

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

САВЧЕНКО АНАТОЛІЙ АНДРІЙОВИЧ

Допускається до захисту:

завідувач кафедри
інформаційних технологій,
доктор технічних наук, доцент

_____ Т. В. Нескородева

« _____ » _____ 2022 р.

**ПРОГРАМНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ВЕРСТАННЯ ВЕБ-
САЙТУ З НАВЧАЛЬНИМ КОНТЕНТОМ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Штовба С. Д., професор
кафедри інформаційних
технологій, д.т.н., професор

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

АНОТАЦІЯ

Савченко А.А. Програмна система для автоматизації верстання веб-сайту з навчальним контентом. Спеціальність 122 «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

Метою кваліфікаційної роботи є розробка програмної системи для автоматизації верстання веб-сайту з навчальним контентом. Зараз популярність веб-ресурсів з навчальним контентом зростає. Але верстати однотипні сторінки часто і у великій кількості не є продуктивним і знижує інтерес до роботи у верстальника. Тому автоматизація цієї діяльності є актуальною. У вступі наведено актуальність автоматизації процесу верстки. У першому розділі наведені теоретичні та статистичні відомості, ключові переваги у відношенні ручної верстки. У другому розділі розглянуто технології та інструменти, які полегшили створення даної системи. У третьому розділі описано користувацький інтерфейс і програмну реалізацію системи, наведено інструкцію оператора та приклади роботи системи та окреслено перспективи розвитку і вдосконалення програмної системи.

Ключові слова: веб-сайт, програма, верстка, автоматизація, програмна система, веб-ресурс.

ABSTRACT

Savchenko A.A. A software system for layout automatization of website with educational content. Specialty 122 "Computer Science". Vasyl' Stus Donetsk National University, Vinnytsia, 2022.

The purpose of the qualification work is to develop a software system to automate the layout of a website with educational content. Nowadays, the popularity of web resources with educational content is growing. But machines of the same type of pages are often and in large quantities is not productive and reduces the interest in the work of the typesetter. Therefore, the automation of this activity is important. The introduction shows the relevance of automating the layout process. The first chapter presents theoretical and statistical information, key advantages in relation to manual typesetting. The second chapter discusses the technologies and tools that have facilitated the creation of this system. The third chapter describes the user interface and software implementation of the system, provides instructions to the operator and examples of system operation and outlines the prospects for development and improvement of the software system.

Keywords: website, program, layout, automation, software system, web resource.

ЗМІСТ

| | |
|--|----|
| ВСТУП | 5 |
| РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ | 7 |
| 1.1 Аналіз предметної області..... | 7 |
| 1.2 Огляд аналогів | 16 |
| 1.3 Деталізація завдань розробки | 26 |
| РОЗДІЛ 2 ОБҐРУНТУВАННЯ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ..... | 29 |
| 2.1 Архітектура системи..... | 29 |
| 2.2 Переваги Microsoft Visual Studio | 39 |
| 2.3 Технології..... | 41 |
| РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ..... | 46 |
| 3.1 Програмна реалізація AutoVerstka | 46 |
| 3.2 Програмна реалізація AutoVerstka2 | 49 |
| 3.3 Інструкція оператора | 52 |
| 3.4 Приклад роботи системи | 54 |
| 3.5 Перспективи..... | 57 |
| ВИСНОВКИ..... | 59 |
| СПИСОК ЛІТЕРАТУРИ..... | 61 |

ВСТУП

Сайти з навчальним контентом давно використовуються у навчальному процесі та стали звичним явищем. За інтернет ресурсами майбутнє системи освіти. Такі ресурси зазвичай складаються з великої кількості однотипних сторінок, наповнених текстом різного формату, ілюстраціями, таблицями, нумерованими та не нумерованими списками. Верстка цих та подібних елементів є монотонною і нудною, що знижує продуктивність, виснажує верстальника і найголовніше - потребує багато часу.

Метою роботи є створення програмної системи, яка автоматизує найбільш рутинні процедури верстання сайту з навчальним контентом. Завдання повної автоматизації верстання не ставиться, через нереальність її вирішення. Мова йде про часткове вирішення цієї проблеми. Часткове тому, що вона не є досконалою і її роботу мусить контролювати і коригувати спеціаліст верстальник. Проте цей інструмент допоможе значно зменшити час, що витрачається на роботу, а відповідно оптимізувати витрати людських ресурсів.

Актуальність даної системи є обумовлена тим, що вона допоможе значно підвищити продуктивність верстальника, що в свою чергу призведе до росту прибутку бізнесу, для якого використовується робота верстальника. Спеціаліст з верстки сайтів позбудеться ефекту «вигорання» на роботі, яка не приносить задоволення.

Для досягнення мети роботи поставлено та вирішено такі завдання:

- 1) аналіз об'єкту автоматизації;
- 2) аналіз конкурентних розробок;
- 3) розробка програмної системи;
- 4) тестування системи.

Об'єкт дослідження: автоматизація верстання веб-сайту.

Предмет дослідження: програмної система для автоматизація верстання веб-сайту з навчальним контентом.

Структура роботи: кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел. Робота містить 64 сторінки, 33 рисунки, 1 таблицю та список літератури з 25 джерел.



РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

На порозі ХХІ ст. організація ефективного управління робочими процесами у різних сферах життєдіяльності суспільства неможлива без використання сучасних інформаційно-комп'ютерних технологій. Особлива роль у цьому процесі належить глобальній мережі Інтернет, яка є інтерактивним засобом масової комунікації та формує віртуальну реальність, що є складовою діяльності інтернет-суспільства. Одним із невід'ємних елементів функціонування мережі Інтернет є веб-сайти, які забезпечують належний комунікаційний процес між усіма учасниками віртуального простору [5].

Перший веб-сайт створено Тімом Бернерсом-Лі 06 серпня 1991 р. Сьогодні поняття веб-сайту все більше використовується в різній документації, зокрема у нормативно-правових актах. Відповідно до наказу Державного комітету інформаційної політики, телебачення та радіомовлення України та Державного комітету зв'язку та інформатизації України «Про затвердження Порядку інформаційного наповнення та технічного забезпечення Єдиного веб-порталу органів виконавчої влади» від 25 листопада 2002 р. №237/225 **веб-сайт** визначено як сукупність програмних та апаратних засобів з унікальною адресою у мережі Інтернет разом з інформаційними ресурсами, що перебувають у розпорядженні певного суб'єкта і забезпечують доступ до юридичних та фізичних осіб до цих інформаційних ресурсів та інші інформаційні послуги через мережу Інтернет; **веб портал** – веб-сайт, організований як системне багаторівневе об'єднання різних ресурсів та сервісів для забезпечення максимальної можливості доступу до інформації та послуг [6].

Веб-сайт – сукупність веб сторінок та залежного вмісту, доступних в мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям [4].

М. В. Гура описує поняття «інтернет-сайт» як відокремлений, логічно завершений елемент мережі Інтернет, створений на основі технології посилань, що розташовані на сервері (інакше кажучи, його називають host), який має свою унікальну адресу (URL). За допомогою цієї унікальної адреси може отримати доступ будь-який користувач мережі Інтернет. Інтернет-сайт у своїй основі містить інтернет-сторінки, які мають графічний вигляд, що можуть бути переглянуті за допомогою браузерів [7].

Виділяють три типи структур веб-сайтів – лінійну, деревоподібну та довільну.

Подорожуючи сайтом із лінійною структурою, з головної сторінки ви перейдете на другу сторінку, з неї – на третю тощо. На сайті з деревоподібною структурою з головної сторінки можна потрапити на одну зі сторінок другого рівня, звідти – на одну зі сторінок третього рівня тощо. Сайт із довільною структурою видається зовсім неорганізованим, але саме в цьому й полягає принцип його створення. Подорожуючи таким сайтом, ви можете переходити з однієї його сторінки на інші в різні способи, і ваш шлях назад не обов'язково має бути таким самим. Вибір структури визначається особливостями завдань, що розв'язуються за допомогою веб-сайту [8].

Отже, під веб-сайтом можна розуміти сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. Веб-сайт існує у віртуальній формі і може нести програмний характер, призначенням якого є розміщення в мережі Інтернет будь-якої інформації тощо. Веб-сайт надає доступ як до фізичних, так і до юридичних осіб інформації (контент сайту), що розміщується на ньому [5].

Верстка дизайну сайту дозволяє зробити із зображення працюючий сайт, у якому всі графічні елементи відповідають закладеному дизайну. Але правильна верстка сайту має особливості, свої складності та специфіку.

Щоб зробити цей макет робочим інструментом, а не просто зображенням, необхідно написати код, який будуть розпізнавати браузері. Цей код відображається візуально так, як виглядає макет сайту.

Дизайн та верстка сайту – це два нерозлучні «брати», які завжди знаходяться разом. Перед дизайнером стоїть завдання розробити стильний макет. Завдання верстальника – втілити плани дизайнера в реальність та технічно, зробити сайт робочим [8].

У докомп'ютерну епоху верстальник був робітником друкарні, в обов'язки якого входила верстка сторінок. Верстальник наводив рядки тексту, виготовлені наборщиком, у сторінки та смуги майбутньої книги, komponував їх з ілюстраціями, додавав колонцифри, колонтитули, стежив за висячими рядками.

У дореволюційній друкарні старшого набирача називали метранпажем (від фр. *metteur en pages* - упорядник сторінок). Метранпаж призначав шрифт, довжину рядка і так далі набирачам, керував ними, а сам виконував найбільш відповідальні роботи. У радянську епоху його змінив технічний редактор видавництва, який робив макет та специфікацію для працівників друкарні.

Проте все частіше макет і верстку виконує одна людина, що суттєво підвищує вимоги до знань та вмінь верстальника у галузях книжкового (газетно-журнального) оформлення та друкарні. У деяких сучасних видавництвах такого універсального фахівця називають метранпажем, за аналогією з головним верстальником старовинної друкарні. У професійному середовищі немає єдиної думки щодо цього.

В англomовних країнах дизайнера, який відповідає нашому верстальнику, називають *layout artist*, *production artist* або *paste up artist*.

Традиційно термін використовується у видавничій справі, але з появою інтернету він набув нового значення – *верстальник веб-сторінок* [1].

Верстка веб-сторінки (англ. *page-proof*) — це процес створення веб-сторінки із попередньо створеного макету дизайну сайту, заздалегідь намальованого за допомогою графічних редакторів.

Існує 2 способи верстки - таблицна (котра була актуальна років 10 тому та блочна, котра будується на основі CSS) [2].

Верстка веб-сторінок відрізняється від поліграфічної тим, що необхідно враховувати різницю відображення елементів у різних браузерях та різницю у розмірах робочого простору пристроїв [3].

Роботи з верстки сайту можуть бути включені до етапу складання сайту.

Іншими словами, верстальник - це спеціаліст, який кодує сторінки web-сайтів, компонує текст з ілюстраціями. Він вміє писати HTML-код, знає, як графічно оформити сторінку.

HTML-верстальник - це професія, але останнім часом це вже не просто професія, а частина більш складної професії веб-розробник, причому поки ще є фронтендери і бекендери, хоча і це може дуже скоро піти в минуле.

HTML-верстальник це вже досить стара і набридла назва людини, який з макета створює веб-сайт, кодований в HTML і CSS, JS. Повний цикл створення повноцінних сайтів та Інтернет-порталів включає в себе наступні етапи:

- ідея або постановка цілей та завдань сайту;
- створення макету дизайну сайту;
- верстка. Створення frontend'a;
- програмування backend'a;
- базове заповнення контентом;
- розгортання на хостингу та інші[10].

По іншому це звучить так: Верстання веб-сайтів - це цілий комплекс процесів, ось лише деякі з них:

- вирізання з макету зображень, іконок та іншої графіки, компонування та збирання в окрему папку;
- збирання шрифтів для коректного відображення;
- розробка сторінок сайту, спираючись на дизайн-макет;
- верстка сторінок у HTML та CSS;
- підключення JS-бібліотек;

- створення динаміки елементів;
- тестування та перевірка валідності верстки.

Верстальник створює живий інструмент із зображення, як тато Карло з поліна створював Буратіно [9]. Це демонструє Рисунок 1.1, наскільки багато різноманітних інструментів потрібно використати, щоб створити повноцінний працюючий веб-сайт:



Рисунок 1.1 – Ілюстрація різноманітності інструментів, потрібних для верстання веб-сайтів [9].

Верстка дизайну сайту підпорядковується законам, які не допускають ніякої поблажливості. В іншому випадку код буде неправильним, а працездатність сайту буде знижена.

Ознаки правильної верстки:

- верстка обов'язково повинна бути кросбраузерною і коректно відображатися в різних браузерах, незалежно від роздільної здатності екрана;
- блочна верстка із застосуванням вважається найбільш якісною, при цьому вона має бути адаптивною до всіх пристроїв;
- валідна верстка сайту – у коді не повинно бути помилок;

- стилі виносяться в окремий файл, код сторінки максимально короткий та легкий;
- вміст HTML і CSS прописано лише малими літерами;
- CSS має переваги перед JS, все, що можна зробити за допомогою стилів, робиться лише через них;
- правильно структуровані заголовки, абзаци, продумані їх стилі;
- код зрозумілий, має чітку структуру, видно теги, що відкривають і закривають;
- усі зображення мають прописані розміри;
- збережено розміри початкового макету.

Лише при дотриманні цих правил сайт працюватиме правильно, швидко, а інші фахівці за необхідності зможуть розібратись в коді [9].

І якщо раніше можна було просто освоїти ази так званої верстки і почуватися дуже комфортно, то тепер ситуація дуже серйозно змінилася не на користь неробства. Сьогодні вже не можна займатися версткою без знання JS, а також без знання пари-трійки фреймворків JS.

Bootstrap сьогодні це просто must have, для будь-якого html-верстальника. Що таке Bootstrap? Bootstrap – це відкритий та безкоштовний HTML, CSS та JS фреймворк, який використовується веб-розробниками для швидкої верстки адаптивних дизайнів сайтів та веб-додатків [11].

Про верстки таблицями вже можна остаточно забути, ця навичка є застарілою і вже зовсім нікому не потрібна.

Якщо верстальник не знає, як верстати адаптивні веб-сайти, то він не зможе працювати у сучасному світі програмування. Рисунок 1.2 демонструє основні інструменти верстання сайтів:



Рисунок 1.2 – основні інструменти верстання сайтів [21].

Зараз відбувається небувалий ріст запитів на грамотних працюючих верстальників, особливо з переходом у період карантину на онлайн-навчання та віддалену роботу в багатьох сферах.

Що ж таке програми автоматичного верстання сайтів?

У Вікіпедії говориться, що верстальник іноді вдається до допомоги таких інструментів:

- WYSIWYG редакторів. Це слово є аббревіатурою від англ. What You See Is What You Get, «що бачиш, те й отримаєш») [16]. В них користувач розкладає всі елементи, які повинні були бути отримані за допомогою HTML, використовуючи графічний інтерфейс користувача. Після цього програма перетворює візуальне представлення в HTML-код. В даному випадку автору не обов'язково мати вичерпне знання HTML
- *програм автоматичної верстки сайтів*, які витягують шари із зображеннями та текстом з дизайн-макета, створеного в графічному редакторі Adobe Photoshop або йому подібному та формують із цих шарів HTML-код. Таким чином створюється базовий каркас веб-сторінки,

готовий до подальшого доопрацювання. Саме такі програми є предметом цієї бакалаврської роботи.

Використання WYSIWYG редакторів і програм автоматичної верстки часто засуджується[12][13] через низьку якість коду. Не дивлячись на це, найчастіше для користувачів використання цих програм зручніше за ручне кодування, а також не вимагає глибокого знання HTML, і тому широко використовуються [14].

Лінь рухає прогрес. Незважаючи на те, що «роботи» не можуть до кінця виключити ручну працю, деякі кроки в цьому напрямку постійно здійснюються. Світ веб-програмування в цьому плані не є винятком. Тому з'являються програми та послуги з *автоматичної верстки веб-сторінок*. Не дивлячись на те, що автоматичне верстання ще далеке за якістю від ручної верстки, воно постійно вдосконалюється. Безперечними конкурентними перевагами такої верстки є велика швидкість та невелика ціна.

Сервіс автоматичної верстки сайтів не претендує повністю замінити ручну верстку. Він зручний для сегменту недорогих сайтів, де головною перевагою є ціна та швидкість виготовлення. Подібних низькобюджетних сайтів в Інтернеті багато, і цей сегмент продовжує зростати. Саме такими сайтами є веб-сайти з навчальним контентом. Не виключено, що незабаром будь-який просунутий користувач зможе легко купувати шаблон сайту та автоматично верстати його під будь-яку систему CMS.

Окрім сегменту недорогих сайтів, сервіс автоматичної верстки допомагає у створенні також складних високобюджетних сайтів. Наприклад, можна виконати частину роботи: швидко зробити нарізку зображень із шарів PSD файлу, витягнути тексти з CSS3 стилями та зверстати каркас сайту, готовий до подальшого доопрацювання. Одним із користувачів було помічено, що сервіс зручний тим, що дозволяє робити проміжні варіанти майбутнього сайту, що дає можливість надати на розгляд ці варіанти замовнику.

Важливою вимогою сервісу автоматичної верстки для отримання якісного результату є правильна підготовка макета [15].

