

2 ОБГРУНТУВАННЯ ПРОГРАМНИХ ЗАСОБІВ, АЛГОРИТМІВ ТА МЕТОДІВ ПРОГРАМНОЇ РОЗРОБКИ

2.1 Обрання засобів розробки

Для розробки конфігуратора товарів було обрано такі технології і платформи:

- JavaScript
- HTML
- CSS
- Tilda
- Wordpress

Для тестування і збору аналітики про ефективність і роботи конфігуратора, були обрані наступні сервіси.

- Google Analytics
- Yandex Metrika
- Hotjar

Далі будуть детально розглянута і аргументірована кожна технологія і сервіс, для розробки дипломного проекту.

2.1.1 Технології розробки

Для розробки Front-end частині сайту будуть використані HTML і CSS. В сукупності, HTML і CSS дозволяє реалізувати адаптивний і кросбраузерності веб інтерфейс[4][5], а також візуалізувати дані, одержувані до клієнтської частини сайту.

Для реалізації логічних операцій, обробки запитів користувача і реалізації веб компонентів буде використовуватися мова програмування JavaScript. Завдяки JavaScript також будуть реалізовані деякі компоненти Back-End сайту, які будуть оброблятися серверною частиною сайту.

2.1.1.1 HTML.

HTML (HyperText Markup Language) - стандартна мова розмітки гіпертекстових сторінок в Інтернеті. Є й інші мови розмітки гіпертексту, але велика частина сторінок сайтів Інтернету розмічена саме на мові HTML[5]. Такі сторінки успішно інтерпретуються браузерами, які відображають їх на екранах різних електронних пристроїв в зручному для людини вигляді.

2.1.1.2 CSS

Це мова стилів, який визначає відображення HTML-документів. CSS працює зі шрифтами, з квітами символів і фону, з полями, з рядками, з висотою і з шириною елементів відображення, з фоновими зображеннями, з позиціонуванням елементів і багато з чим іншим[5]... Використання CSS полегшує створення якісних сайтів, дозволяючи задати стилі окремих елементів сторінок сайту в особливих css-файлах, щоб в подальшому бути впевненим в тому, що всі сторінки сайту будуть витримані в єдиному стилі.

2.1.1.3 JavaScript

JavaScript - один з найпопулярніших мов програмування в світі з більш ніж двадцятирічної історією. Це мову з великими перевагами, які роблять його найкращим вибором серед подібних йому[10]. Всього кілька переваг використання JavaScript:

- Вам не потрібен компілятор, тому що веб-браузер інтерпретує його в HTML;
- Його простіше вивчати, ніж інші мови програмування; Помилки простіше виявити, а значить і справити;
- Він може прив'язуватися до спеціальних елементів сторінок або події на зразок натискання (click) або наведення миші (mouseover) ;
- JS працює в різних браузерах і на різних платформах;
- Ви можете використовувати JavaScript для валідації вхідних даних і зниження необхідності ручної перевірки даних;

- Він робить сайт більш інтерактивним і привабливим для відвідувачів;
- Він швидше і легше, ніж інші мови програмування.

2.1.2 Вибір платформи

Для вибору платформи, під яку буде розроблено конфігуратор товарів, стоїть вибір між двома популярними аналогами - Wordpress і Tilda. Це обумовлено тим, що обидві платформи мають ряд переваг і недоліків[10]. Для розробки дипломного проекту, формально можна використовувати будь-яку. Але для якісного виконання проекту, слід оцінити яка платформа буде більш підходящою.

2.1.2.1 Wordpress

На сьогодні Wordpress одна з найпопулярніших систем управління контентом (Content Management System). Зазвичай, ця CMS використовується для створення блогу, але сайт на WordPress може бути легко перетворений в інтернет магазин, портфоліо, або корпоративний сайт компанії. Під платформу розроблена величезна кількість плагінів, проте їх впровадження неможливе без додаткових знань в області веб розробки. Для створення конфігуратора, вимагає поглиблене розуміння платформи, робота з API, а також додаткове тестування і налагодження. Складність і тривалість розробки конфігуратора під дану платформу НЕ являється ефективним з точки зору витраченого часу і ресурсів.

2.1.2.2 Tilda

Тільда є аналогом Wordpress і дозволяє створити сайт або інтернет-магазин без програмування. За останні кілька років, Tilda стрімко наберать популярність серед веб разработкіков і малого бізнесу. Це обумовлено легкістю створення сайтів електронної комерції, простий інтеграцією з системами облати, можливістю одночасно швидко створити інтернет-магазин або LandingPage. Творці платформи придумали новий спосіб зробити сайт: його

можна збирати з блоків. Можна не мати навички програміста і дизайнера, помилитися складно. Сайт виходить красивим і з перших днів приносить користь.

