

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Радіотехнічний факультет  
Радіотехнічних пристроїв та систем**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Сергій ЖУК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою**

**«Радіотехнічні інформаційні технології»**

**спеціальності 172 «Телекомунікації та радіотехніка»**

**на тему: «Програмний інструмент для адаптивного відображення даних  
в середовищі інформаційно-комунікаційної системи ПОЛІДАР»**

Виконала:

студентка IV курсу, групи РТ-71

Настенко Юлія Андріївна \_\_\_\_\_

Керівник:

к.т.н., асистент

Товкач Ігор Олегович \_\_\_\_\_

Рецензент:

к.т.н., асистент

Нікітчук Артем Валерійович \_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Студент (-ка) \_\_\_\_\_

Київ – 2021 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Радіотехнічний факультет**  
**Радіотехнічних пристроїв та систем**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіотехнічні інформаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Сергій ЖУК

«\_\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**  
**на дипломну роботу студенту**  
**Настенко Юлії Андріївні**

1. Тема роботи «Програмний інструмент для адаптивного відображення даних в середовищі інформаційно-комунікаційної системи ПОЛІДАР», керівник роботи Товкач Ігор Олегович к.т.н., асистент, затверджені наказом по університету від «18» травня 2021 р. №1205-с

2. Термін подання студентом роботи 11.06.2021 року

3. Вихідні дані до роботи: загальні вимоги до розробки програмного інструменту, який буде сумісний із системою «ПОЛІДАР» та будь-яким сайтом.

4. Зміст роботи: 1) Аналіз існуючих технологій для адаптивного відображення контенту на сайті; 2) ПОЛІДАР – паралельна обробка лінійних ірраціональних даних алгоритмами рекомбінації; 3) Розробка структури модульного візуального конструктора

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) презентація по роботі в обсязі не менше 10 слайдів.

6. Дата видачі завдання 12 квітня 2020р.

#### Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд існуючих рішень	13.04 – 18.04	Розділ 1
2	Порівняння різних аналогів	18.04 – 20.04	Розділ 1
3	Дослідження системи «ПОЛІДАР»	21.04 – 01.05	Розділ 2
4	Постановка задачі розробки програмного інструменту для адаптивного відображення даних	02.05 – 03.05	Розділ 3
5	Модифікація вибраного програмного засобу	04.05 – 15.05	Розділ 3
6	Розробка API для передачі даних	16.05 – 01.05	Розділ 3

Студент

Ю.А. Настенко

Керівник

І.О. Товкач

## АНОТАЦІЯ

До бакалаврської дипломної роботи Настенко Юлії Андріївни

На тему: «Програмний інструмент для адаптивного відображення даних в середовищі інформаційно-комунікаційної системи ПОЛІДАР »

В роботі розглядається розроблена інформаційна технологія для адаптивного відображення даних, яка дозволяє створювати або змінювати вже існуючий будь-який сайт, вдосконалювати його, додавати нові компоненти. Дана система містить в собі все необхідне аби створювати довільний контент та стилізувати його, а також редагувати в режимі реального часу, без навичок програмування.

Для реалізації програмного інструменту було використано фреймворк GrapesJs на базі якого написан API та розроблен модуль за допомогою JavaScript для передачі даних між клієнтом (функціоналами системи ПОЛІДАР) та конструктором.

Загальний об'єм роботи: 55 сторінок, 28 рисунків, 3 таблиці.

Ключові слова: Drag & Drop, ПОЛІДАР, інформаційні технології, динамічне відображення контенту.

## ANNOTATION

To the bachelor's thesis of Nastenka Yuliya Andriyivna

On the topic: "Programming tool for adaptive presentation of data in the middle of the information and communication system POLIDAR"

This diploma work considers the developed information technology for dynamic adaptive data display, which allows you to create or modify any existing site, improve it, add new components. This system contains everything you need to create arbitrary content and style it, as well as edit it in live mode.

To implement the software tool, the GrapesJs framework was used, on the basis of which an API was written and a module was developed using JavaScript to transfer data between the client (POLIDAR system functionalities) and the designer.

Total volume of work: 55 pages, 28 figures, 3 tables.

Keywords: Drag & Drop, POLIDAR, information technologies, dynamic display of content.

## ЗМІСТ

Перелік символів та скорочень.....	8
Вступ .....	10
1 Аналіз існуючих підходів до вирішення поставленої задачі.....	12
1.1 Аналіз існуючих технологій для адаптивного відображення контенту на сайті.....	12
1.2 Комерційні конструктори сайтів.....	13
1.3 Безкоштовні конструктори сайтів.....	17
1.4 WYSIWYG-редактори.....	21
1.5 Порівняльний аналіз найбільш функціональних конструкторів і візуальних редакторів.....	24
Висновки.....	26
2 ПОЛІДАР – паралельна обробка лінійних ірраціональних даних алгоритмами рекомбінації.....	27
Висновки.....	31
3 Розробка структури модульного візуального конструктора.....	33
3.1 Постановка задачі розробки програмного інструменту для адаптивного відображення даних.....	33
3.2 Додаткові вимоги до веб-сайту.....	33
3.3 Побудова інфологічної моделі предметної області.....	34
3.4 Побудова даталогічної моделі предметної області.....	36
3.5 Вибір програмних засобів.....	37
3.6 Сервер на Node.js і його особливості.....	37
3.7 JavaScript мова написання сценаріїв.....	38
3.8 Webpack-пакувальник модулів.....	38
3.9 Менеджер пакетів NPM.....	39
3.10 Архітектура програмного засобу.....	40
3.11 Алгоритм додавання контенту на довільний сайт.....	41
3.12 Взаємодія модульного візуального конструктора і функціоналів системи ПОЛІДАР.....	46

3.13 Файлова структура веб-сайту.....	48
Висновки.....	49
ЗАГАЛЬНІ ВИСНОВКИ.....	50
ПЕРЕЛІК ПОСИЛАНЬ.....	51
ДОДАТОК А.....	53

## ПЕРЕЛІК СИМВОЛІВ ТА СКОРОЧЕНЬ

ПОЛІДАР	Паралельна обробка лінійних ірраціональних даних алгоритмами рекомбінації
СЕД	Система електронного документування
HTML	HyperText Markup Language (мова розмітки гіпертекстових документів)
CSS	Cascading Style Sheets(Каскадні таблиці стилів)
WYSIWYG	What You See Is What You Get (що бачиш, те й отримуєш).
NPM	Node Package Manager(менеджер пакунків)
W3C	World Wide Web Consortium(Консорціум Всесвітнього павутиння)
CRM	Customer relationship management(Управління відносинами з клієнтами)
API	Application Programming Interface (Прикладний програмний інтерфейс)
DOM	Document Object Model (об'єктна модель документа)
VPS	Virtual Private Server (послуга, в рамках якої користувачу надається віртуальний виділений сервер)
SEO	Search Engine Optimization( Пошукова оптимізація сайтів)
SSL	Secure Sockets Layer(рівень захищених сокетів)



CMS	Content Management System(Система керування вмістом)
HTTPS	HyperText Transfer Protocol Secure
FTP	File Transfer Protocol
CDN	Content Delivery Network(Мережа доправлення контенту)

## ВСТУП

“

Сталість і завзятість у своєму намірі - речі такі чудові, що і наполеглива лінь вселяє повагу.

”

Л. А. Сенека.

Саме цю влучну цитату видатного філософа і поета античного Риму я хочу привести на початку. Адже що саме стимулює прогрес нашої цивілізації? Прагнення досконалості? Може, інноваційні технології, або навіть капіталізм? Ні - звичайна людська лінь. Ще на зорі свого існування, перші люди виготовляли примітивні інструменти, зброю, винаходили нові способи приготування і зберігання їжі для того, щоби спростити своє і без того не легке життя.

У наші часи, на щастя, можна спостерігати більш прогресивні методи ліквідації зайвих справ, проте проблеми вийшли на новий технологічний рівень. Кожна поважна компанія нині просто зобов'язана мати власну сторінку в інтернеті, магазини фокусуються на прибутку з інтернет-покупок, а ринок цифрових валют потрохи починає гойдати човен звичної нам економіки. За таких обставин, багато підприємців, власників різноманітних інформаційних та розважальних веб-сторінок або просто ентузіастів стикаються із проблемою масштабування свого інтернет-контенту та модернізації і розробки власних веб-платформ. Саме для них було створено безліч веб-застосунків для розробки сайтів, за допомогою яких, можна з абсолютного нуля та без навичок програмування створювати, наповнювати контентом та редагувати власні веб-сайти. В основі таких доступних і корисних програмних рішень лежить не менш корисна технологія генерації програмного JavaScript коду та гіпертекстової розмітки HTML і автоматичних стилів до неї за допомогою візуального композитингу через технологію drag'n'drop.

**Мета даної роботи** є дослідження і розробка програмного інструменту для динамічного адаптивного відображення контенту на будь-якому сайті.

**Практична цінність.** Можливість створювати та додавати на свій сайт компоненти, сторінки з адаптивним контентом, редагувати їх в лайф режимі. Це полегшить створення власного сайту або редагування вже існуючого.

# 1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

## 1.1 Аналіз існуючих технологій для адаптивного відображення контенту

Давайте розглянемо головні інструменти для створення контенту на сайті, розміщенні та стилізації: HTML – мова розмітки гіпертексту та CSS – каскадна мова таблиці стилів.

HTML – це файли, які мають розширення .html. За допомогою HTML можна створювати структуру сторінки, файл можна переглянути в будь-якому веб-браузері. Проект може мати кілька HTML сторінок: головна сторінка, блог, контакти, каталог тощо.

HTML документ складається з тегів або елементів, які містять атрибути

Атрибути – це унікальні властивості елементів, також в самому HTML файлі можна прописувати стилі та скрипти але прийнято це робити в окремих файлах CSS і JavaScript.

CSS – це каскадна мова таблиці стилів, яка використовується для стилювання елементів, написаних мовою розмітки, такою як HTML.

HTML і CSS тісно пов'язані між собою. HTML слугує для створення структури сайту, контенту тоді як CSS - це стиль (вся естетика сайту).

Ще одним з важливих інструментів для створення веб-сайтів є мова програмування JavaScript. Саме завдяки JavaScript можна реалізувати всю логіку на сайті. Більш детально про JavaScript буде описано в 3 розділі.

Але далеко не кожна людина володіє знаннями для створення професійних веб-сайтів, а гарний сайт в наш час необхідний кожній поважній компанії. Саме для того аби полегшити створення власного веб-додатку або вносити зміни до вже існуючого сайту почали розробляти конструктори сайтів

та WYSIWYG-редактори, за допомогою, яких кожна людина без навиків програмування могла б створити власний веб-сайт.

Головна ідея конструкторів та візуальних редакторів полягає в тому аби зробити доступною збірку веб-сайту для людей без технічних знань HTML, CSS, JavaScript, адміністрування веб-ресурсів. Аби створення веб-сайту за складністю було на рівні публікації в соціальних мережах.

Розглянемо найбільш поширені із них.

## **1.2 Комерційні конструктори сайтів**

В даний час існує безліч комерційних конструкторів сайтів. Перерахуємо деякі з них: WIX, Weebly, uKit, Google Sites, Webnode, Jimbo, uCraft, Webstarts, Tilda, Bookmark, Webflow, xWeb.

Дослідимо наступні продукти, відібрані по частоті і якості згадки в пошукових системах Google.com і Yandex.ru, а так само згідно даних рейтингів конструкторів:

- WIX;
- Tilda;
- uKit.

### **Tilda**

Tilda – це модульний конструктор з візуальним редактором. Для роботи з Tilda не потрібно знати мови програмування. Всі дії виконуються в візуальному редакторі, єдина складність - настройка блоків. Це займає досить багато часу. Блоки адаптуються під мобільну версію, можна зробити так, що на комп'ютері користувач бачитиме одні блоки, а на телефоні інші.

