

УДК 004.738.1WWW

DOI: 10.31673/2412-9070.2022.042228

Ю. І. КАТКОВ, доктор техн. наук, доцент;

А. О. ШУЛЯК, студент,

Державний університет телекомунікацій, Київ

ХАРАКТЕРИСТИКА ОСНОВНИХ ЗНАТЬ, УМІНЬ ТА НАВИЧОК FRONT-END ТА BACK-END РОЗРОБНИКІВ

Розглянуто актуальне питання пошуку нових ефективних методів і вдосконалення наявних доволі поширених способів створення та обслуговування вебсайтів. Головну увагу приділено пошуку та підвищенню ефективності застосування інструментів для побудови та обслуговування вебсайтів завдяки впровадженню сучасних технологій. У статті наведено таку постановку завдання: основою якісного створення вебсайту є поєднання знань, умінь та навичок (творчого потенціалу) розробників та залучення ними ефективних інструментів створення вебсайту. Для розв'язання цього завдання потрібно, по-перше, розуміння основних характеристик front-end, back-end та full-stack на базі реальних інтернет-технологій, а по-друге, формування системи підготовки відповідних спеціалістів. Вирішення завдання дає змогу надати кожному проєкту креативність, ефективність та індивідуальність, а сайту — основні стратегічні переваги над потенційними конкурентами. Для досягнення цього результату було проаналізовано способи розроблення вебзастосунків; описано способи виконання front-end, back-end та full-stack для веброзроблення вебзастосунків; запропоновано засоби розроблення вебзастосунків для різних категорій розробників. На основі виконаного аналізу зроблено висновки щодо відмінностей між front-end, back-end та full-stack, що важливо для визначення переліку знань, умінь та навичок, якими має володіти розробник певного напрямку. Такими відмінностями є: їх завдання, мова програмування, алгоритми, структури даних, ролі та обов'язки. Визначено, що за умов стрімкого розвитку інформаційних технологій створення вебсайтів для кожного напрямку підготовки розробників потрібно сформулювати спеціальну систему підготовки, в якій систематизовано знання, вміння та навички на теперішній момент. Як приклад наведено вимоги та перелік необхідних знань, умінь та навичок, актуальних на поточний момент для front-end, back-end та full-stack розробників. Зазначено, що у великі команди входять як front-end, так і back-end розробники, де обидва розробники співпрацюють для розроблення життєздатних продуктів. Показано, жоден із них не працює без іншого — потрібна серверна частина, щоб змусити працювати front-end, а back-end потрібен front-end, щоб користувачі могли мати доступ до вебсайту та взаємодіяти з ним.

Ключові слова: front-end; back-end; full-stack.

Вступ

Сьогодні інтернет досяг усіх аспектів нашого життя, його використовують практично скрізь. Нині він слугує як універсальний посібник для всієї інформації і питань із програмування, приготування їжі, відпочинку, розваг тощо. Сам інтернет також зазнав змін. На початку існування це була статична мережа, призначена для візуального подання інформації або статичних даних без участі серверів та даних користувача. Наприкінці ХХ століття інтернет розширився за своїми масштабами, охопивши можливості динамічного обміну даними. Це зумовило появу концепції Web2.0 і серверної частини вебсайту, яка більше фокусується на даних користувача, що робить її кориснішою для звичайних користувачів. З усім цим сплеском технологічних досягнень з'явилася нова сфера — веброзроблення.

Веброзроблення — це створення та обслуговування вебсайтів; забезпечення заданої функціональності для користувачів, яку створює вебсайт. Воно забезпечує все, що гарантує функціональну роботу вебсайту, а саме: додавання даних та натискання кнопки за допомогою зовнішнього ін-

терфейсу, тоді як раніше всі внутрішні процеси відбувалися без участі користувача на серверах, наприклад, збирання, нагромадження та зберігання даних, шифрування, додавання в базу даних, оброблення цих даних [1; 2].

