

**ЗАКЛАД ВИЩОЇ ОСВІТИ
УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА**

**Факультет суспільних і прикладних наук
Кафедра інформаційних технологій**

на правах рукопису

Мельник Віталій Зіновійович

УДК 004.415

Розробка платформи для вебпублікацій

Спеціальність 121 — «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра

Науковий керівник
к.т.н., доц.
Пашкевич Олег Петрович

Івано-Франківськ — 2021

					КР.ІІЗ – 13.00.000 ІЗ			
Зм.	Лист	№ докум.	Підпис	Дата				
Розроб.		Мельник В.З.			Розробка платформи для електронних публікацій із <i>modern web-application</i>	Літ.	Арк.	Аркушів
Перевір.		Пашкевич О.П.					6	66
Реценз.						ЗВО «УКД», ІІЗс-2017		
Н. Контр.		Зорін В.О.						
Затверд.		Пашкевич О.П.						
					Пояснювальна записка			

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП.....	10
РОЗДІЛ 1. ТЕОРИТИЧНІ ВІДОМОСТІ	12
1.1 Вебдодатки. Їх структура, переваги та недоліки	12
1.2 Технології розробки вебдодатків	18
1.3 Проектування вебдодатків	29
Висновки до розділу 1	34
РОЗДІЛ 2. ІНСТРУМЕНТИ РОЗРОБКИ ВЕБДОДАТКІВ	35
2.1 JavaScript	35
2.2 Поняття одно сторінкового додатку	38
2.3 Технології, які використовуються для реалізації вебдодатку	41
Висновки до розділу 2	44
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПЛАТФОРМИ ДЛЯ ВЕБПУБЛІКАЦІЙ	56
3.1 Реалізація головної сторінки.....	45
3.2 Налаштування середовища розробки.....	47
3.3 Генерування нових компонентів	49
Висновки до розділу 3	53
РОЗДІЛ 4 ОХОРОНА ПРАЦІ	54
4.1 Умови праці на робочому місці з ПК.....	54
4.2 Мікроклімат та ступінь повітря виробничого приміщення.	56
4.3 Вимоги до безпечної експлуатації комп'ютерної техніки.....	63
Висновки до розділу 4	64
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66

					КР.ІПЗс – 13.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- URL (Uniform Resource Locator) - стандартизована адреса певного ресурсу в інтернеті.
- HTML (HyperText Markup Language) - стандартизована мова розмітки документів в всесвітній мережі.
- HTTP (HyperText Transfer Protocol) - протокол передачі гіпертекстових файлів.
- XML (eXtensible Markup Language) - розширювана мова розмітки.
- WSDL (Web Services Description Language) - мова опису вебсервісів і доступу до них, заснований на мові XML.
- CGI (Common Gateway Interface) - стандарт інтерфейсу, який використовує зовнішня програма для зв'язку з вебсервером.
- IIS (Internet Information Services) - набір серверів для декількох служб Інтернету від компанії Майкрософт.
- DLL (Dynamic Link Library) - динамічно приєднувана бібліотека
- API (Application Programming Interface) - набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.
- JVM (Java Virtual Machine) - набір комп'ютерних програм та структур даних, що використовують модель віртуальної машини для виконання інших комп'ютерних програм чи скриптів.
- SSI (Server Side Includes) - технологія дозволяє зручно розробляти вебсторінки з частин, вставляти в них результати виконання CGI-скриптів/
- PHP (Personal Home Page) - скриптова мова програмування, створена для генерації HTML-сторінок на стороні вебсервера.
- JS (JavaScript) - динамічна, об'єктно-орієнтована мова програмування.
- OLE (Object Linking and Embedding) - технологія зв'язування та

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

впровадження об'єктів в інші документи та об'єкти

– COM (Component Object Model) - платформа компонентно-орієнтованого програмування.

– JSP (Java Server Pages) - технологія, що дозволяє динамічно генерувати HTML, XML та інші вебсторінки.

– MVC (Model-View-Controller) - архітектурний шаблон, використовується під час проектування та розробки програмного забезпечення.

– ОС - операційна система.

– DOM (Document Object Model) - специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Перші кроки в побудові глобальної комп'ютерної мережі були зроблені в 60-х роках ХХ століття, а вже в 1969 році відбувся сеанс зв'язку між двома віддаленими її абонентами. У наступні десятиліття Інтернетом користувалося невелика кількість вузьких фахівців. Ситуація кардинально змінилася на початку 90-х, коли в побут увійшли персональні комп'ютери, а загальний доступ у Всесвітню павутину забезпечили браузері з хорошим інтерфейсом. З тих пір пройшла тільки чверть століття, а Інтернетом користується вже більше половини населення Землі. Всесвітня мережа проникла в усі сфери людської діяльності, без неї не обійтися в фінансовій сфері, сервісі, комунікаціях і т.д. Масово стали виникати спеціальності, пов'язані з інформаційними технологіями, одна з найбільш дефіцитних серед них програміст мережевих ресурсів.[1]

На ринку праці сьогодні склалася ситуація, коли попит на програмістів помітно перевищує пропозицію, тому класний веброзробник на вагу золота. Розвиток інтернет торгівлі та інших мережевих сервісів вимагає створення великої кількості сайтів і мобільних додатків. Для цього використовується безліч спеціальних мов, як розмітки, так і вебпрограмування, які часто комбінуються розробниками в рамках одного проекту. Кожен з них відрізняється різним рівнем складності, і одним з найбільш часто вживаних є JavaScript багато-платформна об'єктно-орієнтована скриптова мова програмування, який використовується для реалізації інтерактивних елементів на сторінці. За допомогою JavaScript можна значно розширити функціонал прикладної програми, налагоджуючи таким чином зворотний зв'язок з користувачем.

Ця мова дозволяє реалізовувати складну поведінку вебсторінки. Кожен раз, коли ви бачите вебсторінку, вона не тільки відображає статичний зміст, а й відображає оновлення контенту, виводить інтерактивні карти, 2D, 3D анімацію,

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

прокручує відео і т.д.

Відмінність цієї мови програмування від ще одного поширеного PHP в тому, що він виконується на стороні клієнта, а не сервера. А якщо бути точніше за допомогою його браузера. Звідси виникають і деякі ризики щодо безпеки, але, в той же час, знижується навантаження на сервер.

Актуальність теми. «Розробка платформи для вебпублікацій». Метою даної кваліфікаційної роботи є вивчення проблеми будови і розробки компонентів для інструментальних вебзасобів організації платформи для вебпублікацій. Було вивчено особливості розробки вебдодатків на мові програмування JavaScript, із використанням фреймворку React.

Результатом роботи є розробка платформи для вебпублікацій. Опис процесу розробки компонентів та методів реалізації подібних платформ.

Мета роботи. Розробка платформи електронних публікацій.

Об'єкт роботи. Процес розробки сучасного вебсайту призначеного для постачання музичних інструментів.

Предмет роботи. Методика розробки інтернет-ресурсу згідно сучасних тенденцій розробки.

Завдання роботи. Відповідно до вибраної теми в роботі покладені такі задачі:

- вибір мови програмування, технологій та інших суміжних програм;
- розроблення сучасного та зручного застосунку;
- проведення тестування продукту.

Методи роботи. Для вирішення поставленого завдання були використані: мова гіпертекстової розмітки HTML для розміщення блоків на сторінці, каскадна таблиця стилів CSS для стилізації сторінкии.

Структура роботи. Розділи – 4. Загальний обсяг основної частини – 63 сторінок. Список використаних джерел – 15.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

РОЗДІЛ 1. ТЕОРИТИЧНІ ВІДОМОСТІ

1.1 Вебдодатки. Їх структура, переваги та недоліки.

Користувач за допомогою браузера відправляє HTTP-запит за певною URL адресою, який зіставляється з ресурсами на вебсервері. Якщо дана адреса вказує на динамічний ресурс, то сервер запускає деяку зовнішню програму, яка формує HTML сторінку, яка може бути показана браузером, і повертає її клієнту. Основною частиною вебдодатку є його логіка на стороні вебсервера.

Найчастіше, вебдодаток не є самостійною програмою (окрім технології CGI), а виконується під управлінням деякої зовнішньої програми середовища виконання. В якості такої програми можуть виступати: сервер або контейнер додатків. Середовище виконання робить такі дії:

- приймає від вебсервера всю інформацію, пов'язану з HTTP запитом;
- визначає, яка програма, яке було створене для даного середовища, має бути виконано;
- створює всі допоміжні об'єкти, які можуть бути потрібними для роботи даного додатка;
- завантажує потрібний додаток і передає йому управління;
- вебдодаток виконується і створює HTML сторінку, яку записує в спеціально створений для цього об'єкт і після цього повертає управління вебсерверу;
- вебсервер формує HTTP відповідь, включає в нього сформовану вебдодатком HTML сторінку і відправляє його клієнту, який надіслав запит до нього.

Як правило, HTML сторінки, створені вебдодатком, включають HTML форми, з якими працює користувач. Послідовність викликів вебдодатку користувачем складають сеанс його роботи. Він завершується або в результаті вибору користувачем команди про закінчення його роботи з вебдодатком, або в

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

результаті довгого проміжку часу між відправленнями наступних HTTP запитів.

Варто відзначити, що прикладна програма призначена для вирішення деяких завдань користувачів і обов'язково має графічний інтерфейс, призначений для користувача.

Проаналізувавши сучасний інтернет простір, можна виділити такі приклади типових вебдодатків:

- пошукові системи;
- відео колекції;
- новинні системи;
- мас-меді;
- бізнес: магазини; банки; замовлення послуг;
- соціальні мережі;
- прогноз погоди;
- блоги;
- хостинги файлів;
- обробка зображень;
- робота з музикою;
- ведення бухгалтерії;
- купівля квитків на авто та авіатранспорт.

Онлайн програми, в порівнянні з локальними, мають наступні переваги:

– Простота доступу до додатка. Будь який користувач, яка має комп'ютер, і підключений до мережі Інтернет, може використовувати вебдодаток.

– Простота розгортання (установки). На відміну від локальних додатків, веб, після завершення розробки, не потребує встановлення на комп'ютерах користувачів. Досить тільки повідомити їм URL адреса додатки. При зміні додатку всі користувачі одразу починають працювати зі зміненою версією.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

– Високий рівень розвитку і надійності мережевих з'єднань та інтернет технологій.

– Однак як переваги, вони, відповідно, мають і свої недоліки:

– Відсутність підтримки поточного стану сеансу роботи і затримка при перезавантаженні кожної сторінки. Кожна перезавантаження або оновлення сторінки викликає помітну затримку, викликану необхідністю встановити HTTP з'єднання, обробити запит на сервері, передати по мережі у відповідь HTTP повідомлення і перезавантажити сторінку. Це створює скачки та переривчастий режим роботи користувача.

– Обмежений набір елементів управління для проектування форм прикладних програм. Поточна версія мови HTML підтримує досить обмежений набір елементів управління. Вона не запроваджує підтримку складних алгоритмів взаємодій, які часто використовуються в десктопних додатках, які доступні в навіть найпростіших локальних додатках.

Хоча вони можуть бути розроблені за допомогою JavaScript і CSS, відсутність вбудованої підтримки браузерів веде різноманітності реалізацій з неузгодженими способами роботи з ними.

Як базова технологічна архітектура інтернет мережі, так і обмежений набір доступних елементів керування робить підтримку взаємодії з вебдодатками значно складнішою, в порівнянні з локальними.

Окрім цього, оскільки більшість інтернет додатків робляться незалежно від конкретного браузера, то взаємодії і зовнішнє передання вебдодатки може бути не однаковим в різних операційних системах.

– Неузгоджені між собою підходи до оформлення і способу взаємодії користувача із прикладним додатком. Хоча інтернет надає розробникам і дизайнерам значну гнучкість і можливість проявити свої творчі здібності, неузгодженість призначених для користувача інтерфейсів і способів взаємодії з розробленими додатками, що виходить в результаті цього розмаїття, часто створює проблеми для користувачів. Це викликано тим, що кожен з них пропонує свій персональний перелік об'єктів, дій і візуальних представлень,

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

перемішаних в одному і тому ж додатку.