Під *освітнім веб-сайтом* розуміється сукупність веб-сторінок з повторюваним дизайном, що несуть у собі цілеспрямований процес навчання і виховання в інтересах особистості, суспільства, держави, об'єднаних за змістом, навігаційно і фізично перебувають на одному сервері, використання яких може супроводжуватись атестацією учнів.

До основних характеристик освітніх сайтів належать [17]:

- зміст освітнього Інтернет-сайту – єдність всіх основних елементів (текстових і графічних) освітньої інформації, яка існує і виражена у вигляді веб-сайтів, а також єдність зв'язків цих основних елементів;
- дизайн освітнього Інтернет-сайту – процес вибору й організації графічних компонентів з метою досягнення певної мети, яка може бути або естетичною, або мати функціональне підґрунтя, а часто переслідувати обидві ці цілі;
- технічна реалізація освітнього Інтернет-сайту – вибір компонентів, інтегрування технологій, програмних продуктів і технічних засобів для донесення освітньої інформації до користувача;
- експлуатація освітніх Інтернет-сайтів – характеристики, що інформують про використання освітніх сайтів користувачами.

Останнім часом стали з'являтися так звані освітні портали – великі за обсягом освітньої інформації сайти, в яких поєднані різноманітні функції і типи освітніх сайтів. Тут розміщуються результати проектної навчальної роботи школярів (у тому числі, веб-сторінки різного типу, створені ними), роздуми вчителів і координаторів мережевої роботи з приводу мережевої діяльності в школі, інтеграції її до освітньої програми, виступи фахівців про діяльність освітніх закладів у мережі Інтернет. Також представлена інформація про семінари, конференції, нові веб-сайти, педагогічні видання та медіа засоби, які можуть допомогти вчителям в освітній діяльності та багато іншого [18].

Всі ці особливості впливають на процес автоматизації верстання, а саме – потрібно реалізувати всі вимоги щодо доступності, естетичності, простоти і водночас функціональності даних веб-сайтів. Усі ці особливості навчальних

сайтів будуть враховані у практичній частині даної бакалаврської роботи такими засобами автоматичного верстання веб-сайтів: розпізнавання різних форматів тексту і застосування до них відповідних класів і стилів, формування нумерованих і не нумерованих списків, побудова таблиць, вставка зображень.

1.2 Огляд аналогів

В даний час існує велика кількість комп'ютерних програм, які спрощують працю верстальника. Їх можна розділити на 2 класи:

- різні текстові редактори та IDE з підсвічуванням коду та автодоповненням:
 - Visual Studio Code
 - WebStorm;
- CSS фреймворки (набори фрагментів верстки та коду для прискореної розробки макета сайту):
 - Bootstrap
 - Foundation та тощо.

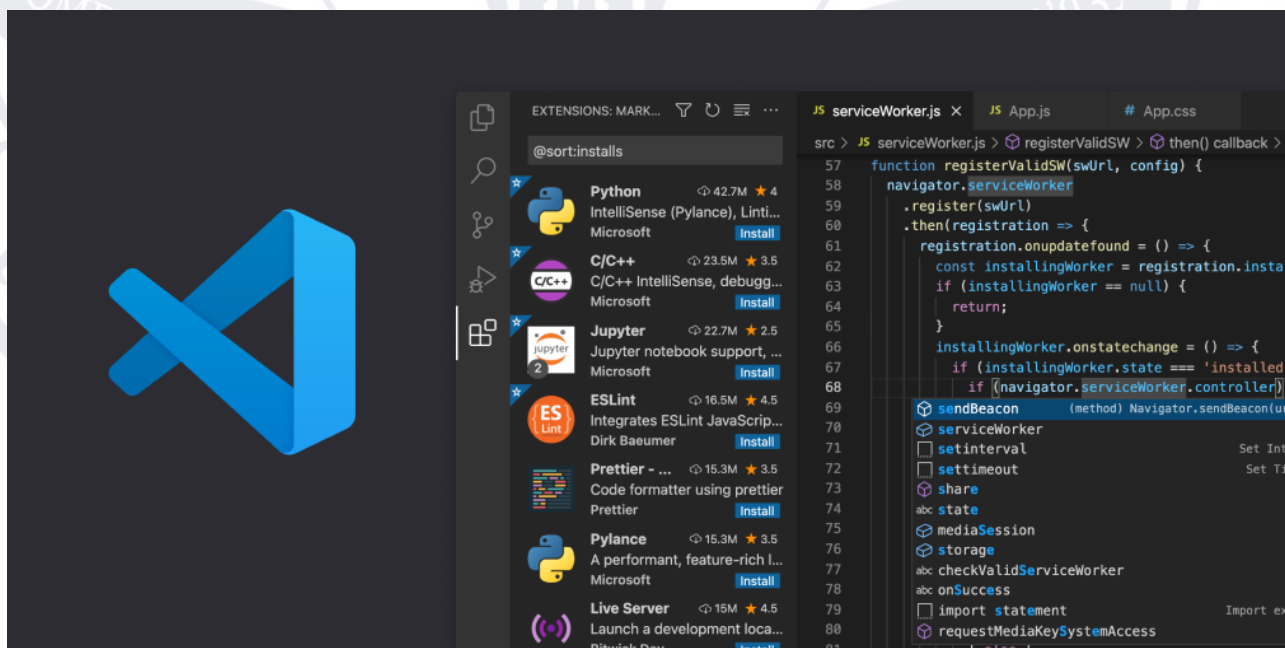


Рисунок 1.3 – Visual Studio Code.

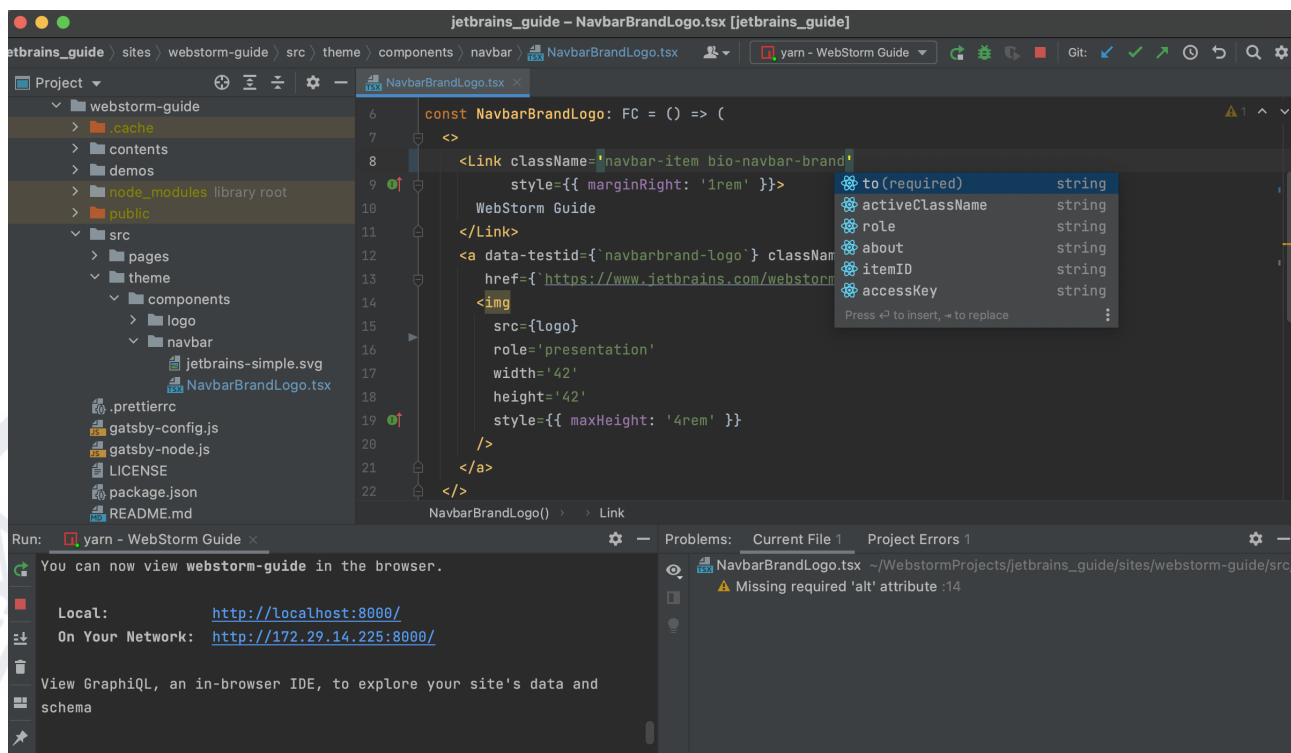


Рисунок 1.4 – WebStorm.

WebStorm – це інтегроване середовище розробки для JavaScript та пов'язаних із ним технологій. Як і інші IDE JetBrains, він робить ваш досвід розробки більш приємним, автоматизує рутинну роботу та допомагає з легкістю справлятися зі складними завданнями.



Рисунок 1.5 – Популярні інструменти верстальників.

SCSS

Офіційно описуваний як «CSS із надздібностями», SCSS (або Sass) пропонує спосіб писати стилі для веб-сайтів із покращеним синтаксисом CSS. Загалом, браузер не знає, як обробляти функції SCSS, такі як функції, міксини та вкладення. Нам потрібно конвертувати їх у звичайні файли CSS, щоб запустити їх у браузері.

Також аналогами програм автоматизації написання сайтів є **візуальні редактори**, наприклад, Adobe Dreamweaver. За їх допомогою можна робити веб-сайти без написання коду, в графічному режимі, тобто користувач має в своєму розпорядженні візуальні блоки, а код генерується автоматично. Але професійний HTML-верстальник цими програмами не користується. Він повинен вміти використовувати мови HTML і CSS без допомоги візуальних редакторів, щоб забезпечити максимальну коректність коду мінімальної ваги.

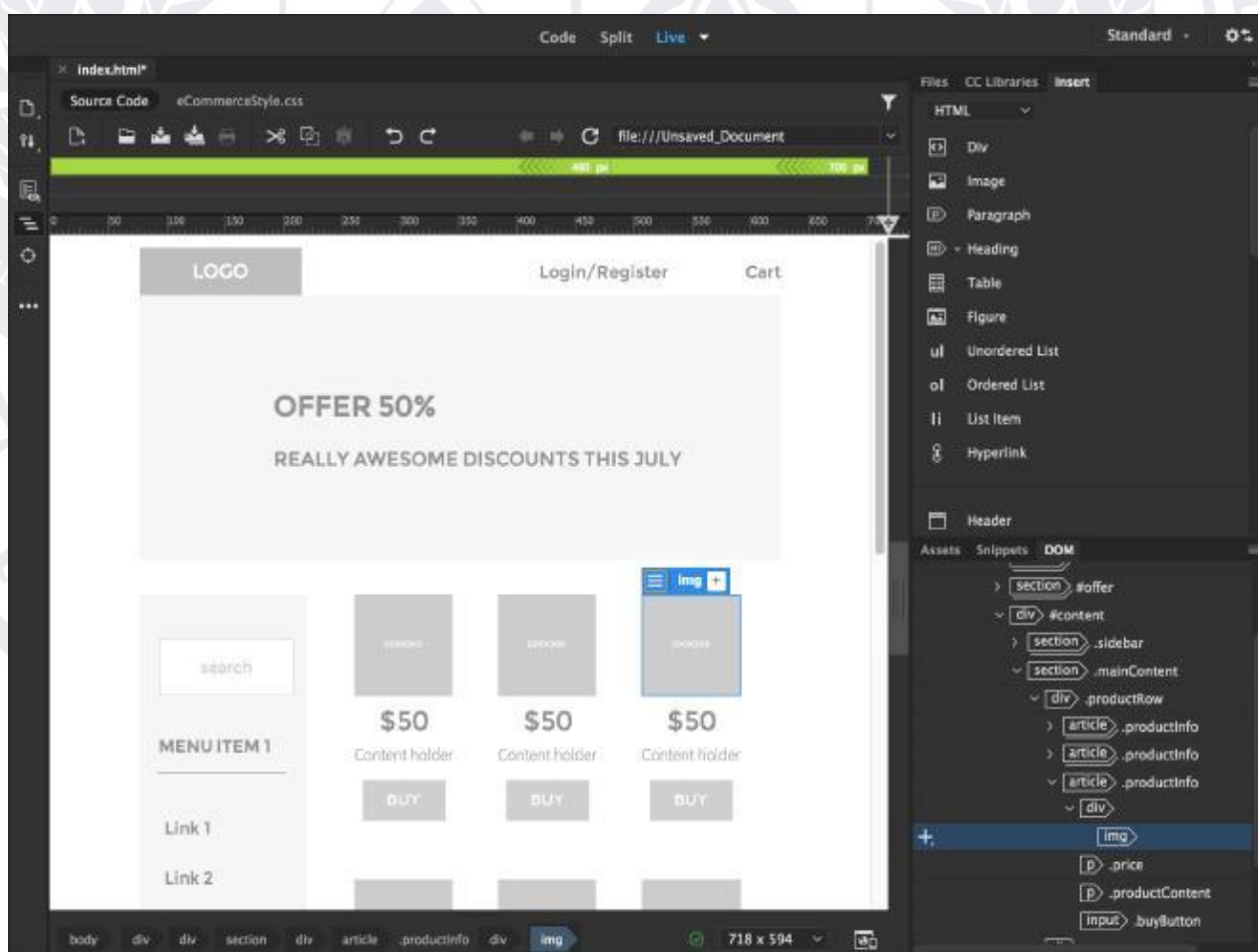


Рисунок 1.6 – Adobe Dreamweaver.

Редактори коду та плагіни до них

Без редактора коду не зверстати сайт. Це основний інструмент верстальника. Він дозволяє писати код десяткою мов програмування, підсвічувати синтаксис, вирівнювати код, тестувати його (іноді у редакторі), заливає в репозиторій github. Який із редакторів вибрати – справа смаку. Найпопулярніші: VSCode, Sublime Text, Brackets, Atom. Всі сучасні редактори коду мають приблизно однаковий функціонал, дозволяють працювати з Git і встановлювати доповнення. А ось плагіни до редакторів можуть набагато полегшити життя верстальщику.

Emmet

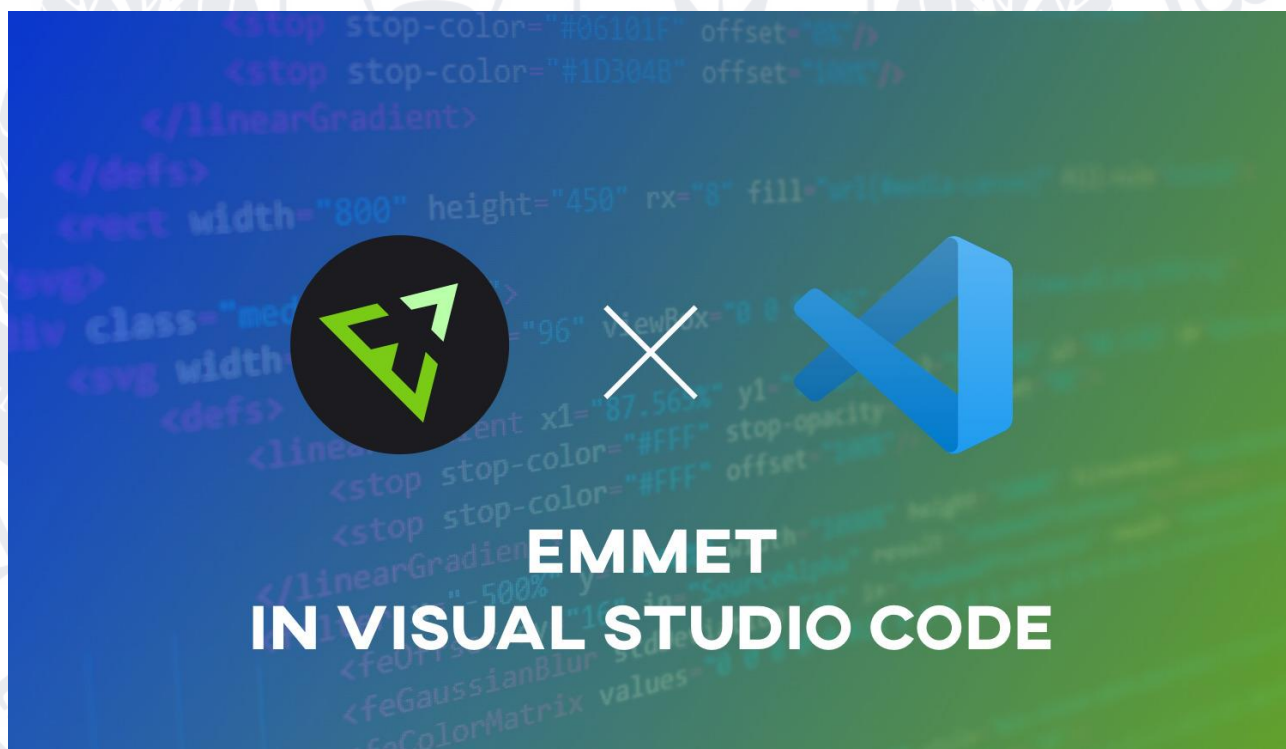


Рисунок 1.7 – Програма Emmet у зв'язці з VSCode[22]

Працює практично з будь-яким редактором коду. У VSCode, наприклад, його навіть не треба спеціально встановлювати: він уже вбудований у його функціонал. Плагін може значно скоротити розробнику кількість написаного тексту. Emmet перетворює короткі аббревіатури на шматки коду HTML. З ним можна швидко побудувати структуру сторінки, написати список з безлічі

елементів, таблицю, кілька вкладених один в одного блоків разом із класами. І виключає можливість втрачених дужок, що закривають. Всі теги, що закривають, з'являються автоматично.

