

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 WEB-ПРИЛОЖЕНИЯ ДЛЯ ОРГАНИЗАЦИИ СПИСКА
ЗАДАЧ И ДОЛГОСРОЧНОГО ПЛАНИРОВАНИЯ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Экгарт Викентия Александровича

Научный руководитель
ассистент

А. А. Трунов

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

Л. Б. Тяпаев

Саратов 2020

ВВЕДЕНИЕ

На данный момент веб-приложения стремительно развиваются, постепенно вытесняя настольные решения и становясь важнейшим компонентом бизнеса в современном мире.

Все чаще компании прибегают к услугам разработки веб-приложений (Web-application), чтобы эффективно решать широкий спектр бизнес-задач.

Для запуска веб-приложения пользователю не нужно устанавливать никаких дополнительных программ, оно запускается на любом устройстве с браузером и с доступом в интернет.

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.

Для создания серверной части веб-приложений используются такие языки программирования, как: PHP, ASP, ASP.NET, Perl, C/C++, Java, Python, Ruby, NodeJS.

Для реализации клиентской части используют HTML, CSS, JavaScript [1].

Актуальность использования фреймворков можно объяснить в масштабном скачке развития области веб-разработки с целью уменьшения времени для создания веб-приложения.

Цель бакалаврской работы – разработка веб-приложения для динамической систематизации и сохранения важных заметок с устройств под управлением разных операционных систем.

Для достижения цели необходимо решить следующие задачи:

- систематизация знаний об объекте исследования, его характеристиках, особенностях;
- провести анализ средств для разработки веб-приложений;
- провести классификацию и анализ фреймворков для создания веб-приложения;
- выбрать фреймворки для разработки веб-приложения для организации списка задач и долгосрочного планирования;
- разработать программную модель системы для организации списка задач и долгосрочного планирования.

Бакалаврская работа состоит из введения, 2 разделов, заключения, списка использованных источников и 6 приложений. Общий объем работы – 63 страницы, из них 41 страница – основное содержание, включая 33 рисунка, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Инструменты для создания веб-страниц» посвящен обзору различных методов и подходов при разработке веб-приложений.

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

Первый способ – привычный разработчикам с использованием JavaScript + CSS + HTML.

Пользователь делает какое-либо действие, например, клик на сайте. При этом действию скрипт обрабатывает этот клик, делает запрос на какой-либо сервер, потом проверяет ответ и если всё хорошо, то происходит отрисовка пришедшего ответа в нужном месте HTML страницы. Таким образом, например работают социальные сети: Instagram, Facebook и ВКонтакте.

Второй способ для создания динамического веб-контента – это использование Server Side Rendering(SSR) [2]. Веб-приложения, построенные с помощью этой технологии, работают следующим образом. Пользователь какое-либо действие, например, клик на сайте, браузер выполняет запрос на сервер, на сервере происходит построение новой HTML страницы с подстановкой нужных данных, и уже готовая HTML страница возвращается клиенту в готовом виде. Далее браузер просто отображает обновленную страницу. Таким образом вся вычислительная работа ложится непосредственно на сервер.

Какой из двух способов выбрать для разработки проекта, зависит от некоторых условий. В первую очередь, многое здесь будет зависеть от умения и навыков каждого члена команды. Также стоит отметить и всевозможные специфические требования и ограничения со стороны браузеров. Например, если требуется поддержка устаревших браузеров, таких как Internet Explorer 6 версии, то конечно, рендеринг страницы лучше делать на сервере.

Также, выбор одного из подходов может зависеть непосредственно от имеющегося бюджета и предполагаемого количества пользователей.

Очевидно, что первый подход существенно снижает нагрузку на сервер. Также, разделение логики на клиентскую и серверную весьма способствует лучшей поддержке приложения во время его жизненного цикла.

С появлением веб-фреймворков все трудности, которые связаны с изменением структуры приложения, были оптимизированы благодаря появлению общего подхода к разработке веб-приложений.

Веб-фреймворк – это платформа для создания сайтов и веб-приложений, облегчающее разработку и объединение разных компонентов большого программного проекта. За счёт широких возможностей в реализации бизнес-логики и высокой производительности эта платформа особенно хорошо подходит для создания сложных сайтов, бизнес-приложений и веб-сервисов [3].

У фреймворков есть две основные функции: работа на серверной стороне (бэкенд) и работа на клиентской стороне (фронтенд) [4].

Фронтенд – фреймворки связаны с внешней частью приложения [4]. Иными словами, фреймворки данного типа отвечают за внешний вид приложения. В свою очередь Бэкенд-фреймворки отвечают за внутреннее устройство приложения [3].

На данный момент имеется множество фреймворков, позволяющие выбрать необходимый для решения определенной задачи. Для этого, можно использовать как один конкретный фреймворк, которого будет достаточно, так и несколько фреймворков, работающих совместно.

Библиотеки и фреймворки делятся на две части по принципу использования – одни из них добавляют UI элементы на страницу, другие же отвечают за манипуляции с данными (способы их отрисовки, форматирование, забрать данные с сервера, отобразить лишь нужные данные в нужный момент).

