2018ICTS13 Tutorial 07

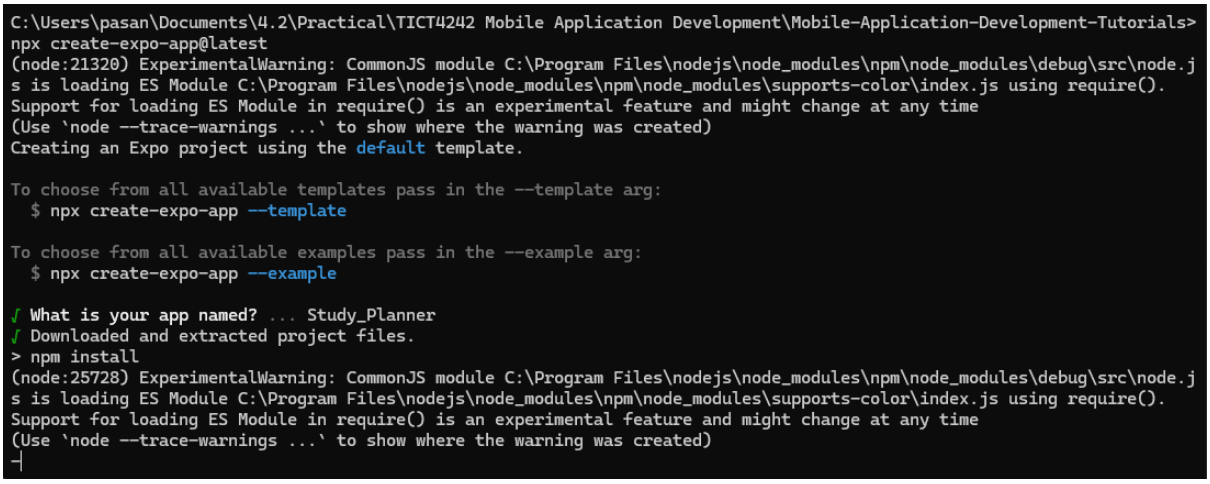
Study Planer Application

Study Planner App using React Native with Expo that helps students organize and manage their study schedules, tasks, and exams.

- **Calendar Screen**
- **Task Management Screen**
- **Settings Screen**
- **Home Screen**
- **Edit Task Screen**
- **Welcome Screen**

Step 1: Project Setup

1. Open a terminal and initialize a new React Native project with Expo:
2. `npx create-expo-app@latest`



- **App.js**

```
3. import React from "react";
4. import { Provider } from "react-native-paper";
5. import { NavigationContainer } from "@react-navigation/native";
6. import { createStackNavigator } from "@react-navigation/stack";
7. import { theme } from "./core/theme";
8. import { AppProvider } from "./context/AppContext";
9. import {
   CalendarScreen,CreateTaskScreen,TaskManagementScreen,WelcomeScreen,Edit
   TaskScreen,SettingsScreen} from "./screens";
```

```
const Stack = createStackNavigator();

export default function App() {
  return (
    <AppProvider>
      <Provider theme={theme}>
        <Stack.Navigator
          initialRouteName="WelcomeScreen"
          options={{ headerShown: false}}

        >
        <Stack.Screen name="CalendarScreen"
   component={CalendarScreen}  options={{ headerShown: false }}/>
        <Stack.Screen name="CreateTaskScreen"
   component={CreateTaskScreen}  options={{ headerShown: false }}/>
        <Stack.Screen name="EditTaskScreen"
   component={EditTaskScreen}  options={{ headerShown: false }}/>
        <Stack.Screen name="WelcomeScreen"
   component={WelcomeScreen}  options={{ headerShown: false }}/>
        <Stack.Screen name="TaskManagementScreen"
   component={TaskManagementScreen}  options={{ headerShown: false }}/>
        <Stack.Screen name="SettingsScreen"
   component={SettingsScreen}  options={{ headerShown: false }}/>
      </Stack.Navigator>
      </Provider>
    </AppProvider>
  );
}
```