Функціональність вебсайтів залежить від того, наскільки ефективно виконуються такі умови: обслуговування клієнтської сторони (те, що ми бачимо навпроти, тобто екран); як підтримується серверна частина (що розміщено всередині вебсайту). Для цього є два взаємозв'язані види програмування: зовнішнє (Front-end) та внутрішнє (Back-end). Терміни з'явилися внаслідок розвитку принципу поділу відповідальності між зовнішнім уявленням та внутрішньою реалізацією.

Ці види програмування, з одного боку, відрізняються один від одного, а з другого боку, доповнюють один одного. Тому функціональність вебсайту залежить від того, наскільки ефективно кожна сторона спілкується та працює одна з одною як єдине ціле. Для забезпечення необхідної функціональності сайту існують такі розробники: Front-end Developer (фронтенд-розробник інтерфейсу вебсайту); Back-end Developer (бекенд-розробник

внутрішньої частини вебсайту); Full-stack Developer (фулстек-розробник повного набору для розроблення зовнішньої та внутрішньої частин вебсайту) [3; 4]. Тому веброзроблення стало невід'ємним елементом сучасного цифрового світу, для якого потрібні спеціалісти, а визначення характеристик основних знань, умінь та навичок для підготовки фронтенд-, бекенд- та фулстек-розробників є актуальним та своєчасним завданням.

Постановка завдання. Основою якісного створення вебсайту є поєднання знань, умінь та навичок (творчого потенціалу) розробників та залучення ними ефективних інструментів створення вебсайту. Для розв'язання цього завдання потрібно, по-перше, розуміння основних характеристик Front-end, Back-end та Full-stack на базі сучасних інтернет-технологій, а по-друге, створення системи підготовки відповідних спеціалістів. Вирішення завдання дає змогу надати кожному проекту креативність, ефективність та індивідуальність, а сайту — основні стратегічні переваги над потенційними конкурентами. Щоб розв'язати це завдання, потрібно виконати аналіз способів розроблення вебзастосунків; описати способи виконання Back-end та Front-end для веброзроблення вебзастосунків; запропонувати засоби розроблення вебзастосунків, а також перелік знань, умінь та навичок для кожного напрямку розробників.

Аналіз останніх наукових досліджень. Розгляд наукових досліджень у галузі створення сучасних сайтів та вебзастосунків свідчить про те, що багато наявних сайтів, вебзастосунків стали надто складними, тому над ними працює чимала команда фахівців [1; 2]. Щоб створити повноцінний, привабливий, функціональний та зручний для користувача сайт, потрібна узгоджена робота дизайнерів, бекенд- і фронтенд-розробників та тих, хто буде всю цю роботу контролювати: аналітиків, проджект-менеджерів або самих замовників — представників бізнесу. Поділ праці прискорює роботу зі створення сайту або вебзастосунків та полегшує його обслуговування [3; 4]. Це призвело до зміни парадигми розроблення сайту або вебзастосунків веброзробниками — від індивідуального кодування сторінок у HTML/CSS, JavaScript, PHP або іншій мові, використовуючи лише текстовий редактор, до застосування нових інтелектуальних технологій, що сформувавши доступні платформи, які допомагають на всіх етапах процесу побудови вебсайту. Основними інструментами бекенд- та фронтенд-розробників є різні інструменти, а саме: мови PHP, Python, Ruby, Java, C#, Node.js (програмна платформа); бібліотека Express для взаємодії платформи Node.js із сервером та Mongo DB (база даних для отримання та зберігання інформації); додаткові засоби — фреймворки Laravel, Symfony, CodeIgniter, Django, Flask, Ruby

on Rails, Spring, Express. Для зберігання даних використовуються MySQL, PostgreSQL, SQLite, а для дизайну — універсальний набір інструментів Sketch, який дає можливість додавати просту анімацію та створювати інтерактивні прототипи. За допомогою Sketch текст та зображення легко трансформуються у динамічний дизайн [4; 5].