Швидкість і простота реалізації, являється ключовим перевагою при виборі платформи під розробку дипломного проекту. Конфігуратор товарів буде розроблений під Tilda, враховуючи всі її переваги перед поставленими завданнями.

2.1.3 Вибір системи аналітики та тестування

Для збору даних і тестування на ринку є кілька популярних технології, які відзначаються функціоналом, простотою впровадження, а також наявністю можливості безкоштовного використання[14]. Вибір системи аналітики і тестування дипломного проекту лежить між Google Analytics, Yandex Metrika і Hotjar.

2.1.3.1 Google Analytics

Google Analytics (GA) - сервіс від компанії Google, призначений для вебмайстрів та оптимізаторів, який дозволяє аналізувати поведінку користувачів на сайті. Зібрана інформація розміщується на віддаленому сервері від Google. Все, що потрібно для підключення до системи - це встановити на сторінках сайту невеликий код JavaScript. Статистика збирається за допомогою лічильника Гугл Аналітікс на сервері компанії. На всіх сторінках веб-ресурсу розміщується так званий код-лічильника JavaScript. Переваги даного сервісу заключається в легкості установки, а також великим набором функцій і можливостей, а також наявністю безкоштовного тарифу з мінімумом обмежень. В якості основної системи аналітики та тестування для дипломного проекту буде використаний цей сервіс.

2.1.3.2 Yandex Metrika

Yandex Metrika - безкоштовний інтернет-сервіс компанії Яндекс, призначений для оцінки відвідуваності веб-сайтів і аналізу поведінки користувачів. На даний момент Яндекс.Метрика є третьою за розміром системою веб-аналітики в Європі. Лічильник «Яндекс.Метрика» працює за принципом звичайного лічильника відвідувань: JS-код - встановлюється веб-майстром на сторінках сайту і збирає дані про кожному відвідуванні. Лічильник Яндекс.Метрики також сумісний з AJAX- і Flash-сайтами. Переваги даного сервісу аналогічні Google Analytics. Також Yandex Metrika має можливість запису сеансів користувачів і Вебвізор. Недоліком даного сервісу є санкції в Україні, через що сервіс некоректно працює з великою кількістю даних. Однак Вебвізор і запис сеансів коректно працює, що дозволяє використовувати Yandex Metrika в якості додаткового сервісу аналітики і тестування.

2.1.3.3 Hotjar

Hotjar - це онлайн-сервіс для аналітики сайту і накопичення зворотного зв'язку. Він покликаний допомогти вникнути в потреби відвідувачів і збільшити конверсію. Система підходить для UX і UI дизайнерів, розробників, менеджерів, аналітиків, маркетологів і ін. Незважаючи на переваги і доступність даного сервісу, у нього відсутня безкоштовний тариф дозволяє повноцінно працювати з сервісів. З комерційної точки зору і доцільності, даний сервіс не буде використовуватися в дипломному проекті.

Кожен вищезгаданий сервіс, технологія і платформа по своєму гарні, проте переваги і недоліки визначають під які завдання їх слід використовувати. У дипломній роботі будуть використані технології JavaScript, HTML, CSS для розробки Front-End і Back-End конфігуратора товарів. Конфігуратор товарів буде розроблений під платформу Tilda. В якості системи аналітики і тестування будуть використані Google Analytics і Yandex Metrika. Google Analytics, дозволить збирати кількісні дані про роботу дипломного проекту. Yandex Metrika, дозволить отримувати записи сеансів користувачів, а також будувати

теплові карти, які дозволять проаналізувати поведінку користувачів і надалі поліпшити і розвивати дипломний проект.

2.2 Огляд середовищ розробки

Для розробки на мові JavaScript, HTML і CSS можна використовувати звичайний текстовий редактор, оскільки код не вимагає компіляції. Для швидкої і зручної роботи, використовуються більш просунуті текстові редактори і середовища розробки. Далі будуть описані найбільш популярні з них на сьогоднішній день. Також слід зазначити, що всі згадані середовища розробки є повністю крос платформеними.

Основними середовищами розробки є:

- Microsoft Visual Studio
- Brackets
- Atom
- Sublime text 3

2.2.1 Microsoft Visual Studio

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному[6], так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight. Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні. Однак незважаючи на всі

переваги Microsoft Visual Studio є платним інструментом, а також має складний інтерфейс і вимагає додаткової настройки при установці.

