

Військовий інститут телекомунікацій та інформатизації
імені Героїв Крут

Кафедра
“Комп’ютерних інформаційних технологій”

Курсовий проект
з дисципліни: “Web-технології та Web-дизайн”
На тему: “Стройова частина”

Виконав:
курсант 292 навчальної групи
старший солдат Андрій САВЧЕНКО

Керівник:
викладач кафедри №22
прац. ЗСУ Андрій ЗАДВОРНИЙ

АНОТАЦІЯ

Курсого проекту на тему: “Стройова частина”

Курсова робота містить: 25 сторінок, 6 рисунків, 6 джерел.

В роботі проведено аналіз проектування графічної реалізації програми, та її внутрішньої логіки. Згідно рішень, які визначено в ході аналізу предметної області, спроектовано діаграму класів, діаграми компонентів та структурно-логічні схему програмного додатку; обґрунтовано вибір відповідних технологій та засобів розробки додатку. За результатом проектування реалізовано web-додаток.

ANNOTATION

Course project on the topic: “Combatant part”

The course work contains: 25 pages, 6 figures, 6 sources.

The paper analyzes the design of the graphic implementation of the program and its internal logic. According to the decisions that were determined during the analysis of the subject area, a class diagram, component diagrams and a structural and logical diagram of the software application were designed; the choice of appropriate technologies and application development tools is justified. Based on the design result, a web application was implemented.

ЗМІСТ

АНОТАЦІЯ	2
ANNOTATION	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
РОЗДІЛ 1	7
ДИЗАЙН WEB-ДОДАТКУ “СТРОЙОВА ЧАСТИНА”	7
1.1 Аналіз дизайну та огляд існуючих рішень	7
1.2 Аналіз існуючих підходів до створення web-сайту	13
РОЗДІЛ 2	15
ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ РОЗРОБКИ WEB-ДОДАТКУ	15
“СТРОЙОВ ЧАСТИНА”	15
2.1 Особливості розробки web-додатку	15
РОЗДІЛ 3	21
ОСОБЛИВОСТІ ВИКОРИСТАННЯ WEB-ДОДАТКУ	21
“СТРОЙОВА ЧАСТИНА”	21
3.1. Порядок використання web-додатку	21
3.2 Аналіз виконання вимог	22
ВИСНОВКИ	23
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	24
ДОДАТОК А	25

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

JVM – Java virtual machine

GUI – graphical user interface

CSS – Cascading Style Sheets

SWT – Standard Widget Toolkit

ВСТУП

Актуальність теми. На сьогоднішній день відсутня реалізація сайту стрийової частини, тому гостро стоїть питання в реалізації такого продукту. З допомогою цього додатку користувачі можуть переглядати новини та події, дізнатися як правильно робити той чи інший рапорт, переглянути накази МОУ.

Мета роботи: розробити web-додаток для автоматизування роботи документообігу у військах.

Виходячи з мети роботи, виникають наступні завдання:

- визначити дизайн web-додатку;
- визначити аналіз існуючих підходів до створення web-додатку;
- розробити інструкцію користувача.

Об’єкт дослідження: процес проектування web-додатку.

Предмет дослідження: методи реалізації, внутрішня логіка web-додатку.

РОЗДІЛ 1

ДИЗАЙН WEB-ДОДАТКУ “СТРОЙОВА ЧАСТИНА”

1.1 Аналіз дизайну та огляд існуючих рішень

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера. Зовнішній інтерфейс веб програми розробляється за допомогою таких мов програмування: HTML, CSS, Javascript, які підтримуються на будь-якому браузері (Opera, Chrome, Mozilla, Yandex). У той час як для написання серверної частини (Back-end) може використовуватися будь-яка інша мова програмування або фреймворк, Python, PHP, Ruby, Java.

Основні переваги веб-застосунків:

- веб додатки можуть застосовуватися на будь-якій операційній системі (Linux, Mac, Windows), оскільки всі вони підтримують сучасні браузери;
- у зв’язку з тим, що у веб-додатку використовується той самий код порівняно з desktop додатками їх набагато легше підтримувати;
- додаток простіше програмувати оскільки він не включає багато роботи з елементами ПК(ядро,процесор, відеокарта);
- на відміну від мобільних додатків, для веб-додатків не потрібно схвалення жодних платформ, щоб випустити свою програму;
- веб додатки це більш економний варіант для будь-якого підприємства

оскільки веб-програми не вимагають підписки або покупки ліцензій, а можуть використовуватися як SaaS-сервіс, що значно дешевше.