Основна частина

Відомо, що дотримання основних етапів побудови сайтів є запорукою успішної роботи ресурсу. Тому розберемося, що потрібно зробити під час його створення. Виокремимо такі етапи створення сайтів:

- окреслюємо цілі (мету сайту, сферу застосування, для якої аудиторії, які відмінності від конкурентів тощо);
- вибираємо домен і хостинг (підбираємо доменне ім'я, надійність та вартість);
- формулюємо технічне завдання (вимоги до структури сайту, дизайну/оформленню сайту, формату контенту, засобів перегляду сайту/браузерів та їх версій, засобів мобільності тощо);
- визначаємо вимоги до системи керування контентом для адміністратора сайту (додавання фотографій у портфоліо або зміну текстів на сторінках, створення нових розділів або зміну структури тощо);
- виконуємо опис структури сайту (основні розділи, зміст контенту, засоби пошуку інформації на сайті тощо);
- здійснюємо написання текстів (головної сторінки, про послуги, про компанію) у такий спосіб, щоб вони були унікальними, оптимізованими;
- розроблюємо дизайн сайту (орієнтуємося на конкурентів і вивчаємо тренди в дизайні, щоб продумати унікальний і сучасний його варіант);
- виконуємо оптимізацію (складаємо карти релевантності, реєструємо сайт в Google Search Console, додаємо карти сайту (sitemap.xml), створюємо robots.txt, налаштовуємо сторінку помилок «404», під'єднуємо до SSL-сертифіката, прописуємо теги і метатеги);
- здійснюємо верстку сайту (зазвичай цей етап виконується майже одночасно з попереднім, але для полегшення сприйняття інформації його виокремлюють у самостійний крок);
- наповнюємо контентом (потрібно заповнити розділи і шаблонні сторінки, серед яких: сторінки карток товарів, портфоліо, різноманітні розділи типу «Блог» або «Новини», пояснення гарантій або умов доставляння й оплати, контактів, відгуків);
- підмикаємо вебаналітику (щоб визначити, наскільки ефективний сайт і чи зрозумілий він користувачам, потрібно встановити лічильники вебаналітики, наприклад, найбільш зручну і поширену систему Google Analytics).

Отже, виконання цих етапів створення сайтів потребує багато різноманітних знань та умінь від розробників. Тому вебсайт має дві частини: front-end та back-end. Інакше кажучи, front-end — це все, що бачить користувач, тобто візуальні ефекти та дизайн, а back-end — це вся логіка, яка змушує процеси відбуватися на сервері. Крім цього є full-stack (розроблення повного стека) — це, коли розроблення здійснюється одним фахівцем, який володіє технікою розробника сайтів як front-end розроблення, так і back-end розроблення. Розглянемо це докладніше, щоб розкрити особливості знань та вмінь фахівців, що розробляють вебсайт.

Front-end — це зовнішній інтерфейс. Це розроблення (front-end developer — фронтенд-розроблення) для зручного візуального відображення даних. Програміст, що його розробляє, має назву Front-end Developer (фронтенд-розробник). Фронтенд-інтерфейс дає змогу користувачу бачити, з чим взаємодіє він у своєму браузері з так званої «клієнтської сторони». «Клієнтська сторона» охоплює все, з чим безпосередньо стикається користувач: від тексту і кольорів до кнопок, зображень і навігаційних меню. Наприклад, ви вирішили розпочати бізнес і хочете надати інформацію про якість послуг або товарів, а отже, вам потрібен рекламний вебсайт. Цей вебсайт потрібен, щоб описати та показати переваги ваших послуг для потенційних клієнтів і повідомити їх, де ви перебуваєте. Для цього вам потрібно за допомогою зовнішнього інтерфейсу, що є на вашому вебсайті, виконати розміщення відповідного контенту. Стосовно програмування це статичний сайт, оскільки його вміст майже не змінюється. Для статичних сайтів уся необхідна інформація, що знаходиться на вебсторінці, міститься в самому коді зовнішнього інтерфейсу. Але, якщо ви будете змінювати контент відповідно до маркетингових планів, то з погляду програмування це не буде динамічним сайтом, оскільки для клієнта не створюватиметься база даних. Для формування такої бази потрібні інші інструменти. Загалом для створення фронтенд-інтерфейсу застосовуються мови фронтенд-розроблення, наприклад мови програмування HTML, CSS, JavaScript та ін. Тут HTML — основна мова програмування позначення коду, яка створює та впорядковує вебконтент, щоб його можна було відобразити у браузері, CSS — це мова опису зовнішнього вигляду вебсайту, яка доповнює HTML і визначає стиль вмісту вебсайту, наприклад макет, кольори, шрифти тощо, а JavaScript — це мова програмування, яку використовують для більш інтерактивних елементів, зокрема меню, модальних вікон і контактних форм. Разом ці мови створюють усе, що візуально представлено, коли ви відвідуєте вебсторінку, будь-то онлайн-покупки,

