Учреждение образования

«Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»

**ОТЧЕТ**

по учебной ознакомительной практике

Исполнитель

студент 1 курса 7 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Костюкевич В.П.

(подпись, дата)

Руководитель практики

от кафедры

преподаватель-стажер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Николайчук А.Н.

(подпись, дата)

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Минск 2022 г.

# Введение

Одно из самых популярных направлений в сфере IT — Веб-разработка. Веб — глобальная сеть, и веб разработчики занимаются именно созданием веб-сайтов, веб-приложений, а также различных Интернет-сервисов и технологий. Хоть специалистов в этом направлении очень много, найти работу веб-разработчику сравнительно легко. Веб-разработку можно поделить на две основных составляющие: frontend и backend разработка. Backend-разработчик создаёт серверную часть веб-приложения, frontend-разработчик — клиентскую.

Целью данной учебной практики является получение практического опыта в разработке клиентской части веб-приложения. Итогом работы является проект, соответствующий требованиям современной веб-разработки.

Тематикой проекта была выбрана следующая: “Сайт знакомств “Mimba””. Данная тематика была выбрана исходя из принципов трудоёмкости процесса и распределения работы между разработчиками и по времени.

Процесс выполнения практики проходил совместно со студентом Саввой Боярчуком Ивановичем, таким образом, данный проект является выполненным совместно.

# **1 Используемые инструменты**

В процессе разработки веб-приложения Интернет-знакомств “Mimba” использовались следующие инструменты разработки: Visual Studio Code, Figma, React, Create React App, React Router.

Visual Studio Code — [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0), разработанный [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) для [Windows](https://ru.wikipedia.org/wiki/Windows), [Linux](https://ru.wikipedia.org/wiki/Linux) и [macOS](https://ru.wikipedia.org/wiki/MacOS), представлен на рисунке 1.6. Позиционируется как «лёгкий» редактор кода для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки [веб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)- и [облачных](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BB%D0%B0%D1%87%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) приложений. Включает в себя [отладчик](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BB%D0%B0%D0%B4%D1%87%D0%B8%D0%BA), инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git), [подсветку синтаксиса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D1%81%D0%B2%D0%B5%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81%D0%B0) и средства для [рефакторинга](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3).

Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

React — JavaScript-библиотека для создания пользовательских интерфейсов. Реакт позволяет создавать инкапсулированные компоненты, управляющие своим состоянием и компоновать их. Ещё одна важная особенность библиотеки — декларативность. С помощью React разработчик описывает, как компоненты интерфейса выглядят в разных состояниях. Декларативный подход сокращает код и делает его понятным.

Create React App (CRA) - это интерфейс командной строки (command line interface, CLI), позволяющий создавать предварительно настроенные React-проекты. Предварительная настройка включает в себя, но не исчерпывается, настройкой Webpack и Babel.

React Router – это стандартная библиотека для маршрутизации в React. Он позволяет обрабатывать маршруты в веб-приложении, используя динамическую маршрутизацию. Это позволяет нам перемещаться между различными компонентами в приложении React без обновления страницы при навигации пользователя.

# **2 Реализация проекта**

В первую очередь между мной и моим коллегой был организован план работы над проектом. Согласно с ним, моей частью работы является создание структуры проекта, общего внешнего вида веб-приложения, сетки профилей, а также редактируемого профиля пользователя. Кроме того, я отвечаю за хранение данных.

## **2.1 Структура веб-страниц**

Поскольку для разработки веб-сайта была выбрана библиотека React, структура веб-сайта представляет собой древо компонентов, каждый из которых является функцией. Мной было принято решение о создании четырёх основных компонентов, входящих в главный тег App: Header, Lenta, Content, Footer (Рисунок 1).