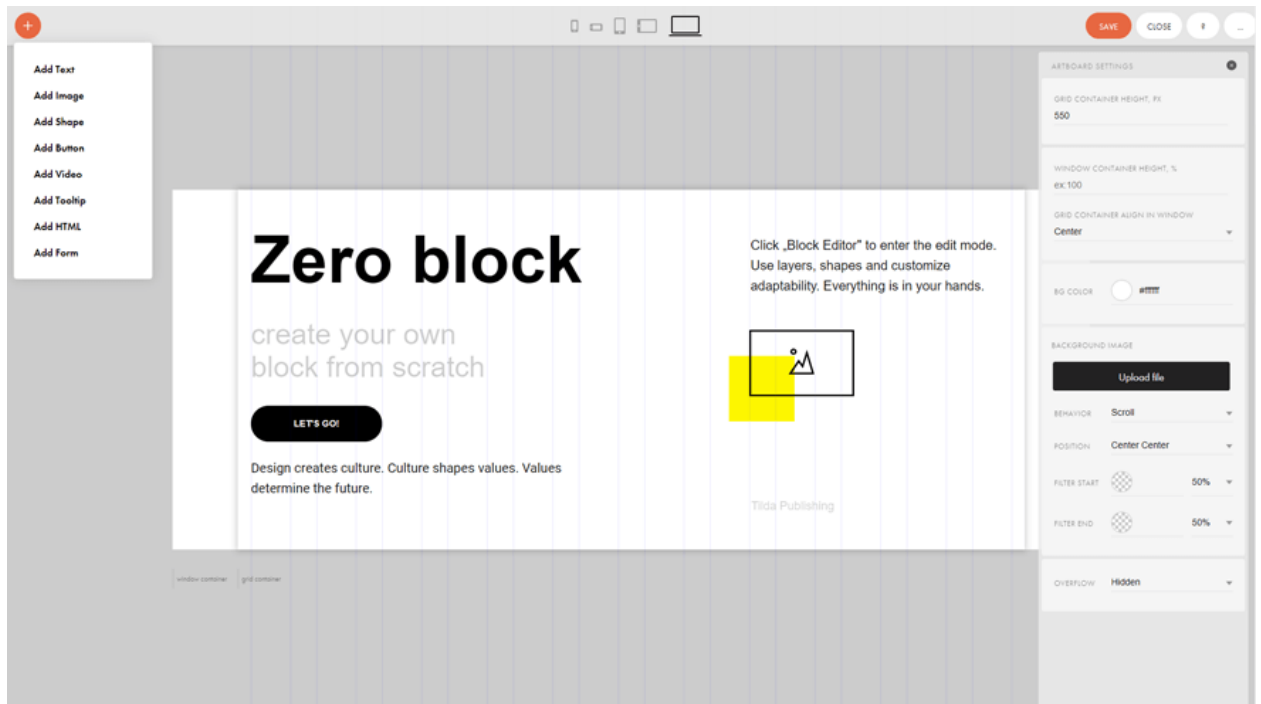


Рисунок 3.1 – Зовнішній вигляд конструктора Tilda

### Переваги

1. Шаблони сторінок і гнучкі настройки дизайну.
2. Наявність вбудованої CRM;
3. Використання технологій для прискорення завантаження сторінок - Lazy Load, CDN, адаптивне стиснення зображень;
4. Інтеграція зі сторонніми сервісами для розширення функціональності: соцмережами, музичними платформам, платіжними системами, картами і т.д.
5. Вбудовані інструменти для створення інтернет-магазину.

### Недоліки

Головний недолік Tilda - висока вартість.

### WIX

WIX – універсальний сервіс для створення сайтів з великим вибором шаблонів і різними способами організації роботи, між якими користувач вибирає в залежності від власного досвіду.

Wix пропонує кілька способів створення сайту. Наприклад, початківцям користувачам буде зручно отримати шаблон за допомогою Wix ADI – штучного інтелекту, який задає кілька питань, а потім аналізує відповіді і пропонує відповідний макет. Досвідчені розробники використовують просунуті інструменти, доступні в режимі Wix Corvid.

Візуальний редактор Wix дозволяє зібрати сайт будь-якого типу, не маючи навичок програмування. Сторінки наповнюються контентом за допомогою блоків, які можна додавати, видаляти і редагувати. Функціональність системи розширюється за рахунок додатків з вбудованого магазину Wix App Market.

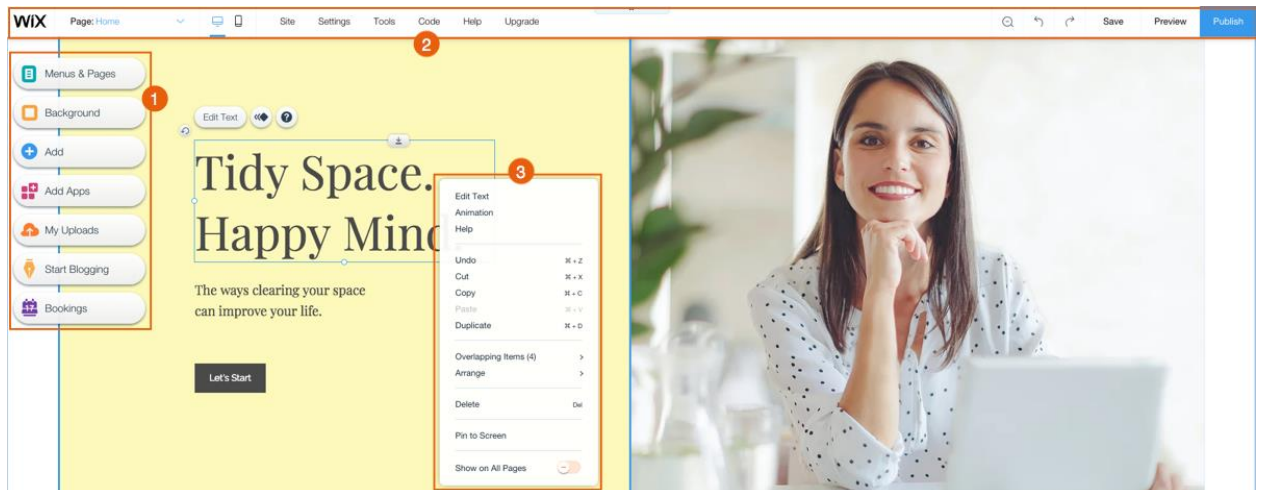


Рисунок 3.2 – Зовнішній вигляд конструктора WIX

### Переваги

1. Потужні інструменти редагування, що забезпечують глибоку кастомізацію дизайну.
2. Інтеграції зі сторонніми сервісами через додатки.
3. Можливість вставки HTML-коду.
4. Інструменти розробника, що дозволяють створювати власні додатки для розширення базової функціональності конструктора.

### Недоліки

1. Підключення власного домену є тільки на платних тарифах.

2. Спочатку важко розібратися в перевантаженій функціоналом адмін панелі.

## uKit

uKit – зручний конструктор сайтів, який пропонує дуже простий в освоєнні візуальний редактор. Завдяки жорсткій верстці дизайн проекту налаштовується за кілька годин навіть при відсутності досвіду. Можливості редактора обмежені, що дозволяє навіть не досвідченим користувачам швидко зробити привабливі варіанти дизайну на основі стандартного шаблону.

Головний плюс uKit – простий редактор, який дозволяє за кілька годин зібрати сайт, не відрізняється за якістю від товарів професійних розробників.

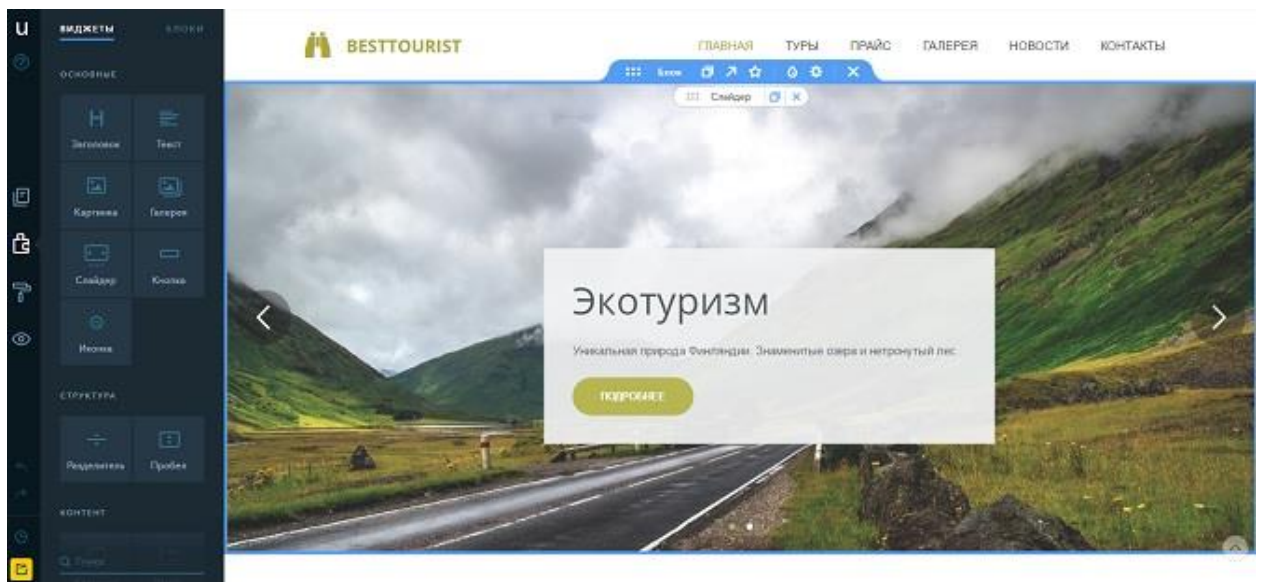


Рисунок 3.3 – Зовнішній вигляд конструктора uKit

## Переваги

1. Багата база віджетів.
2. Підтримка сторонніх додатків через інтеграції.
3. Гнучкі настройки адаптивності з можливістю приховування / показу окремих блоків в залежності від дозволу екрану.

## 1.3 Безкоштовні конструктори сайтів



В даний час існує безліч безкоштовних конструкторів сайтів.

Дослідимо наступні продукти:

- GrapesJs;
- uCoz;
- Ucraft.

## GrapesJs

Конструктор з відкритим кодом. Має простий та зрозумілий інтерфейс, не вимагає спеціальних знань. Підійде для створення Landing Page. Також можна окремо налаштовувати десктопну і мобільні версії. GrapesJs є абсолютно безкоштовним. Основна відмінність GrapesJs від інших конструкторів це відкритий код, розробники можуть удосконалювати конструктор, додавати нову логіку.

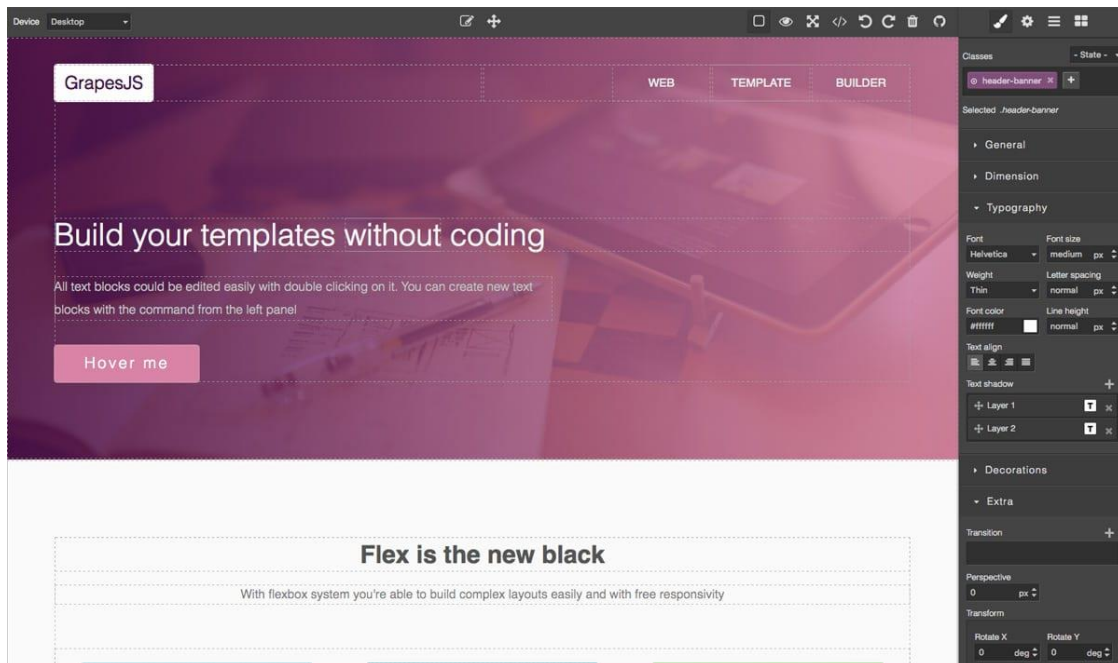


Рисунок 3.4 – Зовнішній вигляд конструктора GrapesJs

## **Переваги**

1. Абсолютно безкоштовний.
2. Відкритий код.
3. Простий і зрозумілий інтерфейс.
4. Постійне оновлення і додавання нових можливостей.

## **Недоліки**

Не можна створювати великі проекти лише Landing Page

## **uCoz**

uCoz – потужний модульний конструктор, який можна порівняти з CMS за своїми можливостями. Пропонує візуальний редактор, зручний для використання початківцями веб-майстрами, відкритий код і доступ до API для досвідчених розробників. uCoz - один з небагатьох конструкторів, на яких можна безкоштовно підключити домен.

Функціональність конструктора залежить від того, які модулі підключені до движку. Наприклад, можна додати блок «Новини» і створити інформаційний портал або підключити форум. Велика частина модулів доступна безкоштовно.