Також Еммет може писати префікси для кросбраузерної верстки і навіть згенерувати «Lorem ipsum» потрібної довжини.

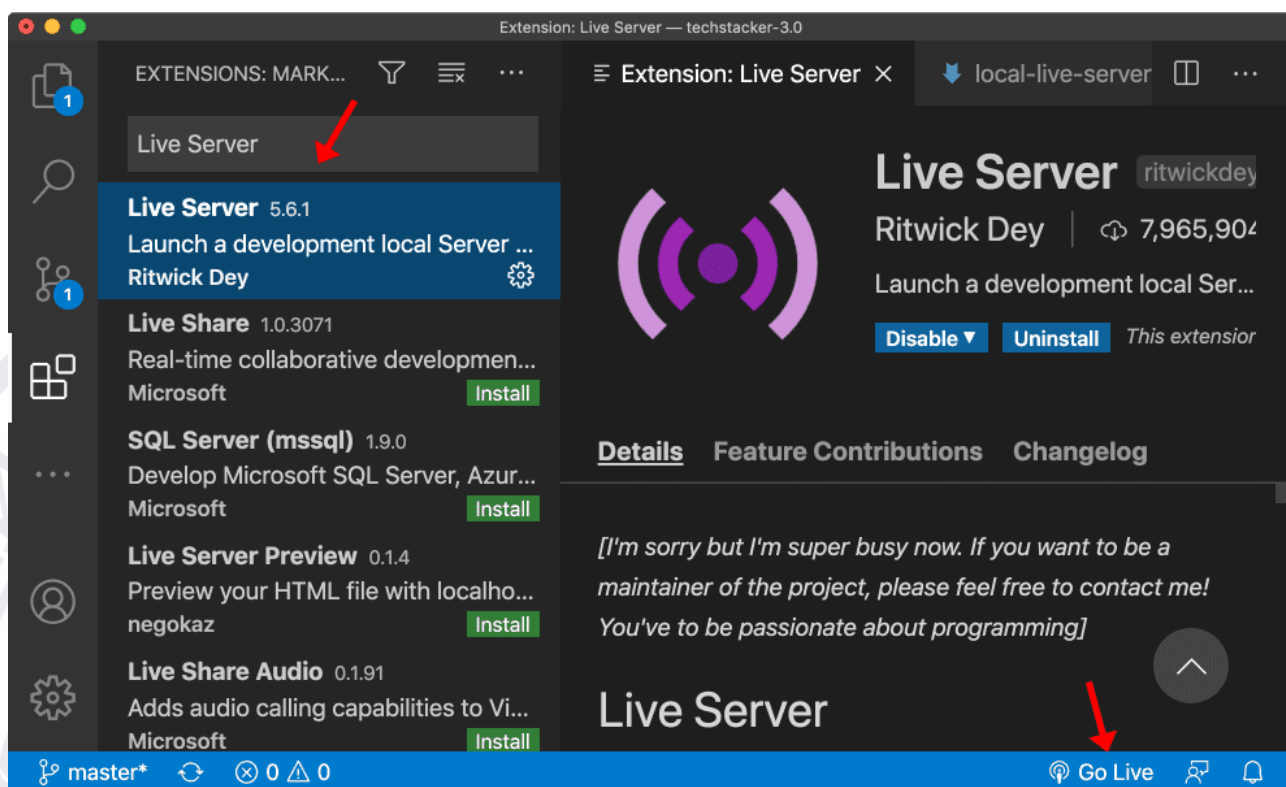
Prettier



Рисунок 1.8 – Логотип програми Prettier

Акуратний і добре читаний код - ознака майстерності, але вирівнювати його марудно і задовго, особливо якщо проект налічує вже не одну тисячу рядків коду. На допомогу придуть плагіни для вирівнювання коду. Один з них – Prettier. Потрібно лише виділити шматок коду і натиснути на комбінацію клавіш, і вуаля — ваш код акуратно побудований відповідно до шаблону. Звичайно, всі правила вирівнювання можна за бажання налаштувати.

Live Server



1.9 – логотип програми Live Server

Плагін дозволяє не перезавантажувати сторінку браузера щоразу, як ви внесли зміни до коду html. Він буде стежити за всіма змінами у ваших файлах і миттєво показувати результат у браузері. Дуже зручно, коли треба внести незначні зміни у стилях або навпаки, перебудувати структуру блоків та прийняти рішення, чи варто залишати зміни.

Браузерні доповнення

Браузер - ще одна річ, без якої робота верстальника немислима. Звичайно, у вас має бути кілька браузерів для перевірки кросбраузерності верстки. Найзручніше користуватися Google Chrome або Mozilla Firefox: по-перше, це найпопулярніші браузери, і по-друге, до них існує безліч цікавих доповнень, у тому числі для розробників. Одним з них є **CSS3 Generator**



Рисунок 1.10 – Ілюстрація доданку CSS3 Generator

Інструмент для генерації CSS-коду для деяких невеликих, але трудомістких завдань. CSS3 Generator може трансформувати колір з одного формату на інший, створити css-код для тіні, градієнта, фільтра картинки. З його допомогою можна написати код трансформації блоку та зробити просту анімацію.

Ще одним з таких доданків є **ColorZilla**

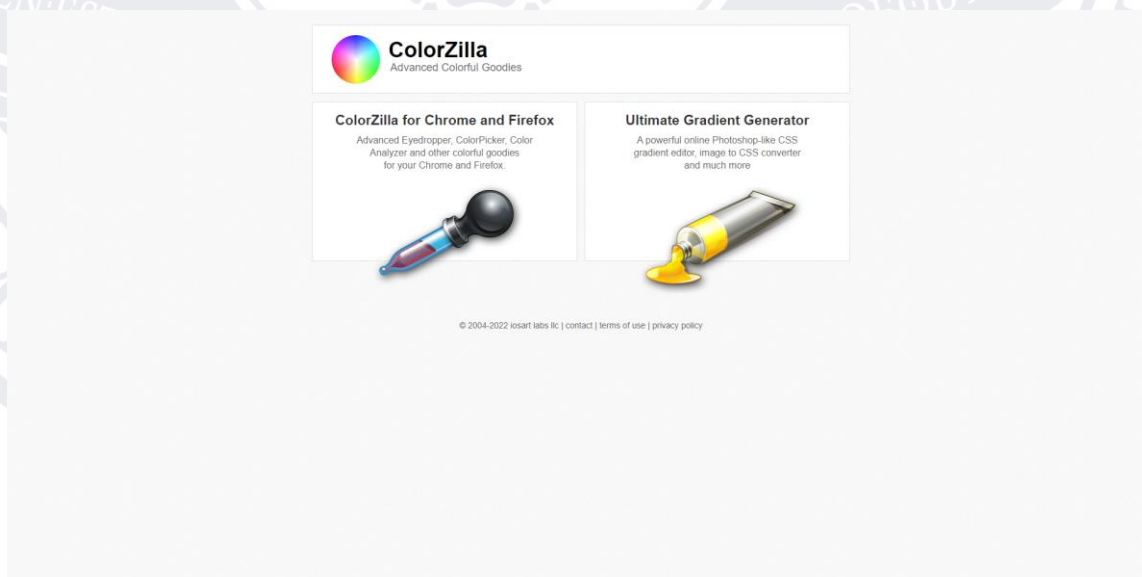


Рисунок 1.11 – Ілюстрація доданку ColorZilla

Інша його назва - Колірна піпетка. Доданок дозволяє дізнатися колірний код вибраної ділянки будь-якого сайту. Код можна отримати в будь-якому з колірних форматів, які використовуються в розробці сайтів, зберегти його в історії та повернутися до нього ще раз у будь-який момент.

Ще одна корисна функція доданку – доступ в один клік на сторінку генератора градієнтів від Колорзилли. Цей сайт дозволяє швидко та легко створити кросбраузерний код для градієнта будь-якої складності.

Онлайн-інструменти

Розробники вигадали багато інструментів, доступних онлайн. Деякі з них незамінні для роботи, інші полегшують роботу, деякі є просто для розваги або натхнення. Розглянемо кілька з них:

Google fonts

<https://fonts.google.com/>

Найкращий безкоштовний каталог шрифтів, налічує близько 100 шрифтів різними мовами. В інтерактивному каталозі можна детально вивчити символи шрифту, оцінити, як він виглядатиме в різних розмірах та на кресленнях. Вибраний шрифт можна завантажити та підключити локально, а можна згенерувати код для підключення його з сервера Гугла.

Font Awesome

<https://fontawesome.ru/>

Незамінний інструмент, якщо до вашого сайту потрібно додати іконки.

Font awesome - колекція векторних іконок, які можна вставити на сайт, як шрифт. Векторні іконки можна масштабувати до будь-якого розміру. І значить, вони будуть чудово виглядати і на retina-дисплеях, що вимагають високої роздільної здатності графіки. Їх легко підключити та легко налаштувати:

розмір, колір, тінь – всі властивості можна прописати в CSS, як для звичайного шрифту.

Пісочниці коду

<https://codepen.io/>

<https://jsfiddle.net/>

<https://repl.it/>

Пісочниці – це майданчики для тестування та демонстрації фрагментів коду HTML, CSS та JavaScript. Інші мови програмування вони теж підтримують, як і деякі бібліотеки та фреймворки, але цей функціонал не є темою цієї роботи.

У пісочниці можна писати або редагувати код і бачити зміни в режимі реального часу. Можна поділитися посиланням на роботу, а інший розробник побачить одночасно і код, і результат. Також у пісочниці можна працювати над чужим кодом, форкнувши його до себе. Codepen, крім роботи над власним кодом, дозволяє набратися ідей та натхнення і навіть взяти участь у колективних челенджах. Проекти кодпен публічні, і в інтерфейсі можна бачити роботи інших користувачів та шукати цікаві піни за ключовими словами.

Перевірка якості коду

Важливий етап роботи — перевірка роботи написаного вами коду. Необхідно перевірити, як відкривається сайт у різних браузерах та за різної ширини екрана. Чи він відповідає дизайнерському макету. Чи його інтерактивні елементи: кнопки, форма зв'язку, посилання працюють як задумано. Для автоматизації цього етапу є кілька популярних інструментів. Ось деякі з них.

Perfect Pixel

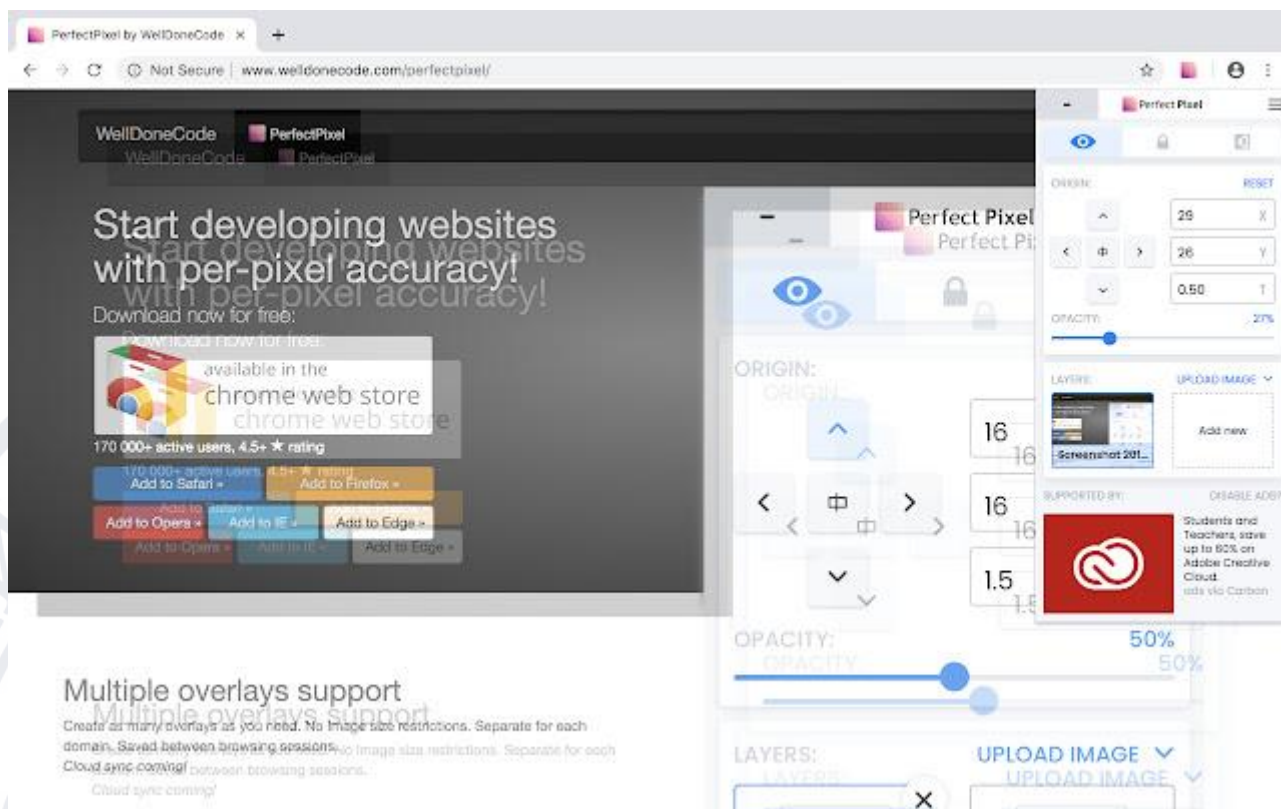


Рисунок 1.12 – Приклад роботи Perfect Pixel[23]

Популярний інструмент для перевірки якості верстки та відповідності її макету. Існує як додаток для будь-якого браузера.

Perfect Pixel дозволяє додати на сторінку jpg з макетом окремим напівпрозорим шаром. Таким чином, можна порівняти точність вашої верстки та її відповідність задуму дизайнера.

Валідатор

<https://validator.w3.org/>

Перевіряє розмітку html та css на наявність помилок та відповідність стандартам Консорціуму World Wide Web. Цей сервіс не напише за вас хороший і правильний код, але вкаже на місця, де можливі проблеми при роботі коду в реальності.

У валідаторі можна перевірити код за посиланням, можна завантажити файл повністю або фрагмент коду. В результаті ви отримаєте список помилок і попереджень. Помилки можуть зашкодити роботі сайту, а попередження

показують код, який не відповідає стандартам, з надмірними або безглуздими елементами. Ділянки коду із попередженнями можуть зашкодити становищу сайту в пошукових системах.

Тепер розглянемо зведену таблицю з порівняння ключових аналогів, програмних продуктів для автоматизації верстки навчальних сайтів:

Таблиця 1.1 – Порівняння ключових аналогів

| Критерії | Emmet | Bootstrap | SCSS |
|---|-------|-----------|------|
| Безкоштовна | + | + | + |
| Можливість генерувати HTML код за попередньо заданими класами | + | - | - |
| Попередньо написаний CSS код | - | + | - |
| Налаштування під стилі конкретного сайту | - | - | + |

На основі даної таблиці можна зробити висновок, що кожен з розглянутих аналогів має свої переваги та недоліки, але жоден з них не задовольняє повністю вимог поставлених у даній роботі. Детальніше про ці вимоги піде мова у наступному підрозділі.

1.3 Деталізація завдань розробки

Завданням даної бакалаврської роботи є реалізація системи автоматизації верстання веб-сайту з навчальним контентом. Подібні веб-ресурси вирізняються з поміж інших тим, що мають великий обсяг матеріалу з однотипною структурою, це буде враховано у програмному продукті наступним чином: завчасно прописані класи, функції розпізнавання та

перетворення для більшої частини типових елементів. Додаток повинен виконувати наступні функції:

1. Читання з документу, або текстового поля. За це відповідатиме функція `Open`, яка буде викликатися відповідною кнопкою, та в свою чергу викликати діалог вибору файлу. Також, текстове поле, що розміщено на формі буде здатне приймати RTF, тобто його можна буде просто вставляти туди з буфера обміну.
2. Перетворення змісту в такі елементи:
 - a. Нумеровані списки. Визначивши уривок тексту, як нумерований список, загорнути кожний його рядок у відповідний тег, якщо присутня нумерація – стерти.
 - b. Ненумеровані списки. Визначивши уривок тексту, як ненумерований список, загорнути його у відповідний тег, кожен рядок розділити тегами нового рядка, додати символ «•» у початок кожного рядка.
 - c. Заголовки. Визначивши уривок тексту, як заголовок, загорнути його у відповідний тег.
 - d. Жирний текст. Визначивши уривок тексту, як жирний текст, загорнути його у відповідний тег.
 - e. Таблиці. Визначивши уривок тексту, як таблицю, загорнути його, та кожен його елемент у відповідний тег.
 - f. Відцентрований текст. Визначивши уривок тексту, як відцентрований текст, загорнути його у відповідний тег.
 - g. Зображення. Визначивши зображення, замінити його на тег, що легко піддається подальшій підстановці посилання з файлового серверу.
3. Додавання відповідних класів до елементів. Кожному доданому тегові автоматично присвоюється один з попередньо заданих класів, для зв'язку зі стилями.

