити файл у чергу на завантаження, як на сервер, так і навпаки. На даний момент існує безліч платних та безкоштовних FTP-клієнтів для різних операційних систем.

Будь-який FTP-сервер вимагає обов'язкової авторизації користувача, тобто введення логіну та паролю. Користувальницькі права доступу встановлює адміністратор сайту. Як правило, гостям сервера файли можна лише качати, але не можна додавати, перейменовувати чи видаляти. Але на багатьох серверах FTP є відкриті для запису каталоги, які призначені для закачування файлів на сервер, що дозволяє оперативно поповнювати його свіжими даними.

Для незареєстрованих відвідувачів практично всі FTP сервери надають можливість анонімного з'єднання. Для цього при запиті логіну вводять "anonymous". У рядку введення пароля можна набрати будь-яку інформацію. Однак при з'єднанні з сервером зазвичай прийнято вводити свою електронну адресу. Після авторизації з'являється запрошення сервера (зазвичай це ftp).

У разі серйозного наміру працювати з FTP-протоколом необхідно вибрати якісний і надійний FTP-сервер, який дає змогу керувати доступом відповідно до типів файлів та категорій користувачів. За наявності загальнодоступних файлів необхідна можливість надання анонімного FTP доступу до вибіркових каталогів. У адміністратора повинен бути повний доступ до всієї інформації, що зберігається на хостингу. При цьому бажано, щоб FTP-сервер мав доступний і зрозумілий інтерфейс - це привабить користувачів.

На практиці зазвичай FTP використовують тільки для завантаження файлів на сервер, тому що завантажувати набагато зручніше браузером. Якщо FTP з невідомих причин не працює, варто перевірити налаштування FireWall. Можливо, вони передбачають використання виключно браузера. У такому випадку необхідно вибрати хостинг, який дозволить закачувати файли не тільки через FTP, але й за допомогою електронної пошти або через особливу форму. Кожен сервер такого типу, як правило, містить посібник із закачування файлів.

Також через цей PHP скрипт можна імпортувати заздалегідь збережений проект. Нажаль підтримується проекти лише збережені через цей веб-застосунок.

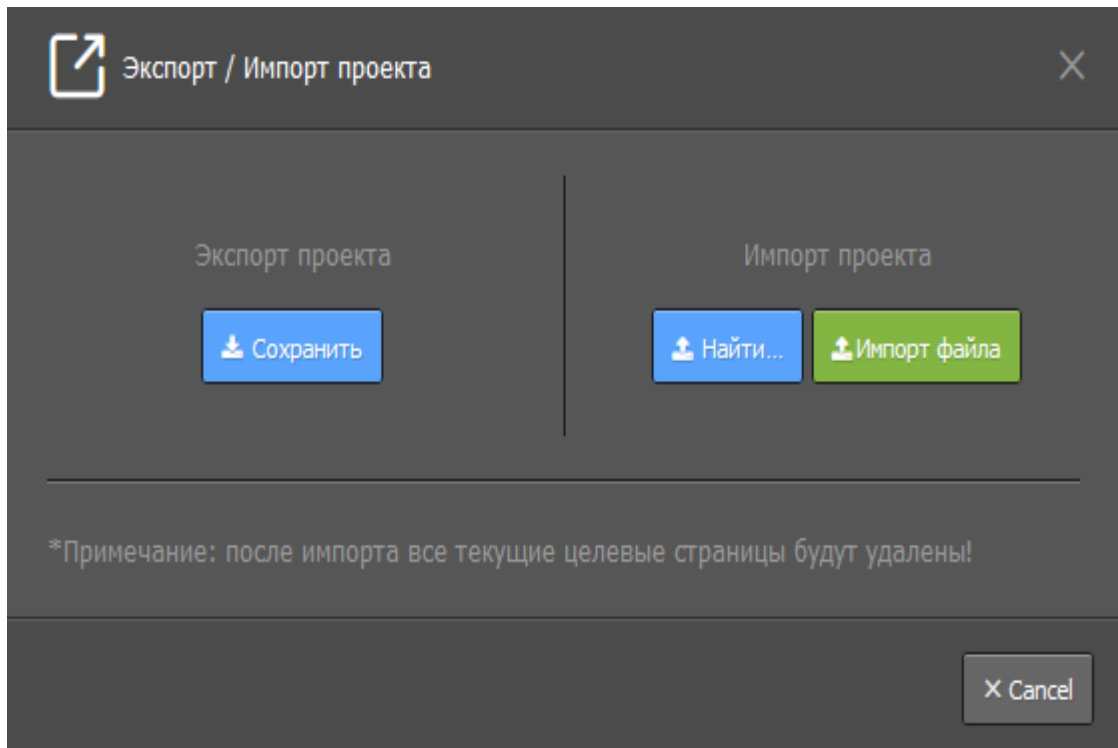


Рисунок 2.6 Модальне вікно для експорту або імпорту проекту

Для збереження проекту був використаний скрипт під назвою `savePage`. Який визиває функцію `saveTextAsFile`.

```

$('#saveprojPage').click(function(e){
    $('#savePage').click();
    if (navigator.appVersion.toString().indexOf('.NET') > 0){

saveTextAsFile('project.pixbuilder',JSON.stringify(localStorage));
        }else{
            console.save(JSON.stringify(localStorage),'project.pixbuilder');
        }
    });

```

Для збереження проекту у готовому вигляді для імпортування його на свій домен був використаний скрипт. За його допомогою уже, що знаходиться у головному макеті(pageContainer) буде експортовано через функцію newDocMainParent. Детальніше скрипт описаний у додатку А.2.

Для необмеженого додавання сторінок в проект було вирішено додати додаткову кнопку, яка автоматично додає нові сторінки до проекту. Оскільки сесія клієнта зберігається за допомогою opensource проекту VebPage v1.1, який використовується як основа проекту. Для додавання був введений скрипт, описаний у додатку А.3.

В ньому порівнюється кількість уже збережених сторінок у сесії клієнта та додається нова сторінка. Усі вони записуються у масив через ідентифікатор new_page_id.

```

$('#addPage').click(function(e){

    e.preventDefault();

    $('#pages li.active').each(function(){

        if( $(this).find('input').size() > 0 ) {

            theLink = $('<a
href="#">'+$(this).find('input').val()+'</a>');

            $(this).find('input').remove();

            $(this).prepend( theLink );

        }

    })

```

Все це згорнуто у функцію addPage через ідентифікатор кнопки на головній сторінці сайту під назвою addPage.

Також вирішено було додати функцію для полегшення роботи SEO спеціалістів, це все було реалізовано за допомогою скрипта, який був записаний у функцію seo_arr.

Для початку його роботи, необхідно щоб хоча б один елемент був добавлений у проект. Далі активується кнопка редагування SEO контенту, тобто додання заголовків, мета-тегів, ключових слів, а також додаткового кода, який необхідно додати. На кнопку був доданий ідентифікатор seoSubmitButton, який потім визивається через скрипт в окремому файлі.

```
$('#seoSubmitButton').click(function(e){
    var seo_arr = [$('#pageData_title').val(),
        $('#pageData_metaDescription').val(),
        $('#pageData_metaKeywords').val(),
        $('#pageData_headerIncludes').val(),
    ];
    localStorage['seo_'+$('#pageTitle span span').text()] =
JSON.stringify(seo_arr);
    temp_seo = JSON.parse(localStorage['seo_'+$('#pageTitle span
span').text()]);
});
```

За допомогою цього скрипта дані будуть додаватись автоматично, за потребою користувача, до тегу <head> в проекті, який створює користувач.

Для реалізації зміни готових елементів використовувався скрипт, який передавався на кнопку через ідентифікатор saveStyling. За допомогою його перезаписувались дані з випадаючого вікна до коду в макеті.

```
$('#button#saveStyling').unbind('click').bind('click', function(){
```

```

$('#styleEditor #tab1 .form-group:not(#styleElTemplate) input,
#styleEditor #tab1 .form-group:not(#styleElTemplate) select').each(function(){
    $(el).css( $(this).attr('name'), $(this).val());

    sandboxID = hasSandbox($(el))
    if( sandboxID ) {
        elementID = $(el).attr('id');

        $('#'+sandboxID).contents().find('#'+elementID).css(    $(this).attr('name'),
$(this).val() )
    }
})

```

2.6 Вибір хостинга

Коли перед користувачем стоїть питання вибору хостингу – платного чи безкоштовного, всі насамперед схиляються до другого варіанту, намагаючись заощадити. Проте мало хто замислюється, які переваги має платний хостинг.

Платний хостинг – це максимально зручний варіант, який надає користувачеві набір додаткових послуг, недоступних у безкоштовному варіанті. Властивості та можливості платного хостингу дещо відрізняються залежно від компаній, що їх пропонують, проте середній їх набір зазвичай однаковий.

Платний хостинг обов'язково буде оснащений веб-панеллю ("адмінкою") - програмним комплексом, що дозволяє користувачам, які не мають спеціальних знань, займатися налаштуваннями акаунта через звичайний браузер. Використовуючи веб-панель, користувач може дізнаватися, наприклад, номер свого договору, змінювати свої контактні дані, виписувати квитанцію або рахунок на оплату послуг, переглядати статистику відвідувань сайту, контролювати стан свого грошового рахунку, налаштовувати поштові скриньки, доменні імена та робити ще багато чого.

іншого. Наявність файлового менеджера ще одна особливість платного хостингу. За допомогою цієї програми користувач може створювати файли та папки у директорії сайту, не вдаючись до послуг FTP-клієнтів. FTP доступ дозволяє завантажувати файли і навіть цілі каталоги з сервера на ваш комп'ютер і навпаки. Безкоштовні варіанти хостингу позбавлені такої можливості – у них все слід виконувати виключно через браузер, що не завжди зручне. Не буде зайвою і можливість анонімного FTP-доступу – цей варіант дає можливість швидко завантажувати певні адміністратором сайту файли без пароля. Є можливість паролювання і самої директорії сайту.

Щоденний Back-up (резервне копіювання всієї інформації) на незалежні HDD, встановлені на кожному сервері, - ще одна прерогатива платного хостингу. Втім, користувач може вибрати самостійний Back-up, який керується через контрольну веб-панель. З його допомогою користувачі самі можуть робити резервні копії свого сайту у зручний для них час.