2.2.2 Brackets

Brackets вдає із себе редактор з відкритим вихідним кодом. Призначений для створення веб-додатків. Це редактор коду загального призначення і призначений для людей, які володіють мовами програмування. Має безліч розширень, які можна завантажити з сайту програми. Як заявляють творці Brackets, їх мета - створити кращий редактор коду для JavaScript, HTML і CSS і пов'язаних з ними відкритих веб-технологій. Редактор написаний на JavaScript, HTML і CSS. Це означає, що більшість користувачів Brackets, які мають навички, можуть зробити доопрацювання і розширення редактора. Brackets не є стабільним редактором, т.к встановлення додатків для оптимізації робочого процесу, може нестабільно впливати на роботу редактора. Незважаючи на численні переваги, нестабільність роботи, може згубно вплинути на роботу над дипломним проектом.

2.2.3 Atom

Atom (в минулому Atomicity) - безкоштовний текстовий редактор з відкритим вихідним кодом для Linux, macOS, Windows з підтримкою плагінів, написаних на Node.js, і вбудованих під керуванням Git. Більшість полігонів мають статус вільного програмного забезпечення, розробляються і підтримуються спільнотою.

2.2.4 Sublime text 3

Sublime Text - пропрієтарний текстовий редактор, який підтримує безліч плагінів і доповнень. Для зручності роботи, плагіни і доповнення можуть розширити можливості даного редактора, що являється відчутною перевагою. Розробник дозволяє безкоштовно і без обмежень ознайомитися з продуктом і використовувати його в роботі. Важливими особливостями програми можна

виділити зручний і налаштовує інтерфейс, можливість виділення стоблцов і множинну правку, автодоповнення, а також підтримку систем збирання. Для розробки дипломного проекту, буде використана цей інструмент через зручності, швидкості роботи, доступності та гнучкості в налаштуванні.

2.3 Обрання методу проектування архітектури системи

Архітектура майбутнього застосунку буде поєднувати у собі принципи MVC (Model-View-Controller) та багатоваршівної архітектури (multilayered architecture), що підвищує ефективність їх роботи через велику модульність проекту, а тому більшу можливість тестування програмного додатку і зменшення кількості помилок у програмному коді.

Для розробки програмного коду також було обрано використання принципів об'єктно-орієнтованого програмування (ООП), що дозволяють створити систему із високим рівнем абстракції від конкретної реалізації, що розширює можливості розробки та вдосконалення системи у майбутньому, а також дозволяє повторне використання написаного коду, як бази у наступних розробках.

2.4 Обрання алгоритмів та моделей даних, використовуваних у розробці

Для гарного та безвідмовного функціонування та швидкої взаємодії із цільовим користувачем, багатифункціональна система повинна використовувати велику кількість алгоритмів. У розроблюваному програмному застосунку будуть використані наступні алгоритми:

- алгоритм багатопотокової обробки даних – потрібний для швидкого реагування на події користувача та обробки і видачі результатів роботи застосунку на екран;
- алгоритми кешування даних – використовуються для оптимізації швидкодії застосунку;
- алгоритм хешування – застосовуються для захисту користувацьких

даних.

Та багато інших алгоритмів, наприклад алгоритми відправки нотіфікацій користувачеві, синхронізації потоків, які забезпечують гарне функціонування системи.

2.4.1 Алгоритми багатопотокової обробки даних

Всі моделі та алгоритми обчислювачів потоків даних мають ту загальну особливість, що їхні компоненти – вузли, оператори, актори або процеси – з'єднані між собою потоками даних. Часто потік даних називають сигналом. В загальному випадку, потік даних – це засіб передачі скінченної або в граничному випадку, нескінченної кількості впорядкованих даних між компонентами обчислювальної моделі.

Обчислювальні моделі відрізняються між собою способом виконання потоків даних. Спосіб такого виконання залежить від певних властивостей або ознак потоку, тобто від того:

- чи потік однонаправлений, чи двонаправлений;
- чи у потоку одне джерело даних, чи кілька джерел;
- чи з потоку зчитує дані один процес, чи кілька процесів;
- як в потоці дані відрізняються за часом виникнення, споживання;
- чи дозволяється в потоці дублювати дані або їх перезаписувати;
- якщо потік є чергою, чи дозволяється перевірка наявності даних в потоці, чи ні;
- чи джерело й приймач даного звертаються до потоку одночасно.

Останні дві приведені вище властивості потоку даних характеризують його існування в часі. Розрізняють синхронні й асинхронні потоки та відповідно – синхронні й асинхронні моделі обчислень. В асинхронному потоці момент передачі даного є довільним і він допускає довільну затримку між завантаженням даного в потік і його споживанням. Це дає можливість будь-яким чином вибирати моменти завантаження й споживання даних або

спрацьовування відповідних акторів при складанні розкладу виконання алгоритму.

