



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и Управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

ДИСЦИПЛИНА: «Кроссплатформенная разработка ПО»

Выполнил: студент гр. ИУК4-62Б _____ (Калашников А. С.)
(Подпись) (Ф.И.О.)

Проверил: _____ (Пчелинцева Н. Н.)
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2023

Цель: Разработка кроссплатформенного мобильного приложения с использованием фреймворка React Native.

Задачи:

1. Изучить возможности фреймворка React Native.
2. Понять принцип создания кроссплатформенных мобильных приложений с использованием React Native.
3. Получить навыки разработки кроссплатформенных мобильных приложений.

Задание: Разработать кроссплатформенное мобильное приложение согласно варианту. Для создания приложения использовать фреймворк React Native. При реализации функционала приложения использовать API сторонних сервисов.

Вариант №6

API: PokeAPI Требования к функционалу:

- 1) Отображение списка покемонов
- 2) Фильтрация списка по типу покемона

Код:

```
import React, {useEffect, useState} from 'react';
import {ActivityIndicator, FlatList, Text, View} from 'react-native';
import { TextInput } from 'react-native-web';

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [data, setData] = useState([]);
  const [count, setCount] = useState([40]);
  const [number, setNumber] = useState([1]);
  const [isLoadingType, setIsLoadingType] = useState(true);
  const [dataType, setDataType] = useState([]);

  const getMovies = async () => {
    try {
      const response = await
fetch('https://pokeapi.co/api/v2/pokemon?offset='+count+'&limit='+count);
      const json = await response.json();
      setData(json.results);
    } catch (error) {
      console.error(error);
    } finally {
      setIsLoading(false);
    }
  };

  const getPoke = async () => {
    try {
      const response = await fetch('https://pokeapi.co/api/v2/type/'+number);
      const json = await response.json();
      setDataType(json.pokemon);
    } catch (error) {
      console.error(error);
    } finally {
      setIsLoadingType(false);
    }
  };

  return (
```

```

<>
<View>

  <button onClick={getMovies()}>Вывод</button>
  <TextInput onChangeText={setCount}/>
  {isLoading ?
    (<ActivityIndicator/>) : (
      <FlatList
        data={data}
        keyExtractor={({id}) => id}
        renderItem={({item}) => (
          <Text>
            {item.name}
          </Text>
        )}
      />
    )}

  <button onClick={getPoke()}>Тип</button>
  <TextInput onChangeText={setNumber}/>
  {isLoadingType ?
    (<ActivityIndicator/>) : (
      <FlatList
        data={dataType}
        keyExtractor={({id}) => id}
        renderItem={({item}) => (
          <Text>
            {item.pokemon.name}
          </Text>
        )}
      />
    )}
</View>
</>
);
};

export default App;

```

Решение:

Вывод	
1	ivysaur
Тип	
3	charizard
	butterfree
	pidgey
	pidgeotto
	pidgeot
	spearow
	fearow
	zubat
	golbat
	farfetchd
	doduo
	dodrio
	scyther
	gyarados
	aerodactyl
	articuno
	zapdos

Вывод	
1	ivysaur
Тип	
4	bulbasaur
	ivysaur
	venusaur
	weedle
	kakuna
	beedrill
	ekans
	arbok

Рис.1 Фильтрация по группам и вывод данных

Вывод: в ходе выполнения лабораторной работы были получены навыки разработки приложения с использованием объектно-реляционного отображения при помощи фреймворка Hibernate на языке Java..