При безкоштовному використанні конструктора користувач отримує 400 Мб вільного місця, мобільну версію сайту, можливість прикріпити свій домен, FTP доступ і постинг в соцмережі з блогу і розділу з новинами до 10 записів в день. Завдяки відкритому коду веб-майстер може повністю змінити зовнішній вигляд і функціональність сайту.

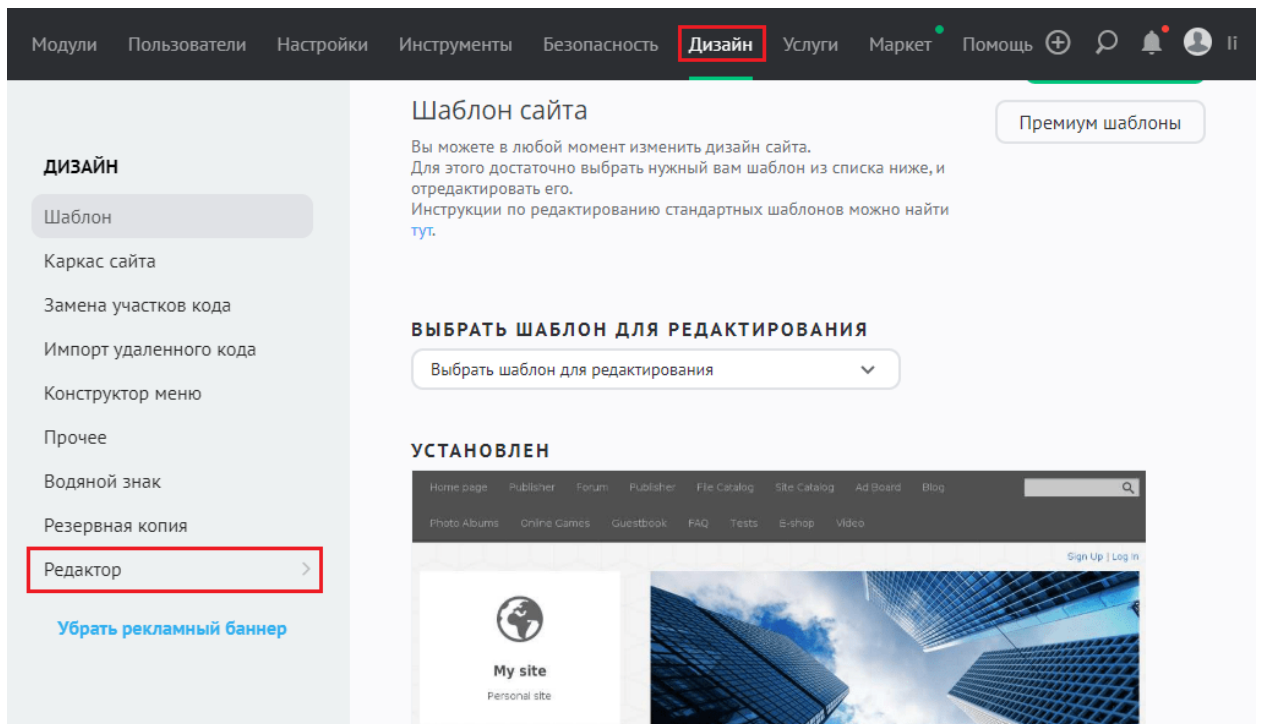


Рисунок 3.5 – Зовнішній вигляд конструктора uCoz

## Переваги

1. Функціональні модулі, велика частина яких доступна безкоштовно.
2. Відкритий код і доступ до API, що дозволяє досвідченим розробникам модифікувати конструктор під свої потреби.
3. Можливість додати власний шаблон.
4. Можна безкоштовно підключити домен

## Недоліки

1. Мала кількість привабливих шаблонів
2. Обмеження дискового простору

## Ucraft

Ucraft – безкоштовний конструктор для малого і середнього бізнесу. Підходить для створення візиток, портфоліо, сайтів компаній, інтернет-магазинів. Пропонує готові шаблони і бібліотеку блоків для вибудовування інтерфейсу з нуля. Відразу після реєстрації ви можете підключити домен, встановити SSL сертифікат і створити необмежену кількість сторінок.

Серед інших можливостей безкоштовного тарифу відзначимо:

1. Безлімітний пропускна здатність.
2. Інструменти для SEO-оптимізації сторінок.
3. Підключення аналітики від Google.
4. Цілодобова підтримка.
5. Захист сторінок паролем.

Якщо ви вирішите створити на Ucraft бізнес-сайт або магазин, то доведеться переходити з безкоштовного тарифу на один з преміум-пакетів послуг.

Залишаються також претензії до швидкодії дашборда і візуального редактора. Іноді відгуку доводиться чекати досить довго. В іншому претензій до функціональності немає.

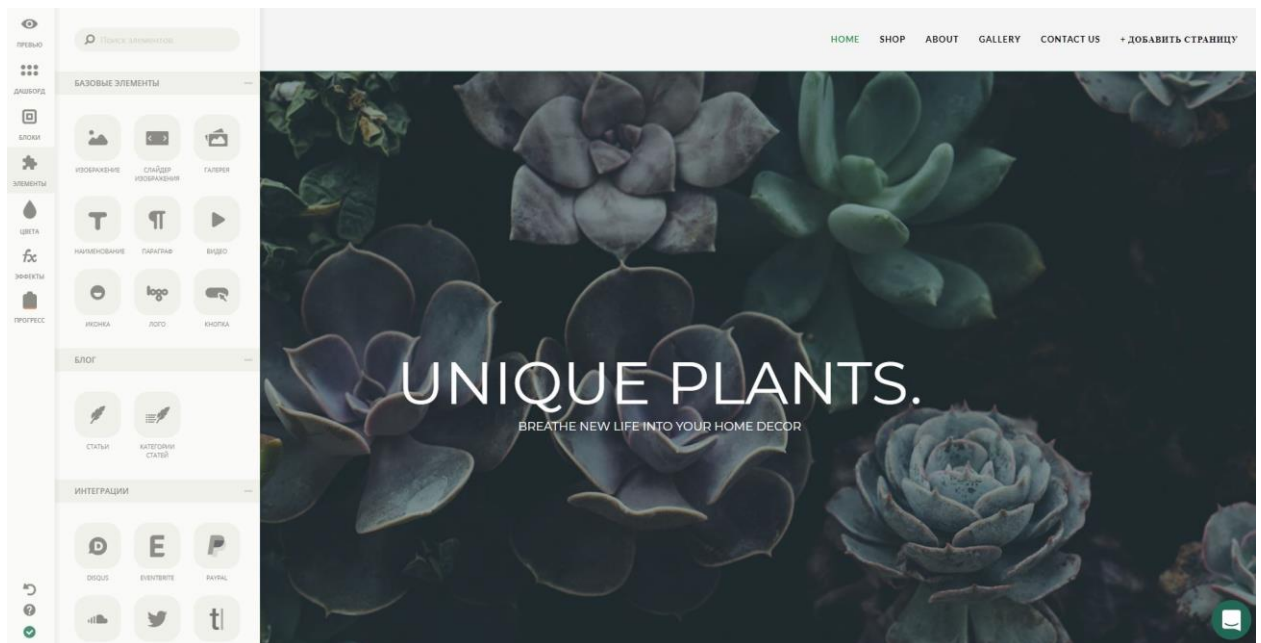


Рисунок 3.6 – Зовнішній вигляд конструктора Ucraft

### Переваги

1. Безкоштовне підключення домену
2. Необмежений трафік.
3. Безлімітна кількість сторінок на сайті.
4. Стандартні можливості для SEO-оптимізації сторінок.

5.Налаштування видимості, що дозволяють створювати різні версії сайту.

### Недоліки

Головний недолік безкоштовного тарифу на Ucraft - відсутність інтеграцій та неможливість додати на сторінки свій код. Це обмежує функціональність конструктора, але для багатьох користувачів не є критичним мінусом.

## 1.4 WYSIWYG-редактори

WYSIWYG-редактор – це візуальне оформлення сайту і його веб-сторінки, починаючи від зовнішнього дизайну і закінчуючи проектуванням структури, включаючи навігацію і движки.

Розглянемо найбільш популярні візуальні редактори. Серед них це:

1. TinyMCE
2. CKeditor
3. Draft.js

### TinyMCE

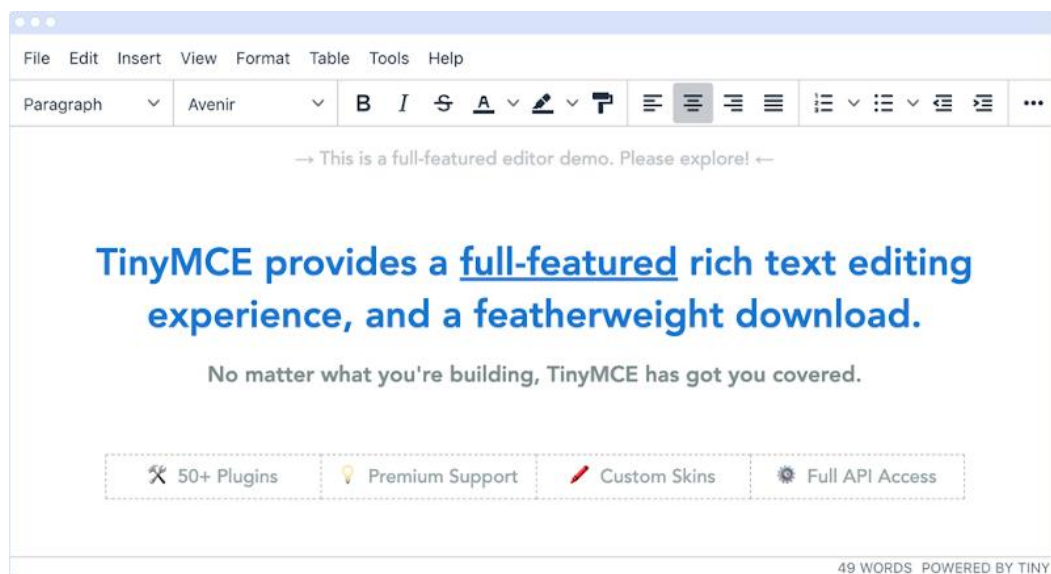


Рисунок 3.7 – Зовнішній вигляд конструктора

Це модульна система для інтеграції блоків контенту в уже готові сайти.

Вона підключається до ресурсів через скрипт з посиланням на CDN-сервер, де зберігаються дані TinyMCE, і дає можливість додавати на ресурс текстові матеріали. TinyMCE Editor легко інтегрується з популярними фреймворками в дусі React, Angular, Vue.js, Bootstrap і jQuery, буквально вбудовуючись в їх інтерфейс у вигляді нативного програмного забезпечення.

Однією з ключових особливостей продукту, яку виділяють розробники, є адаптивність. Редактор TinyMCE спочатку створювався з прицілом на мобільні пристрої і легку розширюваність (при необхідності). Вбудований редактор можна «прокачати» за допомогою плагінів з офіційною галереї. Вони допомагають знаходити помилки, швидко редагувати контент і оформляти його у відповідності зі своїми уявленнями про красу.

## CKeditor

### A simple page with CKEditor

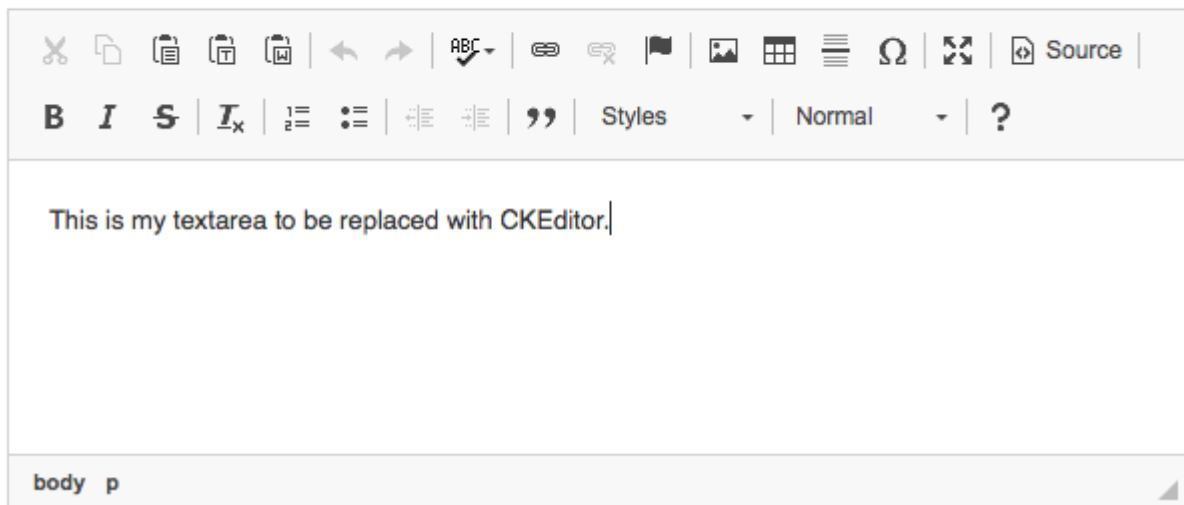


Рисунок 3.8 – Зовнішній вигляд конструктора CKeditor

CKEditor підтримує перенесення тексту з інших редакторів, таких як Word, Google Docs або Excel, в першочерговому вигляді. З Excel можна витягнути всю таблицю, а CKEditor коректно відобразить кожен колонку і дозволить редагувати інформацію в них.