В первом разделе описаны методы разработки веб-страниц, а также описаны фреймворки и библиотеки, благодаря которым существенно облегчается разработка веб-приложений на языке программирования JavaScript.

Второй раздел «Описание работы веб-приложения» посвящен созданию программного продукта для организации списка задач в виде кроссплатформенного веб-приложения.

В основе данного сервиса лежит Google Firebase и Google Cloud Functions. Это облачная база данных в совокупности с backend-логикой от Google, внутри которых находятся WEB-Sockets.

Веб-сокеты позволяют очень быстро получать данные с сервера или отправлять их туда. А также служат для мгновенной синхронизация между устройствами. Иными словами, стоит только добавить задачу с мобильного устройства, как она тут же отобразится на остальных, словно используем привычный всем Google Docs.

Также на данном веб-сервисе используется система оплаты pay as you go

– если придёт десять тысяч новых пользователей, то система автоматически масштабируется и не сломается под натиском.

Cloud Functions позволяют написать функции, выполняются которые будут на серверах Google и делать «тяжелую» работу, которую не выгодно делать на «клиенте». В данном проекте через Cloud Functions реализовано автоматическое удаление всех задач, после того как проект, связанный с этими задачами, удаляется.

В целом облачные технологии Google Firebase позволяет не тратить время на написание примитивных CRUD операций и не задумываться об аренде сервера.

Клиентская часть (frontend) приложения написана с использованием фреймворка vue.js. Также используется webpack для сборки клиентской части и npm – менеджер пакетов и необходимых зависимостей. Для удобной работы с датой, на клиенте используется moment JS.

Данный сервис доступен по адресу: ikiteo.ru.

Вход для пользователей обеспечен с авторизацией с использованием сервисов Google. Выбор именно Google-аккаунта обусловлен тем, что у большинства пользователей имеется учетная запись Google, а непосредственно в основе сервиса лежит Google Firebase и Google Cloud Functions. Функция авторизации позволяет сохранять пользователю свои записи, заметки, данные и синхронизировать их между своими устройствами.

Данное веб-приложение позиционирует себя не сколько как блокнот, куда вписываешь какие-то повседневные задачи, вроде похода по магазинам и списка продуктов, которые необходимо купить, а именно твой планировщик задач, который постоянно будет у тебя под рукой и можно распланировать свой день вплоть до минуты, составить личный распорядок дня и подробно расписать задачи для конкретной заметки.

ЗАКЛЮЧЕНИЕ

Фреймворк – это программный продукт, который в определенной степени упрощает разработчику жизнь – он предоставляет готовые решения, что очень важно при работе с проектом. Фреймворки как правило представляют собой инструмент для увеличения скорости разработки.

Благодаря использованию фреймворков можно выделить некоторые их достоинства:

- Повышение скорости и удобства разработки;
- Сокращение затрат на создание приложения;
- Оптимизация рабочего времени;
- Написание чистого кода, не требующего существенного рефакторинга [5].

Благодаря их использованию существенно сокращается время разработки веб-приложений, которые в отличие от обычных пользовательских приложений не прикреплены к определенной операционной системе, и могут быть доступны конечному пользователю при наличии обычного браузера как с персонального компьютера, так и с мобильного устройства.

В выпускной квалификационной работе был произведен анализ и выбор фреймворков и библиотек, для разработки веб-приложения на JavaScript.

В практической части работы было реализовано веб-приложение для организации списка задач. Пользователь, работая с данным сервисом может получить детальную информацию о количестве запланированных задач, а также с легкостью создавать новые задачи или изменять уже имеющиеся, тем самым получая возможность с легкостью планировать свое время.

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

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Разработка WEB-Приложений. [Электронный ресурс]. – URL: <https://infoshell.ru/blog/razrabotka-veb-prilozhenij/> (дата обращения 05.05.20) – Загл. с экрана – Яз. рус
- 2 Серверный или клиентский рендеринг на вебе: что лучше использовать у себя в проекте и почему. [Электронный ресурс]. – URL: <https://tproger.ru/translations/rendering-on-the-web/> (дата обращения 05.05.20) – Загл. с экрана – Яз. рус
- 3 Фреймворки в веб-разработке. [Электронный ресурс]. – URL: https://webcreator.ru/articles/about_frameworks (дата обращения 05.05.20) – Загл. с экрана – Яз. рус
- 4 Веб-фреймворки: введение для новичков. [Электронный ресурс]. – URL: <https://tproger.ru/translations/web-frameworks-how-to-get-started/> (дата обращения 05.05.20) – Загл. с экрана – Яз. рус
- 5 Что такое фреймворки? [Электронный ресурс]. – URL: <https://saikt-online.ru/chto-takoe-frejmvorki-i-dlya-chego-oni-nuzhny/saikt-online.ru/chto-takoe-frejmvorki-i-dlya-chego-oni-nuzhny> (дата обращения 13.05.20) – Загл. с экрана – Яз. рус