

УДК 621.373.42

Ю.М. Рыбка

## ТЕХНОЛОГІЯ РОЗРОБКИ INTERNET-САЙТІВ З ВІДДАЛЕНИМ АДМІНІСТРУВАННЯМ

**Постановка проблеми.** Бурхливий розвиток Інтернет та швидке зростання обсягу різноманітної інформації буквально унеможливило ефективно та повнокровне функціонування установ та компаній навіть середнього масштабу без наявності в них власних web-сайтів. Зазвичай, це – сукупність складних динамічних сторінок з великим обсягом текстових та графічних даних, таблиць, тощо. Типовим явищем також є інтеграція таких web-сторінок із базами даних (БД). Зважаючи на те, що великий обсяг інформації на таких сайтах потребує значних та частих редагувань, постає актуальною задача їх віддаленого адміністрування.

Аналіз останніх досліджень, невирішені проблеми. На даний час мається певна кількість розроблених Internet-проектів за даним напрямом (автор і сам особисто приймав участь у шести розробках), але ще не вироблено єдиних загальних рекомендацій щодо структури таких сайтів та вибору програмного інструментарію.

**Постановка задачі.** Метою даної роботи є розробка концепцій з проектування складних інформаційно-пошукових платформо-незалежних web-сайтів із віддаленим адмініструванням.

**Основні результати.** Технічні завдання на реалізацію таких проектів зазвичай мають наступні пункти: розробка графічного оформлення сайту, проектування структури БД, розробка web-інтерфейсу, розробка динамічних сторінок за системою “клієнт-сервер” для додавання, отримання та адміністрування змісту БД на клієнтському браузері, розміщення статичної інформації про компанію, введення системи в експлуатацію та розробка інструкції користувача.

Визначившись із цими умовами, розробляють структурну схему, відображаючи порядок розміщення інформації та загальний принцип роботи проекту.

Побудову таких вузлів можна виразити у вигляді трьох рівнів:

1) Відображення інформації. Інтерфейс проекту має бути зрозумілим та легким у використанні, а меню та елементи навігації – інтуїтивними. Форми пошуку та додавання інформації повинні бути максимально простими та схожими на звичайні форми з офісних програм. Також вони мають мати інтерактивну перевірку даних клієнтом, що допоможе заощадити час користувачів при формуванні параметрів пошуку або додаванні інформації.

2) Програмна частина. Майже усі сторінки проекту мають генеруватися у залежності від даних користувача. Цей рівень реалізації є інтерфейсом між клієнтським браузером і БД та відповідає за вибірку даних з бази, видачу їх клієнтові, одержання інформації від клієнта та зберігання її в БД. Мова програмування повинна мати простий та зрозумілий синтаксис, усі основні властивості сучасної мови високого рівня, а також мати добру підтримку обраної БД. Запити до БД повинні бути її простими функціями і вільно транслюватися. Сама БД повинна мати досить простий інтерфейс для можливості швидкого ручного редагування в разі виникнення проблем.

3) Апаратний частина. Сервер для розміщення проекту такого рівня має бути швидким, щоб дозволяти працювати програмам з обробки великих масивів даних та без затримок надавати інформацію клієнтові. Він також повинен відповідати умовам надійності, доступності, можливості модернізації, мати надійний захист від несанкціонованого доступу та бути багатозадачним.

Програмний комплекс, що керує роботою таких сайтів можна поділити на три частини:

1. Модуль селектора типу режиму роботи;
2. Підсистема "Запит користувача";
3. Підсистема "Запит адміністратора".

Перший модуль, зазвичай, реалізується на стартовій сторінці сайту і служить для реєстрації та вибору типу роботи (режим користувача / режим адміністратора).

Режим "Запит користувача" дозволяє отримувати необхідні дані від системи через формування запитів на пошук та перехід до відповідних (завчасно сформованих) розділів-сторінок.

Режим "Запит адміністратора" є основною частиною сайту. Він дозволяє виконувати усі основні дії по веденню БД. Після його

вибору проходить авторизація на права доступу для роботи із системою.

Усі сторінки в обох режимах мають бути модульними, а система меню в них – уніфікованою й однаковою.

