



УДК 004.774

DOI: 10.20535/2077-7264.2 (72).2021.243775

© Г. С. Саюк, магістр, К. І. Золотухіна, канд. техн. наук, доц.,
КПІ ім. Ігоря Сікорського, Київ, Україна

ТЕСТУВАННЯ ВЕБ-РЕСУРСІВ ЗАЛЕЖНО ВІД ЇХ ФУНКЦІОНАЛЬНОГО ПРИЗНАЧЕННЯ

В роботі визначено фактори, що впливають на проектування та якість відтворення веб-ресурсів, систематизовано технології створення веб-ресурсів та види їх тестування, проведено експериментальне дослідження автоматизованого та ручного тестування, увиразнено процедуру тестування веб-ресурсів, розроблено тест-кейс для перевірки веб-сторінок, досліджено час, потрібний для розроблення односторінкових веб-ресурсів. Надано рекомендації з особливостей наповнення контентом веб-сторінок для найшвидшого їх відтворення на пристроях користувачів.

Ключові слова: веб-ресурси; ручне тестування; мова розмітки гіпертекстових документів (HTML); каскадні таблиці стилів (CSS); он-лайн ресурс; швидкодія; автоматизоване тестування; тест-кейс.

Постановка проблеми

З розвитком технологій, співпраця з читачами, клієнтами, користувачами здійснюється шляхом взаємодії в мережі Інтернет. Для цього створюються, розробляються різноманітні веб-ресурси. Такий спосіб зв'язку з цільовою аудиторією має чимало переваг порівняно з живим контактом, оскільки одночасно до веб-ресурсів може мати доступ чимала кількість користувачів, на відміну від, наприклад, відвідування офісу, магазину тощо, де кількість відвідувачів обмежується географічним розміщенням, площами приміщень та часом доби. Однак

недостатні якість такого веб-ресурсу та ступінь захищеності персональних даних можуть значною мірою зіпсувати враження від компанії або й стати причиною зменшення кількості потенційних клієнтів. Кількість та різновиди веб-ресурсів збільшується, однак більшість з них не відповідають критеріям якості. Для покращення цього показника тестування відіграє велику роль. А систематизація підходів тестування залежно від функціонального призначення веб-ресурсів допоможе удосконалити сам процес контролю якості й, відповідно, покращити кінцевий продукт.

© Автор(и) 2021. Видавець КПІ ім. Ігоря Сікорського.
CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>).



Аналіз попередніх досліджень

Аналіз технологій створення веб-ресурсів [1–8] дозволяє поділити їх на групи, а саме: стадія розробки дизайнів (макетів), створення мультимедійних складових, розробки функціональної частини та інтерфейсу й організація збереження даних. Технології створення веб-ресурсів сьогодні достатньо увиразнені, однак не систематизовані. В результаті аналізу наявних досліджень створено систематизацію технологій створення веб-ресурсів, представлену на рис. 1.

Вибір виду тестування залежить від різновидів веб-ресурсів та їх функціонального призначення. Зокрема, за ступенем автоматизації розрізняють ручне та автоматизоване тестування. Види тестувань поділяють на чотири основні групи: функціональне, нефункціональне, тестування пов'я-

зане зі змінами та тестування на етапі прийому та здачі проекту. На основі аналізу понад 50 інформаційних джерел, включаючи публікації в збірниках і журналах, що індексуються наукометричними базами, сайтів виробників і постачальників тощо, було розроблено класифікаційну схему всіх можливих видів тестування веб-ресурсів, наведену на рис. 2 [9–12]. Залежно від обраної технології розробки веб-ресурсу виникає потреба в різних програмних засобах.

Може бути достатньо лише одного браузеру, якщо розробляти за допомогою онлайн-конструкторів (Tilda, Wix, Бітрікс24 тощо). Для більш ґрунтовної розробки необхідно використовувати спеціальні програми, а саме: редактори коду (Visual Studio Code, Sublime Text), інтегровані системи розробки (PhpStorm), засоби для роботи з базами даних (MySQL