– Для вирішення таких проблем проектування і пов'язаних з ними проблем використання, багато компаній розробляють керівництва з проектування користувальницького інтерфейсу і стилю для впорядкування оформлення та роботи прикладних програм.

– Відкритість доступу до додатків. У зв'язку з відкритим доступом до вебдодатків, практично для всіх них потрібно підтримувати безпеку їх використання користувачами.

Структура вебдодатків. Вони зазвичай діляться на логічні частини, при цьому, кожному з них призначається своя роль. Локальні додатки можуть складатися тільки з одного шару, який розміщується на комп'ютері клієнта, в той час, як онлайн додатки можуть складатись з N-шарів. Прикладну вебпрограму, також можна розділити на базові та функціональні підсистеми. До базових підсистем відносяться:

- Одноманітне оформлення вебсторінок, які становлять додаток.
- Підтримка поточного стану сеансу роботи користувачів.
- Налаштування інтернет сторінок для різних користувачів (персоналізація).
- Навігація між різними інтернет сторінками.
- Забезпечення безпеки (розпізнавання, авторизація, реєстрація користувачів).
- Доступ до джерел даних.

До функціональних підсистем відносяться:

- Управління контентом вебдодатків. Він повинен забезпечувати можливість завантаження контенту на сервер (uploading), класифікації, пошуку і отримання з сервера (downloading).
- Пошук інформації в контенті додатки (в документах, файлах, базах даних).
- Підтримка взаємодії користувачів (форуми, обмін повідомленнями, рецензування документів). Нижче наведений (рис. 1.1), з типовими

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

структурами вебдодатку.

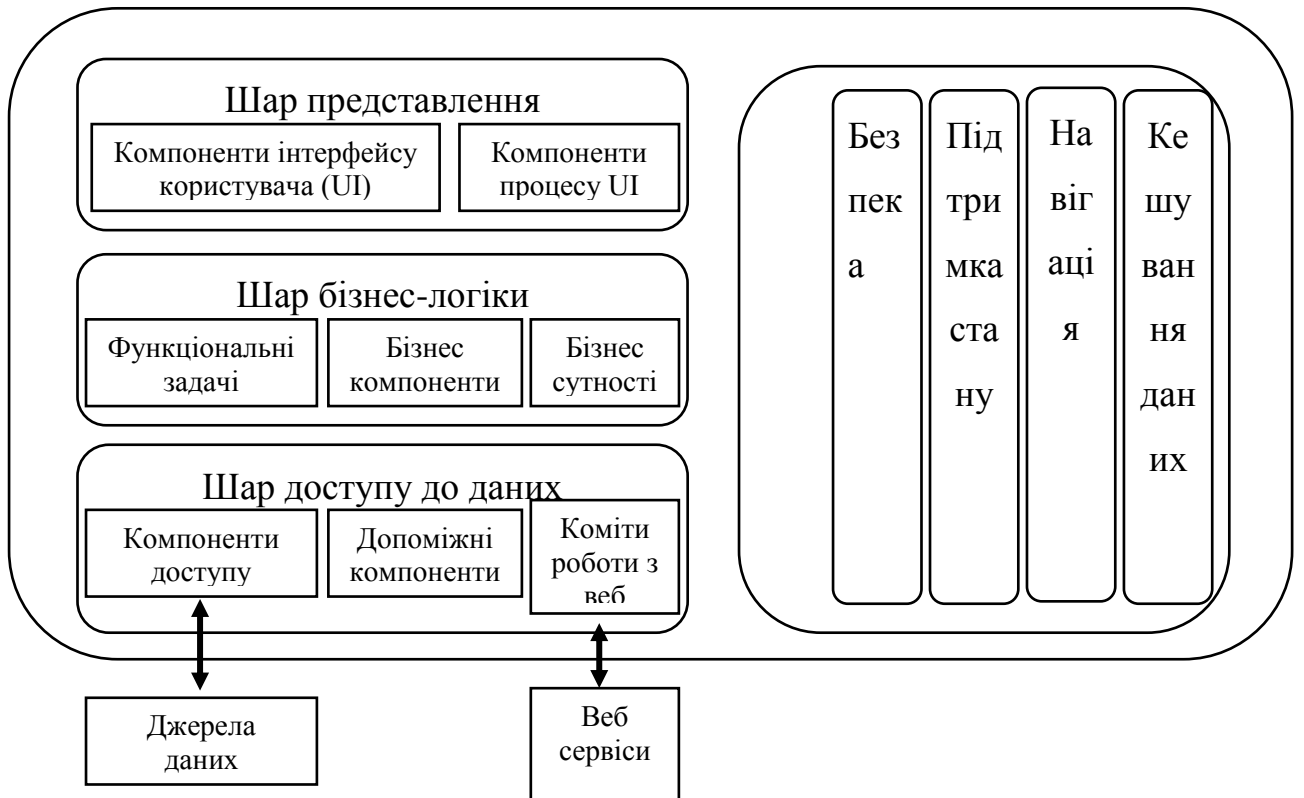


Рисунок 1.1 - Типова структура вебдодатку

Базова функціональність багато в чому реалізується середовищем виконання вебдодатку, а для створення функціональних підсистем потрібно скласти відповідний програмний код.

Вебдодаток складається реалізує технологію клієнт серверної взаємодії. Клієнтська частина реалізує користувальницький інтерфейс, формує запити до сервера і обробляє відповіді від нього. Серверна частина отримує запит від клієнта, виконує обчислення, після цього формує вебсторінку і відправляє її клієнту через мережу з використанням протоколу HTTP.

Для реалізації GUI користувача використовують:

- HTML, XHTML;
- CSS.

Для формування запитів, створення інтерактивного і незалежного від браузера інтерфейсу:

- ActiveX;
- Java;
- JavaScript;
- Silverlight.

Окрім вебдодатків іншим важливим видом програмного забезпечення, використовуваного в інтернет мережі, є вебсервіси.

Перш за все, вебсервіси це технологія. І як і будь-яка інша технологія, вона має досить чіткі окреслену середу застосування. По своїй суті, вебсервіси це реалізація певних інтерфейсів, створених для обміну даними між різними додатками, які написані на різних мовах програмування і знаходяться в різних вузлах мережі.

Вони являються наборами методів, які можна викликати на виконання із заданими параметрами за допомогою HTTP запитів, а результати їх виконання отримувати у вигляді HTTP відповідей. Вони підтримуються, як і вебдодатками, так і вебсерверами. Вони призначені для використання будь якими програмами, які можуть формувати правильні HTTP запити і розрізняти отримані відповіді, але не користувачами, оскільки вони не мають графічного інтерфейсу.

XML дані сформовані в якості кореневого елемента використовують елемент «Envelope», в який включені два під елементи «Header" (не є обов'язковим) та «Body». В елементі «Header» описується загальна інформація, пов'язана із запитом або відповіддю.

Елемент «Body» в запиті містить інформацію про метод, який викликається, і переданих йому параметрах. Складений таким чином XML документ передається за допомогою протоколів HTTP або SMTP.

При використанні протоколу HTTP, даний XML документ (Envelope) включається в тіло HTTP запиту, яке проходить через пустий рядок після рядка запиту і відповідних заголовків.

Після отримання даного запиту вебсервер передає управління середовищі виконання, яка викликає необхідний метод і передає йому отримані параметри.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

В процесі виконання даного методу формується XML документ, що містить результати роботи методу.

Для того, щоб користувачі сервісів могли правильно складати запити і розуміти відповіді, вебсервіси також є відповідальними за надання метаданих, що описують повідомлення, які вони можуть приймати і створювати.

Кожну вебслужбу за спеціальним HTTP запитом надає WSDL документ, в якому описується вся необхідна користувачу інформація, для організації роботи з даним сервісом. WSDL-документ дозволяє розробникам описувати синтаксис виклику для кожного створеного методу. Більш того, він дозволяє застосовувати інструменти та методи для автоматичного генерування спеціальних програмних засобів, які роблять виклик методів вебсервісу такими ж простими, як і застосування об'єктів локальних класів.

1.2 Технології розробки вебдодатків

– В загальному, способи розробки вебдодатків поділяють на 3 великі категорії:

– Підходи, котрі засновані на програмуванні або скриптах: зовнішні програми або скрипти; розширення вебсервера.

– Підходи, які базуються на застосуванні готових шаблонів вебсторінок, що включають вставки коду скриптів і спеціальних серверних тегів.

– Об'єктні середовища каркаси, фрейм ворки.

– Хоч ці підходи пересікаються між собою, більшість широко відомих підходів пов'язана з однією конкретною категорією.

При запровадженні даного підходу, вебдодатком (динамічним ресурсом, пов'язаним з URL адресою) являється зовнішня програма, складена на певній універсальній високо рівневій мові програмування (наприклад, Java або C++) або скрипт, написаний за допомогою відповідної мови, виконання якого

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

відбувається також за допомогою зовнішньої програми інтерпретатора скриптів.

Одним з головних недоліків з даним підходом до розробки є зосередження великої кількості часу на написанні коду. Розмітка HTML та інші існуючі конструкції форматування вбудовуються напряму в логіку роботи програмного коду за допомогою операторів виведення.

Це суттєво обмежує можливості дизайнерів вносити зміни та корективи в оформлення новостворюваної додатком сторінки. Вебдизайнер може проектувати макет сторінки, проте, саме програміст повинен потім буде перетворити його на код і зв'язати його зі скриптом, або програмою. По суті, для зміни практично будь якого елементу вже сформованої сторінки потрібне втручання програміста, чи стосується це зміни самої логіки роботи програми, чи тільки лишень зміни її оформлення і розташування певних структурних елементів даної сторінки.

Зовнішні програми. Найпростіший спосіб утворювати вебсторінки внаслідок HTTP запиту, в процесі виконання сценаріїв, полягає в тому, щоб передати роботу по вирішенню потрібного завдання і утворення HTML сторінки зовнішньої програми, яка повинна отримувати передані в HTTP запиті вхідні параметри і сформувану вихідну сторінку на мові HTML.

Найпершою подібною програмною технологією створення і реалізація вебдодатків, яка широко використовувалась, незалежно від типу від сервера, була технологія CGI. Вона визначала масив правил поведінки, які програма повинна була враховувати, для того, щоб різноманітні HTTP сервера і ОС могли її використовувати.

Відповідно до даній технології, під час вступу до вебсерверу HTTP запиту, який включає посилання не на статичну сторінку, а на CGI програму, генерується новий процес, в якому запускається необхідна прикладна програма. Технологія CGI задає спосіб передачі такій програмі параметрів, які входять до складу HTTP запиту. Передача вхідних даних може виконуватися або за допомогою фіксованого набору змінних середовища, які можуть створюватися

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

однією програмою і використовуватися іншими програмами, або через вхідні дані функції, з якої починається робота програми, а результати роботи програми повертаються за допомогою стандартного потоку виводу.

Ця технологія дозволяє використовувати будь-яку мову програмування, яка може працювати зі стандартними пристроями вводу/виводу. Окрім цього, CGI програми можна писати з використанням скриптових мов, які називаються "CGI скриптами". Прикладами таких CGI мов є Perl, Python або Tcl. При використанні скрипта вебсервер викликає на виконання зовнішню програму - інтерпретатор скриптів, якій передаються дані HTTP запиту і ім'я файлу, в якому міститься запитуваний користувачем скрипт. Як наслідок, потім ця програма виконує вказаний скрипт і повертає серверу повністю сформовану HTML сторінку.

Технологія CGI являється одним із найпростіших способів утворювати інформацію в вебмережі динамічним чином, але вона має істотні недоліки, які роблять його не практичною в більшості випадків:

– Основним і найсуттєвішим недоліком є продуктивність: для кожного HTTP програмного запиту до CGI вебсервер запускає новий процес, який завершує свою роботу тільки після завершення самої програми. Робота зі створення і завершення процесів є досить трудомісткою, що може дуже швидко знизити продуктивність всієї системи, крім цього, різні активні процеси починають конкурувати між собою за права використання системних ресурсів, а також їхніми об'ємами, такі як оперативна пам'ять.

– Для створення і налагодження CGI програм, програміст повинен володіти якісними знаннями програмування на одній із конкретного списку мов, які підтримують процес написання CGI програми.