Декілька доменів і піддоменів для одного сайту можна розмістити тільки на платних хостингах. Веб-майстрам, які працюють зі складними проектами, припадуть до смаку підтримка SSI для складання веб-сторінки з блоків, що зберігаються в інших файлах, наявність захищеного сервера Secure Socket Layer (SSL), який дозволяє перевіряти автентичність користувачів, гарантувати збереження вмісту та шифрувати дані, що передаються через мережу. Останнє дуже важливо, якщо потрібне суворе дотримання конфіденційності даних, що передаються.

Встановлювати свої власні скрипти на сайти на безкоштовному хостингу не вдасться, а ось платні варіанти дають можливість створювати особисті CGI-BIN директорії. Комусь корисною виявиться і технологія Parser, що дозволяє зручно створювати сайти, використовуючи автоматичний збір інформації з інших ресурсів. Варто також згадати і функцію підтримки Frontpage Server Extensions, що дозволяє коректно відображати сайти, які були зроблені за застарілими на сьогоднішній день технологіями.

Окремо слід поговорити про статистику. При розміщенні сайту на платному хостингу його адміністратору в будь-який момент буде доступна докладна статистика відвідування ресурсу. Це Log-файли, за допомогою яких можна отримувати якісну та докладну статистику заходів на сайт, обробляти та аналізувати помилкові ситуації, що виникали під час роботи ресурсу. Це ще й список посилань Referrers Log, які показують, звідки відвідувачі приходили на Ваш сайт і протокол помилок Error Log.

Якісна робота з електронною поштою також доступна лише за умови розміщення сайту на платному хостинговому майданчику. У такому разі ви зможете завести собі деяку (а іноді і необмежену) кількість поштових скриньок POP3/IMAP. Це будуть абсолютно окремі поштові скриньки, і кожен з них матиме свою адресу, свій пароль і певний розмір, і налаштовувати ці дані може сам користувач. Таку пошту можна перевіряти будь-якою поштовою програмою – The Bat, Outlook Express або іншими, а можна читати кореспонденцію через адмін-панель. Переадресація електронної пошти – ще одна корисна функція. Будь-яка поштова скринька типу ivanov@mysite.com може автоматично переслати лист на іншу вашу поштову адресу. А ще можна налаштувати SMS-повідомлення про надходження нової пошти до скриньки на сайті або влаштувати масове поштове розсилання новин, оглядів, іншої інформації для постійних відвідувачів та клієнтів.

При розміщенні сайту на вибраному хостингу бажано періодично робити резервні копії або, як їх ще називають, бекапи. Ці бекапи дозволяють швидко повернути працездатність сайту та втрачену інформацію, якщо вона з тієї чи іншої причини була видалена, пошкоджена вірусами або просто зникла. Досить часто хостингові компанії, що надають платний хостинг, самі роблять резервні копії та зберігають їх на своїх серверах, що дуже зручно для веб-майстра і бажано уточнити це у представника компанії при замовленні хостингу. Для розміщення сайту на сервері часто використовуються FTP-клієнти. Для цього необхідно встановити на хостингу FTP-сервер і

налаштувати його роботу, що можна зробити досить швидко і легко. А далі необхідно встановити FTP-клієнт, і можна закачувати на сервер потрібну інформацію, видаляти стару, робити бекапи і у разі потреби завантажувати файли на свій комп'ютер, щоб зберегти їх або прогнати антивірусною програмою.

3 ПРАКТИЧНЕ ВИКОРИСТАННЯ WEB-ЗАСТОСУНКУ

Для демонстрації практичного використання WEB-застосунку покажемо його роботу в додаванні нових елементів до проекту та їх редагування.

3.1 Додавання та редагування елементів до проекту

При заході на користувач бачить перед собою головну сторінку сайту, з якої він відразу може почати конструювання свого проекту. Для додавання нових елементів необхідно вибрати та перенести елементи з списку, який знаходиться в лівій частині сайту до полотна, яке знаходиться у центрі екрану. Якщо користувач не зрозуміє що необхідно робити, по центру екрану знаходиться підказка.

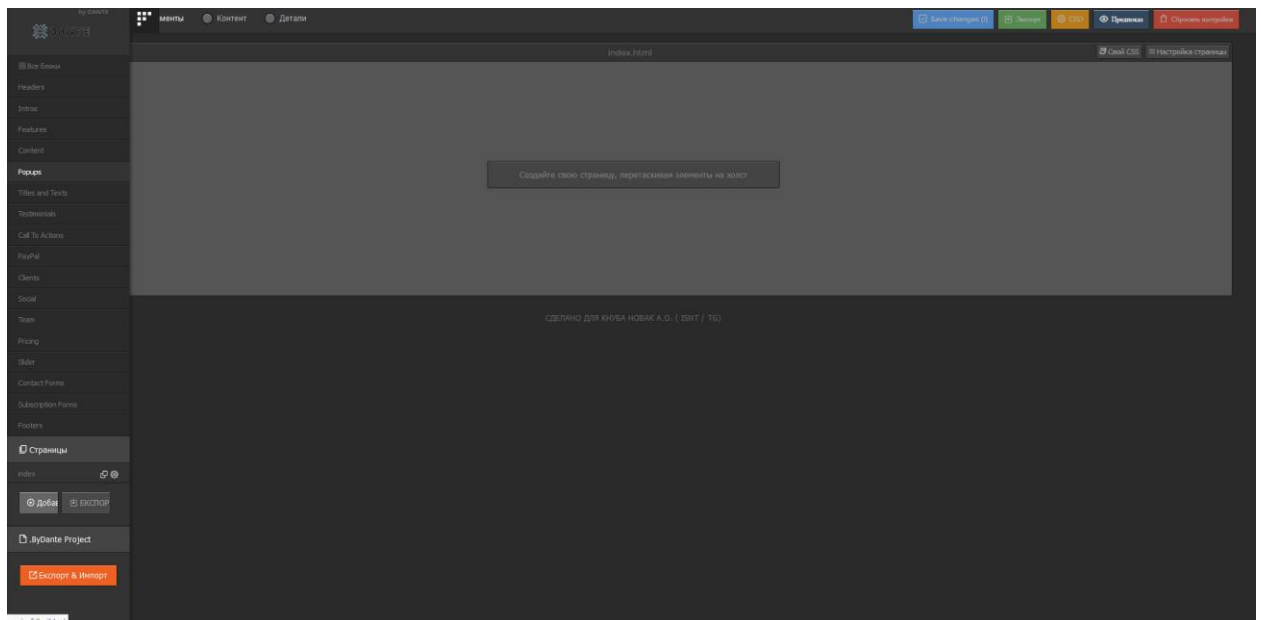


Рисунок 3.1 головна сторінка сайту з відкритим списком елементів

Це достатньо просто зробити, тому що вибране ядро сайту підтримує функцію Drag-and-drop.

Користувач може вибрати необхідні елементи та сконструювати необхідний собі сайт. Список має заздалегідь готові частини популярних інтерфейсів.

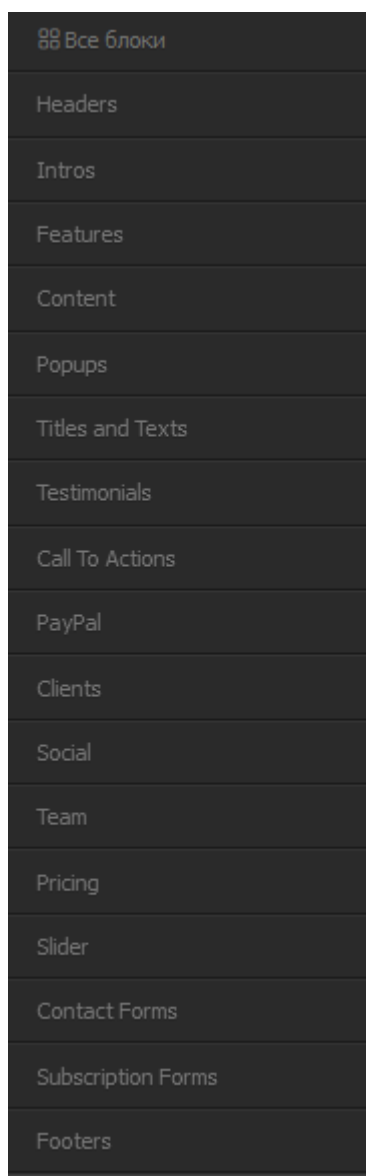


Рисунок 3.2 список елементів для проекту сайту

Елементи сайту найрізноманітніші, є заготовлені header та footer сайту, частини для компонування контенту, контактних форм, коментаріїв інших клієнтів.