- **Screens**

```
export { default as CalendarScreen} from './CalendarScreen'
export { default as TaskManagementScreen} from './TaskManagementScreen'
export { default as SettingsScreen} from './SettingsScreen'
export { default as CreateTaskScreen} from './CreateTaskScreen'
export { default as EditTaskScreen} from './EditTaskScreen'
export { default as WelcomeScreen} from './WelcomeScreen'
```

- **contexts**

```
import React, { createContext, useState, useContext, useEffect } from "react";
import AsyncStorage from '@react-native-async-storage/async-storage';

const AppContext = createContext();

export const AppProvider = ({ children }) => {
  const [data, setData] = useState([
    // Sample data structure
    { id: 1, type: 'class', date: '2024-12-13', title: 'Math Lecture' },
    { id: 2, type: 'exam', date: '2024-12-15', title: 'Physics Exam' },
    { id: 3, type: 'task', date: '2024-12-13', title: 'Complete Homework' },
  ]);

  // Add new item to data
  const addData = (newItem) => {
    setData((prevData) => [...prevData, { id: Date.now(), ...newItem }]);
  };

  // Edit an existing item
  const editData = (id, updatedItem) => {
    setData((prevData) =>
      prevData.map((item) => (item.id === id ? { ...item, ...updatedItem } : item))
    );
  };

  // Delete an item
  const deleteData = (id) => {
    setData((prevData) => prevData.filter((item) => item.id !== id));
  };

  const [reminderTime, setReminderTime] = useState('09:00 AM');
  const loadReminderTime = async () => {
    try {
      const savedTime = await AsyncStorage.getItem('reminderTime');
      if (savedTime) setReminderTime(savedTime);
    } catch (error) {
      console.error('Error loading reminder time:', error);
    }
  };

  const saveReminderTime = async (time) => {
    try {
      await AsyncStorage.setItem('reminderTime', time);
```

```
      setReminderTime(time);
    } catch (error) {
      console.error('Error saving reminder time:', error);
    }
  };

  useEffect(() => {
    loadReminderTime();
  }, []);

  return (
  <AppContext.Provider value={{ data, addData, editData, deleteData,
   reminderTime, saveReminderTime }}>
      {children}
    </AppContext.Provider>
  );
}

export const useAppContext = () => useContext(AppContext);
```

Building the App

Calendar Screen

```
import React, { useState } from "react";
import { View, Text, TextInput, Button, StyleSheet,TouchableOpacity } from
"react-native";
import { useAppContext } from "../context/AppContext";

export default function CreateTaskScreen({ route, navigation }) {
  const { addData } = useAppContext();
  const selectedDate = route.params?.selectedDate; // Get selected date from
navigation params
  const [taskTitle, setTaskTitle] = useState("");
  const handleCreateTask = () => {
    if (taskTitle) {
      addData({ type: "task", date: selectedDate, title: taskTitle });
      navigation.navigate('CalendarScreen')
    }
  };
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Create Task</Text>
      <Text style={styles.label}>Date:<Text
style={styles.date}>  {selectedDate}</Text></Text>
      <TextInput
```

```
        style={styles.input}
        placeholder="Enter Task Title"
        value={taskTitle}
        onChangeText={setTaskTitle}
      />
      <View style={styles.btnwrap}>
                <TouchableOpacity
                style={styles.customButton}
                onPress={handleCreateTask}
                >
                <Text style={styles.buttonText}>Create Task</Text>
            </TouchableOpacity>
      </View>
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    justifyContent: "center",
  },
  title: {
    fontSize: 20,
    fontWeight: "bold",
    marginBottom: 10,
  },
  date: {
    color: "#9A82F6",
    fontSize: 16,
    fontWeight: "bold",
  },
  label: {
    fontSize: 16,
    marginBottom: 10,
  },
  input: {
    borderWidth: 1,
    padding: 10,
    marginBottom: 20,
    borderRadius: 5,
  },
  btnwrap: {marginBottom: 10,marginTop: 10,},
  buttonText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold",
  },
```

```
  customButton: {
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
});
```