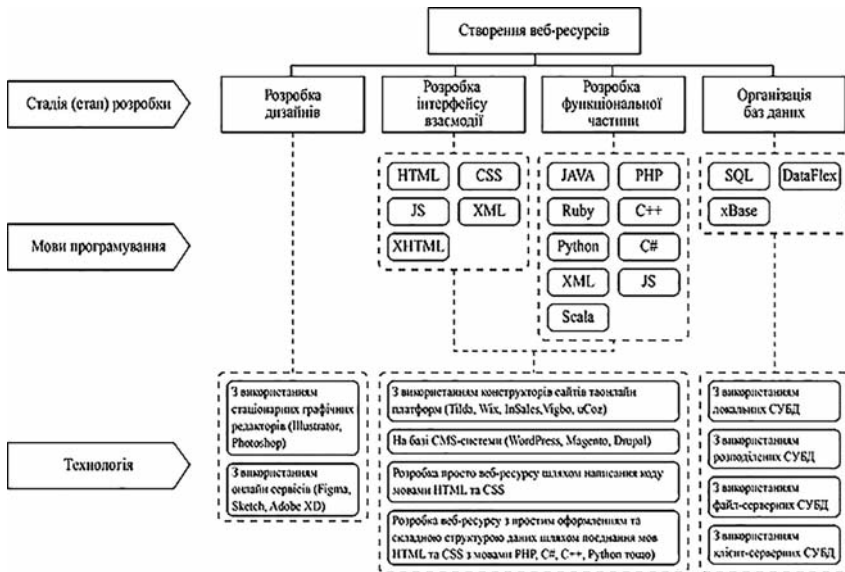


Рис. 1. Систематизація технологій створення веб-ресурсів



Workbench, PHPMyAdmin, dbForge Studio), браузери з підключеними плагінами, що допомагають в розробці тощо. Для підготовки інформаційних матеріалів також необхідні текстові та графічні редактори, програмні продукти для обробки аудіо та відео, програми для створення спеціальних ефектів тощо.

Мета роботи

Визначення параметрів та режимів, що впливають на створення веб-ресурсів та дослідженні їх тестування залежно від функціонального призначення.

Результати проведених досліджень

Об'єктом дослідження є технологічний процес розроблення та тестування веб-ресурсів, а саме: сайту-візитки та двох односторінкових веб-ресурсів. Предметом дослідження є режими та параметри створення та тестування веб-ресурсів. Серед режимів та параметрів тестування для дослідження обрано наступні: об'єм сторінки веб-ресурсу, кількість та розмір запитів до серверу, складність об'єктної моделі документа (DOM) та маніпуляції з нею, наявність та показники проблем, пов'язаних з JavaScript контентом, складність каскадної таблиці стилів (CSS) та проблеми, пов'язані з нею, характеристика шрифтів та показники швидкості завантаження сторінки. Обрано два види, що будуть досліджуватись детально, а саме сайти-візитки та односторінкові веб-ресурси. Для обраних видів розроблено структури типових сторінок. Для сайту-візитки розроблено шість сторінок, а саме: титульна

сторінка, сторінка з текстовою інформацією, сторінка з відео-інформацією, сторінка зі спливаючими вікнами та сторінка з контактною формою. Дослідження процесу тестування проведено для двох видів тестування, як ручного так і автоматизованого. За допомогою методів автоматизованого тестування тестові сторінки перевірялися за такими параметрами, як швидкість завантаження, як сторінки в цілому, так і окремих складових, кількість та швидкість отримання відповідей на запити до серверу, складність моделі документа, параметри налаштування сторінки тощо. Всі параметри з допустимими та/чи рекомендованими значеннями представлені в табл. 1. Цей вид тестування проведено за допомогою онлайн-сервісів PageSpeed Insights та Yellow Lab Tools [13, 14]. За допомогою ручного методів ручного тестування перевірялися контактні форми, спрацьовування гіперпосилань та кнопок, відтворення відео. Також за допомогою ручного тестування тестові сторінки перевірено на кросбраузерність та кросплатформеність, а також на адаптивність до різних розмірів екрану. Тестування здійснювалося за «Тест-кейсом», що створено та наведено нижче.

Перевірка проводилась для типових тестових сторінок. Показники першої групи мають сталі значення й не залежать від кількості проведених перевірок, а отже після того, як це було точно встановлено дослідним шляхом, було вирішено перевіряти сторінки по одному разу. Показники другої групи (швидкості завантаження) не є сталими, а тому кожна сторінка



Таблиця 1

Параметри автоматизованого тестування

Група показників	Показник	Допустимі/рекомендовані значення
Об'єм сторінки	Повний об'єм	до 2 Мб
	Оптимізація зображень	до 50 Кб
	Зображення надвеликого розміру	Ширина зображень більше 800px для мобільних девайсів та 1500px — для повноекранної версії
	Мінімізація файлів	до 20 Кб
Співвідношення запитів залежно від наповнення сторінки	Кількість запитів	до 60
	Кількість різних доменів	до 20
	Кількість посилань, що ведуть на неіснуючі сторінки (код помилки 404)	0
	Ідентичний контент	0
	Порожні запити	0
	Малі запити	Запити розміром до 2 Кб
	Приховані зображення	0
Складність об'єктної моделі документа (DOM) та маніпуляції з нею	Кількість DOM-елементів	до 1500
	Максимальна глибина моделі	до 15
	Дублікати ідентифікаторів	0
	Доступи до DOM моделі	до 1000
	Запити без результату	до 20
	Дублікати DOM запитів	до 20
Проблеми, пов'язані з JavaScript контентом	Помилки	0
	Синхронні Ajax запити	0
	Кількість коментарів в інтерфейсі розробників (консолі)	0
	Глобальні змінні	до 80
Параметри jQuery	Кількість завантажень бібліотеки jQuery	1
	Запити до порожніх об'єктів	до 40
	Неделеговані події	до 30



