



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Кафедра информационно-измерительных систем

ОТЧЕТ ПО УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКЕ

«Веб-приложение для знакомств среди студентов

ФГБОУ ВО «Московский государственный университет геодезии и
картографии» (МИИГАиК)»

Выполнил:

студент группы 2022-ФГиИБ-ИБ-16

(подпись) Никитин Д. С.

Проверил:

преподаватель кафедры информационно-
измерительных систем,

(подпись) Бондарев И. Н.

Москва 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ТЕХНИЧЕСКОЕ ЗАДАНИЕ	5
ПРИМЕР РЕАЛИЗАЦИИ ТЕХНИЧЕСКОГО ЗАДАНИЯ	7
ОТОБРАЖЕНИЕ АРХИТЕКТУРЫ И СЦЕНАРИЕВ ВНУТРЕННЕГО ВЗАИМОДЕЙСТВИЯ В НОТАЦИЯХ	11
1. Диаграмма вариантов использования	11
2. Диаграмма «сущность-связь»	12
3. Диаграмма потоков данных	13
4. Модель бизнес-процессов	14
ТЕСТИРОВАНИЕ ПРОДУКТА	15
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	21

ВВЕДЕНИЕ

В современном мире информационные технологии играют ключевую роль в жизни общества, проникая во все сферы деятельности человека. Одной из таких сфер является образование, где внедрение информационных технологий способствует повышению качества обучения и эффективности образовательного процесса. В рамках учебной ознакомительной практики было разработано веб-приложение для знакомств среди студентов ФГБОУ ВО “Московский государственный университет геодезии и картографии” (МИИГАиК).

Целью данной практики является разработка веб-приложения, которое позволит студентам университета знакомиться друг с другом, находить общие интересы и создавать новые социальные связи. Это приложение призвано стать дополнительным инструментом для социализации студентов, расширения их круга общения и повышения уровня удовлетворенности образовательным процессом.

Для достижения поставленной цели были определены следующие задачи:

1. Изучить теоретические основы разработки веб-приложений.
2. Проанализировать существующие решения в области приложений для знакомств.
3. Разработать концепцию и дизайн веб-приложения.
4. Реализовать функционал приложения, включая регистрацию пользователей, поиск по интересам, обмен контактами и другие функции.
5. Протестировать разработанное приложение на предмет работоспособности и удобства использования.
6. Подготовить отчет о проделанной работе, включающий описание процесса разработки, результаты тестирования и выводы.

В ходе выполнения практики были использованы различные методы исследования, такие как анализ научной литературы, изучение существующих

решений, проектирование и программирование. Результаты работы могут быть полезны для дальнейшего развития информационных технологий в сфере образования и повышения социальной активности студентов.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. Выбор объекта реализации

Проект:

Реактивное веб-приложение для знакомств среди студентов, развития нетворкинга и студенческого сотрудничества

Начальные условия:

- Защита от веб-краулеров/скреперов/парсеров;
- Использование websockets;
- Использование serversentevents;
- Реактивный web-интерфейс;
- Интеграция с провайдером аутентификации;

2. Проектирование, проработка, формальное описание продукта

Назначение разработки:

Создать удобный для студентов сервис, который позволит выстраивать новые социальные связи на основе фильтра по интересам и запросу пользователей

Функциональные требования:

- Создание анкеты с добавлением данных;
- Просмотр анкет других пользователей и корректное отображение введенных данных;
- Обеспечение целостности, доступности и конфиденциальности пользователей;

Требования к внешним (пользовательским и программным) интерфейсам:

- Доступность;
- Кроссплатформенность;

- Адаптивность;
- Кроссбраузерность;
- Интуитивный дизайн;

Требования к производительности:

- Отказоустойчивость;
- Эффективность;

Требования к надёжности:

- Устойчивость к ошибкам;
- Ресурсы системы должны быть доступны, но защищены от несанкционированного доступа.

ПРИМЕР РЕАЛИЗАЦИИ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Защита от веб-краулеров/скреперов/парсеров представлена в виде использования в программном продукте решения от компании Google – reCAPTCHA v2 (рис. 1)

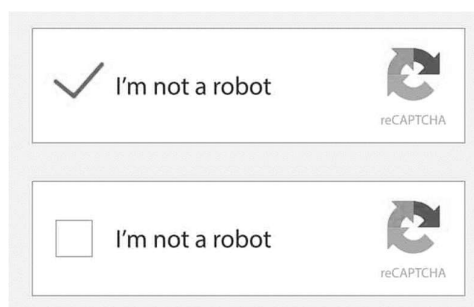


Рис. 1. reCAPTCHA v2

Флажок «Я не робот» требует, чтобы пользователь установил флажок, указывающий, что пользователь не является роботом. Это либо немедленно пропустит пользователя (без CAPTCHA), либо попросит его подтвердить, является ли он человеком (рис. 2). [1]

The image displays a user registration form on a website. At the top right, there is a logo for 'МИИГАиК'. The form itself has two buttons at the top: 'Войти' (Login) and 'Создать аккаунт' (Create account). Below these are several input fields: 'Введите никнейм телеграмма' (Enter Telegram nickname), 'Введите пароль' (Enter password), 'Повторите пароль' (Repeat password), 'Имя' (Name), 'Возраст' (Age), 'Пол' (Gender) with a dropdown arrow, 'Факультет' (Faculty) with a dropdown arrow, 'Увлечения' (Hobbies) with a dropdown arrow, and 'Немного о себе' (A little about you). There is a file upload section with a button 'Выберите файл' (Choose file) and the text 'Файл не выбран' (File not selected). Below this is a 'Зарегистрироваться' (Register) button. A paragraph of text follows, stating that by clicking 'Зарегистрироваться', the user agrees to the processing and transfer of their personal data in accordance with the Federal Law of 27.07.2006 No 152-FZ 'On personal data'. Below this text is a reCAPTCHA v2 checkbox with the text 'I'm not a robot' and the reCAPTCHA logo.

Рис. 2. Страница регистрации пользователя

Реализация в виде кода JavaScript (файл Login.jsx):

```
<ContainerCaptha>
  <Recaptcha
    sitekey="6LeIxAcTAAAAAJcZVRqyHh71UMIEGNQ_MXjiZKhl"
    render="explicit"
    verifyCallback={() => {
      setTimeout(() => {
        setFlagCaptcha(false)
      }, 60000)
      setFlagCaptcha(true)
    }}
    onloadCallback={() => { console.log("loaded") }}
  />
</ContainerCaptha>
```

Использование serversentevents представлено в виде HTTP -запросов (post, get, put). В виде кода JavaScript (файл index.js):

```
app.post("/Auth/Regist", validation, UserController.add);
app.post("/Auth/Login", UserController.login);
```

```
app.put("/Asses", AssesController.like);
app.post("/Asses", AssesController.load)
```

```
app.post("/Like", LikeController.showWhoLikeU)
app.put("/Like", LikeController.removeUserFromListLike)
```


Для улучшения пользовательского опыта и повышения эффективности работы приложения мы используем реактивный веб интерфейс на основе библиотеки React.

Он позволяет приложению быстро реагировать на действия пользователя и изменения в данных, обеспечивая плавное и интерактивное взаимодействие. Реактивный интерфейс способствует улучшению производительности приложения, так как он позволяет оптимизировать использование ресурсов и минимизировать задержки в обработке запросов.

Пример (файл Logo.jsx):

```
import React from "react";
import GlobalStyle from "../styles/Global.css";
import Logo from "../imgs/МИИГАиК_logo.png"
const { ContainerLogo } = GlobalStyle
function LogoComp() {
  return (
    <ContainerLogo>
      <img style={{maxWidth: "100px"}} src={Logo} alt="МИИГАиК" />
    </ContainerLogo>
  )
}
export default LogoComp
```

Реализация интеграции с провайдером аутентификации выполнена на серверной части проекта: введенные данные сверяются с данными в базе данных, при корректно введенных данных (рис. 3) – пользователь получает доступ к своей учетной записи.

Пример (UserController.js):

```
export const login = async (req, res) => {
```

```

try {
  let user = await Users.find({ Login: req.body.login })
  console.log(user[0])
  if (!!user[0]) {
    if ( bcrypt.compare(req.body.pass, user[0].Pass)) {
      res.json(user)
    } else {
      console.log("incorrect")
      res.json("incorrect")
    }
  } else {
    console.log("incorrect")
    res.json("incorrect")
  }
} catch (error) {
  res.json(error)
}
}

```



МИИГАиК

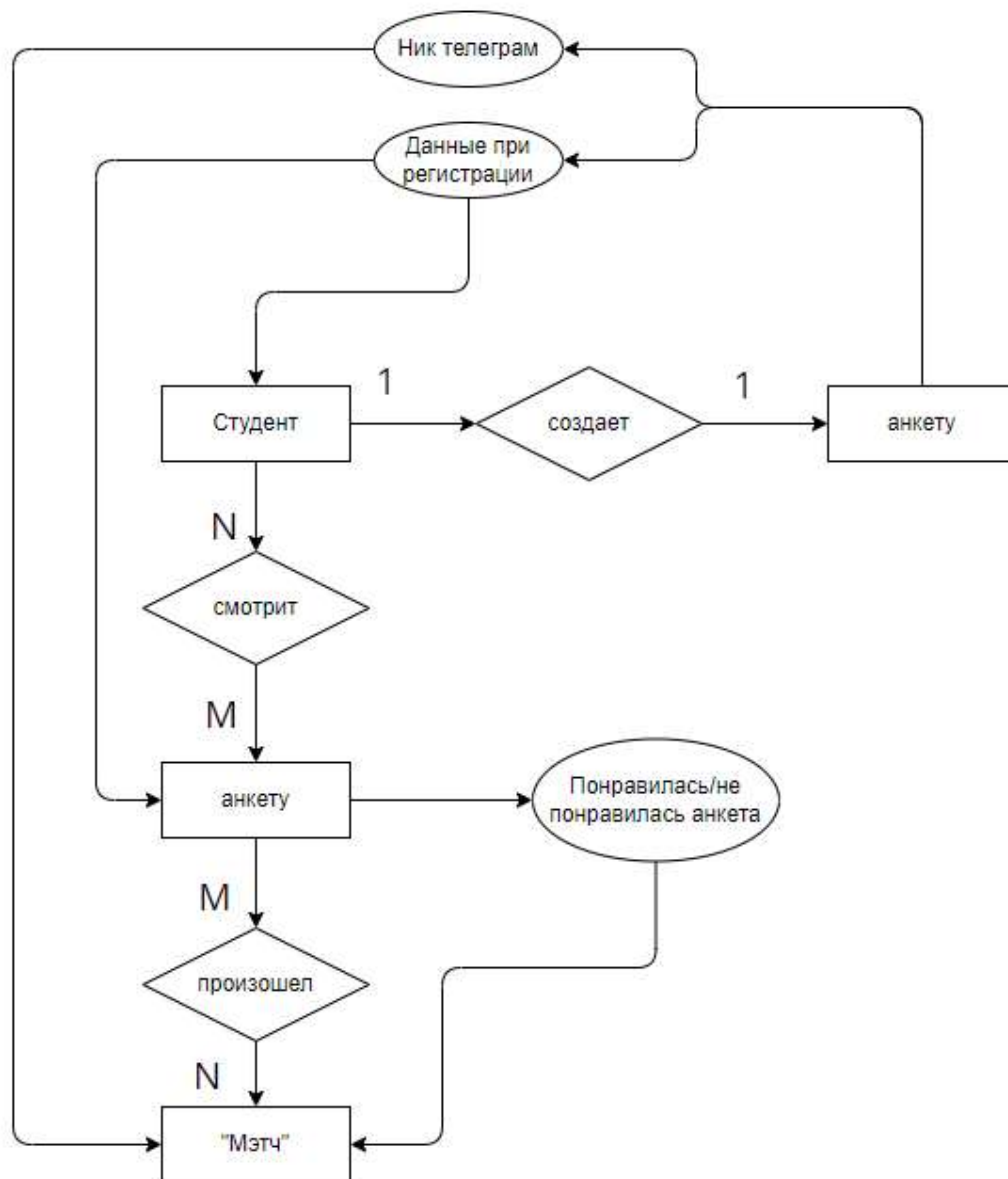
Рис. 3. Страница идентификации пользователя

ОТОБРАЖЕНИЕ АРХИТЕКТУРЫ И СЦЕНАРИЕВ ВНУТРЕННЕГО ВЗАИМОДЕЙСТВИЯ В НОТАЦИЯХ

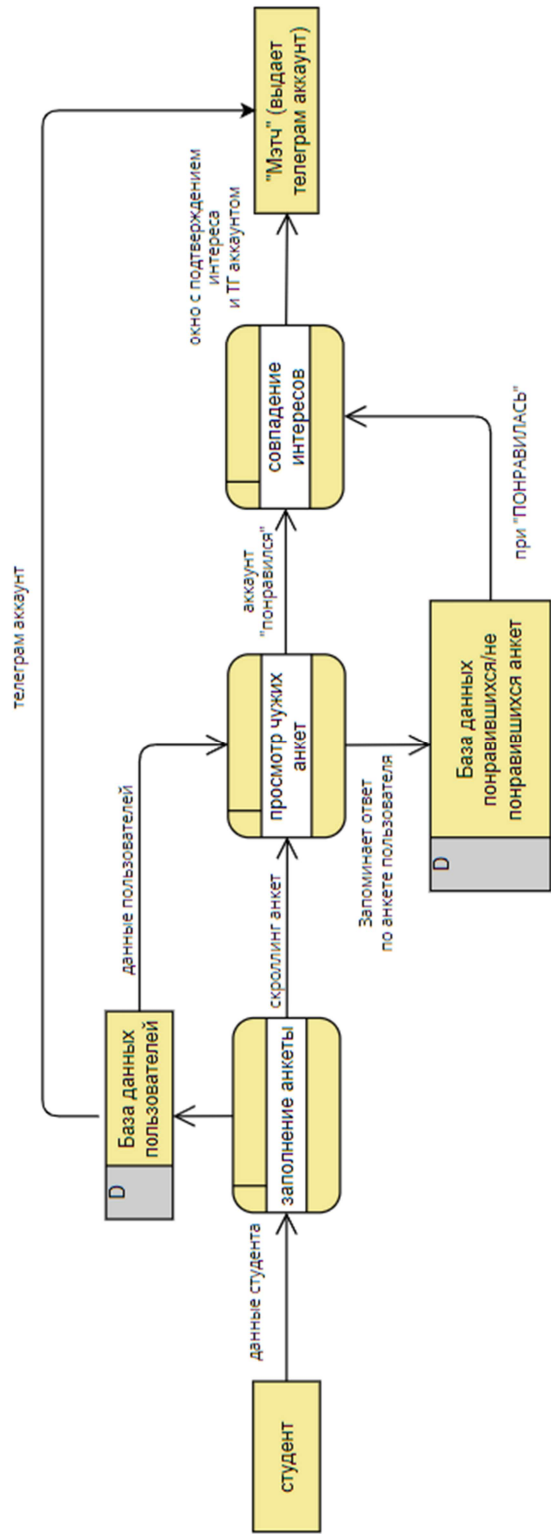
1. Диаграмма вариантов использования



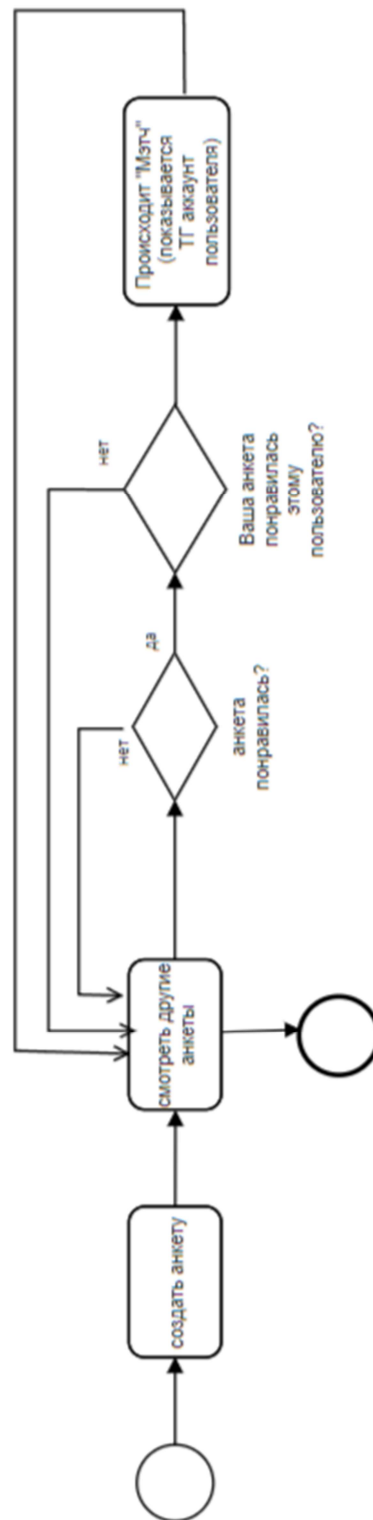
2. Диаграмма «сущность – связь»



3. Диаграмма потоков данных



4. Модель бизнес-процессов



ТЕСТИРОВАНИЕ ПРОДУКТА

1) Регистрация нового пользователя

Шаг 1: Открыть веб-приложение.

- Открыть веб-браузер.
- Ввести URL адрес веб-приложения.
- Нажать Enter.

Шаг 2: Перейти на страницу регистрации.

- Найти и нажать на кнопку «Создать аккаунт».
- Дождаться загрузки страницы (рис. 4).

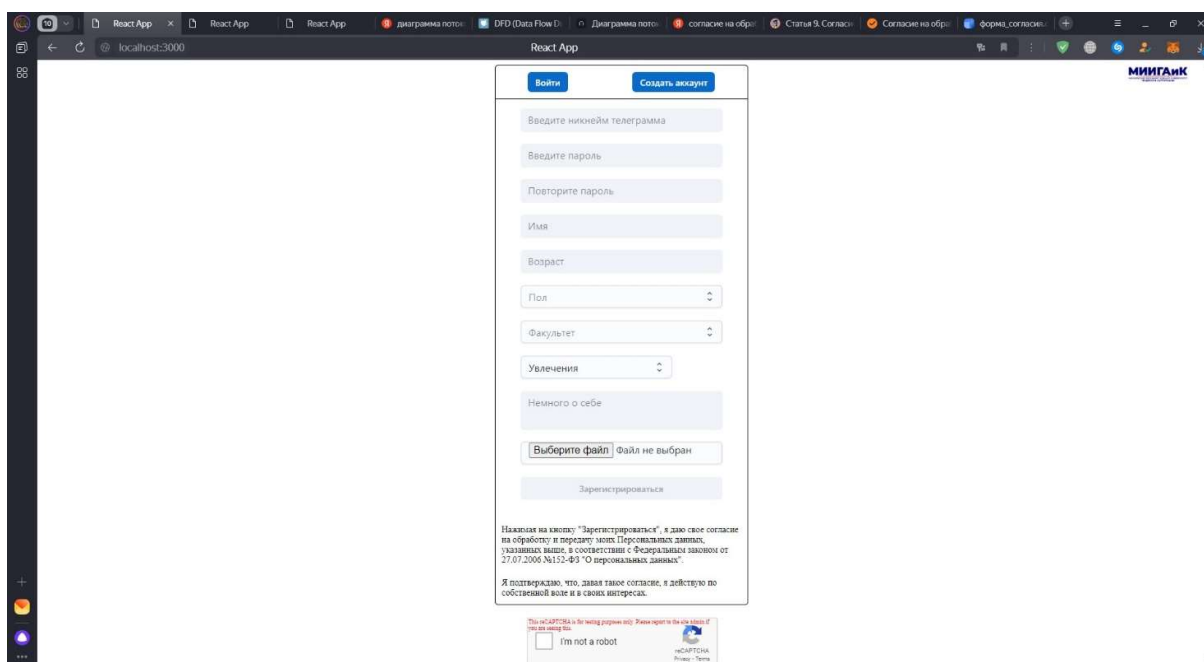


Рисунок 4. Страница «Создать аккаунт»

Шаг 3: Ввести данные для регистрации.

- Ввести уникальное имя пользователя в поле «Введите никнейм телеграмма».
- Ввести пароль в поле «Введите пароль».
- Повторно ввести пароль для подтверждения в поле «Повторите пароль».
- Ввести имя в поле «Имя».
- Ввести возраст в поле «Возраст».

- Выбрать свой пол.
- Выбрать факультет.
- Выбрать увлечения из списка.
- Написать информацию о себе в поле «Немного о себе».
- Загрузить фотографию.
- Подтвердить, что Вы не являетесь роботом.
- Нажать на кнопку «Зарегистрироваться».

Шаг 4: Проверить результат регистрации.

- Дождаться загрузки страницы.
- Проверить, что на странице отображается панель входа.

2) Вход в приложение

Шаг 1: Открыть веб-приложение.

- Открыть веб-браузер.
- Ввести URL адрес веб-приложения.
- Нажать Enter.

Шаг 2: Ввод данных

- Дождаться загрузки страницы (рис. 5).

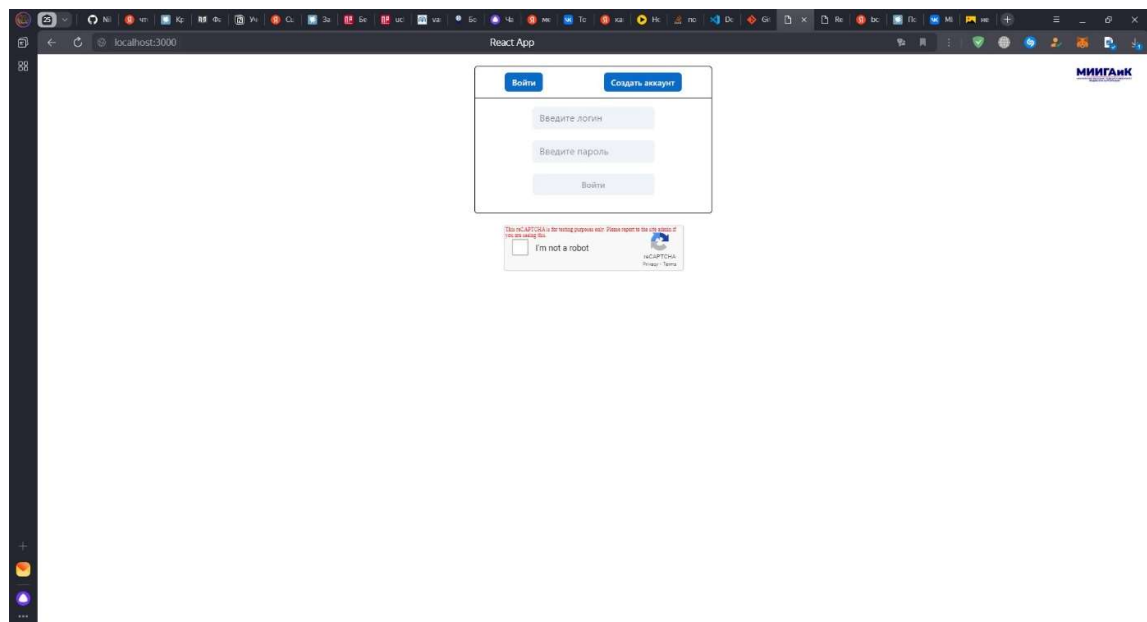


Рисунок 5. Страница «Войти»

- Ввести имя пользователя в поле «Введите логин».
- Ввести пароль в поле «Введите пароль».
- Нажать на кнопку «Войти».

Шаг 3: Проверить результат входа.

- Дождаться загрузки страницы (рис.6).

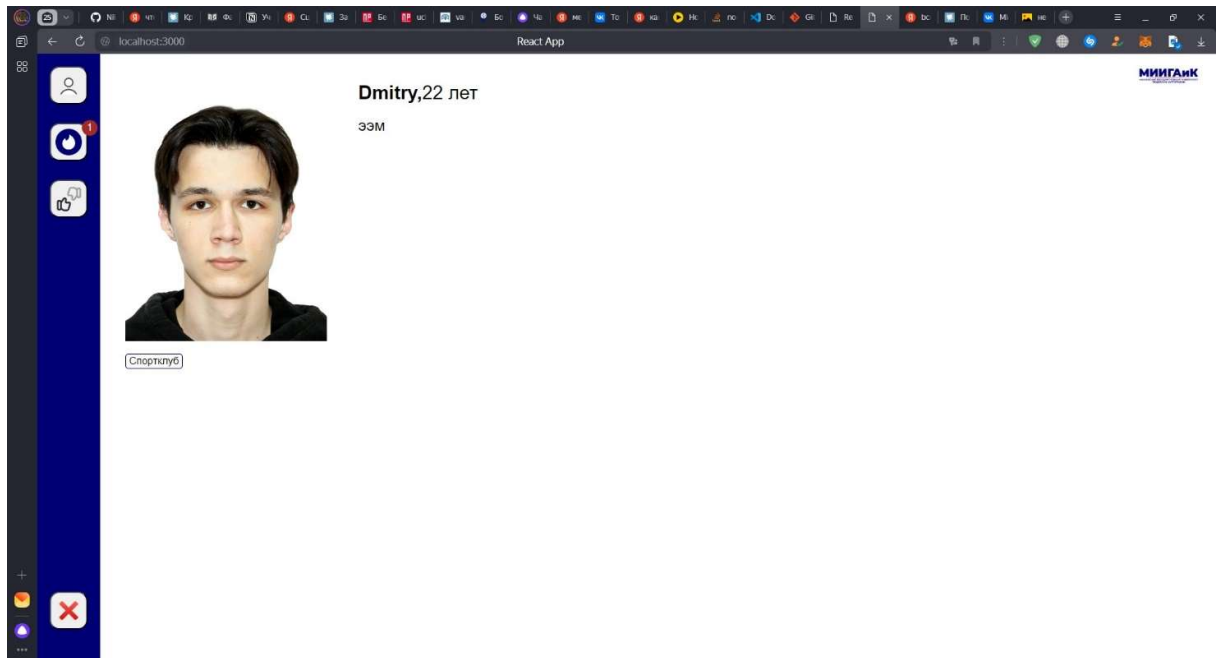


Рисунок 6. Страница «Ваша анкета»

- Проверить что отображается Ваша анкета.
- Проверить, что слева на синем фоне доступны 4 иконки:
 1. Ваша страница;
 2. Анкеты для знакомства;
 3. Анкеты, которым Вы интересны;
 4. Выход из приложения;

3) Просмотр анкет

Шаг 1: Перейти во вкладку «Анкеты»

- Выбрать слева в списке иконку с огонечком.
- Нажать на нее.
- Дождаться загрузки страницы (рис. 7).

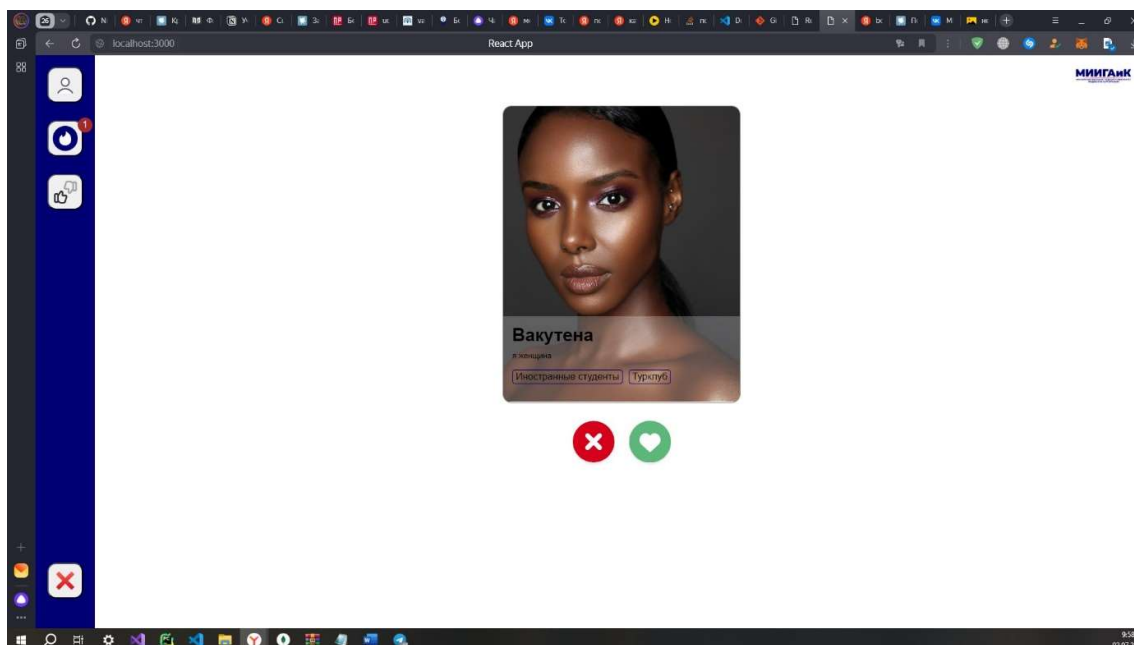


Рисунок 7. Страница «Анкеты»

Шаг 2: Выбор анкет по интересам

- На выбор пользователя:

1. Нажать на красный крестик, если человек не подходит по интересам;
2. Нажать на зеленое сердечко, если подходит;

- При взаимном интересе появится модальное окно с телеграмм-аккаунтом пользователя (рис. 8).

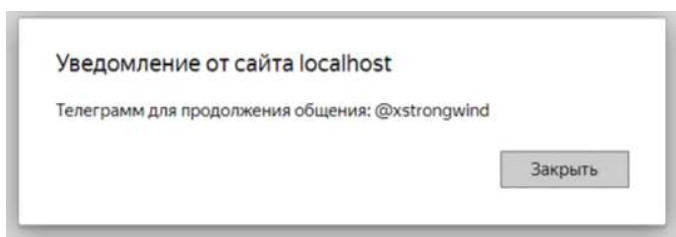


Рисунок 8. Уведомление от сайта

- Если анкеты закончатся, появится предупреждение от сайта (рис. 9).



Рисунок 9. Предупреждение от сайта

ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы по созданию Веб-приложения для знакомств среди студентов ФГБОУ ВО «Московский государственный университет геодезии и картографии» (МИИГАиК) были достигнуты следующие результаты:

- Изучены основные принципы разработки веб-приложений, написания технического задания и проработки сопутствующей документации;
- Исследованы существующие приложения для знакомств и выявлены их основные функции и возможности;
- Разработано приложение для знакомств среди студентов, которое позволяет пользователям создавать профиль, искать потенциальных партнёров на основе отбора по заданным ими критериям, обмениваться контактами и заводить новые знакомства;

В процессе работы над проектом были решены следующие задачи:

- Определены основные функции и возможности приложения;
- Разработан дизайн приложения на основе основных цветов Университета и Факультета Геоинформатики и Информационной безопасности, с использованием логотипов вышеупомянутых структурных организаций;
- Написана программа на выбранном языке программирования и созданы нотации в виде Диаграмм на языке UML;
- Приложение протестировано и отлажено;

Приложение для знакомств среди студентов может стать полезным инструментом для общения и поиска новых друзей и знакомых. Оно может быть дополнено новыми функциями и возможностями в будущем.

Выполнение практической работы позволило получить ценный опыт в разработке веб-приложений и понять, как важно учитывать потребности целевой

аудитории при создании продукта. При создании продукта обращалось внимание на наиболее актуальные похожие приложения в данном сегменте.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- [1] Google for Developers: официальный сайт. -URL: <https://developers.google.com/recaptcha/docs/versions?hl=ru> (дата обращения 06.07.2023)
- Дэвид Флэнаган. JavaScript. Подробное руководство: учеб. Пособие – Москва : 2020. – 1080 с. – ISBN 978-5-93286-215-5.
- Фаулер Мартин. UML. Основы – Москва, Символ-Плюс: 2020. – 192 с. – ISBN 978-5-93286-215-5.