Редактор за замовчуванням дозволяє вставляти в текст фотографії, відеоролики, тощо. Також підтримуються віджети, математичні формули і вставка коду з підсвічуванням. Для CKEditor розроблено з десяток плагінів, які допомагають автоматично завершити частину тексту або замінити його іншим контентом.

### Draft.js

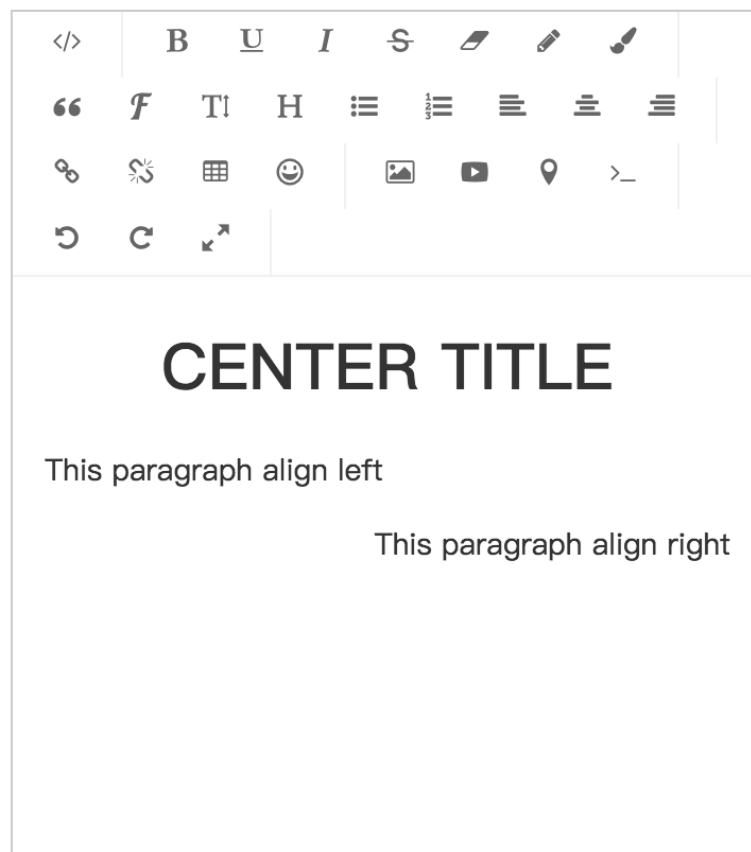


Рисунок 3.9 – Зовнішній вигляд конструктора Draft.js

WYSIWYG-редактор, розроблений спеціально для фреймворка React і включає в себе набір всіх необхідних інструментів для роботи з текстом.

Як і у випадку з іншими подібними редакторами, контент відображається у вигляді блоків. Кожен блок містить певний тип даних. Це може бути як текст, так і медіаконтент.

Draft сповідує базові принципи React і строго дотримується їх. Тому він легко інтегрується в додатки, створені з використанням цього фреймворка, а технічні аспекти, що стосуються рендеринга або введення і виведення тексту, зводять в абстракцію. Доступ до функцій Draft здійснюється через добре задокументоване API. А ще, щоб уникнути витоків пам'яті і проблем в продуктивності редактора, тут використовується immutable-js, що вигідно відрізняє його від конкурентів.

### 1.5 Порівняльний аналіз найбільш функціональних конструкторів та візуальних редакторів

На основі дослідження проведеного вище побудуємо таблицю, в якій дамо порівняльні характеристики (табл.1.1).

Таблиця 1.1 – Порівняльний аналіз найбільш функціональних конструкторів та редакторів

Порівняння конструкторів сайту	
Плюси безкоштовних конструкторів	Мінуси безкоштовних конструкторів
<ul style="list-style-type: none"> <li>- Вони безкоштовні.</li> <li>- Є безліч шаблонів. Можна не витратитися на дизайн.</li> <li>- Простий і зрозумілий інтерфейс.</li> <li>- Документація. Без проблема можна знайти в мережі вичерпну документацію.</li> <li>- Постійне оновлення і додавання нових можливостей.</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежений дизайн.</li> <li>- Більшість безкоштовних конструкторів не має безкоштовного підключення домену</li> <li>- Не можна створити великі проекти</li> </ul>
Плюси платних движків	Мінуси платних движків
<ul style="list-style-type: none"> <li>- Великий вибір готових шаблонів.</li> </ul>	<ul style="list-style-type: none"> <li>- Вони платні.</li> </ul>



<ul style="list-style-type: none"> <li>- Зручний інтуїтивно зрозумілий інтерфейс.</li> <li>- Мобільна і десктопна версії редагуються окремо.</li> <li>- Можливість створити свій унікальний макет з нуля.</li> <li>- Можливість додати свої елементи за допомогою HTML, CSS і JS.</li> <li>- Інтеграція з CRM, сервісами зворотних дзвінків, онлайн-чатами і електронними платіжними системами.</li> <li>- Можливість зберегти і використовувати для подальшої роботи власні шаблони.</li> <li>- Підключення HTTPS в інтерфейсі конструктора.</li> <li>- Інструменти для SEO</li> </ul>	<ul style="list-style-type: none"> <li>- Складно розібратися в налаштуваннях.</li> </ul>
<p><b>Порівняння WYSIWYG-редакторів</b></p>	
<p><b>Плюси WYSIWYG-редакторів</b></p>	<p><b>Мінуси WYSIWYG-редакторів</b></p>
<ul style="list-style-type: none"> <li>- Робота з WYSIWYG редакторами не вимагає попередньої підготовки</li> <li>- Підтримка перенесення інформації безпосередньо з Word.</li> <li>- Робота з WYSIWYG редакторами подібна роботі з Word.</li> <li>- Мінімізація витрат на підтримку web-сайту.</li> </ul>	<ul style="list-style-type: none"> <li>- Надлишкова кількість зайвих тегів</li> <li>- збільшення коду сторінки і менша привабливість сторінки для пошукових систем</li> <li>- можливе порушення єдиного стилю web-сайту</li> </ul>

## Висновки

У результаті аналізу існуючих конструкторів веб-сайтів (див. таб. 1.1) як кращих безкоштовних GrapesJs, uCoz, Ucraft, так і кращих платних Tilda, WIX і uKit, приходимо до висновку, що в залежності від того для якої мети та який саме сайт необхідний (Landing Page, корпоративний сайт або інтернет-магазин тощо) можна використовувати як платні так і безкоштовні конструктори. Наприклад, для сайту-візитка, де немає великої кількості даних можна використовувати безкоштовні конструктори але необхідно враховувати, що у більшості безкоштовних конструкторів платний домен. Для таких сайтів добре підійдуть конструктори Ucraft, GrapesJs і uCoz оскільки в них можна безкоштовно підключити домен. Для корпоративних сайтів, великих проектів краще використовувати платні конструктори такі як WIX, uKit або Tilda оскільки їх функціонал набагато більший, більше можливостей, а також не має обмеження дискового простору.

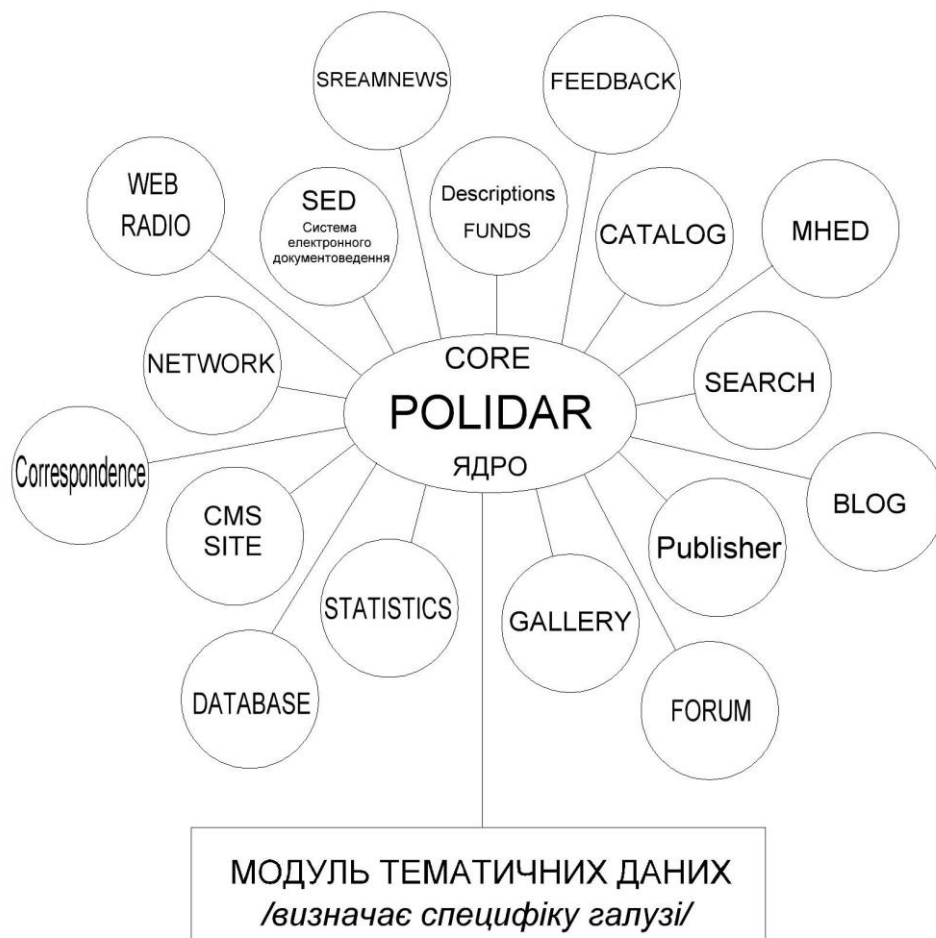
У результаті аналізу існуючих WYSIWYG-редакторів таких як CKeditor, TinyMCE та Draft.js, приходимо до висновку, що з візуальними редакторами добре працювати якщо у вас вже є готовий сайт і ви хочете додати контент без написання коду. З таб. 1.1 можна побачити, що візуальні редактори мають чимало недоліків, які можуть вплинути на відображення контенту.

Для створення компоненту для адаптивного відображення даних не підійшов жодний із візуальних редакторів, оскільки вони мають ряд суттєвих недоліків і тому використання редактора є неприйнятним. Підійшов конструктор GrapesJs з відкритим кодом, в якому є все необхідне для реалізації адаптивного відображення контенту. Необхідно буде розробити модуль, а також написати API завдяки чому можна буде використовувати компонент в функціоналі системи «ПОЛІДАР», а також в будь-якому сайті.

## 2 «ПОЛІДАР» – ПАРАЛЕЛЬНА ОБРОБКА ЛІНІЙНИХ ІРАЦІОНАЛЬНИХ ДАНИХ АЛГОРИТМАМИ РЕКОМБІНАЦІЇ

**Інформаційно-комунікаційна система «ПОЛІДАР»** – це програмний менеджер-органайзер, який керує та узгоджує роботу on-line доступних тематичних функціоналів, які по суті є окремими незалежними системами спеціалізованого призначення, що пов’язані між собою і функціонують як єдине ціле (тобто, як система), яких в свою чергу, може бути практично необмежена кількість. На даний час їх вже більше двадцяти і їхня кількість постійно збільшується. Їх кількість визначається кількістю розробок спеціалістами галузі алгоритмів потрібних їм інструментів чи окремих функціоналів, які програмуються та доповнюють Систему [13].

Структурна схема інформаційно-комунікаційної системи «ПОЛІДАР» показана на рис.2.1.



