UNICHECK

Ім'я користувача:
приховано налаштуваннями конфіденційності

ID перевірки:
1015591211

Дата перевірки:
13.06.2023 20:14:13 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
13.06.2023 20:15:13 EEST

ID користувача:
100011372

Назва документа: ОК42 Остапчук

Кількість сторінок: 52 Кількість слів: 10385 Кількість символів: 88439 Розмір файлу: 452.72 KB ID файлу: 1015240402

8.01%
Схожість

Найбільша схожість: 5.76% з джерелом з Бібліотеки (ID файлу: 1015240367)

Пошук збігів з Інтернетом не проводився

8.01% Джерела з Бібліотеки | 136 ..... Сторінка 54

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи | 792

1 ОГЛЯД СУЧАСНИХ CMS СИСТЕМ

1.1 Класифікація та характеристики сучасних CMS

Розпочнімо з того, що ж значить CMSсистема? CMSсистема або ж Система управління контентом (CMS) - це програмне забезпечення, яке дозволяє користувачам створювати, керувати та публікувати цифровий контент. Зазвичай CMS надає інструменти для створення, редагування та публікації контенту, а також функції для організації, зберігання та пошуку контенту. У цій статті ми розглянемо ключові особливості CMS, різні типи CMS, їхні переваги та недоліки, а також приклади популярних платформ CMS.

CMS надає зручний інтерфейс для створення та редагування контенту без потреби в технічних знаннях. Вона включає в себе інструменти для форматування тексту, додавання зображень і відео, а також управління мультимедійним контентом.

Зберігання контентузберігає контент у централізованій базі даних, що дозволяє користувачам легко отримувати до нього доступ і вилучати його. Вміст можна організувати за категоріями, тегами або папками, що полегшує його пошук і керування.

Публікація контенту CMS дозволяє користувачам публікувати контент на різних цифрових каналах, таких як веб-сайти, мобільні додатки або платформи соціальних мереж, одним клацанням миші.

Керування користувачами: CMS надає функції для управління користувачами та призначення ролей і дозволів. Це дозволяє адміністраторам контролювати, хто має доступ до контенту та може його редагувати.

Управління робочим процесом: CMS надає інструменти для управління процесом створення контенту, включаючи робочі процеси рецензування та затвердження. Це допомагає забезпечити високу якість контенту та його відповідність стандартам організації.

Аналітика та звітність CMS надає функції аналітики та звітності, які дозволяють користувачам відстежувати ефективність контенту, наприклад, перегляди сторінок, залучення та конверсії.

Типи CMS. Існує кілька типів CMS, кожен з яких має свої сильні та слабкі сторони.CMS з відкритим кодом: Платформи CMS з відкритим кодом є безкоштовними у використанні і можуть бути налаштовані відповідно до потреб користувача. Прикладами CMS з відкритим кодом є WordPress, Joomla та Drupal.

Пропрієтарні CMS: Пропрієтарні CMS-платформи - це комерційні продукти, для використання яких потрібна ліцензія. Вони надають більше можливостей і підтримки, але можуть бути дорожчими. Прикладами пропрієтарних CMS є AdobeExperienceManager та Sitecore.

Хмарні CMS: Хмарні CMS-платформи розміщуються в хмарі, що дозволяє користувачам отримувати доступ до них з будь-якого місця, де є інтернет-з'єднання. Прикладами хмарних CMS є Wix, Squarespace і Shopify.

Переваги CMS:

- Простота створення контенту: CMS надає зручний інтерфейс для створення та редагування контенту без потреби в технічних знаннях. Це полегшує створення та публікацію контенту для нетехнічних користувачів.
- Управління контентом: CMS надає інструменти для організації, зберігання та пошуку контенту. Це дозволяє легко керувати великими обсягами контенту і забезпечувати його актуальність та релевантність.
- Управління робочими процесами: CMS надає інструменти для управління процесом створення контенту, включно з робочими процесами рецензування та затвердження. Це допомагає забезпечити високу якість контенту та його відповідність стандартам організації.
- Багатоканальна публікація: CMS дозволяє користувачам публікувати контент на різних цифрових каналах, таких як веб-сайти, мобільні додатки або платформи соціальних мереж, одним кліком. Це економить час і забезпечує узгодженість між каналами.

- Кастомізація: CMS можна налаштувати відповідно до потреб користувача. Це включає в себе налаштування зовнішнього вигляду веб-сайту, додавання нових функцій або інтеграцію з іншими системами.

Недоліки CMS:

- Складність: Платформи CMS можуть бути складними в налаштуванні та управлінні, що вимагає технічних знань. Це може бути бар'єром для малого бізнесу або нетехнічних користувачів.
- Безпека: Платформи CMS можуть бути вразливими до загроз безпеки, таких як хакерські атаки або шкідливе програмне забезпечення. Це вимагає регулярних оновлень і патчів безпеки, щоб забезпечити безпеку платформи.
- Продуктивність: Платформи CMS можуть бути повільними та ресурсоємними, особливо для великих веб-сайтів з великою кількістю контенту. Це може вплинути на продуктивність веб-сайту та зручність користування ним.
- Вартість: деякі платформи CMS можуть бути дорогими, особливо пропрієтарні CMS, які вимагають ліцензії навикористання. Це може бути перешкодою для малого бізнесу або неприбуткових організацій з обмеженим бюджетом.
- Обмеження кастомізації: Хоча платформи CMS можна налаштовувати, можуть існувати обмеження на те, що можна налаштувати без технічних знань або додаткових ресурсів для розробки.

Системи управління контентом (CMS) змінили спосіб, у який організації створюють, керують і поширюють цифровий контент у наш час. Їхні зручні інтерфейси, ефективні процеси роботи з контентом, можливості кастомізації, масштабування та інтеграції зробили їх необхідними інструментами для бізнесу в різних галузях. CMS-системи дають користувачам змогу ефективно керувати контентом, оптимізувати співпрацю та створювати цікавий досвід для своєї цільової аудиторії.

1.2 Огляд популярних CSM систем

1. WordPress

WordPress - це популярна система управління контентом (CMS), на якій працюють понад 40% веб-сайтів в Інтернеті. Її створили МеттМулленвег і Майк Літтл у 2003 році як форк існуючої платформи для ведення блогів під назвою b2/cafelog. З того часу WordPress перетворився на універсальну CMS, яку можна використовувати для створення веб-сайтів, блогів та інтернет-магазинів. На рис. 1.1 зображенолого програми.

Рисунок 1.1 – Логотип Wordpress

Мулленвег і Літтл почали працювати над WordPress, щоб покращити b2/cafelog, якому, на їхню думку, не вистачало функцій і гнучкості. Вони вирішили випустити свою вдосконалену версію платформи як проект з відкритим вихідним кодом, що означало, що будь-хто може використовувати, модифікувати та розповсюджувати програмне забезпечення безкоштовно.

Однією з ключових особливостей WordPress є простота використання. Навіть люди, які не мають досвіду кодування чи веб-розробки, можуть використовувати WordPress для створення та управління своїми веб-сайтами. Платформа дуже добре налаштовується, з тисячами безкоштовних і платних тем і плагінів, які дозволяють користувачам змінювати зовнішній вигляд і функціональність своїх сайтів.

WordPress також є дуже масштабованою платформою, що робить її придатною як для невеликих особистих блогів, так і для великих корпоративних

веб-сайтів. Відкритий характер платформи означає, що розробники можуть створювати власні плагіни та теми для розширення можливостей платформи та задоволення конкретних потреб своїх клієнтів.

Ще однією значною перевагою WordPress є потужна підтримка спільноти. Користувачі мають доступ до безлічі ресурсів, включаючи документацію, форуми та навчальні посібники, які допоможуть їм отримати максимальну віддачу від платформи. Крім того, велика кількість користувачів WordPress означає, що вразливості безпеки швидко виявляються і виправляються, гарантуючи, що користувачі можуть захистити свої веб-сайти.

Однією з найбільших переваг WordPress є те, що він є безкоштовним у використанні. Однак користувачам доведеться платити за веб-хостинг і будь-які преміум-теми або плагіни, які вони вирішать використовувати. Вартість веб-хостингу може варіюватися в залежності від потреб користувача, але є багато доступних варіантів.

Плюси:

- Простий у використанні та зручний інтерфейс
- Велика спільнота розробників та користувачів для підтримки
- Легко налаштовується завдяки великому вибору доступних тем
- Безкоштовний і з відкритим вихідним кодом
- Добре підходить для SEO та управління контентом

Мінуси:

- Може бути повільним та ресурсоємним
- Уразливості безпеки можуть вимагати додаткових плагінів
- Не завжди сумісний з певними темами або плагінами
- Обмежена вбудована функціональність для сайтів електронної комерції та членських сайтів
- Може вимагати частих оновлень та обслуговування

Загалом, WordPress - це чудова платформа CMS, яка пропонує широкий спектр функцій та переваг. Простота використання та універсальність роблять її чудовим вибором як для початківців, так і для досвідчених користувачів. Drupal

2. Drupal

Drupal - це система управління контентом (CMS), яка є вільним програмним забезпеченням з відкритим вихідним кодом. Її створив у 2001 роціДрісБуйтаерт, коли був студентом Антверпенськогоуніверситету. СпочаткуБуйтаерт назвав її "Drop", щобулоабревіатуроювід "DrupalObject-OrientedProgrammingtool" (об'єктно-орієнтованийінструментпрограмуванняDrupal). Пізнішеназву "Drupal" булообрано як варіаціюголландського слова "druppel", щоозначає "крапля".На рис 1.2 зображено логотип Drupal.

Рисунок 1.2 – Логотип Drupal

СпочаткуDrupalбуврозроблений як система дошкиоголошеньпідназвою "Розширений форум", але пізнішеБуйтаерт додав іншіфункції, такі як агрегатор новин і спільна книга, щобзробитийогобільш комплексною системою управління контентом. Перша версіяDrupalбулавипущена в 2001 році, і з тих пір вона постійнорозвивається.

Однією з унікальнихособливостейDrupal є те, щовін написаний на PHP - серверніймовісценаріїв, яка зазвичайвикористовується для створеннядинамічних веб-сторінок. Drupalтакожвикористовуємодульнуархітектуру, щоозначає, щойогофункціональністьможе бути розширена за рахуноквикористаннясторонніхмодулівабонаписаннявласного коду.

Drupal широко використовується для створення веб-сайтів, блогів та веб-додатків. Деякі з йоговизначнихособливостейвключають:

Гнучкість і масштабованість: Drupalдужегнучкий і масштабований, щодозволяєвикористовуватийого для широкого спектру проектів, від невеликих веб-сайтів до масштабнихдодатків.

Управління контентом: Drupalнадаєкомплексну систему управління контентом, яка дозволяєкористувачам легко створювати, керувати та публікувати контент.

Безпека: Drupalвідомийсвоїминадійнимифункціямибезпеки, що робить йогопопулярнимвибором для веб-сайтів, якіпотребуютьвисокогорівнязахисту.

Підтримкаспільноти: Drupalмає велику та активнуспільнотурозробників та користувачів, якіроблятьсвійвнесок у йогопостійнийрозвиток та надаютьпідтримкуіншим.

Плюси:

- Високанастроюваність і гнучкість
- Добре підходить для створенняскладнихсайтів та додатків
- Велика спільнотарозробників і користувачів для підтримки та ресурсів
- Безкоштовний і з відкритимвихідним кодом
- Добре підходить для SEO та управління контентом

Мінуси:

- Тяжке в навчанні і можевимагатитехнічноїекспертизи
- Ресурсомістка і можевимагативиділеного сервера
- Обмеженийвибір тем і плагінів у порівнянні з іншими CMS
- Не завждисумісна з певними модулями абоплагінами

Отже, Drupal - цепотужна і гнучка система управління контентом, яка розвивається з моменту їїстворення у 2001 році. Вона широко використовується для створення веб-сайтів, блогів та веб-додатківзавдякисвоїйкомплекснійсистеміуправління контентом, функціямбезпеки, масштабованості та підтримціспільноти. Як програмнезабезпечення з відкритимвихідним кодом, Drupal є безкоштовним для завантаження та використання, але вартістьйоговикористанняможеваріюватися в залежностівідрізнихфакторів.