Web додаток і веб-сайт, здається два абсолютно ідентичних поняття, проте варто зауважити, що у них різні функції, а також компоненти розробки. Як ви вже зрозуміли веб-додаток це те, що запускається в браузері, проте він включає більшу кількість елементів з якими ми можемо взаємодіяти:

Більш детальну характеристику чим відрізняється веб-додаток і веб-сайт ви можете побачити в таблиці (дивись рис. 1.1):

Параметр	Веб-додаток	Веб-сайт
Основне призначення	Створюється для того, щоб взаємодіяти з користувачем	Має на увазі лише наявність статей, без будь-якого можливого впливу на читача
Взаємодія з користувачем	Користувач може проводити маніпуляції з даними, але з обмеженим доступом	Користувач може читати інформаційний контент, проте не може його ніяк міняти
Аутентифікація	Більшість веб додатків вимагають автентифікації, щоб користувач міг скористатися програмою	Для даних сайтів не потрібна обов'язкова автентифікація. Сайт може лише мати пароль, який відсилає адміністратор ресурсу
Завдання та складність	Веб-програма має багато функцій і закриває безліч проблем	Веб-сайт відображає лише статичну сторінку, на якій зображена текстова інформація
Зміна проекту	Щоб внести будь-які зміни в проект, потрібно робити ревью всього коду і після вписувати те, що ви хочете змінити.	Досить просто вносити зміни, лише зробивши пару змін у html коді сторінки.

Рис. 1.1 Різниця веб-додатку і веб-сайту

Давайте детальніше розберемо як відбувається робота самого веб-додатку. Користувач створює запит, який повинен буде відправитися на веб-сервер за допомогою запиту в інтернет-браузері, або в самому веб-додатку.

Далі цей запит має направитися на закріплений за цим веб-додатком веб-сервер. Після того як сервер отримав запит, він іде до бази даних для формування відповіді на запит користувача. Фінальним етапом йде відправка необхідних даних на Front-частину програми, а потім відображення користувачеві.

І так повторюється знову і знову, поки користувач створює запити у веб-застосунку. Хотілося б відзначити, що чим більше і складніше надходитимуть запити, тим довше буде надсилатися відповідь з сервера.

Види веб-додатків:

- SPA (Single Page Application) – односторінковий інтерактивний додаток. Важливо, що він не тільки знаходиться на одній сторінці, а й, подібно до повноцінної програми, є інтерактивним. Так інформаційний веб-сайт може складатися з однієї сторінки, але, по суті, не бути SPA. В односторінковому веб додатку користувач, перемикаючись між вкладками, залишається на одній сторінці. Причому підвантажуються та оновлюються лише необхідні частини

контенту, що грає на користь швидкості SPA. Приклад односторінкової програми – Gmail. Зверніть увагу – коли ви перемикаєтеся між списками повідомлень, адреса сторінки не змінюється. Це відмінна ознака SPA. Основна мова створення SPA – JavaScript. Невеликий односторінковий додаток можна зробити за допомогою бібліотеки jQuery. Однак цей варіант не найоптимальніший для великих проектів. Найкраще використовувати фреймворки Vue, React або Angular.

- MPA (Multi Page Application) – традиційні багатосторінкові веб-програми. Коли користувач взаємодіє з веб-сайтом, завантажуються нові HTTP-сторінки. Тому обмін даними відбувається повільніше, ніж у SPA. Особливо, якщо є проблеми з інтернет-з'єднанням або хостингом веб сайту. Приклади MPA – інтернет-магазини, такі як Rozetka та Amazon.

- PWA Прогресивна програма близька за своїми можливостями, функціями та якістю користувацького досвіду до нативних комп'ютерних та мобільних додатків. Чіткого кордону між не-PWA та PWA додатком немає. Але можна виділити низку характеристик. Зокрема PWA має містити проксі-шар (Service Worker) та Web App маніфест. По суті, браузер виступає віртуальною машиною для запуску веб-додатків, подібно до того, як Windows запускає exe-файли, а Android – apk.

- Service Worker – це проксі-шар між серверною та клієнтською частиною. Він знаходиться у браузері, через нього проходять усі запити. Таким чином, є два фронтенди шару – в одному прописується інтерфейс, в іншому – логіка. Це дозволяє виконувати повноцінні програми для Інтернету. Service Worker зазвичай пишеться на чистому JS.

Крім технічної класифікації існує також класифікація веб додатків на основі їх призначень. Нижче ми наведемо найпопулярніші види веб-додатків для бізнесу:

- E-commerce системи. Дані системи створюються для того, щоб клієнти могли замовляти і продавати товари без сторонніх осіб, тільки дві людини в ланцюжку продажу. Найбільш яскравими видами e-commerce платформ виступають: маркетплейси, онлайн-каталоги, інтернет-магазини.

- CRM-системи. Ці системи розробляються для автоматизації відділу продажу та всіх заявок, що надходять у вашу фірму. За рахунок CRM-системи

ви зможете: бачити та відслідковувати всю воронку продажів компанії, призначати зустрічі та бачити історію взаємодії з усіма клієнтами.

