**STUDENT WORK EXPERIENCE PROGRAM (S.W.E.P)**

**NAME: Kayode Oluwasogo Esther**

**MATRIC NO: EES/21/22/0265**

**COURSE CODE: FEG 390**

**PROJECT REPORT**

**TOPIC: CREATE A TODO LIST APP USING EXPO GO**.

**INTRODUCTION**

This app is a simply TODO APP which allows users to generate, modify, and remove tasks, assisting them in maintaining organization and concentration.

To begin, you will be required to configure your development setup. This entails the **[installation of Node.js](https://www.geeksforgeeks.org/installation-of-node-js-on-windows/)**, which serves as the runtime environment for executing JavaScript code beyond the confines of a web browser.

* **Installation:** We will use the Expo CLI edition, which offers a more seamless experience for executing your React Native apps. Proceed sequentially through the provided instructions to establish your React native ecosystem.
* **Expo** is a JavaScript and React-based platform that enables developers to create cross-platform mobile applications for iOS, Android, and the web using a unified codebase. This free and open-source framework offers a range of tools and services that streamline the mobile app development process, empowering developers to construct top-notch applications.

**Prerequisites:**

* [Introduction to React Native](https://www.geeksforgeeks.org/introduction-react-native/)
* [Introduction React Native Components](https://www.geeksforgeeks.org/react-native-view-component/)
* [React Native State](https://www.geeksforgeeks.org/react-native-state/)
* [React Native Props](https://www.geeksforgeeks.org/what-are-props-in-react-native/)
* [React useState Hook](https://www.geeksforgeeks.org/what-is-usestate-in-react/)
* Expo CLI

**Steps to Create React Native Application:**

**Step 1:**Create a react native application by using this command

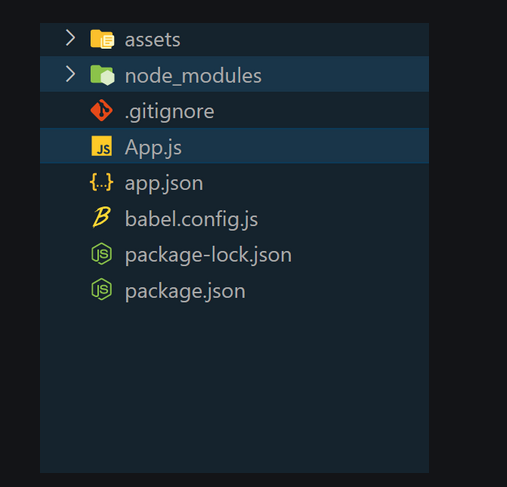
npx create-expo-app todo-app-in-native

**Step 2:**After creating your project folder, i.e. todo-app-in-native, use the

following command to navigate to it:

cd todo-app-in-native

**Project Structure:**



**Example:**

**Step 3:** Open the **App.js**file. Simply paste the source code into the App.js file.

import React, { useState } from "react";

import {

View,

Text,

TextInput,

TouchableOpacity,

FlatList,

StyleSheet,

} from "react-native";

const App = () => {

const [task, setTask] = useState("");

const [tasks, setTasks] = useState([]);

const [editIndex, setEditIndex] = useState(-1);

const handleAddTask = () => {

if (task) {

if (editIndex !== -1) {

// Edit existing task

const updatedTasks = [...tasks];

updatedTasks[editIndex] = task;

setTasks(updatedTasks);

setEditIndex(-1);

} else {

// Add new task

setTasks([...tasks, task]);

}

setTask("");

}

};

const handleEditTask = (index) => {

const taskToEdit = tasks[index];

setTask(taskToEdit);

setEditIndex(index);

};

const handleDeleteTask = (index) => {

const updatedTasks = [...tasks];

updatedTasks.splice(index, 1);

setTasks(updatedTasks);

};

const renderItem = ({ item, index }) => (

<View style={styles.task}>

<Text

style={styles.itemList}>{item}</Text>

<View

style={styles.taskButtons}>

<TouchableOpacity

onPress={() => handleEditTask(index)}>

<Text

style={styles.editButton}>Edit</Text>

</TouchableOpacity>

<TouchableOpacity

onPress={() => handleDeleteTask(index)}>

<Text

style={styles.deleteButton}>Delete</Text>

</TouchableOpacity>

</View>

</View>

);

return (

<View style={styles.container}>

<Text style={styles.heading}>Kayode Oluwasogo Esther</Text>

<Text style={styles.title}>Matric Number: EES/21/22/0265</Text>

<Text style={styles.title}>ToDo App</Text>

<TextInput

style={styles.input}

placeholder="Enter task"

value={task}

onChangeText={(text) => setTask(text)}

/>

<TouchableOpacity

style={styles.addButton}

onPress={handleAddTask}>

<Text style={styles.addButtonText}>

{editIndex !== -1 ? "Update Task" : "Add Task"}

</Text>

</TouchableOpacity>

<FlatList

data={tasks}

renderItem={renderItem}

keyExtractor={(item, index) => index.toString()}

// style={styles.myflatlist}

/>

</View>

);

};

const styles = StyleSheet.create({

container: {

flex: 1,

padding: 40,

marginTop: 40,

backgroundColor: 'purple'

},

title: {

fontSize: 24,

fontWeight: "bold",

marginBottom: 20,

color: 'white'