3. Joomla

Joomla - це популярна система управління контентом (CMS) з відкритимвихідним кодом, яка використовується для створення та управління веб-сайтами. Вперше вона булавипущена в 2005 році і написана на мовіPHP. ТворцямиJoomla є групарозробників і волонтерів, якібуличастиною проекту Mambo, іншоїпопулярноїCMS. Команда вирішиларозгалузитиMambo і створитиJoomla як проект, керованийспільнотою, якийбув би більшдоступним і простим у використанні. На рисунку 1.3 зображено логотип Joomla.

Рисунок 1.3 – Логотип Joomla

До основноїкомандирозробниківJoomla входили ЕндрюЕдді, колишнійпровіднийрозробникMambo, та Йохан Янссенс, якийтакож брав участь у проектіMambo. Вони працювали над розробкоюJoomla разом з групоюволонтерів, якідопомагалитестувати, документувати і перекладатипрограмнезабезпечення.

ВпершеJoomlабулавипущена у версії 1.0 у вересні 2005 року і швидкозавоювалапопулярність як безкоштовна альтернатива іншимкомерційним CMS з відкритим кодом. За ці роки Joomla пережила кілька великих оновлень і випусків, останняверсія - Joomla 4, випущена в серпні 2021 року.

СьогодніJoomlamає велику та активнуспільнотурозробників, дизайнерів та користувачів, якіроблятьсвійвнесок у їїпостійнийрозвиток та вдосконалення. Проектом Joomlaкерує Команда лідерів проекту Joomla, яка стежить за розвиткомпрограмногозабезпечення і гарантує, щовонозалишаєтьсявірнимсвоїм принципам відкритого коду.

Joomla доступна безкоштовно, і немаєніякихліцензійнихплатежів. Однаккористувачамможезнадобитисязаплатити за веб-хостинг, реєстрацію домену та будь-якіпреміум-розширенняабошаблони, які вони бажаютьвикористовувати.

Joomlaмає велику спільнотурозробників, якістворюютьрозширення і шаблони, якіможназавантажити і встановити на веб-сайтибезкоштовноабо за певну плату.

Joomlaмає широкий спектр функцій і можливостей, що робить їїчудовимвибором для створення веб-сайтів. Вона маєзручнийінтерфейс, якийдозволяєкористувачам легко створювати, редагувати та керувати контентом. Joomlatакожмаєвбудованиймедіа-менеджер, якийдозволяєкористувачамзавантажувати і керуватизображеннями, відео та іншими типами файлів. Крім того, Joomlaмаєнадійніфункції управліннякористувачами, якідозволяютьадміністраторамконтролювати, хтоможемати доступ до контенту на сайті та редагуватийого.

Joomlatакожмаєпотужну систему розширень, яка дозволяєкористувачамрозширюватифункціональністьсвоїх веб-сайтів. Для Joomlaдоступнітисячірозширень, включаючиплагіни, модулі та компоненти.

Плюси:

- Зручнийінтерфейс
- Добре підходить для створенняскладнихсайтів та додатків
- Велика спільнотарозробників і користувачів для підтримки та ресурсів
- Високаступінькастомізації з великим виборомдоступнихрозширень
- Безкоштовний і з відкритимвихідним кодом

Мінуси:

- Не така популярна, меншукількістьдоступнихресурсів та підтримки
- Деякіфункціїможутьвимагатидодатковихресурсів для розробки
- Обмеженийвибіршаблонів у порівнянні з іншими платформами CMS
- Може бути ресурсномістким і вимагативиділеного сервера
- Можевимагатичастихоновлень та обслуговування

На закінчення, Joomla - цепотужна і гнучкаCMS, яка підходить для створення веб-сайтіввсіхтипів і розмірів. Завдякивідкритому коду та великійспільнотірозробників вона є економічновигіднимрішенням для бізнесу та

приватнихосіб. Завдякизручномуінтерфейсу, надійнимфункціям і широкійсистемірозширень, Joomla є популярнимвибором для розробки веб-сайтів.

4. Magento

Magento - це популярна платформа електронноїкомерції з відкритимвихідним кодом, яку запустили у 2008 році Рой Рубін та ЙоавКатнер. Платформа швидкозавоювалапопулярністьсередінтернет-магазинівзавдякисвоїйгнучкості, масштабованості та потужнимфункціям. У 2011 роціeBayпридбавMagento, щощебільшезміцнилоїїпозиції як провідноїплатформиелектронноїкомерції.На рис 1.4 зображено логотип компанії.

Рисунок 1.4 – Логотип Magento

ТворціMagento**виявилипрогалинунаринкуплатформиелектронної**комер якабулабпростоюувикористанні, аледостатньопотужною, щобзадовольнитипотребивеликихпідприємств. Magentобуларозроблена, щоб бути гнучкою та настроюваною, дозволяючироздрібнимторговцям легко створювати та керуватисвоїмиінтернет-магазинами. З того часу платформа перетворилася на одне з найбільшнадійних і багатофункціональнихрішень для електронноїкомерції.Magento поставляється у двохредакціях: Community Edition та Enterprise Edition. Community Edition є безкоштовною у використаннітаорієнтовананамалийтасереднійбізнес, в тойчасяк Enterprise Edition - цепреміум-версія, якапризначенадлявеликихпідприємств. Enterprise Edition пропонуєдодатковіфункції, такіяксегментаціяклієнтів, цільовіпромо-акціїтапостановкаконтенту.

Magentoпропонує широкий спектр функцій, якіроблятьйого потужною платформою для електронноїкомерції. Деякі з цихфункційвключають:

Кастомізація: Magentoпропонуєвисокийступінькастомізації, щодозволяє ритейлерам пристосовуватисвоїінтернет-магазини до їхніхконкретних потреб.

Масштабованість: Magentorозроблена так, щобмасштабуватися разом з вашим бізнесом, а цеозначає, що вона можевпоратися з високимрівнемтрафіку та великими обсягамитранзакцій.

SEO-дружність: Magentooптимізовано для пошукових систем, щополегшує ритейлерам підвищення рейтингу в пошукових системах та залученнятрафіку до їхніхінтернет-магазинів.

Адаптивність для мобільнихпристроїв: Magentoпропонуєадаптивнішаблони дизайну, щоозначає, що ваш інтернет-магазин буде оптимізований для мобільнихпристроїв.

Інтеграція з платіжними шлюзами: Magentoінтегрується з широким спектром платіжнихшлюзів, щополегшує ритейлерам прийомплатежіввідсвоїхклієнтів.

Плюси:

- Високагнучкість та можливістьналаштування для комерції
- Добре підходить для SEO та управління контентом
- Велика спільнотарозробників та користувачів для підтримки
- Вбудованіфункції для управління продуктами
- Безкоштовний та з відкритимвихідним кодом

Мінуси:

- Тяжке в навчанні і можевимагатитехнічноїекспертизи
- Може бути ресурсоємним і вимагативиділеного сервера
- Обмеженийвибір тем і плагінів у порівнянні з іншими платформами
- Не таказручна для користувача, як деякііншіплатформи CMS
- Можевимагатичастихоновлень та обслуговування

Отже, Magento - цепотужна та гнучка платформа для електронноїкомерції, яка стала найкращимвибором для роздрібнихторговцівусіхрозмірів.. У той час як корпоративна версіяможе бути дороговартісною, громадськаверсія є

безкоштовною, що робить її доступною для бізнесу, якийтількипочинає свою діяльність.

5. Shopify

Shopify - це популярна платформа електронноїкомерції, яка дозволяєпродавцямстворюватиінтернет-магазини та керувати ними. Їїзаснували у 2006 роціТобіас Лютке, ДаніельВейнанд і Скотт Лейк, які шукали спосібстворитиінтернет-магазин для свого магазину сноубордів в Оттаві, Канада. Троєзасновниківбулинезадоволенідоступними на той час рішеннями для електронноїкомерції і вирішилистворитивласну платформу.На рис 1.5 зображено логотип компанії.

Рисунок 1.5 – Логотип Shopify

Перша версія Shopify була випущена в 2006 році і швидко завоювала популярність серед малого та середнього бізнесу завдяки простоті використання та доступній ціновій політиці. З роками Shopify перетворився на одну з найбільших платформ електронної комерції у світі, яка обслуговує понад 1,7 мільйона підприємств у більш ніж 175 країнах.

На додаток до своєїосновноїплатформиелектронноїкомерції, Shopifyтакожрозробив ряд іншихпродуктів і послуг. До них відносятьсяShopifyPayments, якийдозволяєпродавцямприйматиплатежібезпосередньо через свій магазин Shopify, а такожShopify Capital, якийнадаєфінансуваннявідповіднимпродавцям на основіїхньоїісторіїпродажів.

Shopifyтакожздійснив ряд придбаньпротягомбагатьохроків, щоброзширитисвоїможливості та пропозиції. Наприклад, у 2018 роцівінпридбав онлайн-маркетплейс Oberlo, якийдозволяєпродавцям легко імпортуватитовари з

AliExpress та іншихпостачальників у свій магазин Shopify. Вона такожпридбала низку іншихкомпаній, щопрацюють у таких сферах, як доповненареальність, виробництвовідео та аналітикаданих.

Однією з ключовихособливостейShopify є простота використання. Платформа надаєзручнийінтерфейс, якийдозволяєпродавцямстворитиінтернет-магазин без будь-якихтехнічнихнавичок. Вона такожпропонує широкий вибіршаблонів і тем, якіпродавціможутьвикористовувати для налаштуваннязовнішньоговиглядусвого магазину.

Shopifyтакожнадає широкий спектр інструментів і функцій, якідопомагаютьпродавцямкеруватисвоїмінтернет-магазином. Крім того, Shopifyмаєпотужний магазин додатків, якийнадаєпродавцям широкий вибірсторонніхдодатків.

Щостосуєтьсяціни,Shopifyпропонує ряд тарифнихпланів, яківідповідаютьрізнимбізнес-потребам і бюджетам. Базовий план починаєтьсявід $29 на місяць і включаєвсіосновніфункції, необхідні для роботиінтернет-магазину. Розширений план, якийкоштує 299 доларів на місяць, включаєдодатковіфункції, такі як розширенізвіти та тарифи на доставку відтретіхсторін.

Плюси:

- Зручнийінтерфейс і простий у налаштуванні
- Добре підходить для створеннясайтівелектронноїкомерції
- Вбудованіфункції для управління продуктами
- Велика спільнотарозробників і користувачів для підтримки та ресурсів
- Не потребує хостингу читехнічногообслуговування

Мінуси:

- Обмежені можливості кастомізації у порівнянні з іншими CMS
- Додаткова плата за певніфункціїабостороннідодатки
- Не такагнучка для не-сайтівелектронноїкомерції
- Може бути дорогим для великих сайтівабо великих обсягівпродажів
- Обмежений контроль над хостинговимсередовищем

Загалом, Shopifyдокоріннозмінивспосіб продажу товарів в Інтернеті, надавшипотужну та зручну платформу, доступну для бізнесу будь-якогорозміру. Простота використання, доступність та широкий спектр функційзробилийогопопулярнимвибором для компаній, якібажаютьпродавати онлайн.

2 ВИБІРЗАСОБІВРОЗРОБКИCMSСИСТЕМИ

2.1 Вибір фреймворків та розширень

Фреймворки - це, по суті, заздалегідь створені програмні компоненти багаторазового використання, які надають набір бібліотек, інструментів і шаблонів, що допомагають розробникам швидко та ефективно створювати додатки. Фреймворки стають дедалі популярнішими у розробці програмного забезпечення завдяки перевагам, які вони пропонують з точки зору структури, стандартизації та простоти використання. Вони пропонують структурований підхід до розробки програмного забезпечення, що дозволяє розробникам зосередитися на створенні унікальних функцій і можливостей, а не турбуватися про базову інфраструктуру та організацію коду.

Однією з головних переваг використання фреймворків є те, що вони дозволяють розробникам створювати складні додатки легше і швидше. Надаючи готові бібліотеки та інструменти, фреймворки позбавляють розробників необхідності писати код з нуля. Це не лише економить час та зусилля, але й зменшує ймовірність помилок та багів у коді. Крім того, оскільки фреймворки заздалегідь зібрані і протестовані, вони, як правило, більш надійні і міцні, ніж користувацький код.