## Рисунок 2.1 Інформаційно-комунікаційна система «ПОЛІДАР»

Основними функціоналами системи є:

**CMS SITE** – функціонал для створення галузевого сайту;

**NETWORK** – сервіс для об'єднання окремих сайтів в мережу за допомогою порталу (потребує подальшої розробки);

**SEARCH** – пошукові системи різного призначення, електронні довідники (каталоги);

**Descriptions FUNDS** – описи фондів, довідник з розташування фондів у сховищах, і т.д. і т.п.);

**DATABASE** – база даних (система для упорядкованого зберігання цифрових копій документів);

**USER MODULE** – користувацький модуль (забезпечує в автоматичному режимі регламентований доступ користувачів до цифрових копій документів);

**GALLERY** – галерея (функціонал, який призначений для створення різної складності багатопрофільних мультимедійних виставок документів та перегляду їх в режимі on-line;

**FORUM** – сервіс для створення та обговорення тем;

**CHAT** – сервіс для швидкого обміну повідомленнями;

**PLAN-REPORT** – план-звіт (дозволяє в напівавтоматичному режимі готувати плани-звіти);

**WEB-RADIO** – інтернет-радіо (функціонал, який призначений для створення архівістами радіопередач для трансляції їх не в ефірі, а в інтернеті;

**Publisher** – переглядач-ілюстратор (функціонал, який красиво листає документи, що викладені на сайті архівної установи, і водночас забезпечує їх захист від копіювання відвідувачами сайту;

**BLOG** – діловий щоденник установи сервіс для фіксації поточної діяльності установи;

**BACKUP** – резервна копія сайту (сервіс для збереження свого сайту на власному комп'ютері);

**STREAM AUDIO&VIDEO** – сервіс для проведення аудіо та відео конференцій;

**ASSISTANCE** – сервіс для дистанційної допомоги користувачам системи «ПОЛІДАР»;

**SED** – система електронного документообігу (на даний час забезпечує функціонування десяти електронних журналів, які мають власні пошукові системи та багатoproфільні системи збирання і підрахунку статистичних даних);

**MHED** – система шифрування даних для передачі їх відкритим каналом зв'язку (мережею інтернет);

**STREAMNEWS** – інформаційний модуль (функціонал, який автоматично збирає новини з усіх сайтів архівних установ області та розміщує їх на порталі мережі);

**Correspondence** – листування в мережі та пересилання даних (дані можуть шифруватись за допомогою методу шифрування MHED /Multilayer Hybrid Encryption and Decryption/);

**FEEDBACK** – сервіс для інтерактивного зворотнього зв'язку з користувачами архівів;

**STATISTICS** – система для глобального збирання відкритих статистичних даних про діяльність архівних установ всієї Київської області (функціонує в реальному часі), та ін.

Система «ПОЛІДАР» має можливість підлаштовуватись під різні вимоги користувача. Зв'язує користувача з програмним забезпеченням, яке розміщено на VPS сервері.

Принцип роботи показаний на рис. 2.2.

Система «ПОЛІДАР» може підлаштовуватись під будь-які вимоги користувачів.

Наприклад:

Користувач №1. Користувач має вже розроблений сайт, який написаний ним самим, або за допомогою будь-якої CMS системи. В такому випадку система «ПОЛІДАР» обслуговує користувача наступним чином. Користувач звертається до системи, безпосередньо до Ядра (Командер), Ядро зв'язує його з хостингом, місце, де зберігається сайт. В наслідок цього встановлюється зв'язок «КОРИСТУВАЧ - ЯДРО-ХОСТИН».

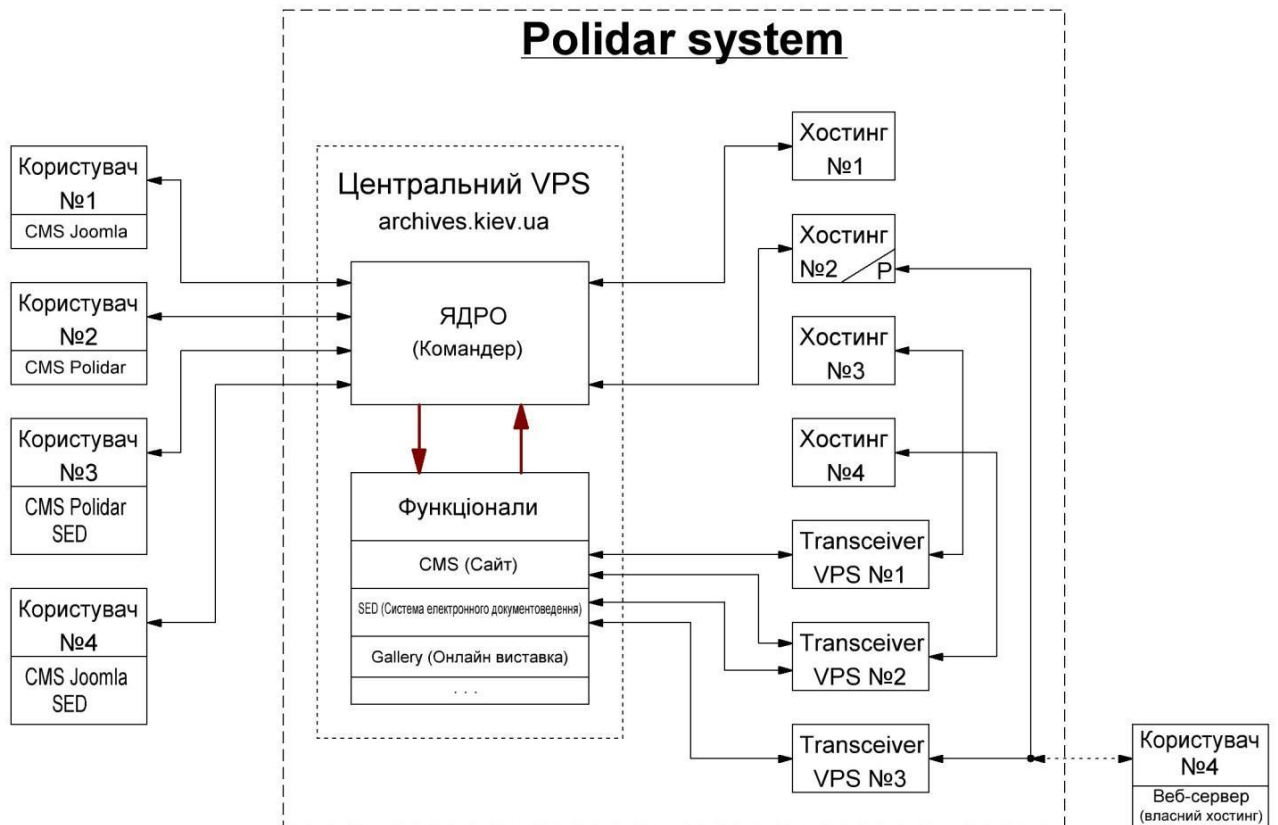


Рисунок 2.2 Схема, що пояснює принцип роботи системи

Користувач №2. Користувач не має взагалі ніякого сайту, та навичок щоб його створити, але йому потрібен сайт. В такому випадку система «ПОЛІДАР» автоматично створює сайт і обслуговує користувача наступним чином. Користувач звертається до системи, до Ядра (Командер), Ядро зв'язує його з функціоналами «ПОЛІДАР», одним із них є CMS, та з'єднує з VPS Transceiver, де розміщується допоміжні компоненти функціонала, які мають доступ до

хостинга, де зберігаються всі дані від сайту. В наслідок цього встановлюється зв'язок «КОРИСТУВАЧ-ЯДРО-ФУНКЦІОНАЛ-ХОСТИНГ».

Користувач №3. Користувач не має взагалі ніякого сайту, та навичок щоб його створити, але йому потрібен сайт, а також йому потрібнен інструмент документоведення SED. В такому випадку система «ПОЛІДАР» автоматично створює сайт та надає доступ до SED, і обслуговує користувача наступним чином. Користувач звертається до системи, до Ядра (Командер), Ядро зв'язує його з функціоналами «ПОЛІДАР» та з'єднує з VPS Transceiver, де розміщується допоміжні компоненти функціонала, які мають доступ до хостинга, де зберігаються всі дані від сайту та від інструмента документоведення. В цього встановлюється зв'язок «КОРИСТУВАЧ-ЯДРО-ФУНКЦІОНАЛИ-ХОСТИНГ».

Користувач №4. Користувач має вже розроблений сайт, який написаний ним самим, або за допомогою будь-якої CMS системи, а також йому потрібен інструмент документоведення SED. В такому випадку система «ПОЛІДАР» надає доступ до SED, і обслуговує користувача наступним чином. Користувач звертається до системи, до Ядра (Командер), Ядро зв'язує його з хостингом, місце, де зберігається сайт, зв'язує його з функціоналами «ПОЛІДАР» та з'єднує з VPS Transceiver, де розміщується допоміжні компоненти функціонала, які мають доступ до хостинга, де вже розміщені дані від сайту, поряд з ними зберігаються дані від інструмента документоведення. В наслідок цього встановлюється зв'язок «КОРИСТУВАЧ-ЯДРО-ХОСТИНГ або ФУНКЦІОНАЛИ-ХОСТИНГ».

## **Висновки**

Таким чином система «ПОЛІДАР» має достатню кількість гнучких функціоналів, які можуть задовольнити користувача з будь якими вимогами, і

ефективно використовуватися без допомоги висококваліфікованих комп'ютерних спеціалістів.



## **3 РОЗРОБКА СТРУКТУРИ МОДУЛЬНОГО ВІЗУАЛЬНОГО КОНСТРУКТОРА**

### **3.1 Постановка задачі розробки програмного інструменту для адаптивного відображення даних**

Необхідно розробити систему, яка буде функціонувати як модульний візуальний drag'n'drop конструктор, створений на базі бібліотеки GrapesJs.

Передачу даних необхідно організувати за допомогою веб-сервера написаного на Node.js. Також для реалізування даної програми необхідно створити модуль на JavaScript, який потрібно буде встановити на веб-сайт аби ми могли отримати дані з конструктора. Дана система передбачає можливість створення адаптивного контенту, який буде імпортуватися та динамічно відображатися на будь-якому сайті. А також буде оновлюватися в режимі реального часу.

### **3.2 Додаткові вимоги до веб-сайту**

Грунтуючись на виборі технології для створення модульного редактора можна також сформулювати додаткові вимоги до веб-сайту:

- дана система легка в використанні, потребує знань і навиків на рівні користувача комп'ютера середнього рівня;
- необхідність імпортування спеціального модуля для функціонування створенної системи;
- модифікація веб-сайта з незначними трудовими витратами;
- сайт повинен бути розроблений за допомогою сучасних технологій Webpack;
- npm та node.js необхідні бути встановленими на комп'ютері користувача.

Відповідно до висунутих вимог необхідно сформувати структуру сайту. Для цього розробити інфологічну та даталогічну моделі предметної області,

визначити програмні засоби, сервер, мову для написання модуля, розробити архітектуру програмного засобу та файлову структуру сайту.

### **3.3 Побудова інфологічної моделі предметної області**

Інфологічна модель предметної області – це опис предметної області, виконаний з використанням спеціальних мовних засобів, що не залежать від використання надалі програмних засобів. Вимоги до інфологічної моделі:

- адекватність відображення предметної області;
- несуперечливість;
- відображення потреби всіх користувачів майбутньої інформаційної системи;
- допущення можливих розширень моделі, а також при великій розмірності інфологічної моделі;
- композиції і декомпозиції; бажано застосування мови опису предметної області як при ручному, так і при автоматизованому проектуванні;
- легкість сприйняття різними категоріями користувачів.

Інфологічна модель є ядром системи проектування. Вона містить всю необхідну інформацію для проектування програми.

Для опису інфологічної моделі даних можна виділити вісім об'єктів, які беруть участь в обробці інформації:

- "Девайс";
- "Перегляд";
- "Повноекранний режим";
- "Перегляд коду";
- "Стилі";
- "Налаштування";
- "Структура блоків";
- "Блоки";

Кожен об'єкт має свої властивості:

Об'єкт "Девайс" змінює пристрій під екран якого ми робимо сайт. Це необхідно для адаптивного відображення контенту. Даний об'єкт містить в собі десктопну версію, планшетну та мобільну.

Об'єкт "Перегляд" прибирає панель інструментів, дозволяє нам побачити макет як він буде виглядати .

Об'єкт "Повноекранний режим" прибирає вкладки браузера зверху, працює як кнопка F11.

Об'єкт "Перегляд коду" дозволяє нам переглянути верстку та стилі сформованого блоку або блоків. Містить в собі кнопку збереження ("Export to ZIP"), при натисканні якої зберігає наш контент у вигляді ZIP архіва. Кнопка "Upload" передає дані до сервера.

Об'єкт "Стилі" дозволяє стилізувати контент. Включає в себе загальні стилі, розмірність, стилізацію шрифтів, декор та додаткові стилі. В загальні стилі входить розміщення об'єкту та його позиція відносно екрану або інших елементів. До розмірності входить висота, ширина та відступи компонента. До декору входять колір та тіні, а додаткові стилі містять переходи.

Об'єкт "Налаштування" дозволяє змінювати атрибути елемента. В залежності від того, який елемент вибран атрибути цього елемента ми можемо змінювати.

Об'єкт "Структура блоків" дозволяє змінювати структуру DOM-дерева.