– В CGI код самої програми і розмітки повністю перемішані між собою. Дизайнер повинен мати хороші знання у програмуванні, для того щоб змінювати структуру вебсторінок.

Розроблений на початку 1990-х років, CGI був найдавнішим розповсюдженим методом, який дозволяв вебсторінці бути інтерактивною.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

Незважаючи на те, що CGI все ще використовується, він є порівняно неефективний порівняно з новішими технологіями, внаслідок чого замінений ними.

Спробою об'єднати переносимість CGI додатків з ефективністю є технологія FastCGI. Дана технологія ґрунтується на простій ідеї: замість необхідності кожен раз запускати абсолютно новий процес для обробки CGI-програми, FastCGI дозволяє не закривати процеси, пов'язані з CGI скриптами, після закінчення обробки, а використовувати їх для обробки нових запитів до CGI програмам. Це означає, що нові процеси не потрібно постійно запускати і видаляти, так як один і той же процес може неодноразово використовуватися для обробки запитів. Такі процеси можуть ініціалізовуватись тільки один раз при їх створенні.

Модулі сервера, які виконують функціональність FastCGI, взаємодіють з HTTP сервером за допомогою своїх власних API. Вони намагаються приховати деталі реалізації і конфігурації від FastCGI додатків, але розробники все одно повинні знати особливості реалізації технології FastCGI, так як модулі різних типів серверів не сумісні між собою.

Розширення вебсерверів. Недоліки технологій CGI можливо нівелювати за рахунок розширення існуючих можливостей вебсерверів, запроваджуючи спеціальні компоненти. Використовуючи дані розширення, для програм, які формують HTTP відповіді, можливо підвищити загальну ефективність, без потреби завершення процесу їхньої роботи, після переробки кожного запиту, і за допомогою спільного використання потрібних ресурсів декількома додатками. Такий вид технологій, як правило, надає спроможність залишати в основній пам'яті статистичні дані ітерацій роботи користувачів, які працюють із зазначеним додатком напротязі великої кількості HTTP запитів.

Інтерфейс ISAPI. Спеціально для вебсервера Microsoft IIS (Internet Information Server) був розроблений програмний інтерфейс ISAPI, який дозволяв створювати прикладні програми, які б розширювали стандартні можливості даного вебсервера. ISAPI являє собою бібліотеку функцій, за

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

допомогою якої програмісти можуть створювати вебдодатки у вигляді DLL модулів, які формують HTML сторінки. Такі вебдодатки виконують свою функцію набагато швидше звичайних CGI програм, за рахунок того, що вони більш тісно інтегровані в вебсервер.

За допомогою технології ISAPI можуть створюватися два види компонентів: розширення і фільтри, які компілюються в DLL файли та запускаються вебсервером динамічним чином. ISAPI-розширення можуть зв'язуватись із викликом файлів, котрі мають спеціальні розширення, або із файлами, котрі містяться в певних заданих каталогах або в усьому вебсайті. ISAPI-фільтри використовуються для зміни або підвищення вже наявної функціональності IIS сервера. Зазвичай вони обробляють (фільтрують) кожен HTTP запит, який надходить. Фільтри можуть застосовуватися для аналізу і модифікації вихідних HTTP відповідей.

Інтерфейс JavaServlet API. Інший широко цією технологією розширення архітектури вебсервера є прикладний інтерфейс Java Servlet API, котрий пов'язує сервер із віртуальною машиною Java. JVM підтримує виконання спеціальної Java програми, яка відповідає за управління даними сеансу роботи і виконання Java сервлетів. Хоча сервлети можуть відповідати на багато типів запитів, вони найчастіше реалізують вебконтейнери для розміщення вебдодатків на вебсерверах і, отже, кваліфікуються як вебAPI сер влетів на стороні сервера.

Сервлети це спеціальні класи на мові Java (програми), котрі мають доступ до інформації з HTTP запитів. Вони формують HTTP відповіді, які повертаються браузерам.

Контейнер сер влетів відповідає за отримання від вебсервера HTTP запитів на виконання сер влетів; створення сеансу роботи користувача, якщо це потрібно; виклик сервелату пов'язаного з HTTP запитом; передачу сервелату параметрів, які містяться в HTTP запиті, представлених у вигляді Java об'єктів.

На відміну від ISAPI розширень, дана технологія може переноситись між різними вебсерверами, операційними системами і комп'ютерними

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

платформами. Сер влети виконуються однаково в будь-якому середовищі, яка надає сумісний з ними контейнер сер влетів. Технологія Servlet API використовується великою кількістю розробників і підтримується багатьма відомими вебсерверами.

При зміні статичної розмітки логіка формування динамічної інформації не змінюється, але в програму необхідно вносити зміни і виконувати їх перекомпіляцію (якщо це не скрипт).

Підходи, які засновані на шаблонах використовують в якості об'єктів (що мають URL адреса) не програми або скрипти, а «шаблони». По суті шаблони є HTML файлами із додатковими серверними "тегами", які задають методи інтеграції сформованого контенту динамічним способом. Тобто, файл такого шаблону містить HTML код, який задає загальну структуру сторінки, і додаткові серверні теги, розміщені таким чином, щоб сформований за їх допомогою зміст сторінки мав необхідний вигляд.

На сьогодні, до найбільш поширених технологій розробки вебдодатків, на основі шаблонів є SSI, Cold Fusion, PHP, ASP і JSP.

Технологія SSI. Технологія вставок на стороні сервера є старою, яка з'явилася майже одночасно з технологією CGI. SSI надає можливість вставки додаткових файлів (або результатів виконання CGI скриптів) в HTML сторінку. SSI інструкції вставляються в вигляді HTML коментарях, тим самим гарантуючи, що не оброблені на сервері інструкції будуть ігноруватися браузером, який отримав такий файл. Прикладами SSI інструкцій є, echo для виведення значень змінних середовища; include для вставки вмісту інших файлів; ехес для виконання CGI програми на стороні сервера і включення сформованого нею HTML коду в шаблон.

Технологія SSI надає простий і зручний спосіб додати динамічний зміст до вже існуючих сторінок, без необхідності генерувати всю сторінку цілком.

Технологія Cold Fusion. Іншою досить популярною технологією, заснованою на шаблонах, є технологія Cold Fusion, розроблена компанією Adobe.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

Перевага даного підходу полягає в тому, що даний шаблон може створюватися і підтримуватися дизайнером сторінки, котрий має базові знання мови HTML і вебграфіки, але не має досвіду програмування. Спеціальні теги, які є "розширенням" HTML в деякій мірі схожі на інструкції SSI, тим, що дозволяють швидко навчитися використовувати їх дизайнерами, які мають не великий досвід роботи з кодом, оскільки використання таких тегів вимагає менше досвіду, ніж написання програмного коду.

Технологія PHP Hypertext Preprocessor. Технологія "PHP Hypertext Preprocessor" або просто PHP дозволяє розробникам вбудовувати програмний код в шаблони, за допомогою мови, схожого з мовою скриптів Perl.

Код PHP, як правило, обробляється на вебсервері інтерпретатором PHP, реалізованим як модуль або виконуваний файл CGI. На сервері результат інтерпретованого та виконаного PHP коду, який може бути у вигляді будь-якого типу даних, буде утворювати цілу відповідь або його частину на HTTP запит. Різні системи вебшаблонів, системи управління вебвмістом та вебфреймворки існують, які можливо пристосувати для організації або сприяння генерації такої відповіді. PHP код також можливо реалізувати безпосередньо із командного рядка.

Стандартний інтерпретатор PHP, що працює на базі Zend Engine, безкоштовне програмне забезпечення, випущене під ліцензією PHP. PHP широко переноситься, і його можна безкоштовно розгорнути на більшості вебсерверів майже на будь-якій операційній системі та платформі.

Технологія Active Server Pages. Компанія Microsoft розробила технологію ASP, яка об'єднала можливості створення шаблонів, включаючи скрипти, з доступністю до низки OLE і COM об'єктів, які наявні в операційній системі Windows, в тому числі і до ODBC джерел даних. Як і PHP шаблони, ASP сторінки можуть включати блоки скриптів (використовують посилання на COM об'єкти), перемішуючись із HTML форматуванням. На відміну від таких об'єктно-орієнтованих мов, як Java або C++, мова, яка використовується в ASP сторінках, була плоскою, лінійною і строго процедурною.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

На відміну від технології PHP, ASP не пов'язаний з однією конкретною скриптовою мовою програмування. У ASP в якості стандартного мови використовується мову Visual Basic Scripting Edition (VBScript), але може використовуватися і JS.

У ASP шаблони можуть включатися блоки, виділені за допомогою певних тегів, які містять код скрипта, що виконується інтерпретатором ASP шаблонів, при формуванні відповіді. HTML розмітка, яка знаходиться поза таких блоків, розглядається як вихідний HTML код і просто переписується в формується HTML сторінку.

Той факт, що технологія ASP входила до складу вебсервера Microsoft IIS, зробило її дуже привабливою для використання розробниками, які працюють в ОС Windows. У зв'язку з популярністю технології ASP, вона була реалізована і на інших платформах. Основною перевагою даної технології є прискорення розробки і установки щодо простих вебдодатків.

Технологія Java Server Pages. Технологія JSP була відповіддю на популярність технології Microsoft ASP. Як і технологія PHP, виконання JSP сторінок реалізується за допомогою препроцесора, який перетворює (трансює) їх в початковий код сервлету. HTML розмітка, яка стоїть поза виділених блоків трансюється в оператори print мови Java.

Технологія JSP розвивалася і з часом до неї було додано такі нові можливості, як бібліотеки JSP, набір спеціальних (серверних) тегів, які не передаються в HTTP відповіді браузеру, а використовуються при обробці JSP сторінки в контейнері сер влетів на стороні вебсервера.

Фактично кожен спеціальний тег це деяка функціональність, для реалізації якої, в іншому випадку, потрібно було би включати за допомогою вбудованих блоків, які містили би Java код. В ході виконання трансляції JSP сторінки, котра виконується до компіляції і виконання створеного сервлету, виконується перетворення таких спеціальних тегів в Java код.

Технологія Java Standard Tag Library. Компанія Sun продовжила розвивати технологію JSP, як платформу розробки, включивши неї можливість

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

створення своїх власних JSP тегів. В результаті цього з'явилася велика кількість спеціалізованих тегів, розроблених локально різними групами розробників, що призвело до ще більшої плутанини. Спочатку компанія Sun не надала ні стандартів, ні вказівок щодо організації і структурування спеціалізованих тегів. Однак потім компанія запропонувала специфікацію бібліотеки стандартизованих тегів на мові Java JSTL, які розділені на кілька категорій:

- Базові теги, що надають стандартну функціональність, звичайну для шаблонного підходу: включення, виконання ітерацій, перевірка умов і т.п.
- XML теги, що надають аналогічну функціональність в XML контексті, а також можливість обходу елементів XML документа, із використанням виразів XPath.
- SQL теги, які надають інструменти для опису джерел даних, подання запитів та виконання ітерацій по отриманим результатами вибірки;
- формуючі теги, які включають функції інтернаціоналізації і локалізації, а також засоби для форматування дат і чисел.

Підстановка параметрів виконується за допомогою мови виразів, пов'язаного з JSTL, що надає доступ до змінних, визначених у різних областях: на сторінці, в запиті, сеансі роботи користувача або всієї прикладної програми в цілому.

JSTL зробили можливим складання JSP сторінок без включення в неї коду на мові Java. Тим самим дана технологія надала корисні можливості для реалізації компонента View в JSP Model 2, особливо в такий об'єктної MVC середовищі, як Struts. Хоча JSTL має всі можливості для розробки повнофункціональних вебдодатків, в основному вона використовується в якості технології розробки представлень в контексті об'єктних MVC середовищ.

Звичайні скриптові технології на стороні сервера використовують різні об'єкти, але не дозволяють розробляти і використовувати власні класи і на їх основі створювати об'єкти. У зв'язку з цим подальший розвиток вебтехнологій було пов'язано зі створенням спеціальних об'єктно орієнтованих технологій розробки вебдодатків. Використання даних технологій дозволяє зробити

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

розробку вебдодатків більш схожою з розробкою звичайних об'єктно орієнтованого програмного забезпечення.