Ще однією перевагою використання фреймворків є те, що вони сприяють узгодженості та стандартизації додатків. Фреймворки зазвичай включають в себе стандартизовані підходи до загальних завдань розробки, таких як обробка користувацького вводу, управління зберіганням даних, а також автентифікація та безпека. Використовуючи ці стандартні підходи, розробники можуть гарантувати, що їхній код буде узгодженим і простим в обслуговуванні, зменшуючи ймовірність помилок і полегшуючи оновлення та модифікацію коду з часом.

Незважаючи на ці переваги, використання фреймворків також має деякі потенційні недоліки. Наприклад, фреймворки можуть бути обмеженими з точки

зору гнучкості та кастомізації. Оскільки фреймворки надають готовий набір інструментів та бібліотек, розробникам може бути важко або неможливо налаштувати код під свої потреби. Крім того, фреймворки можуть мати тяжке навчання, особливо для розробників, які не знайомі з конкретним фреймворком.

Один із способів зменшити ці недоліки - вибрати фреймворк, який добре підходить для конкретних потреб конкретного додатку. Наприклад, якщо програма потребує складної, керованої даними функціональності, фреймворк на кшталт Django може бути гарним вибором, оскільки він забезпечує надійне моделювання даних та можливості інтеграції з базами даних. З іншого боку, якщо програма потребує високоінтерактивного, чуйного користувацького інтерфейсу, фреймворк на кшталт React може бути кращим вибором, оскільки він надає потужні інструменти для побудови користувацьких інтерфейсів.

Отже, фреймворки є важливим інструментом у сучасній розробці програмного забезпечення. Вони пропонують структурований підхід до розробки, сприяючи узгодженості та стандартизації, а також полегшуючи створення складних додатків швидко та ефективно. Однак вони також можуть бути обмеженими з точки зору гнучкості та кастомізації, а також можуть мати тяжке навчання. Щоб отримати максимальну користь від використання фреймворку, розробники повинні ретельно зважити свої конкретні потреби і вибрати фреймворк, який добре підходить для виконання поставленого завдання.

Одним з найпоширенішихтипівфреймворків є фреймворки для веб-додатків. Ці фреймворки спеціальнорозроблені для полегшеннярозробки веб-додатків, доступ до якихзазвичайздійснюється через веб-браузер. Фреймворки веб-додатківнадаютьрозробникамінструменти для виконанняпоширенихзавдань веб-розробки, таких як маршрутизація, шаблонування та обробкаданих, введенихкористувачем. Популярними прикладами фреймворків веб-додатків є Django, RubyonRails і Laravel.

Інший тип фреймворків - це фреймворки для мобільнихдодатків, якіспеціальнорозроблені для створеннямобільнихдодатків для iOS, Android та іншихмобільних платформ. Фреймворки для

мобільнихдодатківнадаютьрозробникамінструменти для вирішеннятиповихзавданьмобільноїрозробки, таких як дизайн інтерфейсукористувача, сумісність з пристроями та доступ до специфічнихфункцій пристрою, таких як камера абоGPS. Популярними прикладами фреймворків для мобільнихдодатків є ReactNative, Xamarin та Ionic.

Фреймворки для десктопнихдодатків - цеще один тип фреймворків, яківикористовуються для створеннядодатків, щозапускаютьсянативно в настільнійопераційнійсистемі, такій як WindowsабомасOS. Фреймворки десктопнихдодатківнадаютьрозробникамінструменти для виконаннятиповихзавдань з розробкидесктопнихдодатків, таких як дизайн інтерфейсукористувача, керування файлами та доступ до системнихресурсів, таких як буфер обмінуабо принтер. Популярними прикладами фреймворків для десктопнихдодатків є Qt, Electron та WPF.

Інший тип фреймворків - це фреймворки для розробкиігор, якіспеціальнорозроблені для створеннявідеоігор. Фреймворки для розробкиігорнадаютьрозробникамінструменти для вирішенняпоширенихзавданьрозробкиігор, таких як рендеринг графіки, моделюванняфізики та обробкакористувацького вводу. Популярніприкладифреймворків для розробкиігорвключаютьUnity, UnrealEngine та Godot.

Нарешті, існуютьтакож фреймворки, розроблені для конкретнихмовпрограмування, таких як JavaабоPython. Ці фреймворки надаютьрозробникамінструменти та бібліотеки, специфічні для певноїмовипрограмування, щополегшуєнаписання коду цією мовою. Прикладами мовнихфреймворків є Spring для Java та Flask для Python.

Однією з головнихперевагвикористанняфреймворків веб-додатків є те, що вони надаютьрозробникамстандартнийнабірінструментів і конвенцій для створення веб-додатків. Цедозволяєрозробникамзосередитися на створенніунікальнихфункцій і можливостей, а не турбуватися про низькорівневізавдання, такі як маршрутизація, шаблони та

обробкакористувацького вводу. Фреймворки такожнадаютьстандартизованіпідходи до загальнихзавдань веб-розробки, таких як інтеграція з базами даних, автентифікація та безпека, а такожобробка веб-запитів і відповідей.

Щеоднієюперевагоювикористанняфреймворків для веб-додатків є те, що вони допомагаютьзабезпечитистабільність, безпеку та масштабованістьдодатку. Фреймворки розробляються з урахуваннямнайкращих практик, і вони часто включаютьфункціїбезпеки, такі як захист CSRF та шифрування з коробки. Крім того, фреймворки полегшуютьвирішенняпоширених проблем веб-розробки, таких як кешування, балансуваннянавантаження та управліннясесіями. Цедопомагаєгарантувати, щододатокможепрацювати з великою кількістюкористувачів без проблем з продуктивністюабовразливостейбезпеки.

Існуєбагаторізнихтипівфреймворків для веб-додатків, кожен з якихмаєсвоїсильні та слабкісторони. Деякі фреймворки, такі як Django і Ruby onRails, розроблені для того, щоб бути високоціненими, надаючирозробникампевнийнабірумовностей та інструментів для створення веб-додатків. Інші фреймворки, такі як Flask та Express, є більш легкими та гнучкими, дозволяючирозробникамналаштовуватиархітектурудодатків та обиративласніінструменти та бібліотеки.

Одним з основнихфакторів, якіслідвраховувати при виборі фреймворку для веб-додатків, є мовапрограмування, на якійвінпобудований. Деякіпопулярні фреймворки для веб-додатківпобудовані на певнихмовахпрограмування, наприклад, Ruby onRails для Ruby абоFlask для Python. Інші, такі як Node.js та Express, призначені для роботи з різнимимовамипрограмування.

Отже, фреймворки для веб-додатків - цепотужніінструменти, якіможутьдопомогтирозробникамстворювати веб-додаткишвидше та ефективніше. Вони надаютьготовібібліотеки та інструменти, якідопомагаютьспроститипроцесрозробки, гарантуючи при цьомустабільність, безпеку та масштабованістьотриманогододатку. Оскількиіснуєбагаторізнихтипівфреймворків для веб-додатків,

розробникамважливоретельнопроаналізувати потреби свогододатку і вибрати фреймворк, якийнайкращевідповідаєцим потребам.

Існуєбагаторізнихтипівфреймворків, кожен з якихмаєсвоїунікальнісильні та слабкісторони. Для розробниківважливоретельнооцінити потреби свогододатку перед вибором фреймворку, щобпереконатися, що вони обралинайкращий фреймворк.

Одним з популярнихфреймворків є Ruby onRails, якийпобудований на мовіпрограмування Ruby і відомийсвоїмпідходом до конфігурації, щопереважає над конвенціями. Django, побудований на Python, маєвисокумасштабованість і безпеку, що робить йогопопулярнимвибором для великомасштабних веб-додатків. Laravel, побудований на PHP, пропонуєелегантний синтаксис і простоту використання, тоді як Express.js, побудований на Node.js, є дужегнучким і легким та ідеальнопідходить для створення API.

Flask - це легкий фреймворк для веб-розробки, дужегнучкий і простий у використанні, що робить йогокращимвибором для малих і середніх веб-додатків. AngularJS та React, обидвапопулярні фреймворки для веб-розробки, надаютьрозробникампотужніінструменти для створеннядинамічних та високоінтерактивнихкористувацькихінтерфейсів.

Vue.js - ще один фреймворк JavaScript, якийвідрізняєтьсявисокоюгнучкістю та масштабованістю, що робить йогочудовимвибором для малих та середніх веб-додатків, якіпотребуютьвисокоїпродуктивності. Нарешті, ASP.NET, розроблений Microsoft, - це фреймворк веб-розробки, призначений для створення веб-додатків корпоративного рівня, якийнадаєрозробникам ряд потужнихінструментів для створенняскладних веб-додатків, включаючиінтеграцію з базами даних, автентифікаціюкористувачів та безпечнуобробкуплатежів.

Отже, існуєбагаторізнихтипівфреймворків для веб-розробки, кожен з якихмаєсвоїсильні та слабкісторони. Розробникиповинніретельнооцінити потреби свогододатку перед тим, як вибрати фреймворк, щобпереконатися, що вони обралинайкращий. Вибравшиправильний фреймворк,

розробникиможутьзначноприскоритипроцесрозробки, гарантуючи при цьомустабільність, безпеку та масштабованістьотриманогододатку.

Трипопулярнихфреймворків для веб-розробки:

- React.JS
- Angular
- Vue

Розпочнімо з фреймворкаJS,а саме React.

React - це JavaScript бібліотека для побудови користувацьких інтерфейсів. Вона дозволяє розробникам створювати складні інтерфейси, використовуючи декларативний синтаксис, де вони визначають, що вони хочуть відобразити, а не як вони хочуть це відобразити. React використовує віртуальну DOM (DocumentObjectModel) для оптимізації продуктивності та рендерингу лише необхідних змін в інтерфейсі, що призводить до швидшого оновлення та зручнішого користувацького досвіду. На рис 2.1 зображений логотип фреймворку.

Рисунок 2.1 – Логотип React

React був розроблений для використання з іншими бібліотеками JavaScript, такими як Redux або ReactRouter, щоб забезпечити комплексне рішення для створення складних веб-додатків. Він також підтримує рендеринг на стороні сервера, що дозволяє розробникам рендеритиReact-компоненти на сервері та надсилати HTML клієнту, що може покращити час початкового завантаження сторінки.

Історія React починається з розробки стрічки новин Facebook. Стрічка новин стала поворотним моментом для компанії, оскільки дозволила користувачам бути в курсі подій своїх друзів та членів сім'ї в режимі реального часу. Однак ця функція мала проблеми з продуктивністю, оскільки вимагала постійного оновлення користувацького інтерфейсу для відображення нового контенту. Команда Facebook потребувала рішення, яке могло б покращити продуктивність стрічки новин, зберігаючи при цьому високий рівень інтерактивності.

У 2011 році ДжорданВок (JordanWalke), інженер-програміст Facebook, почав працювати над проектом, який мав на меті вирішити цю проблему. Проект Волка називався "FaxJS" і являв собою експериментальну бібліотеку JavaScript, призначену для покращення продуктивності стрічки новин Facebook. FaxJS використовувала концепцію під назвою "віртуальний DOM" для управління користувацьким інтерфейсом, що значно покращило продуктивність стрічки новин. Однак, FaxJS ніколи не був випущений публічно, оскільки на той час вважався експериментальним проектом.

Через кілька місяців команда Уолке вирішила переписати всю функцію стрічки новин за допомогою FaxJS. Нова версія стрічки новин була швидшою, гнучкішою та масштабованішою, ніж попередня. Успіх нової стрічки новин спонукав Facebook випустити FaxJS публічно як бібліотеку з відкритим кодом.

Однак, FaxJS не була ідеальною назвою для публічної бібліотеки. Тому команда почала придумувати нову назву для бібліотеки. Їм потрібна була проста назва, яка б легко запам'ятовувалася і привертала увагу. Після тривалого мозкового штурму вони зупинилися на назві "React".