- ERP-системи. Це веб-системи, які включають не тільки автоматизацію відділу продажу, але й усіх ресурсів та підрозділів компанії. Завдяки ERP-системі можна бачити ефективність кожного підрозділу та ставити відповідні завдання.

- Корпоративні портали. Це веб-програми, які виступають у ролі соціального модуля для вашого холдингу. За рахунок цієї веб-програми можуть вирішуватись такі проблеми як: швидке інформування всіх співробітників компанії, корпоративне навчання, контроль співробітників (hr-модуль).

Розробка веб програми ділиться на 6 етапів. Кожен етап по-своєму важливий і не варто нехтувати кожним з них:

Постановка мети. На початку ви як власник бізнесу або стартапу, повинні визначитися з тим, які цілі ви переслідуєте при розробці програми, які бізнес-завдання воно має вирішити.

Складання технічного завдання. Під час цього етапу прописується технічна документація для проекту, стек технологій який буде використовуватись, а також список усіх технічних елементів, які будуть розроблені.

Прототипування. Створюється черновий макет веб-додатка, де буде показано взаємодію системи з користувачем.

Створення дизайну. Відображення дизайну відбувається відповідно до останніх тенденцій UX/UA. Це сприяє тому, що веб додатком буде зручніше користуватися, а також він буде більш впізнаваний.

Програмування. Команда програмістів створює всі веб-сторінки вашого сайту, а також логіку яка повинна виконуватись у веб-додатку (відображення цін, місцезнаходження товару, відображення знижок залежно від програми лояльності).

Тестування. Після того, як написано весь код, важливо зробити перевірку, щоб кожен елемент системи працював стабільно і правильно, адже найменша помилка може коштувати компанії великих коштів. Важливо робити обидва види тестування (Manual і Unit), щоб усунути всі можливі баги.

Веб-додаток отримує запит від клієнта і виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту через мережу з використанням протокола HTTP.

Веб-додаток может бути клієнтом інших служб, наприклад, бази даних або іншого веб-додатку, розташованого на іншому сервері.

Останнім часом набуває популярності новий підхід до розробки веб-додатків, який називається Ajax. Сторінки веб-додатка не перезавантажуються повністю, а лише завантажують з сервера зміни, що робить їх більш інтерактивними і продуктивними.

Незалежність від операційної системи клієнта. Додаток створюється один раз для довільно вибраної платформи і на ній розгортається. Проте різна реалізація HTML, CSS, DOM і інших специфікацій в браузерях може викликати проблеми при розробці веб-додатків і подальшої підтримки.

Простота використання. Відвідувач сайта має протягом 10-15 секунд зрозуміти, як почати користуватися сайтом. Не сподівайтесь, що хтось читатиме інструкції або витратить 1-2 години на вивчення системи меню та команд.

Успіх веба багато в чому зобов'язаний тому, що велика частина теоретичних побудов, присвячених гіпертексту, була відкинута на користь простих прагматичних рішень, які і послужили основою ідеальної конструкції. RSS став, можливо, єдиним широко поширеним веб-сервісом саме тому, що він простий. А складні корпоративні набори все ще чекають своєї години.

Підтримуйте спрощені моделі програмування і ви отримаєте вільно-зв'язаних партнерів. Проблема корпоративних веб-сервісів в тому, що вони передбачають жорстко обумовлене партнерство. У багатьох випадках це виправдано, але частенько найцікавіші додатки можуть бути побудовані на вельми крихкій основі.

Думайте про синдикацію, а не про координацію. Прості веб-сервіси - як RSS або сервіси на базі REST - займаються синдикацією даних, не намагаючись контролювати, що відбувається з інформацією на іншому кінці ланцюжка. Ідея кризної передачі даних є однією з базових ідей самого інтернету.

Великий час реакції на дії користувача. Від моменту натискання на посилання до появи результату часто проходить більше 1 секунди (а буває й

більше). Тому людина встигає помітити затримку і навіть занудьгувати. Задача програміста - максимально скоротити час реакції додатка. Як мінімум, початок відповіді має з'явитися менш ніж за 10 секунд.

Велика кількість користувачів. В успішних веб-проектах на сайт приходять десятки користувачів на секунду. Тому варто замислитися над оптимізацією швидкості виконання програми.

Стійкість до атак. Через відкритий доступ до керування web-додатком, кількість хакерських атак на такі програми дуже велика.

Гігантські масиви даних. Бази даних з мільярдами записів перестали бути екзотикою зовсім недавно, і тому ще немає інструментів для ефективного пошуку та маніпуляції такими великими кількостями інформації.

Мережні, розподілені, паралельні обчислення, системи агентів

Сучасні додатки internet часто спілкуються між собою. Наприклад, сайт "електронний магазин" використовує сайт банка, щоб прийняти оплату за товар. Розважальні сайти показують рекламні блоки з інших сайтів і т.д.