Граф потоків даних (ГПД) представляє собою натуральну модель для задавання алгоритмів обробки таких потоків. Динамічні ГПД мають найширші можливості для задавання алгоритмів. Через це вони застосовуються, в основному, тільки в системах з динамічним розкладом як у нашому випадку.

2.4.2 Алгоритми кешування даних

Швидке збільшення темпів використання в сучасному світі новітніх інформаційних технологій вимагає розробку ефективних технологій, які забезпечують зберігання, аналіз і обробку значних масивів різноманітної інформації.

Великі об'єми даних неможливо одночасно розмістити в оперативній та кеш-пам'яті, тому більша частина даних під час роботи системи залишатиметься на зовнішніх носіях, що зводить продуктивність системи до мінімуму. Практично єдиним ефективним способом зменшення кількості звертань до зовнішніх носіїв є механізми кешування даних, реалізовані кешконтролером, основною задачею якого є прогнозування звернень процесора до пам'яті.

На теперішній час існує велика кількість алгоритмів кешування, основними з яких є наступні:

- **LRU.** В даному алгоритмі витісняються елементи, які довше за інші не використовувались. Для реалізації цього алгоритму використовується лічильник «віку» – числове значення, яке показує на час останнього звернення до елементу. Алгоритм LRU простий в реалізації, але, очевидно, що його можливості прогнозування досить обмежені;
- **MRU.** Алгоритм протилежний до LRU, так як витісняються елементи до яких відбулось останнє звернення. Цей алгоритм інколи є більш ефективним за LRU, наприклад у випадках циклічного сканування великих наборів даних;

- **LFU**. Алгоритм, заснований на підрахунку числа звернень до кожного елемента, при якому витісняються елементи, які використовуються рідше за інші. Для реалізації потрібно збереження частоти звертань до кешу;

Для обробки даних, а також ефективного кешування, обран алгоритм MRU. Оскільки специфіка конфігуратора товарів полягає у том, що при великих обсягах даних, та їх циклічної обробки, нам необхідно мати високу швидкість та стабільність роботи.

2.4.3 Алгоритми хешування даних

Оскільки розроблюваний додаток має можливості багатокористувацької системи, то постає питання створення захисного механізму. Для цього користувач буде створювати свій особистий профіль, який буде охоронятися паролем. Щоб паролі були більш захищені, було вирішено використати метод хешування.

Серед відомих методів хешування є:

- MD5;
- SHA-0;
- SHA-1;
- SHA-2;
- SHA-256.

MD5 та SHA-0 є застарілими та майже не використовуються через свої вразливості. SHA-2 використовується у проектах, що повинні мати дуже якісний захист, SHA-256 застосовується виключно у проектах із великим рівнем захисту та є дуже повільним у використанні. Оптимальним для нашої задачі є SHA-1, виходячи із швидкості хешування та рівня захисту у розроблюваному програмного забезпеченні.

2.4.4 Інші алгоритми

Даний перелік алгоритмів не є повним. Також потрібними є алгоритми кешування даних, багатопотокової обробки даних та синхронізації потоків, а також багато інших. Однак, їх важливість є меншою за вищевказані. Також вони розроблювались як допоміжні, тому не заслуговують більш детального розбору.

2.5 Огляд та обрання методології розробки та тестування

На сьогоднішній день використовується багато методологій для розробки програмного продукту та його тестування. Серед популярних є наступні: Scrum, Cleanroom Software Engineering, RAD (Rapid application development), DSDM (Dynamic Systems Development Method,) та TDD (test-driven development).

2.5.1 Scrum

Scrum – це набір принципів, на яких будується процес розробки, що дозволяє в жорстко фіксовані і невеликі за часом ітерації, звані спринти (sprints), надавати кінцевому користувачеві працює ПО з новими можливостями, для яких визначено найбільший пріоритет. Можливості ПО до реалізації в черговому спринті визначаються на початку спринту на етапі планування і не можуть змінюватися на всьому його протязі. При цьому строго фіксована невелика тривалість спринту надає процесу розробки передбачуваність і гнучкість.

