

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ДИНАМИЧЕСКОЙ
СИСТЕМАТИЗАЦИИ ЗАКАЗОВ И ПЛАНИРОВАНИЯ
ЛОГИСТИЧЕСКИХ ЗАДАЧ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Дмитришина Евгения Михайловича

Научный руководитель
ассистент

А. А. Трунов

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

Л. Б. Тяпаев

Саратов 2021

ВВЕДЕНИЕ

С появлением интернета наиболее эффективным средством распространения рекламы и продвижения товаров и услуг является одним из важных элементов современной цивилизации. Интернет может удовлетворить большинство потребностей современного человека. Например, в сфере торговли и продвижения товаров и услуг, всевозможные веб-приложения для торговли онлайн стали неотъемлемой частью общества. Для организации и планирования развития собственного предприятия не обязательно открывать несколько магазинов или арендовать офисное помещение – достаточно иметь хорошее веб-приложение, с описанием, например, товаров и услуг, и место для хранения этих товаров.

Создание веб-приложений сегодня считается одним из перспективных направлений деятельности для многих компаний, занятых в сфере высокотехнологичных цифровых и компьютерных технологий [1]. Для запуска веб-приложения пользователю не нужно устанавливать никаких дополнительных программ, оно запускается на любом устройстве с браузером и с доступом в интернет.

Для создания серверной части веб-приложений используются такие языки программирования, как: PHP, ASP, ASP.NET, Perl, C/C++, Java, Python, Ruby, NodeJS [2]. Для реализации клиентской части используют HTML, CSS, JavaScript, Ajax [2].

Актуальность данной работы обусловлена масштабным развитием новых небольших предприятий, для которых создание веб-приложения, станет эффективным решением широкого спектра бизнес-задач.

Целью выпускной квалификационной работы является разработка веб-приложения для динамической систематизации заказов и планирования логистических задач.

Для достижения цели необходимо решить следующие задачи:

- Систематизация знаний о веб-приложениях;
- систематизация знаний о веб-фреймворках;
- обзор и анализ средств разработки для разработки серверной части веб-приложения при помощи Node.js;
- обзор и анализ средств разработки для создания клиентской части веб-приложения при помощи Node.js;

- разработка веб-приложения для упрощения логистической деятельности предприятия.

Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 110 страниц, из них 47 страниц – основное содержание, включая 43 рисунка, список использованных источников информации – 30 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Веб-приложения» посвящен обзору преимуществ веб-приложений. Также в первом разделе рассматривается основная классификация веб-приложений и программный инструментарий для их создания.

Веб-приложение – это прикладное программное обеспечение, логика которого распределена между сервером и клиентом, а обмен информацией происходит по сети. Клиентская часть реализует пользовательский интерфейс, а серверная – получает и обрабатывает запросы от клиента, выполняет вычисления, формирует веб-страницу и отправляет её клиенту согласно протоколу HTTP [3].

Работа клиента не зависит от операционной системы, стоящей на компьютере пользователя, поэтому при разработке веб-приложений нет необходимости писать отдельные версии для Windows, Linux, Mac OS и других операционных систем [2].

Веб-приложения отличаются от обычных веб-страниц, как правило, более сложным функционалом. В веб-приложениях пользователь работает с динамически сгенерированным контентом. То есть, если какая-то информация будет отображаться на экране, будь это смартфон или привычный всем персональный компьютер, то это напрямую будет зависеть от действий конкретного пользователя.

В современном мире веб разработки существует множество инструментов для создания веб-страниц с динамическим контентом.

Как и привычные каждому пользователю обычные приложения для персонального компьютера, веб-приложения имеют ряд преимуществ:

- Доступ с любого устройства.
- Экономия.
- Адаптивность.
- Отсутствие клиентского ПО.
- Сетевая безопасность.
- Масштабируемость.
- Защита от потери данных.

Существует три основных подхода к разработке веб-приложений: одностраничные, многостраничные и прогрессивные. Они выделяются среди

других подходов простотой разработки, удобством для пользователей и широкими возможностями для развития бизнеса [4].

Веб-фреймворк – это каркас для написания веб-приложений. Он определяет структуру, задаёт правила и предоставляет необходимый набор инструментов для разработки [5].

У фреймворков есть две основные функции: работа на серверной стороне (бэкенд) и работа на клиентской стороне (фронтенд) [6].

К основным типам веб-фреймворков можно отнести:

- Бэкенд-фреймворки.
- Фронтенд-фреймворки.
- Фуллстек-фреймворки.

