

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**ИЗУЧЕНИЕ И ДЕМОНСТРАЦИЯ ТЕХНОЛОГИЙ WEB-РАЗРАБОТКИ
НА ПРИМЕРЕ РЕКЛАМНОГО ПОРТАЛА
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Репенинга Павла Алексеевича

Научный руководитель
доцент, к. ф.-м. н.

А. Д. Панферов

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2020

ВВЕДЕНИЕ

В настоящее время мир во многом стал зависим от услуг, предоставляемых в сети Интернет. Каждый человек, имея свои предпочтения и интересы, стремится оставаться в курсе событий, касающихся его увлечений. Для этого создаются различные тематические сайты, с помощью которых происходит распространение большого количества информации через удобное устройство, будь то смартфон, планшет, ноутбук или персональный компьютер. Таким образом из-за быстрого прорыва в этой сфере, распространение рекламы перешло на новый уровень. Сейчас рекламу можно увидеть повсюду, начиная с социальных сетей и заканчивая посещением любой страницы поисковой системы.

Потоки информации, циркулирующие в мире, который нас окружает, настолько огромны, что мы уже не обращаем внимание на рекламу, которую перелистываем по много раз каждый день. Таким образом рекламодатели заинтересованы в таком способе донесения информации, чтобы они были уверены, что пользователь посмотрел рекламу, усвоил информацию и даже отправил обратную связь.

Из сказанного выше, актуальность разработки такого приложения в котором будут заинтересованы как рекламодатели, так и пользователи обусловлена непрерывным увеличением числа активных пользователей сети Интернет и востребованностью рекламы в современном Интернет-пространстве.

Дипломная работа состоит из введения, 4 разделов, заключения и списка литературы.

КРАТКОЕ СОДЕРЖАНИЕ

1) Постановка задачи

Подраздел 1.1 «Цель работы» знакомит с основными используемыми в работе языками для разработки и описывает цель работы.

Сейчас в Интернете используется довольно много различных технологий и из-за технических возможностей были выбраны следующие: JavaScript, jQuery, Webix, Bootstrap, HTML, CSS, Java, Tomcat, Memcached, PostgreSQL, EnjoyHint. Современные веб-технологии предоставляют разработчикам неограниченные возможности для реализации своих идей. Для того, чтобы вы могли использовать весь их потенциал, необходимо знать как правильно их использовать [1].

Целью данной работы является изучение и освоение технологий web-разработки, которые были представлены выше, для применения этих навыков в разработке рекламного портала адаптированного под разного рода устройства, в котором рекламодатели смогут выкладывать свои материалы для рекламы и получать статистику о пользователях, которые посмотрели её и обратную связь от них, а пользователи будут получать какое-то вознаграждение, в данном случае материальное. Полный список требований описан в подразделе 1.2 «Техническое задание».

Цель работы предполагает разработку нескольких приложений, которые будут объединяться в программный комплекс, позволяющий рекламодателю предоставлять рекламу любого вида (видео, картинка, опрос) авторизованным пользователям для ознакомления с ней и получением вознаграждения за уделённое время. Так же рекламодатели смогут получать статистику о пользователях, просмотревших их материал, а пользователи смогут выводить материальное вознаграждение через различные сервисы. Этот комплекс включает в себя администрирование приложения, которое отвечает за работоспособность проекта, получение необходимой информации об ошибках как на стороне сервера, так и на стороне браузера, также рассмотрен функционал контроля версий и своевременного тестирования этих версий.

2) Выбор средств разработки

Подраздел 2.1 «Языки программирования и инструментарий» кратко описывается используемый инструментарий в работе.

В качестве основных сред разработки выбраны:

1. Atom - бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом. Разработчики GitHub Inc [3]. Данный редактор хорошо подходит для создания приложений под веб, т.е. Писать код на HTML, CSS, JavaScript;
2. NetBeans - бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения. Среда предоставляет все средства, необходимые для создания профессиональных десктоп приложений, корпоративных, мобильных и веб-приложений на платформе Java, а также C/C++, PHP, JavaScript, Groovy и Ruby. Разработчики Apache Software Foundation, Oracle и Sun Microsystems [4].
3. JavaScript (часто просто JS) — это легковесный, интерпретируемый или JIT-компилируемый, объектно-ориентированный язык с функциями первого класса. Наиболее широкое применение находит как язык сценариев веб-страниц, но также используется и в других программных продуктах, например, node.js или Apache CouchDB. JavaScript это прототипно-ориентированный, мультипарадигменный язык с динамической типизацией, который поддерживает объектно-ориентированный, императивный и декларативный (например, функциональное программирование) стили программирования [5].
4. jQuery — набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержанию элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX [6]. JQuery хорошо помогает в использовании разных функций и сокращает код в разы, например если функция на чистом JS была бы написана в 10 строк кода, то JQuery сократит её до 2-3.
5. Webix — это набор инструментов JavaScript / HTML5 / CSS3 UI для разработки сложных и динамических кроссплатформенных веб-приложений. Каркас разработан ИТ-аутсорсинговой компанией XB Software со штаб-квартирой в Минске, Беларусь [7]. Библиотека webix имеет обширный

набор виджетов которые удобно использовать и добавлять на свои сайты, она избавляет от громоздкого кода HTML и CSS, сворачивая его в простые и понятные конструкции.

6. HTML (HyperText Markup Language — «язык гипертекстовой разметки») — самый базовый строительный блок Веба. Он определяет содержание и структуру веб-контента. HTML использует разметку для отображения текста, изображений и другого контента в веб-браузере. HTML-разметка включает в себя специальные "элементы такие как `<head>`, `<title>`, `<body>`, `<header>`, `<footer>`, `<article>`, `<section>`, `<p>`, `<div>`, ``, ``, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`, `<embed>`, `<nav>`, `<output>`, `<progress>`, `<video>` и многие другие [8].
7. CSS (Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL [9].
8. Bootstrap - HTML/CSS/JS-фреймворк. Это набор CSS-стилей и JavaScript-скриптов для быстрого создания современных адаптивных сайтов [10].
9. Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения. Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее [11].
10. Tomcat — это контейнер сервлетов с открытым исходным кодом, который также выполняет функцию веб-сервера. Большинство приложений Java запускаются с помощью командной строки и выполняют некоторые действия. Такие приложения реализуют одну заранее заданную

функцию, после чего больше не выполняются. У таких программ, как правило, есть метод `main`, через который их можно запустить. Веб-приложение рассчитано на взаимодействие с клиентом. Если есть запрос от клиента, он обрабатывается, и пользователю отправляется ответ. Если нет, приложение простаивает. Как реализовать такую логику в стандартном приложении, учитывая, что нужно поддерживать сессии, принимать HTTP-запросы и прочее? Цикл `while-true`? Нет, здесь нужно надежное решение. Для этого и существует Tomcat. Фактически он представляет собой Java-приложение, которое заботится об открытии порта для взаимодействия с клиентом, настройке сессий, количестве запросов, длине заголовка и еще многих операциях [12].

11. Memcached – это удобное высокопроизводительное хранилище данных в памяти. Продуманное масштабируемое решение с открытым исходным кодом обеспечивает время отклика на уровне долей миллисекунды, что позволяет использовать его в качестве кэша или хранилища сессий. Memcached широко применяется для поддержки рекламных технологий, площадок интернет-коммерции, игровых, мобильных и интернет-приложений, а также других приложений, работающих в режиме реального времени [13].
12. EnjoyHint – Бесплатный веб-инструмент, который был создан, чтобы помочь пользователям упростить навигацию по сайту или приложению, делая этот процесс интуитивным и простым. Он отлично подходит для создания интерактивных туториалов с подсказками, которые можно добавить в свое веб-приложение, тем самым повысив удобство и эффективность его использования. EnjoyHint позволяет добавлять простые подсказки наряду с мгновенной автофокусировкой на соответствующей зоне или элементе с его подсветкой [14].

Подраздел 2.2 «Технологий программирования» знакомит с технологиями разработки серверной части и веб-приложения.

Общий подход к созданию программного комплекса – это декомпозиция приложений по зоне ответственности и клиент-серверная архитектура ПО

Серверная часть комплекса предоставляет программный интерфейс (API) для своих клиентов (мобильного приложения и веб-сайта) и отвечает за сохранение и выдачу данных, а также инкапсулирует (скрывает реализацию)

базы данных. API выполнен по технологии «передачи состояния представления» (Representational State Transfer или REST). Технология REST позволяет эффективно строить распределенные клиент-серверные комплексы, так как между клиентом и сервером отсутствует понятие сессии и постоянного подключения, и в каждом обращении к серверу и ответе от сервера передается достаточно информации, чтобы модели клиента или сервера находились в непротиворечивом состоянии. Это позволяет горизонтально масштабировать подобные комплексы, и любой запрос от клиента может быть отправлен на случайный экземпляр серверной части. Также, особенностью REST технологии является то, что она работает поверх протокола HTTP – протокола прикладного уровня, являющегося основным средством передачи данных в современном Интернете [15]. Это позволяет одинаково просто интегрировать с REST API все типы программного обеспечения, в том числе и веб-сайты. Из этой особенности вытекает и удобство отладки API на базе REST: протокол HTTP позволяет сделать запрос на сервер прямо из браузера.

