МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Гродненский государственный университет имени Янки Купалы»

Факультет математики и информатики

Кафедра современных технологий программирования

ШУЛЯК АРТЕМ ВАЛЕРЬЕВИЧ

ОТЧЕТ  
ПО ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКЕ

студента 3 курса специальности   
1-40 01 01 «Программное обеспечение информационных технологий»

Руководитель от кафедры:   
Урбан О.И,

старший преподаватель кафедры СТП

\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

**Утверждаю** 30.06.2023 г.

Должность: инженер-программист:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Е.Р. Михаил

*(подпись)* М.П.

**Гродно, 2023**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc107432941)

[1. ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ООО «Вибэнд» 4](#_Toc107432942)

[2. ЭТАПЫ ПРОХОЖДЕНИЯ ПРАКТИКИ 8](#_Toc107432943)

[3. ОБЗОР ТЕХНОЛОГИЙ И ФРЕЙМВОРКОВ ДЛЯ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ 10](#_Toc107432944)

[3.1. Язык программирования JavaScript 10](#_Toc107432945)

[3.2. React и Node.js 10](#_Toc107432946)

[4. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ 12](#_Toc107432947)

[4.1. Подготовка индивидуальной темы: Разработка web приложения на React 12](#_Toc107432948)

[4.2 Проектирование и программная реализация приложения 13](#_Toc107432949)

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 19

# ВВЕДЕНИЕ

В период с 5 июня по 30 июня 2023 г. проходил технологическую практику в ООО «Вибэнд».

Wiband — это компания предоставляющая большой спектр услуг в области разработки ПО и приложений. Wiband разрабатывает программное обеспечение для клиентов в самых различных отраслях, включая образование, финансы, страхование, здравоохранение, системы безопасности. Компания предлагает консультационные услуги для стартап-проектов и ориентируется на тесное сотрудничество с клиентами по всему миру.

Цель Wiband – добиться успеха своих клиентов. Благодаря большому опыту в области разработки веб-платформ, подбираются наиболее подходящие решения для эффективного развития бизнеса заказчиков. Разрабатывая ПО для различных индустрий, сотрудничая с партнерами из разных стран, компания постоянно обогащает свой опыт и экспертизу, что содействует получению все более интересных и масштабных проектов.

1. **ОБЩАЯ ХАРАКТЕРИСТИКА ДЕЯТЕЛЬНОСТИ ООО «Вибэнд»**

Являясь резидентом Парка Высоких Технологий, **Wiband специализируется на**:

* Веб и мобильной разработке
* Продуктовой разработке ПО

В процессе реализации проектов используются следующие **технологии**:

* Языки программирования – JavaScript, , PHP, Node.js, HTML5, CSS5
* Фронтенд – Angular, ReactJS, EmberJS, Vue.js, Backbone.js, Nuxt.js HTML5
* UI Фреймворки – Bootstrapm, Foundation, Vuetify, Compass
* CSS просессоры – SASS/SCSS, Less
* CMS – Wordpress, Joomla
* Базы данных – MySQL

**Рабочий процесс** компании строится на последовательности следующих этапов разработки ПО:

1. **Определение проблемы**

Перед тем как приступать к кодированию, необходимо четко сформулировать проблему, которую будущая программа должна решать. Так как, не имея хорошего определения проблемы, можно потратить много усилий и времени на решение не той проблемы, которую требуется решить.

1. **Выработка требований**

Требования к программе – это подробное описание всех возможностей программы и действий, которые должна выполнять программа. Такие требования иногда также называют «Функциональной спецификацией» или просто «Спецификацией».

Требования вырабатывают для того, чтобы свести к минимуму изменения системы после начала непосредственной разработки. Такие требования должны быть обязательно официальными, т.е. документально оформлены. Так как это гарантирует то, что функциональность системы определяется заказчиком, а не программистом.

Выработка требований очень важна, так как она позволяет определить функциональность программы до начала программирования.

1. **Создание плана разработки**

На данном этапе нужно в формальном виде составить план разработки программного обеспечения с учётом существующей проблемы и выработанных требований. Иными словами, необходимо составить план того, как будем действовать дальше.

1. **Разработка архитектуры системы или высокоуровневое проектирование**

Архитектура системы – это каркас программы, это высокоуровневое проектирование программы.

Архитектура системы обычно включает:

* Общее описание системы;
* Основные компоненты;
* Формат и способ хранения данных;
* Специфические бизнес-правила;
* Способ организации пользовательского интерфейса;
* Подход к безопасности системы;
* Оценки производительности;
* Возможности масштабирования;
* Моменты, связанные с интернациональностью, т.е. будет ли система интернациональной.

Кроме того, в архитектуру необходимо включить подтверждение того, что при разработке этой архитектуры рассматривались альтернативные варианты в каждом из вышеперечисленных направлений, с обоснованием окончательного выбора и подхода.

1. **Детальное проектирование**

На этом этапе проводится проектирование программы на низком уровне, иными словами, здесь проектируются классы и методы, рассматриваются, оцениваются и сравниваются различные варианты и причины выбора окончательных подходов и способов реализации.