Об'єкт "Блоки" дозволяє створювати різні блоки. Містить додавання навігаційної панелі, картинок, відео, блоків з текстом, колонки, посилання, а також додавання форми зворотнього зв'язку.

## Структура взаємозв'язків інфологічної моделі предметної області

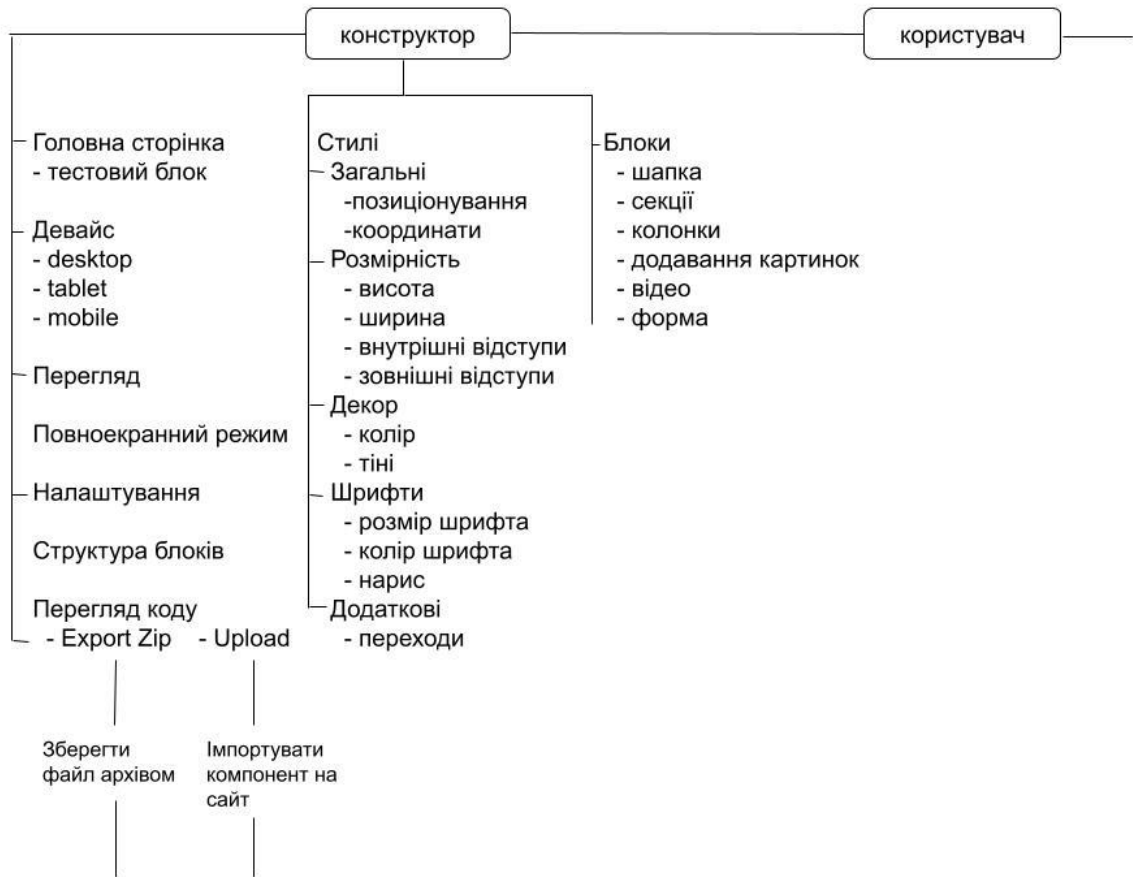


Рисунок 3.1 Структура взаємозв'язків інфологічної моделі предметної області

### 3.4 Побудова даталогічної моделі предметної області

В даній розробці відсутня база даних. Верстка та стилі компонента зберігаються в json об'єкт в масиві.

В масиві містяться об'єкти з трьома параметрами:

1. html
2. css
3. id

### 3.5 Вибір програмних засобів

Сучасні веб сайти стають все складнішими та громісткими, багато даних потрібно передавати та зберігати. Тому для полегшення веб-сайтів створюють різні фреймворки, збірки модулів, які полегшують завантаження сайту

Для розробки у даній роботі було доцільно обрати мову JavaScript для написання модуля разом з сервером написаним за допомогою Node.js. Для роботи потрібна установка Node.js.

### 3.6 Сервер на Node.js і його особливості

Node.js – це не мова програмування, а сфера виконання мови програмування. Node.js – це програмне забезпечення на базі сервера, яке містить заздалегідь визначені процеси для виконання конкретних завдань. Кожен процес Node.js виконується на сервері як серверний час виконання; працює по суті над внутрішнім аспектом програми управління даними. Наприклад, якщо необхідне зберегти деякі дані у файлі чи базі даних, вам потрібно використовувати мову або додаток на стороні сервера. Node.js позначено як час виконання JavaScript, оскільки він використовує JavaScript для запуску внутрішніх процесів.

Node.js побудований на движку JavaScript V8, який використовується для компіляції і виконання вихідного коду JavaScript. Тому, коли виконуєте JS-скрипт за допомогою Node.js, цей код спочатку передається в движок JavaScript V8. Потім механізм JavaScript V8 компілює сценарій і передає результат компіляції назад в Node.js, де його можна використовувати в додатку. Розробники JavaScript тепер можуть створювати і виконувати серверні процеси (такі як створення файлу і збереження в ньому даних) за допомогою Node.js.

### 3.7 JavaScript мова написання сценаріїв

Для написання модуля, з допомогою якого можна було динамічно відображати контент, який ми хочемо аби був на сайті, було обрано мову програмування JavaScript. Мова JavaScript є універсальною мовою для написання сценаріїв, виконуваних на веб-сторінках.

JavaScript об'єктно-орієнтована прототипна мова програмування. Це надає можливість керувати браузером, асинхронно отримувати дані з сервера та передавати їх. Також змінювати структуру веб-сторінки та її зовнішній вигляд. На сьогодні JavaScript є найпопулярніших мов програмування в інтернеті. Постійно створюють бібліотеки та фреймворки для роботи с JavaScript.

#### Переваги JavaScript

1. Не потрібен компілятор;
  2. Він може прив'язуватися до спеціальних елементів сторінки або подій, таких як клацання або наведення миші;
  3. Ви можете використовувати JavaScript для перевірки вхідних даних та зменшення необхідності ручної перевірки даних;
  4. Він швидший і легший за інші мови програмування.
- Працює в усіх браузерах.

### 3.8 Webpack - пакувальник модулів

Webpack – це система збирання. Це означає, що він збирає кілька файлів в один. В більш-менш великих програмах, необхідно поділяти код на кілька файлів з різних причин, і, відповідно, потрібно всіх їх підключити до сторінки. Це перше завдання, яке вирішує Webpack. Після того як Webpack збере проект, на виході отримаємо файл, який і потрібно підключити до сторінки.

## Які проблеми вирішує Webpack

1. Підтримка сучасного коду старими браузерми
2. Зменшення розмірів файлів
3. Підтримка модулів Javascript
4. оптимізація зображень
5. Використання препроцесорів
6. Організація коду. Можна розділяти код на кілька файлів не турбуючись про те, що це тільки ускладнить розробку. По-друге, налаштування, можна використовувати тільки ті плагіни і завантажувачі, в яких потребуєте.

Весь Webpack це його файл конфіга, тому якщо у є додатки зі схожими структурами, можна використовувати той же конфіг, досить просто перенести його в новий проект.

Оптимізація. Не потрібно самому видаляти, наприклад, коментарі з файлів для завантаження на сервер.

### 3.9 Менеджер пакетів NPM

Для того аби спеціальний модуль можна бул імпортувати на свій веб-сайт необхідно встановити менеджер пакетів npm.

NPM – це менеджер пакетів, який керує модулями і залежностями проекту У Node.js npm вже є за замовчуванням, тому його окрема установка не потрібно.

Всі пакети зберігаються в хмарному сховищі на офіційному сайті. Також можна створити свій модуль і опублікувати його в репозиторії для загального використання. Node.js npm також позбавляє розробника від необхідності

зберігати всі використовувані сторонні модулі в віддалених репозиторіях. За це відповідає файл `package.json`.

Модулі – це додатки, які реалізують деякий функціонал і можуть бути використані для розробки більш складних додатків

### **3.10 Архітектура програмного засобу**

Як і більшість веб-додатків, конструктор контенту має архітектуру клієнт-сервер. Серверна частина системи має доступ до бази даних, і виконує функції обробки і зберігання даних програми. Клієнтська частина системи – це набір програмних модулів, виконуваних на сторінці сайту. Клієнт і сервер взаємодіють один з одним.

Принцип роботи програми наступний: коли створили довільний блок з контентом та стилізували його, потрібно аби він відобразився на веб-сайті. Для цього за допомогою POST-запиту до сервера передається HTML та CSS у вигляді рядків, які є значеннями об'єкта `json` і він зберігається на сервері. Також коли сервер отримує дані генерується унікальний ключ(`id`).

Програмний модуль, який встановлено на сторінці сайту, кожні три секунди відправляє GET-запит до серверу і якщо на нього надійшли дані з конструктора, то за допомогою функцій HTML рядок перетворюється на DOM-елемент і додаються стилі.

Щоб веб-сайт отримав необхідний елемент потрібно імпортувати модуль і створити об'єкт класу.

Схема роботи системи показана на рис.4.4.





Рисунок 3.2 – Схема роботи програми

### 3.11 Алгоритм додавання контенту на довільний веб-сайт

Для прикладу було розроблено базову сторінку з панеллю, текстом та заголовком. Саме на цю сторінку буде імпортовано компонент.

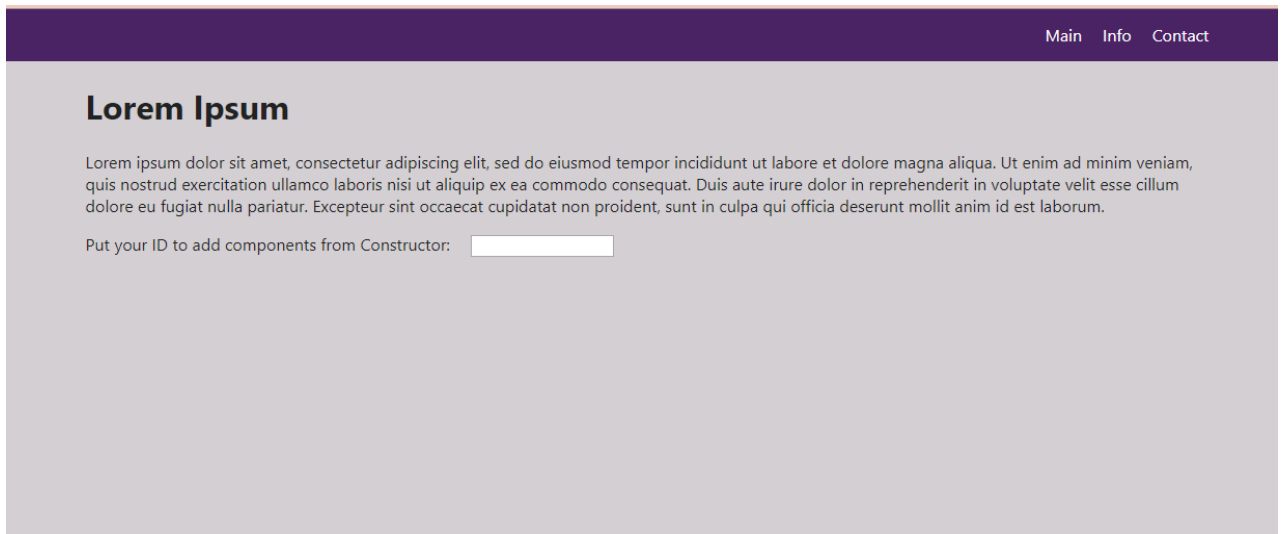


Рисунок 3.3 – Веб-сторінка на яку буде імпортовано компонент

Для того аби сайт міг бачити зміни, які відбуваються в конструкторі необхідно встановити модуль

```
import ConstructorJS from "constructorJS"
```

```
new ConstructorJS("API_KEY", "server_url").Init();
```

А також необхідно обрати елемент, який потрібно змінити або створити новий. В даному випадку було створено блок який буде змінюватися.

```
<div id={API_KEY}></div>
```

id - це унікальний набір символів, який буде отримано з серверу.

Після того як було встановлено необхідний модуль необхідно створити контент, який буде відображатися на сайті за допомогою конструктора.

За допомогою спеціальних інструментів такі як "2 Columns", "Text", "Button" та "Link" було створено шаблон.