читання новин, перевірка електронної пошти або пошук Google [4].

Back-end має назву «серверна частина». Це все те, що користувачі не бачать, і містить дії, які відбуваються під час виконання будь-яких кроків на вебсайті. Розроблення серверної частини вебсайту має назву back-end developer (бекенд-розроблення). Програміст, що його розробляє, має назву Back-end Developer (бекенд-розробник). Бекенд-розроблення з погляду програмування створює програмне забезпечення для серверної програмно-апаратної частини вебсайту (вебзастосунку). Ця програмно-апаратна частина відповідає за функціонування внутрішньої частини вебсайту і забезпечує функціонування внутрішньої архітектури сервера для збирання, зберігання, шифрування та додавання даних у базу даних та їх оброблення з подальшим висновком для візуального відображення даних. Програмно-апаратна частина вебсайту потребує наявності і розроблення програмного забезпечення, яке створює зручності візуального відображення даних і охоплює все, що забезпечує роботу вебсайту: від нагромадження і зберігання даних до оброблення цих даних, тобто увага приділяється базам даних, внутрішній логіці, Application Programming Interface (API) у серверах. Особливістю цього виду програмування є те, що серверна частина бекенд-вебсайту є комбінацією серверів, застосунків і баз даних. Код, написаний бекенд-розробниками, допомагає браузерам користувачів вебсайту взаємодіяти з базами даних та зберігати дані в базі даних, зчитувати дані з бази даних, оновлювати та видаляти дані або інформацію з бази даних [4]. Наприклад, у вас є рекламний сайт під бізнес. Клієнти хочуть купувати ваші послуги чи товари онлайн. Тому ви вирішуєте зробити вебсайт інтернет-магазину, де клієнти можуть здійснювати покупки та розміщувати замовлення у будь-який час. Тобто, ваш вебсайт тепер має зберігати інформацію про продукти, покупки, профіль користувачів, кредитні картки та багато іншого. Для керування цими даними потрібне бекенд-розроблення. Тут треба звернути увагу на те, що статичні вебсайти (які раніше було розглянуто) цілком підходять для демонстрації таких речей, як реклама послуг або товарів підприємства, портфоліо веброзроблень або професійні профілі. Але якщо необхідно перетворити свій сайт на щось, із чим користувачі можуть взаємодіяти, потрібно організувати роботу серверної частини. Вона відповідає за зберігання та організацію даних, а також за те, щоб все на боці клієнта справді працювало. Серверна частина взаємодіє з інтерфейсом, надсилаючи та отримуючи інформацію для відображення у вигляді вебсторінки. Щоразу, коли ви заповнюєте контактну форму, вводите вебадресу або робите

покупку (будь-яка взаємодія з користувачем на боці клієнта), ваш браузер надсилає запит на бік сервера, який повертає інформацію як зовнішній код, котрий браузер може інтерпретувати та відображати. Отже, такий вебсайт повинен мати додаткові серверні компоненти, щоб зробити його динамічним вебзастосунком — вебсайтом, вміст якого може змінюватися залежно від того, що саме є в його базі даних, і який може бути змінено введенням користувача. Цим він відрізняється від статичного вебсайту, якому не потрібна база даних.