Task Management Screen

```
import React, {useState } from 'react';
import { View, Text, FlatList, TextInput, Button, StyleSheet,TouchableOpacity
} from 'react-native';
import { useAppContext } from '../context/AppContext';

export default function TaskManagementScreen({navigation}) {
  const {data, addData, editData, deleteData } = useAppContext();
  const [newTask, setNewTask] = useState('');
  const [selectedDate, setSelectedDate] = useState('');
  const addTask = () => {
    if (newTask && selectedDate) {
      addData({ type: 'task', date: selectedDate, title: newTask });
      setNewTask('');
    }
  };
  return (
    <View style={styles.container}>
      <View style={styles.headerContainer}>
          <TouchableOpacity
            style={styles.BackButton}
            onPress={() => navigation.navigate('CalendarScreen')}
          >
            <Text style={styles.buttonText}>Back</Text>
          </TouchableOpacity>

          <Text style={styles.title}>Task Management</Text>
      </View>
      <TextInput
        style={styles.input}
        placeholder="Enter Task"
        value={newTask}
        onChangeText={setNewTask}
      />
      <TextInput
        style={styles.input}
        placeholder="Enter Date (YYYY-MM-DD)"
```

```
        value={selectedDate}
        onChangeText={setSelectedDate}
      />

      <View style={styles.btnwrap}>
                <TouchableOpacity
                  style={styles.CommonButton}
                  onPress={addTask}
                  >
                <Text style={styles.buttonText}>Add Task</Text>
                </TouchableOpacity>
     </View>

      <FlatList
        style={styles.flatlist}
        data={data.filter((item) => item.type === 'task')}
        renderItem={({ item }) => (
          <View style={styles.taskContainer}>
            <Text><Text style={styles.date}>{item.date}</Text>:
{item.title}</Text>

            <View style={styles.btnwrap}>
                <TouchableOpacity
                  style={styles.EditButton}
                  onPress={() => navigation.navigate('EditTaskScreen', { task:
item })}
                  >
                <Text style={styles.buttonText}>Edit</Text>
                </TouchableOpacity>
            </View>

            <View style={styles.btnwrap}>
                <TouchableOpacity
                  style={styles.DeleteButton}
                  onPress={() => deleteData(item.id)}
                  >
                <Text style={styles.buttonText}>Delete</Text>
                </TouchableOpacity>
            </View>
          </View>
        )}
        keyExtractor={(item) => item.id.toString()}
      />
    </View>
  );
};
const styles = StyleSheet.create({
  container: {
```

```
  flex: 1,
  padding: 20,
},
title: {
  fontSize: 20,
  fontWeight: 'bold',
  marginBottom: 10,
},
input: {
  borderWidth: 1,
  padding: 10,
  marginBottom: 10,
  borderRadius: 30
},
taskContainer: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  alignItems: 'center',
  marginVertical: 5,
},
date: {
  color: "#9A82F6",
  fontSize: 15,
  fontWeight: "bold",
},
buttonText: {
  color: "white",
  fontSize: 16,
  fontWeight: "bold",
},
EditButton: {
  backgroundColor: "#9A82F6",
  paddingVertical: 5,
  paddingHorizontal: 10,
  borderRadius: 30,
  alignItems: "center",
},
DeleteButton: {
  backgroundColor: "red",
  paddingVertical: 5,
  paddingHorizontal: 10,
  borderRadius: 30,
  alignItems: "center",
},
flatlist: {
  paddingVertical: 10
},
CommonButton: {
```

```
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
  headerContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between',
    paddingVertical: 10,
  },
  BackButton: {
    backgroundColor: '#9A82F6',
    paddingVertical: 8,
    paddingHorizontal: 12,
    borderRadius: 5,
  },
});
```