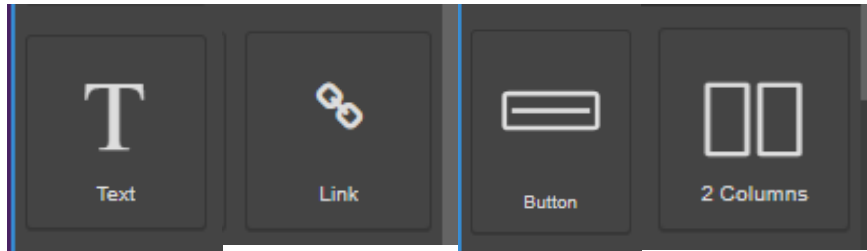


Рисунок 3.4, 3.5, 3.6, 3.7 – Інструменти, які було використано при створенні контенту

Після стилізації було отримано компонент, який готов до імпорту.

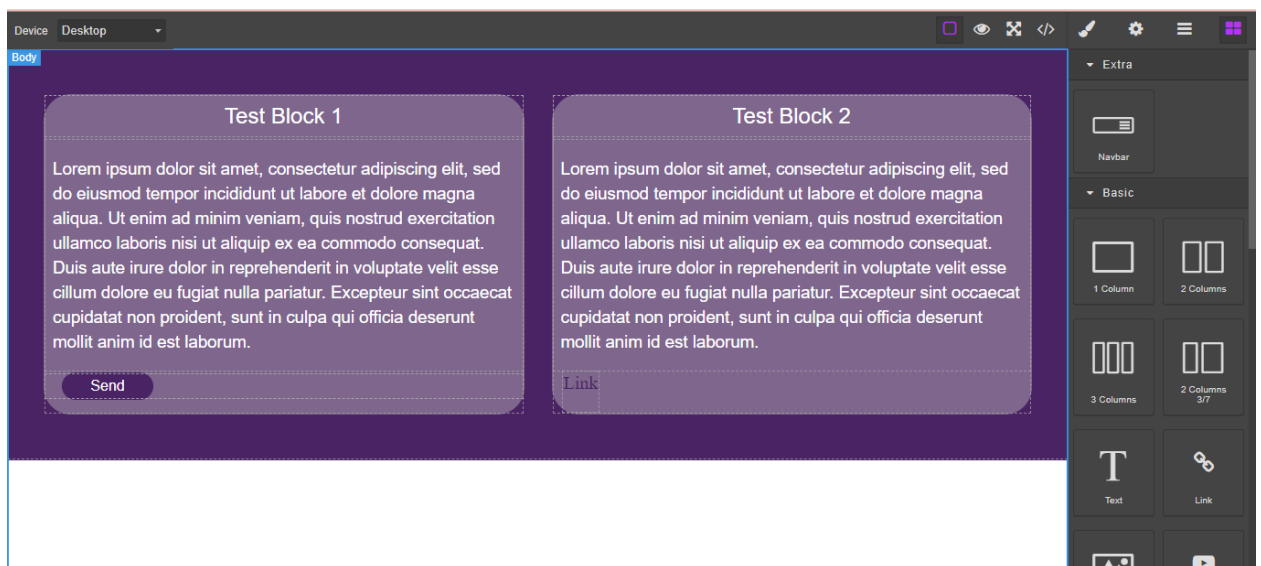


Рисунок 3.8 – Готовий компонент, вигляд на desktop

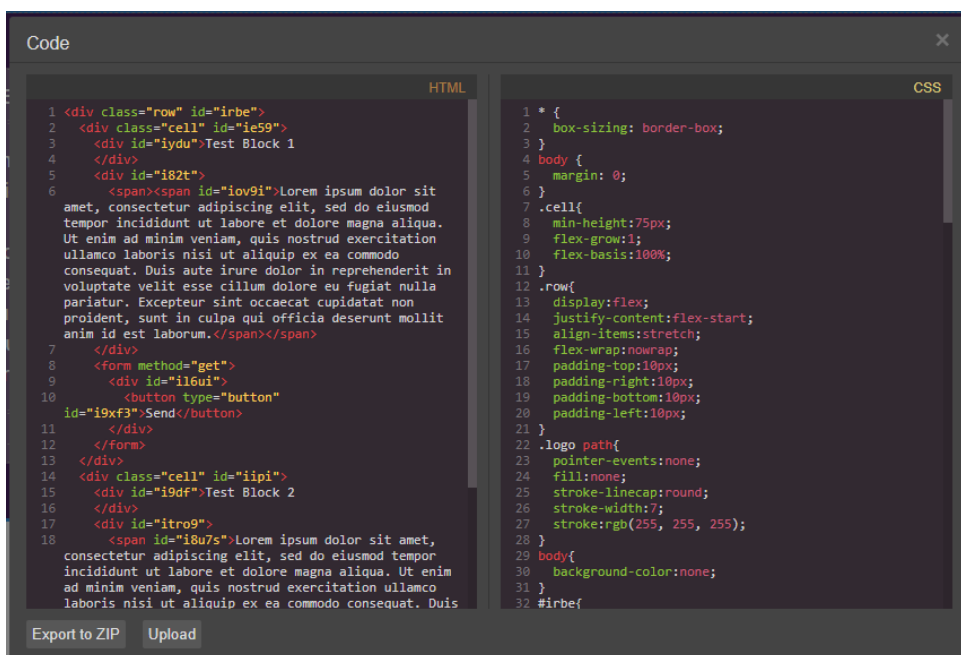


### Рисунок 3.9, 3.10– Мобільна версія контенту

З рисунку 3.9, 3.10 видно що даний контент є абсолютно адаптивний

Наступний крок це імпорт компонента на сайт

Необхідно скористатися інструментом "Перегляд коду" та натиснути на кнопку "Upload". Після чого з'явиться повідомлення з ключем, який необхідно копіювати.



The image shows a code editor window titled "Code" with two tabs: "HTML" and "CSS".

**HTML Code:**

```

1 <div class="row" id="irbe">
2   <div class="cell" id="ie59">
3     <div id="iydu">Test Block 1
4   </div>
5   <div id="i82t">
6     <span><span id="iov9i">Lorem ipsum dolor sit
7       amet, consectetur adipiscing elit, sed do eiusmod
8       tempor incididunt ut labore et dolore magna aliqua.
9       Ut enim ad minim veniam, quis nostrud exercitation
10      ullamco laboris nisi ut aliquip ex ea commodo
11      consequat. Duis aute irure dolor in reprehenderit in
12      voluptate velit esse cillum dolore eu fugiat nulla
13      pariatur. Excepteur sint occaecat cupidatat non
14      proident, sunt in culpa qui officia deserunt mollit
15      anim id est laborum.</span></span>
16   </div>
17   <form method="get">
18     <div id="i16ui">
19       <button type="button"
20         id="i9xf3">Send</button>
21     </div>
22   </form>
23 </div>
24 <div class="cell" id="iipi">
25   <div id="i9df">Test Block 2
26 </div>
27   <div id="itro9">
28     <span id="i8u7s">Lorem ipsum dolor sit amet,
29       consectetur adipiscing elit, sed do eiusmod tempor
30       incididunt ut labore et dolore magna aliqua. Ut enim
31       ad minim veniam, quis nostrud exercitation ullamco
32       laboris nisi ut aliquip ex ea commodo consequat. Duis

```

**CSS Code:**

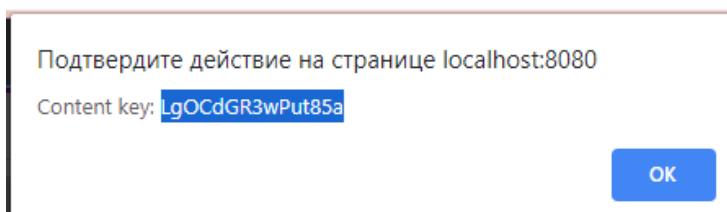
```

1 * {
2   box-sizing: border-box;
3 }
4 body {
5   margin: 0;
6 }
7 .cell{
8   min-height:75px;
9   flex-grow:1;
10  flex-basis:100%;
11 }
12 .row{
13   display:flex;
14   justify-content:flex-start;
15   align-items:stretch;
16   flex-wrap:nowrap;
17   padding-top:10px;
18   padding-right:10px;
19   padding-bottom:10px;
20   padding-left:10px;
21 }
22 .logo path{
23   pointer-events:none;
24   fill:none;
25   stroke-linecap:round;
26   stroke-width:7;
27   stroke:rgb(255, 255, 255);
28 }
29 body{
30   background-color:none;
31 }
32 #irbef

```

At the bottom of the editor, there are two buttons: "Export to ZIP" and "Upload".

### Рисунок 3.11 – Вікно "Перегляд коду"



### Рисунок 3.12 – Повідомлення зі спеціальним ключем

Необхідно скопіювати content key та вставити його в поле на сайті.  
Після чого наш контент одразу завантажеться на сайт

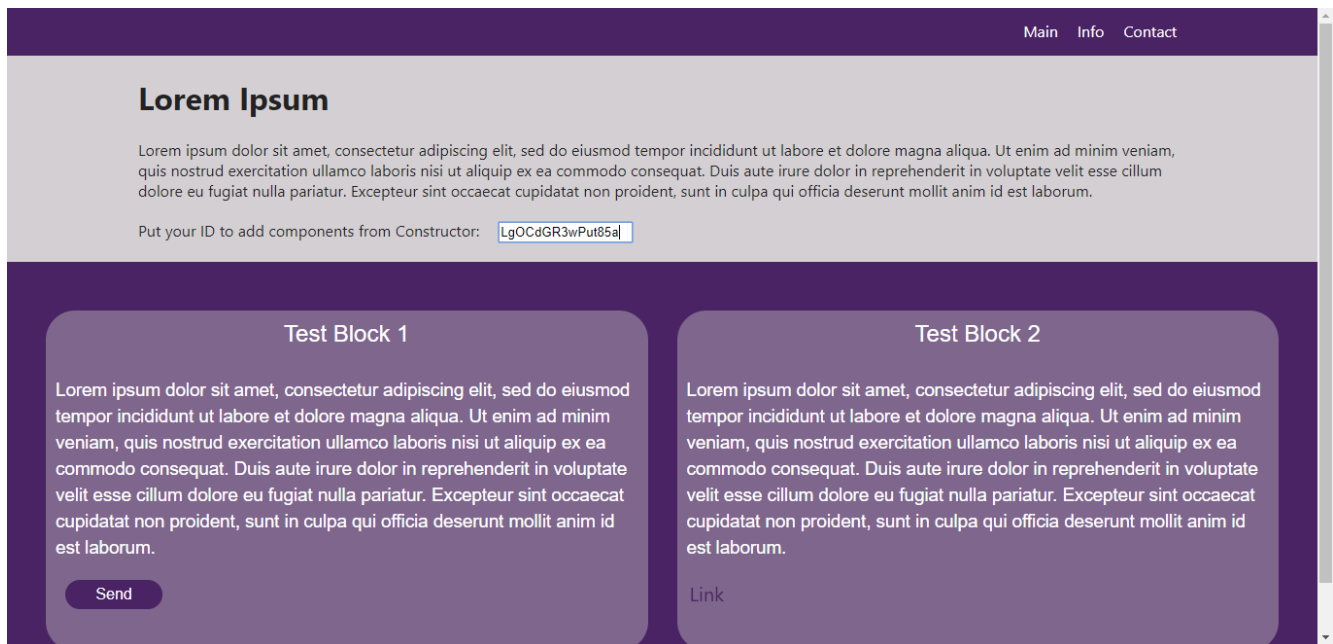


Рисунок 3.13 – Сайт з імпортованим компонентом

На даному прикладі було продемонстровано поетапну роботу з конструктором та базовою сторінкою. Компонент відображається правильно як на десктоп так і на мобільній версії.

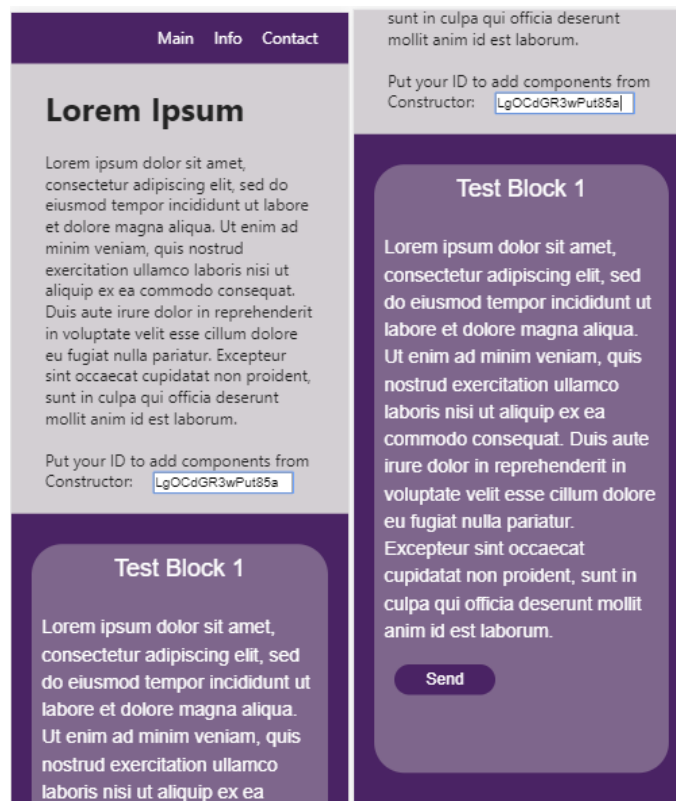


Рисунок 3.14, 3.15 – Мобільна версія веб-сторінки

### 3.12 Модульний візуальний конструктор і система «ПОЛІДАР»

Система «ПОЛІДАР» має чималий набір інструментів. Даний конструктор буде одним із функціоналів системи «ПОЛІДАР». Завдяки розробленому інструменту в системі «ПОЛІДАР» можна змінювати та вдосконалювати компоненти і додавати нові.