Основну дію з обробки даних і представлення їх користувачеві можуть виконати два модулі. Перший – інтерфейс обробника даних, дозволить йому працювати з даними (уведення, редагування, перегляд, вибір і т.д.). Другий – обробник даних, виконає всі операції над даними, тобто, додавання, відновлення та вилучення записів у БД. Усі результати обробки даних передаються в інтерфейс обробника даних. Саме він працює з БД. Запит до бази виконується окремим модулем, на вхід якого поступає запит у вигляді, зручному для користувача, а на виході він дає запит у вигляді, зручному для сервера БД.

У зв'язку з тим, що при проектуванні таких систем постають вимоги до захисту від введення некоректної інформації, після модуля інтерпретації запитів до БД повинен працювати модуль перевірки на помилку. Це виконується на машині клієнта за допомогою введення в систему "обробника даних". Модуль проводить перевірку на відповідність типів та на заповнення обов'язкових полів. Після успішного проходження цього етапу запит поступає в БД.

Після відповіді сервера БД дані поступають на вхід модулю інтерпретування відповідей. Він передає вже інтерпретовані дані в модуль редагування. У ньому дані відображаються у вигляді таблиці, де для кожного запису є можливість перейти в режим корекції.

Таким чином, реалізований модульний підхід до проектування системи і функції редагування та видалення інформації поєднані. Після редагування система знов переходить у режим претрансляції запиту, після чого знов запускається модуль визначення запиту.

Структуру сайту доцільно робити максимально простою. Основна сторінка повинна мати головне меню, де користувач може заходити під своїм ім'ям у систему, реєструватись як новий користувач, або здійснювати коректний вихід із системи.

Сторінка додавання нового користувача містить форму для заповнення, де маються такі поля: ім'я для входу, пароль, що є обов'язковими для заповнення і поля для введення повного імені, а також електронної адреси. Після відправлення на сервер даних

відбувається перевірка на наявність такого імені для входу. У разі успішного додавання в базу можна здійснювати вхід на сайт.

Сторінка для авторизованого входу містить 2 поля для введення. Це ім'я для входу й пароль.

Зміна особистої інформації надає можливість поміняти дані, введені при реєстрації. Для зменшення навантаження на сервер передбачається виконання скрипту при наборі цієї інформації. Так, наприклад, при спробі зміни пароля, відсутності підтвердження чи пароля, його невідповідності старому паролю, видається попередження, що інформація ще не надходить на сервер. Це заощаджує час і трафік.

Якщо вхід на сайт здійснено одним з адміністраторів, то з'являється форма, де можна продивитися перелік зареєстрованих користувачів системи і, при бажанні, видалити їх і їхні бази. Це відбувається на рівні роботи із сайтом “на віддалі” і не вимагає яких-небудь спеціальних знань програміста.

Аргументацію вибору програмних інструментальних засобів для розробки таких проектів треба проводити, зважаючи на те, що їх реалізація не повинна залежати від програмних платформ.

При виборі бази даних аналізувались такі фактори, як швидкість, поширеність у хостерів (планується клієнт-серверне рішення, а хостери, зазвичай, ставлять оптимальні комплекти ПО), надійність, ціна. За всіма показниками ідеально підходить база даних MySQL.

При виборі мови програмування, а саме вона потрібна для динамічного створення сторінок, вибір був між Perl, PHP, ASP. Perl не підходить із-за складності інтеграції з HTML і важкою розширюваністю. ASP є розробкою фірми Microsoft, що спричиняє покупку ОС Windows. Залишається PHP, який, після розгляду всіх можливостей, виявився найбільш зручнішим для розробки. Доповненням до серверної мови, а саме такими є всі перераховані мови, необхідно також використовувати клієнтську мову програмування, але після вибору основних засобів, тут практично немає альтернатив, окрім JavaScript.

Вибір web-сервера був практично однозначний – Apache. Як найбільш безпечний, швидкий, і встановлений на, практично, усіх хостингах.

**Висновки.** Таким чином, в роботі розглянуто концепції проектування інформаційних платформонезалежних web-сайтів із віддаленим адмініструванням та подано рекомендації з вибору програмного інструментарію.

Получено 21.11.2006