React був випущений для громадськості в травні 2013 року і швидко завоював популярність серед розробників. Популярність бібліотеки була зумовлена її простотою та ефективністю, а також використанням віртуальної DOM, що дозволило розробникам створювати складні користувацькі інтерфейси без шкоди для продуктивності.

З моменту свого випуску React зазнав кількох оновлень та вдосконалень. Творці бібліотеки також додали кілька нових функцій, щоб полегшити

розробникам створення сучасних веб-додатків. Деякі з помітних особливостей React включають можливість створювати багаторазові компоненти інтерфейсу користувача, рендеринг на стороні сервера та підтримку нативної розробки додатків.

Творці React, серед яких ДжорданУок і Том Оккіно, продовжують працювати над бібліотекою і зробили її одним з найпопулярніших інструментів фронтенд-розробки у світі. Зараз React використовується такими компаніями, як Airbnb, Dropbox, Instagram та Netflix для створення швидких, адаптивних та масштабованих користувацьких інтерфейсів.

Використання React для створення веб-додатків вимагає знання JavaScript, HTML та CSS. Ось загальні кроки, яких слід дотримуватися при використанні React:

Налаштуйте середовище розробки: Вам потрібно встановити Node.js, середовище виконання JavaScript, та редактор коду, наприклад, VisualStudioCode. Вам також потрібно встановити бібліотеку React за допомогою NodePackageManager (NPM).

Створіть новий React-проект: Ви можете створити новий React-проект за допомогою інструменту CreateReactApp, який створить для вас базову структуру проекту.

Визначте компоненти: У React ви створюєте користувацькі інтерфейси, визначаючи компоненти - багаторазові будівельні блоки, які містять HTML, CSS та JavaScript, необхідні для візуалізації певної частини інтерфейсу.

Пишіть JSX-код: JSX - це синтаксичне розширення для JavaScript, яке дозволяє писати HTML-подібний код у ваших JavaScript-файлах. Ви можете використовувати JSX для визначення компонентів і пов'язаних з ними HTML і CSS.

Керування станами та даними: React використовує систему станів для управління даними, які відображаються в інтерфейсі користувача. Ви можете визначати змінні стану для своїх компонентів і оновлювати їх за потреби.

Використовуйте пропси: Пропси - це спосіб передачі даних та функціональності між компонентами. Ви можете визначити пропси для своїх компонентів і використовувати їх для налаштування поведінки компонента.

Рендерити компоненти: Щоб відрендеритиReact-компонент, ви використовуєте бібліотеку ReactDOM, щоб створити новий екземпляр компонента і відрендерити його в DOM.

Додавання обробників подій: Ви можете додавати обробники подій до ваших компонентів, щоб обробляти взаємодії користувачів, такі як натискання кнопок або відправлення форм.

Тестуйте та розгортайте ваш додаток: Після того, як ви створили свій React-додаток, ви можете протестувати його локально, а потім розгорнути на веб-сервері або хостингу.

Переваги React:

- Компонентна архітектура: Компонентно-орієнтована архітектура React є однією з його найважливіших переваг. Вона дозволяє розробникам розбивати додаток на менші компоненти, які можна використовувати повторно. Такий підхід полегшує управління складністю коду і дозволяє розробникам працювати над різними частинами програми незалежно. Крім того, оскільки кожен компонент має свій власний стан, це полегшує міркування про стан програми.
- Віртуальний DOM: Віртуальний DOM - ще одна перевага React. Це полегшене представлення реального DOM, яке відстежує зміни, що відбуваються в інтерфейсі користувача. Коли відбувається зміна, React оновлює віртуальний DOM, порівнює його з попередньою версією і оновлює реальний DOM тільки з урахуванням змін. Такий підхід робить додаток швидшим та ефективнішим, оскільки в DOM вносяться лише необхідні зміни.
- Продуктивність: Продуктивність React - ще одна перевага. Віртуальний DOM та компонентна архітектура дозволяють створювати високопродуктивні додатки. React оновлює лише необхідні частини інтерфейсу, а це означає, що додаток може працювати з великими наборами даних без

уповільнення. Крім того, React оптимізований для рендерингу на стороні сервера, що може покращити час початкового завантаження додатку.

- Спільнота та екосистема: React має велику та активну спільноту, яка робить свій внесок у його розвиток та надає підтримку розробникам. Екосистема навколо React також велика, з багатьма сторонніми бібліотеками та інструментами, доступними для використання розробниками. Це полегшує розробникам створення складних додатків, не вигадуючи велосипед.

Мінуси React:

- Тяжке в навчанні: У порівнянні з іншими фреймворками, React має тяжке навчання. Він вимагає ґрунтовного розуміння JavaScript та концепції віртуального DOM. Розробникам також потрібно вивчити JSX, синтаксичне розширення, яке дозволяє використовувати HTML-подібний синтаксис у JavaScript-коді. Процес навчання може бути складним для початківців, і розробникам, можливо, доведеться витратити значну кількість часу на освоєння React.
- Шаблонний код: Шаблонний код React може стати викликом для розробників. React спирається на компонентну архітектуру, і розробникам потрібно створювати компоненти та керувати ними, що може зайняти багато часу. Розробникам також потрібно керувати станами та пропсами, що може зробити код більш складним.
- Відсутність офіційної документації: Офіційна документація React не така вичерпна, як у інших фреймворків. Знайти відповіді на конкретні питання може бути складно, і розробникам, можливо, доведеться покладатися на форуми спільноти або зовнішні ресурси.
- Інструментарій: React покладається на інструментарій, що може бути недоліком для розробників. Їм потрібно встановлювати середовище, налаштовувати інструменти та керувати залежностями, що може зайняти багато часу.

React - популярний фреймворк для створення користувацьких інтерфейсів для веб та мобільних додатків. Хоча React має кілька недоліків, таких як Тяжке в

навчанні та шаблонний код, його переваги роблять його кращим вибором, ніж інші фреймворки. Продуктивність, гнучкість, підтримка спільноти та масштабованість React роблять його ідеальним вибором для розробників, які хочуть створювати ефективні та масштабовані додатки. Хоча React може бути не найкращим вибором для кожного проекту, його варто розглянути через його численні переваги.

Познайомившись з React тепер глянемо й на AngularJS

Angular - цепотужний і популярнийфронтенд-фреймворк для створенняскладних веб-додатків. ВінбувстворенийкомпанієюGoogle у 2010 році, і з того часу став одним з найпоширенішихфреймворків для створеннявеликомасштабниходносторінковихдодатків. У цьому ми розглянеморізніможливостіAngular, йогопереваги та недоліки, а такожпорівняємойого з іншимипопулярними фреймворками для фронтенду. На рис 2.2 зображено логотип Angular.

Рисунок 2.2 – Логотип Angular

Першим творцем AngularJS був МіскоХевері, інженер-програміст з Google, який працював над проектом зі створення поштового клієнта на основі веб-технологій. Хевері виявив, що традиційні інструменти та технології веб-розробки, такі як jQuery та HTML, не підходять для створення великомасштабних додатків, і почав експериментувати з різними підходами до веб-розробки.

Зрештою Хевері розробив новий фреймворк, який дозволив розробникам створювати веб-додатки з використанням декларативного синтаксису та двостороннього зв'язування даних, який він назвав AngularJS. Фреймворк швидко набув популярності в Google та серед ширшої спільноти веб-розробників і став основою для низки масштабних веб-додатків.

У 2014 році команда Angular в Google вирішила переписати фреймворк з нуля, використовуючи сучасні інструменти та методи веб-розробки. Нова версія Angular була розроблена як більш модульна та гнучка, ніж оригінальна AngularJS, і була випущена в 2016 році під назвою Angular.

СьогоднніAngular є одним з найпоширенішихфронтенд-фреймворків для створенняскладних веб-додатків. Вінрозроблений як високомодульний, з кожним компонентом програми, організованим у власномумодулі. Цеполегшуєпідтримку та масштабуваннядодатків з часом, оскількиокремікомпонентиможнаоновлюватиабозамінювати, не впливаючи на рештупрограми.

Angularтакожвключає ряд вбудованихінструментів і функцій, якіполегшуютьстворенняскладних веб-додатків. Наприклад, цепотужна система ін'єкціїзалежностей, яка дозволяє легко керуватизалежностямиміжрізнимичастинамипрограми. Вінтакожвключає в себе повнийнабірінструментівтестування, в тому числівбудований фреймворк для тестування та інструменти для наскрізноготестування.

Однією з ключовихособливостейAngular є використанняTypeScript, статично типізованоїпідмножини JavaScript, яка додаєтакіможливості, як необов'язкова статична типізація та об'єктно-орієнтованепрограмування на основікласів. TypeScriptпокликанийдопомогтирозробникамписатибільшзручний для підтримки і масштабований код, що робить йогоідеальним для великомасштабнихдодатків.

ЩобвикористовуватиAngular, розробникизазвичайпочинаютьзістворення нового проекту за допомогоюAngular CLI - інтерфейсу командного рядка для створення та керуванняAngular-додатками. CLI надає ряд кориснихінструментів і команд для створенняновихкомпонентів, сервісів і модулів, а також для побудови і розгортаннядодатків.

Післястворення нового проекту розробникиможутьпочатистворюватикомпоненти, використовуючидекларативний синтаксис шаблонівAngular та двостороннєзв'язуванняданих. Компоненти, як

правило, організовані в модулі, якіможна легко імпортувати та повторно використовувати в іншихчастинахпрограми.

Angular - цекомпонентний фреймворк, якийвикористовуєдекларативнішаблони та прив'язкуданих для створеннянасичених і динамічних веб-додатків. Вінпобудований на основіTypeScript, статично типізованоїпідмножиниJavaScript, яка додаєтакіможливості, як необов'язкова статична типізація та об'єктно-орієнтованепрограмування на основікласів. TypeScriptпокликанийдопомогтирозробникамписатибільшзручний для підтримки та масштабований код, що робить йогоідеальним для великомасштабнихдодатків.

Однією з головнихпереваг Angular є використанняTypeScript, статично типізованоїпідмножини JavaScript. TypeScriptдодає до JavaScript такіможливості, як необов'язкова статична типізація та об'єктно-орієнтованепрограмування на основікласів, щополегшуєнаписання коду, якийлегшепідтримувати та масштабувати. TypeScriptтакождопомагаєвідловлюватипомилкипід час компіляції, зменшуючиймовірністьпомилокпід час виконання та покращуючизагальнуякість коду.

ЩеоднієюперевагоюAngular є йогомодульнаархітектура. Кожен компонент Angular-додаткуорганізовано у власний модуль, щодозволяє легко підтримувати та масштабуватидодатки з часом. Такиймодульнийпідхідтакождозволяєоновлюватиабозамінюватиокремікомпонен , не впливаючи на рештупрограми, щоможе бути особливо корисним при роботі над масштабними проектами.

Angularтакожвключає ряд вбудованихінструментів і функцій, якіполегшуютьствореннясладних веб-додатків. Наприклад, цепотужна система ін'єкціїзалежностей, яка дозволяє легко керуватизалежностямиміжрізнимичастинамипрограми. Вінтакожвключає в себе повнийнабірінструментів для тестування, в тому числівбудований фреймворк для тестування та інструменти для наскрізноготестування.

Щостосуєтьсяпродуктивності, Angularвідомийсвоєюздатністюоброблятидані в режимі реального часу. Вінвключаєвбудованупідтримкудвостороннього зв'язуванняданих, щодозволяєвідображатионовлення в реальному часі в інтерфейсікористувача. Це робить йогогарнимвибором для додатків з вимогами до даних у реальному часі, таких як чат-додатки, онлайн-ігри та фінансовіторговіплатформи.

Однак у використанніAngular є й певнінедоліки. Однією з головних проблем є йогоТяжке в навчанні. ОскількиAngularвключає в себе багатопотужнихінструментів і функцій, розробникамможезнадобитисядеякий час, щобнавчитисякористуватися фреймворком. Цеможе бути особливо складно для новачків у веб-розробціабо для тих, хтозвикпрацювати з іншими фреймворками.

Ще одним потенційнимобмеженнямAngular є його великий розмір. ОскількиAngularміститьбагатовбудованихінструментів і функцій, фреймворк може бути досить великим і завантажуватисядовше, ніждеякіінші фреймворки. Цеможе бути проблемою для додатків, яківимагаютьшвидкогозавантаження, особливо на мобільнихпристрояхабо при повільномуінтернет-з'єднанні.