Бекенд-розробник забезпечує для функціонування вебсайту таке:

- керування базою даних для всієї інформації про клієнтів та продукти, яка зберігається на віддаленому комп'ютері, тобто сервері;
- заснування фундаментального коду, який дає змогу вебсайтам обробляти дії, що вживаються користувачами у зовнішньому інтерфейсі, та надавати правильну інформацію.

Серверна частина об'єднує всі компоненти зовнішнього інтерфейсу і передбачає такі дії в базах даних, як збереження історії покупок і відомостей про продукт, створення безпечних облікових записів користувачів, що редагуються тощо. Існує багато різних інструментів застосування баз даних, зокрема SQLite, MongoDB, MySQL, SQL Server, PostgreSQL та Oracle. Для опису цих баз даних використовуються спеціальні мови програмування. Наприклад, під час бекенд-розроблення для створення зовнішнього інтерфейсу, який з'єднує інтернет з базою даних, керує підімкненнями користувачів та забезпечує роботу самого вебзастосунку, застосовуються мови: Java, PHP, .NET (C#, VB, F#), Ruby, Python, SQL, JavaScript, Go тощо. Ці мови програмування часто працюють на платформах, які полегшують процес веброзроблення [5].

Треба також зазначити, що бекенд-програмування, може бути об'єктно-орієнтованим або функційним. В *об'єктно-орієнтованому бекенд-програмуванні* застосовується метод, який фокусується на створенні об'єктів, а оператори мають виконуватися в певному порядку. Популярними мовами об'єктно-орієнтованого програмування є Java, .NET та Python. У *функційному бекенд-програмуванні* застосовується метод, що ґрунтується на «дії». Функційне програмування використовує декларативну мову, а отже, оператори можуть виконуватись у будь-якому порядку. Популярними мовами функційного бекенд-програмування є SQL, F# та R. Також використовують такі інструменти, як MySQL, PostgreSQL, MongoDB, Oracle та SQL-сервер для створення, читання, оновлення або видалення даних та передавання їх на клієнтський бік або в зовнішній інтерфейс програми. Крім цьо-

го, до основних мов зовнішнього інтерфейсу використовуються вебфреймворки, зокрема Bootstrap, Django та Angular, бібліотеки JavaScript, такі як jQuery, та розширення CSS (Sass та LESS). Існує довгий перелік схожих ресурсів із набором різних інструментів для кожної мови та потенційної функції.

Раніше було показано, що створення сучасних сайтів та вебзастосунків стали надто складними, тому над ними працює ціла команда фахівців. Відомо, що програмісти — розробники сайтів мають поділ спеціалізацій: фронтенд-розробник (Front-end Developer); бекенд-розробник (Back-end Developer); фулстек-розробник (Full-stack Developer).

Розглянемо тепер завдання, які розв'язують фронтенд-, бекенд- та фулстек-розробники.

Фронтенд-розробленням (Front-end Develop) займаються фронтенд-розробники зовнішнього інтерфейсу або точніше, дизайнер інтерфейсу користувача («клієнтської сторони»). Інакше кажучи, фронтенд-розробники дбають про зовнішній вигляд вебсайту, його структуру, кнопки, зображення і навігаційне меню та точки взаємодії з користувачем. Вони зацікавлені у функціональності, інженерії, яка перетворює ці проекти на живий інтерактивний вебсайт. Хоча розроблення інтерфейсу пов'язане з візуальною та інтерактивною частиною вебсайту, із дизайном та естетикою сайту, це не те саме, що вебдизайн. Фронтенд-розробники не проєктують зовнішні аспекти вебсайту. Мета вебфреймворків та зовнішніх бібліотек — просто зробити код (і процес його написання) більш керованим та організованим завдяки різноманітним інструментам та шаблонам, сумісним із поширеними мовами програмування.