Закінчення табл. 1

Група показників	Показник	Допустимі/рекомендовані значення
Складність каскадної таблиці стилів (CSS) та проблеми, пов'язані з нею	Кількість правил	до 2500
	Складні селектори	до 10
	Кількість кольорів	до 80
	Подібні кольори	0
	Синтаксичні помилки	0
	Дублікати селекторів	до 10
	Дублікати властивостей	до 10
	Порожні правила	0
	Використання декларації «!important»	до 50
	Використання застарілих префіксів	до 50
	Надлишкові селектори тіла документа	0
	Надлишкові селектори тегів	до 15
Веб шрифти	Кількість шрифтів	1
	Шрифти надвеликого розміру	Сумарний розмір шрифту більше 40Кб
	Невикористовуваний діапазон Юнікоду	0
Налаштування серверу	Кількість з'єднань,, що не закриваються після відпрацювання запитів	0
	Відповіді з відімкненим кешуванням	до 5
	Кешування на занадто короткий час	до 10
Швидкість завантаження сторінки	Час завантаження першого контенту	до 3 с
	Індекс швидкості завантаження	до 3,5 с
	Час завантаження достатнього для взаємодії	до 4 с
	Час завантаження достатньої частини контенту	до 1,5 с
	Час закінчення роботи процесору	до 5 с
	Приблизний час затримки при відображенні	до 100 мс



перевірялась п'ять разів, потім знаходилось середнє значення з всіх вимірювань та заносилось до таблиці. Отримані результати тестування для сайту-візитки представлено в табл. 2.

На основі отриманих результатів швидкодії типових сторінок сайту-візитки складено результуючий графік для загального порівняння швидкодії різних сторінок, представлений на рис. 3.

Таким чином, можна зробити висновок, що всі показники окрім часу завантаження достатньої частини контенту знаходяться в допустимих межах. Сторінка з текстовою інформацією та титульна сторінка мають найкращі показники, а найгірше завантажується сторінка з відеоінформацією. Для пришвидшення часу завантаження слід максимально оптимізувати складові сторінки. А також можна застосувати технологію, за якої всі сили йдуть спочатку на завантаження основних складових,

а той матеріал, що йде нижче на сторінці підвантажується вже, коли користувач переглядає сторінку.

Сторінка з контактною формою містить додатковий функціонал, перевірку якого слід здійснювати згідно тест-кейсу, що розроблюється безпосередньо для цієї сторінки. Результат ручного тестування за тест-кейсом представлено в табл. 3.

Таким чином, після проходження ручної перевірки сторінки з контактною формою було виявлено три критичних недоліки: перевірка коректності введених значень є лише для поля електронної пошти; не відбувається перевірка коректності введених значень кожного разу при натисканні на кнопку «Відправити»; немає перевірки для формату файлу, що завантажується. Аналогічно було перевірено швидкодію для односторінкового веб-ресурсу. Для цього виду веб-ресурсів перевірялись