4. Вивід тексту в поле, копіювання в буфер обміну або запис у документ. За це відповідатиме функція Save, яка буде викликатися відповідною кнопкою, та в свою чергу зберігатиме файл з відповідною назвою, яку також можна вибрати зі списку на формі. Також, текст буде виводитись у текстове поле, де можна буде його скопіювати в буфер обміну.

Отже потрібні кнопки та текстові поля. Також платформа має бути зручною для користування, легкою для розуміння і доступною для більшості. Про обрані для реалізації інструменти піде мова в наступному розділі.

Висновок до розділу 1

У цьому розділі було розглянуто актуальність верстки, потребу у верстальниках та інструментах, що полегшують їм роботу. Також було описано необхідні функції програми автоматизації. Розглянуто альтернативи програм автоматизації на даний момент, обговорено їхні особливості, а також переваги і недоліки.

РОЗДІЛ 2 ОБГРУНТУВАННЯ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ

2.1 Архітектура системи

Для доступності і простоти написання програми автоматизації було обрано наступні засоби:

- мова програмування C#,
- IDE Microsoft Visual Studio,
- інтерфейс програмування додатків Windows Forms,
- формат файлів RTF,
- бібліотеки RtfPipe та HtmlAgilityPack.

Про кожен з них окремо піде мова у наступному підрозділі.

Розробку поділено на дві частини, окремі програми. Назвемо їх: AutoVerstka та AutoVerstka2. Перша програма, під назвою AutoVerstka, призначена для обробки окремих елементів з більшою точністю, натомість друга, під назвою AutoVerstka2, здатна конвертувати усю сторінку цілком. Детальніше про кожну з них піде мова у третьому розділі.

2.2 Інструменти та утиліти для пришвидшення роботи

Програмування передбачає застосування різних інструментів та навичок. Ключовими є мови. Серед багатого вибору та різноманітності таких особливої уваги гідний C#.

C# – об'єктно-орієнтована мова програмування. Розроблений у 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберга та Скотта Вільтаумота [20] як мова розробки додатків для платформи Microsoft .NET Framework. По суті, він об'єднав у собі потужність Java та простоту C++. Ця мова використовується для створення програм на платформі Microsoft.NET Framework. Розроблявся насамперед як інструмент прикладного рівня CLR. Тому C# багато в чому залежить від її можливостей.

Попередниками C# є такі мови програмування, як: Pascal Smalltalk; Модула; C++ Java. Від останніх двох було перейнято особливо багато. При

цьому мова програмування сі шарп не включає ті моделі, які на практиці застосування C++ і Java зарекомендували себе, як дуже проблемні. Так, C# допускає множинну реалізацію інтерфейсів, але не підтримує множинного успадкування. Ключовою перевагою даної мови є те, що вона цілком доступна і зрозуміла тим, хто вже знайомий з C-синтаксисом.

Використання саме мови C# обумовлено тим, що програмна система має бути доступною на найпопулярнішій операційній системі серед користувачів (верстальників) Microsoft Windows OS. Також, дуже легкий для освоєння синтаксис і наявність великої кількості інтегрованих бібліотек робить цю мову програмування найкращим вибором для розробки даної програмної системи.

Microsoft Visual Studio

Visual Studio IDE (інтегроване середовище розробки) — це програмний засіб для розробників. Його інтерфейс користувача використовується для розробки програмного забезпечення для редагування, налагодження та створення коду. Visual Studio містить редактор коду, який підтримує IntelliSense (компонент завершення коду), а також рефакторинг коду. Інтегрований налагоджувач працює і як налагоджувач на рівні джерела, і як налагоджувач на рівні машини. Інші вбудовані інструменти включають профайлер коду, конструктор для створення додатків із графічним інтерфейсом користувача, веб-дизайнер, конструктор класів і дизайнер схем баз даних.

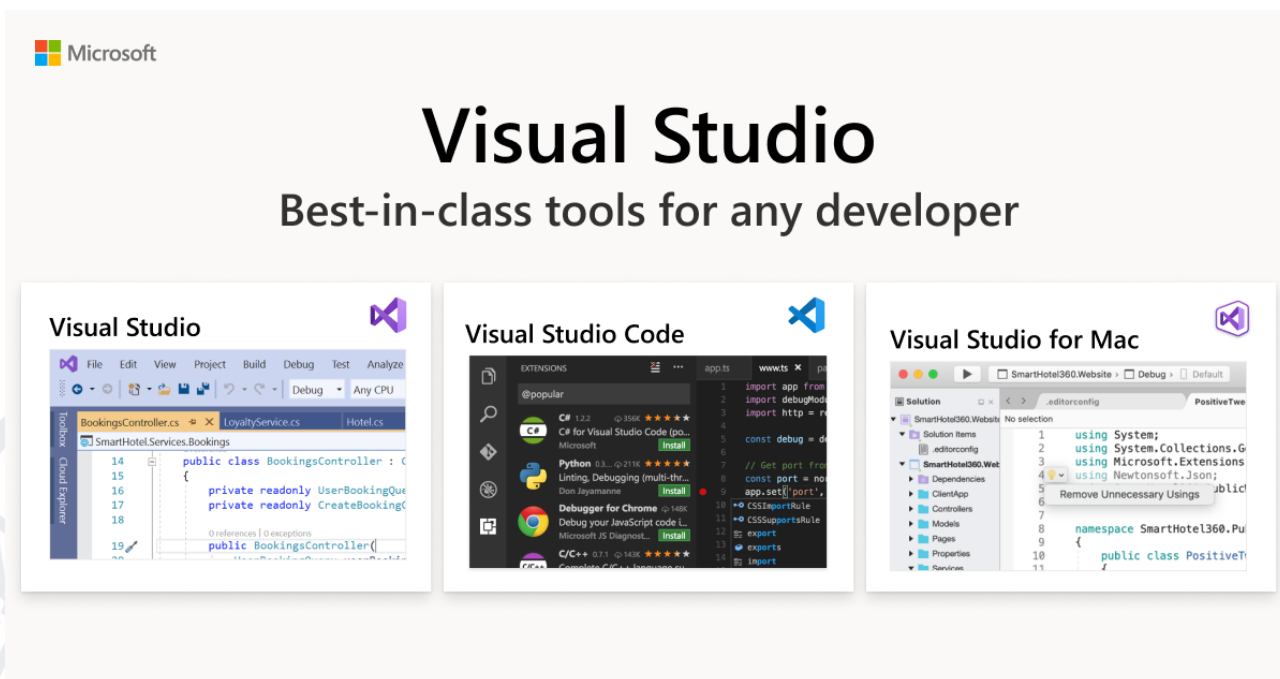


Рисунок 2.1 – Ілюстрація Visual Studio

Microsoft Visual Studio — це IDE, створена Microsoft і використовується для різних типів розробки програмного забезпечення, наприклад комп'ютерних програм, веб-сайтів, веб-програм, веб-сервісів та мобільних додатків. Він містить інструменти завершення, компілятори та інші функції для полегшення процесу розробки програмного забезпечення.

Microsoft Visual Studio - одне з інтегрованих середовищ розробки, розроблена на C++ і C#, підтримується Windows OS. Це середовище розробки перекладено десятима мовами (також і російською). У Visual Studio творець може вести розробку веб-сайтів, веб-служб, писати консольні програми, а також програми з графічним інтерфейсом. Також VS підтримує різноманітні доповнення. Найвідоміші доповнення - це ReSharper (виконує пошук помилок у коді під час написання коду програми розробником до компіляції); Visual Assist (на відміну від ReSharper підтримує також C++); AnkhSVN (використовує Visual Studio систему контролю версій, яка зветься Subversion).



Рисунок 2.2 – Логотип Visual Studio

Visual Studio — найпопулярніший інструмент розробки IDE та програмного забезпечення для створення додатків .NET за допомогою C#. Це продукт від Microsoft, який також дуже популярний серед розробників, які створюють програми за допомогою екосистеми .NET [24].

Для розробки програмної системи було обрано Visual Studio через міцну інтеграцію з C# та додатковими плагінами необхідними у процесі розробки. А також тому, що це зручне і функціональне середовище розробки.

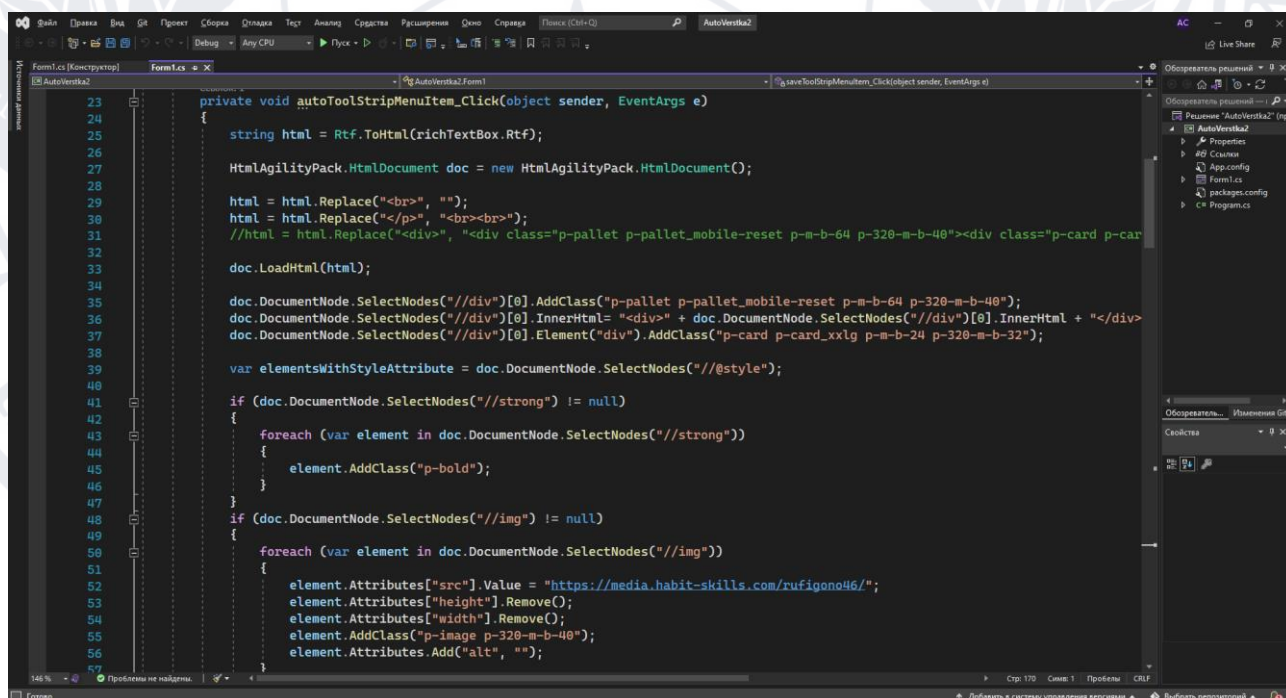


Рисунок 2.3 – Visual Studio IDE.

Невід'ємною частиною Visual Studio є NuGet Package Manager. NuGet — це менеджер пакетів для платформи розробки Microsoft, включаючи .NET. Інструменти клієнта NuGet надають можливість створювати та використовувати пакети. Галерея NuGet (nuget.org) — це центральне сховище пакетів, яке використовується всіма авторами та споживачами пакетів [25].

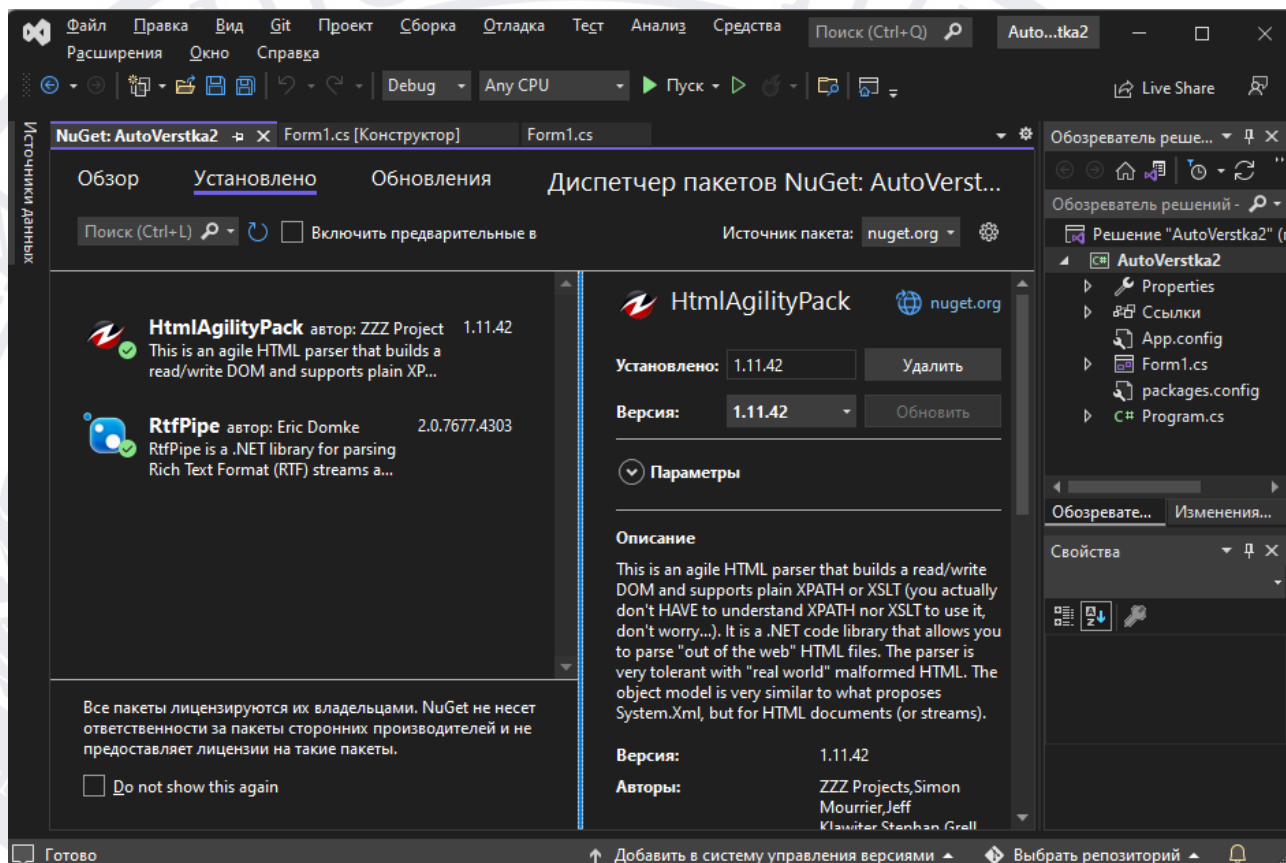


Рисунок 2.4 – Встановлені бібліотеки у вікні NuGet Package Manager.

Завдяки NuGet Package Manager інструменту вдалося легко встановити необхідні бібліотеки RtfPipe та HtmlAgilityPack про які йде мова нижче.

Windows Forms

Windows Forms дозволяє розробляти інтелектуальні клієнти. Інтелектуальний клієнт — це програма з повнофункціональним графічним інтерфейсом, проста в розгортанні та оновленні, здатна працювати за наявності або відсутності підключення до Інтернету та використовує безпечніший доступ

до ресурсів на локальному комп'ютері в порівнянні з традиційними програмами Windows.