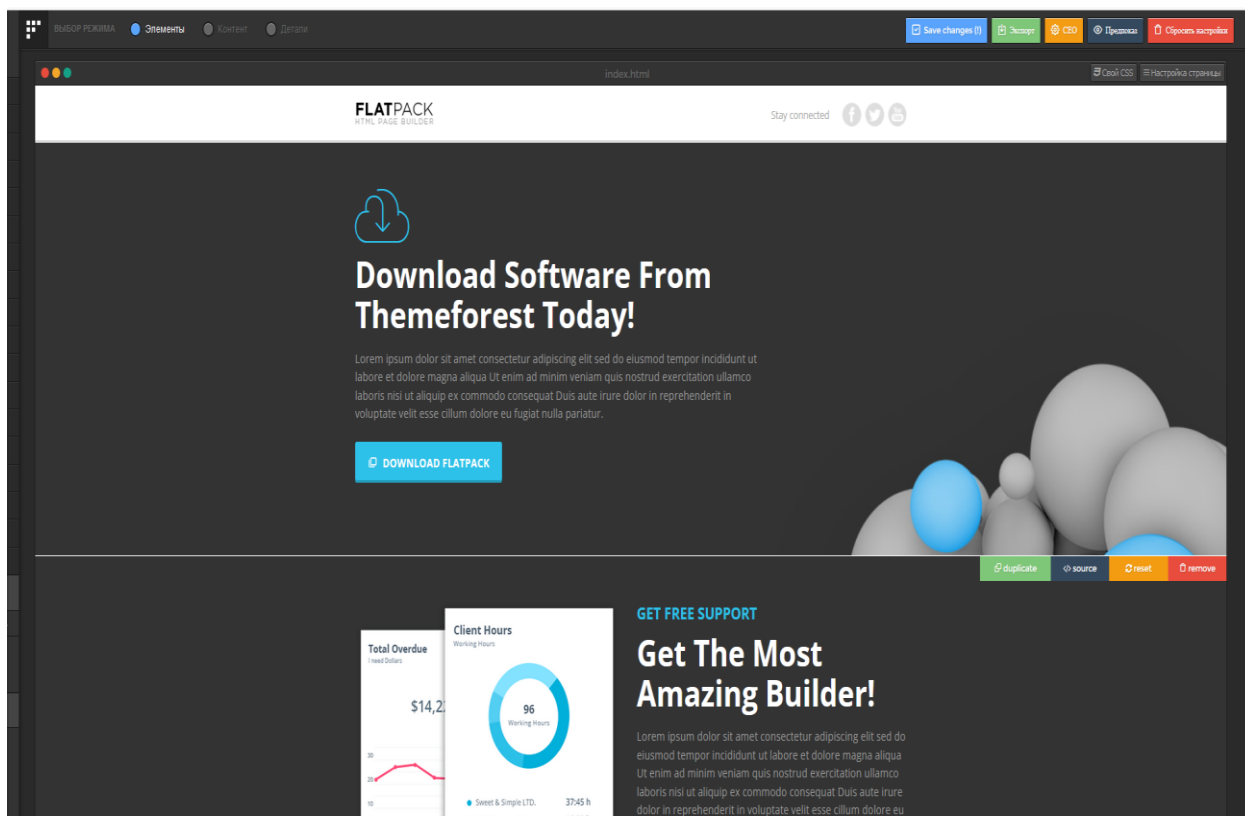


Рисунок 3.3 макет сайту

Після того, як користувач перетягнув необхідні йому елементи майбутнього сайту, йому необхідно буде відредагувати данні шаблонів під свої. Для цього необхідно вибрати режим редагування у лівому верхньому куту. Наразі присутні три режими : елементи, контент, деталі.

Режим елементи відповідає за редагування окремих елементів сайту які користувач додавав з шаблону. В цьому режимі користувач може продублювати елемент, замінити готовий CSS код цього елементу, скинути поточні налаштування або видалити його.

Режим контент відповідає за редагування контенту елементів сайту які користувач додавав з шаблону. В цьому режимі користувач може змінити увесь наявний текст в елементах або видалити його.

Режим деталі відповідає за редагування деталей окремих елементів сайту які користувач додавав з шаблону. В цьому режимі користувач може редагувати колір тексту, його розмір, задній фон та шрифт. За необхідності користувач може додати анімацію для окремого елемента.

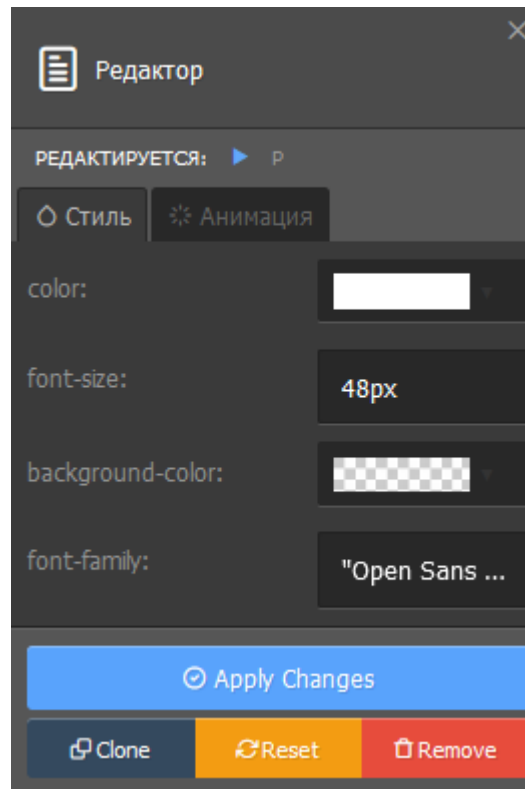


Рисунок 3.4 редагування в режимі «деталі»

3.2 Редагування SEO налаштувань

За необхідності редагування SEO налаштувань користувач може натиснути спеціальну кнопку у правому верхньому куті “SEO”. Але це можливо лише за умови, якщо на проекті уже доданий хоч один елемент. Після натискання на кнопку, відкриється модальне вікно.

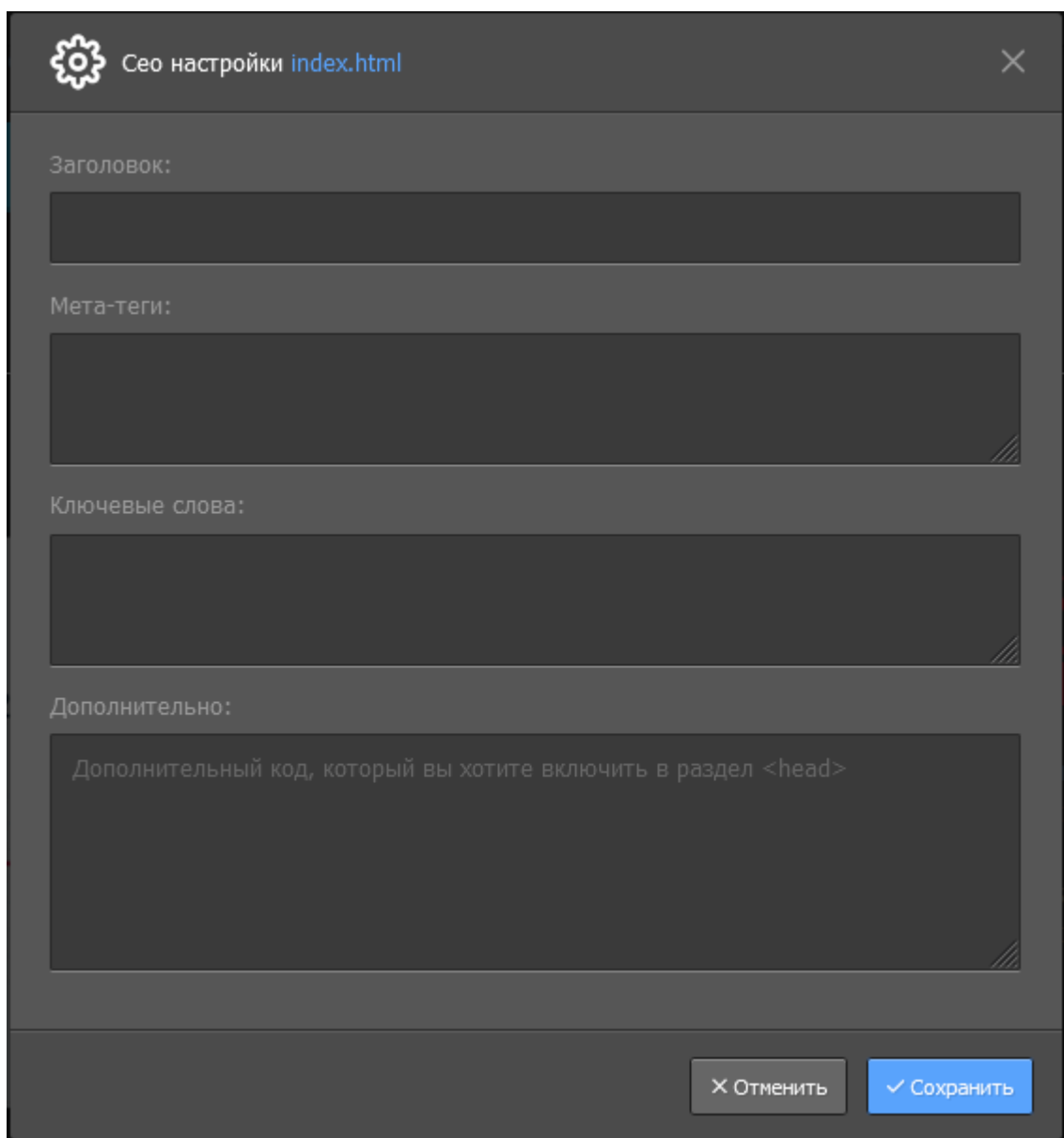


Рисунок 3.5 модальне вікно «SEO»

У цьому вікні користувач може додати необхідні SEO налаштування до тегу <head> свого проекту.

3.3 Экспорт проекта

Якщо користувачу необхідний бекап його проекту, то він може скористатись функцією експорту. У користувача є два вибори експорту :

Перший варіант експорту, це варіант експорту самого проекту з сайту. Для подальшого його редагування на іншому комп'ютері. Оскільки у web-застосунку немає бази даних, то уся інформація записується у сесії, тому для використання на інших комп'ютерах необхідний файл проекту. Для його

отримання необхідно звернутись до кнопки експорту, яка знаходиться у лівому нижньому куті під назвою «експорт & імпорт». Після її натискання відкриється модальне вікно, у якому можна буде зробити бекап свого проекту.