При реализации крупных проектов данному процессу отводится отдельный этап и проектирование в этом случае проводится с очень высокой степенью детальности.

1. **Кодирование и отладка**

Если все вышеперечисленные этапы выполнены, то данный этап подразумевает чисто механическую работу, т.е. кодинг.

После того как код написан, программисту необходимо отладить этот код, чтобы в нем не было никаких ошибок.

1. **Тестирование компонентов**

После того, как код написан, и проведена отладка, необходимо провести тестирование реализованного функционала. Если программа состоит из нескольких компонентов, сначала тестируют каждый компонент в отдельности, так как очень крупные программы включают огромный функционал, который часто разделяют на отдельные компоненты, разработка которых осуществляется по отдельности. В менее крупных проектах этот этап может включать просто тестирование отдельных классов.

1. **Интеграция компонентов**

Когда тестирование всех компонентов закончено, можно переходить к интеграции всех компонентов в единый программный комплекс, этот этап как раз и подразумевает процесс интеграции, т.е. слияния всех компонентов в единую систему.

1. **Тестирование всей системы**

На данном этапе проводится тестирование всей системы, уже с учётом интеграции всех компонентов. На этом этапе можно выявить проблемы взаимодействия компонентов и устранить их. Также на этом этапе основным предметом тестирования является безопасность, производительность, утечка ресурсов и другие моменты, которые невозможно протестировать на более низких уровнях тестирования.

1. **Сопровождение, внесение изменений, оптимизация**

После запуска программы в промышленную эксплуатацию осуществляется сопровождение этой программы, т.е. внесение изменений на основе выявленных недочетов в процессе эксплуатации системы, а также проводится оптимизация функционала или добавление нового.

1. **ЭТАПЫ ПРОХОЖДЕНИЯ ПРАКТИКИ**

Основные этапы прохождения технологической практики в ООО «Техартгруп» в период с 5 июня по 30 июня 2023 г. приведены в таблице 1.

Таблица 1 – Этапы прохождения практики

|  |  |
| --- | --- |
| Дата | Содержание выполненных работ |
| 05/06/223 | Инструктаж по охране труда (обучение мерам безопасности) в университете. Ознакомление с правилами оформления дневника и отчета |
| 06/06/23 | Инструктаж по охране труда (обучение мерам безопасности) по месту прохождения практики. Ознакомление с работой отдела |
| 07/06/23 | Получение и обсуждение индивидуального задания |
| 08/06/23 | Ознакомление с документацией веб фреймворка React |
| 09/06/23 | Проработка архитектуры Web-API приложения |
| 12/06/23 | Проработка интерфейса приложения |
| 13/06/23 | Изучение веб фреймворка React |
| 14/06/23 | Изучение веб фреймворка React |
| 15/06/23 | Добавление сервисов для CRUD операций к сущностям |
| 16/06/23 | веб фреймворк React + изучение Хуки |
| 19/06/23 | веб фреймворк React + изучение Хуки |
| 20/06/23 | веб фреймворк React + изучение Context API |
| 21/06/23 | веб фреймворк React + изучение Redux |
| 22/06/23 | веб фреймворк React + изучение Redux |
| 23/06/23 | Программирование клиентского приложения |
| 26/06/23 | Изучение Node.js |
| 27/06/23 | Связывание клиентской части приложения с серверной |
| 28/06/23 | Доработки некоторые часи приложения |
| 29/06/23 | Подведение итогов практики, оформление дневника и отчета |
| 30/06/23 | Подведение итогов практики, оформление дневника и отчета |

Задания выдавались руководителем практики от организации. Все задания были выполнены индивидуально с соблюдением необходимых требований техники безопасности. В исключительных случаях при выполнении заданий проводилась консультация с руководителем, а также предоставлялся дополнительный теоретический материал.

# 3. ОБЗОР ТЕХНОЛОГИЙ И ФРЕЙМВОРКОВ ДЛЯ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ

# 3.1. Язык программирования JavaScript

JavaScript – это язык, который позволяет вам применять сложные вещи на web странице — каждый раз, когда на web странице происходит что-то большее, чем просто её статичное отображение — отображение периодически обновляемого контента, или интерактивных карт, или анимация 2D/3D графики, или прокрутка видео в проигрывателе, и т.д. — можете быть уверены, что скорее всего, не обошлось без JavaScript. Это третий слой слоёного пирога стандартных web технологий, два из которых (HTML и CSS) мы детально раскрыли в других частях учебного пособия.

Ещё более увлекательным является функциональность, созданная поверх основного языка JavaScript. Так называемые интерфейсы прикладного программирования (API) предоставляют вам дополнительные сверхспособности для использования в вашем коде JavaScript.

API - это готовые наборы блоков кода, которые позволяют разработчику реализовывать программы, которые в противном случае было бы трудно или невозможно реализовать. Они делают то же самое для программирования, что готовые комплекты мебели делают для домашнего строительства - гораздо проще брать готовые панели и скручивать их вместе, чтобы сделать книжную полку, чем самому разрабатывать дизайн, ходить в поисках правильной древесины, вырезать все панели необходимого размера и формы, найти подходящие винты, а затем собрать их вместе, чтобы сделать книжную полку.