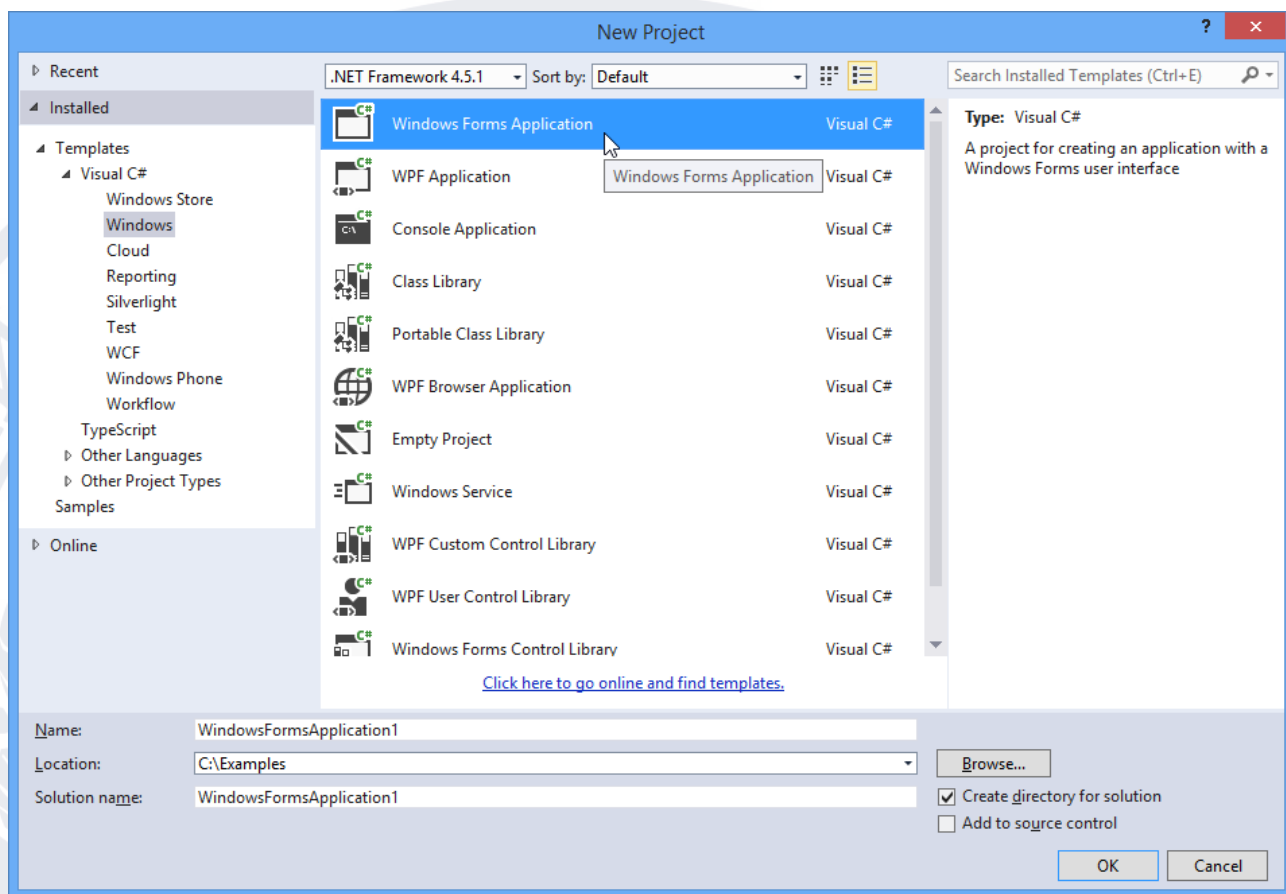


Рисунок 2.5 – Створення проекту Windows Forms у Visual Studio.

Щоб створити GUI програми в Microsoft .NET, потрібно використати Windows Forms. Windows Forms – новий стиль побудови програми на базі класів .NET Framework class library. Вони мають власну модель програмування, яка більш досконала, ніж моделі, що базуються на Win32 API або MFC, і вони виконуються в керованому середовищі .NET Common Language Runtime (CLR). Ця стаття дає уявлення про те, що таке Windows Forms, розглядаючи її від моделі програмування до Microsoft Intermediate Language та JIT-транслятора.

Microsoft .NET – нова платформа, яка базується на Windows. Це ціла нова парадигма програмування, яка змінить шлях, яким ви думаєте про написання програм для Windows. Вона реалізована на бібліотеці класів .NET Framework

class library та містить більш єдину модель програмування, покращений захист та багатші можливості для написання повнофункціональних веб-додатків. І це лише початок.

Windows Forms – одна з найцікавіших можливостей Microsoft .NET. Якщо розробник знає MFC (або Windows API), то Windows Forms є гарним початком для роботи з .NET Framework class library, тому що вона дозволяє писати традиційні GUI додатки з вікнами, формами і т.п. речами. Одного разу, почавши працювати з Windows Forms, можна швидко зрозуміти .NET Framework.

Головна вигода від написання Windows-додатків з використанням Windows Forms - це те, що Windows Forms гомогенізують (створюють більш однорідну (гомогенну) структуру) програмну модель і усувають багато помилок і протиріч від використання Windows API. Наприклад, кожен досвідчений програміст під Windows знає, що деякі стилі вікна можуть застосовуватися лише до вікна, коли вже створено. Windows Forms значною мірою усувають таку суперечність. Якщо ви хочете існуючому вікну встановити стиль, який може бути присвоєний тільки в момент створення вікна, то Windows Forms спокійно знищить вікно і знову створить його із зазначеним стилем. Крім того, .NET Framework class library набагато багатший, ніж Windows API, і коли ви писатимете програми, використовуючи Windows Forms, ви отримаєте в розпорядження більше можливостей. Написання програми за допомогою Windows Forms потребує менше коду, ніж програми, які використовують Windows API або MFC.

Інша вигода від Windows Forms - ви використовуєте той самий API, незалежно від мови програмування, яку ви вибрали. У минулому вибір мови програмування керував вибором API. Якщо ви програмували у Visual Basic, ви використовували один API (реалізований мовою Visual Basic), тоді як програмісти C використовували Win32 API, а програмісти C++, взагалі кажучи, використовували MFC. MFC-програмісту було важко перейти на Visual Basic і навпаки. Але тепер такого немає. Всі програми, які використовують Windows

Forms, використовують один API із .NET Framework class library. Знання одного API дозволить програмісту писати програми фактично будь-якою мовою, яку він вибере.

Windows Forms не менше, ніж сучасна модель програмування для GUI додатків. На відміну від моделі програмування Win32, в якій багато що йде ще від Windows 1.0, нова модель була розроблена з урахуванням усіх сучасних вимог. Ціль цієї статті полягає в тому, щоб познайомити читача з моделлю програмування Windows Forms. Щоб компілювати та виконувати приклади коду, наведеного далі, на вашому комп'ютері має бути встановлений пакет Microsoft .NET Framework SDK (.NET Framework SDK Beta 1 доступний на сайті Microsoft).

Після відкриття (або створення) проекту Windows Forms ми можемо відкрити форму в редакторі. Це покаже нам фактичний перегляд конструктора, який відображає форму або елемент керування користувача (1), з яким ми працюємо. З сітки властивостей ліворуч (2) ми можемо маніпулювати доступними властивостями, такими як текст, елементи та шрифти, а також можемо підписатися на події для подвійних клацань, наведення миші тощо. З правого боку панель інструментів конструктора містить список компонентів, які можна додати до форми (4). Унизу ми можемо знайти вкладки навігації для перемикання між виглядом конструктора та кодом позаду (3):

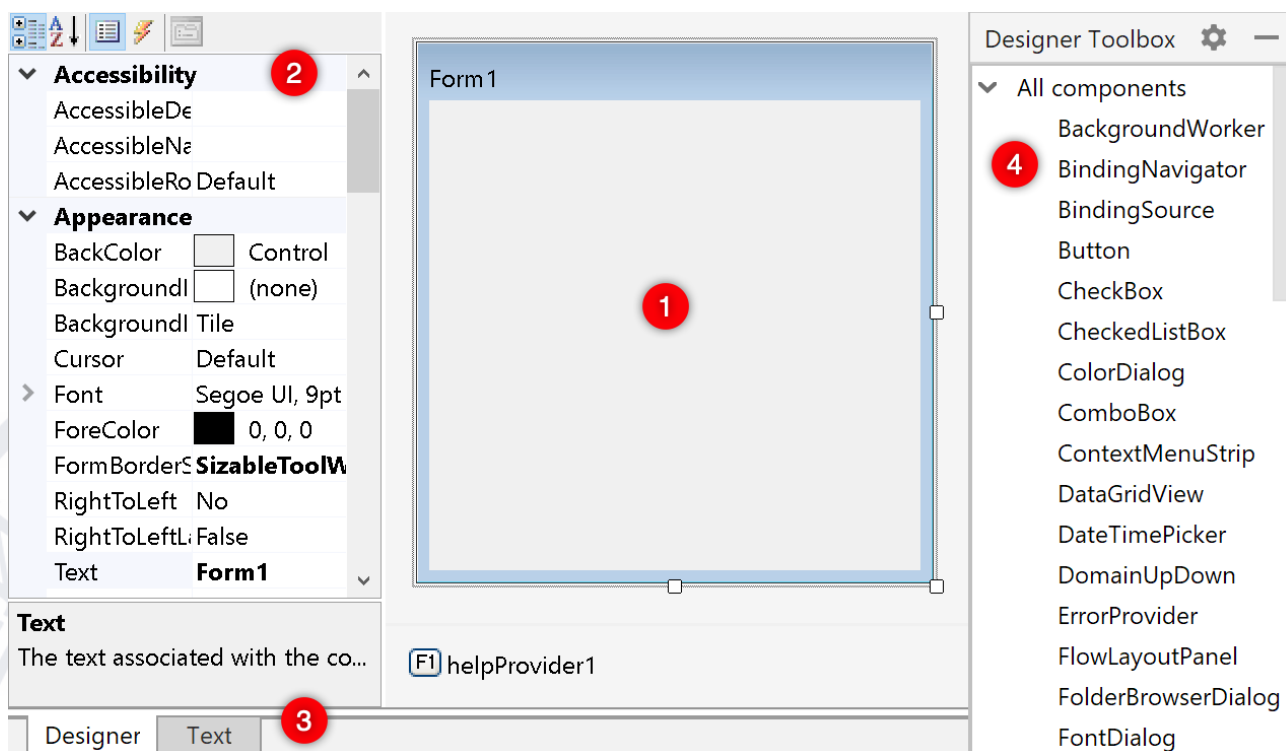


Рисунок 2.6 – елементи редактора Windows Forms.

У Windows Forms термін "форма" – синонім вікна верхнього рівня. Головне вікно програми – форма. Будь-які інші вікна верхнього рівня, які має програму – також форми. Вікна діалогу також вважаються формами. Незважаючи на назву, програми, які використовують Windows Forms, не виглядають як форми. Подібно до традиційних Windows-додатків програми здійснюють повний контроль над подіями у власних вікнах.

Програмісти бачать Microsoft .NET через лінзу .NET Framework class library. Уявіть MFC на порядок більше і ви отримаєте точну картину про ширину та глибину .NET Framework class library. Щоб полегшити протиріччя в позначеннях і надати організацію багатьом сотням класів, .NET Framework class library розбита на ієрархічні розділи на ім'я. Кореневий розділ, System, визначає фундаментальні типи даних, що використовуються всіма програмами .NET.

Програми, які використовують Windows Forms, використовують класи System.WinForms. Цей розділ включає такі класи, як Form, який моделює поведінку вікон чи форм; Menu, яке представляє меню; Clipboard, який дозволяє програмам Windows Forms використовувати буфер обміну. Він також

містить численні класи, що надають засоби керування, наприклад: `Button`, `TextBox`, `ListView`, `MonthCalendar` і т.д. Ці класи можуть бути включені до програми або з використанням тільки імені класу, або з використанням повного імені, наприклад: `System.Windows.Forms.Button`.

В основі майже кожної програми, написаної із застосуванням `Windows Forms`, - похідний клас від `System.Windows.Forms.Form`. Зразок цього класу представляє головне вікно програми. `System.Windows.Forms.Form` має безліч властивостей та методів, які мають багатий програмний інтерфейс до форм. Бажаєте знати розміри клієнтської області форми? У `Windows` ви б викликали функцію `API GetClientRect`. У `Windows Forms` потрібно використовувати властивості `ClientRectangle` або `ClientSize`.

Програми, засновані на `Windows Forms`, які використовують кнопки, списки та інші типи компонентів `Windows`, використовують класи керування `System.Windows.Forms`, які значно спрощують програмування керування. Бажаєте створити стилізовану кнопку із зображенням у вигляді фону? Немає проблем. Увімкніть потрібне зображення в об'єкт `System.Drawing.Bitmap` та призначте його властивості кнопки `BackgroundImage`. Як щодо керування кольором? Ви коли-небудь намагалися налаштувати колір тла текстового поля? У `Windows Forms` це просто: потрібно просто надати колір властивості `BackColor`, все відстальне система зробить сама.

Інший важливий "будівельний" блок програми, який використовує `Windows Forms` – клас `System.Windows.Forms.Application` на ім'я `Application`. Цей клас містить статичний метод `Run`, який завантажує програму та відображає вікно.

Якщо програми, які є `Windows Forms`, не обробляють повідомлення, як вони відповідають на введення користувача або знають коли малювати? Багато класів мають віртуальні методи, які можна перевизначити... Наприклад, `System.Windows.Forms.Form` містить віртуальний метод `OnPaint`, який викликається, коли клієнтська область форми потребує оновлення. `OnPaint` - один із багатьох віртуальних методів, який можна перевизначити у похідному класі для формування інтерактивних форм.

Інша важлива грань моделі програмування Windows Forms - механізм, який форми використовують для відповіді на введення в меню, засобів керування та інших елементів GUI програми. Традиційні програми Windows обробляють повідомлення WM_COMMAND і WM_NOTIFY використовуючи події процесу Windows Forms. У C# та іншими мовами, які підтримують .NET Common Language Runtime (CLR), події - члени типу першого класу нарівні з методами, полями та властивостями. Фактично всі класи, що управляють (control classes) Windows Forms (а також і багато не керуючих класів) створюють події. Наприклад, кнопка (примірник System.WinForms.Button) після натискання створює подію Click.

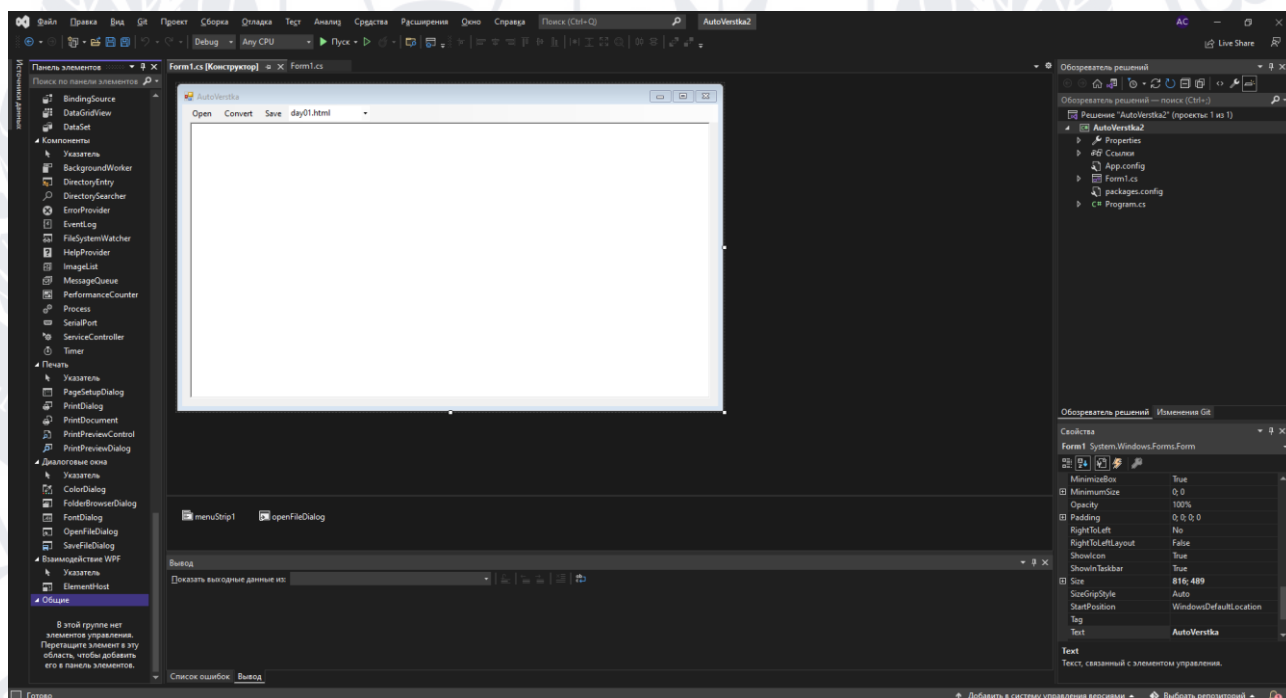


Рисунок 2.7 – редактор Windows Forms.

2.2 Переваги Microsoft Visual Studio

Нижче наведено ключові переваги IDE Microsoft Visual Studio:

- середовище розробки підтримує роботу з декількома мовами програмування, до яких відносяться найпопулярніші – C# / C++.

- зручна робота із системами версії контролю;
- зручний та функціональний редактор коду;
- підказки до коду та автозавершення;
- багатофункціональний графічний редактор, який дає можливість створювати Windows Forms додатки;
- велика кількість додаткових модулів, які роблять розробку швидшою;
- підтримка усіх нововведень;
- зручний інтерфейс та робота із структурою проекту.
- безліч «гарячих клавіш» для швидкості роботи.