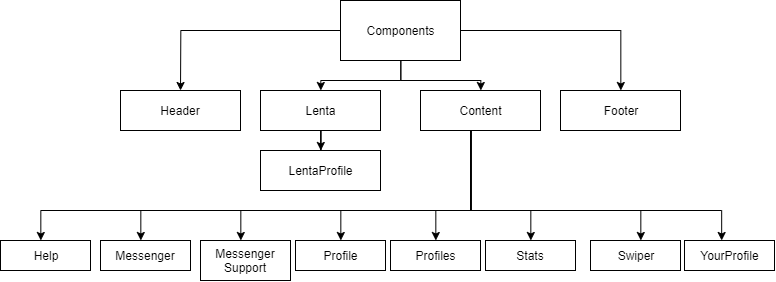


Рисунок 1. Первые 3 уровня структуры приложения “Mimba”

Из схемы рисунка 1 можно заметить, что наибольшее количество подкомпонентов имеет компонент Content. Причиной тому служит большое количество различных адресов, по которым можно перейти, и которые отобразятся внутри компонента Content. Эти переходы реализованы при помощи библиотеки React Router.

Компонент Lenta в данном случае отображает вертикальную линию профилей активных участников сообщества.

При разработке структуры веб-приложения необходимо придерживаться принципов дробления компонентов на минимальные составляющие и создания отдельной директории для каждого из них. В этой директории я помещал JSX-файл, а также CSS-модуль. CSS-модуль отличается от простого файла формата CSS тем, что в случае импорта классов из CSS-модуля в JSX-файл классы передадутся как свойства некоторого объекта. Это позволяет прописывать стили для отдельного компонента, не опасаясь, что они повлияют на отображение других компонентов.

## **2.2 Вёрстка**

Вёрстка веб-страниц и компонентов осуществлена при помощи технологии CSS Grid. Грид — это макет для сайта (его схема, проект). Грид-модель позволяет размещать контент сайта (располагать его определенным образом, позиционировать). Она позволяет создавать структуры, необходимые для обеспечения отзывчивости сайтов на различных устройствах.

Я использовал Grid-вёрстку следующим образом: большинству компонентов задавал свойства grid-area, а для родительских элементов строил схему расположения элементов при помощи grid-template-areas. (Листинг 1)

.app-wrapper

{

  display: grid;

  grid-template-rows:60px 100px 1fr 100px;

  grid-gap:5px;

  grid-template-areas:

  "h"

  "l"

  "c"

  "f";

  position: relative;

}

.content

{

  grid-area:c;

  grid-template-areas:

  "help main"

  "stats main";

  grid-gap: 5px;

}

Листинг 1 – CSS Grid

Как видно из листинга, дочерний элемент с классом content также представлен в виде сетки.

**2.3 React Router**

Библиотека React Router является одной из наиболее полезных и широко используемых в веб-разработке библиотек. Благодаря этой библиотеке можно настроить отрисовку тех или иных компонентов в зависимости от URL-адреса страницы. Для работы с React Router я сделал следующие действия:

1. Установил библиотеку с помощью команды терминала:

npm i react-router-dom -save;

2. Импортировал компоненты BrowserRouter, Routes, Route, NavLink;

3. Обернул содержимое главного компонента <App /> компонентом <BrowserRouter></BrowserRouter>. Это обязательное условие для функционирования библиотеки.

4. В компоненте <Content /> создал компонент <Routes></Routes>

5.Внутри с помощью компонентов <Route path=”” element=””></Route> создал компоненты, которые будут отрисованы только в случае перехода на URL, указанный в атрибуте path. Атрибут element отвечает за компонент, который будет отрисован.

6. С помощью компонента <NavLink to=””></NavLink> создал ссылки на необходимые URL, которые принимает атрибут to.

После проделанных действий в компоненте <Content /> теперь возможно поместить множество различных компонентов, которые будут отрисованы только в том случае, если пользователь перейдёт по соответствующей ссылке <NavLink />.

**2.4 Сетка профилей и хранение данных.**

Сетка профилей — это компонент, отображаемый на странице “Поиск” и состоящий из компонентов, каждый из которых является профилем отдельной Интернет-персоны (Рисунок 2). Изображения людей были сгенерированы на сайте thispersondoesnotexist.com.