Об'єктні середовища (фреймворки) представляють собою наступний рівень вдосконалення розробки вебдодатків. Замість об'єднання розмітки і логіки в єдиний модуль, об'єктні середовища підтримують принцип відокремлення змісту від представлення. Модулі відповідальні за створення контенту відділяється від модулів, які показують це зміст в конкретному форматі.

Відокремлення змісту від представлення є важливим у зв'язку з тим, що збільшується гнучкість програми (можливість її зміни з невеликими витратами); поліпшується поділ обов'язків та відповідальності між вебдизайнерами і програмістами.

В даний час є два підходи до створення об'єктно орієнтованих вебдодатків:

- підходи, котрі базуються на наборі спеціальних вебсторінок, котрі пов'язані із описами класів, об'єкти яких будуть створюватися і запроваджуватись в процесі їхнього виклику;

- підходи, засновані на використанні наборів класів, відповідних шаблоном MVC.

Об'єктний підхід на основі форм. Підхід на основі вебформ є подальшим розвитком скриптових серверних технологій. В даному підході в HTML документи додаються спеціальні теги, які обробляються на стороні сервера. Крім цього, можна описувати і використовувати власні теги у вигляді класів на універсальних мовах програмування (Java, C #, VisualBasic і т.п.). Створення на стороні сервера об'єктної моделі програми, аналогічної об'єктної моделі локального додатку.

Основними об'єктними підходами на основі форм є наступні:

- технологія Microsoft ASP.Net Web Form;
- технологія JavaServer Faces (JSF).

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Технологія JavaServer Faces була відповіддю компанії Sun на появу технології ASP.Net, і зокрема на використовувану в ній підтримку інтерфейсу користувачів.

На відміну від технології JSTL, JSF надає більш високо-рівневий підхід за допомогою збільшених візуальних компонент.

Підхід на основі архітектурного шаблону MVC. Відповідно до архітектурним шаблоном MVC всі класи, які складають додаток, діляться на три основні групи (компоненти): Модель (Model), Представлення (View) і Контролер (Controller).

Кожен з цих компонентів відповідає за свої завдання: моделі набір класів, що реалізують всю бізнес логіку вебдодатки. Ці класи відповідають за обробку даних, а також за взаємодію між самими об'єктами, які складають такі дані.

Представлення набір класів, що відповідають за інтерфейс взаємодії з користувачами (UI). З їхньою допомогою утворюються HTML сторінки, які показують користувачам дані. Воно використовує дані з Моделі і надає користувачам можливість виконувати їх редагування.

Контролер сполучна ланка між першими двома компонентами. Класи даного компонента отримують дані про запит до сервера, і передає їх в Модель для обробки і збереження. Після обробки отриманих даних Контролер вибирає, яким чином показати їх клієнту за допомогою використання деякого класу із Представлення.

В результаті такого поділу вебдодатки на компоненти, розробник отримує повний контроль над формується HTML документів, спрощується структура додатка; полегшується завдання виконання тестування програми; досягається повне відокремлення логіки роботи програми від представлення даних

Прикладами технологій розробки на основі MVC є:

- технологія Struts (заснована на мові Java);
- технологія ASP.Net MVC, що входить до складу набору технологій ASP.Net платформи .Net Framework;
- технологія Ruby on Rails.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

1.3 Проектування вебдодатків

Вебтехнології дозволяють розробляти як прості вебдодатки, так і складні інформаційні вебсистеми. Для успішної розробки таких програм потрібно розуміти загальний процес написання програмного забезпечення: основні дії по розробці, які повинні бути виконані; як ці дії взаємопов'язані між собою; який порядок їх виконання; які спеціалісти, повинні приймати участь в цьому процесі.

Розробка реальних вебдодатків можлива тільки в результаті колективної роботи. Залежно від розміру програми і беруть участь в його створенні фахівців, розробка може бути складною справою, схильним до різних ризиків, які можуть впливати на успіх кінцевого результату.

Розробка програмного забезпечення це творчий процес, який ведуть до створення інноваційних програмних продуктів і систем. Зазвичай даний процес не є єдиним монолітним блоком робіт, який приймає на вході деякі ідеї про новий додатку, а в якості результату видає рішення (програмний продукт), повністю відповідно до цієї ідеї.

Процес розробки може бути розділений на набір базових робіт з добре визначеними межами і здоровим глуздом:

- розуміння проблеми;
- планування подальшої стратегії;
- виконання плану;
- перевірка точності отриманого результату;
- доопрацювання з метою видалення можливих помилок або не точностей.

Основні види робіт по розробці вебдодатків. У процесі розробки будь-якого можна виділити наступні основні види діяльності:

- Визначення вимог до (інжиніринг вимог).
- Проектування: призначене для планування вирішення проблеми.
- Реалізація: перетворення плану в працюючий програмний код.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

- Перевірка і оцінка якості: призначено для виявлення помилок кодування або певних розбіжностей між певними вимогами і їх реалізацією.
- Розгортання, надання користувачам можливості працювати зі створеним.
- Підтримка, призначене для відстеження використання діючої системи і збереження її працездатності.
- Розвиток: призначене для поліпшення з часом розробленого рішення, надання нових входних даних для процесу розробки в формі нових вимог.

Інжиніринг вимог призначений для розуміння затверджених можливостей і характеристик. Даний аналіз спрямований на визначення функціональних (які функції система повинна виконувати) і не функціональних вимог (якість пропонованого рішення). Він також передбачає виявлення загальної, основоположної ідеї, яка стоїть за системою, яка розробляється, основних зацікавлених осіб, яким потрібна нова система і умови, в яких буде використовуватися система.

Проектування призначене для опису рішення, яке повинно відповідати функціональним вимогам і вимогам ефективності, а також обмеженням того середовища, в якій вона буде працювати. Попередньо зібрані вимоги уточнюються і поліпшуються для того, щоб задовольняти можливим технологічним обмеженням. Проектування включає такі дії, як:

- Проектування схеми даних і класів.
- Проектування компонент.
- Проектування графічного інтерфейсу.
- Проектування архітектури системи.

Допомагає краще сформулювати специфічні особливості системи, такі, як: структура, поведінка, взаємодія, дані і потік управління. Дозволяє розділити сфери відповідальності, основний принцип програмної інженерії: вирішення проблеми шляхом її розподілу на різні під задачі, може допомогти впоратися зі

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

складністю і досягти необхідних технічних якостей, таких, як адаптованість, простота підтримки, розширюваність і т.д.

На етапі реалізації проекту розроблені проектні рішення перетворюються у відповідний програмний код (вручну або за допомогою інструментів автоматизації програмування).

Можуть бути потрібні бібліотеки програм, різні мови програмування, різні комунікаційні протоколи і технічні пристрої.

Перевірка і оцінка якості зазвичай проводиться паралельно з реалізацією, так як правильність і надійність проміжних результатів, а не тільки кінцевого продукту є дуже важливим для гарантування якості всього програмного рішення. Якість в значній мірі пов'язано з наступними критеріями:

- оцінкою функціональності, тобто правильності поведінки додатку, щодо заданих функціональних вимог.
- оцінкою продуктивності (тобто час очікування відповіді при розкладанні в звичайних умовах і при пікових навантаженнях).
- оцінкою зручності використання (usability), тобто легкість використання, комунікаційна ефективність і відповідність стандартам використання.

Розгортання додатки надає користувачам можливість використання цього додатка. Залежно від типу програми, процес розгортання може включати:

- установка на комп'ютерах клієнтів;
- установка центрального додатки і баз даних на сервері;
- конфігурування проміжного комунікаційного;
- інструктування і навчання майбутніх користувачів, в особливості, якщо встановлюється зовсім новий додаток, а не нова версія вже наявної програми.

Підтримка розгорнутого і ефектів у програмному забезпеченні означає забезпечення його робочого стану, яке складається в гарантуванні його доступності та зменшенні збоїв його роботи. Може включати: періодичну

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

перевірку файлів журналу; звітів про помилки та очищення тимчасових файлів; виправлення помилок; установку виправлень.

Розвиток програми. Додаток призначений для вирішення реальних завдань користувачів. Однак, у зв'язку з розвитком організації, ускладненням розв'язуваних завдань, поліпшенням розуміння користування користувачем можливостей, його обов'язково потрібно буде постійно розвивати. Нові потреби користувачів з'являються тільки після того, як вони деякий час попрацюють.

Після цього вони починають давати свої пропозиції та коментарі. Поява нових вимог може викликати необхідність запуснути весь процес розробки заново. Незважаючи на суворе дотримання рекомендаціям правильної організації процесу розробки програмного забезпечення, часто тільки після розгортання і накопичення деякого досвіду роботи користувачів з ним стає ясно, що деякі вимоги не були виконані достатньо якісно та в повному обсязі, і додаток повинен бути доопрацьовано.

- Вебдодатки мають архітектурні, технологічні особливості.
- Вебдодатки повинні розгортатися дуже швидко і вимоги до них, ймовірно, змінюються в ході етапу розробки;
- все більш частою практикою у веброзробці стає заміна документів на реальні прототипи додатку;
- дуже раннє включення кінцевих користувачів до тестування і розвитку вебдодатки;
- в той час, як при розробці звичайних додатків створюється тільки одна версія додатка, яка задовольняє всім заданим вимогам, для вебдодатків все більш частою (і бажаним) стає публікація в вебмережі додатку, який ще не задовольняє всім заданим вимогам ;
- отримання ранніх відгуків користувачів стають все більш важливим для розробки вебдодатки, а цикл розвитку дійсно починає розглядатися в якості реальної можливості для поліпшення вебдодатки, а не як додаткова витрата часу на доведення додатки до робочого стану;

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

– робота з проектування вебдодатків ділиться на проектування даних і проектування навігації.

– Розробка вебдодатків включає специфічні процеси:

– Вебдодатки є спеціальним видом звичайних про програмних додатків і тому вебінженерія може розглядатися в якості спеціальної різновиди програмної інженерії;

– розробка додатків для вебмережі передбачає виконання кількох добре певних правил і угод, які надають стабільно працюючу, стійку і масштабовану середу розробки і виконання;

– існують спеціальні процеси розробки, які враховують специфічні можливості вебдодатків.

Дана модель складається з 5 основних видів діяльності і семи переходах між ними. До основних видів діяльності по розробці вебдодатків відносяться:

- аналіз вимог;
- проектування;
- реалізація;
- тестування і оцінка;
- підтримка і розвиток.

Основна відмінність даної моделі від моделі розробки зазвичайго програмного забезпечення полягає в розумінні "розгортання програми" не як виду діяльності (роботи), а як перехідного процесу між роботами.

У інтернет мережі розгортання додатків для його користувачів дійсно не є трудомісткою роботою, так як в зв'язку з централізованою архітектурою звичайних вебдодатків; відсутністю спеціального коду програми на стороні користувачів; використанням браузера в якості середовища виконання, робота з розгортання вебдодатку в значній мірі полегшується і прискорюється.

Дана модель пропонує явне з'єднання роботи по «підтримки і розвитку» з «аналізом вимог».

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

Висновки до розділу 1

Було розглянуто вебдодатки, їх структуру, переваги та недоліки. Також були описані технології розробки вебдодатків і проектування вебдодатків.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ІНСТРУМЕНТИ РОЗРОБКИ ВЕБДОДАТКІВ

2.1 JavaScript

JavaScript мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний та функціональний стилі. Поряд з HTML та CSS, JS є однією з основних технологій Всесвітньої павутини. JS є домінуючою мовою сценаріїв на стороні клієнта в Інтернеті, і 97% вебсайтів використовують його для цієї мети [13]. Скрипти вбудовані в документи HTML або включені до них і взаємодіють із DOM. Усі основні веббраузери мають вбудований механізм JavaScript, який виконує код на пристрої користувача.

Програми написані на ній називаються скриптами. Вони можуть запроваджуватись в HTML і автоматично запускається в процесі завантаження вебсторінки. Вони інтегруються, як простий текст, і не потребують спеціальної попередньої підготовки або компіляції для запуску. На сьогоднішній день, сценарій JS може відбуватися не тільки в браузері, а й напряму на сервері або на будь-якому іншому доступному пристрої, який має підтримку движка JS.