Нарешті, ще одним потенційнимобмеженнямAngular є йогосумісність з іншимибібліотеками та фреймворками. ХочаAngularможнавикористовувати з іншимибібліотеками та фреймворками JavaScript, інодіможе бути складно інтегруватийого з іншимиінструментами та технологіями. Цеможе бути проблемою для розробників, якіпрацюють над проектами, щовимагаютьінтеграціїзістороннімиінструментамиаботехнологіями.

Google, творецьAngular, широко використовує фреймворк у багатьохсвоїх продуктах і сервісах. Наприклад, Google CloudConsole, веб-інтерфейс для Google Cloud Platform, побудований з використаннямAngular. Google такожвикористовуєAngular для створеннявласнихвнутрішніхінструментів та додатків, включаючи систему відстеженняпомилок та інструмент для перегляду коду.

Окрім цих компаній, є багато інших організацій, які використовують Angular для своїх веб-додатків. Наприклад, Forbes, компанія, що спеціалізується на

фінансових новинах та медіа, використовує Angular для своєї веб-платформи. TheNewYorkTimes, одна з провідних світових газет, використовує Angular для свого веб-кросворду. Airbnb, онлайн-майданчик для орендижитла на час відпустки, використовуєAngular для своєї веб-системипошуку та бронювання.

Загалом, Angular - цепотужний і гнучкийфронтенд-фреймворк, який добре підходить для створенняскладних веб-додатків. ВикористанняTypeScript та RxJS робить йогогарнимвибором для великомасштабнихдодатків з вимогами до даних у реальному часі, а модульнаархітектура та вбудованіінструментидозволяють легко підтримувати та масштабуватийого з часом. Однак, перед тим, як обрати Angularзамістьіншихфреймворків, слідврахуватийогострімкукривунавчання та потенційніпроблеми з продуктивністю.

Після ознайомлення з минулими двома час познайомитись з останнім фреймворком – Vue. Це популярний прогресивний JavaScript-фреймворк, який використовується для створення користувацьких інтерфейсів та односторінкових додатків. СтворенийЕваном Ю у 2014 році, Vue.jsшвидкозавоювавпопулярністьсеред веб-розробниківзавдякисвоїйпростоті, гнучкості та легкості у використанні. У цьому ми розглянемоключовіособливостіVue.js, йогопереваги та недоліки, а такожзастосування в реальнихсценаріях. На рис 2.3 зображено логотип фреймворка.

Рисунок 2.3 – Логотип Vue

ОднакЕван Ю виявив, щоAngularJSзанадтоскладний і йоговажковикористовувати для невеликих проектів. Вінхотівстворити фреймворк, якийбув би простим у використанні, гнучким і легким. Він почав працювати над

Vue.js у вільний час і врешті-рештзалишив Google, щобзосередитися на розробці фреймворку на повну ставку.

Еван Ю надихавсякількомаіншими JavaScript-фреймворками, зокремаReact та AngularJS. Вінхотівоб'єднатинайкращірисицихфреймворків у новий фреймворк, якийбув би простим у використанні та вивченні. Вінтакожхотівнадатирозробникамгнучкий і масштабованийінструмент для створення веб-додатків.

Назва "Vue" походить відфранцузького слова "vue", щоозначає "вигляд". Цевідображає фокус фреймворку на створеннікористувацькихінтерфейсів та представлень для веб-додатків. НазватакожвідображаєбажанняЕвана Ю (Evan You) створити фреймворк, якийбув би простим для розуміння і використання.

Vue.js буввпершевипущений у лютому 2014 року і швидконабувпопулярностісеред веб-розробників. Простота та зручністьвикористання фреймворку припала до душірозробникам, які шукали легку та гнучку альтернативу AngularJS та React. Фреймворк такожотримаввисокуоцінку за відміннупродуктивність і швидкість, щозробилойогоідеальнимвибором для створенняодносторінковихдодатків.

З роками Vue.js продовжуваврости і розвиватися. Фреймворк зазнавкілька великих оновлень, останняверсія (Vue 3) вийшла у вересні 2020 року. У новійверсії фреймворку з'явилосякількановихфункцій та вдосконалень, зокрема нова система реагування та покращенапродуктивність.

Vue.jsбазується на архітектурномушаблоніModel-View-ViewModel (MVVM), якийдозволяєрозробникамрозділятидані, представлення та логічнийрівеньсвоїхдодатків. Цеполегшуєпідтримку та оновленнядодатків, оскількизміни, внні в один рівень, не впливають на інші. Vue.jstакожзабезпечуєреактивний і декларативнийпідхід до побудовикористувацькихінтерфейсів, щоозначає, щозміни, внні до базовихданих, автоматично відображаються в інтерфейсікористувача.

Переваги Vue.js:

- Простота та легкість у вивченні: Vue.js простий і легкий у вивченні для розробників, якімаютьбазовізнання HTML, CSS та JavaScript. Фреймворк маєчіткий і лаконічний синтаксис, що робить його легким для розуміння і використання. Це робить йогоідеальним як для досвідченихрозробників, так і для початківців.
- Гнучкий та масштабований: Vue.js - цегнучкий і масштабований фреймворк, якийможнавикористовувати для створеннямалих і великих веб-додатків. Він легкий і легко інтегрується з іншимибібліотеками та фреймворками. Це робить йогопопулярнимвиборомсередрозробників, якимпотрібенгнучкий і масштабованийінструмент для створення веб-додатків.
- Продуктивність і швидкість: Vue.js відомийсвоєювідмінноюпродуктивністю та швидкістю. Вінвикористовуєвіртуальну DOM (Document Object Model), яка дозволяєшвидшерендерити веб-сторінки. Це робить йогоідеальнимвибором для створенняодносторінковихдодатків, яківимагаютьшвидкої та плавноїроботи.
- Велика та активна спільнота: Vue.js має велику та активнуспільнотурозробників, якіроблятьсвійвнесок у розвиток та вдосконалення фреймворку. Спільнота створила широкий спектр плагінів, інструментів та бібліотек для Vue.js, щополегшуєрозробникамствореннясклadних та багатофункціональних веб-додатків.
- Реактивнеприв'язуванняданих: Vue.js використовує систему реактивного зв'язуванняданих, яка дозволяє автоматично оновлюватиподання при змінiданих. Цедозволяє легко створюватидинамічні та інтерактивнікористувацькіінтерфейси, якіреагують на введеннякористувача.

Недоліки Vue.js:

- Відсутністьофіційноїпідтримки: Vue.js - це фреймворк з відкритимвихідним кодом, а цеозначає, щовін не маєофіційноїпідтримкивід великих корпорацій, таких як AngularабоReact. Цеможе бути недоліком для деякихрозробників, яківважають за кращевикористовувати фреймворки з офіційноюпідтримкою.

- Обмеженіресурси та документація: Хоча Vue.js має велику та активнуспільноту, вінможемати не так багаторесурсів та документації, як інші фреймворки, такі як React та Angular. Цеможеускладнитирозробникампошукінформації, необхідної для створенняскладнихдодатків.
- Обмеженийінструментарій: Vue.js маєобмеженийінструментарій у порівнянні з іншими фреймворками, щоможеускладнитирозробникамналагодження та усуненнянесправностей у своїхдодатках. Цеможе бути недоліком для розробників, якіпотребуютьрозширеногоінструментарію та можливостейналагодження.

Чому Vue.js кращий за інші фреймворки?

Vue.js вважаєтьсякращим за інші фреймворки з кількох причин. По-перше, вінпростіший і легший у вивченні, ніжінші фреймворки, такі як React та Angular. Це робить йогоідеальним як для досвідченихрозробників, так і для початківців, якимпотрібен фреймворк, простий у використанні та розумінні.

По-друге, Vue.js більшгнучкий і масштабований, ніжінші фреймворки. Йогоможна легко інтегрувати з іншимибібліотеками та фреймворками, що робить йогоідеальним для створеннямалих і великих веб-додатків. Цягнучкістьтакожполегшуєрозробникамналаштування фреймворку підсвоїконкретні потреби.

Незважаючи на цінедоліки, Vue.jsнабувзначноїпопулярності і широко використовується в реальнихсценаріях. Vue.jsвикористовується такими компаніями, як Alibaba, Xiaomi та Baidu, середінших, для створеннясвоїх веб-додатків. Фреймворк такожвикористовується в проектах з відкритимвихідним кодом, таких як Laravel, Nuxt.js та Qasar.

Отже, Vue.js - цепопулярний і потужний фреймворк JavaScript, якийвикористовується для створеннякористувацькихінтерфейсів та односторінковихдодатків. Фреймворк забезпечуєпростий і гнучкийпідхід до створення веб-додатків, що робить його легким у вивченні та використанні для розробників. Vue.jstакожзабезпечуєвідміннупродуктивність і легко інтегрується з

іншимибібліотеками та фреймворками. Хоча фреймворк маєдеякінедоліки, він широко використовується в реальнихсценаріях і маєзростаючуспільнотурозробників, якіроблятьсвійвнесок у йогорозвиток і вдосконалення.

2.3Вибір бази даних

Бази даних - це колекції структурованих і організованих даних, до яких можна легко отримати доступ, керувати ними та оновлювати. Вони є важливими інструментами для бізнесу, організацій та приватних осіб для ефективного зберігання, управління та пошуку великих обсягів даних. Бази даних бувають різних типів, включаючи реляційні бази даних, об'єктно-орієнтовані бази даних, бази даних NoSQL тощо. У цьому ми розглянемо основи баз даних, їх важливість, типи і те, як вони використовуються в різних галузях.

База даних - це сукупність даних, які організовані та зберігаються таким чином, щоб забезпечити легкий доступ, пошук та управління ними. Бази даних складаються з таблиць, які зберігають пов'язані між собою дані, і ці таблиці пов'язані між собою зв'язками, утворюючи логічну структуру. Вони управляються системами управління базами даних (СУБД), які є програмним забезпеченням, що дозволяє користувачам зберігати дані, керувати ними та маніпулювати ними.

Бази даних відіграють важливу роль в ефективному управлінні та організації великих обсягів даних. Вони забезпечують централізоване місце для зберігання даних, що полегшує пошук інформації та відстеження змін. Бази даних також необхідні підприємствам, організаціям і приватним особам для прийняття обґрунтованих рішень на основі точних і своєчасних даних.

Типи баз даних:

- Реляційні бази даних: Ці бази даних організовують дані в таблиці зі стовпчиками і рядками. Дані в одній таблиці можуть бути пов'язані з даними в іншій таблиці за допомогою спільного поля або ключа.

- Об'єктно-орієнтовані бази даних: Ці бази даних зберігають дані у вигляді об'єктів, які є екземплярами класів, що містять атрибути та методи.
- Бази даних NoSQL: Ці бази даних використовують гнучку схему, яка дозволяє зберігати неструктуровані дані, такі як пости в соціальних мережах або дані з датчиків.
- Графові бази даних: Ці бази даних зберігають дані у вигляді графової структури з вузлами та ребрами, що робить їх ідеальними для складних взаємозв'язків між точками даних.

Бази даних використовуються в різних галузях, включаючи фінанси, охорону здоров'я, роздрібну торгівлю тощо. Ось кілька прикладів того, як використовуються бази даних:

- Охорона здоров'я: Бази даних використовуються для зберігання записів пацієнтів, історій хвороб та інших даних, пов'язаних зі здоров'ям. Ця інформація може бути використана лікарями та іншими медичними працівниками для прийняття обґрунтованих рішень щодо лікування пацієнтів.
- Фінанси: Бази даних використовуються для зберігання фінансових даних, таких як транзакції, інформація про клієнтів тощо. Ці дані можна використовувати для аналізу тенденцій, прогнозування доходів і прийняття інвестиційних рішень.
- Роздрібна торгівля: Бази даних використовуються для зберігання інформації про товари, дані про клієнтів і дані про продажі. Цю інформацію можна використовувати для оптимізації запасів, створення цільових маркетингових кампаній та підвищення рівня задоволеності клієнтів.