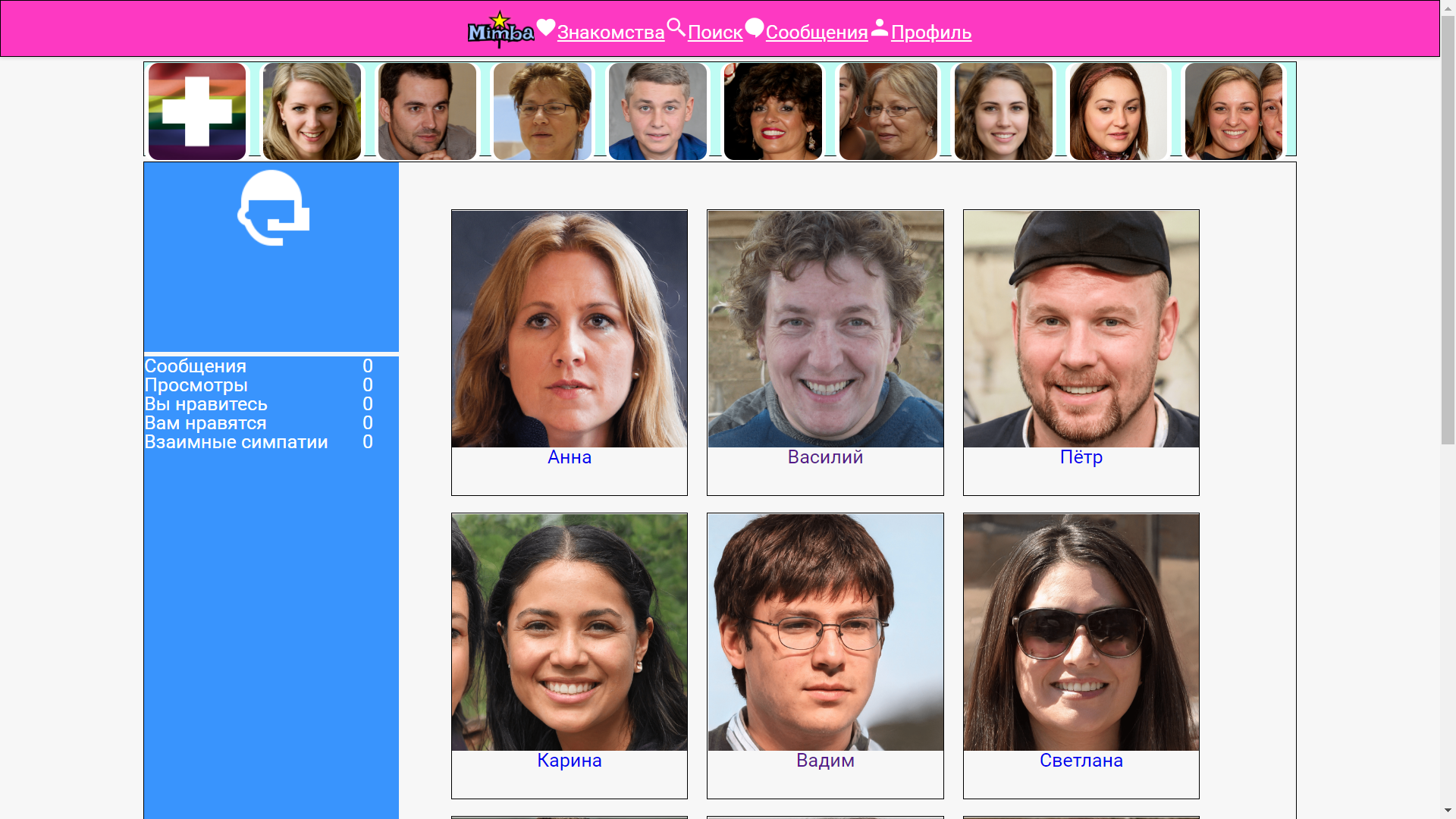


Рисунок 2 – Сетка профилей

Пользователь может перейти по имени-ссылке и попасть на страницу профиля. На этой странице представлена основная информация о субъекте и интересующие его знакомства. Также есть ссылка на страницу переписки с ним.

Всего на странице Profiles представлено 12 профилей. Информацию каждый профиль хранит разную. Данные хранятся в массиве объектов. Один объект представляет собой полную информацию об одном профиле. С помощью метода массива map() осуществлена отрисовка всей сетки. Метод принимает функцию, аргумент которой – элемент обрабатываемого массива. Функция возвращает компонент, свойствами которого являются значения свойств объекта массива. Метод map() обрабатывает все элементы, начиная с первого.

**2.5 Возможность изменения профиля**

При желании пользователь может изменить информацию о своём профиле. Это касается лишь текстовой информации. Для осуществления данной функции, был создан отдельный компонент под названием <Your Profile /> В него вложен компонент <Inf />, содержащий всю текстовую информацию о пользователе, а в <Inf./> вложено 5 компонентов. Вот они слева направо:

<Name /> <Interes /> <Meet /> <About /> <WhoFind />

Все эти компоненты имеют одинаковый принцип работы. Они возвращают текст, хранящийся в переменной состояния inf (Листинг 2).

import React from 'react';

import S from './Name.module.css'

import { useState } from 'react';

const Text = (props) =>

{

const [val,SetVal] = useState("");

const HandleChange = (event) =>

{

SetVal(event.target.value)

}

if(props.show)

{

    props.updateData(val)

    return <textarea value={props.saveInf} onChange={HandleChange}></textarea>

}

else

{

    return <div></div>

}

}

const Butt = (props) =>

{

if(props.show)

{

    return  <button>

    Готово

    </button>

}

else

{

    return <div></div>

}

}

const Name = () =>

{

    const [showVvod, setShowVvod] = useState(false)

    const [inf, setInf] = useState("Ваше имя")

    return(

        <div className={S.name}>

            <div onClick={() =>setShowVvod(true)} className={S.stroka}>{inf}</div>

          <Text show={showVvod} updateData={setInf} saveInf = {inf}/>

          <div onClick={() =>{setShowVvod(false); }}>

   <Butt show={showVvod} />

     </div>

     </div>

    )

}export default Name;

Когда пользователь кликает по тексту, срабатывает функция setShowVvod(), изменяющая состояние ShowVvod на true. Компоненты <Text /> и <Butt />, представляющие собой текстовую область и кнопку, будут отрисованы только тогда, когда состояние ShowVvod = true, поскольку они получают значение этой переменной в качестве свойств, от которых зависит отрисовка.

При изменении информации в текстовом поле будет вызвана функция HandleChange компонента <Text />. Эта функция изменит состояние переменной val на значение, находящееся в данный момент внутри <textarea>. Состояние компонента изменилось, его можно поднять вверх, добавив в свойства компонента <Text /> Функцию, изменяющую состояние родительского компонента (В данном случае <Name />) И мы присваиваем переменной состоояния inf значение переменной состояния val. Поскольку состояние компонента <Name /> inf изменилось, реакт рендерит компонент заново и строка меняется. Нажав на кнопку с подписью “Готово”, пользователь закрывает <Text /> и <Butt />.

# **3 Заключение**

В процессе работы над проектом для учебной практики мной было изучено большое количество материалов, касающихся принципов Frotend-разработки и Project-менеджмента. Кроме того, были получены навыки, необходимые при приёме на работу Frontend-разработчиком. Разработанное веб-приложение по использованным в нём технологиям является современным, структурированным, соответствующим принципам Single Page Application. Приложение предоставляет достаточно возможностей, чтобы выполнять функцию сайта знакомств. При этом веб-приложение может быть доработано и улучшено, могут быть созданы новые дополнительные функции.