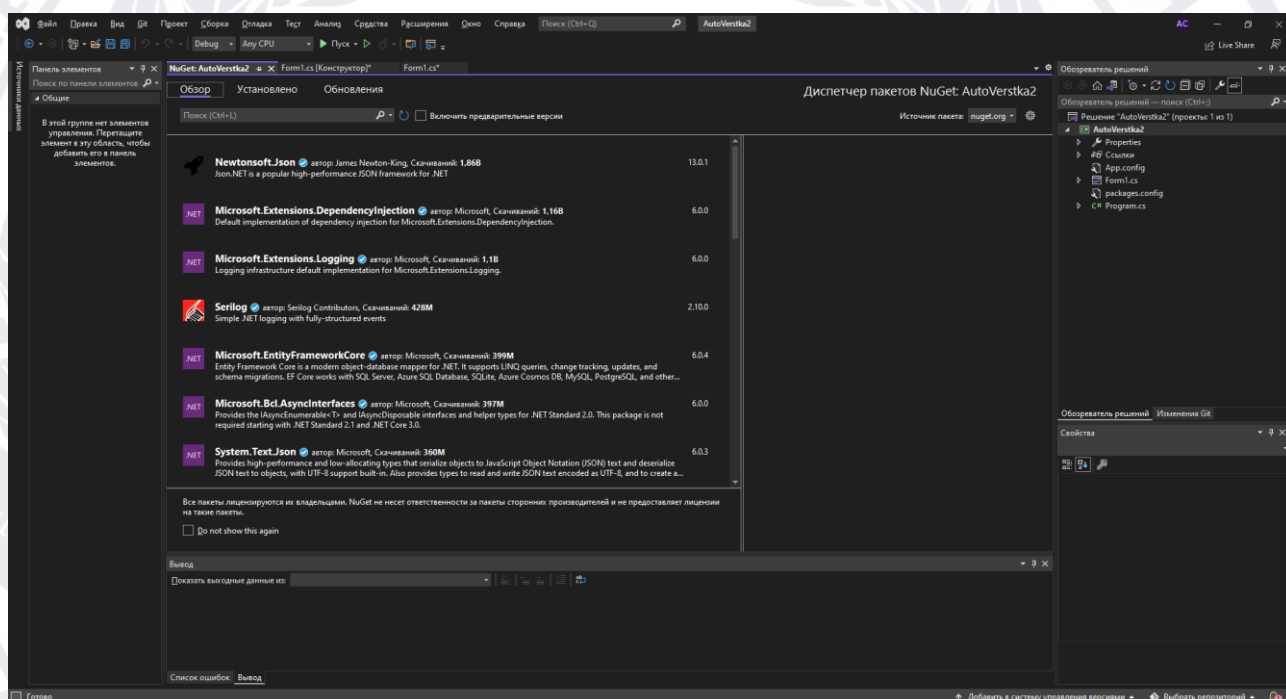


Рисунок 2.8 – nuget package manager.

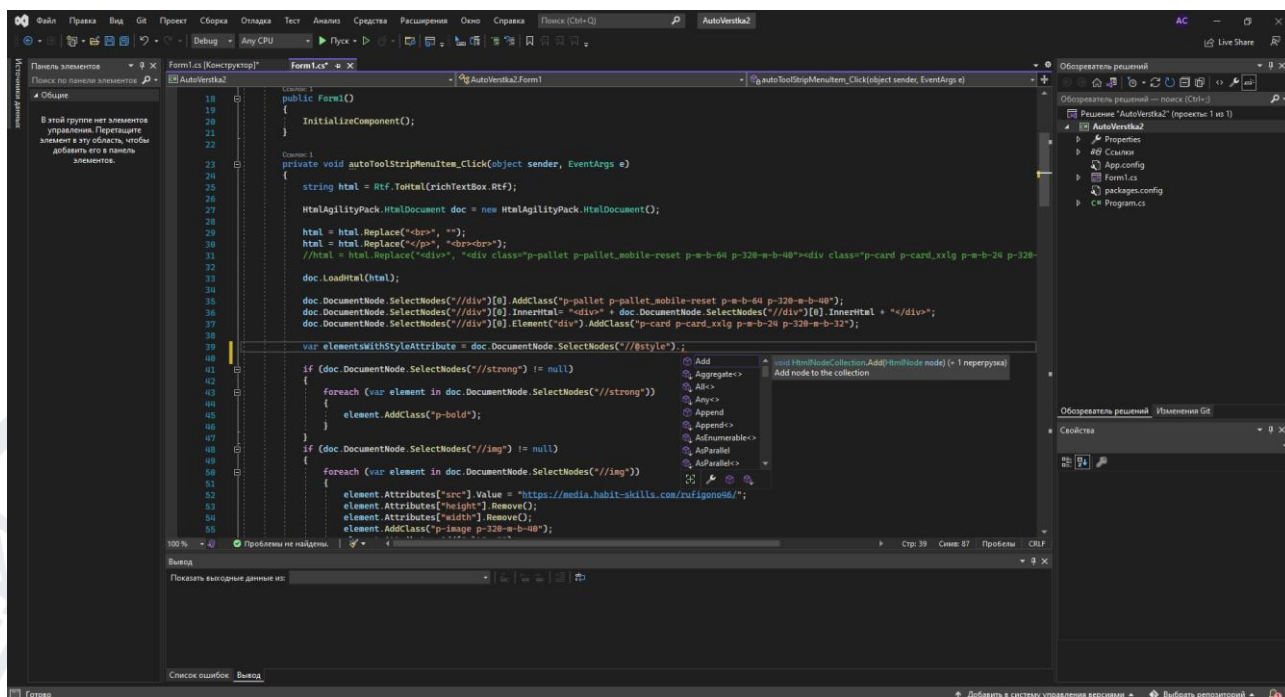


Рисунок 2.9 – автозавершення відповідно до коду із бібліотек та фреймворків

2.3 Технології

RTF

Розширений текстовий формат (RTF), введений і задокументований Microsoft, являє собою метод кодування відформатованого тексту та графіки для використання в програмах. Формат полегшує між платформний обмін документами з іншими продуктами Microsoft, таким чином слугує меті взаємодії. Ця можливість робить його стандартом передачі даних між програмним забезпеченням для обробки текстів, і, отже, вміст можна передавати з однієї операційної системи в іншу без втрати форматування документа. Специфікації формату файлів доступні Microsoft для загальнодоступного завантаження і на них можна посилатися з точки зору розробника.

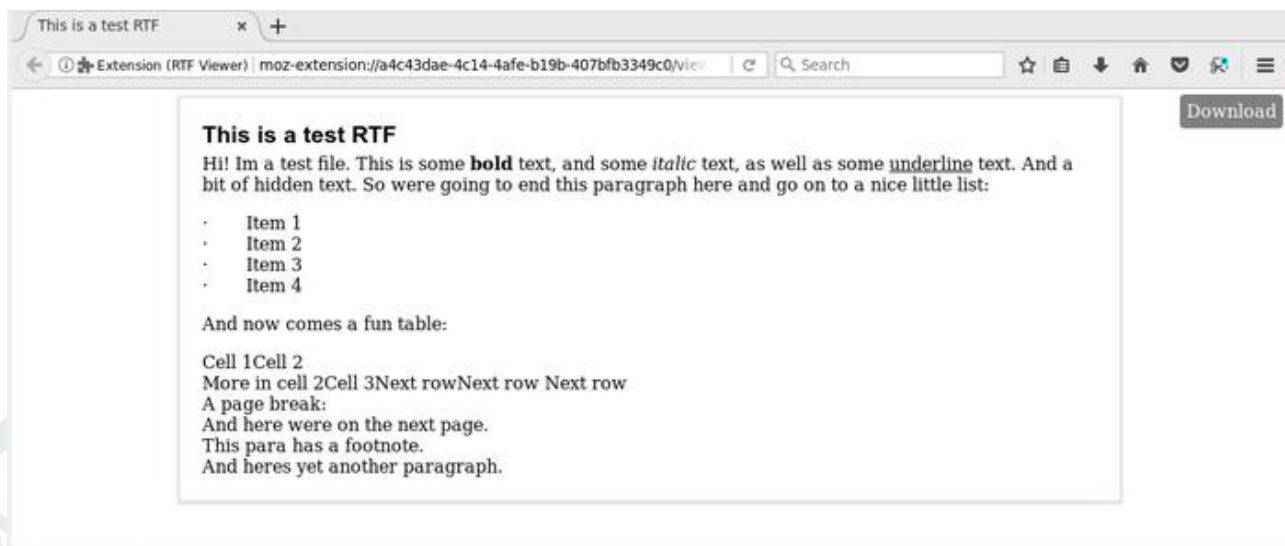


Рисунок 2.10 Тестовий файл RTF

Rich Text Format, RTF (англ. rich - багатий; "формат збагаченого тексту") – пропрієтарний [19] міжплатформний формат зберігання текстових документів з форматуванням, запропонований групами програмістів, що заснували компанії Microsoft і Adobe, як метатеговий формат для редактора Word 1982 року. Він був призначений як універсальний формат, який може використовуватися більшістю текстових процесорів, що полегшує людям обмін документами Word з людьми, які не використовують Word. Він також був включений як формат за замовчуванням, який використовується вбудованою програмою WordPad у Windows — легким текстовим процесором.

До того, як вони були замінені файлами HTML, RTF також використовувався як основа для файлів довідки Windows.

Оскільки більшість текстових процесорів можуть читати та записувати файл RTF, це означає, що якщо ви створите його в Windows, то зможете надіслати його колезі, який використовує Mac, Linux або Chromebook, без будь-яких проблем. Він також широко використовується в інших типах програм, таких як клієнти електронної пошти.

Microsoft припинила розробку RTF у 2008 році, але вона все ще широко підтримується програмами майже в кожній операційній системі.

RtfPipe

RtfPipe — це бібліотека .NET для аналізу потоків формату RTF (Rich Text Format) і перетворення їх у HTML. Хоча спочатку він був адаптований з роботи, розпочатої Яні Джанудісом, він був повністю переписаний для підтримки додаткових функцій. У поєднанні з бібліотекою BracketPipe цю бібліотеку також можна використовувати для перетворення потоків RTF у різні текстові формати, такі як Markdown і Textile.

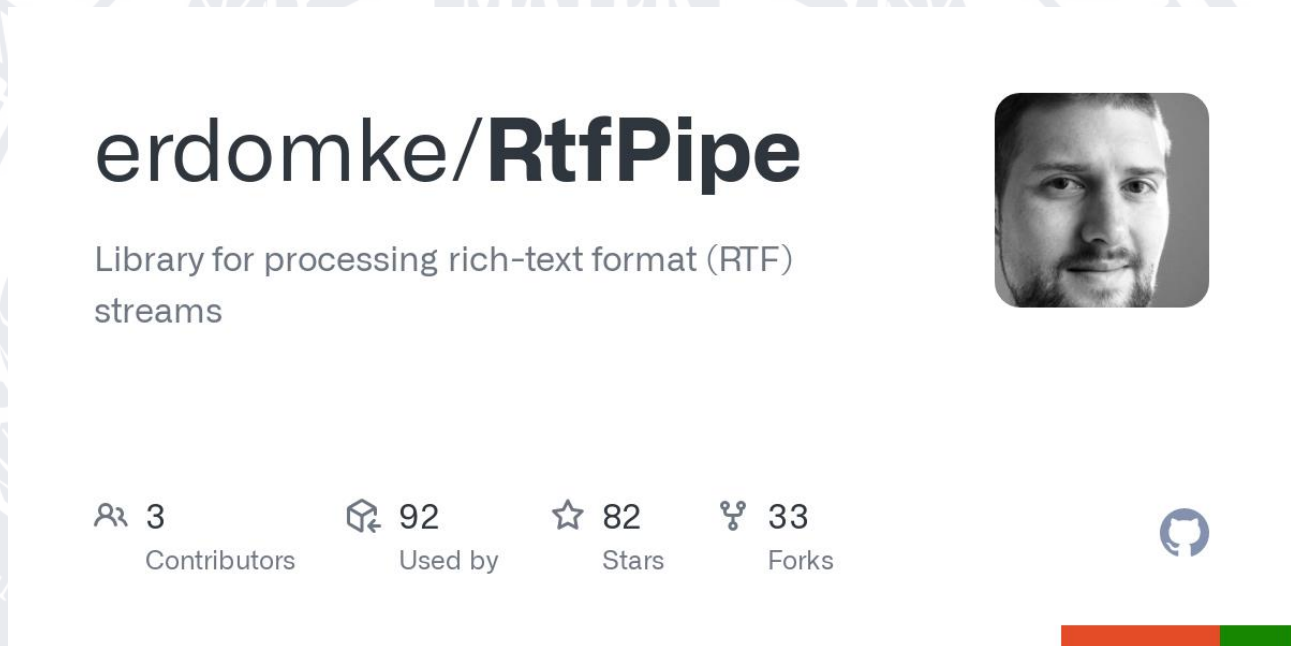


Рисунок 2.11 Сторінка RtfPipe

Ця бібліотека намагається підтримувати основні функції RTF, описані в Специфікації RTF 1.9.1. Ці функції включають:

- Форматування символів (жирний, курсив, колір, ...)
- Таблиці (включаючи вкладені таблиці)
- Списки
- Гіперпосилання
- Картинки
- Рівні заголовків
- Інкапсуляція HTML (наприклад, як виконується Outlook)

З огляду на це, є численні випадки для нетривіальних документів, коли бібліотека не буде створювати «правильне» візуальне уявлення в порівнянні з іншими засобами для читання RTF (наприклад, MS Word).

HtmlAgilityPack

tomap/ HtmlAgilityPack



HtmlAgilityPack - Fork of codeplex original.
Introduced a few fixes



1

Contributor



0

Issues



7

Stars



3

Forks



Рисунок 2.12 Сторінка RtfPipe

Це швидкий синтаксичний аналізатор HTML, який створює DOM для читання/запису та підтримує звичайний XPATH або XSLT (вам насправді не потрібно розуміти ні XPATH, ні XSLT, щоб використовувати його, не хвилюйтеся...). Це бібліотека коду .NET, яка дозволяє аналізувати HTML-файли «поза мережею». Синтаксичний аналізатор дуже толерантний до "реального світу" неправильного HTML. Об'єктна модель дуже схожа на те, що пропонує System.Xml, але для HTML-документів (або потоків).

Бібліотека RtfPipe покликана вирішувати одне завдання. Вона конвертує вміст rtf у html. Після цього нам залишається тільки почистити рядок, що вийшов, від непотрібних тегів, що робиться досить легко за допомогою того ж HtmlAgilityPack.

Висновок до розділу 2

Розроблено загальну архітектуру програмної системи для автоматизації верстання веб-сайту з навчальним контентом. Обрано наступні засоби для її написання:

- мова програмування C#,
- формат файлів RTF,
- бібліотеки RtfPipe та HtmlAgilityPack.

Також підведено підґрунтя для використання наступних технологій:

- IDE Microsoft Visual Studio,
- інтерфейс програмування додатків Windows Forms,

Обґрунтовано доцільність використання даних інструментів та технологій для розробки програмної системи для автоматизації верстання веб-сайту з навчальним контентом. У наступному розділі буде доведено, що ці інструменти та технології обрано вірно і вони повністю підходять для цієї цілі.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

3.1 Програмна реалізація AutoVerstka

AutoVerstka – обробка окремих елементів.

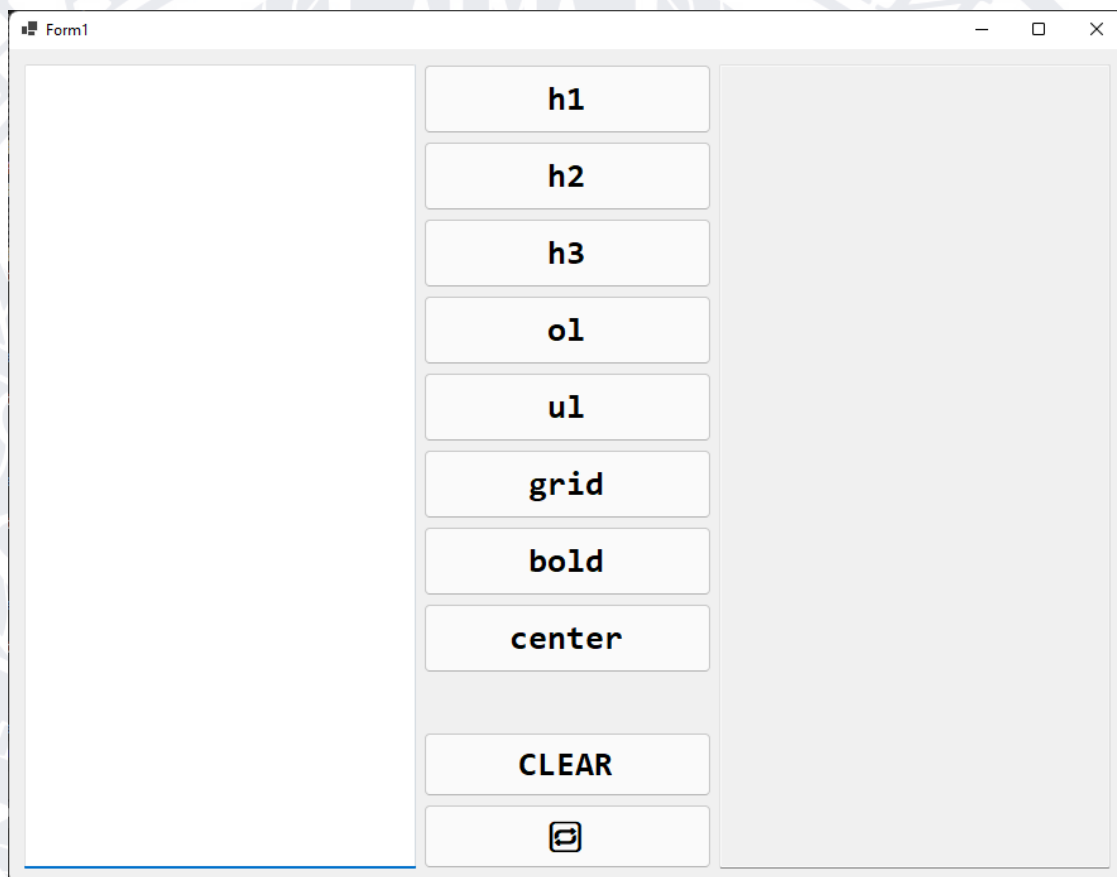


Рисунок 3.1 – зовнішній вигляд програми AutoVerstka.