Сучасний JS не надає низько рівневий доступ до процесора пристрою, або його даних, тому що, з самого початку, він був розроблений саме для браузерів, які не передбачають їхнє використання. Спроможності JS сильно залежать від середовища, в якій він буде реалізовуватись. У браузері для JS є все, що пов'язано з маніпулюванням вебсторінками, їхньою взаємодією з сервером та кінцевим користувачем.

Області застосування JS:

– Вебдодатки JS використовується в клієнтській частині вебдодатків: клієнт серверних програм, які мають чітко розділену логіку між сервером і клієнтом. Процес взаємного обміну інформацією в таких прикладних додатках відбувається по мережі інтернет. Одним з основних переваг даного підходу є

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

те, що користувачі абсолютно ніяк не залежать від виду конкретної ОС самого користувача, завдяки чому, такі вебдодатки є між-платформними сервісами.

– AJAX дану мову програмування застосовують в AJAX, підході до утворення інтерфейсів вебдодатків, назначених для кінцевого користувача, який полягає в неявному обміні даними браузера із вебсервером, який відбувається не синхронно. В такому випадку, в процесі оновлення даних, вебсторінка не буде повністю перезавантажуватись і робота інтерфейсу такого вебдодатку стає в рази швидше, ніж при аналогічних умовах в традиційному підході.

– Comet величезне поняття, котре характеризує сам механізм реалізації режиму роботи вебдодатків, які застосовують сталі HTTP з'єднання, що дозволяє серверам переправляти дані браузеру без процесу виклику додаткового запиту з його боку.

– Для подібних прикладних програм запроваджуються технології, які підтримуються браузерами безпосередньо, в них широко застосовується мова JS, зокрема.

– Браузерні операційні системи.

– Букмарклеті невеликі програм, які розташовуються в закладки браузера.

– Призначені для користувача скрипти в браузері окремий вид програм, написані за допомогою мови JS, які реалізують свою роботу в браузері користувача в процесі завантаження вебсторінки.

– серверні додатки програми, написані на JS, які мають властивість запускатись і реалізовуватись на серверах, що використовують версію Java 6 і новіші.

– Мобільні додатки.

– Віджети невеликі допоміжні програми, графічний модуль яких розміщується в робочому просторі відповідної програми, що служить для різноманітних цілей. JS використовується як для реалізації самих віджетів, так і

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

для їхніх движків.

- Прикладне програмне забезпечення.
- Маніпуляція об'єктами додатків дана мова також застосовується в якості скриптової мови доступу до об'єктів додатків.
- Офісні програми JavaScript використовується в офісних прикладних програмах для різноманітних функцій: доведення до автоматизації звичайних дій, написання необхідних макросів, організації доступу до них, зі сторони вебслужб.

Можливості JS в браузері користувача обмежені заради його безпеки, для запобігання доступу до особистої інформації кінцевого користувача або нанесення шкоди його персональним даним. Приклади таких обмежень:

JavaScript на вебсторінці не здатні читати або записувати довільні файли на дисковому просторі пристрою користувача, копіювати їх або запускати програми. Він не має прямого доступу до системних функцій ОС.

Сучасні браузери надають дозвіл на роботу коду із файлами, але з обмеженими правами, і тільки у випадку, якщо користувач виконує відповідні дії.

Існують певні можливості використання програмами камери чи мікрофону і інших пристроїв для запису аудіо та відеоінформації, але вони потребують прямого явного доступу в користувача. Таким чином, сторінка з підтримкою JS не може непомітно включити вебкамеру, спостерігати за ним і зберігати інформацію.

Різні вікна вкладки не передають інформацію між собою. Іноді одне вікно, використовуючи JavaScript, відкриває інше вікно, однак і в цьому випадку JavaScript з однієї сторінки не має прямого доступу до іншої, якщо вони прийшли з різних джерел (олітика однакового джерела).

Для обходу даного обмеження, обидві сторінки мають містити узгоджений код, який певним чином дозволив би їм обмінюватися даними. Дане обмеження існує для безпеки кінцевого користувача.

JS може легко взаємодіяти з сервером, з якого прийшла поточна сторінка.

					КР.ІПЗ – 13.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Проте його спроможність отримувати дані з інших сайтів або доменів обмежена.

Подібні обмеження не діють, якщо JavaScript використовується поза браузера, наприклад на сервері. Сучасні браузери надають плагіни та розширення, за допомогою яких можна запитувати додаткові дозволи.

Дана мова програмування має, як мінімум, три сильні сторони:

- Повна інтеграція написаного коду із HTML та CSS.
- Прості речі робляться простими алгоритмами.
- Дана мова програмування підтримується всіма основними

браузерами і включений за замовчуванням.

Лише JavaScript єдина технологія, котра об'єднує ці три фактори, що і робить JavaScript найпоширенішим інструментом для його використання, зокрема, задля створення інтерфейсів в браузері.

2.2 Поняття одно сторінкового додатка

Single page application (SPA) програма, яка працює в браузері і не перезавантажує сторінку під час поточної роботи. Як і будь-який інший додаток, воно призначене для того, щоб допомогти користувачеві у вирішенні конкретного завдання, що стоїть перед ним. У SPA можуть застосовуватись всі серверні технології існуючі на даний момент.

Існують деякі відмінності між підходами при реалізації звичайного вебсайту і одно-сторінкового додатку (рис. 2.1).

Код JavaScript можна застосовувати у веббраузері для відображення інтерфейсу користувача (UI), запуску логіки самої програми та його взаємодії із

Доступні бібліотеки з відкритим кодом, які підтримують побудову одно-сторінкових додатків, кількісно зменшують процес написання JS коду.

SPA промальовується як персональний додаток. Воно утворює лише ті частини інтерфейсу, які змінилися, і в той момент, коли це необхідно.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

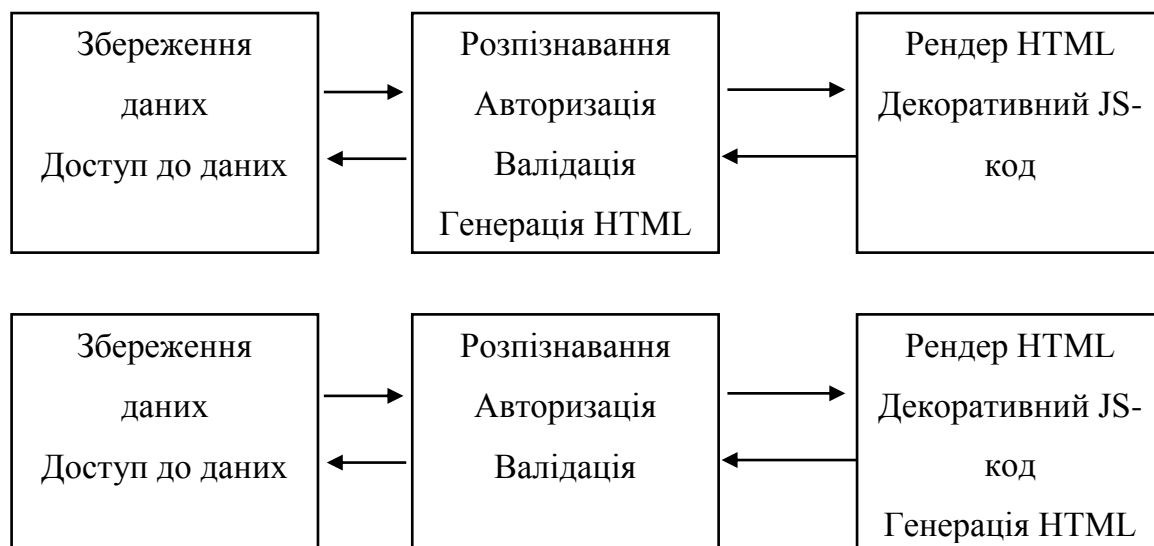


Рисунок 2.1. - Схема порівняння роботи одно сторінкового додатку (знизу) і традиційного підходу до створення веб сайтів (зверху)

Традиційний сайт навпаки перебудовує абсолютно всю сторінку в залежності які дії були здійснені користувачем, що, в свою чергу, продукує та генерує певні затримки, по тій причині, що сам браузер повинен отримати інформацію про сторінку від сервера і згенерувати її на екрані. І у випадку, якщо поточна сторінка має великий розмір, сервер на даний момент перевантажений або з'єднання з мережею інтернет є дуже повільним, то затримка може скласти кілька секунд, і кінцевому користувачу залишається лише здогадуватись, коли буде можливо відновити роботу з нею. Це найбільша проблема, якщо порівнювати стандартні підходи зі швидкою генерацією і миттєвим зворотним зв'язком одно-сторінкових прикладних програм.

Також, SPA може діяти як персональна прикладна програма. Воно мінімізує час реагування, за рахунок того факту, що перенаправляє поточні робочі дані і частину процесу обробки із сервера напряму в браузер.

В інструментарії SPA наявні дані і бізнес-логіка, необхідні для прийняття більшості рішень локально, а значить, швидко. На сервері мають лишатись тільки процеси розпізнавання користувача, валідація та постійного збереження поточних даних. У випадку застосування звичайних додатків велика частина його логіки знаходиться на сервері, тому, щоб отримати реакцію на якісь дії,

користувач повинен дочекатися завершення циклу запит, відповідь, оновлення, що може зайняти кілька секунд, в той самий час, як відповідна реакція SPA майже миттєва.

SPA може повідомити кінцевого користувача про поточний стан програми, як і персональний додаток. Якщо він повинен очікувати відповіді від сервера, то він може оновлювати індикатор по ходу виконання алгоритмів або зайнятості в реальному часі, для зручності користувача.

Доступ до одно сторінкового додатку, як і до звичайного сайту, майже завжди наявний, на відміну від більшості переносних програм. Кінцевий користувач може звернутися до SPA, маючи тільки лише з'єднання з мережею і браузер.

SPA повністю погружається при початковому завантаженні сторінки, а потім регіони сторінок замінюються або оновлюються новими фрагментами сторінок, завантаженими із сервера на вимогу. Щоб уникнути надмірного завантаження невикористаних функцій, SPA часто поступово завантажує більше функцій, коли вони стають необхідними, або невеликі фрагменти сторінки, або повні екранні модулі.

Таким чином існує аналогія між "станами" в SPA та "сторінками" на традиційному вебсайті. Оскільки "навігація станом" на одній сторінці аналогічна навігації сторінками, теоретично будь який вебсайт на основі сторінки може бути перетворений на одно сторінкову сторінку, змінивши на тій самій сторінці лише змінені частини.

SPA, як і сайт, здатний працювати на різноманітних платформах. На відміну від більшості переносних програм, добре написаний одно сторінковий додаток може адекватно працювати абсолютно в будь якій ОС, де є сучасний браузер з підтримкою стандарту HTML5.

Процес оновлення і поширення даного типу одно-сторінкових додатків є надзвичайно легким, і, переважно, він не вимагає абсолютно ніяких дій з боку користувача.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Технологія SPA це еволюція від моделі, для якої від самого спочатку були розроблені браузері, і внаслідок цього, виникають декілька пов'язаних з цим проблем. Можливі рішення включають:

- Клієнтські бібліотеки JavaScript .
- Серверні фреймворки, які спеціалізуються конкретно на моделі SPA.
- Еволюція браузерів та специфікація HTML5, спеціально розроблена для моделі SPA.

2.3 Технології, які використовуються для реалізації одно-сторінкового вебдодатку

JS-фреймворки існують для того щоб допомогти нам створювати додатки, що володіють подібними можливостями, використовуючи узагальнений підхід. Однак багатьом додаткам вся та міць, яку дають фреймворки, не потрібна. Використання будь-якого фреймворка в проекті невеликого або середнього масштабу, до якого пред'являються певні специфічні вимоги, цілком може виявитися непотрібною тратою сил і часу.

На даний момент існує багато фреймворків за допомогою яких можна реалізувати одно сторінкове додаток. У кожного є свої плюси і недоліки.