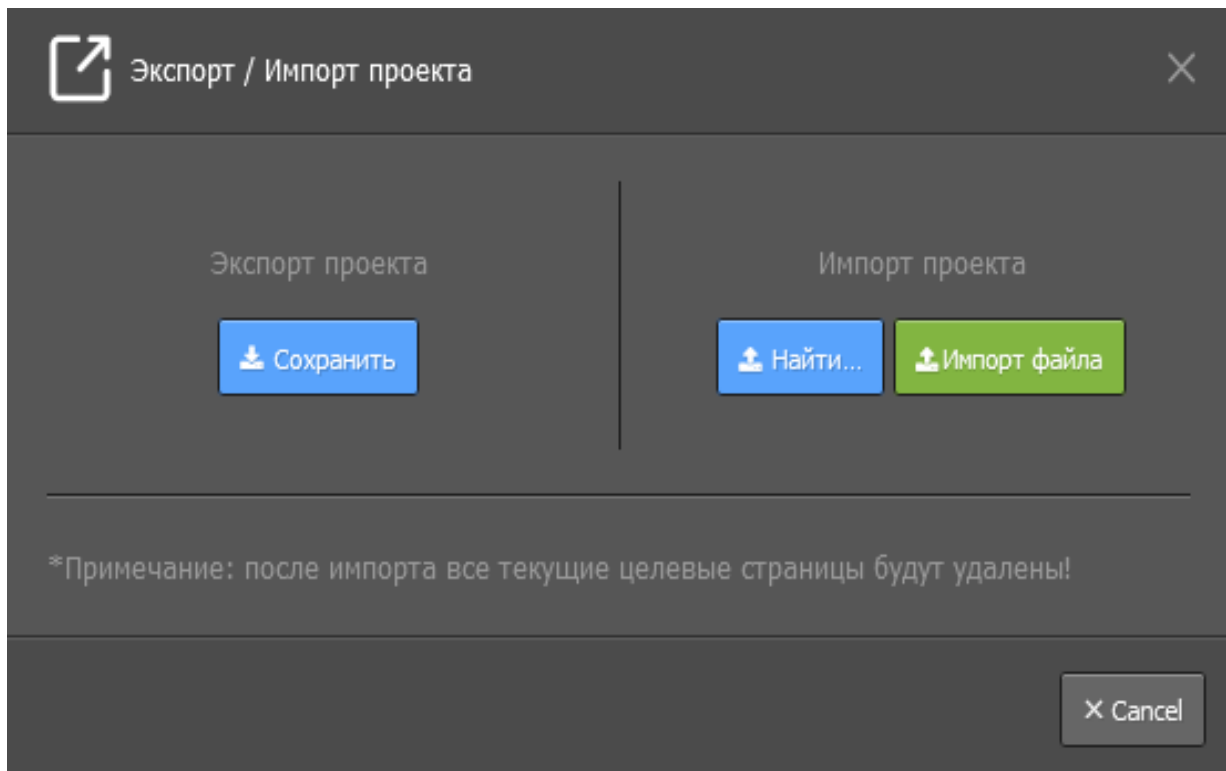


Рисунок 3.6 модальне вікно «експорт & імпорт»

Другий варіант експорт, це варіант експорту проекту для FTP передачі його на свій домен. Це рекомендується робити, коли проект уже повністю готовий. Для такого експорту необхідно звернутись до кнопки яка знаходиться у правому верхньому куті. Після її натискання відкриється модальне вікно, через яке можна експортувати макет у готовий до FTP передачі проект

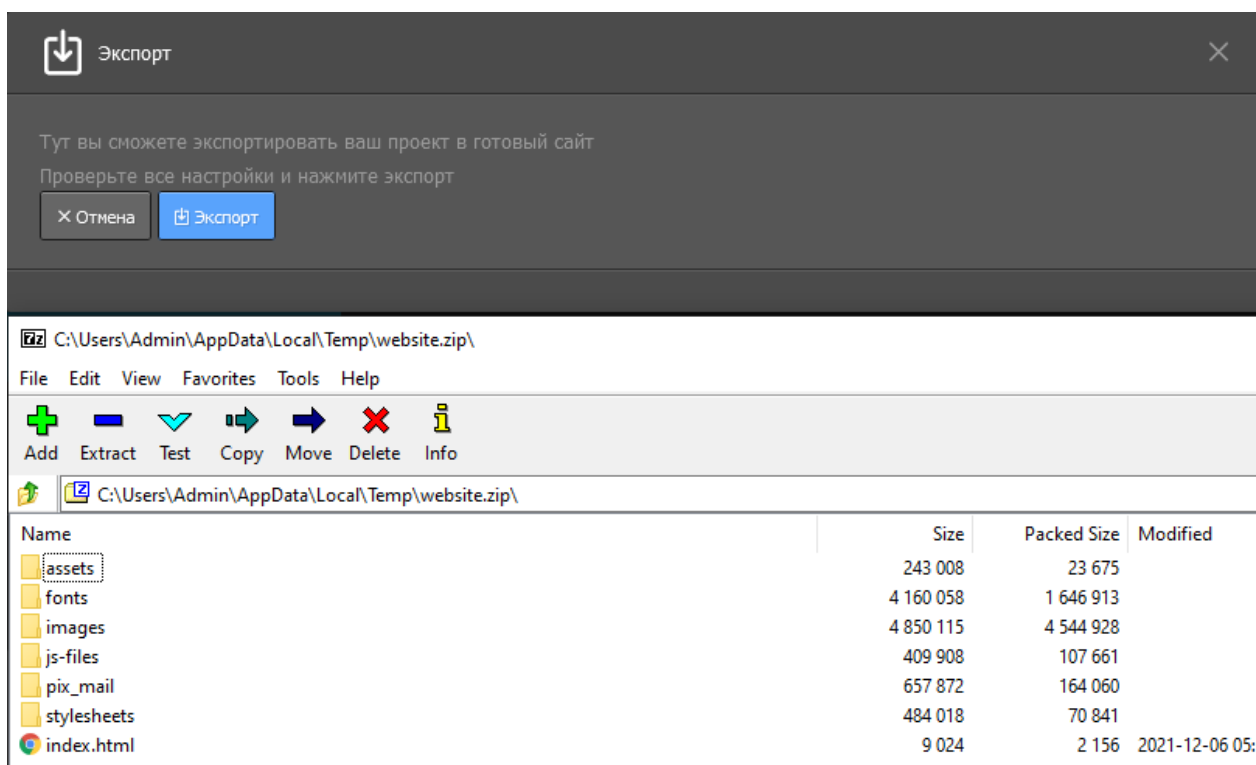


Рисунок 3.7 модальне вікно «експорт»

3.4 Передпоказ проекту

Хоча не потрібно забувати про передпоказ проекту, перед його експортом. Щоб подивитись як буде виглядати майбутній сайт, необхідно натиснути на кнопку у правому верхньому куті. Після чого у новій вкладці браузера відкриється окремий сайт з уже зкомпільованим проектом, без редактору.

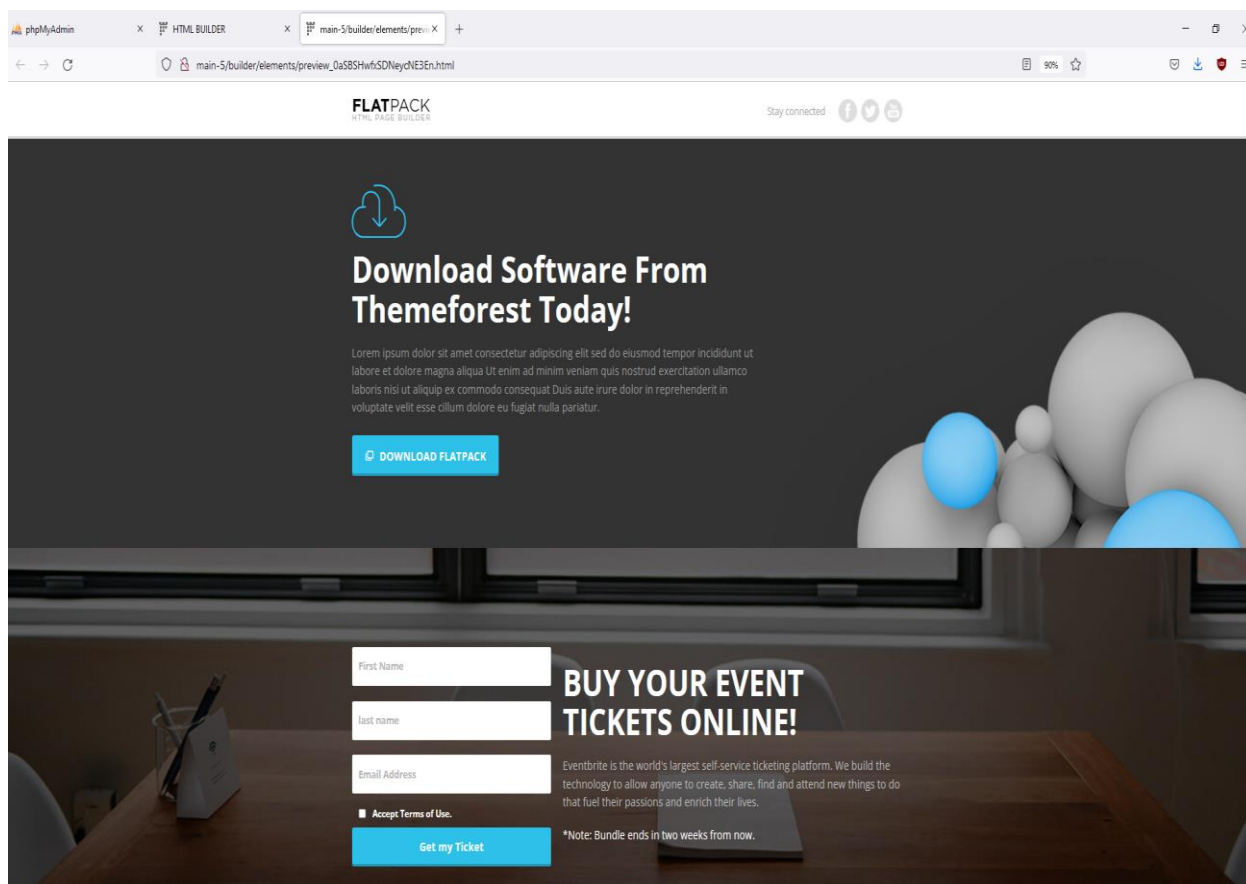


Рисунок 3.8 нова вкладка з проектом

3.5 Редагування css елементу

Для редагування css вибраного елементу, користувачу необхідно у режимі «елементи» вибрати у правому верхньому куті елемента на кнопку «source». Після чого відкриється модальне вікно з редагуванням css.