В первом разделе были рассмотрены основные виды веб-приложений, а также основные фреймворки для их разработки. Второй раздел «Программные средства для разработки веб-приложения» посвящен описанию программных средств для разработки клиентской и серверной части веб-приложения.

Node.js – среда исполнения Java Script. Позволяет исполнять код среды Java Script при помощи любой платформы и с использованием любой операционной системы, не задумываясь о поддержке.

С другой стороны Node.js – это среда выполнения программного кода JavaScript, построенная на основе JavaScript V8 Chrome. Node.js выступает в роли как асинхронная среда выполнения JavaScript, управляемая событиями, и предназначена для создания масштабируемых сетевых веб-приложений.

При каждом подключении запускается обратный вызов, однако если никаких действий не требуется, то Node.js перейдет в спящий режим. Это контрастирует с более распространенной на сегодняшний день моделью параллелизма, в которой используются потоки ОС. Сеть на основе потоков относительно неэффективна и очень сложна в использовании. Более того, пользователи Node.js могут не беспокоиться о тупиковой блокировке процесса, поскольку блокировок нет. Практически ни одна функция в Node.js напрямую не выполняет ввод-вывод, поэтому процесс никогда не блокируется. Поскольку ничто не блокирует, масштабируемые системы очень разумно разрабатывать на Node.js.

Node.js похож по дизайну на такие системы, как Ruby Event Machine и Python Twisted, и находится под их влиянием. Node.js продвигает модель

событий немного дальше. Он представляет цикл событий как конструкцию среды выполнения, а не как библиотеку. В других системах всегда есть блокирующий вызов для запуска цикла событий. Обычно поведение определяется через обратные вызовы в начале скрипта, а в конце сервер запускается через блокирующий вызов, например `EventMachine::run()`. В Node.js нет такого вызова цикла события. Node.js просто входит в цикл обработки событий после выполнения входного скрипта. Node.js выходит из цикла обработки событий, когда больше нет обратных вызовов для выполнения. Это поведение похоже на JavaScript в браузере – цикл событий скрыт от пользователя.

К основным библиотекам для разработки клиентской части можно отнести `react` и `sass`.

React – это JavaScript-библиотека для разработки пользовательского интерфейса.

Отвечает за перерисовку отдельных компонент что ускоряет рендеринг страницы [7].

К примеру, для демонстрации примеров можно работать в своем текстовом редакторе. Для этого необходимо скачать тестовый HTML-файл, добавить в него код и запустите на своём компьютере. При открытии тестового файла браузер преобразует JSX в обычный код на JavaScript. Такое преобразование достаточно медленно, поэтому мы рекомендуем использовать этот файл только для простых демонстрационных примеров.

React изначально был спроектирован так, чтобы его можно было внедрять постепенно. Другими словами, вы можете начать с малого и использовать только ту функциональность React, которая необходима вам в данный момент [7].

Sass – это один из самых популярных CSS-препроцессоров. Он помогает писать чистый, повторно используемый и модульный CSS.

`node-sass` используется для сборки стилей проекта, поддерживает больше возможностей, ускоряющих разработку и возможность использовать код многократно, нежели CSS [8].

Во втором разделе были рассмотрены программные средства, применяемые при разработке веб-приложения.

Третий раздел «Описание структуры веб-приложения» посвящен описанию разработанного веб-приложения, с примерами работы его функцио-

нальной части.

Главной концепцией данного веб-сервиса является автоматизация и планирование маршрутов доставки товаров для пользователей, поэтому для удобства составления и планирования маршрутов доставки, все совершенные заказы сортируются по районам доставки и количеству заказов по заданному району. Сортировка по районам происходит по количеству заказов и выводится по убыванию в виде небольшой «карусели», в которой показаны какие товары были заказаны для конкретного района, в каком количестве и информация о пользователе, сделавшим заказ. Каждый пользователь имеет возможность зайти на сайт и посмотреть ассортимент товаров, представленный в каталоге. Для каждого товара в каталоге доступно краткое описание.

У администратора портала есть доступ к системе для проверки заказов или редактирования описания товара.

Главным преимуществом данного веб-приложения является кроссплатформенность. Иными словами, пользователь может воспользоваться данным сервисом не только используя персональный компьютер, а также и воспользоваться со смартфона.

Таким образом, можно планировать свой маршрут, например начиная доставку в том районе, где было выполнено наибольшее число заказов, или наоборот, построить удобный маршрут, состоящий из нескольких районов с наименьшими заказами. Причем, возможность зайти с любого мобильного устройства позволяет оперативно откликаться на поступление новых заказов.