Веб-приложение выполнено по технологии «Одностраничное приложение» (Single Page Application или SPA). Это архитектурный подход, при котором сайт состоит из одной, основной HTML-страницы, являющейся оболочкой для всех остальных страниц сайта, а взаимодействие с пользователем и логика выполнена при помощи динамически подгружаемых HTML и JavaScript модулей. В основном приложении для реализации данного архитектурного подхода был использован JavaScript Framework jQuery создателем которого является Джон Резит. Данный Framework является одним из самых распространённых для такого типа веб-приложений, хорошо документирован и имеет множество примеров использования. В части администрирования приложения используется Webix.

3) Проектирование структуры программного комплекса

В подразделе 3.1 «Общая архитектура программного комплекса» описывается общая архитектура программного комплекса.

Программный комплекс выполнен по принципу клиент-серверной архитектуры, состоит из следующих компонентов: серверная часть (сервер, предоставляющий API), основное приложение и администрирования (клиент API).

Взаимодействие между компонентами комплекса происходит по протоколу HTTPS. HTTPS – это обычный HTTP протокол, работающий шифро-

ванные транспортные механизмы SSL (в данном случае TLS). Данный протокол обеспечивает защиту от прослушивания соединения и перехвата пересылаемых данных [16].

На уровне приложения серверная часть предоставляет API по технологии REST, описанной выше в подразделе 2.2 «Технологии программирования».

В подразделе 3.2 «Архитектура серверной части комплекса» описывается основная архитектура серверной части.

HAProxy— серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов [17], в нашем случае перенаправляет входящие запросы java-приложению, которое находит на веб-сервере Tomcat, java-приложение берёт данные любого вида из базы данных, файлового хранилища и кеша. Серверная часть программного комплекса взаимодействует с веб-приложением, архитектура которого описана в подразделе 3.3 «Архитектура веб-приложения».

Как упоминалось ранее, в подразделе 2.2 Технологии программирования, сайт с точки зрения архитектуры построен по технологии «Одностраничного приложения». В отличие от традиционного подхода, когда на каждое действие пользователя происходит перезагрузка и очередной вывод страницы, в одностраничном приложении, загрузка приложения в браузер происходит один раз и далее все взаимодействие с сервером идёт в фоне, по технологии AJAX при помощи языка JavaScript. То есть, когда пользователь работает с элементами управления, не происходит полной перезагрузки страницы. JavaScript подгружает необходимую порцию информации и обновляет конкретные элементы страницы (узлы DOM-дерева).

4) Программная реализация компонентов комплекса

В подразделе 4.1 «Структура данных в СУБД PostgreSQL» описывается работа СУБД.

Программная реализация серверной части программного комплекса, архитектура которого описывалась в подразделе 3.2 «Архитектура серверной части комплекса» рассматривается в подраздел 4.2 «Серверная часть комплекса». Используемый веб-сервер для серверной части приложения, описывается в подраздел 4.3 «Использование веб-сервера Tomcat».

Следующие подразделы показывают результаты работы веб-приложения: подраздел 4.4 «Пользовательский интерфейс основного приложения и его возможности в использовании»; подраздел 4.5 «Пользовательский интерфейс и использование конструктора подсказок»; подраздел 4.6 «Пользовательский интерфейс администрирования и его возможности в использовании».

В подразделе 4.7 «Архитектура загрузки веб-приложения» описывается архитектура загрузки страниц веб-приложения из базы данных.

ЗАКЛЮЧЕНИЕ

Для выполнения поставленных требований был разработан программный комплекс из четырёх приложений: серверной части, мобильная и десктопная версия основного приложения и приложение для администрирования. Поставленные цели и требования были достигнуты. В качестве последующего развития комплекса имеет смысл добавить поддержку большего количества языков, также разработать мобильное приложения на платформы (iOS, Windows Phone, Android).