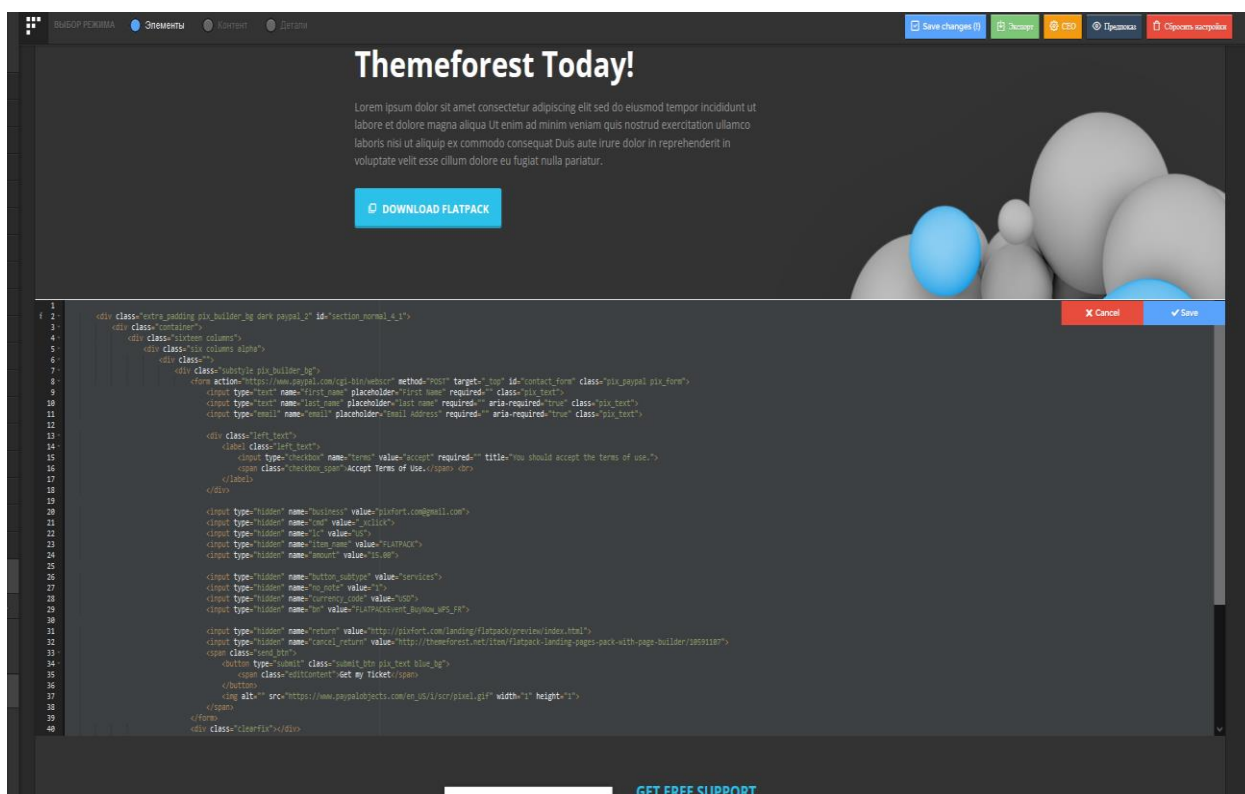


Рисунок 3.9 редагування css елемента

Після редагування у правому верхньому куті користувачу необхідно зберегти або ж відмінити усі зміни.

3.6 Редагування контенту

Для редагування контенту всередині елемента, користувачу необхідно вибрати режим «контент» та натиснути на необхідний елемент. Після цього відкриється модальне вікно з редагуванням тексту. В ньому користувач зможе змінити текст, зробити його жирним/курсивом/закресленим.

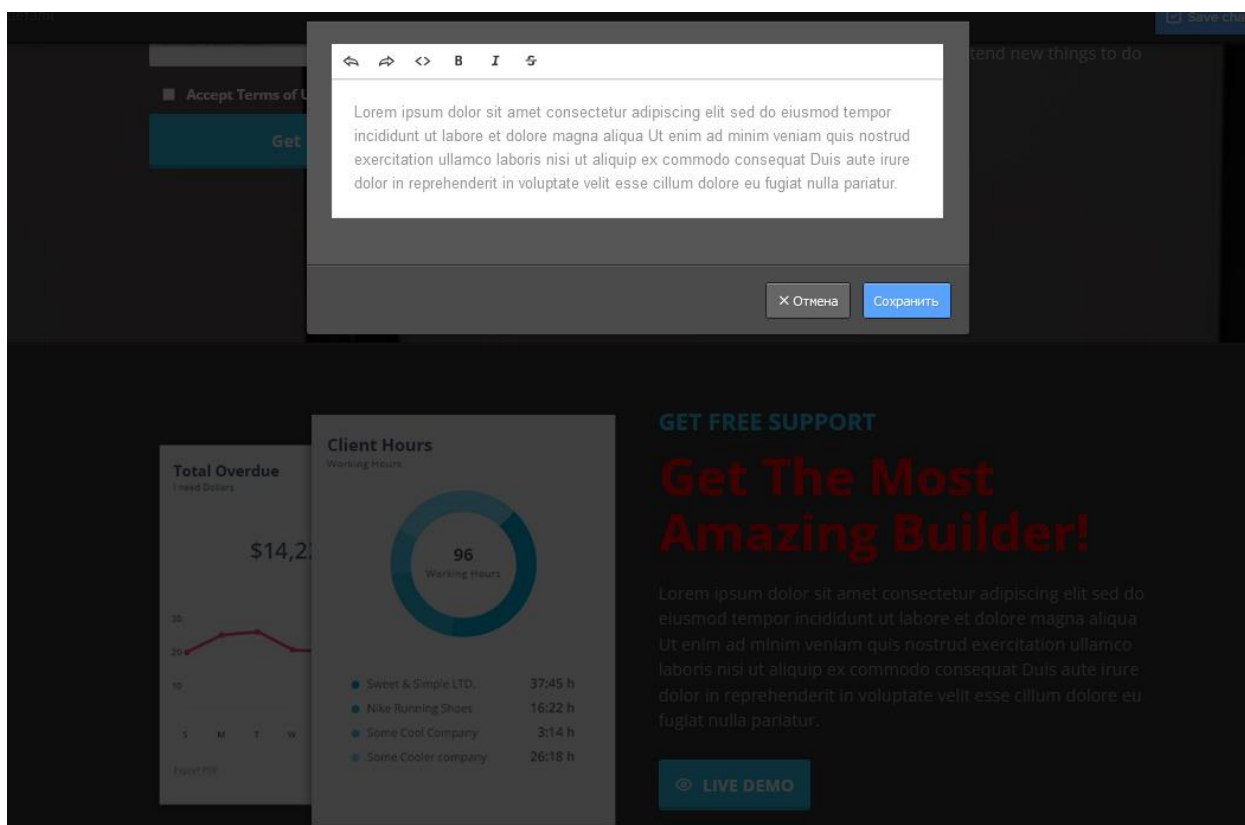


Рисунок 3.10 редагування css елемента

4 ЕРГОНОМІКА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

4.1 Розрахунок часу евакуації людей при пожежі в приміщенні

Підприємство є одноповерховою будівлею, що відображена на рис. 4.1 розмірами 10 м. на 20м.; кількість робочих кімнат 8; кількість працюючих 13; кількість виходів 1.

Для розрахунку загального часу евакуації необхідно розрахувати час на кожній ділянці руху людей, починаючи від максимально віддаленої точки.

Рух людей під час процесу евакуації є вимушеним, тобто пов'язаним із необхідністю покинути приміщення чи будівлю через виниклу небезпеку. Вимушений рух людей має свої специфічні особливості, вже на початковій стадії, людині погрожує небезпека в результаті того, що пожежа супроводжується виділенням теплоти, продуктів повного й неповного згорання, токсичних речовин, обвалення конструкцій, що так чи інакше погрожує людині. Із цього слід зробити висновок, що при плануванні будівлі і устрої приміщень в них необхідно прийняти заходи, щоб процес евакуації міг закінчитися безпечно і в необхідний час.

Друга особливість полягає у тому, що в силу погрожуючої людині небезпеки рух інстинктивно починається одночасно в один і той же напрям – у сторону виходів. Це призводить до того, що проходи швидко заповнюються людьми при визначеній щільності потоків. Із збільшенням щільності потоків швидкість руху зменшується, що створює певний визначений ритм руху. В цій ситуації з'являється погроза утворення затору, і дуже важко запобігти їй.

Показником ефективності процесу вимушеної евакуації є час, на протязі якого люди можуть при необхідності покинути окремі приміщення і будівлю в цілому. Безпечність, досягнута тоді, коли цей час менший, ніж тривалість пожежі. Короткочасність процесу евакуації повинна досягатися не тільки конструктивно-планувальними рішеннями, на які звертали увагу раніше, але й організаційними рішеннями.

Процес евакуації людей можна поділити на три етапи:

- рух людей від найбільш віддаленої точки приміщення до евакуаційних виходів;
- рух людей від евакуаційних виходів до виходів на зовні;
- рух людей від виходів із будівлі та їх розсіювання.

При евакуації основними параметрами, які характеризують процес руху людей є:

- щільність людського потоку – D , люд/м²;
- швидкість руху людського потоку – v , м/хв;
- пропускна спроможність шляху (виходів) – Q , чол/хв;
- інтенсивність руху людського потоку – q м/хв;

1) Щільність людського потоку D , яка складається з N людей, дорівнює:

$$D_1 = \frac{N_1 f}{A} \text{ м}^2/\text{м}^2, \quad (4.1)$$

де $A = g \cdot l$ – площа шляху евакуаційної ділянки [м²];

l – довжина ділянки, м; g – ширина ділянки, м;

f – площа горизонтальної проекції людини м².

Якщо $D < 0.05$ людина має повну свободу пересування.

Якщо $0.05 < D < 0.15$ людина не може вільно змінювати напрямок свого руху.

Якщо $0.15 < D \leq 0.92$ люди рухаються вкупі. Величина 0.92 є верхньою межею, коли люди рухаються вкупі, та нею обмежується щільність при проектуванні евакуаційних шляхів.

2) Швидкість руху людського потоку v залежить від його щільності D та виду шляху (горизонтальні чи похилі). Значення швидкості V , а також інтенсивності руху людського потоку q в залежності від його щільності D приведено в табл. 4.1.