Бази даних є важливим інструментом для управління та організації великих обсягів даних. Вони забезпечують централізоване місце для зберігання даних, що полегшує пошук інформації та відстеження змін. Бази даних бувають різних типів, включаючи реляційні, об'єктно-орієнтовані, NoSQL та графові бази даних. Вони використовуються в різних галузях, включаючи охорону здоров'я, фінанси, роздрібну торгівлю тощо. Оскільки кількість даних продовжує зростати, бази даних залишатимуться невід'ємною частиною нашого життя та бізнесу.

У веб-програмуванні найчастіше всього використовують одну з трьох баз данних, а саме: PostgreSQL, MongoDB, Oracle.Зараз ми розберем ми їх розберем окремо щоб мати уявлення про кожне, коли і для чого воно використовується.

Розпочнімо знайомство з PostgreSQL.

PostgreSQL, також відома як Postgres, є популярною реляційною системою управління базами даних (СКБД) з відкритим вихідним кодом. Вперше вона була випущена в 1989 році і з тих пір здобула репутацію надійної, масштабованої та безпечної системи. У цьому ми розглянемо особливості, переваги та застосування PostgreSQL.PostgreSQL - це багатофункціональна СУБД, яка пропонує користувачам безліч переваг. Ось деякі з ключових особливостей PostgreSQL. На рис 3.1 зображено логотип бази данних.

Рисунок 3.1 – Логотип PostgreSQL

Доктор Стоунбрейкер та його команда розробили PostgreSQL в рамках своїх досліджень в Каліфорнійському університеті в Берклі. Вони працювали над проектом під назвою Postgres, який був об'єктно-реляційною системою управління базами даних. Метою проекту було створення системи управління базами даних, яка могла б працювати зі складними типами даних і зв'язками, зберігаючи при цьому переваги традиційної реляційної системи управління базами даних.

Проект Postgres був успішним, і в 1989 році команда випустила першу версію PostgreSQL як проект з відкритим вихідним кодом. З тих пір PostgreSQL продовжує розвиватися і вдосконалюватися завдяки внеску великої спільноти розробників.

Однією з ключових переваг PostgreSQL є її гнучкість. Вона підтримує широкий спектр типів даних і може обробляти як структуровані, так і неструктуровані дані. Це робить її добре придатною для широкого спектру

додатків, включаючи веб-додатки, фінансові послуги, охорону здоров'я та державні установи.На рисунку 3.2 зображено процес створеня нової бази данних.

Рисунок 3.2 – Процес створення нової бази данних (Консоль)

Щоб створити нову базу даних за допомогою psql, спочатку потрібно відкрити термінал або командний рядок і ввести наступну команду:

createdbmydatabase

Це створить нову базу даних з назвою "mydatabase". Після цього ви можете підключитися до бази даних за допомогою наступної команди:

psqlmydatabase

Це відкриє інтерфейс командного рядка psql, де ви можете вводити команди SQL для створення таблиць, вставки даних і виконання інших операцій над базою даних.

Якщо ви віддаєте перевагу використанню графічного інтерфейсу, ви можете завантажити і встановити pgAdmin, який є популярним інструментом з відкритим вихідним кодом для PostgreSQL. Після встановлення pgAdmin ви можете підключитися до бази даних, ввівши дані для підключення, такі як хост, порт, ім'я користувача та пароль. На рис 3.3зображено процес створення нової бази данних.

Рисунок 3.3 – Процес створення нової бази данних (Графічний інтерфейс)

Після підключення до бази даних ви можете використовувати pgAdmin для створення таблиць, вставки даних і виконання інших операцій з базою даних. pgAdmin має зручний інтерфейс, який дозволяє легко керувати базою даних.

Сумісність з ACID: PostgreSQL сумісна зі стандартом ACID (Atomicity, Consistency, Isolation, andDurability - атомарність, узгодженість, ізоляція та довговічність), що означає, що вона забезпечує цілісність та узгодженість даних навіть у випадку апаратних або програмних збоїв.

З відкритим вихідним кодом: PostgreSQL - це база даних з відкритим вихідним кодом, що означає, що її можна вільно використовувати та модифікувати. Вона також має велику спільноту розробників, які роблять свій внесок у її розвиток і підтримку.

Об'єктно-реляційна база даних: PostgreSQL - це об'єктно-реляційна база даних, що означає, що вона підтримує як реляційні, так і об'єктно-орієнтовані моделі даних. Це робить її дуже гнучкою та пристосованою до широкого спектру випадків використання.

Розширюваність: PostgreSQL є дуже розширюваною і може бути налаштована для задоволення конкретних потреб. Вона має модульну архітектуру, що дозволяє розробникам додавати нові функції та можливості за потреби.

Масштабованість: PostgreSQL може обробляти великі обсяги даних і є дуже масштабованою. Вона підтримує розбиття на розділи, що дозволяє розподіляти дані між декількома серверами для підвищення продуктивності.

Безпека: PostgreSQL має сильні функції безпеки, які включають шифрування, контроль доступу та аудит. Вона також підтримує шифрування SSL/TLS для безпечної передачі даних.

Переваги PostgreSQL:

- PostgreSQL має кілька переваг над іншими системами СКБД. Ось деякі з ключових переваг PostgreSQL:
- Надійність: PostgreSQL відома своєю надійністю та цілісністю даних. Вона має репутацію стабільної та узгодженої, навіть у середовищах з великими обсягами даних та великою кількістю транзакцій.
- Продуктивність: PostgreSQL розроблена для швидкої та ефективної роботи, навіть при обробці великих обсягів даних. Вона використовує передові методи індексування та оптимізації запитів, щоб забезпечити швидке та ефективне виконання запитів.
- Економічно ефективна: PostgreSQL безкоштовна у використанні і не вимагає жодних ліцензійних платежів. Це робить її економічно вигідним рішенням для організацій, яким потрібно управляти великими обсягами даних.
- Гнучкість: PostgreSQL підтримує широкий спектр типів даних і може обробляти як структуровані, так і неструктуровані дані. Її також можна легко налаштувати для задоволення конкретних потреб.
- Підтримка спільноти: PostgreSQL має велику і активну спільноту розробників і користувачів, які надають підтримку, поради та ресурси, щоб допомогти іншим отримати максимальну віддачу від системи.

Застосування PostgreSQL:

- PostgreSQL використовується в широкому спектрі додатків і галузей. Ось деякі з найпоширеніших застосувань PostgreSQL:
- Веб-додатки: PostgreSQL є популярним вибором для веб-додатків, особливо тих, які вимагають високого рівня надійності, масштабованості та безпеки. Її використовують такі компанії, як Apple, Fujitsu та Cisco для своїх веб-додатків.
- Фінансові послуги: PostgreSQL використовується багатьма фінансовими установами для управління фінансовими даними, такими як торгові дані, ринкові дані та дані про клієнтів. Вона відома своєю надійністю та безпекою, що робить її популярним вибором для цієї галузі.
- Охорона здоров'я: PostgreSQL також використовується в додатках для охорони здоров'я для управління даними пацієнтів, медичними записами та даними клінічних досліджень. Завдяки своїй надійності та масштабованості вона добре підходить для обробки великих обсягів конфіденційних даних.
- Уряд: PostgreSQL використовується державними установами та організаціями для управління широким спектром даних, включаючи дані переписів населення, виборчі дані та податкові дані. Її функції безпеки та надійність роблять її надійним вибором для урядових додатків.

На закінчення, PostgreSQL - це потужна і гнучка СУБД, яка була створена командою розробників на чолі з доктором Майклом Стоунбрейкером (MichaelStonebraker). Це проект з відкритим вихідним кодом, який розвивався і вдосконалювався протягом багатьох років завдяки внеску великої спільноти розробників. Щоб використовувати PostgreSQL, вам потрібно встановити її у вашій системі, а потім створити нову базу даних за допомогою інструменту командного рядка або графічного інтерфейсу. Після підключення до бази даних ви можете використовувати команди SQL або графічний інтерфейс для управління даними.

А тепер познайомимось з його аналогом - MongoDB

MongoDB - цеNoSQL-документно-орієнтована система баз даних, яка з роками набуває все більшоїпопулярностізавдякисвоїйгнучкості, масштабованості

та простотівикористання. Вона булавпершевипущена в 2009 роцікомпанієюMongoDBInc. і з тих пір стала провідною платформою для створеннясучаснихдодатків, яківимагаютьзберігання і пошукувеличезнихобсягівнеструктурованихабонапівструктурованихданих. У цьому ми надамодетальнийоглядMongoDB, включаючиїїархітектуру, можливості та варіантивикористання. На рис 3.4 зображено логотип.

Рисунок 3.4 – Логотип MongoDB

MongoDBбулавинайденаЕліотомГоровіцем, Дуайтом Мерріманом та КрістіноюХодороу. ЕліотГоровіц та Дуайт МерріманбуликолишнімиспівробітникамирекламноїкомпаніїDoubleClick, яку згодомпридбав Google. У 2007 році вони почали працювати над новим проектом підназвою 10gen, метою якогобуларозробка нового типу базиданих, яка була б швидшою та масштабованішою, ніжтрадиційнібазиданих. Крістіна Ходоров приєдналася до команди як основнийрозробник у 2009 році.

Команда почала з розробкисистеми баз даних з відкритимвихідним кодом, яка використовуваладокументну модель данихзамістьтрадиційноїтабличноїмоделіданих. Це дозволило розробникамзберігати та отримуватидані у більшгнучкий та масштабованийспосіб, щополегшило роботу з неструктурованими та напівструктурованимиданими. Вони назвали систему баз даних "MongoDB", щоозначає "величезнідані".

MongoDBбулавипущена в 2009 році і швидкозавоювалапопулярністьсередрозробників, які шукали систему баз даних, здатнуоброблятивеликідані і масштабуватися по горизонталі. MongoDB Inc. була заснована в 2010 році для наданнякомерційноїпідтримки та послуг для MongoDB.

АрхітектураMongoDBбазується на документніймоделіданих, яка дозволяєрозробникамзберігати та отримуватидані у вигляді JSON-подібнихдокументів. Документ - ценабір пар ключ-значення, де ключами є рядки, а значеннямиможуть бути будь-які типи данихBSON, включаючи рядки, цілі числа, числа з плаваючою комою, масиви та вбудованідокументи. На відмінувідтрадиційнихреляційних баз даних, MongoDB не вимагаєсхемиабо структур даних, що робить їїдужегнучкою та адаптивною до швидкозмінних середовищданих.

MongoDBзберігаєдані в колекціях, які є аналогом таблиць у традиційнихреляційних базах даних. Однакколекції в MongoDB не маютьпевноїсхеми, і кожен документ можематирізну структуру. Колекціїорганізовані в базиданих, які є логічнимигрупамиколекцій. MongoDBвикористовуєрозподіленуархітектуру, де данізберігаються в шардах - горизонтальних розділах даних.

Шардингдозволяєздійснюватигоризонтальнемасштабування і допомагаєрівномірнорозподілятиданіміждекількома серверами.

Переваги MongoDB:

- Гнучка модель даних. Модель даних документів MongoDB є дуже гнучкою, що дозволяє розробникам зберігати та отримувати дані різними способами. Це полегшує роботу з неструктурованими та напівструктурованими даними, які стають все більш поширеними в сучасних додатках.
- Масштабованість. MongoDB має високу масштабованість, як вертикальну, так і горизонтальну. Вона може обробляти великі обсяги даних і може бути розподілена між декількома вузлами для обробки ще більших робочих навантажень.
- Продуктивність. MongoDB відома своєю швидкою швидкістю читання та запису, що робить її популярним вибором для додатків, які потребують високої продуктивності.

- Індексування. MongoDB підтримує різноманітні варіанти індексування, включаючи однополюсні, складені та геопросторові індекси. Це дозволяє розробникам оптимізувати запити для підвищення продуктивності.
- Простота у використанні. Синтаксис і команди MongoDB прості у вивченні та використанні, що робить її популярним вибором серед розробників, які тільки починають працювати з базами даних.
- Відкритий вихідний код. MongoDB - це технологія з відкритим вихідним кодом, що означає, що вона є вільно доступною і може бути налаштована відповідно до індивідуальних потреб.