Програмне забезпечення – сервіс, а не товар. Однією з головних характеристик сучасних інтернет-додатків є те, що вони поширюються у вигляді сервісу, а не товару. Це, у свою чергу, веде до фундаментальних змін в бізнес-моделях компаній-розробників програм.

Компанія повинна уміти управляти процесами. Мистецтво розробки додатків повинне супроводжуватись умінням організувати щоденні операції для підтримки роботи цих додатків. Розрив між софтом-артефактом і софтом-сервісом такий великий, що вже зараз не можна написати хороший продукт і забути про нього - його потрібно підтримувати щодня.

Не випадково інформація про системне адміністрування, обслуговування мереж, балансування навантаження і тому подібне охороняється Google, мабуть, навіть краще, ніж самі пошукові алгоритми. Google навчився автоматизувати згадані процеси, а це - ключова частина його цінової переваги перед конкурентами.

Відстежування поведінки користувачів в реальному часі дозволяє бачити, які нові властивості використовуються і як вони використовуються - і це ще одна ключова складова успіху технології. Веб-розробник одного з розкручених мережесервісів відзначає: "ми додаємо дві-три нових

властивості в різні частини сайту щодня, і якщо користувачам вони не подобаються - ми відмовляємося від цих нововведень. Якщо подобаються - упроваджуємо на всьому сайті".

Проектуйте з врахуванням можливих переробок і поліпшень. Ослаблений захист інтелектуальної власності Системи, подібні вебу, RSS і AJAX, схожі тим, що особливих перешкод для їх повторного використання не існує. Велика частина корисного програмного забезпечення має відкриті тексти, а якщо і немає, то існує мало способів захистити свою інтелектуальну власність. Стандартна функція оглядача інтернет "преглянути вихідний код" дозволяє будь-якій людині скопіювати будь-яку веб-сторінку.

RSS був спроектований для того, щоб користувач міг читати контент тоді, коли це зручно йому, а не постачальникові інформації. Найуспішніші веб-сервіси - це, як правило, такі служби, які можуть бути змінені несподіваним для їх творців чином (some rights reserved).

1.2 Аналіз існуючих підходів до створення web-сайту

Сайти можна класифікувати за доступністю сервісів, фізичним розташуванням і призначенням.

За доступністю сервісів їх поділяють на:

- відкриті – усі сервіси повністю доступні для будь-яких відвідувачів і користувачів;
- напіввідкриті – потрібно зареєструватися (зазвичай безкоштовно);
- закриті – повністю закриті службові сайти організацій (наприклад, корпоративні сайти), особисті сайти приватних осіб. Такі сайти доступні для вузького кола людей.

Доступ нових людей можливий через запрошення. За фізичним розташуванням. Якщо сайт доступний користувачам з Інтернету, він вважається зовнішнім, натомість сайт, доступ до якого можуть здійснювати лише користувачі локальної мережі, є внутрішнім. Прикладами внутрішнього сайту можуть бути корпоративний сайт підприємства або сайт приватної особи в локальній мережі провайдера.

За призначенням сайти поділяють на:

- бізнес-сайти – сайти, що містять інформацію про компанії та їхні послуги, здійснюють функцію електронної торгівлі;
- інформаційні сайти – призначені для інформування відвідувачів, поширення новин, тематичні сайти, енциклопедії, словники тощо;
- сайти соціальних мереж – інтерактивні багатокористувацькі веб-сайти, які наповнюються самими учасниками мережі. Сайт являє собою автоматизоване соціальне середовище, що дозволяє спілкуватися групі користувачів, об'єднаних спільним інтересом;
- веб-портали – універсальні сайти, через які можна вийти на інші ресурси Інтернету;
- сайти сервісів – сайти служб, які існують у мережі Інтернет, зокрема, сайти пошукових служб (Google, Bing), поштові сайти, веб-форуми, он-лайніві сховища даних (Skydrive), сайти служб онлайнного документообігу (Google Docs), зберігання та обробки фотографій (Picnik, ImageShack, Panoramio, Photobucket), зберігання відео (You Tube).

Для створення, редагування та керування вмістом сайту використовують спеціальну систему керування контентом — CMS (від англ. Content Management System). Це програмна оболонка, яка дозволяє легко вводити і редагувати дані — текст, рсунки тощо, додавати і видаляти сторінки, тобто керувати сайтом у режимі онлайн, без знання HTML, мов програмування та інших спеціальних навичок.

Оскільки створити сайт за допомогою CMS можна швидко і без спеціальних навичок, цей інструмент стає дедалі популярнішим.