Таблиця 4.1

Значення швидкості v і інтенсивності q руху людського потоку залежно від його щільності D

Щільність потоку m^2/m^2 , D	Горизонтальний шлях		Дверний проем	Сходи вниз		Сходи вверх	
	Швидкість м/хв. v	Інтенсивність, q м/хв.	Інтенсивність, q м/хв.	Швидкість м/хв. v	Інтенсивність, q м/хв.	Швидкість м/хв. v	Інтенсивність, q м/хв.
0,01	100	1	1	100	1	60	0,6
0,05	100	5	5	100	5	60	3
0,1	80	8	8,7	95	9,5	53	5,3
0,2	60	12	13,4	68	13,6	40	8
0,4	40	16	18,4	40	16	26	10,4
0,6	27	16,2	19	24	14,4	18	10,8
0,8	19	15,2	17,3	13	10,4	13	10,4
0,9 и більше	15	13,5	8,5	8	7,2	11	9,9

3) Пропускна спроможність шляху Q (м/хв чи люд/хв) визначається:

$$Q = D \cdot v \cdot \delta \text{ м}^2/\text{хв.} \quad (4.2)$$

4) Інтенсивністю руху людського потоку q (м/хв чи люд/хв) визначається:

$$q = D \cdot v \quad (4.3)$$

Інтенсивність руху не залежить від ширини шляху і являється характеристикою потоку. Інтенсивністю руху людського потоку на кожному відрізку дорівнює:

$$q_i = \frac{q_{i-1} \delta_{i-1}}{\delta_i} \text{ м/хв.} \quad (4.4)$$

де: δ_i , δ_{i-1} – ширина розглядаючого i -го і перед ним ($i - 1$) відрізків шляху, м;

q_i, q_{i-1} – значення інтенсивності руху потоку на розглядаючому i -му і перед ним ($i - 1$) відрізках шляху, м/хв.

Якщо q_i менше чи рівно q_{\max} , то час руху на відрізку можна визначити по формулі:

$$t_1 = \frac{l_1}{v_1} \text{ хв} \quad (4.5)$$

при цьому значення q_{\max} треба приймати рівним, м/хв:

-для горизонтальних шляхів	16,5
-для дверних проємів	19,6
-для сходів вниз	16
-для сходів вверх	11

Розрахунковий час евакуації людей із приміщення й будівлі t_p встановлюється по розрахунку часу руху людських потоків від найбільш віддалених місць розташування. При розрахунку весь шлях руху людського потоку поділяється на ділянки (прохід, коридор, сходишковий марш, дверний проріз, тамбур) довжиною l_i і шириною g_i .

Початковими ділянками являються проходи між робочими місцями.

Розрахунковий час евакуації дорівнює:

$$t_p = t_1 + t_2 + t_3 + \dots + t_i = t \text{ хв},$$

$$t_i = \frac{l_i}{v_i} \text{ хв},$$

де t_i – час руху людського потоку на кожній окремій ділянці.

Умова безпечної евакуації характеризується виразом $t_p \leq t_{нб}$, тобто розрахункова тривалість вимушеної евакуації на різноманітних ділянках при розрахункових швидкостях людей і розрахунковій пропускній спроможності евакуаційних дверей повинна бути рівна або менша необхідного часу тривалості евакуації. Необхідний час евакуації $t_{нб}$ визначається по таблиці.

Використовуючи вище зазначений опис, за винятком таких ділянок як дверний проріз та тамбур (не передбачена у будівлі), проведемо розрахунок часу евакуації людей для прийнятого приміщення.

Маршрут евакуації розбивається на дев'ять етапів (ділянок). Для проведення розрахунку задаємося планом евакуації людей (рис. 4.1).

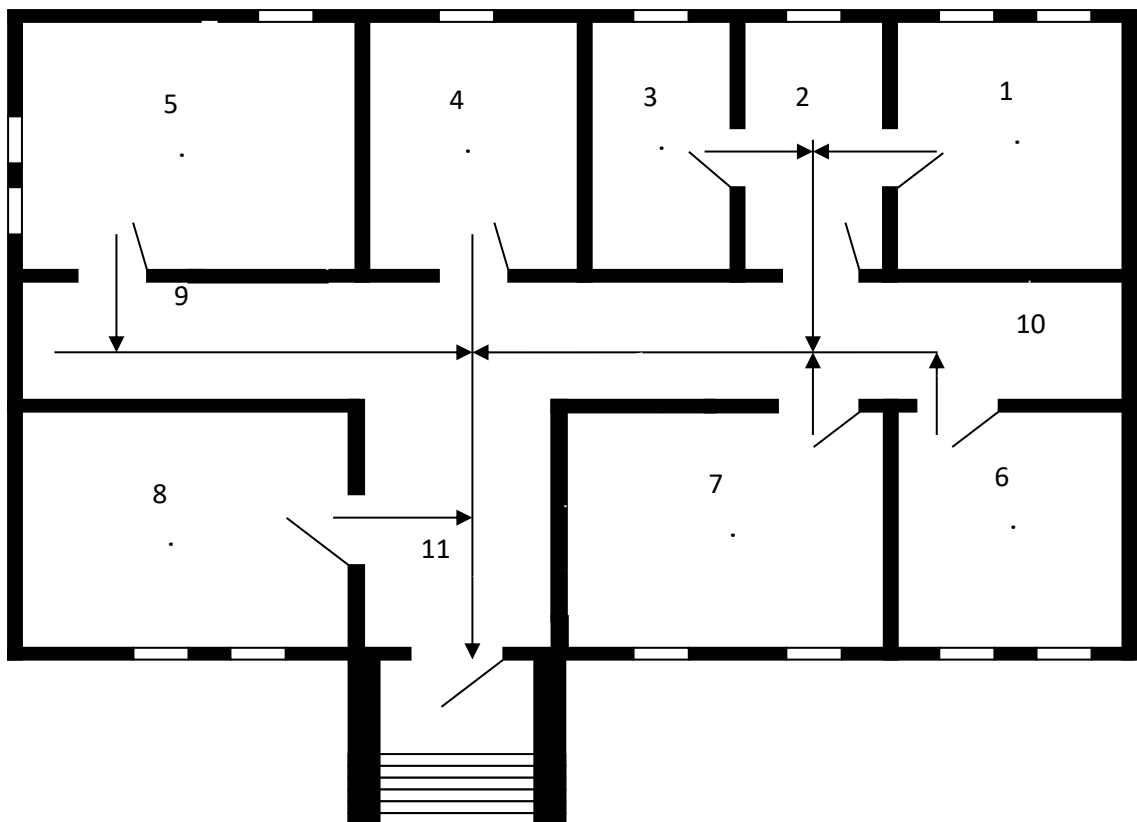


Рисунок 4.1 План евакуації людей

Перша ділянка.

Час руху людського потоку – вихід людей з кімнати № 1:

де $l = 13$ м – довжина ділянки; v – швидкість руху на ділянці.

$f = 0.113$ м² – середня площа горизонтальної проекції людини ;

$N = 2$ – кількість людей ; $S = 3$ м – ширина ділянки .

$$D_1 = 2 \left(\frac{0.113}{3 \cdot 13} \right) = 0.006 \text{ м}^2/\text{м}^2, \text{ тоді } v_1 = 100 \text{ м/хв} ; q_1 = 1 \text{ м/хв.}$$

$$t_1 = 13/100 = 0,13 \text{ хв.}$$

Друга ділянка.

Час руху людського потоку – вихід людей з кімнати № 2:

$$D = 3 \left(\frac{0.113}{11 \cdot 3} \right) = 0.01 \text{ м}^2/\text{м}^2, \text{ тоді } v_3 = 100 \text{ м/хв} ; q_3 = 1 \text{ м/хв.}$$

$$t_2 = 11/100 = 0,11 \text{ хв.}$$

де $l = 11$ м; $f = 0.113$ м²; $N = 3$; $S = 3$ м.

Третя ділянка.

Час руху людського потоку – вихід людей з кімнати № 3:

$$D = 1 \left(\frac{0.113}{12 \cdot 3} \right) = 0.003 \text{ м}^2/\text{м}^2, \text{ тоді } v_2 = 100 \text{ м/хв} ; q_2 = 1 \text{ м/хв.}$$

$$t = 12/100 = 0,12 \text{ хв.}$$

де $l = 12$ м; $f = 0.113$ м²; $N = 1$; $S = 3$ м.

Четверта ділянка.

Час руху людського потоку – вихід людей з кімнати № 4:

$$D = 2 \left(\frac{0.113}{5 \cdot 3} \right) = 0.01 \text{ м}^2/\text{м}^2, \text{ тоді } v_4 = 100 \text{ м/хв} ; q_4 = 1 \text{ м/хв.}$$

$$t = 5/100 = 0,05 \text{ хв.}$$

де $l = 5$ м; $f = 0.113$ м²; $N = 2$; $S = 3$ м.

П'ята ділянка.

Час руху людського потоку – вихід людей з кімнати № 5:

$$D = 2 \left(\frac{0.113}{12 \cdot 3} \right) = 0.007 \text{ м}^2/\text{м}^2, \text{ тоді } v_5 = 100 \text{ м/хв} ; q_5 = 1 \text{ м/хв.}$$

$$t = 12/100 = 0,12 \text{ хв.}$$

де $l = 12$ м; $f = 0.113$ м²; $N = 2$; $S = 3$ м.

Шоста ділянка.

Час руху людського потоку – вихід людей з кімнати № 6:

$$D = 2 \left(\frac{0.113}{12 \cdot 3} \right) = 0.007 \text{ м}^2/\text{м}^2, \text{ тоді } v_6 = 100 \text{ м/хв}; q_6 = 1 \text{ м/хв.}$$

$$t = 12/100 = 0,12 \text{ хв.}$$

де $l = 12$ м; $f = 0.113$ м²; $N = 2$; $S = 3$ м.

Сьома ділянка.