Мінуси MongoDB

- Відсутність транзакцій.Відсутність підтримки багатодокументних транзакцій в MongoDB може бути недоліком для деяких додатків, які вимагають складних операцій.
- Використання пам'яті.MongoDB може споживати багато пам'яті, особливо при роботі з великими наборами даних. Це може бути проблемою для додатків, які працюють на обмежених ресурсах.
- Узгодженість даних.Гнучка модель даних MongoDB може ускладнювати підтримку узгодженості даних у документах. Це може бути недоліком для додатків, які вимагають суворої узгодженості даних.
- Складність запитів.Модель даних документів MongoDB може ускладнювати написання складних запитів. Це може бути недоліком для додатків, які потребують складних запитів для отримання даних.
- Обмежені можливості аналітики.Аналітичні можливості MongoDB обмежені в порівнянні з деякими іншими системами баз даних. Це може бути недоліком для додатків, які потребують розширеної аналітики та звітності.

MongoDB є популярним вибором для створення сучасних додатків, які потребують гнучкого та масштабованого сховища даних. Його модель даних документів, масштабованість і продуктивність є одними з найбільших переваг. Однак відсутність підтримки транзакцій, використання пам'яті та узгодженості даних можуть бути недоліками в деяких сценаріях.

MongoDB використовується різноманітними організаціями, від невеликих стартапів до великих підприємств, у різних галузях. У цьому ми розглянемо, хто використовує MongoDB і чому.

Стартапи та малий бізнес часто обирають MongoDB за простоту використання, гнучкість та масштабованість. Документна модель даних MongoDB добре підходить для роботи з неструктурованими та напівструктурованими даними, які часто зустрічаються в сучасних додатках. Це полегшує стартапам і малому бізнесу швидкий початок роботи без необхідності турбуватися про складне моделювання даних або жорсткі вимоги до схем.

Крім того, масштабованість MongoDB робить його популярним вибором для стартапів і малих підприємств, які швидко зростають і потребують системи баз даних, здатної обробляти зростаючі обсяги даних і трафіку.

MongoDB також використовується багатьма великими підприємствами завдяки своїй масштабованості, продуктивності та гнучкості. Багато великих підприємств мають складні вимоги до даних, а документна модель даних MongoDB полегшує роботу з цими вимогами без необхідності створювати та керувати складними схемами реляційних баз даних.

Крім того, здатність MongoDB до горизонтального масштабування робить її придатною для обробки великих робочих навантажень і великих обсягів даних. Це робить її популярним вибором для великих підприємств, яким потрібно обробляти великі обсяги даних у різних місцях.

MongoDB також використовується в галузі охорони здоров'я, де вона використовується для зберігання та управління великими обсягами даних про пацієнтів. Ці дані можуть включати електронні медичні картки, дані медичних зображень та інші типи даних про пацієнтів.

Масштабованість, продуктивність і гнучкість MongoDB роблять її добре придатною для обробки великих обсягів даних про пацієнтів, які генеруються медичними організаціями. Крім того, здатність MongoDB обробляти неструктуровані та напівструктуровані дані полегшує медичним організаціям зберігання та управління різноманітними типами даних.

MongoDB також використовується у сфері фінансових послуг, де її застосовують для обробки великих обсягів фінансових даних, включаючи дані про транзакції, дані про клієнтів та ринкові дані.

Масштабованість і продуктивність MongoDB роблять її добре пристосованою для обробки великих обсягів даних, які генеруються організаціями, що надають фінансові послуги. Крім того, гнучкість MongoDB полегшує організаціям, що надають фінансові послуги, роботу зі складними та швидкозмінними вимогами до даних у цій галузі.

MongoDB також використовується в індустрії електронної комерції, де вона використовується для зберігання та управління даними клієнтів, продуктами та транзакціями.

Отже, MongoDB - це популярна система управління базами даних, яка використовується широким колом організацій, від невеликих стартапів до великих підприємств, а також у різних галузях, включаючи охорону здоров'я, фінансові послуги та електронну комерцію. Масштабованість, продуктивність і гнучкість MongoDB роблять її добре пристосованою для роботи зі складними вимогами до даних сучасних додатків, а її документна модель даних і здатність обробляти неструктуровані та напівструктуровані дані полегшують управління та аналіз даних. Незважаючи на деякі потенційні недоліки, такі як відсутність ACID-транзакцій і необхідність ретельного проектування схем, багато переваг MongoDB роблять її популярним вибором для організацій, які шукають сучасну і гнучку систему управління базами даних.

3 РОЗРОБКА КЛІЄНТСЬКОЇ ЧАСТИНИ CMSСИСТЕМИ

3.1 Розробка інтерфейсу

У сучасному бізнес-середовищі, що швидко розвивається, організації визнають першочергову важливість ефективного управління обслуговуванням клієнтів. Добре розроблена система управління обслуговуванням клієнтів (CSM) відіграє життєво важливу роль у забезпеченні задоволеності та лояльності клієнтів. Одним з найважливіших аспектів системи CSM є її інтерфейс, який слугує основною точкою взаємодії між користувачами та системою. Метою цього підрозділу є поглиблене вивчення ключових принципів та найкращих практик створення ефективного та зручного інтерфейсу для системи управління взаємовідносинами з клієнтами (CSM). Головні пункти які потрібні бути дотримані щоб інтерфейс був близьким до ідеалу:

- Підхід до проектування, орієнтований на користувача, має фундаментальне значення для успіху інтерфейсу системи CSM. Розуміння потреб, уподобань та больових точок основних користувачів системи - агентів з обслуговування клієнтів - має вирішальне значення. Проведення комплексного дослідження користувачів, включаючи інтерв'ю, опитування та спостереження, допомагає визначити вимоги та проблеми користувачів. Юзабіліті-тестування за участю репрезентативних користувачів дає змогу дизайнерам збирати відгуки та вносити корективи в дизайн інтерфейсу, забезпечуючи ефективне задоволення потреб користувачів.
- Простота і зрозумілість мають першорядне значення у створенні інтуїтивно зрозумілого та ефективного інтерфейсу для системи ЦВС. Складні інтерфейси можуть перевантажити користувачів, що призводить до плутанини та зниження продуктивності. Чистий і мінімалістичний дизайн з чіткою візуальною ієрархією та добре організованою інформаційною архітектурою дозволяє користувачам легко орієнтуватися в системі. Інтерфейс повинен представляти інформацію в стислій і зрозумілій формі, уникаючи непотрібного візуального безладу. Добре продумані меню, піктограми та написи сприяють впорядкованій та інтуїтивно зрозумілій роботі користувача.
- Дотримання послідовності в інтерфейсі має важливе значення для запобігання когнітивному перевантаженню та плутанині серед користувачів.

Послідовні шаблони дизайну, включаючи навігаційні меню, піктограми, кнопки та термінологію, створюють відчуття впізнаваності та дозволяють користувачам застосовувати свої наявні знання до системи CSM. Дотримання встановлених галузевих конвенцій і включення впізнаваних елементів з широко використовуваних додатків скорочує криву навчання для нових користувачів. Узгодженість поширюється не лише на візуальні елементи, а й на схеми взаємодії та робочі процеси, що сприяє підвищенню ефективності роботи користувачів та зменшенню кількості помилок.

- Створення інклюзивного інтерфейсу гарантує, що всі користувачі, в тому числі з обмеженими можливостями, можуть отримати доступ до системи CSM та ефективно використовувати її. Включення функцій доступності, таких як комбінації клавіш, альтернативний текст для зображень та підтримка програм для читання з екрану, є життєво важливим. Дотримання стандартів і рекомендацій з веб-доступності, таких як Рекомендації з доступності веб-контенту (WCAG), підвищує зручність використання для користувачів з порушеннями зору, слуху або опорно-рухового апарату. Проектування з урахуванням доступності з самого початку демонструє прихильність до різноманітності та забезпечує рівний доступ до функціональності системи.

В епоху зростаючого розмаїття пристроїв адаптивний підхід до дизайну є незамінним для інтерфейсу системи CSM. Інтерфейс повинен бути розроблений таким чином, щоб легко адаптуватися до різних розмірів та роздільної здатності екранів, включаючи настільні комп'ютери, планшети та смартфони. Адаптивний дизайн не лише забезпечує однаковий користувацький досвід на різних пристроях, але й задовольняє потреби мобільності агентів з обслуговування клієнтів, яким може знадобитися доступ до системи в дорозі. Гнучкі макети, гнучкі сітки та масштабовані ресурси є важливими компонентами адаптивного інтерфейсу, який може задовольнити потреби різних пристроїв.

Механізми зворотного зв'язку відіграють вирішальну роль у створенні позитивного користувацького досвіду в інтерфейсі системи CSM. Надання своєчасного та змістовного зворотного зв'язку допомагає користувачам зрозуміти

результати їхніх дій та зміцнює довіру до системи. Підтверджувальні повідомлення, індикатори прогресу та оновлення в режимі реального часу запевняють користувачів, що їхні дії реєструються та обробляються. Крім того, ефективна обробка помилок має вирішальне значення для допомоги користувачам у вирішенні проблем. Чіткі повідомлення про помилки, які ідентифікують проблему та надають вказівки щодо її вирішення, допомагають користувачам ефективно виправляти помилки. Проактивні заходи, такі як перевірки валідації та запобігання помилкам у реальному часі, сприяють безперебійній роботі користувачів, зменшуючи кількість помилок.

3.2 Організація взаємодії з базою данних

У сучасному світі, що базується на даних, бази даних стали основою різних додатків і систем. Від управління великими обсягами інформації до забезпечення цілісності та доступності даних, бази даних відіграють важливу роль у зберіганні, організації та пошуку даних. Цей має на меті надати всебічний огляд того, що таке бази даних, їх призначення та значення в сучасних обчисленнях.

За своєю суттю, база даних - це організована колекція структурованих даних, які зберігаються і управляються в електронному вигляді. Вона слугує центральним сховищем для зберігання інформації, до якої можна легко отримати доступ, керувати нею та оновлювати. Основна мета бази даних - забезпечити ефективний і надійний спосіб зберігання та пошуку даних, гарантуючи їх цілісність, безпеку та масштабованість.

База даних складається з кількох ключових компонентів, які працюють разом для ефективного управління та маніпулювання даними:

Дані: Дані є основним компонентом будь-якої бази даних. Вони являють собою інформацію, яка зберігається, організовується та обробляється в системі. Дані можуть бути в різних формах, включаючи текст, числа, дати, зображення або навіть складні структури, такі як мультимедійні файли або документи.

Таблиці: Таблиці - це фундаментальні будівельні блоки бази даних. Вони складаються з рядків і стовпців, утворюючи структурований формат для зберігання пов'язаних даних. Кожна таблиця представляє певну сутність або концепцію, а стовпці визначають атрибути або характеристики цієї сутності. Рядки, також відомі як записи або кортежі, містять фактичні екземпляри даних.

Зв'язки: Зв'язки визначають асоціації між різними таблицями в базі даних. Вони встановлюють зв'язки або залежності між сутностями, забезпечуючи цілісність і узгодженість даних. Найпоширеніші типи зв'язків включають один-до-одного, один-до-багатьох і багато-до-багатьох.

Запити: Запити дозволяють користувачам отримувати певну інформацію з бази даних. Вони уможливлюють пошук даних на основі умов, критеріїв фільтрації або конкретних вимог. Запити можуть варіюватися від простого пошуку до складних операцій, що включають кілька таблиць і перетворення даних.

Мова маніпулювання даними (DML): DML надає необхідні інструменти для взаємодії з базою даних. Вона дозволяє користувачам вставляти, оновлювати, видаляти та отримувати дані. Найпоширеніші операції DML включають оператори SELECT, INSERT, UPDATE і DELETE.

Мова визначення даних (DDL): DDL відповідає за визначення та керування структурою бази даних. Вона включає такі команди, як CREATE TABLE, ALTER TABLE і DROP TABLE, які використовуються для створення, модифікації або видалення об'єктів бази даних.