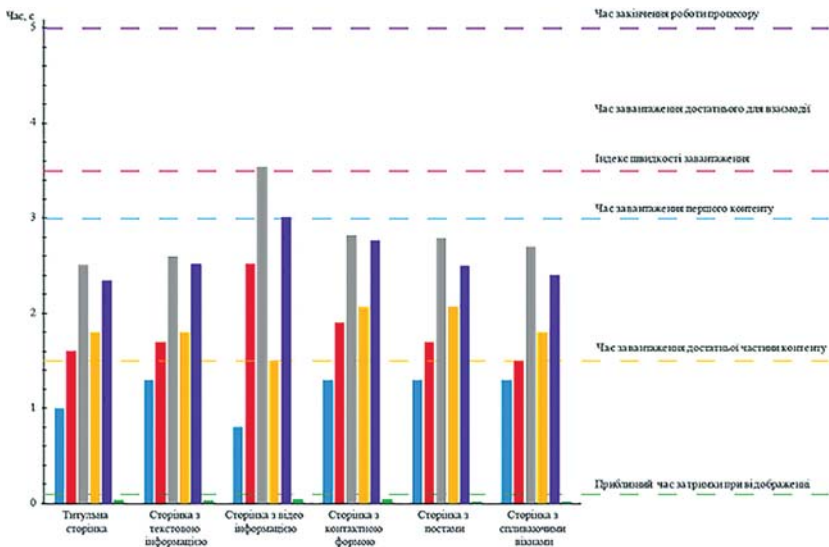


Рис. 3. Швидкість завантаження сторінок залежно від їх наповнення



Таблиця 2

Результати автоматизованого тестування

Група показників	Показник	Допустимі/ рекомендовані значення	Сторінки, що були перевірені						
			Титульна сторінка	Сторінка з текстовою інформацією	Сторінка з відео інформацією	Сторінка з контактною формою	Сторінка з постами	Сторінки з спливаючими вікнами	
	Повний об'єм	до 2 Мб	1,39 Мб	1,43 Мб	2,19 Мб	1,43 Мб	1,43 Мб	1,43 Мб	1,43 Мб
	Оптимізація зображень	до 50 Кб	344 Кб	121 Кб	203 Кб	121 Кб	121 Кб	121 Кб	121 Кб
	Зображення надвеликого розміру (ширина зображень більше 800px для мобільних девайсів та 1500px — для повноекранної версії)	0	0	0	0	0	0	0	0
	Мінімізація файлів	до 20 Кб	11,7 Кб	11,6 Кб	28,6 Кб	11,6 Кб	11,6 Кб	11,8 Кб	1,7 Кб
	Кількість запитів	до 60	53	48	86	49	48	48	49
	Кількість різних доменів	до 20	11	11	14	11	11	11	11
	Кількість посилань, що ведуть на неіснуючі сторінки (код помилки 404)	0	0	0	0	0	0	0	0
	Ідентичний контент	0	0	0	0	0	0	0	0
	Порожні запити	0	1	1	1	1	1	1	1
	Малі запити (запити розміром до 2 Кб)	до 10	9	9	17	10	9	9	10
	Приховані зображення	0	0	0	0	0	0	0	0



Продовження табл. 2

Група показників	Показник	Допустимі/ рекомендовані значення	Сторінки, що були перевірені					
			Титульна сторінка	Сторінка з текстовою інформацією	Сторінка з відео інформацією	Сторінка з контактною формою	Сторінка з постами	Сторінки з спливаючими вікнами
Складність об'єктної моделі документа (DOM) та маніпуляції з нею	Кількість DOM-елементів	до 1500	710	565	957	670	746	895
	Максимальна глибина моделі	до 15	13	13	16	19	20	20
	Дублікати ідентифікаторів	0	0	0	1	0	3	3
	Доступи до DOM моделі	до 1000	769	628	9882	773	796	1189
	Запити без результату	до 20	87	88	1065	89	86	95
Проблеми, пов'язані з JavaScript контентом	Дублікати DOM запитів	до 20	742	142	1468	148	147	158
	Помилки	0	0	0	0	0	0	0
	Синхронні Ajax запити	0	0	0	0	0	0	0
	Кількість коментарів в інтерфейсі розробників (консолі)	0	0	0	0	0	0	0
	Глобальні змінні	до 90	74	74	86	74	74	74
Параметри jQuery	Кількість завантажень бібліотеки jQuery	1	1	1	1	1	1	1
	Запити до порожніх об'єктів	до 40	36	37	37	35	33	38
	Неделеговані події	до 30	12	12	74	52	128	



Продовження табл. 2

Група показників	Показник	Допустимі/ рекомен- довані значення	Сторінки, що були перевірені					
			Титульна сторінка	Сторінка з тексто- вою ін- форма- цією	Сторінка з відео ін- формацією	Сторінка з контакт- ною формою	Сторінка з постами з постами	Сторінки з спливаю- чими вікнами
Складність каскад- ної таблиці стилів (CSS) та проблеми, пов'язані з нею	Кількість правил	до 2500	6088	5238	5494	5237	5239	5244
	Складні селектори	до 100	483	480	483	480	480	480
	Кількість кольорів	до 80	369	366	369	366	366	366
	Подібні кольори	0	98	96	98	96	96	96
	Синтаксичні помилки	0	0	0	0	0	0	0
	Дублікати селекторів	до 10	72	72	72	72	72	72
	Дублікати властивостей	до 10	31	31	31	31	31	31
	Порожні правила	0	0	0	0	0	0	0
	Використання декларації «!important»	до 50	452	399	440	398	400	404
	Використання застарілих префіксів	до 50	535	522	562	522	522	522
Надлишкові селектори тіла документа	0	97	97	97	97	97	97	
Надлишкові селектори тегів	до 15	7	7	7	7	7	7	