Бекенд-розробленням (Back-end Develop) займаються бекенд-розробники серверної частини вебзастосунку. Бекенд-розробник — це той, хто використовує технології, потрібні для розроблення продуктів для серверної частини будь-якого вебсайту. Він відповідає за побудову структури програмного застосування, тобто передусім зосереджений на тому, як вебсайт працює. Він пише код, зорієнтований на функціональність і логіку програми та технологій, на яких працює вебсайт і які ніколи не бачать користувачі безпосередньо. Бекенд-розробник опікується вебдизайном, проєктує зовнішні аспекти вебсайту, виконує налаштування на боці сервера, перевіряє якість коду.

Бекенд-розробники зазвичай працюють у групах або в команді. Щоб стати бекенд-розробником, ви повинні добре знатися на таких темах: мови внутрішнього програмування; бази даних; сервер; API. Також невід'ємною частиною роботи бекенд-розробника — це вміння працювати з фреймворками як допоміжними засобами для спрощення розв'язання типових завдань.

Потрібно зауважити, що у великі команди входять як бекенд-, так і фронтенд-розробники, де обидва розробники співпрацюють для розроблення життєздатних продуктів. Однак жоден із них не працює без іншого — потрібна серверна частина, щоб змусити працювати фронтенд, і вам потрібен фронтенд, щоб користувачі могли отримати доступ до вебсайту та взаємодіяти з ним. Вважається, що бекенд-розроблення є складнішою і більш «програмістською» частиною створення сайту, а фронтенд — більш простий та пов'язаний.

Фулстек-розроблення (Full-stack Develop) здійснюють фулстек-розробники (Full-stack Developer), які можуть працювати як із бекенд-, так і з фронтенд-технологіями. Тобто вони одночасно займаються як зовнішнім інтерфейсом, так і вебдизайном.

Тепер розглянемо ролі та обов'язки веброзробників.

Оскільки технологія серверної частини є комбінацією серверів, застосунків і баз даних, то **до обов'язків бекенд-програмістів** може входити:

- написання програмного інтерфейсу спеціального протоколу для взаємодії комп'ютерних програм, який дає змогу використовувати функції однієї програми всередині іншої, тобто API (Application Programming Interface);

- написання коду для взаємодії з базою даних;
- створення бібліотек;
- робота над бізнес-процесами й архітектурою даних і багато іншого.

Також є ще кілька окремих обов'язків бекенд-програміста, а саме: координувати свої дії з розробниками зовнішнього інтерфейсу та розробляти серверні алгоритми для ефективного передавання даних клієнтським вебзастосункам; взаємодіяти з менеджером проєкту (Project manager) та інженером забезпечення якості (Quality assurance) для оптимізації та розроблення якісної взаємодії з користувачем (UX/UI — User Experience / User Interface); переконуватися, що програма працює швидко і працює однаково в разі зміни трафіку користувача; співпрацювати із зацікавленими сторонами, щоб зрозуміти їх конкретні потреби, аби потім перетворити їх на технічні вимоги та запропонувати найбільш ефективно та дієво технічне рішення; оптимізувати програми для збільшення часу відгуку та ефективності; аналізувати вимоги та цілі, обробляти помилки та знаходити ефективні та швидкі вирішення; взаємодіяти з базою для зберігання даних; керувати та розробляти API-інтерфейси (вони допомагають двом програмам взаємодіяти одна з одною через інтернет), які виконуються на пристроях; створювати архітектуру системи з огляду на масштабованість, швидкість та стабільність програм; упроваджувати структуру безпеки та передовий досвід; повторно викорис-

товувати код та бібліотеки для майбутньої реалізації.