Бази даних можна класифікувати на різні типи на основі моделей і структур даних, що лежать в їх основі. Деякі з найпоширеніших типів включають

Реляційні бази даних: Реляційні бази даних організовують дані в таблиці із заздалегідь визначеними зв'язками. Вони використовують мову структурованих запитів (SQL) для управління та маніпулювання даними. Популярні системи управління реляційними базами даних (СКБД) включають MySQL, Oracle і Microsoft SQL Server.

Бази даних NoSQL: Бази даних NoSQL відрізняються від традиційної реляційної моделі і надають гнучкі схеми для зберігання неструктурованих або

напівструктурованих даних. Вони відмінно справляються з великими обсягами даних і підтримують розподілені та масштабовані архітектури. Прикладами баз даних NoSQL є MongoDB, Cassandra та Redis.

Об'єктно-орієнтовані бази даних: Об'єктно-орієнтовані бази даних зберігають дані в об'єктно-орієнтованих форматах, що дозволяє безпосередньо зберігати та отримувати складні структури даних. Вони підходять для додатків, які значною мірою покладаються на об'єктно-орієнтовані парадигми програмування.Бази даних стали життєво важливими компонентами в різних галузях і сферах. Їх важливість зумовлена наступними факторами:

Організація та управління даними: Бази даних забезпечують структурований та ефективний підхід до зберігання та організації даних. Вони забезпечують цілісність, узгодженість і стандартизацію даних, гарантуючи точність і достовірність інформації.Доступність даних: Бази даних дозволяють декільком користувачам і програмам одночасно отримувати доступ до даних і маніпулювати ними. Вони забезпечують контрольований доступ і заходи безпеки для захисту конфіденційної інформації.Аналіз даних та прийняття рішень: Бази даних полегшують аналіз даних, дозволяючи організаціям отримувати цінну інформацію, виявляти закономірності та приймати обґрунтовані рішення. Такі технології, як інтелектуальний аналіз даних, зберігання даних та бізнес-аналітика, покладаються на бази даних.

Масштабованість і продуктивність: Бази даних призначені для обробки величезних обсягів даних і підтримки зростаючих робочих навантажень. Вони використовують методи індексування, кешування та оптимізації, щоб забезпечити швидкий та ефективний пошук і обробку даних.

У сфері сучасної розробки програмного забезпечення зв'язок між базами даних та інтерфейсними додатками є вирішальним кроком у створенні надійних та динамічних систем. Цей має на меті дослідити найкращі практики та підходи для встановлення зв'язку між базами даних та інтерфейсними додатками, що уможливлює безперешкодний пошук даних, маніпулювання ними та їх представлення кінцевим користувачам.

Перш ніж зануритися в процес підключення, важливо зрозуміти природу взаємодії інтерфейсу і бази даних. Інтерфейс, що складається з HTML, CSS та JavaScript, відповідає за представлення користувацького інтерфейсу та обробку взаємодії з користувачем. З іншого боку, база даних зберігає та управляє даними програми. Подолання розриву між цими двома компонентами є життєво важливим для створення функціональних та інтерактивних додатків.

Найпоширенішим підходом для підключення баз даних до інтерфейсних додатків є використання внутрішнього сервера як посередника. Бекенд діє як інтерфейс між інтерфейсом і базою даних, забезпечуючи пошук даних, маніпуляції з ними та іншу бізнес-логіку. Такий підхід має низку переваг, зокрема підвищену безпеку, контрольований доступ до даних і розподіл проблем між інтерфейсом і базою даних.

Найпоширенішим методом зв'язку інтерфейсу з бекендом і базою даних є використання API (інтерфейсів прикладного програмування). API визначають набір правил і протоколів для зв'язку між різними програмними компонентами. Розробляючи та впроваджуючи RESTful API, бекенд може відкривати кінцеві точки, які дозволяють фронтенду запитувати та отримувати дані з бази даних у структурований та контрольований спосіб.

Для підключення внутрішнього сервера до бази даних можна використовувати різні методи і технології, залежно від конкретної системи управління базами даних (СУБД), що використовується. Деякі загальні підходи включають:

Драйвери для конкретної бази даних: Більшість СУБД надають власні драйвери, які полегшують зв'язок між внутрішнім сервером і базою даних. Ці драйвери, такі як JDBC для додатків на основі Java або ODBC для з'єднань загального призначення, дозволяють розробникам взаємодіяти з базою даних за допомогою стандартизованих API.

Інструменти об'єктно-реляційного відображення (ORM): Інструменти ORM абстрагують взаємодію з базою даних, дозволяючи розробникам працювати з об'єктно-орієнтованими представленнями даних вищого рівня. Популярні ORM-

фреймворки, такі як Hibernate для Java або Django ORM для Python, спрощують процес відображення таблиць бази даних в об'єктні моделі та забезпечують безшовну інтеграцію з бекендом.Бібліотеки підключення до баз даних: Бібліотеки підключення до баз даних, такі як ADO.NET для Microsoft SQL Server або psycopg2 для PostgreSQL, забезпечують пряме підключення до бази даних з внутрішнього сервера. Ці бібліотеки надають API та методи для встановлення з'єднань, виконання запитів та отримання даних.

При обміні даними між бекендом і фронтендом дуже важливо вибрати відповідні формати обміну даними. Найпоширеніші формати включаютьJSON (JavaScriptObjectNotation): JSON - це легкий, зрозумілий людині формат, який широко підтримується сучасними веб-технологіями. Він добре підходить для передачі структурованих даних між бекендом і фронтендом, дозволяючи легко аналізувати та маніпулювати ними в JavaScript.

XML (eXtensibleMarkupLanguage): XML - це гнучка мова розмітки, яка використовується для представлення та обміну даними. Хоча вона менш поширена в сучасній веб-розробці, вона залишається життєздатним варіантом, особливо в сценаріях, що вимагають більш складних структур даних або застарілих систем.

Для з'єднання інтерфейсу з бекендом і базою даних розробники можуть використовувати фреймворки і бібліотеки, які спрощують пошук даних і маніпуляції з ними. Деякі популярні варіанти включають

Фреймворки JavaScript: Сучасні фреймворки JavaScript, такі як React, Angular або Vue.js, надають надійні інструменти та бібліотеки для створення інтерактивних користувацьких інтерфейсів. Ці фреймворки можуть використовувати HTTP-бібліотеки, такі як Axios або Fetch, для надсилання запитів до бекенд API та отримання даних у форматі JSON для рендерингу у фронтенді.Рендеринг на стороні сервера (SSR): фреймворки SSR, такі як Next.js або Nuxt.js, дозволяють рендерити динамічний контент на сервері перед відправкою його клієнту. Такий підхід дозволяє серверу отримувати дані з бекенду і попередньо заповнювати ними фронтенд під час першого завантаження сторінки.

Движки шаблонів: Движки шаблонів, такі як Handlebars або Jinja2, дозволяють створювати динамічні HTML-шаблони з даними, отриманими з бекенда. Ці механізми надають механізми для вбудовування даних у розмітку HTML, що дозволяє розробникам створювати динамічні інтерфейси.

При з'єднанні інтерфейсу з бекендом і базою даних дуже важливо визначити пріоритети безпеки для захисту конфіденційних даних. Деякі основні заходи безпеки включають

Перевірка вхідних даних: Перевіряйте і дезінфікуйте дані, що вводяться користувачем, щоб запобігти потенційним SQL-ін'єкціям або іншим зловмисним атакам.Аутентифікація та авторизація: Впроваджуйте надійні механізми автентифікації, щоб гарантувати, що тільки авторизовані користувачі можуть отримувати доступ до даних і змінювати їх. Використовуйте такі методи, як управління сеансами, токени або OAuth для безпечної автентифікації користувачів.Шифрування даних: Шифруйте конфіденційні дані як у стані спокою, так і під час передачі, щоб захистити їх від несанкціонованого доступу.

Підключення баз даних до інтерфейсних додатків є критично важливим аспектом побудови функціональних та інтерактивних програмних систем. Використовуючи бекенд як посередника, встановлюючи зв'язок на основі API та використовуючи відповідні формати обміну даними, розробники можуть легко інтегрувати бази даних у свої інтерфейсні додатки. Дотримуючись найкращих практик, використовуючи безпечні методи підключення та враховуючи масштабованість і продуктивність, розробники можуть створювати ефективні, безпечні та зручні додатки, які використовують можливості баз даних.

Схожість

Джерела з Бібліотеки 136

1	Студентська робота	ID файлу: 1015240367	Навчальний заклад: Lviv Polytechnic National University	4 Джерело	5.76%
2	Студентська робота	ID файлу: 1015232303	Навчальний заклад: Lviv Polytechnic National University	17 Джерело	0.25%
3	Студентська робота	ID файлу: 1015079352	Навчальний заклад: Taras Shevchenko National Universit	4 Джерело	0.19%
4	Студентська робота	ID файлу: 1015079900	Навчальний заклад: National Technical University of Ukraine “Kyi...		0.18%
5	Студентська робота	ID файлу: 1015048391	Навчальний заклад: Donetsk National Technical Univers	30 Джерело	0.16%
6	Студентська робота	ID файлу: 1015196499	Навчальний заклад: National Technical University of Ukraine “Kyi...		0.15%
7	Студентська робота	ID файлу: 1015076422	Навчальний заклад: National Technical University of Ukra	2 Джерело	0.15%
8	Студентська робота	ID файлу: 1015039697	Навчальний заклад: Taras Shevchenko National University of Kyiv		0.13%
9	Студентська робота	ID файлу: 1014705673	Навчальний заклад: National Technical University of Ukraine “Kyi...		0.12%
10	Студентська робота	ID файлу: 1015119262	Навчальний заклад: National University of Life and Environmenta...		0.11%
11	Студентська робота	ID файлу: 1014713650	Навчальний заклад: Lviv Polytechnic National University	3 Джерело	0.11%
12	Студентська робота	ID файлу: 1015202480	Навчальний заклад: National Technical University of Ukra	3 Джерело	0.11%
13	Студентська робота	ID файлу: 1015236136	Навчальний заклад: Lviv Polytechnic National University		0.1%
14	Студентська робота	ID файлу: 1000788191	Навчальний заклад: National Technical University of Ukra	3 Джерело	0.1%
15	Студентська робота	ID файлу: 1015080991	Навчальний заклад: National University of Life and Environmenta...		0.1%
16	Студентська робота	ID файлу: 1015192306	Навчальний заклад: National Technical University of Ukra	4 Джерело	0.09%
17	Студентська робота	ID файлу: 1011479405	Навчальний заклад: National Technical University of Ukr	11 Джерело	0.09%
18	Студентська робота	ID файлу: 1007696943	Навчальний заклад: National University of Life and Envir	4 Джерело	0.09%
19	Студентська робота	ID файлу: 1015000454	Навчальний заклад: Interregional Academy of Personnel Managem...		0.09%
20	Студентська робота	ID файлу: 1003435233	Навчальний заклад: Lviv Polytechnic National University		0.09%

21	**Студентська робота**	ID **файлу**: **1005672077**	**Навчальний заклад**: Lviv Polytechnic National University		0.09%
22	**Студентська робота**	ID **файлу**: **1013072514**	**Навчальний заклад**: Uzhhorod National University	2 Джерело	0.08%
23	**Студентська робота**	ID **файлу**: **1005732766**	**Навчальний заклад**: National Aviation University	25 Джерело	0.08%
24	**Студентська робота**	ID **файлу**: **1011498027**	**Навчальний заклад**: National Technical University of Ukraine “Kyi...		0.08%
25	**Студентська робота**	ID **файлу**: **1014813918**	**Навчальний заклад**: National Technical University of Ukraine “Kyi...		0.08%
26	**Студентська робота**	ID **файлу**: **1014825732**	**Навчальний заклад**: Taras Shevchenko National Universit	8 Джерело	0.08%
27	**Студентська робота**	ID **файлу**: **1013013165**	**Навчальний заклад**: Lutsk National Technical University		0.08%
28	**Студентська робота**	ID **файлу**: **1015131821**	**Навчальний заклад**: National Aviation University		0.08%
29	**Студентська робота**	ID **файлу**: **1014993611**	**Навчальний заклад**: State University Kyiv National Economic Univ...		0.08%
30	**Студентська робота**	ID **файлу**: **1015221385**	**Навчальний заклад**: National Technical University of Ukraine “Kyi...		0.08%