Хостинг (від англ. host) — це виділення місця на вінчестері серверу, що має цілодобове підключення до Інтернету, для розміщення сайту. Цю послугу надають як Інтернет-провайдери, так і компанії, що спеціалізуються на наданні послуг хостингу. Зазвичай на хостинг-серверах розміщуються від кількох до декількох тисяч сайтів.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА ОСОБЛИВОСТІ РОЗРОБКИ WEB-ДОДАТКУ

“СТРОЙОВ ЧАСТИНА”

2.1 Особливості розробки web-додатку

Web-додатки мають архітектурні, технологічні і призначені для користувача особливості:

- web-додатки повинні розгортатися дуже швидко, і вимоги до них, ймовірно, змінюються в ході етапу розробки;
- все більш частою практикою в web-розробці стає заміна документів на реальні прототипи додатки;
- дуже раннє підключення кінцевих користувачів до тестування і розвитку web-додатки;
- в той час як при розробці звичайних додатків створюється тільки одна версія додатка, яка задовольняє всім заданим вимогам, для web-додатків все більш частою (і бажаною) стає публікація в web-мережі додатки, яке ще не задовольняє всім заданим вимогам;
- отримання ранніх відгуків користувачів стає все більш важливим для розробки web-додатки, а цикл розвитку дійсно починає розглядатися в якості реальної можливості для поліпшення web-додатки, а не як додаткова витрата часу на доведення додатки до робочого стану;
- робота з проектування web-додатків ділиться на проектування даних і проектування навігації.
- розробка web-додатків включає специфічні процеси:
- web-додатки є спеціальним видом звичайних програмних додатків, і тому web-інженерія може розглядатися в якості спеціальної різновиди програмної інженерії;
- розробка додатків для web-мережі передбачає використання декількох добре певних правил і угод, які надають стабільно працюючу, стійку і масштабовану середу розробки і виконання;
- існують спеціальні процеси розробки, які повинні враховувати особливі можливості web-додатків.

Модель життєвого циклу сучасних web-додатків показана на рис. 2.1.



Рис. 2.1. Модель життєвого циклу сучасних web-додатків

Основна відмінність даної моделі від моделі розробки звичайного ПО полягає в розумінні "розгортання програми" не як виду діяльності (роботи), а як переходу між роботами.

У web-мережі розгортання додатки для його користувачів дійсно не є трудомісткою роботою: у зв'язку з централізованою архітектурою звичайних web-додатків, відсутністю спеціального коду програми на стороні користувачів, використанням браузера в якості середовища виконання робота з розгортання web-додатки в значній мірі полегшується і прискорюється.

Дана модель пропонує явне з'єднання роботи по "підтримці та розвитку" з "аналізом вимог". Саме цей перехід характеризує модель, він створює другий цикл моделі, який включає "аналіз вимог". Даний цикл називається "циклом розвитку". Перший цикл пов'язує "проектування", "реалізацію" "тестування і розвиток". Даний цикл називається "цикл побудови і перевірки". Такі два циклу відповідають двом етапам, які характерні для сучасних web-додатків: оффлайн-розробка та онлайн-розробка. Оффлайн-розробка полягає в створенні web-додатки в середовищі розробки (наприклад. Visual Studio) без використання реального web-сервера. Розробка в режимі онлайн виконується після перенесення web-додатки під управління реального web-сервера: створення віртуального каталогу, файлів настройки; управління web-сервером.

Веб-дизайнери проектують логічну структуру веб-сторінок, знаходять найбільш зручні рішення надання інформації, а також займаються художнім оформленням веб-проекта. У результаті поєднання двох галузей людської діяльності гарний веб-дизайнер повинен бути знайомий з останніми веб-технологіями і мати відповідні художні якості.

Це визначення має суттєві недоліки.

По-перше, нема чіткого визначення місця Web-дизайнера у розробці та подальшому супроводженні сайтів. Супроводження для сайтів це такий самий важливий етап як і створення. Сайту без інформаційного наповнення не існує, а його дає замовник.

По-друге, логічну структуру сайту, як і його інформаційне наповнення, теж задає замовник.

По-третє, подальше існування сайту, його розвиток, його супроводження теж залежать не тільки від Веб-дизайнера.

По-четверте, сайт може мати певні вразливості (іноді вони мають назву - "дірки") і їх наявність залежить від кваліфікації Веб-дизайнера, а точніше від сумісної роботи Веб-дизайнера та Веб-програміста. Фактично у розробці сайту приймають, чи повинні приймати, участь декілька спеціалістів різних фахів.

1. Замовник

Замовник володіє інформацією, що розміщується на сайті. Він вміє структурувати інформацію, виділяти найбільш важливу, зробити її лінгвістичну перевірку чи доручити зробити цю перевірку відповідним фахівцям, виділити інформацію, що буде поновлюватись, визначити засоби online та offline зв'язку. І цей перелік може у кожному конкретному випадку бути продовжений.

2. Web-дизайнер

Функції Web-дизайнера в основному співпадають з наданим у Вікіпедії визначенням. Але його треба розглядати як координатора робіт по створенню сайту. Він повинен опікуватися і про подальше існування розробленого сайту.