Час руху людського потоку – вихід людей з кімнати № 7:

$$D = 2 \left(\frac{0.113}{9 \cdot 3} \right) = 0.008 \text{ м}^2/\text{м}^2, \text{ тоді } v_7 = 100 \text{ м/хв}; q_7 = 1 \text{ м/хв.}$$

$$T = 9/100 = 0,09 \text{ хв.}$$

Де $l = 9$ м; $f = 0.113$ м²; $N = 2$; $S = 3$ м.

Восьма ділянка.

Час руху людського потоку – вихід людей з кімнати № 8:

$$D = 2 \left(\frac{0.113}{3 \cdot 3} \right) = 0.02 \text{ м}^2/\text{м}^2, \text{ тоді } v_8 = 100 \text{ м/хв}; q_8 = 1 \text{ м/хв.}$$

$$t = 3/100 = 0,03 \text{ хв.}$$

де $l = 3$ м; $f = 0.113$ м²; $N = 2$; $S = 3$ м.

Дев'ята ділянка.

Час руху людського потоку – вихід людей з кімнати № 9:

$$D = 7 \left(\frac{0.113}{9 \cdot 3} \right) = 0.03 \text{ м}^2/\text{м}^2, \text{ тоді } v_9 = 100 \text{ м/хв}; q_9 = 1 \text{ м/хв.}$$

$$t = 9/100 = 0,09 \text{ хв.}$$

де $l = 9$ м; $f = 0.113$ м²; $N = 7$; $S = 3$ м.

Десята ділянка.

Час руху людського потоку – вихід людей з кімнати № 10:

$$D = 11 \left(\frac{0.113}{5 \cdot 3} \right) = 0.08 \text{ м}^2/\text{м}^2, \text{ тоді } v_{10} = 100 \text{ м/хв}; q_{10} = 1 \text{ м/хв.}$$

$$T = 5/100 = 0,05 \text{ хв.}$$

Де $l = 5$ м; $f = 0.113$ м²; $N = 11$; $S = 3$ м.

Одинадцята ділянка.

Час руху людського потоку – вихід людей з кімнати № 11:

$$D = 13 \left(\frac{0.113}{3 \cdot 3} \right) = 0.1632 \text{ м}^2/\text{м}^2, \text{ тоді } v_{11} = 60 \text{ м/хв}; q_{11} = 12 \text{ м/хв.}$$

$$t = 3/60 = 0,05 \text{ хв.}$$

$$\text{де } l = 3 \text{ м}; f = 0.113 \text{ м}^2; N = 13; S = 3 \text{ м.}$$

Загальний час евакуації: $t = t_1 + t_2 + \dots + t_{18} = 1,01 \text{ хв.}$

$t_{нб} = 2,5$ хвилин для одноповерхового будинку (з СНиП 2.01.02-85, табл. 12) $t = 1,01 < t_{нб} = 2,5$ хв, тобто вимоги пожежної безпеки виконуються.

В зв'язку з можливістю виникнення пожежі на території будівлі внаслідок несправної роботи комп'ютерної техніки, яка підключена до електромережі, я вирішив вибрати вуглекислотні вогнегасники моделі ОУ-8 та порошкові – моделі ОП-8Б. Розмістити їх необхідно на пожежних щитах в вестибюлі та біля пожежного, по одному екземпляру кожного типу.

За допомогою вогнегасника ОУ-8 можна гасити різні речовини, крім тих, які можуть горіти без доступу повітря. Також їм можна тушити пожежу в пристроях під напругою до 1000V, при умові приближення по струмопровідних частин не ближче одного метру.

Механізм припинення горіння за допомогою використання вуглекислого газу базується на його властивостях шляхом розбавлення знижувати концентрацію реагуючих речовин до рівня, при якому горіння становиться неможливим.

За допомогою вогнегасника ОП-8Б можна тушити палаюче електрообладнання під напругою до 1000V, легкозаймисті рідини, тліючі матеріали (навіть ті що горять без доступу повітря) праці в робочому приміщенні.

4.2 Ергономічні вимоги до організації і обладнання робочих місць з комп'ютерною технікою

Оператор обробки інформації при виконанні своєї роботи майже весь робочий час знаходиться в сидячому положенні за робочим столом, на якому розташоване його робоче обладнання. Для запобігання виникнення, пов'язаних з таким видом робіт, хвороб (сколіоз, хвороби очей та ін.), а також для усунення загального дискомфорту, зменшення втомлюваності працівника, підвищенню його продуктивності необхідно правильно організувати робоче місце.

Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічного обґрунтованого робочого положення, виробничих меблів з урахуванням антропометричних характеристик людини;
- раціональну компоновку обладнання на робочих місцях;
- урахування характеру та особливостей трудової діяльності.

ДНАОП 0.00-1.31-99, ГОСТ 12.2.032-78, ДСанПІН 3.3.2.007-98 регламентує такі вимоги до організації робочого місця користувача ВДТ (візуальний дисплейний термінал):

1) Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Рекомендовані розміри столу: висота – 725 мм, ширина – 600-1400 мм, глибина – 80-1000 мм. Робочий стіл повинен мати простір для ніг висотою не менше ніж 450 мм, на рівні витягнутої ноги не менше 650 мм.

Робоче місце має бути обладнане підставкою для ніг шириною не менше ніж 300 мм, глибиною не менше ніж 400 мм, з можливістю регулювання по висоті в межах 150 мм та кута нахилу опорної поверхні – в

межах 20° . Підставка повинна мати рифлену поверхню і бортик по передньому краю заввишки 10 мм.

2) Робочий стілець користувача ВДТ повинен мати такі основні елементи: сидіння, спинку та стаціонарні або знімні підлокітники. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, за кутом нахилу сидіння та спинки і за відстанню від спинки до попереднього краю сидіння. Поверхня сидіння має бути плоскою, передній край заокругленим.

Висота поверхні сидіння має регулюватися в межах 400-500 мм, а ширина і глибина становити не менше ніж 400 мм. Кут нахилу сидіння – до 15° вперед і до 5° назад.

Висота спинки має становити 300 ± 20 мм, ширина – не менше ніж 380 мм, радіус кривизни горизонтальної площини – 400 мм. Кут нахилу спинки має регулюватися в межах $0 - 30^\circ$ від вертикального положення. Відстань від спинки до переднього краю сидіння має регулюватися в межах 260 – 400 мм.

Для зниження статичного навантаження м'язів верхніх кінцівок слід використовувати стаціонарні або знімні підлокітники довжиною не менше ніж 250 мм, шириною не менше ніж 50 – 70 мм, що регулюються за висотою над сидінням у межах 230 – 260 мм і відстанню між підлокітниками в межах 350 – 500 мм.

Поверхня сидіння і спинки стільця має бути напівм'якою з нековзним, повітронепроникненим покриттям, що легко чиститься і не електризується.

Конструкція виробничих меблів для користувача ВДТ має бути такою, щоб забезпечувати йому підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; верхні частини рук – вертикальні; кут ліктьового суглоба (між плечем та передпліччям) – $70 - 90^\circ$; зап'ястки зігнуті під кутом не більше 20° відносно горизонтальної площини, нахил голови вперед в межах $15 - 20^\circ$ до вертикалі.

3) Дисплей має розташуватися на столі на відстані від очей користувача не більше 700 мм (оптимальна відстань 450 – 500 мм).

Розташування екрану має забезпечувати зручність зорового спостереження у вертикальній площині під кутом $+ 30^\circ$ до нормальної лінії погляду працюючого. В горизонтальній площині кут спостереження екрану не повинен перевищувати 60° .

4) Клавіатуру слід розташувати на поверхні столу на відстані 100 – 300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій, який дає змогу змінювати кут нахилу поверхні клавіатури у межах $5 - 10^\circ$. Висота середнього рядка клавіш має не перевищувати 30 мм. Поверхня клавіатури має бути матовою з коефіцієнтом відбиття 0,4.

5) Документ для вводу даних розташовується на відстані 450 – 500 мм від очей працівника, переважно зліва, кут між екраном дисплея та документом в горизонтальній площині має бути $30 - 40^\circ$.

б) Розміщення принтера або іншого пристрою введення – виведення інформації на робочому місці має забезпечувати добру видимість екрана ВДТ, зручність ручного керування пристроєм введення – виведення інформації в зоні досяжності: по висоті 900 – 1300 мм, по глибині 400 – 500 мм. Під принтери ударної дії потрібно підкладати вібраційні килимки для гасіння вібрації та шуму.

На рис. 4.2 зображено вид робочого місця з ВДТ:

А-принтер.

В-монітор.

С-системний блок.

Д-клавіатура.

Е-папка для документів.