Рисунок 3.16 – Сторінка до імпорту компонента





## Макарівський архівний сектор

архівного відділу Бучанської райдержадміністрації  
Київської області



 
 **Online**
Архів України в Інтернеті
Мережа архівів Київщини

- [Історія архіву](#)
- [Наші новини](#)
- [Контакти](#)
- [Зворотній зв'язок](#)
- [Нормативна база](#)
- [Регуляторні акти](#)
- [Діяльність](#)
- [Послуги](#)
- [Фонди](#)
- [Пошук](#)
- [Штати](#)
- [Корисні посилання](#)
- [Статистика](#)
- [Карта сайту](#)

### Історія архіву



На території сучасного Макарівського району Київської області до Великої Вітчизняної війни були розташовані Макарівський та Бишівський райони Київської області. В Макарівському районному державному архіві (таку назву мав архів) до Великої Вітчизняної війни знаходилися на зберіганні документи 55 фондів за 1923-1936 роки в кількості 18284 справ, в Бишівському районному державному архіві – документи 25 фондів за 1929-1937 роки в кількості 5706 справ.

Станом на 01.01.2020 в архівному відділі Макарівської районної державної адміністрації Київської області зберігається 38372 справи за 1932-2018 роки 152 архівних фондів, з них: 38070 справ постійного зберігання, 302 – з кадрових питань (особового складу).

Станом на 01.01.2021 в архівному відділі Макарівської райдержадміністрації зберігаються 40165 од.зб. за 1932-2020 роки 151 архівного фонду, з них: 39780 справ постійного зберігання, 385 – з кадрових питань (особового складу). Найдавніший документ архівного відділу – Звіт колгоспу (сільськогосподарської артілі) «Воля» с.Мар'янка Брусилівського району (пізніше – Бишівського, Макарівського) Київської області за 1932-1933 роки.

#### Інформація про попередніх керівників архіву

Герасименко Анатолій Онуфрійович – завідуючий Макарівським районним державним архівом з 20 квітня 1951 року до 1956 року

Палієнко Юрій Максимович – завідуючий Макарівським районним державним архівом в 1956-1958 роках.

Ільченко Марія Дементівна – завідувача Макарівським районним державним архівом, архівного відділу Макарівської районної державної адміністрації, архівного відділу Макарівської районної Ради народних депутатів з червня 1958 року до 1 лютого 1996 року

Соломенко Таміла Володимирівна – завідувача архівним відділом Макарівської районної державної адміністрації з 31 січня 1996 року до 11 лютого 2000 року.

З 11 квітня 2000 року начальником архівного відділу Макарівської районної державної адміністрації працює Кравець Юлія Володимирівна, нагороджена відповідно до Указу Президента України від 21.12.2009 №1078/2009 Орденем княгині Ольги III ступеня, в 2003 році – Почесною грамотою Київської обласної державної адміністрації, в 2006 році – Грамотою Макарівської районної державної адміністрації, в 2012 році – Грамотою Президії Київської обласної організації профспілки працівників державних установ, в 2014 році – Грамотою Макарівської районної державної адміністрації та профспілкового комітету Макарівської районної державної адміністрації і Макарівської районної ради, в 2017 році – Грамотою Макарівської районної державної адміністрації з нагоди Дня Державної служби і Грамотою Макарівської районної державної адміністрації та Макарівської районної ради з нагоди Дня працівників архівних установ [5].

Сторінки, на які є посилання в тексті:

1. Герасименко Анатолій Онуфрійович (Переглянути)
2. Палієнко Юрій Максимович (Переглянути)
3. Ільченко Марія Дементівна (Переглянути)
4. Соломенко Таміла Володимирівна (Переглянути)
5. Грамоти Кравець Ю.В. (Переглянути)
6. Грамоти Зінченко Л.М. (Переглянути)

Copyright © 2012-2021 StarlineTelecom. Всі права захищені.

Рисунок 3.17 – Сторінка після імпорту компонента

### 3.13 Файлова структура системи

Всі файли програми розміщуються в папці src окрім index.html. Проект містить кілька підкаталогів:

1. src/test тести для компілювання конструктора;
2. src/style/scss зберігає всі файли зі стилями;
3. src/style/fonts зберігає шрифти;

В табл.3.1. описані найголовніші файли та їх призначення.

Таблиця 3.1 – Опис файлової структури програми

№ п/п	Назва файлу	Призначення файлу
1	index.html	Верстка
2	src/asset_manager/index.js	Додавання картинок
3	src/block_manager/index.js	Створення блоків
4	src/commands/index.js	Модуль з функціями
5	src/traits_manager/index.js	Зміна значень атрибутів
6	src/style_manager/index.js	Формує стилі
7	src/storage_manager/index.js	Збереження даних
8	src/plugin_manager/index.js	Створення плагінів
9	src/component_manager/index.js	Менеджер компонентів
10	src/device_manager/index.js	Зміна девайсу
11	src/code_manager/index.js	Перегляд коду
12	src/utils/colorPicker.js	ColorPicker для зміни кольору компонента
13	src/style/scss/main.scss	Стилі конструктора
14	script/build-locale.js	Створює locale/index.js файл



## **Висновки**

Для модульного візуального конструктора побудована інфологічна та даталогічна модель, яка повністю задовольняє вимогам вітчизняних ДСТУ[18] та вимогам загального міжнародного стандарту ISAD[19].

Застосовано програмне забезпечення для створення системи – Node.js для написання сервера та JavaScript для написання модуля.

На основі прикладу можна зробити висновок, що дана система є досить легкою та зрозумілою у використанні для користувачів.

## ЗАГАЛЬНІ ВИСНОВКИ

1. Проведений аналіз існуючих конструкторів сайтів, як кращих безкоштовних GrapesJs, uCoz, Ucraft, так і кращих платних Tilda, Wix і uKit, а також аналіз візуальних редакторів такі як CKeditor, TinyMCE, Draft.js показав, що не підійшов жодний із візуальних редакторів, оскільки вони мають ряд суттєвих недоліків і тому використання редактора є неприйнятним. Для розробки системи підійшов конструктор GrapesJs з відкритим кодом, в якому є все необхідне для реалізації адаптивного відображення контенту.

2. На базі GrapesJs було розроблено API та створений модуль для динамічного відображення контенту на будь-якому сайті. А також встановлено в функціонали системи «ПОЛІДАР» , що дає можливість змінювати її контент та додавати нові компоненти.

3. Побудовані для модульного візуального конструктора інфологічна та даталогічна моделі повністю задовольняють вимогам вітчизняних ДСТУ та вимогам загального міжнародного стандарту ISAD.

4. Показано, що вибрана мова програмування JavaScript для модуля та Node.js для написання сервера відповідає вимогам, які висуваються до конструкторів, і можуть бути використані для створення модульного візуального конструктора.

5. Розроблений інструмент є досить легким та зрозумілим у використанні для користувачів та розробників.

6. Створена система, містить всі необхідні функціонали та інструменти, які потрібні для створення адаптивних стилізованих блоків, а також для відображенні цих блоків на будь-якому сайті.

## ПЕРЕЛІК ПОСИЛАНЬ

1. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/HTML>.
2. CSS Документація і самоучителі [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/CSS>.
3. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/JavaScript>.
4. Лучшие бесплатные конструкторы для создания сайтов [Електронний ресурс] – Режим доступу до ресурсу: <https://konstruktorysajtov.com /luchshie-besplatnye-konstruktory-sajtov>.
5. 10 лучших WYSIWYG-редакторов текста и HTML для ваших проектов [Електронний ресурс]. – Режим доступу: <http://www.coolwebmasters.com/codes-and-scripts/691-10-best-wysiwyg-text-and-html-editors.html> [12 травня 2015, 09:58:47].
6. Сайт Tilda [Електронний ресурс] – Режим доступу до ресурсу: <https://tilda.cc/ru/>.
7. Сайт GrapesJs [Електронний ресурс] – Режим доступу до ресурсу: <https://grapesjs.com/>.
8. Сайт Ucraft [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ucraft.com/>
9. Сайт uKit [Електронний ресурс] – Режим доступу до ресурсу: <https://ukit.com/ru>.
10. Редактор TinyMCE [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tiny.cloud/>.
11. Редактор CKeditor [Електронний ресурс] – Режим доступу до ресурсу: <https://ckeditor.com/>.
12. Редактор Draft.js [Електронний ресурс] – Режим доступу до ресурсу: <https://draftjs.org/>.

13. Товкач І.О. Програмний модуль для забезпечення багатоканального інтерактивного зв'язку /І.О.Товкач, В.О.Піддубний // VII Міжнародних науково-технічних симпозіум «Новые технологии в телекоммуникациях». Карпати, Вишков, 21 — 25 січня 2013 р.: збірник тезисів конференції — Карпати, Вишков 2013. — С. 141—142.

14. Товкач І.О. Інтерактивний інструмент «ПОЛІДАР» для керування структурою та наповненням веб-сайту /І.О.Товкач, В.О.Піддубний // Информатика, математика, автоматика – 2013. Сумской государственной университет // – Режим доступу до матеріалів конференції: <http://elitconf.sumdu.edu.ua:8080/index.php/ima/ima13/paper/download/740/557>

15. Сайт Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/>.

16. Сайт Webpack [Електронний ресурс] – Режим доступу до ресурсу: <https://webpack.js.org/>.

17. Сайт Npm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/>.

18. Правила описування архівних документів (ISAD (G):1999, NEQ): ДСТУ 4331:2004. — [Чинний від 2005-07-01]. — К.: Держспоживстандарт України, 2004. — 20 с. — (Національний стандарт України).

19. ISAD (G):1999 General International Standart Archival Description: Adopted by the Committee on Descriptive Standards, Stockholm, Sweden, 19-22 September 1999.

## ДОДАТОК А

```

const express = require("express");
const bodyParser = require("body-parser");

const axios = require("axios");

const app = express();

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

const contentsArray = [];

const makeid = (length) => {
  var result = [];
  var characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';
  var charactersLength = characters.length;
  for ( var i = 0; i < length; i++ ) {
    result.push(characters.charAt(Math.floor(Math.random() *
charactersLength)));
  }
  return result.join('');
}

app.get('/allcontent', (request, response) => {
  response.setHeader('Access-Control-Allow-Origin', '*');
  response.send(JSON.stringify(contentsArray));
});

app.post('/edit', (request, response) => {
  const id = request.body.id;
  const htmlMin = request.body.htmlMin;
  const cssMin = response.body.cssMin;

  let status = false;
  contentsArray.forEach((block) => {
    if (block.id == id) {
      block.htmlMin = htmlMin;
      block.cssMin = cssMin;
      status = true;
    }
  });

  if (status) {
    response.send(
      {
        status: 200
      }
    );
  } else {
    response.send({
      status: "not_found"
    });
  }
});

```

```

    }
  });

  app.get('/content', (request, response) => {
    const id = request.query.id;
    let data = null;

    response.setHeader('Access-Control-Allow-Origin', '*');

    contentsArray.forEach((block) => {
      if (block.id == id) {
        data = block;
      }
    })

    if (data != null) {
      response.send(
        {
          status: 200,
          content: data
        }
      );
    } else {
      response.send({
        status: 404,
        content: data
      });
    }
  });

  app.post('/addcontent', (request, response) => {
    /**
     {
       "htmlMin": "<></>",
       "cssMin": ".clas...."
     }
    */
    const data = request.body;
    const ID = makeid(15);
    data.id = ID;

    contentsArray.push(data);
    console.log(contentsArray)

    response.send(
      {
        status: 200,
        id: ID
      }
    )
  });

```

```
app.listen("1488", () => {
  console.log("Server runing on port 1488...");
});

setTimeout(() => {
  axios.post('http://localhost:1488/addcontent', {
    htmlMin: "<div class=`test`></div>",
    css: ".test { color: `white` }"
  })
  .then((res) => {
    console.log("Response: ", res.data);
  });
}, 3000);
```