3. Web-програміст

Роль Web-програміста все збільшується в процесі розвитку Web-технологій. На початку існування Web-сайтів його функції зосереджувались на розробці скриптів, які розширювали можливості сайтів. При технологіях, коли генерація сторінок виконується на боці сервера, функції Web-програміста при розробці Web-сайта вважаються основними. До речі, він є і основним джерелом наявності вразливостей на сайті, і тим, хто їх може прибирати. Розробкою Веб додатків займаються програмісти.

Розглянемо поняття Веб додатку.

Принцип роботи Web-додатків. Web-додатки використовують архітектуру "клієнт- сервер". Власне, Web-додаток знаходиться на сервері і

обробляє запити, які передають йому через Інтернет. На стороні клієнта Web-додаток працює у браузері. Призначений для користувача інтерфейс Web-додатка передається у вигляді сторінок на мові HTML, де браузер інтерпретує і відображає їх. На стороні сервера Web-додаток працює під управлінням IIS (Internet Information Services). IIS управляє роботою додатка, передає йому клієнтські запити і повертає клієнтам результати виконання їх запитів. Запити і результати їх виконання передаються через Інтернет по протоколу HTTP (Hypertext Transport Protocol). Протокол реалізується через середовище Інтернет. Взаємодія клієнта і сервера при роботі Web-додатка. Фактично Web-додаток komponує відгук з серверних ресурсів, до яких належать виконуваний код, що працює на сервері: Web-форми, HTML-сторінки, графічні файли і інший вміст, що становить інформаційне наповнення додатків. Web-додатки багато в чому нагадують традиційні Web-сайти, але на відміну від них відображають користувачу динамічний вміст, що генерується виконуваним кодом додатка, а не статичні сторінки, що зберігаються на сервері в готовому вигляді. На рис. 3 показано, як Web-додаток komponує HTML-сторінки, що повертаються користувачам. Виконавча частина Web-додатка здатна робити багато що з того, чого не можуть статичні Web-сайти, а саме:

- приймати дані від користувача і зберігати їх на сервері;
- виконувати для користувача різні дії: розмістити замовлення, робити складні обчислення і витягувати інформацію з баз даних (БД);
- розпізнавати користувача і відображати інтерфейс, настроєний відповідно до його переваг;
- відображати постійно змінний вміст, наприклад інвентарні списки, оброблювані замовлення і відомості про відвантажувані товари

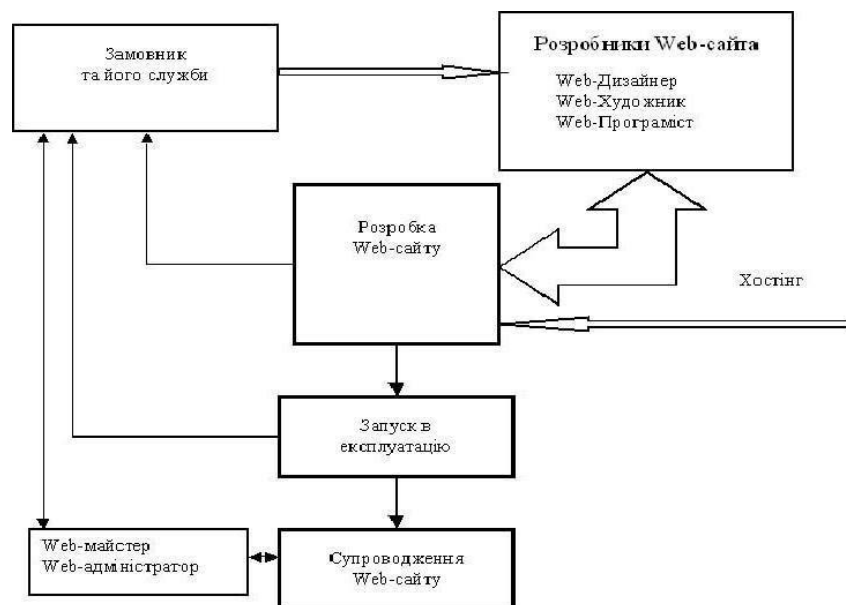
4. Web-художник

Ще одним спеціалістом, який може бути залучений до роботи над Web-сайтом, є Web-художник. Часто без спеціальної освіти та таланту художника знайти художні рішення не вдається.