React інтерфейсна бібліотека JS із відкритим кодом, для побудови користувальницьких інтерфейсів або компонентів інтерфейсу. Його можна застосовувати як ядро при створенні одно сторінкових або мобільних додатків. Однак він займається лише управлінням станом та наданням цього стану в DOM, тому для створення програм React зазвичай потрібно використовувати додаткові бібліотеки для маршрутизації, а також певну функціональність на стороні клієнта.

React бібліотека компонентів для користувача інтерфейсу (UI-компонентів). UI-компоненти створюються з React на мові JavaScript, а не на спеціальній мові шаблонів. Цей підхід грає основоположну роль в філософії

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

React. UI-компоненти React це автономні блоки функціональності, призначені для конкретної мети. Такі компоненти мають як візуальним представленням, так і динамічної логікою. Деякі компоненти навіть здатні на самостійні взаємодії з сервером

Він, по суті, є абсолютно незалежним від серверної частини (back-end) для розробки клієнтської частини, тобто, від Node.js або MERN для подальшого його використання. Ніщо не заважає застосовувати React з іншими технологіями серверної частини, такими як Java, Python, Ruby, Go. Зрештою, React - UI-бібліотека. Її можна інтегрувати як з будь-якою серверною частиною, так і бібліотекою даних клієнтської частини.

Дана технологія має ряд особливостей, які виділяють даний фреймворк на фоні інших:

Компоненти. Код React складається з елементів компонентів. Вони можуть бути відтворені до певного елемента в DOM за допомогою спеціальної бібліотеки React DOM. При візуалізації такого компонента є можливість передавати певні значення. Існують два основних способи оголошення компонентів, функціональні (оголошуються функцією, яка потім повертає деяку кількість JSX) та компоненти на основі класу (на основі класів оголошуються за допомогою класів ES6).

Віртуальний DOM. У внутрішній реалізації React використовує віртуальну модель DOM для визначення відмінностей між поточним вмістом браузера і новим представленням. Цей процес називається пошуком відмінностей в DOM, чи узгодженням стану з поданням. Це означає, що розробникам не потрібно турбуватися про явну зміну представлення, оскільки він буде оновлюватися автоматично, у міру потреби, а їм досить оновити стан.

Крім необхідних оновлень DOM, фреймворк може виконати н-потрібні оновлення, які тільки знижують швидкодію складних інтерфейсів користувача. Ефект особливо помітний і неприємний при великій кількості динамічних UI-елементів на вебсторінці.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

З іншого боку, віртуальна модель DOM React існує тільки в пам'яті JavaScript. Кожен раз, коли відбувається зміна в даних, React спочатку порівнює відмінності по своїй віртуальній моделі DOM, тільки коли бібліотека знає, що в рендері відбулися зміни, вона оновлює фактичну модель DOM. Зрештою React оновлює тільки ті частини, для яких це абсолютно необхідно, щоб внутрішній стан (віртуальна модель DOM) і надання (реальна модель DOM) не відрізнялися. Ще одна перевага віртуальної моделі DOM можливість проведення модульного тестування без браузерів, які не мають графічного інтерфейсу.

React не намагається надати повну "бібліотеку програм". Він розроблений спеціально для побудови користувальницьких інтерфейсів, тому не включає багато інструментів, які деякі розробники можуть вважати необхідними для створення додатка. Це дозволяє вибрати, яку бібліотеку розробник воліє виконувати такі завдання, як забезпечення доступу до мережі або локального зберігання даних. У міру дозрівання бібліотеки з'являються загальні моделі використання.

Існують такі недоліки даного фреймворку:

– React не є повноцінним фреймворком на всі випадки життя. Розробнику доводиться використовувати React в поєднанні з такою бібліотекою, як Redux або React Router, щоб досягти функціональності, порівнянної з Angular або Ember. Це також може стати перевагою, якщо вам потрібна мінімалістська UI-бібліотека для інтеграції з існуючим стеком.

– За зрілості React дещо поступається іншим фреймворками.

– React використовує дещо нетрадиційний підхід до веброзробки, а JSX і Flux (часто використовувані з React як бібліотеки даних) можуть здатися страхітливими для новачків. Є нестача практичних методів, хороших книг, навчальних курсів і ресурсів для освоєння React.

– React створює тільки одностороннє зв'язування. Хоча одностороннє зв'язування краще для складних додатків, оскільки воно усуває більшу частину

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

складнощів, деякі розробники, які звикли до двостороннього зв'язування, побачать, що їм доводиться писати трохи більше коду.

– React не є реактивним (в сенсі реактивного програмування і архітектури, які в більшій мірі управляються подіями, володіють гнучкістю і швидкістю реакції) в початковому стані.

Висновки до розділу 2

Було описано поняття одно сторінкового додатку. Перелічені та описані технології які використовуються для реалізації вебдодатку.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДЛ 3. РЕАЛІЗАЦІЯ ПЛАТФОРМИ ДЛЯ ВЕБПУБЛІКАЦІЇ

3.1 Реалізація головної сторінки

Для написання програми буде використано такий інструмент розробки, як Next.js безкоштовний і відкритий JavaScript фреймворк, заснований на React, який дозволяє створювати вебдодатки з поліпшеною продуктивністю і покращеним досвідом за допомогою додаткових функцій попереднього рендеринга, таких як повноцінний рендеринг на стороні сервера (SSR) і статична генерація сторінок (SSG). Можливості Next.js цілком підходять для більшості проектів.

Таким чином додатки Next.js використовують всі переваги бібліотеки React і просто додають додаткові функції:

– Server Side Rendering. SSR дозволяє отримувати доступ до всіх необхідних даних для побудови сторінки на сервері. Потім сторінка повністю відправляється назад в браузер і відразу ж відображається. SSR дозволяє вебсторінкам завантажуватися за менший час і підвищує зручність роботи користувачів за рахунок підвищення швидкості відгуку.

– SEO або просто пошукова оптимізація. Використання SSR також дає вам перевагу в SEO, що допомагає вашому сайту займати більш високі позиції на сторінках результатів пошукових систем. SSR підвищує рейтинг вебсайтів для SEO, тому що вони завантажуються швидше і більше контенту сайту можна сканувати за допомогою трекерів SEO.

– Тег `<head>` Next.js також дозволяє вам редагувати тег `<head>` сайту, що ви не можете зробити в React. Тег є основною частиною метаданих вебсторінки і сприяє підвищенню SEO-рейтингу сайту.

Основна перевага Next.js - вбудована підтримка SSR для підвищення продуктивності і SEO. Візуалізація на стороні сервера працює шляхом зміни потоку запитів додатка React, так що всі компоненти, крім клієнта,

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

відправляють свою інформацію на сервер.

Маючи всю інформацію на сервері, клієнт може отримати попередню версію HTML-коду сторінки. Нижче наведено (рис. 3.1) із різновидами схем рендерінгу .

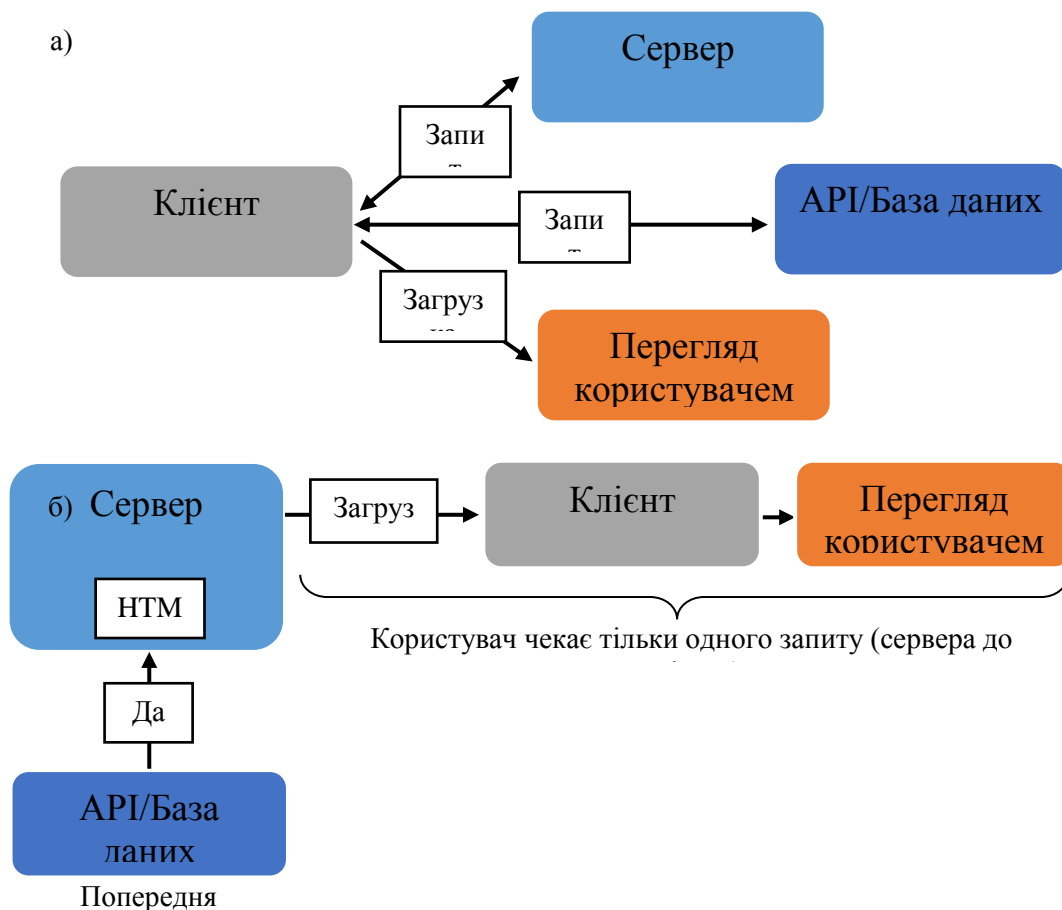


Рисунок 3.1 - Різновиди схем рендерінгу: а) зі сторони клієнта; б) зі сторони сервера

Клієнт може відправити один запит на сервер і отримати повноцінну HTML сторінку замість того, щоб запитувати кожен компонент окремо з рендерингом на стороні клієнта.

Переваги даного фреймворку:

- Додатки Next.js завантажуються значно швидше, ніж додатки React, завдяки вбудованому рендерингу на стороні сервера;
- Підтримує функції експорту статичних сайтів;
- Швидкий вхід, для тих хто вже працював з бібліотекою React.js;

- Автоматичне розділення коду для сторінок;
- Легко створювати внутрішні API-інтерфейси за допомогою вбудованих маршрутів API і створювати кінцеві точки API;
- Вбудована підтримка маршрутизації сторінок, CSS, JSX і TypeScript;
- Швидке додавання плагінів для настройки Next.js відповідно до потреб вашої сторінки;
- Підтримує такі переваги React, як інтуїтивно зрозуміле створення на основі компонентів, інтерфейсна система станів і висока популярність.

Єдиним реальним недоліком Next.js є те, що це самостійний фреймворк, тобто у нього є певний метод і набір інструментів, які він хоче, щоб ви використовували для створення своїх додатків.

3.2 Налаштування середовища розробки

Перед початком роботи потрібно буде створити каталог для вашого проекту та ініціалізувати його як сховище npm за допомогою такої команди:

```
npm init -y
```

Після цього вам потрібно буде встановити залежності для вашого проекту. У цьому підручнику я буду використовувати TypeScript та Sass.

Отже, поверх звичайних залежностей Next.js вам потрібно буде встановити всі типізації, а також `@zeit/next-typescript`, `@zeit/next-sass` та `node-sass`.