ЗАКЛЮЧЕНИЕ

Последние годы веб-приложения стремительно развиваются, постепенно вытесняя настольные решения и становясь важнейшим компонентом бизнеса в современном мире.

Веб-приложение – это веб-сайт, с частично разработанными страницами, конечное содержание которых определяется по запросу пользователя. Первая страница, на которую попадает посетитель после входа в браузер называется статической, т.е. ее содержимое всегда стабильно, неизменно. Большинство остальных страниц, на которые пользователь попадает, нажав на кнопку или зайдя во вкладку – динамические, т.к. формируются под определенный запрос пользователя [9].

При разработки собственного веб-приложения зачастую требуется использование различных подключаемых модулей. С помощью фреймворков можно реализовать большинство важных функций, таких как [10]:

- Маршрутизация URL-адресов
- Шаблоны вывода
- Управление базами данных
- Управление сеансом
- Безопасность от стандартных атак

Также, при использовании фреймворков упрощается разработка веб-приложений, которая избавляет пользователя от необходимости многократного написания одних и тех же фрагментов кода.

При выполнении выпускной квалификационной работы был произведен анализ и выбор фреймворков и библиотек, для разработки веб-приложения на Node.js.

В практической части работы было разработано веб-приложение для динамической систематизации заказов и планирования логистических задач для доставки товаров. Главным преимуществом подобных систем является их кроссплатформенность. Иными словами, для пользователя не важно, какая операционная система установлена на устройстве его устройстве для того, чтобы воспользоваться веб-приложением.

Таким образом, цель и задачи выпускной квалификационной работы были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка WEB-приложений: [Электронный ресурс] – URL: <http://clashdash.ru/directions/web-applications> (дата обращения 25.05.21) – Загл. с экрана – Яз. рус.
2. РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ [Электронный ресурс] – URL: <https://infoshell.ru/blog/razrabotka-veb-prilozhenij/> (дата обращения 20.05.21) – Загл. с экрана – Яз. рус.
3. Разработка веб-приложений. Шасанкар К. СПб.: Питер, 2014. — 208 с.
4. Одностраничные (spa) и многостраничные (pwa) веб-приложения: [Электронный ресурс] – URL: <https://vc.ru/seo/108149-odnostranichnye-spa-i-mnogostranichnye-pwa-veb-prilozheniya> (дата обращения 10.04.21) – Загл. с экрана – Яз. рус
5. Веб-фреймворки для начинающих: простое объяснение с примерами: [Электронный ресурс] – URL: <https://tproger.ru/translations/web-frameworks-how-to-get-started/#part1> (дата обращения 19.04.21) – Загл. с экрана – Яз. рус.
6. Веб-фреймворки: введение для новичков: [Электронный ресурс] – URL: <https://tproger.ru/translations/web-frameworks-how-to-get-started/> (дата обращения 13.05.21) – Загл. с экрана – Яз. англ.
7. Факультет веб-разработки [Электронный ресурс] – URL: https://gb.ru/geek_university/web?utm_source=google&utm_medium=cpc&utm_campaign=google_s_russia_web_razrabotchik_main_lp_geekuniversity_web_prog&utm_term=html%20javascript&utm_content=cid:6451099602|gid:116088789761|aid:487898103546|pos:|st:g|src:|dvc:c|reg:1012036&gclid=CjwKCAjw47eFBhA9EiwAy8kzNGVcrFqip0qtmXU3EKYI2YFWTCt339S0jmqM3T1owXvOrlIwgQDjtBoC1s0QAvD_BwE (дата обращения 20.05.21) – Загл. с экрана – Яз. рус.
8. Отслеживаем и компилируем Sass в 5 быстрых шагов: [Электронный ресурс] – URL: <https://webformyself.com/otslezhivaem-i-kompiliruem-sass-v-5-bystryx-shagov/> (дата обращения 20.05.21) – Загл. с экрана – Яз. англ.
9. WEB-приложение или мобильное приложение. Что выбрать?. [Электронный ресурс]. – URL: <https://punicapp.com/blog/pages/893/web-prilozhenie-ili-mobilnoe-prilozhenie-chto-vybrat> (дата обращения 23.04.21) – Загл. с экрана – Яз. англ.

10. Введение в создание веб-приложений на Python. [Электронный ресурс].
– URL: <https://proglib.io/p/python-web-development/> (дата обращения 23.04.21) – Загл. с экрана – Яз. англ.