Основне завдання бекенд-розробника можна подати так: розроблення моделі предметної галузі (домена); формування платформи та основного функціоналу, тобто бізнес-логіки; створення безпечних застосунків, що підтримують інтерфейс користувача; налаштування серверів (основного, тестового та робочого) та програм із моніторингу їх станів; використання системи контролю версій (це загальна вимога до всіх програмістів); налаштування баз даних, створення моделей предметної галузі та взаємодій між ними; налаштування процесів неперервної інтеграції та постачання, робота з докер-контейнерами, які необхідно запускати в kubernetes.

До обов'язків фронтенд-розробника входить: визначитись із дизайном та структурою вебсторінки; розробити функції для покращення читабельності сайту; досягнути балансу між функціональним та естетичним дизайном; переконатися, що вебдизайн підтримує смартфон; створити зрозумілий для подальшого використання код; переконатися, що вебсторінки оптимізовані для найкращої швидкості та масштабованості; зберігати бренд у всьому дизайні.

Основне завдання фронтенд-розробника — це зв'язати надані дизайнером графічні макети у вебзастосунку (сторінках сайту) з бекендером і за потреби реалізувати обчислювальний функціонал на боці користувача. Основні технології в арсеналі фронтендера це HTML, CSS і JavaScript. Зазвичай під час роботи в команді потрібно знати і розбиратися в багатьох процесах, суміжних із роботою фронтенд-розроблення. Будучи вже досвідченим фронтенд-розробником, потрібно бути знайомим із бекенд-технологіями і розуміти принципи взаємодії користувача із застосунками (UX/UI).

Для прикладу розглянемо певний перелік знань, умінь та навичок. Щоб стати затребуваним та досвідченим спеціалістом у галузі фронтенд-розроблення, потрібно знати та вміти використовувати такі технології: HTML та CSS (HTML5 API, HTML5/CSS3 Polyfills, сітки та CSS-фреймворки, специфікації W3C та WHATWG); знати та мати навички для вільного працювання з JavaScript; знати логіку роботи клієнт-серверної архітектури; знати та вміти використовувати популярні бібліотеки та фреймворки типу: React.js, jQuery, Angular, JS, Redux, js.; знати принципи побудови сучасних односторінкових застосунків (Single Page Application); знати роботу препроцесорів CSS (Sass, Less, Stylus тощо); знати популярні CMS (WordPress, Drupal, Joomla тощо); знати та вміти використовувати OOCSS/BEM/SMACSS; знати ECMAScript 6; знати та розуміти DOM; знати принципи побудови back-end (Node.js, PHP, Ruby, .NET тощо); знати

JavaScript транспайлер (Babel); вміти застосовувати інструменти дебагінгу (Chrome Dev Tools, Firebug тощо); вміти застосовувати графічні редактори (Photoshop, Illustrator тощо); мати навички застосовувати інструменти контролю версій (Git, GitHub, CVS тощо); вміти застосовувати дані та мови запитів (SQL, MySQL, NoSQL, MongoDB тощо).

Проте різниця між фронтендом і бекендом не завжди настільки відчутна, щоб відрізнити бекенд-розробника від фронтенд-розробника. Деякі розробники володіють як інтерфейсом, так і сервером; це звані фулстек-розробники повного стека. Але для розуміння ключова відмінність полягає в тому, що тоді як розробники бекендів встановлюють, як працює вебсайт, фронтенд-програмісти створюють і проєктують інтерфейс, визначаючи, який вигляд матиме сайт для користувачів. На практиці бекенд-розробники зазвичай отримують більшу заробітну плату, ніж фронтенд-розробники, оскільки бекенд-мови переважно більш технічні. Середня зарплата бекенд-розробника становить майже 100 тис. дол. на рік, а фронтенд-розробника — 80 тис. дол. на рік. Фронтенд-розробнику не потрібно знати особливості реалізації сервера, а бекенд-розробнику — реалізацію фронтенду. Однак їх спільна робота зрештою визначає досвід користувача і робить можливим функціонування вебсайту взагалі [4; 5].