Для цього просто запустіть цю команду npm:

```
npm install --save react @types/react react-dom @types/react-dom next @types/next @zeit/next-typescript @zeit/next-sass node-sass
```

Вам також потрібно буде зробити кілька змін у конфігурації. Створіть файл у кореневій частині проекту з іменем `next.config.js` із таким кодом:

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

```
const withTypescript = require('@zeit/next-typescript')
const withSass = require('@zeit/next-sass')
module.exports = withTypescript(withSass());
```

Це вкаже Next.js, що ви хочете використовувати TypeScript та Sass у проєкті. Вам також потрібно буде створити файл з іменем `.babelrc.jsy` тому самому місці з таким кодом:

```
module.exports = {
  presets: ['next/babel', '@zeit/next-typescript/babel']
}
```

Крім того, у свій `package.json` файл додайте такі сценарії. Додавши ці сценарії, ви зможете локально обслуговувати свою програму за допомогою `npm run dev` команди. Не панікуйте, якщо це не спрацює на цьому етапі, оскільки ваша заявка ще не готова.

```
{
  "scripts": {
    "dev": "next",
    "build": "next build",
    "start": "next start"
  }
}
```

Тепер, коли ваше середовище готове до роботи, ви можете почати додавати сторінки на сайт. Всередині `pages` каталогу створіть `index.tsx` файл із таким кодом:

```
const Index = () => {
  return (
    <div className="app">
      <p>Diplom app</p>
      <p>Vitalii Melnyk app</p>
    </div>
  )
}
export default Index
```

На цьому етапі робота `npm run dev` в консолі має служити додаток за наступною адресою: `localhost:3000`.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

3.3 Генерування нових компонентів

Так як ви створюєте додаток електронної комерції, вам необхідно створити чотири основні компоненти всередині components каталогу: Header.tsx, Footer.tsx, ProductList.tsx і Product.tsx.

Усередині заголовка ви можете імпортувати next/link пакет, який дозволяє конвертувати більшість елементів HTML у посилання. У цьому випадку логотип та назва повернуть нас на домашню сторінку.

Майте на увазі, що Link компонент може мати лише один вкладений елемент HTML і повинен використовуватися лише для пересилання користувача на ваш вебсайт. Все що посилається поза вашим вебсайтом, повинно залишатися всередині атегу.

Крім того, використовуючи Next.js, ви можете обслуговувати будь-який статичний вміст, наприклад зображення, розміщуючи їх у static каталозі в кореневій папці.

Product компонент буде виводити будь яку інформацію ви хочете відобразити про конкретний продукт. Ви можете створити IProduct інтерфейс, який відповідає визначенню продукту Snipcart, та IProductProps інтерфейс для визначення типів нашого реквізиту, який передається як параметр функції.

```
import {withRouter, RouterProps} from 'next/router'
export interface IProduct {
  id: string
  name: string
  price: number
  url: string

  description: string
  image: string
}
interface IProductProps {
  product: IProduct
  router: RouterProps
}
```

Маршрутизатор дозволяє отримати URL адресу як на клієнті, так і на сервері, що чудово на багатьох рівнях 1), додаток Next.js може бути

					КР.ІПЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

відтворений на стороні сервера і 2). Snipcart зможе сканувати назад сторінки для перевірки цілісності продукту без будь яких проблем.

ProductList.tsx Компонент буде використовуватися для відображення списку товарів на головній сторінці. Тому ви можете створити IProductListProps інтерфейс, який описує масив, IProduct який врешті-решт буде переданий нашим вебсайтом.

React потрібно мати унікальний ключ для кожної дочірньої компоненти, яку ви вирішили відтворити більше одного разу. Ось чому вам потрібно передати індекс як підказку нашому Product компоненту.

```
import Product, { IProduct } from "./Product"
interface IProductListProps {
  products: IProduct[]
}
const ProductList = (props: IProductListProps) => {
  return (
    <div className="product-list">
      {props.products.map((product, index) => <Product product={product}
        key={index}/>)}
    </div> export default ProductList
```

Отримання даних та імпорт компонентів. На цьому етапі ви, мабуть, захочете заповнити свої продукти ProductList компонентом. Ви можете використовувати useEffect життєвий цикл React усередині, ProductList щоб заповнити дані. Однак це не відобразатиметься на сервері.

```
const Product = (props: IProductProps) => {
  return (
    <div className="product">
      <h2 className="product__title">{props.product.name}</h2>
      <p className="product__description">{props.product.description}</p>

      <img src={props.product.image} alt="" className="product__image"/>
      <div className="product__price-button-container">
        <div className="product__price">${props.product.price.toFixed(2)}</div>
        <button className="snipcart-add-item product__button"
          data-item-id={props.product.id}
          data-item-name={props.product.name}
          data-item-price={props.product.price}
          data-item-url={props.router.pathname}
          data-item-image={props.product.image}>
          Add to cart
        </button> </div> </div>
    </div>
  )
  export default withRouter(Product)
```

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

На щастя, Next.js додає новий метод життєвого циклу для іменованих сторінок `getInitialProps`, який є асинхронним методом, який може повернути все, що можна вирішити в JavaScript Object. Тут ви зазвичай хочете отримати з API або CMS.

У цьому випадку ви просто повернете новий об'єкт JavaScript, що містить список усіх продуктів.

Ви можете імпортувати щойно створені компоненти всередині `index.tsx` сторінки та додати `getInitialProps` метод, змінивши код на такий:

```
import Header from "../components/Header"
import ProductList from "../components/ProductList"
import { IProduct } from "../components/Product"
import Footer from "../components/Footer"
import Head from "next/head"
interface IIndexProps {
  products: IProduct[]}
const Index = (props: IIndexProps) => {
  return (
```

Тепер давайте встановимо Snipcart на вебсайт. По перше, вам потрібно буде імпортувати `Head` компонент з `next/head` вашої `index.tsx` сторінки, що дозволить вам додати HTML всередину `<head>` елемента. Ви можете зробити це, додавши наступний код всередині `Index` речення функції `return`. Не забудьте поміняти `data-api-key` атрибут своїм ключем API. Стиль програми

Наразі ви вже налаштували всі конфігурації, необхідні для використання Sass у вашому вебдодатку. Тому залишається лише створити `.scss` файл та імпортувати його на сторінку, яка вам подобається.

```
import "../styles.scss"
```

Тим не менш, Next.js пропонує безліч інших способів стилізації вебпрограми. Наприклад, ви можете додати вбудовані стилі або використовувати `styled-jsx`, який за замовчуванням входить до складу додатків Next.js. Розгортання програми, програмі Next.js існує два основних способи розгортання програми.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

```

<div className="app">
  <Header />
  <main className="main">
    <ProductList products={props.products} />
  </main> <Footer />
  Index.getInitialProps = async ({ req }) => {
  return { products: [
    {id: "nextjs_halfmoon", name: "Halfmoon Betta", price: 25.00, image:
    "./static/halfmoon.jpg", description: "The Halfmoon betta is arguably one of the
    prettiest betta species. It is recognized by its large tail that can flare up to 180
    degrees."} as IProduct,
  export default Index

```

Ви можете скористатися більш традиційним підходом, який відображається на стороні сервера, що чудово підходить для вебдодатків з великою кількістю динамічного вмісту, або експортувати кожен сторінку у .html файл і обслуговувати ці файли через мережу доставки вмісту.

Висновки до розділу 3

В розділі було розроблена платформа для вебпублікацій. А саме реалізація головної сторінки. Було налаштовано середовище розробки і генерування головних компонентів.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Умови праці на робочому місці з ПК

Навчальний процес з використанням комп'ютера, відбувається в середовищі, яке в певній мірі впливає на функціональний стан студентів, які перебувають у комп'ютерній аудиторії. Найважливішими несприятливими факторами середовища при роботі за комп'ютером є :

фізичні параметри мікроклімату; освітлення; електромагнітні випромінювання різних частотних діапазонів; виробничий шум та вібрація; іонізація повітря; статична електрика;

психофізіологічні перенапруження зорового аналізатора; недостатня рухома діяльність; нервово-емоційне напруження.

Аналіз умов праці виконується для приміщення аудиторії №330/4 яка знаходиться, на 3 поверсі.

Відповідно до НПАОП 0.00-1.28-10 є неприпустимим розташування приміщень, призначених для роботи з ВДТ у підвалах та цокольних поверхах. Також забороняється розташування вибухонебезпечних приміщень категорії А і Б (ОНТП 24-86) та виробництв з мокрими технологічними процесами поряд з приміщенням, де розташовуються ЕОМ (ПЕОМ), а також над такими приміщеннями, або під ними.

Окрім того, виробничі приміщення для роботи з ВДТ не повинні межувати з приміщеннями, у яких рівень шуму та вібрації перевищує допустимі значення.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці і відпочинку. У протилежному випадку в працюючого за ПК відзначаються значна напруга зорового апарата з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втома і хворобливі відчуття в очах, у попереку, в області шиї і руках.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

Розташування робочого місця особи, що працює з ПК, в даному випадку студента, повинне відповідати вимогам ДСТУ 22.269-79 «Робоче місце оператора».

Облаштування робочого місці повинно забезпечувати: правильне розміщення робочого місця у виробничому приміщенні;

належні умови освітлення приміщення і робочого місця, відсутність відблисків;

належні ергономічні характеристики основних елементів робочого місця; характер та особливості трудової діяльності.

Для зменшення втоми, місця користувачів ЕОМ мають бути зручними.

Конструкція робочого місця користувача ПК, (при роботі сидячи) має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками:

- ступні ніг на підлозі або на підставці для ніг;
- стегна в горизонтальній площині;
- передпліччя вертикально;
- лікті під кутом 70-90 до вертикальної площини;
- зап'ястя зігнуті під кутом не більше 20 відносно горизонтальної площини;
- нахил голови 15-20 відносно вертикальної площини.

Робоче місце користувача ПК, обладнується робочим столом, стільцем і підставкою для ніг. Висота робочого стола має бути в межах від 0,65 до 0,8 м, а ширина повинна забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Клавіатуру слід розташовувати на поверхні столу на відстані 200 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає його переміщенню), який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5...15.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

4.2 Мікроклімат та ступінь повітря виробничого приміщення.

Як фактор виробничого середовища, мікроклімат впливає на теплообмін організму людини з цим середовищем. Необхідною умовою життєдіяльності людини є терморегуляція, тобто здатність організму регулювати віддачу тепла в оточуюче середовище. Робоче місце користувача ПК, обладнується робочим столом, стільцем і підставкою для ніг. Висота робочого стола має бути в межах від 0,65 до 0,8 м, а ширина повинна забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Цей процес визначається параметрами мікроклімату. Метеорологічні умови визначаються такими параметрами:

- температурою повітря в приміщенні, С;
- відносною вологістю повітря %;
- рухливістю повітря м/с;
- тепловим випромінюванням, Вт/м³.

Принцип нормування мікроклімату створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Параметри мікроклімату, вміст шкідливих речовин на робочих місцях, оснащених моніторами, відповідають вимогам ДСТУ 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень».

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні.

В приміщеннях, де встановлені комп'ютери, повинні дотримуватися оптимальні параметри мікроклімату, які визначають комфортні умови. Ці параметри залежать від періоду року, категорії робіт за важкістю, і від теплових характеристик виробничого приміщення (табл. 4.1).

Робота за комп'ютером характеризується малими фізичним навантаженнями, цей вид діяльності належить до категорії легких робіт-Ia за критерієм загальних енерговитрат організму (ДСН 3.3.6.042-99)

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

Таблиця 4.1
Параметри мікроклімату

Період року	Категорія робіт	Параметр мікроклімату	Величина
Холодний	Роботи легкі I а	Температура повітря в приміщенні	22÷24°C
		Відносна вологість	40÷60%
		Швидкість руху повітря	до 0,1м/с
Теплий		Температура повітря в приміщенні	23÷25°C
		Відносна вологість	40÷60%
		Швидкість руху повітря	0,1,0,2м/с

Системи опалення та кондиціонування повітря в приміщеннях з ПК повинні відповідати. Під час роботи комп'ютерної техніки в повітряному середовищі відбувається суттєва трансформація іонного складу, істотно знижується концентрація легких, середніх та важких негативно зарядних частинок. Така зміна балансу іонного складу призводить до негативного впливу на здоров'я працюючих.

Рівні іонізації повітря приміщень при роботі на персональних комп'ютерах визначені в (табл. 4.2), (відповідно до НПАОП 0.03-3.06-80).

Таблиця 4.2
Рівні іонізації повітря в приміщенні з ПК

Рівні	Кількість іонів в 1 см ³ повітря	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально допустимі	50000	50000

Для підтримки оптимальних значень мікроклімату та підтримання нормальної концентрації позитивних та негативних іонів в приміщенні аудиторії пропонується удосконалити системи опалення, природної вентиляції

та встановити кондиціонер.