Дизайн програми обрано мінімалістичний - для доступності і простоти у користуванні. Присутні кілька кнопок з відповідними підписами та два текстових поля: ліве для вводу інформації, праве для виводу. На вхід програма отримує звичайний текст, після конвертації на виході отримуємо html код елементу готового до розміщення на сторінці. Після конвертації програма сама зберігає результат у буфер обміну, що забезпечує додаткову економію часу.

Далі наведено частину лістингу програмного коду з відповідними коментарями:

Функція, що відповідає за заголовки першого рівня (h1):

```
private void button1_Click(object sender, EventArgs e)
{
    textBox2.Text = "<span class=\"p-h1\">" +
    textBox1.Text + "</span>";
}
```

Подібним чином працюють і функції, що відповідають за заголовки h2, h3, center (відцентрований текст) та bold (жирний текст).

Функція, що відповідає за копіювання тексту до буферу обміну після конвертації:

```
private void textBox2_TextChanged(object sender,
EventArgs e)
{
    if (textBox2.Text != "")
        Clipboard.SetText(textBox2.Text);
}
```

Функція, що відповідає за нумеровані списки:

```
private void button4_Click(object sender, EventArgs e)
{
    textBox2.Text = "<ol class=\"p-list\">";
    string[] newText = textBox1.Text.Split("\r\n");
    for (int j = 0; j < newText.Length; j++)
    {
        if (newText[j] != "")
        {
            int k = 0;
            for (int i = 0; i < newText[j].Length; i++)
            {
                if (newText[j][i] == '.')
                {
                    k++;
                    break;
                }
            }
        }
    }
}
```



```

        else
        {
            k++;
        }
    }
    if (k != newText[j].Length)
    {
        newText[j] = newText[j].Substring(k,
newText[j].Length - k);
    }
}
for (int j = 0; j < newText.Length; j++)
{
    if (newText[j] != "")
        textBox2.Text += "<li class=\"p-list__item\">"
+ newText[j] + "</li>";
}
textBox2.Text += "</ol>";
}

```

Функція, що відповідає за перенесення тексту з вихідного поля форми у вхідне:

```

private void button7_Click(object sender, EventArgs e)
{
    textBox1.Text = textBox2.Text;
}

private void button8_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
}

```

Функція, що відповідає за нумеровані списки:

```

private void button9_Click(object sender, EventArgs e)
{
    textBox2.Text = "";
    string[] newText = textBox1.Text.Split("\r\n");
    for (int j = 0; j < newText.Length - 1; j++)
    {
        if (newText[j] != "")
            textBox2.Text += newText[j] + "<br><br>";
    }
}

```

Функція, що відповідає за сітку (grid):

```
private void button10_Click(object sender, EventArgs e)
{
    textBox2.Text = "<div class=\"p-grid p-grid_four p-m-  
b-72 p-320-m-b-40\">";
    string[] newText = textBox1.Text.Split("\r\n");
    for (int j = 0; j < newText.Length; j++)
    {
        if (newText[j] != "")
            textBox2.Text += "<div class=\"p-card p-  
card_lg\">" + newText[j] + "</div>";
    }
    textBox2.Text += "</div>";
}
```

Програма AutoVerstka допомагає у роботі з конкретними елементами тексту, на противагу другій, яка конвертує повністю сторінки, проте може робити це з меншою точністю.

3.2 Програмна реалізація AutoVerstka2

AutoVerstka2 – обробка сторінок цілком.

Нижче наведений результат роботи даного модуля програмної системи:

Ця програма здатна конвертувати RTF в HTML з деякими змінами, що задані в коді, та налаштовані під специфіку роботи Автора.

```

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    using (OpenFileDialog ofd = new OpenFileDialog() {
        ValidateNames = true, Multiselect = false,
        Filter = "Word Document|*.docx|Word 97 - 2003 Document|*.doc" })
    {
        if (ofd.ShowDialog() == DialogResult.OK)
        {
            object readOnly = true;
            object visible = true;
            object save = false;
            object fileName = ofd.FileName;
            object missing = Type.Missing;
            object newTemplate = false;
            object docType = 0;
            Microsoft.Office.Interop.Word._Document oDoc = null;
            Microsoft.Office.Interop.Word._Application oWord =
                new Microsoft.Office.Interop.Word.Application()
                { Visible = false };
            oDoc = oWord.Documents.Open(
                ref fileName, ref missing, ref readOnly, ref missing,
                ref missing, ref missing, ref missing, ref missing,
                ref missing, ref missing, ref missing, ref visible,
                ref missing, ref missing, ref missing, ref missing);
            oDoc.ActiveWindow.Selection.WholeStory();
            oDoc.ActiveWindow.Selection.Copy();
            IDataObject data = Clipboard.GetDataObject();
            richTextBox.Rtf = data.GetData(DataFormats.Rtf).ToString();
            oWord.Quit(ref missing, ref missing, ref missing);
        }
    }
}

```

Рисунок 3.2 – функція відкриття файлу

Реалізована функція, що дозволяє відкривати документи в форматі .docx і зчитувати з них RTF. Завдяки цій функції користувач матиме можливість конвертувати файли цілком. Вміст файлу далі виводиться в текстове поле де його можна переглянути і відредагувати, за потреби.

Функція конвертації має поетапну структуру, де кожен елемент форматування конвертується у свою чергу. Спочатку видаляються непотрібні

теги та стилі. Потім додаються необхідні класи, відповідно до кожного елементу.

```
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    var path = DayComboBox.Text;
    string text = richTextBox.Text;
    File.WriteAllText(path, text);
    MessageBox.Show("Seved!");
}
```

Рисунок 3.3 – функція збереження

Коли конвертацію завершено логічним кроком є зберегти результат. І ця функція також наявна у програмі, крім виводу в текстове поле вона може зберігати файл результат у файл з розширенням .html.

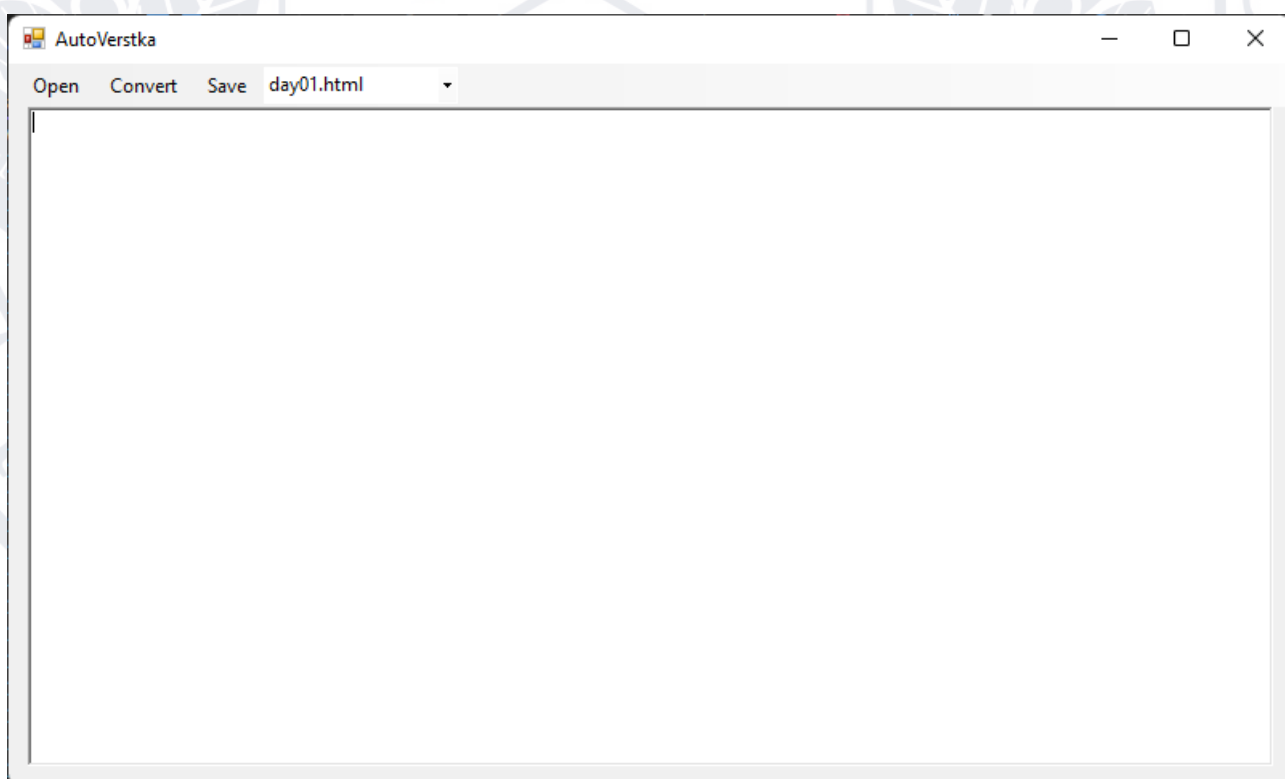


Рисунок 3.4 – вигляд програми AutoVerstka2.

Програма AutoVerstka2 має досить практичний і легкозрозумілий дизайн, достатньо одного разу розібратися. як вона працює і далі питань у роботі з нею виникати не має.

Програма призначена для конвертації сторінок повністю. Точніше, конвертації всього змісту документа, який подається на вхід.

Перед тим як ознайомитись з результатом конвертації варто оглянути вхідний текст, за потреби відредагувати його до належного вигляду.

Переваги програми у тому, що є кілька можливих шляхів внесення даних такі як:

- 1) Відкривати обрані файли
- 2) Вставляти з буфера обміну

Якщо доцільно конвертувати лише частину тексту то краще скористатися другим. Натомість, у більшості випадків, щоб зайвий раз не відкривати файл, навантажувати систему і витрачати час, доцільніше відкрити і конвертувати файл повністю.

Звичайно, обидві ці програми не можуть працювати повністю самостійно, їх повинен контролювати верстальник. Проте вони значно полегшують його монотонну працю.

3.3 Інструкція оператора

Запуск

Стосується обох модулів програми: Для запуску програми необхідно відкрити файл застосунку.

Далі описано дії, які користувачу необхідно виконати, щоб успішно використовувати програму у роботі з програмою AutoVerstka.

Використання програми AutoVerstka

Необхідно:

- 1) Обрати фрагмент тексту, який містить необхідний елемент.
- 2) Виділити його курсором та скопіювати у буфер обміну будь-яким зручним способом.
- 3) Вставити скопійований фрагмент у ліве текстове поле програми.
- 4) Натиснути одну з кнопок, що розміщені в центральній частині форми, відповідно до обраного елемента.

Після цього програма автоматично конвертує та скопіює в буфер обміну користувача результат.

Далі описано дії, які користувачу необхідно виконати, щоб успішно використовувати програму у роботі з програмою AutoVerstka.

Використання програми AutoVerstka2

Необхідно:

- 1) Обрати фрагмент тексту, який містить необхідний елемент. Або файл цілком.
 - a. У першому випадку: виділити його курсором та скопіювати у буфер обміну будь-яким зручним способом. Вставити скопійований фрагмент у ліве текстове поле програми.
 - b. У другому випадку: натиснути кнопку Open розташовану на верхній панелі програми. Обрати потрібний файл за допомогою файлового діалогу. Після цього вміст обраного файлу автоматично буде поміщений у текстове поле.
- 2) Натиснути кнопку Convert, що розміщена на верхній панелі програми.
- 3) Скопіювати вміст текстового поля самостійно, або зберегти його у файл за допомогою кнопки Save, що розміщена на верхній панелі програми. Поруч з кнопкою Save знаходиться випадаючий список, завдяки якому можна обрати назву файлу в який буде збережено результат.

AutoVerstka2 працює подібним до AutoVerstka чином та має кілька суттєвих відмінностей. Програма AutoVerstka2 не пропонує користувачу обирати який елемент він бажає конвертувати. Натомість, вона автоматично розпізнає всі елементи з структурою подібною до необхідних.

3.4 Приклад роботи системи

Приклад роботи AutoVerstka

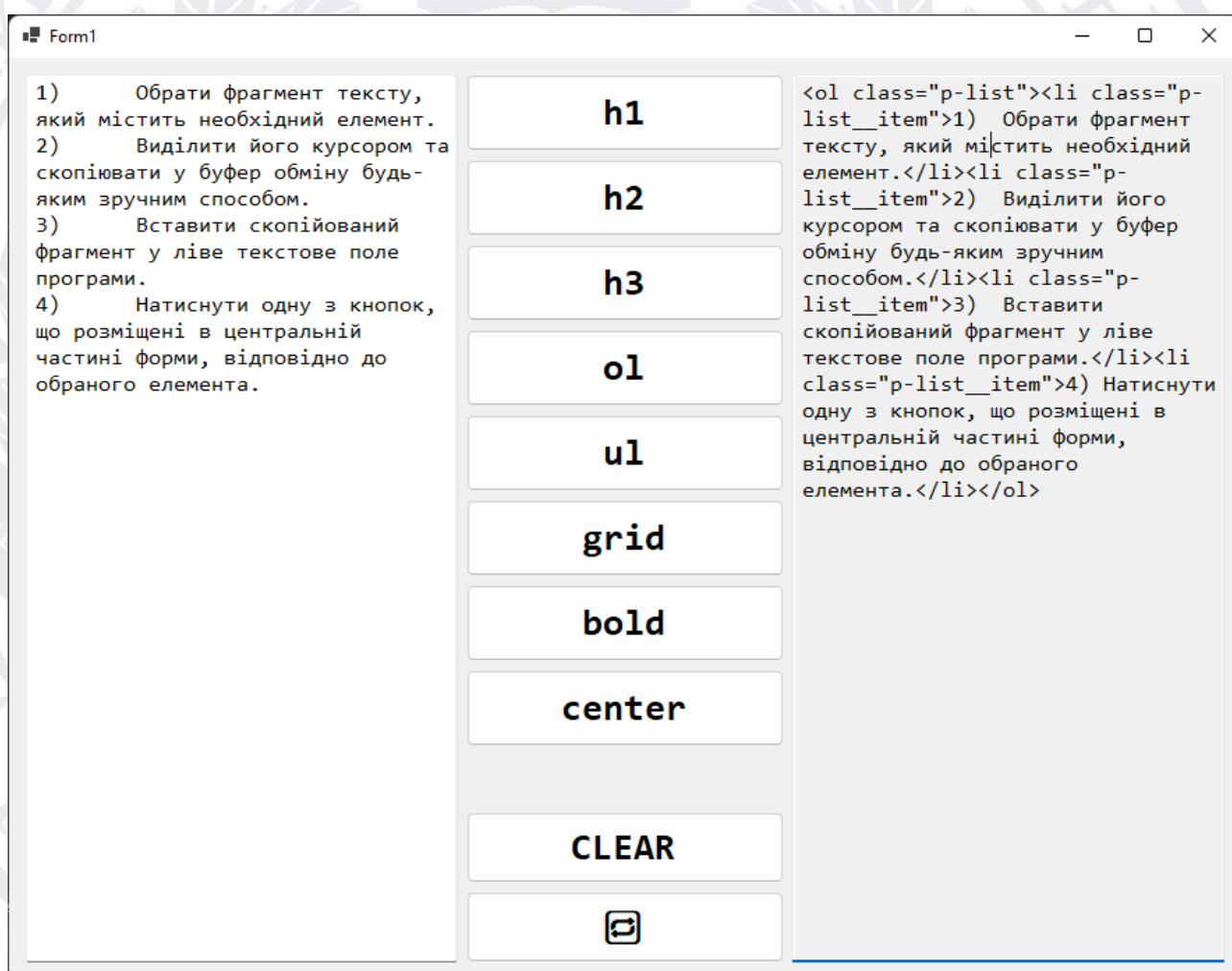


Рисунок 3.5 – результат конвертації на прикладі нумерованого списку

Як можемо бачити - програма успішно конвертує нумеровані списки у потрібний верстальнику формат, коректно загортає пункти у теги, додає класи.

Подібним чином відбувається перетворення інших елементів, швидко і легко. На відміну від ручної праці, яка вимагає у десятки разів більше натискань на клавіші користувачем. Щоб вручну зробити подібний список необхідно скопіювати та вставити окремо кожен пункт, загорнути його у відповідний тег.