В ходе разработки программного комплекса, автором был получен опыт: разработки приложений используя современные web-технологии; подготовки приложения к выпуску; адаптации интерфейса на мобильные устройства основываясь на его особенностях; сетевого взаимодействия клиент-серверного приложения с применением клиентской аутентификации. Подводя итоги проделанной работы, можно утверждать, что поставленные требования выполнены, а главная цель – достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Web-технологии для разработчиков [Электронный ресурс] : [сайт]. - URL: <https://developer.mozilla.org/ru/docs/Web> (Дата обращения: 17.05.2020). - Загл. с экр. - Яз. рус.
- 2 Яндекс.Директ - контекстная реклама на Яндексе [Электронный ресурс]: [сайт]. - URL: <https://direct.yandex.ru/> (Дата обращения: 22.05.2020). - Загл. с экр. - Яз. рус.
- 3 Atom (текстовый редактор) - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/Atom> (Дата обращения: 15.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 22:13, 25 марта 2020 года).
- 4 NetBeans - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/NetBeans> (Дата обращения: 15.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 02:18, 4 ноября 2019 года).
- 5 JavaScript | MDN [Электронный ресурс]: [сайт]. - URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (Дата обращения: 17.05.2020). - Загл. с экр. - Яз. рус.
- 6 Адам Фримен jQuery для профессионалов/ Адам Фримен - Вильямс 2012. — 960 с.
- 7 webix — JavaScript UI Фреймворк [Электронный ресурс]: [сайт]. - URL: <https://ru.webix.com/> (Дата обращения: 20.05.2020). - Загл. с экр. - Яз. рус.
- 8 HTML | MDN [Электронный ресурс]: [сайт]. - URL: <https://developer.mozilla.org/ru/docs/Web/HTML> (Дата обращения: 18.05.2020). - Загл. с экр. - Яз. рус.
- 9 CSS - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/CSS> (Дата обращения: 15.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 07:58, 23 апреля 2020 года).

- 10 Что такое bootstrap - Hexlet Guides [Электронный ресурс]: [сайт]. - URL: <https://guides.hexlet.io/bootstrap/> (Дата обращения: 18.05.2020). - Загл. с экр. - Яз. рус.
- 11 Java - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/Java> (Дата обращения: 25.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 18:21, 23 мая 2020 года).
- 12 Tomcat в Java [Электронный ресурс]: [сайт]. - URL: <https://javarush.ru/groups/posts/tomcat-v-java> (Дата обращения: 21.05.2020). - Загл. с экр. - Яз. рус.
- 13 Memcached [Электронный ресурс]: [сайт]. - URL: <https://webcreator.ru/technologies/webdev/legacy/memcached> (Дата обращения: 17.05.2020). - Загл. с экр. - Яз. рус.
- 14 EnjoyHint - XB Software [Электронный ресурс]: [сайт]. - URL: <https://xbsoftware.ru/produkty/enjoyhint/> (Дата обращения: 17.05.2020). - Загл. с экр. - Яз. рус.
- 15 HTTP - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/HTTP> (Дата обращения: 15.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 09:47, 13 мая 2020 года).
- 16 HTTPS - Словарь | MDN [Электронный ресурс]: [сайт]. - URL: <https://developer.mozilla.org/ru/docs/Словарь/https> (Дата обращения: 18.05.2020). - Загл. с экр. - Яз. рус.
- 17 НАРоху - Википедия [Электронный ресурс]: [сайт]. - URL: <https://ru.wikipedia.org/wiki/НАРоху> (Дата обращения: 15.05.2020). - Загл. с экр. - Яз. рус. (Последнее изменение страницы: 23:05, 13 апреля 2020 года).
- 18 Пример работы tomcat с сервлетами java [Электронный ресурс]: [сайт]. - URL: <http://javaway.info/primer-raboty-s-tomcat-i-servletami/> (Дата обращения: 21.05.2020). - Загл. с экр. - Яз. рус.

- 19 WEB — Что такое Servlet? [Электронный ресурс]: [сайт]. - URL: <http://java-course.ru/student/book1/servlet/> (Дата обращения: 21.05.2020). - Загл. с экр. - Яз. рус.
- 20 Пример сервлета, hello [Электронный ресурс]: [сайт]. - URL: <http://java-online.ru/servlet-example.html> (Дата обращения: 04.06.2020). - Загл. с экр. - Яз. рус.