Закінчення табл. 2

Група показників	Показник	Допустими/ рекомендовані значення	Сторінки, що були перевірені					
			Титульна сторінка	Сторінка з текстовою інформацією	Сторінка з відео інформацією	Сторінка з контактною формою	Сторінка з постами	Сторінки з спливаючими вікнами
Веб шрифти	Кількість шрифтів	до 4	8	7	7	7	7	7
	Шрифти надвеликого розміру	до 60 Кб	735 Кб	679 Кб	679 Кб	679 Кб	679 Кб	679 Кб
	Невикористовуваний діапазон Юнікоду	0	27	26	26	26	26	26
Налаштування серверу	Кількість з'єднань, що не закриваються після відпрацювання запитів	0	0	0	0	0	0	0
	Відповіді з не специфікованим/відміченим кешуванням	до 5	0/1	0/1	8/1	0/1	0/1	0/1
	Кешування на занадто короткий час	до 10	7	7	11	7	7	7
Швидкість завантаження сторінки	Час завантаження першого контенту	до 3 с	1 с	1,42 с	0,84 с	1,4 с	1,4 с	1,4 с
	Індекс швидкості завантаження	до 3,5 с	1,7 с	1,78 с	2,14 с	1,78 с	1,76 с	1,76 с
	Час завантаження достатнього для взаємодії	до 4 с	2,52 с	2,62 с	3,38 с	2,84 с	2,82 с	2,78 с
	Час завантаження достатньої частини контенту	до 1,5 с	1,9 с	1,9 с	1,42 с	2,06 с	2,06 с	1,98 с
	Час закінчення роботи процесору	до 5 с	2,2 с	2,48 с	3 с	2,76 с	2,7 с	2,66 с
	Приблизний час затримки при відображенні	до 100 мс	40 мс	30 мс	50 мс	50 мс	23 мс	22 мс



однакові сторінки, створені за допомогою різних засобів. Таким чином, одну сторінку було створено на базі CMS системи WordPress, другу — за допомогою онлайн-платформи Tilda, а третю сторінку створено за допомогою конструктора Бітрікс24 [13, 14]. Отримані середні значення швидкостей наведено на рис. 4.

Таким чином, можна зробити висновок, що найбільш оптимізовані сторінки створюються за допомогою Tilda.cc, а найгіршим варіантом з досліджених є Бітрікс24. Однак, незважаючи на переваги в швидкодії, Tilda має ряд суттєвих недоліків, вона не є універсальною системою і призначена в більшій мірі саме для створен-

ня односторінкових веб-ресурсів, в той час, як WordPress є більш універсальною системою, яка дозволяє як створювати більш складні веб-ресурси, так і розміщувати на сторінці нестандартні, персоналізовані блоки, що, в свою чергу, забезпечує більшу варіативність отриманих веб-ресурсів. Щодо показників швидкодії, то за допомогою підключення спеціальних плагінів, проведення операцій з оптимізації інформації, можна досягти результатів навіть краще, чим у Tilda.

Досліджено час, що потрібний на розроблення односторінкових веб-ресурсів. За результатами досліджень побудовано циклограму, що наведена на рис. 5.

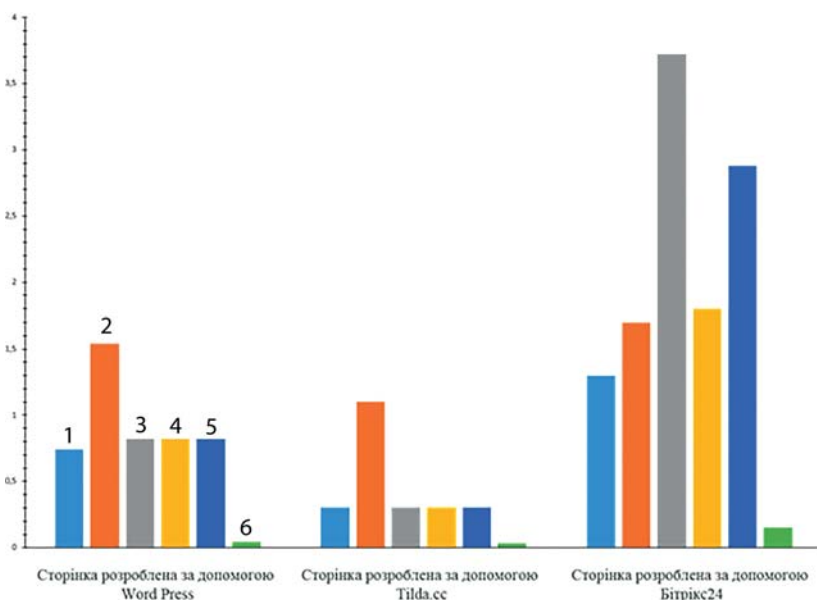


Рис. 4. Швидкість завантаження сторінок залежно від їх наповнення:

1 — час завантаження першого контенту, 2 — час завантаження достатньої частини контенту, 3 — час завантаження достатнього для взаємодії, 4 — індекс швидкості завантаження, 5 — час закінчення роботи процесору, 6 — приблизний час затримки при відображенні