5. Менеджер проекту

Звичайно керує роботою менеджер проекту

Для розробки Web-сайту на замовлення ідеальним було б об'єднання цих трьох спеціалістів в одній особі. Фактично це не можливо і тому йдуть, по-перше, на використання попередньо розроблених шаблонів (щоб зменшити необхідність у художниках), по-друге, на використання розроблених плагінів (щоб зменшити необхідність у програмістах). Але розробка Web-додатків ("Web-приложений"- російською мовою) виконується Web-програмістами. Складності полягають і в тому, що Web-дизайнер це наполовину технічна професія, а Web-програміст – технічна. Якщо розглянути глибше, то і програміст має в собі декілька рівнів (Ф.Брукс "Мифический человеко-месяц или как разрабатываются программные системы"). Все більше значення набуває Web-програміст. Якщо дизайнеру ще іноді вдається засвоїти програмування з боку клієнта (звичайно JavaScript), то серверні додатки створюються професійними програмістами (програмування на мовах PHP і це нижній рівень, а дійсно професійні програмісти працюють на мовах C++ та Java



).

Рис. 2.2 Структура розробки

В деяких публікаціях є висловлювання, що сучасний сайт це програмний продукт. Якщо сайт і звичайний та простий, але вимагає засоби спілкування, то вони створюються програмістами. До засобів спілкування чи активних компонентів відносяться форми, гостьові кніжки, форуми та ін. Можна взяти стандартну розробку, але фільтрацію інформації для більшості активних компонентів треба робити. Обумовлено це багатьма причинами і в основному пов'язано з шумом, який генерує користувач - нецензурна лексика,

просте клацання,образи та інше.Структура процесу розробки Web-сайту.Якщо розглядати Веб-дизайн як Веб-проектування, то загальна структура процесу розробки може бути представлена у вигляді, наданому на рисунку 2.2

Функції замовника та розробників Web-сайту. Розглянемо дещо ширше функції замовника та розробників Web-сайту. Замовник є однією з найбільш важливих ланок розробки. Він, по-перше, вирішує задачу обґрунтування необхідності сайту, по-друге, повинен мати інформаційне наповнення сайту, пропозиції з його структури та графічному оформленню, і що не менш важливо, так це фінансові ресурси на розробку та супроводження сайту. Замовник це ланцюг розробки, який приймає чи повинен приймати участь майже в усіх етапах розробки сайту. Він визначає і назви розділів, і можливе скорочення назв, і першорядні та другорядні розділи та інше. Якість сайту теж суттєво залежить від замовника. Сайт не повноцінний, якщо є не повним інформаційне наповнення. Сайт вмирає, якщо він не супроводжується, а ця функція звичайно лягає на замовника та його спеціалістів. Фінансує і контролює роботу Web-майстра замовник.

Форма та методи передачі інформації від замовника до розробників сайту може бути різною, але завжди стандартизованою. Так, зважаючи на ієрархічну графову модель сайту, доцільно використовувати файлову передачу інформації, що відноситься до кожної "Вершини виконання дій" При цьому номери вершин можуть співпадати з іменами файлів. Замовник може використовувати різні текстові процесори (але більшість буде використовувати WORD, а це при створенні Веб-сторінок накладає деякі обмеження і додаткові дії).

Питання супроводження. Загалом про сайт можна казати як про живу істоту. Вона народжується, живе і вмирає. Щоб вона (істота) жила їй треба годувати. Для сайта "годувати" це поновлювати інформацію, додавати нові функції та можливості. Сайт не тільки дивляться через Інтернет користувачі, але й збирається про нього інформація, оцінюються його можливості. І, доречі, деякі параметри залежать від того скільки і як успішно живе сайт.

РОЗДІЛ 3

ОСОБЛИВОСТІ ВИКОРИСТАННЯ WEB-ДОДАТКУ

“СТРОЙОВА ЧАСТИНА”

3.1. Порядок використання web-додатку

Експлуатація не складна, щоб потрапи на сайт потрібно перейти на посилання: <https://luigi-samponi.github.io/kursova/> Сайт не складний і зорієнтуватися в ньому не складно, головна сторінка виглядає наступним чином (дивись рис. 3.1-3.3).