# 3.2. [React и Node.js](#_Toc107432946)

React — популярная фронтенд-библиотека JavaScript для создания интерактивных пользовательских интерфейсов. Node.js — среда бэкенд-разработки JavaScript, которая позволяет создавать масштабируемые веб-приложения с высокой производительностью и низкой задержкой.

Node.js — это среда выполнения JavaScript, которая создается на виртуальных машинах Google Chrome, специально предназначенных для разработки быстрых и масштабируемых сетевых приложений. Node.js использует событийно-ориентированную модель неблокирующего ввода-вывода, что делает этот инструмент легким и эффективным.

Более того, он позволяет создавать веб-приложения с использованием JavaScript на фронтенде и бэкенде. Это дает множество преимуществ компаниям, которые хотят использовать один язык для создания кода как на стороне сервера, так и на стороне клиента.

С помощью React фреймворка Javascript можно создавать повторно используемые компоненты пользовательского интерфейса. Он позволяет разрабатывать крупные веб-приложения для динамического представления управляемых данных, а также обеспечивает высокую производительность и отзывчивость на различных устройствах.

Простая React-модель дает пользователям возможность легко разрабатывать масштабируемые веб-приложения на чистом Javascript без необходимости изучать сложные языки программирования или фреймворки.

# 4. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Задача для прохождения технологической практики представляла собой выполнение следующих пунктов:

1.Изучение теоретических аспектов вопроса разработки веб-приложений.

2.Получить практические навыки разработки web-приложений.

## **4.1. Подготовка индивидуальной темы:** **Комплексная разработка веб-приложений с помощью React и Node.js**

Сочетание этих двух технологий дает огромные преимущества при комплексной разработке веб-приложений.

**Данные в реальном времени:**

Node.JS имеет событийно-ориентированную архитектуру, что делает его идеальным для приложений, работающих с данными, поступающими в реальном времени. При использовании Node.JS с React можно легко создавать мощные веб-приложения, которые эффективно обрабатывают большие объемы трафика.

**Ускорение процесса разработки**:

Использование React и Node.js обеспечивает высокую окупаемость инвестиций (return on investment, ROI), а также экономию времени и денег. Эти две технологии хорошо совмещаются, обеспечивая эффективную платформу для разработки быстрых сайтов, которые легко поддерживать в течение долгого времени.

**Масшабируемость**:

React.js и Node.js позволяют создавать веб-приложения для динамических представлений, управляемых данными и реагирующих на различных устройствах.

## **Один язык для фронтенда и бэкенда:**

Используя React с Node.js, разработчикам не нужно изучать сложные языки бэкенда, такие как Ruby или Python. Они могут использовать один язык для React-разработки пользовательского интерфейса и Node.js-разработки на стороне сервера. Это избавит от необходимости переключаться между различными языками программирования и фреймворков, сэкономив время, деньги и ресурсы.

**Высокая производительность**:

Node.js — идеальная платформа для высокопроизводительных приложений. Она использует событийно-ориентированную модель неблокирующего ввода-вывода, что делает ее легкой и эффективной.

При использовании Node.js с React можно создавать мощные веб-приложения, не беспокоясь о медленном времени отклика или загрузки страниц.

**Одностраничные приложения**:

Node.js и React обеспечивают идеальную платформу для разработки мощных одностраничных приложений. Эти приложения более эффективны, чем традиционные, поскольку они загружаются только один раз, а затем динамически обновляются по мере взаимодействия с ними.

Это может пригодиться при разработке сложных веб-приложений, которые должны обрабатывать большой объем трафика.\

Обе описанные здесь технологии хорошо дополняют друг друга. Node.js — бэкенд-платформа, которая обеспечивает быструю обработку данных и может использоваться для создания веб-серверов на JavaScript. React — фронтенд-фреймворк, позволяющий легко создавать пользовательские интерфейсы из компонентов с помощью Javascript.

**ЗАКЛЮЧЕНИЕ**

В процессе прохождения практики в ООО «Вибэнд» в период с 5 июня по 30 июня 2023 г. были выполнены задачи, поставленные руководителем. В ходе выполнения заданий применялись знания, полученные в университете. Условия практики полностью соответствовали требованиям программы.

Выполнение заданий состояло из теоретической и практической частей, результатом которых являются закрепление и расширение знаний о языке программирования и технологиях JavaScript, фреймворке React. Созданный в результате проделанной работы продукт полностью отвечает всем запланированным функциональным требованиям.

Обучающийся

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_А.В.Шуляк\_\_\_\_\_/

(подпись) (И.О. Фамилия)

Руководитель практики от организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_Е.Р. Михаил\_\_\_\_\_\_\_\_/

(подпись) (И.О. Фамилия)

*М П*

**CПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Node.js [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript/> – Дата доступа: - 02.06.2022.
2. React [Электронный ресурс] – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/JavaScript/> – Дата доступа: - 02.06.2022.
3. JavaScript [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Web/JavaScript/ – Дата доступа: - 02.06.2022.