},

heading: {

fontSize: 30,

fontWeight: "bold",

marginBottom: 7,

color: "white",

},

input: {

borderWidth: 3,

borderColor: "#ccc",

padding: 10,

marginBottom: 10,

borderRadius: 10,

fontSize: 18,

},

addButton: {

backgroundColor: "white",

padding: 10,

borderRadius: 5,

marginBottom: 10,

},

addButtonText: {

color: "black",

fontWeight: "bold",

textAlign: "center",

fontSize: 18,

},

task: {

flexDirection: "row",

justifyContent: "space-between",

alignItems: "center",

marginBottom: 15,

fontSize: 18,

},

itemList: {

fontSize: 19,

color: 'white'

},

taskButtons: {

flexDirection: "row",

},

editButton: {

marginRight: 10,

color: "white",

fontWeight: "bold",

fontSize: 18,

},

deleteButton: {

color: "white",

fontWeight: "bold",

fontSize: 18,

},

// myflatlist: {

// color: "white",

// fontWeight: "bold",

// fontSize: 18,

// },

});

export default App;

**Step 5:** To run the react native application, open the Terminal and enter the command listed below.

npx expo start

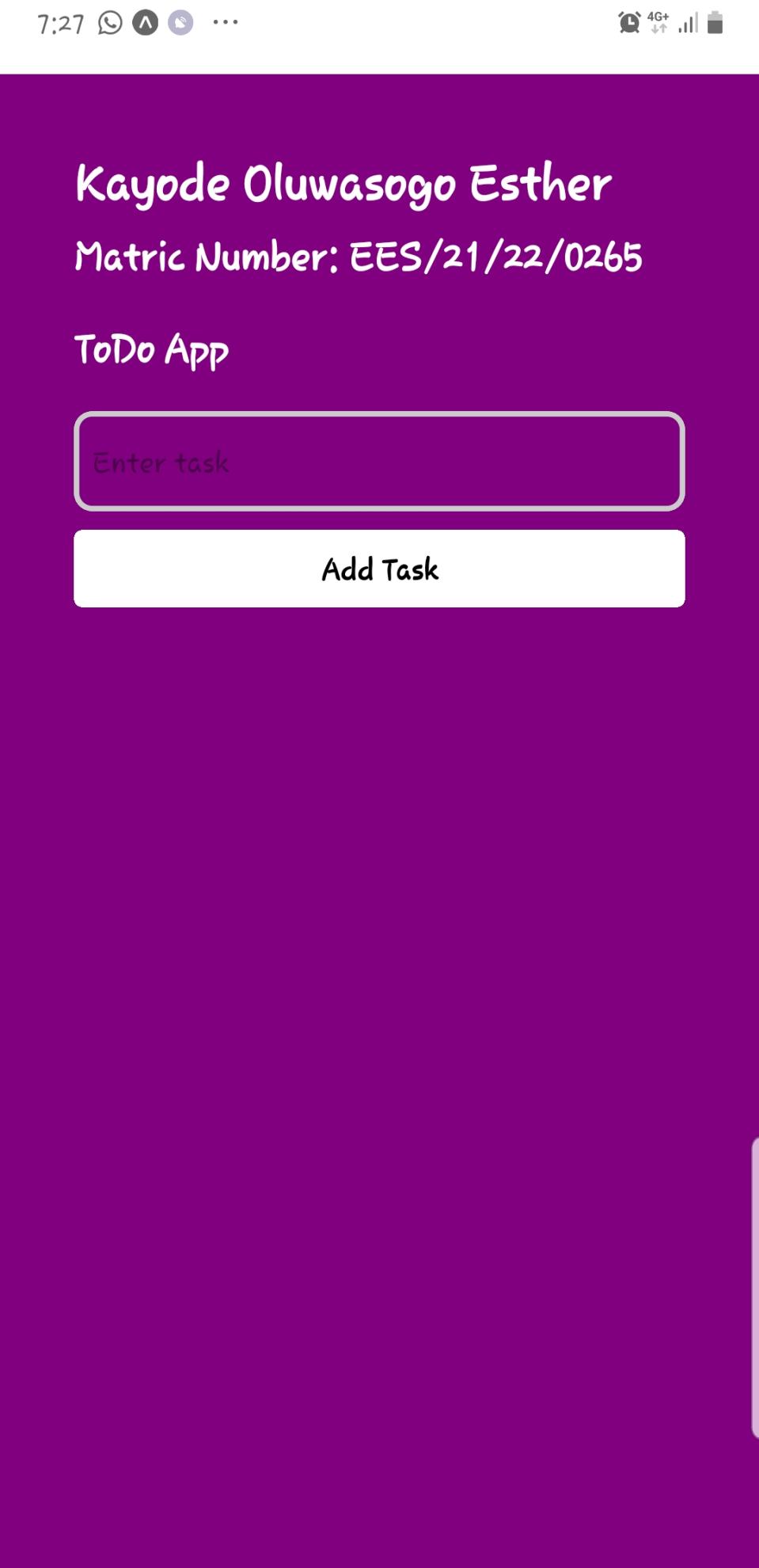
**To run on Android:**

npx react-native run-android

**To run on Ios:**

npx react-native run-ios

**Output:**

****