Дисплеї на основі ЕПТ є потенційним джерелом випромінювання кількох діапазонів електромагнітного спектра: рентгенівського, оптичного, радіочастотного. Кожний вид випромінювання відрізняється своїми особливими характеристиками впливу на організм людини.

Рентгенівське випромінювання. Дослідження показують, що відео термінал не несе небезпеки для користувача ПЕОМ, оскільки інтенсивність такого випромінювання нижча за гранично допустимі норми (ГДН). Відповідно до «Норм радіаційної безпеки України» (НРБУ-97) гранично допустима потужність експозиційної дози рентгенівського випромінювання на відстані 5 см від поверхні екрана відеотермінала становить $7,74 \cdot 10^{-12}$ Кл/кг, що відповідає еквівалентній дозі 0,1 мбер/год. (100 мкР/год.).

Оптичне випромінювання. Оптичні види випромінювання виникають завдяки взаємодії електронів з шаром люмінофору, нанесеного на екран ВДТ. Область оптичного випромінювання включає ультрафіолетове (УФ), світлове та інфрачервоне (ІЧ) випромінювання.

УФ-випромінювання впливає на шкіру та очі людини. Такий вплив на шкірі проявляється досить швидко, а для очей характерним є період прихованої дії. Рівень УФ- випромінювання, який був виявлений, досить низький і становить 1 середньому $0,001 \text{ Вт/м}^2$.

Світлове випромінювання впливає в основному на око і призводить до втоми очей, запалення райдужної оболонки. Однак ці симптоми швидко минають і не викликають патологічних змін.

ІЧ-випромінювання - довжина хвиль обмежена від 0,76мм до 1мм. Для більшості біологічних матеріалів випромінювання цього діапазону вважаються непрозорими. Інтенсивність інфрачервоних випромінювань нижча за значення, передбачені ДсанПіН 3.3.2. 007-98.

Електромагнітні випромінювання (ЕМВ) радіочастотного діапазону. Джерелом ЕМВ є відеотермінал. Тому, обираючи робоче місце для комп'ютера, необхідно пам'ятати, що його задня і бокові стінки можуть бути джерелом

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

значно більшого ЕМВ, ніж екран.

З метою профілактики несприятливого впливу електромагнітного випромінювання від ВДТ на користувача необхідно:

- встановити на робочому місці відеотермінал, що відповідає сучасним вимогам стосовно захисту від випромінювань;
- не переобтяжувати приміщення значною кількістю робочих місць з ВДТ;
- не концентрувати на робочому місці великої кількості радіоелектрон-них пристроїв.

Гранично допустимі рівні випромінювань на робочих місцях з ПК. Відомо, що шум несприятливо діє на слуховий аналізатор та інші органи та системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності. Ті види діяльності, у яких поєднується напружена розумова робота та інтенсивне використання комп'ютера (редагування тексту, верстка оригіналу, "запуск" та відлагодження програм тощо) характеризується відчутним впливом навіть незначних рівнів шуму. Цей вплив виражається у зниженні розумової працездатності, швидкій втомлюваності, послабленні уваги, появі головного болю та ін.

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ і ПК визначені ДСанПіН 3.3.2-007-98 не повинні перевищувати 65 дБА. Для приміщення аудиторії, рівень шуму не перевищує 60 дБА.

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

Рівні вібрації під час виконання робіт з ЕОМ у виробничих приміщеннях не повинні перевищувати допустимих значень 70 дБ, визначених 12.1.012-90 та СН 3044-84 "Санітарні норми робочих місць"

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

Для зниження вібрації обладнання, пристрої, пристосування необхідно встановлювати на спеціальні амортизуючі прокладки, передбачені документами.

Близько 90% всієї інформації, що отримується людиною, приходиться на органи зору. Організація освітленості робочих місць грає велику роль у житті людини. Недостатнє та нераціональне освітлення веде до втомлення очей, розладу центральної нервової системи, зниженню розумової та фізичної працездатності, а у ряді випадків може бути причиною травматизму (близько 5% травм приходиться на частку нераціонального та недостатнього освітлення).

Щоб уникнути перевтоми, а також для профілактики професійних захворювань та виробничого травматизму потрібно дотримуватись наступних вимог:

- створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;
- забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частоті преадаптації органів зору;
- не створювати засліплювальної дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;
- не створювати на робочій поверхні різких та глибоких тіней (особливо рухомих);
- повинен бути достатній для розрізнення деталей контраст поверхонь, що освітлюються;
- не створювати небезпечних та шкідливих виробничих чинників (шум, теплові випромінювання, небезпека ураження струмом, пожежо- та вибухонебезпека світильників);
- повинно бути надійним і простим в експлуатації, економічним та естетичним .

Приміщення аудиторії №330/4 має природну та штучну системи

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

освітлення, які повинні відповідати вимогам ДБН В.2.5-28-2996 «Інженерне обладнання будинків та споруд. Природне та штучне освітлення».

Штучне освітлення в приміщенні з комп'ютеризованим робочим місцем здійснюється системою загального освітлення.

Як джерела світла в разі штучного освітлення застосовуються світильники серії ЛПО353. Світильники укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА). Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 град. з вертикаллю в повздовжній та поперечній площинах становить не більше 200 кд/м², захисний кут світильників не менше ніж 40 град.

Рівень освітленості у аудиторії, $E = 300$ лк.

Електробезпека система організаційних та технічних заходів і засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

Приміщення аудиторії №330/4 відноситься до приміщень з підвищеною безпекою.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ, електропроводи та кабелі за виконанням та ступенем захисту мають відповідати класу зони за ПВЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Використання нульового робочого провідника як нульового захисного провідника забороняється.

Нульовий захисний провід прокладається від склейки групового розподільчого щита до розеток живлення.

Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі повинна бути не менше площі перерізу фазового провідника. Усі провідники повинні відповідати номінальним параметрам

					КР.ІПЗ – 13.00.000 ПЗ	Арк.
						60
Зм.	Арк.	№ докум.	Підпис	Дата		

мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ПВЕ.

З метою підвищення рівня електробезпеки в приміщенні аудиторії №330/4 неприпустимим є:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- залишення під напругою неізольованих кабелів та проводів;
- застосування саморобних подовжувачів, які не відповідають вимогам ПВЕ до переносних електропроводок;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання.

ВДТ є джерелом електростатичних зарядів. Тривале перебування в електростатичному полі, створеному цими зарядами, негативно впливає на здоров'я працюючих: бронхо-легеневі захворювання, порушення серцево-судинної та нервової систем, ураження шкіри тощо .

Напруженість електростатичного поля на робочих місцях, в тому числі й з ВДТ, не повинна перевищувати 20 кВ/м відповідно до 12.1.045-84 "ССБТ.

Для запобігання створенню значної напруженості поля та захисту від статичної електрики необхідно:

- встановити нейтралізатори статичної електрики;
- підтримувати в приміщенні з ВДТ відносну вологість повітря не нижче 45-50% (чим сухіше повітря тим більше електростатичних зарядів); можна для цього використати навіть побутові зволожувачі;
- застелити підлогу в приміщеннях з ВДТ антистатичним лінолеумом і проводити щоденне вологе прибирання;

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

- складати всі полімерні покриття (чохли) ВДТ у найбільш віддаленому від користувачів місці розміщення;
- протирати екран та робоче місце спеціальною антистатичною серветкою або зволоженою тканиною.

4.3 Вимоги до безпечної експлуатації комп'ютерної техніки

Пожежна безпека при роботі за комп'ютером повинна забезпечуватись у відповідності з вимогами Закону України «Про пожежну безпеку» та НПАОП 0.01-1.01-95 «Правил пожежної безпеки в Україні» та інших нормативних документів.

Найбільш імовірною причиною виникнення пожеж в приміщенні є порушення вимог при експлуатації комп'ютерної техніки та займання електропроводки внаслідок коротких замикань.

Для того щоб уникнути виникнення пожежі, потрібно дотримуватися наступних заходів:

- дотримання правил пожежної безпеки при роботі з комп'ютером, електрообладнанням та освітлювальними приладами;
- періодичний контроль цілісності і надійності електроізоляції;
- наявність інструкцій з пожежної безпеки;
- навчання, атестація і переатестація персоналу з пожежної безпеки;
- наявність системи захисту від атмосферної електрики;
- періодичне зняття зарядів статичної електрики;
- заборона куріння в приміщенні.
- застосування будівельних конструкцій із ступенем вогнестійкості не нижче II, а також використання важко горючих або негорючих матеріалів в інтер'єрі виробничого приміщення;
- наявність схеми евакуації;
- наявність пристроїв автоматичного вимкнення ПЕОМ та іншого

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

електроустаткування на випадок пожежі;

– наявність первинних засобів пожежогасіння (вогнегасник ВВК-5-2шт).

Висновки до розділу 4

Було розібрано основні правила безпеки роботи за персональним комп'ютером, та необхідні умови в приміщенні в якому проводиться робота. Особливу увагу було надано до правил експлуатації ПК.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В кваліфікаційній роботі було розроблено платформу для електронних публікацій. Під час виконання роботи, було реалізований особистий кабінет для реєстрації з можливістю створити і доповнити свій профайл та детальну інформацію про теперішнє місце роботи, спілкуватись з іншими учасникам платформи та залишати відгуки. Також розроблено адміністративний кабінет роботодавця з можливістю переглядати профайл усіх учасників, наповнення ресурсу контентом та обробки коментарів.

Даний ресурс розроблено відповідно до потреб потенційних користувачів. Потреби користувачів визначались методами порівняльного аналізу. Інтерфейс системи розроблено згідно створеного макету.

Сайт розроблений за допомогою мови Nodejs з розширенням Express для з'єднання з базою даних. Вибір даної мови був зумовлений тим, що вона є ефективною і добре працює з даними. В якості бази даних використана нереляційна база MongoDB.

Розробка web-ресурсу дала змогу розширити свої знання з розробки вебзастосунків. Починаючи з аналізу потреб користувача і завершуючи проектуванням бази даних.

Детально описано усі етапи розробки сайту, від планування архітектури та бази даних до розробки макетів та інтерфейсів. Здійснено огляд функціональних можливостей розробленої системи.

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MVC. *Wikipedia*: вебсайт. URL: <https://ru.wikipedia.org/wiki/Model-View-Controller>
2. Клієнт-серверна архітектура. *Wikipedia*: вебсайт. URL: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура
3. JavaScript. *Wikipedia*: вебсайт. URL: <https://uk.wikipedia.org/wiki/JavaScript>
4. JWT. *jwt.io*: вебсайт. URL: <https://jwt.io/introduction>
5. React. *Wikipedia*: вебсайт. URL: <https://ru.wikipedia.org/wiki/React>
6. Node.js. *Wikipedia*: вебсайт. URL: <https://ru.wikipedia.org/wiki/Node.js>
7. Express.js. *Wikipedia*: вебсайт. URL: <https://ru.wikipedia.org/wiki/Express.js>
8. MongoDB. *Wikipedia*: вебсайт. URL: <https://ru.wikipedia.org/wiki/MongoDB>
9. Empathy Mapping: The First Step in Design Thinking URL: <https://www.nngroup.com/articles/empathy-mapping/>
10. Структурні елементи JavaScript. URL: https://developer.mozilla.org/ru/docs/Learn/JavaScript/Building_blocks/Events
11. Майкл Миковски, Джош Пауэлл. Разработка одностраничных вебприложений. URL: <https://dmkpress.com/catalog/computer/web/978-5-97060-072-6/>
12. Usage statistics of JavaScript as client-side programming language on websites. URL: <https://w3techs.com/technologies/details/cp-javascript>
13. The session data is kept server-side, the ID is saved as a HTTP Cookie. URL: Режим доступу: <https://docs.microsoft.com/en-us/previous-versions/officedeveloper-servertechnologies>

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

14. Embedding NodeJS in HTML. O'Reilly. URL: <https://www.oreilly.com/radar/>.

15. Services Mashups: The New Generation of Web Applications. URL: https://dsg.tuwien.ac.at/Staff/sd.papers-ServicesMashups_IC.pdf

					КР.ІІЗ – 13.00.000 ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66