Таблиця 3

Тест-кейс для перевірки сторінки з формою

№	Опис	Крок	Дані	Бажаний результат	Одержаний результат	Статус
1	Перевірка запуску сторінки	Відкрити сторінку	—	Сторінка запустилась	Сторінка запустилась	Пройдено
2	Перевірка складових елементів	Зверити з ТЗ наявність та розташування всіх елементів сторінки	—	Всі елементи відтворились на сторінці на своїх місцях	Всі елементи відтворились на сторінці на своїх місцях	Пройдено
3		Зверити з ТЗ, що відмічені всі обов'язкові поля форми	—	Всі обов'язкові поля мають відповідну відмітку	Всі обов'язкові поля мають відповідну відмітку	Пройдено
4	Перевірка валідності контактної форми	Заповнити контактну форму коректними значеннями, прикріпити файл в форматі .pdf та натиснути «Відправити»	ПІБ: Тест email: test123@gmail.com НК: Тест П: Тест	Форма відправилась, з'явилося повідомлення про вдалу відправку форми	Форма відправилась, з'явилося повідомлення про вдалу відправку форми	Пройдено
5		Спробувати відправити незаповнену форму	—	Поля виділились як незаповнені, форма не відправилась	Лише поле email відлились як незаповнені, форма не відправилась	Не пройдено
6	Перевірка валідності контактної форми	Заповнити поля пробілами	ПІБ: email: НК: П:	Поля виділились як незаповнені, форма не відправилась	Лише поле email відлились як незаповнені, форма не відправилась	Не пройдено
7		Заповнити лише обов'язкові поля	ПІБ: Тест email: test123@gmail.com	Форма відправилась, з'явилося повідомлення про вдалу відправку форми	Форма відправилась, з'явилося повідомлення про вдалу відправку форми	Пройдено



Закінчення табл. 3

№	Опис	Крок	Дані	Бажаний результат	Одержаний результат	Статус
8	Перевірка валідації контактної форми	Заповнити поля спец. символами, прикріпити файл в форматі .pdf та натиснути «Відправити»	ПІБ: !»№; /- =+%.*()#\$^&@ email: !»№; /- =+%.*()#\$^&@ НК: !»№; /- =+%.*()#\$^&@ П: !»№; /- =+%.*()#\$^&@	Всі поля, окрім «П» виділились як некоректно заповнені, форма не відправилась	Лише поле «email» виділилось як некоректно заповнені, форма не відправилась	Не пройдено
9		Поле «email» заповнити кириличними символами	ПІБ: Тест Тест email: йцукен НК: Тест П: Тест	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Пройдено
10		В поле «email» додати пробіл	ПІБ: Тест Тест email: test123@gmail.com НК: Тест П: Тест	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Пройдено
11	Перевірка валідації контактної форми	В поле «email» заповнити без «@»	ПІБ: Тест Тест email: test123gmail.com НК: Тест П: Тест	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Поле «email» виділилось як некоректно заповнене, форма не відправилась	Пройдено
12		Заповнити всі поля, окрім «email» коректно, після того, як з'явиться повідомлення про некоректно заповнене поле, видалити значення всіх полів та повторно заповнити форму так, щоб лише поле «email» було заповнено коректно, одне з обов'язкових полів не заповнювати	Спочатку: ПІБ: Тест Тест email: йцукен НК: Тест П: Тест Потім: ПІБ: Тест Тест email: test123@gmail.com НК: !»№; /- =+%.*()#\$^&@ П: !»№; /- =+%.*()#\$^&@	Відбулась повторна перевірка коректності заповнення форми, поля відмітились як незаповнені або некоректно заповнені, форма не відправилась	Повторна перевірка форми не відбулась, форма відправилась, з'явилось повідомлення про вдалу відправку форми	Не пройдено
13		Спробувати завантажити файл в не-допустимому форматі	—	Файл не зміг завантажитись, з'явилось повідомлення про невірний формат	Вдалося завантажити файл в форматі .doc	Не пройдено

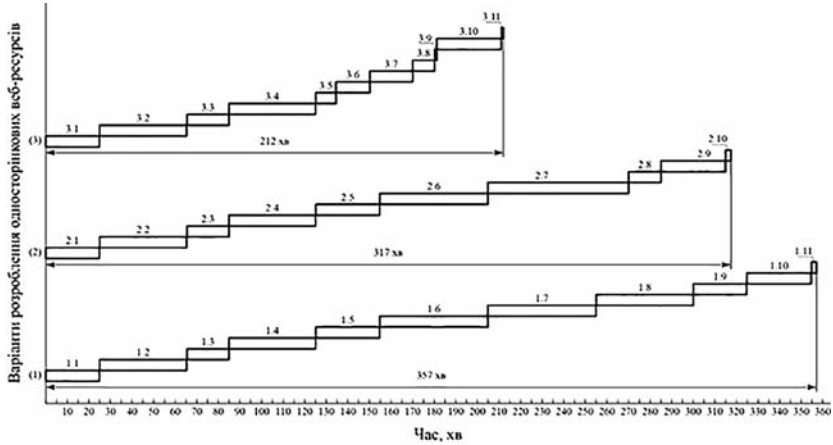


Рис. 5. Циклограми технологічних процесів розроблення односторінкових веб-ресурсів за допомогою технологій: написання коду використовуючи HTML, CSS та JS (1), використання CMS-систем (2) та використання онлайн-конструктор (3)

Пояснення до рис. 5:

1.1 — введення та опрацювання текстової інформації в текстовому редакторі Microsoft Word; 1.2 — опрацювання ілюстративного матеріалу Adobe Photoshop CC та Adobe Illustrator CC; 1.3 — опрацювання аудіо інформації; 1.4 — опрацювання відео інформації; 1.5 — створення розмітки html в програмі Visual Studio Code; 1.6 — додавання каскадних таблиць стилів (CSS) в програмі Visual Studio Code; 1.7 — створення інтерактивної складової шляхом написання коду мовою JavaScript (JS) в програмі Visual Studio Code; 1.8 — створення необхідного функціоналу шляхом написання коду мовою JavaScript (JS) в програмі Visual Studio Code; 1.9 — наповнення сторінок; 1.10 — тестування та, за потреби, редагування коду; 1.11 — завантаження на віддалений сервер.

2.1 — введення та опрацювання текстової інформації в тексто-

вому редакторі Microsoft Word; 2.2 — опрацювання ілюстративного матеріалу Adobe Photoshop CC та Adobe Illustrator CC; 2.3 — опрацювання аудіо інформації; 2.4 — опрацювання відео інформації; 2.5 — підбір теми в CMS системі WordPress; 2.6 — редагування теми за допомогою написання коду каскадних таблиць стилів (CSS) в програмі Visual Studio Code та імпорту до CMS системі WordPress; 2.7 — створення інтерактивної складової шляхом написання коду мовою JavaScript (JS) в програмі Visual Studio Code та імпорт коду CMS системі WordPress; 2.8 — наповнення сторінок; 2.9 — тестування та, за потреби, редагування коду; 2.10 — завантаження на віддалений сервер.

3.1 — введення та опрацювання текстової інформації в текстовому редакторі Microsoft Word; 3.2 — опрацювання ілюстративного матеріалу Adobe Photoshop



СС та Adobe Illustrator СС; 3.3 — опрацювання аудіо інформації; 3.4 — опрацювання відео інформації; 3.5 — вибір необхідного шаблону з бази онлайн-платформи; 3.6 — створення розмітки типової сторінки за допомогою модулів; 3.7 — корегування стилів; 3.8 — наповнення сторінок; 3.9 — тестова публікація; 3.10 — тестування та, за потреби, редагування веб-ресурсу; 3.11 — остаточна публікація сторінок.

Таким чином онлайн-конструктор дозволяє значно пришвидшити процес створення веб-сторінок.

Висновки

1. Для пришвидшення часу завантаження рекомендується максимально оптимізувати складові сторінки, застосувавши технологію, за якої всі сили йдуть спочатку на завантаження основних складових, а той матеріал, що йде нижче на сторінці підвантажується вже, коли користувач переглядає сторінку.

Список використаної літератури

1. Куликов С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов. Минск: Четыре четверти, 2017. 312 с.

2. 10 тенденций тестирования программного обеспечения, на которые стоит обратить внимание в 2019 году [Електронний ресурс]. Режим доступу: <https://habr.com/ru/company/otus/blog/436942/>, вільний. Загл. з екрану. Мова рос.

3. Тестирование веб-ориентированных приложений. Часть 1: инсталляционное тестирование [Електронний ресурс]. Режим доступу: https://svyatoslav.biz/technologies/web_testing_installation/, вільний. Загл. з екрану. Мова рос.

4. Тестирование веб-ориентированных приложений. Часть 2: тестирование форм [Електронний ресурс]. Режим доступу: https://svyatoslav.biz/sft/web_testing_forms/, вільний. Загл. з екрану. Мова рос.

5. Тестирование веб-ориентированных приложений. Часть 3: ссылки [Електронний ресурс]. Режим доступу: https://svyatoslav.biz/sft/web_testing_links/, вільний. Загл. з екрану. Мова рос.

6. Тестирование веб-ориентированных приложений. Часть 4: безопасность в PHP [Електронний ресурс]. Режим доступу: https://svyatoslav.biz/sft/php_security/, вільний. Загл. з екрану. Мова рос.

2. Найбільш оптимізовані сторінки створюються за допомогою Tilda.cc, а найгіршим варіантом, з досліджених, є Бітрікс24. Tilda.cc має переваги в швидкодії, проте й має ряд недоліків. Tilda.cc не є універсальною системою і призначена в більшій мірі саме для створення односторінкових веб-ресурсів. WordPress є більш універсальною системою, яка дозволяє як створювати більш складні веб-ресурси, так і створювати та розміщувати на сторінці нестандартні, персоналізовані блоки, що, в свою чергу, забезпечує більшу варіативність отриманих веб-ресурсів. Щодо показників швидкодії, то за допомогою підключення спеціальних плагінів, проведення операцій з оптимізації інформації, можна досягти результатів навіть краще, ніж у Tilda.

3. Для економії часу, створення веб-сторінок доцільно виконувати із застосуванням он-лайн-конструкторів.