2.5.2 Cleanroom Software Engineering

Cleanroom Software Engineering (методологія «чистої кімнати») – процес розробки програмного забезпечення, призначений для створення програмного забезпечення з сертифікується рівнем надійності. Cleanroom був спочатку розроблений Харланом Міллзом і декількома його колегами, в тому числі Аланом Хевнером з IBM. Основний принцип cleanroom полягає в тому, що попередження дефектів краще, ніж їх усунення. Назва Cleanroom («чиста

кімната») взято з електронної промисловості – так називаються приміщення з високим ступенем захисту від забруднень, що дозволяють запобігти появі дефектів у процесі виробництва напівпровідників. Вперше процес був застосований в середині-кінці 80-х років.

2.5.3 Rapid Application Development

RAD (від англ. Rapid application development – швидка розробка додатків) – концепція створення засобів розробки програмних продуктів, що приділяє особливу увагу швидкості і зручності програмування, створення технологічного процесу, що дозволяє програмісту максимально швидко створювати комп'ютерні програми.

RAD передбачає, що розробка ПО здійснюється невеликою командою розробників за термін близько трьох-чотирьох місяців шляхом використання інкрементного прототипування із застосуванням інструментальних засобів візуального моделювання та розробки. Технологія RAD передбачає активне залучення замовника вже на ранніх стадіях – обстеження організації, вироблення вимог до системи. Останнє із зазначених властивостей передбачає повне виконання вимог замовника як функціональних, так і не функціональних, з урахуванням їх можливих змін в період розробки системи, а також отримання якісної документації, що забезпечує зручність експлуатації і супроводу системи. Це означає, що додаткові витрати на супровід відразу після поставки будуть значно менше. Таким чином, повний час від початку розробки до отримання прийнятного продукту при використанні цього методу значно скорочується.

2.5.4 DSDM

Метод розробки динамічних систем (Dynamic Systems Development Method, DSDM) – це головним чином методика розробки програмного забезпечення, заснована на концепції швидкої розробки додатків (Rapid Application Development, RAD). DSDM – це ітеративний і інкрементний підхід,

який надає особливого значення тривалого участі в процесі користувача / споживача [4].

Мета методу – здати готовий проект вчасно і вкластися в бюджет, але в той же час регулюючи зміни вимог до проекту під час його розробки. DSDM входить в сімейство гнучкою методології розробки програмного забезпечення, а також розробок не входять в сферу інформаційних технологій.

2.5.5 TDD

Розробка через тестування (англ. Test-driven development, TDD) – техніка розробки програмного забезпечення, яка ґрунтується на повторенні дуже коротких циклів розробки: спочатку пишеться тест, що покриває бажану зміну, потім пишеться код, який дозволить пройти тест, і під кінець проводиться рефакторинг нового коду до відповідних стандартів. Кент Бек, який вважається винахідником цієї техніки, стверджував, що розробка через тестування заохочує простий дизайн і вселяє впевненість (англ. Inspires confidence).

Розробка через тестування вимагає від розробника створення автоматизованих модульних тестів, що визначають вимоги до коду безпосередньо перед написанням самого коду. Тест містить перевірки умов, які можуть або виконуватися, або ні. Коли вони виконуються, кажуть, що тест пройдено. Проходження тесту підтверджує поведінку, передбачуване програмістом. Розробники часто користуються бібліотеками для тестування (англ. Testing frameworks) для створення і автоматизації запуску наборів тестів. На практиці модульні тести покривають критичні і нетривіальні ділянки коду. Це може бути код, який схильний до частих змін, код, від роботи якого залежить працездатність великої кількості іншого коду, або код з великою кількістю залежностей.

Середовище розробки повинна швидко реагувати на невеликі модифікації коду. Архітектура програми повинна базуватися на використанні безлічі сильно пов'язаних компонентів, які слабо зчеплені один з одним, завдяки чому тестування коду спрощується.

TDD не тільки передбачає перевірку коректності, але і впливає на дизайн програми. Спираючись на тести, розробники можуть швидше представити, яка функціональність необхідна користувачеві. Таким чином, деталі інтерфейсу з'являються задовго до остаточної реалізації рішення.

Тому дана методологія розробки є найвідповіднішою для розробки даної дипломної роботи, а також її тестування, в порівнянні із вищезазначеними методологіями.

2.6 Висновки

Для розробки дипломного проекту, було обрано мову програмування JavaScript, а також HTML та CSS. Конфігуратор товарів буде розроблено на базі платформи Tilda використовуючи JavaScript. Tilda має вбудований функціонал, CRM систему, та MVC архітектуру, які дозволяють гнучко підійти до розробки. У якості систем аналітики було обрано Google Analytics та Yandex Metrika, Об'єднавши можливості обох систем, можливо отримати більше даних для глибокого аналізу. Методологія розробки було обрано TDD, завдяки її перевагам. Середовищем розробки було обрано Sublime Text 3 як найбільш зручне середовище розробки.