Висновки

Інтернет — це всесвіт, що швидко розвивається. Усе, що ми використовуємо і бачимо в повсякденному житті, перемістилося до мережі. З цим підйомом технологій зростає і попит на фронтенд- та бекенд-розробників. Якщо вам подобається ідея працювати з візуальним дизайном і втілювати його в життя, створюючи першокласний досвід користувача, то вам, імовірно, сподобається працювати в зовнішньому інтерфейсі. Якщо вам більше до душі працювати з даними, розбиратися в алгоритмах та вигадувати способи оптимізації склад-

них систем, ви можете віддати перевагу роботі бекенд-розробника. Щоб стати бекенд-розробником потрібно добре знати хоча б одну з мов програмування для бекенда та серверної частини. Також треба розуміти, як створювати, читати, оновлювати та видаляти (відомі як операції CRUD) дані з бази даних та передавати їх стороні сервера, зверненому до користувача. Бекенд-розробники повинні вміти працювати з API та протоколами HTTP, а також із системами контролю версій, зокрема Git. Кожна мова має свої варіанти використання, переваги та недоліки. І від розробника залежить, якою мовою він/вона почувається комфортно.

Список використаної літератури

1. **Best web development tools in 2022** [Електронний ресурс]. URL:

<https://www.techradar.com/news/best-web-development-tool> (дата звернення: 12.05.2022).

2. **15 Tools That Will Help You Build Your Website (No Coding Required)** [Електронний ресурс]. URL:

<https://blog.producthunt.com/15-tools-that-will-help-you-build-your-website-no-coding-required-33e9e51a2df6#tvaky89dd> (дата звернення: 12.05.2022).

3. **14 полезных инструментов, ускоряющих и упрощающих веб-разработку** [Електронний ресурс]. URL:

<https://habr.com/ru/company/ruvds/blog/546974/> (дата звернення: 12.05.2022).

4. **33 полезных инструмента для веб-разработки** [Електронний ресурс]. URL:

<https://academy.yandex.ru/posts/33-poleznykh-instrumenta-dlya-veb-razrabotki> (дата звернення: 12.05.2022).

5. **Backend Developer Skills You Must Have. June 21, 2021** [Електронний ресурс]. URL:

<https://www.interviewbit.com/blog/backend-developer-skills/#what-is-backend-development> (дата звернення: 12.05.2022).

Yu. I. Katkov, A. O. Shulyak

CHARACTERISTICS OF BASIC KNOWLEDGE, COMPREHENSION AND SKILLS FRONT-END AND BACK-END OF THE DEVELOPER

The article is devoted to the topical issue of finding new effective methods and improving existing common ways of creating and maintaining websites. The main focus is on finding and improving the efficiency of tools for creating and maintaining websites through the introduction of modern technologies. The article presents the following problem statement: The basis of high-quality website creation is a combination of knowledge, skills and abilities (creative potential) of developers and their involvement in effective tools for creating a website. To solve this problem it is necessary to understand the basic characteristics of front-end, back-end and full-stack based on modern Internet technologies. Solving the problem allows you to give each project creativity, efficiency and individuality, and give the site the main strategic advantages over potential competitors. To solve this problem in the article performed: analysis of ways to develop Web-applications; describes the methods of front-end, back-end and full-stack for web development of Web-applications; offered tools for developing WEB-applications for different categories of developers. Based on the analysis, the article draws conclusions about the differences between front-end, back-end and full-stack, which is important to determine the list of knowledge, skills and abilities that

should have a developer in a particular area. These differences are: their tasks, programming language, algorithms, data structures, roles and responsibilities. It is shown that in the conditions of rapid development of information technologies of creating websites for each direction of training of developers it is necessary to create a special training system, which systematizes knowledge, skills and abilities at the moment. As an example, the requirements and list of required knowledge, skills and abilities relevant at the moment for front-end, back-end and full-stack developers are given. It is emphasized that the big teams include both front-end and back-end developers, where both developers work together to develop viable products. It is shown that none of them works without the other — you need a server part to make the front-end work, and the back-end needs a front-end so that users can access and interact with the website.

Keywords: front-end; back-end; full-stack.