7. Какие технологии используются при создании сайтов? [Электронный ресурс]. Режим доступа: <http://uhdesign.com.ua/faq/3.html>, вільний. Загл. з екрану. Мова рос.

8. Технология создания Веб-сайта [Электронный ресурс]. Режим доступа: <https://poznayka.org/s75172t1.html>, вільний. Загл. з екрану. Мова рос.

9. Как тестировать веб-сайт [Электронный ресурс]. Режим доступа: <https://geteasyqa.com/ru/qa/test-website/>, вільний. Загл. з екрану. Мова рос.

10. Этапы тестирования веб-сайтов [Электронный ресурс]. Режим доступа: <https://art-lemon.com/site-test>, вільний. Загл. з екрану. Мова рос.

11. Types of software testing [Электронный ресурс]. Режим доступа: <https://geteasyqa.com/qa/software-testing-types/>, вільний. Загл. з екрану. Мова англ.

12. Тестирование программного обеспечения — основные понятия и определения [Электронный ресурс]. Режим доступа: <http://www.protesting.ru/testing/>, вільний. Загл. з екрану. Мова рос.

13. PageSpeed Insights [Электронный ресурс]. Режим доступа: <https://developers.google.com/speed/pagespeed/insights/>, вільний. Загл. з екрану. Мова рос.

14. Yellow Lab Tools [Электронный ресурс]. Режим доступа: <https://yellowlab.tools/>, вільний. Загл. з екрану. Мова англ.

References

1. Kulikov, S. S. (2017). *Testirovanie programmnoho obespecheniya. Bazyvyy kurs [Software Testing. Basic Course]*. Minsk: Chetyre chetverti, 312 p. [in Russian].

2. *10 tendentsiy testirovaniya programmnoho obespecheniya, na kotorye stoit obratit' vnimanie v 2019 godu [10 Software Testing Trends to Look Out for in 2019]*. Retrieved from <https://habr.com/ru/company/otus/blog/436942/> [in Russian].

3. *Testirovanie veb-orientirovannykh prilozheniy. Chast' 1: installatsionnoe testirovanie [Testing Web-Oriented Applications. Part 1: Installation Testing]*. Retrieved from https://svyatoslav.biz/technologies/web_testing_installation [in Russian].

4. *Testirovanie veb-orientirovannykh prilozheniy. Chast' 2: testirovanie form [Testing Web-Oriented Applications. Part 2: Testing Forms]*. Retrieved from https://svyatoslav.biz/sft/web_testing_forms/ [in Russian].

5. *Testirovanie veb-orientirovannykh prilozheniy. Chast' 3: ssylki [Testing Web-Oriented Applications. Part 3: Links]*. Retrieved from https://svyatoslav.biz/sft/web_testing_links/ [in Russian].

6. *Testirovanie veb-orientirovannykh prilozheniy. Chast' 4: bezopasnost' v PHP [Testing Web-Oriented Applications. Part 4: Security in PHP]*. Retrieved from https://svyatoslav.biz/sft/php_security/ [in Russian].

7. *Kakie tekhnologii ispol'zuyutsya pri sozdanii saytov? [What Technologies are Used to Create Websites?]*. Retrieved from <https://uhdesign.com.ua/faq/3.html> [in Russian].

8. *Tekhnologiya sozdaniya Veb-sayta [Website Creation Technology]*. Retrieved from <http://poznayka.org/s75172t1.html> [in Russian].

9. *Kak testirovat' veb-sayt [How to Test a Website]*. Retrieved from <http://geteasyqa.com/ru/qa/test-website/> [in Russian].

10. *Etapy testirovaniya veb-saytov [Website Testing Stages]*. Retrieved from <https://art-lemon.com/site-test> [in Russian].



11. *Types of software testing*. Retrieved from <https://geteasyqa.com/qa/software-testing-types/> [in English].
12. *Testirovanie programmogo obespecheniya – osnovnye ponyatiya i opredeleniya [Software Testing – Basic Concepts and Definitions]*. Retrieved from <http://www.protesting.ru/testing/> [in Russian].
13. *PageSpeed Insights*. Retrieved from <http://developers.google.com/speed/pagespeed/insights/> [in Russian].
14. *Yellow Lab Tools*. Retrieved from <http://yellowlab.tools/> [in English].

The paper identifies the factors influencing the design and reproduction of web resources, systematizes technologies for creating web resources and types of testing, conducted an experimental study of automated and manual testing. Typical page structures have been developed for selected species. The procedure for testing web resources was clarified, a test case to verify web pages was developed, the time required to develop one-page web resources was explored. Recommendations on the features of filling the content of web pages for the fastest playback on users' devices were developed.

Keywords: web resources; manual testing; hypertext markup language (HTML); cascading style sheets (CSS); online resource; performance' automated testing; test case.

Надійшла до редакції 02.04.21