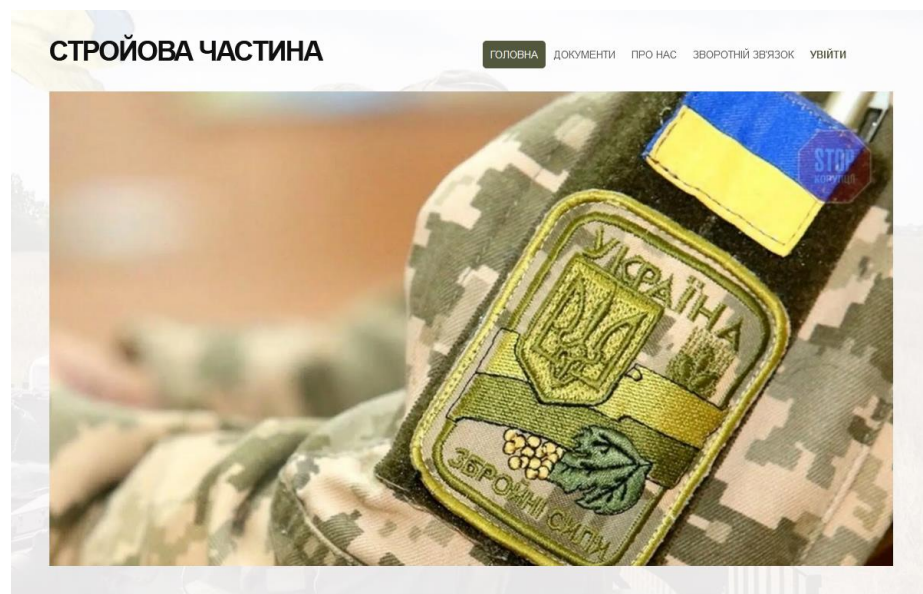


Рис. 3.1 Верх головної сторінки



Рис. 3.2 Перша частина головної сторінки

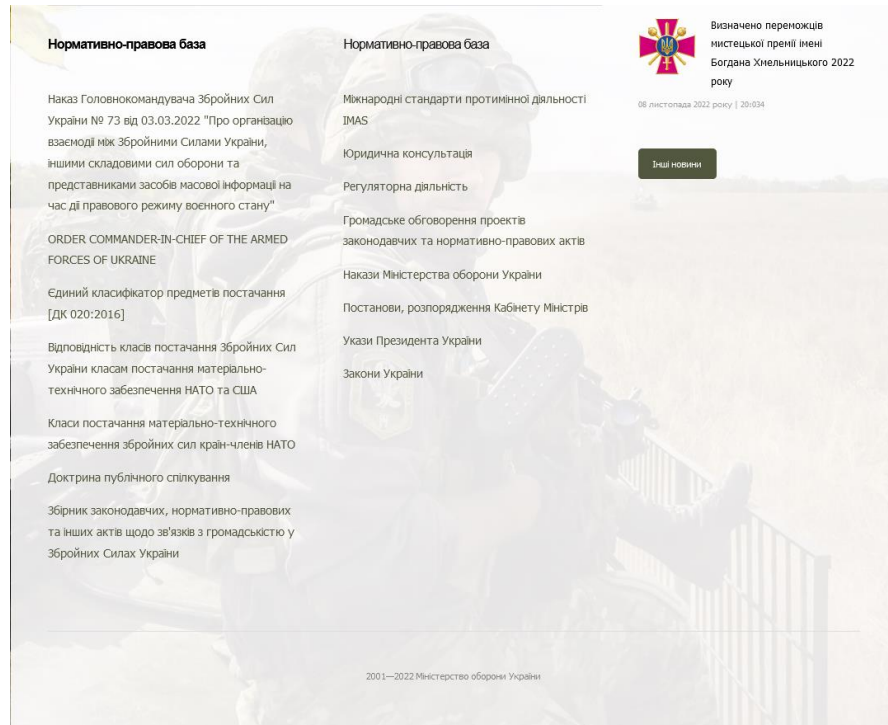


Рисунок 3.3 Низ головної сторінки

3.2 Аналіз виконання вимог

Вимоги курсового проекту виконано повною мірою, а саме реалізовано графічну частину, розроблено внутрішню логіку web-додатку. Програма має високу динаміку, завдяки оригінальному дизайну.

ВИСНОВКИ

На сьогоднішній день лишається актуальним питання відсутності реалізації сайту бойової частини Міністерства Оборони України, тому гостро стоїть питання в реалізації такого продукту. Одним з шляхів вирішення цієї проблеми є створення сайту.

У ході виконання курсового проекту досліджено питання дизайну сайту. Додаток дає можливість підвищити рівень освіченості. Також в наслідок створення динамічної, простої та яскравої графічної реалізації додатку, підіймається морально-психологічний стан користувача.

Розроблений додаток згідно поставленого завдання. Розглянуто принцип реалізації та роботи додатку, його основні класи і методи.

Також вказано особливості використання сайту. Як відбувається запуск, початкові дії та подальша робота сайту. Вказані особливості використання додатку. Розроблена інструкція для користувача з допомогою якої він може зрозуміти принцип роботи сайту і вміти нею користуватися.

Розглянувши всі питання курсової роботи та вирішення цих питань створюється можливість зробити висновок, що всі завдання і вимоги курсового проекту було досягнуто. Тому мета курсового проекту досягнута.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://getbootstrap.com/>
2. <https://css.in.ua/html/tags>
3. <https://html5book.ru/css-css3/>
4. <https://jquery.com/>
5. <https://www.mil.gov.ua/ministry/struktura-aparatu-ministerstva/upravlinnya-fizichnoi-kulturi-i-sportu-ministerstva-oboroni-ukraini.html>
6. https://vpzhr.gp.gov.ua/ua/news.html?_m=publications&_t=rec&id=211730&fp=80

ДОДАТОК А

<https://luigi-samponi.github.io/kursova/>