Приклад роботи AutoVerstka2

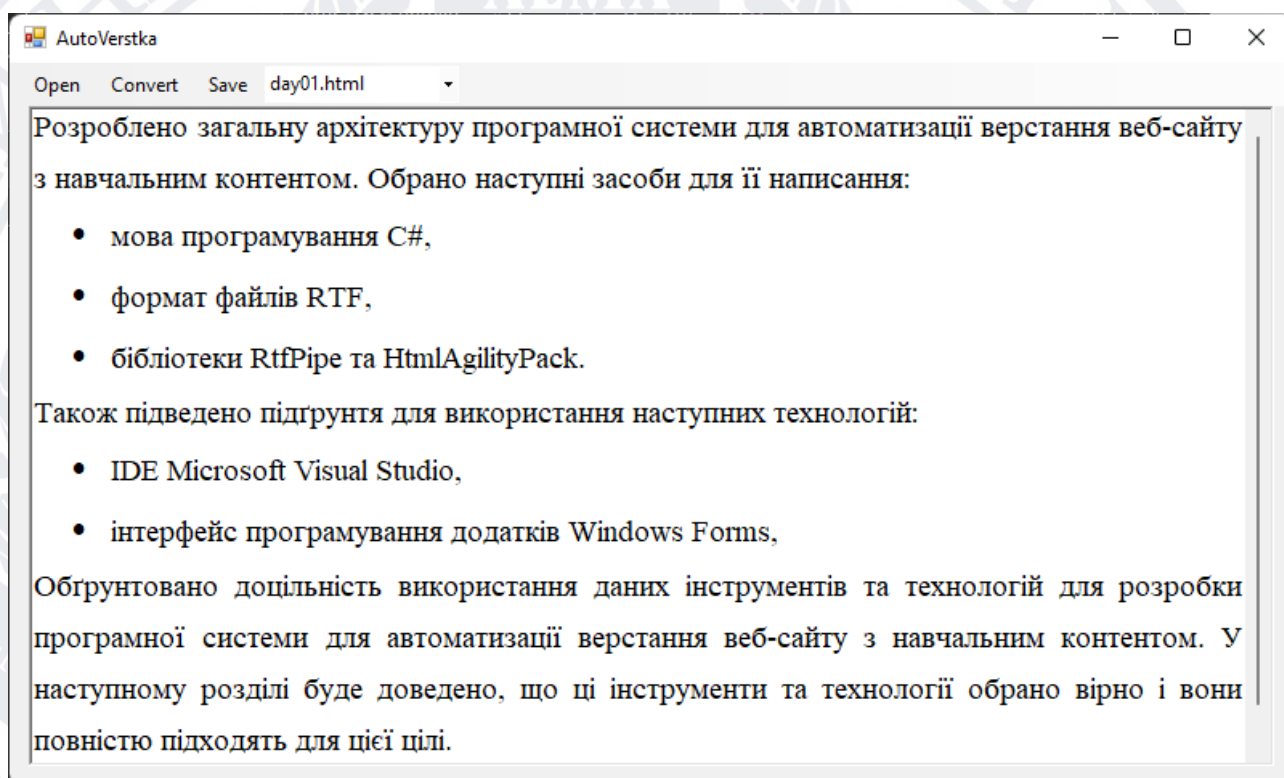


Рисунок 3.6 – до конвертації, на прикладі частини з бакалаврської роботи.

На цьому прикладі можемо бачити, що RichTextBox достатньо добре сприймає RTF, що й не дивно, бо саме для подібних задач він і був створений.

Вивід вмісту файлу необхідний для того, щоб користувач перевінив його перед конвертацією. Трапляються випадки коли користувач випадково обирає не той файл і розуміє це лише після того, як зберіг та переглянув результат. Щоб мінімізувати кількість таких випадків текст виводиться в текстове поле на формі програми.

Далі відбувається конвертація, результат якої видно на наступному зображенні:

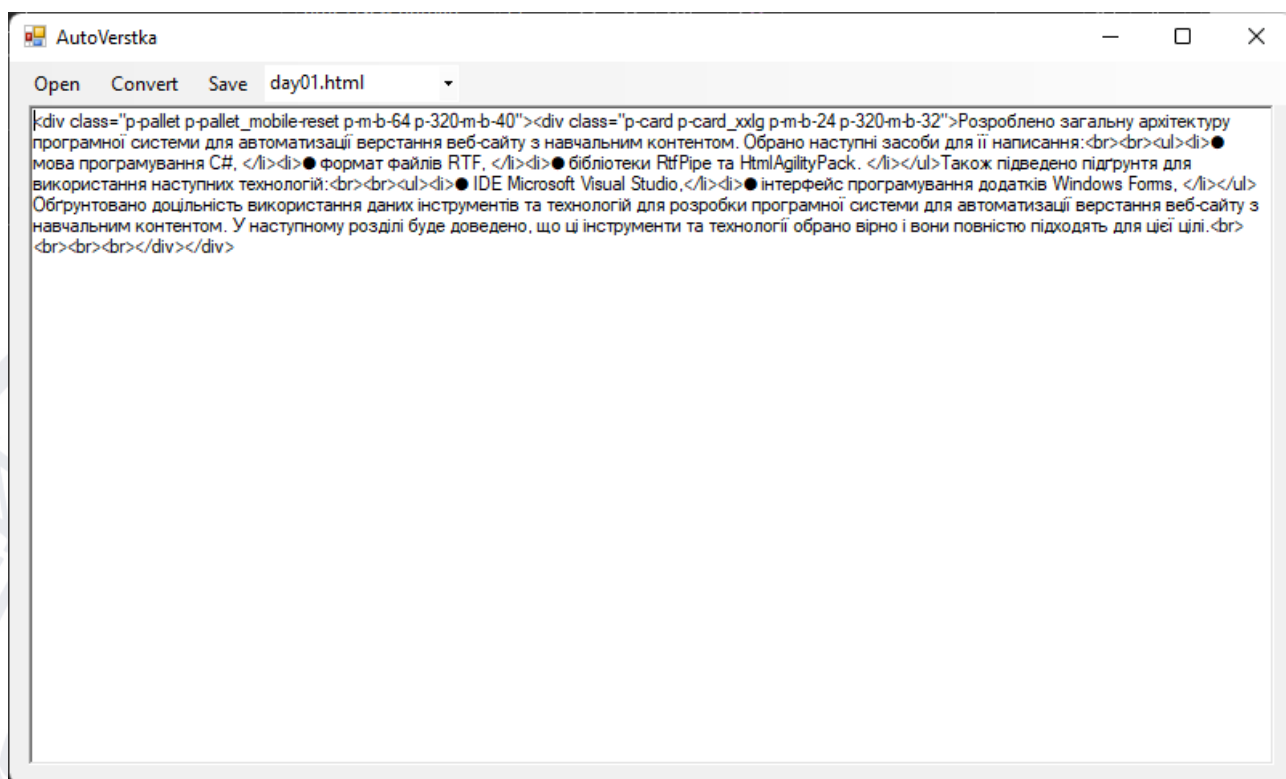


Рисунок 3.7 – результат конвертації, на прикладі частини з бакалаврської роботи.

HTML код, який є результатом конвертації мініфікований.

Після збереження результату, подальша робота з кодом, його доопрацювання та публікація лягає на верстальника.

```
<div class="p-pallet p-pallet_mobile-reset p-m-b-64 p-320-m-b-40">
  <div class="p-card p-card_xxl p-m-b-24 p-320-m-b-32">Розроблено загальну архітектуру програмної
  системи для автоматизації верстання веб-сайту з навчальним контентом.<br><br>Обрано наступні засоби
  для її написання:<br>
  <ul>
    <li>● мова програмування C#, </li>
    <li>● формат файлів RTF, </li>
    <li>● бібліотеки RtfPipe та HtmlAgilityPack. </li><br>
  </ul>Також підведено підґрунтя для використання наступних технологій:<br>
  <ul>
    <li>● IDE Microsoft Visual Studio,</li>
    <li>● інтерфейс програмування додатків Windows Forms, </li><br>
  </ul>Обґрунтовано доцільність використання даних інструментів та технологій для розробки
  програмної системи для автоматизації верстання веб-сайту з навчальним контентом. У
  наступному розділі буде доведено, що ці інструменти та технології обрано вірно і вони
  повністю підходять для цієї цілі.<br>
  </div>
</div>
```

Рисунок 3.8 – результат конвертації, форматований у редакторі VS Code.

Розроблено загальну архітектуру програмної системи для автоматизації верстання веб-сайту з навчальним контентом.

Обрано наступні засоби для її написання:

- мова програмування C#,
- формат файлів RTF,
- бібліотеки RtfPipe та HtmlAgilityPack.

Також підведено підґрунтя для використання наступних технологій:

- IDE Microsoft Visual Studio,
- інтерфейс програмування додатків Windows Forms,

Обґрунтовано доцільність використання даних інструментів та технологій для розробки програмної системи для автоматизації верстання веб-сайту з навчальним контентом. У наступному розділі буде доведено, що ці інструменти та технології обрано вірно і вони повністю підходять для цієї цілі.

Рисунок 3.9 – кінцевий результат застосування програми.

Кінцевий результат, зображений на рисунку вище, дає змогу побачити, що програма справляється з поставленою задачею і коректно відображає усі необхідні елементи.

Для розуміння кількісного приросту ефективності роботи, проведемо приблизний розрахунок. Для верстки цього простого фрагменту веб-сторінки в ручну необхідно: виконати операцію копіювання-вставки приблизно 35 рази (враховуючи кількість тегів 26 та окремих фрагментів тексту 9).

Натомість за допомогою програми, скопіювати-вставити достатньо буде лише близько 5 разів (з RTF документа в програму, з програми в HTML документ і ще 3 рази при внесенні правок). Отже в результаті ми отримуємо у 7 разів менше операцій, ніж при верстці ручним методом.

3.5 Перспективи

У розробленої програмної системи є перспективи у вигляді допрацювань:

- оптимізації швидкості роботи алгоритмів,
- опрацювання декількох файлів за один підхід,
- можливість змінювати налаштування, назви класів,
- додавання та видалення тегів,
- створення файлу налаштувань і можливість його резервного копіювання та відновлення,
- інші.

Висновок до розділу 3

У розділі розглянуто деталі реалізації та перспективи розвитку області автоматизації написання коду. З цією ціллю описано програмну реалізацію модулів цієї системи з урахуванням обраних у 2 розділі інструментів та технологій. Було опрацьовано та показано на прикладах роботу обох додатків, які призначені для автоматизації верстання веб-сайту з навчальним контентом. В конкретно взятому випадку проілюстровано семи кратний приріст продуктивності, що загалом не є обмеженням. Розкрито, як додатки написані та працюють, їх логічну структуру, опис вхідних та вихідних змінних, лістинг коду, приклади. Це допомогло на практиці побачити цінність даної роботи для майбутнього використання у поставленні таких сайтів на потік, адже скоротить час для написання коду.

ВИСНОВКИ

У рамках даної роботи була розглянута актуальність додатків для автоматизації верстки на сьогоднішній день, їх темпи розвитку. Особливості навчальних сайтів були враховано у практичній частині даної бакалаврської роботи такими засобами автоматичного верстання веб-сайтів: розпізнавання різних форматів тексту і застосування до них відповідних класів і стилів, формування нумерованих і не нумерованих списків, побудова таблиць, вставка зображень. Досягнуто цілі створення автоматичних програм для верстання сайтів з освітнім контентом. Дана бакалаврська робота наглядно показала, що додатки для автоматичної верстки є сучасним і актуальним інструментом для верстання навчальних сайтів та інших низькобюджетних веб-сайтів, що потребують марудної роботи верстальника. Було досягнуто наступних цілей:

- ❖ Проаналізовано об'єкт автоматизації.
- ❖ Здійснено аналіз конкурентних розробок.
- ❖ Розроблено додатки для автоматичного верстання навчальних сайтів.
- ❖ Проведено тестування системи.

Було виявлено наступні проблеми: верстка однотипних сторінок є монотонною і нудною, що знижує продуктивність, виснажує верстальника і найголовніше - потребує багато часу та реалізовано наступні шляхи їх вирішення: розробка інструменту, який допоможе значно зменшити час, що витрачається на роботу, а відповідно оптимізувати витрати людських ресурсів.

Досягнуто наступних функцій:

1. Читання з документу, або текстового поля. Також, текстове поле, що розміщено на формі здатне приймати RTF, тобто його можна буде просто вставляти туди з буфера обміну.
2. Перетворення змісту в такі елементи: нумеровані списки, нумеровані списки, заголовки, жирний текст, таблиці, відцентрований текст, зображення.

3. Додавання відповідних класів до елементів. Кожному доданому тегові автоматично присвоюється один з попередньо заданих класів, для зв'язку зі стилями.

4. Вивід тексту в поле, копіювання в буфер обміну або запис у документ. Створено потрібні кнопки та текстові поля.

Отримано наступні результати: розроблено програмну систему, що частково або повністю сприяє автоматизації верстання односторінкових сторінок цілком та їх елементів окремо. Усі поставлені завдання реалізовані повною мірою. Приріст ефективності для кожного окремого випадку різниться, але на прикладі проілюстровано семи кратний приріст, що без умовно є показником досягнення поставлених цілей. Висновок з усього вище переліченого - розроблена платформа є зручною для користування, легкою для розуміння і доступною для більшості майбутніх користувачів.

СПИСОК ЛІТЕРАТУРИ

1. Каплан Л. Е. Памятка верстальщику. — [2-е изд., доп.]. — М.: Гизлегпром, 1945. — 28 с. — (Библиотечка молодого полиграфиста).
2. Клименко Р.А. Веб-мастеринг на 100 % - [2-е изд] – С.-Пб.:Издательство «Питер», 2015. – 560 с.
3. Влад Мержевич. Основы верстки. htmlbook.ru
4. Веб-сайт. URL: <https://sites.google.com/view/bug-informatika/10-клас/веб-технології/проектування-та-верстка-веб-сторінок/проектування-та-верстка-веб-сторінок>
5. Федоренко Анастасія Юріївна (наук. керівник – д-р філол. наук, професор Лукаш Г. П.) Донецький національний університет імені Василя Стуса, м. Вінниця ПОНЯТТЯ, СТРУКТУРА ТА ОЗНАКИ ВЕБСАЙТУ. Інформаційні технології і системи в документознавчій сфері: Збірник матеріалів VI Всеукраїнської наукової студентської конференції(м.Вінниця, 02 квітня 2021 р.) Вінниця: ДонНУ імені Василя Стуса, 2021. С. 50-51.
6. Закон України «Про затвердження Порядку інформаційного наповнення та технічного забезпечення Єдиного веб-порталу органів виконавчої влади та Порядку функціонування веб-сайтів органів виконавчої влади». URL: <https://zakon.rada.gov.ua/laws/show/z1021-02#Text>
7. Гура М.В. Цивільно-правова охорона Інтернет-сайту в Україні: Автореф. дис. канд. юрид. наук. – К., 2006. – 20 с.
8. Структура веб-сайтів. URL: <http://www.ndu.edu.ua/liceum/web.pdf>
9. Веб-сайт. URL: <https://wezom.com.ua/blog/verstka>
10. Кириченко А.В., Дубовик Е.В., Никольский А.П. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков. – К., 2021
11. Веб-сайт. URL: <https://itchief.ru/bootstrap/introduction>

12. Sauer, C.: WYSIWIKI — Questioning WYSIWYG in the Internet Age. In: Wikimania (2006)
13. Spiesser, J., Kitchen, L.: Optimization of html automatically generated by WYSIWYG programs. In: 13th International Conference on World Wide Web, pp. 355—364. WWW '04. ACM, New York, NY (New York, NY, U.S., May 17-20, 2004)
14. Веб-сайт. URL: https://ru.wikipedia.org/wiki/Вёрстка_веб-страниц#Инструменты_верстальщика
15. Автоматическая верстка сайтов для тех, у кого мало времени. Design FactoRu - 11 июля 2017 года. Веб-сайт. URL: <http://www.coolwebmasters.com/web-services/3300-avtomaticheskaya-verstka-saytov-dlya-teh-u-kogo-malo-vremeni.html>
16. "Dictionary.com Unabridged (v 1.1)". Archived from the original on 30 November 2007. Retrieved 9 November 2007.
17. Рекомендації до сайтів навчальних закладів. - [Електронний ресурс]. - Веб-сайт. URL: <http://www.nmc.org.ua/metodicnirekomendacii/proorganizaciurobotizistvorenatafunkcionuvannavebsajtivnavcalnihzakladiv>
18. Боднар Л. В. Методичні рекомендації щодо створення Інтернет- сайту освітнього закладу / Л. В. Боднар. – Одеса, 2019. – с. 20-23.
19. Веб-сайт. URL: https://ru.wikipedia.org/wiki/Rich_Text_Format
20. Либерти Д. Язык программирования С#: пер. с англ. Санкт-Петербург. Символ-Плюс, 2003. 688 с.
21. Александр Шут, Верстальщик — кто это такой и почему от его работы зависит многое. Веб-сайт. URL: <https://zwebra.com.ua/verstalshhik-%E2%80%94kto-eto-takoj-i-pochemu-ot-ego-raboty-zavisit-mnogoe.html>
22. Веб-сайт. URL: <https://webdesign-trends.net/entry/13588>
23. Веб-сайт. URL: <https://chrome.google.com/webstore/detail/perfectpixel-by-welldonec/dkaagdgmdbnecmcefdhjekcoceebi?hl=ru>

24. Веб-сайт. URL: <https://www.codeguru.com/tools/7-best-c-sharp-ide/>

25. Веб-сайт.

URL:

<https://marketplace.visualstudio.com/items?itemName=NuGetTeam.NuGetPackageManager>



Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

Савченко Анатолій Андрійович

Прізвище, ім'я, по батькові

Факультет інформаційних і прикладних технологій

Факультет

122 Комп'ютерні науки

Шифр і назва спеціальності

Сучасні інформаційні технології та програмування

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «ПРОГРАМНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ВЕРСТАННЯ ВЕБ-САЙТУ З НАВЧАЛЬНИМ КОНТЕНТОМ» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис