НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

з дисципліни

“Програмування мобільних систем / Розроблення клієнтських додатків для мобільних платформ”

Виконав:

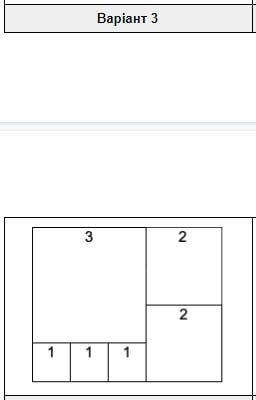
студент групи ІВ-81

ЗК ІВ-8102

Бєлов Микита Сегрійович

Київ 2021

**Варіант № 3**



**Лістинг коду**

**Picture.js**

import React, { useState, useEffect } from 'react';

import { Text, View, ScrollView, Dimensions, Platform, StyleSheet, TouchableOpacity, StatusBar, SafeAreaView } from 'react-native';

import { MaterialCommunityIcons } from '@expo/vector-icons';

import \* as ImagePicker from 'expo-image-picker';

import Layout from "../layout";

import { styleConfig } from "../style";

*const* window = Dimensions.get("window");

*const* screen = Dimensions.get("screen");

*const* maxSizeOfArray = (*arr* = [], *maxArrSize* = 6) *=>* {

*const* lst = [];

    for (*let* i = 0; i < Math.ceil(*arr*.length / *maxArrSize*); i++) {

        lst[i] = *arr*.slice(i \* *maxArrSize*, (i \* *maxArrSize*) + *maxArrSize*);

    }

    return lst;

};

export default *function* Picture({ *navigation* }) {

*const* [dimensions, setDimensions] = useState({ window, screen });

*const* [pictureConstructor, setPictureConstructor] = useState([]);

*const* imageSize = {

        width: dimensions.window.width / 5,

        height: dimensions.window.width / 5,

    }

*const* onChange = ({ *window*, *screen* }) *=>* {

        setDimensions({ window, screen });

    };

    useEffect(() *=>* {

        Dimensions.addEventListener("change", onChange);

        return () *=>* {

            Dimensions.removeEventListener("change", onChange);

        };

    });

    useEffect(() *=>* {

        (async () *=>* {

            if (Platform.OS !== 'web') {

*const* { status } = await ImagePicker.requestMediaLibraryPermissionsAsync();

                if (status !== 'granted') {

                    alert('We need camera roll permissions to make this work!');

                }

            }

        })();

    }, []);

    React.useLayoutEffect(() *=>* {

*navigation*.setOptions({

            headerRight: () *=>* (

                <*TouchableOpacity*

                    activeOpacity={0.5}

                    onPress={pickImage}>

                    <*MaterialCommunityIcons* style={styles.addIcon} name="plus" color={'#808082'} size={30} />

                </*TouchableOpacity*>

            ),

        });

    }, [*navigation*]);

*const* pickImage = async () *=>* {

*let* pickedImage = await ImagePicker.launchImageLibraryAsync({

            mediaTypes: ImagePicker.MediaTypeOptions.All,

            allowsEditing: false,

            quality: 1,

        });

        if (!pickedImage.cancelled) {

            setPictureConstructor(*prevState* *=>* [...*prevState*, { uri: pickedImage.uri }])

        }

    };

*const* PictureComponent = maxSizeOfArray(pictureConstructor).map(

*image* *=>* (

            <*Layout*

                key={*image*[0].uri}

                layout={*image*}

                width={imageSize.width}

                height={imageSize.height}

            />

        )

    );

    return (

        <*SafeAreaView* style={styles.container}>

            {

                pictureConstructor.length === 0 ?

                <*View* style={styles.textContainer}>

                    <*Text* style={styles.text}>No items found</*Text*>

                </*View*> :

                <*ScrollView*>

                        {PictureComponent}

                </*ScrollView*>

            }

        </*SafeAreaView*>

    );

}

*const* styles = StyleSheet.create({

    container: {

        backgroundColor: styleConfig.bg,

        flex: 1,

        borderWidth: 1,

        borderColor: styleConfig.bg,

    },

    textContainer: {

        flex: 1,

        marginTop: '10%'

    },

    text: {

        textAlign: 'center',

        backgroundColor: styleConfig.bg,

        fontSize: 18,

        color: styleConfig.color

    },

    addIcon: {

        textAlign: 'right',

        marginHorizontal: 16,

        marginBottom: 5,

        marginTop: 2,

        color: 'white'

    },

});

**index.js**

import React from 'react';

import { View, StyleSheet } from 'react-native';

import Image from 'react-native-image-progress';

import { styleConfig } from "../style";

*const* Layout = ({ *layout*, *width*, *height* }) *=>* {

*const* imageBoxStyle = (*size* = 1) *=>* {

        if (*size* === 1) {

            return (

                {

                    width: *width*,

                    height: *height*,

                    borderWidth: 1,

                    borderColor: styleConfig.bg,

                }

            )

        } else if (*size* === 2) {

            return (

                {

                    width: *width* \* 2,

                    height: *height* \* 2,

                    borderWidth: 1,

                    borderColor: styleConfig.bg,

                }

            )

        } else if (*size* === 3) {

            return (

                {

                    width: *width* \* 3,

                    height: *height* \* 3,

                    borderWidth: 1,

                    borderColor: styleConfig.bg,

                }

            )

        }

    }

*const* ImageBox = (*uri*, *style* = imageBoxStyle()) *=>* (

        <*View* style={*style*}>

            <*Image*

                style={styles.imageStyle}

                source={*uri*}

            />

        </*View*>

    );

    return (

        <*View*>

            <*View* style={styles.row}>

                <*View* style={styles.column}>

                    {*layout*[0] && ImageBox(*layout*[0], imageBoxStyle(3))}

                    <*View* style={styles.row}>

                        {*layout*[3] && ImageBox(*layout*[3], imageBoxStyle(1))}

                        {*layout*[4] && ImageBox(*layout*[4], imageBoxStyle(1))}

                        {*layout*[5] && ImageBox(*layout*[5], imageBoxStyle(1))}

                    </*View*>

                </*View*>

                <*View* style={styles.column}>

                    {*layout*[1] && ImageBox(*layout*[1], imageBoxStyle(2))}

                    {*layout*[2] && ImageBox(*layout*[2], imageBoxStyle(2))}

                </*View*>

            </*View*>

        </*View*>

    );

};

*const* styles = StyleSheet.create({

    row: {

        flexDirection: "row",

    },

    column: {

        flexDirection: "column",

    },

    imageStyle: {

        height: "100%",

        width: '100%',

        flexDirection: 'row',

        justifyContent: 'center',

        alignItems: 'center',

    }

})

export default Layout

**Висновок**

У даній роботі ми розібралися з тим, як правильно росташовувати картинки на екрані. Навчилися створювати логіку для додавання картинок на екран. Макет відображається правильно, як у лендскейпі так і в портреті. Ми покращили навички оформлення сторінок додатка.