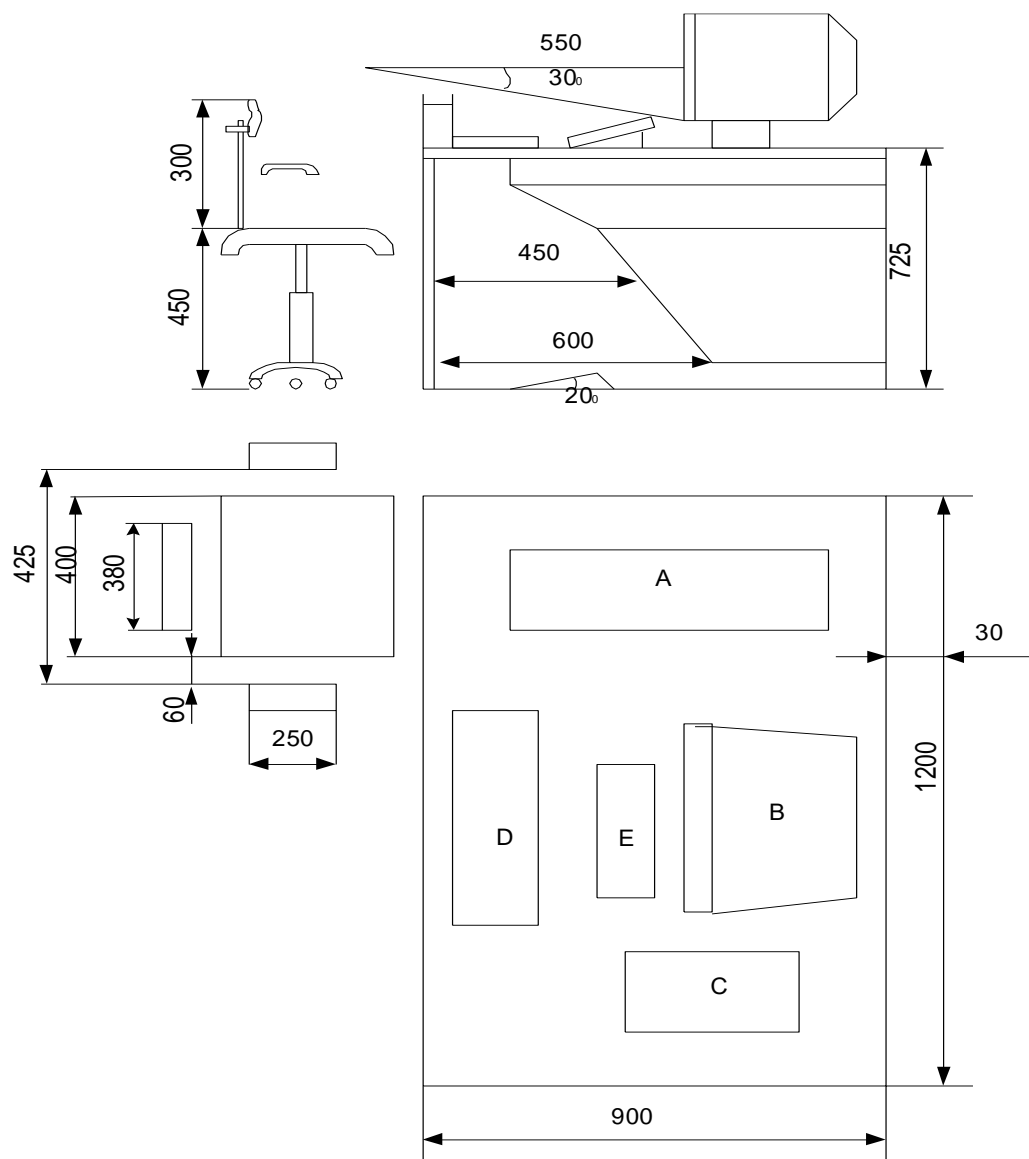


Рисунок 4.2 Вид робочого місця з ВДТ

ВИСНОВКИ

В роботі представлено аналіз поставленої задачі, вивчено специфіку предметної області. Також опрацьовано теоретичний матеріал, викладено суть та теоретичні основи досліджуваної проблеми, обрано інструментальні засоби, визначенні основні завдання проекту, проведено огляд сучасної літератури, присвяченої питанням, що розглядаються.

В результаті роботи було проведено:

- Аналіз предметної області.
- Проектування веб-ресурсу
- Розробка веб-ресурсу в сучасних умовах.
- Проведені ергономічні дослідження в області інформаційних технологій.

В роботі представлено, вивчено специфіку предметної області. Також опрацьовано теоретичний матеріал, викладено суть та теоретичні основи досліджуваної проблеми, аналіз поставленої задачі, обрано інструментальні засоби, визначенні основні завдання проекту, проведено огляд сучасної літератури, присвяченої питанням, що розглядаються.

Створений веб-ресурс відповідає всім сучасним вимогам в плані швидкодії, надійності і реалізовує поставлені цілі в повній мірі.

Використання CMS – досить простий шлях побудувати власний сайт. Вибравши одну з них, вам знадобиться:

- Придбати хостинг та доменне і'мя під ваш сайт;
- Встановити движок на хостинг;
- Придбати або створити власними зусиллями шаблон. Також можна використати один із стандартних (вони є скрізь), переробивши його стиль під свій проект;
- Визначитися із необхідним функціоналом, що не входить в базову версію, та підібрати надійні плагіни для його реалізації.
- Налаштувати базові параметри системи.

Розроблений сайт відповідає всім вимогам технічного завдання. Під час його розробки були використані сучасні технології, що використовуються для створення сайтів. Була застосована програмна платформа, що має високий рівень захисту, швидкодії та масштабованості. Також була розроблена та реалізована модульна архітектура сайту, яка також гарантує високий рівень відмовостійкості та швидкості роботи сайту.

Під розроблену архітектуру було створено структуру бази даних, що забезпечує надійне зберігання даних сайту та користувачів.

Подальше вдосконалення сайту є можливим завдяки реалізованій модульній архітектурі сайту. Така архітектура дозволяє створювати та впроваджувати нові модулі безболісно і не торкаючись основної екосистеми.

Основними напрямками для подальшої розробки може розглядатись додання адмін доступу, а також передання даних користувачів на базу даних і заміна сесії.

Основні результати курсової роботи полягають у наступному:

1 Під час аналізу та порівняння різних систем для управління вмістом web-сайтів стало очевидно, що ринок надає безліч рішень, кожне з яких добре справляється зі своїм завданням. Залежно від цілей та завдань організації сайту можна використовувати як безкоштовні, так і платні CMS. Так для забезпечення роботи сайту комерційної організації, де важлива стабільність роботи продукту, оперативність технічної підтримки, і не критична вартість продукту та технічної підтримки найдоцільніше використовувати комерційну CMS. Це може бути Бітрікс або DLE, не виключено, що будь-яка інша комерційна система зможе задовольнити всі запити організації.

2 Так організаціям некомерційним, тим, кому критична вартість продукту, і критично відсутність техпідтримки доцільно використовуватиме побудови сайту одну з безкоштовних CMS систем. З розглянутих у цій роботі CMS найбільш універсальною, з найбільшою кількістю модулів та найбільшою російською спільнотою розробників є CMS Joomla.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Boiko B., Content Management Bible, 2nd Edition, Wiley Publishing, Inc., 2005, ISBN: 0-7645-7371-3
2. 2. Bolin, D., Information Architecture for the World Wide Web, 3rd Edition, O'Reilly Media, Inc., 2006 , ISBN: 0596527349
3. 3. Floridi L., Information – A Very Short Introduction, Oxford University Press, 2010 ISBN 0-19-160954-4.
4. 4. Jenkins, T., D. Glazer, H. Schaper, Enterprise Content Management Technology: What You Need to Know. Open Text Corporation. 2004, ISBN 0-9730662-5-3
5. 5. Stallman R., Free Software, Free Society, Selected Essays of Richard M. Stallman, Second Edition, Free Software Foundation, Inc., 2010, ISBN 978-0-9831592-0-9
6. 6. Sabin-Wilson L., C. Miller, K. Palmer, A. Rennick, M. Torbert. WordPress All-in-One For Dummies, Wiley Publishing, 2011, ISBN 978-0-470-87701-2
7. What is Web CMS (or WCM)?
8. <<http://www.aiim.org/What-is-Web-CMS-WCM-System-Content-Management>>
9. Web Content Management Systems (Wcms): High-Impact Strategies – What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. Kevin Roebuck. Emereo Pty Limited, 2011.
10. CCNP Self-Study CCNP BCRAN Exam Certification Guide Second Edition [Електронний ресурс]. – Режим доступу: <https://docstore.mik.ua/cisco/pdf/routing/Cisco%20Press%20CCNP%20BCRAN%20Certification%20Guide.pdf>
11. Hierarchical internetworking model [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Hierarchical_internetworking_model
12. Ethernet [Електронний ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/Ethernet>

13. Frame (networking) [Электронный ресурс]. – Режим доступа:
[https://en.wikipedia.org/wiki/Frame_\(networking\)](https://en.wikipedia.org/wiki/Frame_(networking))
14. Frame synchronization [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/Frame_synchronization
15. CRC-based framing [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/CRC-based_framing
16. Asynchronous transfer mode (ATM) [Электронный ресурс]. – Режим доступа:
[https://en.wikipedia.org/wiki/Asynchronous_transfer_mode_\(ATM\)](https://en.wikipedia.org/wiki/Asynchronous_transfer_mode_(ATM))
17. Octet (computing) [Электронный ресурс]. – Режим доступа:
[https://en.wikipedia.org/wiki/Octet_\(computing\)](https://en.wikipedia.org/wiki/Octet_(computing))
18. Broadband Integrated Services Digital Network [Электронный ресурс]. – Режим доступа:
https://en.wikipedia.org/wiki/Broadband_Integrated_Services_Digital_Network
19. BPDU [Электронный ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/BPDU>
20. IEEE 802 [Электронный ресурс]. – Режим доступа:
https://uk.wikipedia.org/wiki/IEEE_802
21. LMSC, LAN/MAN Standards Comitee (Project 802) [Электронный ресурс]. – Режим доступа: <http://www.ieee802.org>
22. <<http://www.scribd.com/doc/62607306/Web-Content-Management-Systems-WCMS-High-impact-Strategies-What-You-Need-to-Know-Definitions-Adoptions-Impact-Benefits-Maturity-Vendors>>
23. Crash Course for Content Management: What is content management?. Vasont Systems.
24. <<http://www.vasont.com/resources/what-is-content-management.html>>
25. CMS Wiki. History Of CMS
26. <<http://www.cmswiki.com/tiki-index.php?page=HistoryOfCMS>>
27. The Content Management Comparison Tool
28. <<http://www.cmsmatrix.org/>>

29. March 2015 Web Server
Survey <<http://news.netcraft.com/archives/category/web-server-survey/>>
30. Programming Languages Market Share
<<http://www.w3cook.com/programminglanguage/python/>>
31. PHP vs ASP vs ASP.NET vs JavaScript vs CGI
32. <<http://www.wrensoft.com/zoom/benchmarks.html>>
33. Performance benchmarking – PHP, ASP, JSP, Coldfusion
<<http://www.linuxdocs.org/HOWTOs/PHP-HOWTO-13.html>>
34. Final Report on the CMS Selection for the Web Central Refresh
35. <http://sites.utexas.edu/refresh/files/2009/12/cms_evaluation_final_report.pdf>
36. A Guide to Web Content Management System Evaluation
37. <http://omniupdate.com/_resources/pdfs/white_papers/cms_evaluation_guide.pdf>
38. Comparing Open Source CMSs: WordPress, Joomla, Drupal and Plone
<<http://www.idealware.org/reports/2010-os-cms>>