Settings Screen

```
import React, { useState, useEffect } from 'react';
import { View, Text, TextInput, TouchableOpacity, StyleSheet } from 'react-
native';
import { useAppContext } from '../context/AppContext';

export default function SettingsScreen({navigation}) {
  const { reminderTime, saveReminderTime } = useAppContext();
  const [localReminderTime, setLocalReminderTime] = useState(reminderTime);
  // Update local state when context changes
  useEffect(() => {
    setLocalReminderTime(reminderTime);
  }, [reminderTime]);

  const handleSave = () => {
    saveReminderTime(localReminderTime);
    alert(`Reminder time saved: ${localReminderTime}`);
    navigation.navigate('CalendarScreen')
  };
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Settings</Text>
      <TextInput
        style={styles.input}
        placeholder="Set Reminder Time (e.g., 09:00 AM)"
        value={localReminderTime}
```

```
        onChangeText={setLocalReminderTime}
      />
      <View style={styles.btnwrap}>
                <TouchableOpacity
                style={styles.customButton}
                onPress={handleSave}
                >
                <Text style={styles.buttonText}>Save Reminder Time</Text>
            </TouchableOpacity>
      </View>
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    justifyContent: 'center',
  },
  title: {
    fontSize: 20,
    fontWeight: 'bold',
    marginBottom: 20,
  },
  input: {
    borderWidth: 1,
    padding: 10,
    marginBottom: 20,
    borderRadius: 30,
  },
  btnwrap: {marginBottom: 10,marginTop: 10,},
  buttonText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold",
  },
  customButton: {
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
});
```

Home Screen

```jsx
import React from 'react';
import { View, Text, Button, StyleSheet,TouchableOpacity } from 'react-
native';
import Logo from '../components/logo'
export default function WelcomeScreen({ navigation }) {
  return (
    <View style={styles.container}>
        <Logo/>
      {/* Welcome Title */}
      <Text style={styles.title}>Welcome to Task Manager</Text>
      <Text style={styles.subtitle}>
        Organize your tasks effectively and stay productive!
      </Text>
        <View style={styles.btnwrap}>
            <TouchableOpacity
            style={styles.customButton}
            onPress={() => navigation.navigate("CalendarScreen")}
            >
            <Text style={styles.buttonText}>Go to Calendar</Text>
            </TouchableOpacity>
        </View>
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f5f5f5',
    padding: 20,
  },
  logo: {
    width: 150,
    height: 150,
    marginBottom: 20,
  },
  title: {
    fontSize: 24,
    fontWeight: 'bold',
    marginBottom: 10,
    color: '#000000',
  },
  subtitle: {
    fontSize: 16,
    color: '#666',
    textAlign: 'center',
```

```
    marginBottom: 20,
  },
  btnwrap: {marginBottom: 10,marginTop: 10,},
  buttonText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold",
  },
  customButton: {
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
});
```

Create Task Screen

```
import React, { useState } from "react";
import { View, Text, TextInput, Button, StyleSheet,TouchableOpacity } from
"react-native";
import { useAppContext } from "../context/AppContext";

export default function CreateTaskScreen({ route, navigation }) {
  const { addData } = useAppContext();
  const selectedDate = route.params?.selectedDate; // Get selected date from
navigation params
  const [taskTitle, setTaskTitle] = useState("");
  const handleCreateTask = () => {
    if (taskTitle) {
      addData({ type: "task", date: selectedDate, title: taskTitle });
      navigation.navigate('CalendarScreen')
    }
  };
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Create Task</Text>
      <Text style={styles.label}>Date:<Text
style={styles.date}>  {selectedDate}</Text></Text>
      <TextInput
        style={styles.input}
        placeholder="Enter Task Title"
        value={taskTitle}
        onChangeText={setTaskTitle}
      />
      <View style={styles.btnwrap}>
                <TouchableOpacity
```

```
                style={styles.customButton}
                onPress={handleCreateTask}
                >
                <Text style={styles.buttonText}>Create Task</Text>
            </TouchableOpacity>
      </View>
    </View>
  );
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    justifyContent: "center",
  },
  title: {
    fontSize: 20,
    fontWeight: "bold",
    marginBottom: 10,
  },
  date: {
    color: "#9A82F6",
    fontSize: 16,
    fontWeight: "bold",
  },
  label: {
    fontSize: 16,
    marginBottom: 10,
  },
  input: {
    borderWidth: 1,
    padding: 10,
    marginBottom: 20,
    borderRadius: 5,
  },
  btnwrap: {marginBottom: 10,marginTop: 10,},
  buttonText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold",
  },
  customButton: {
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
```

```
});
```

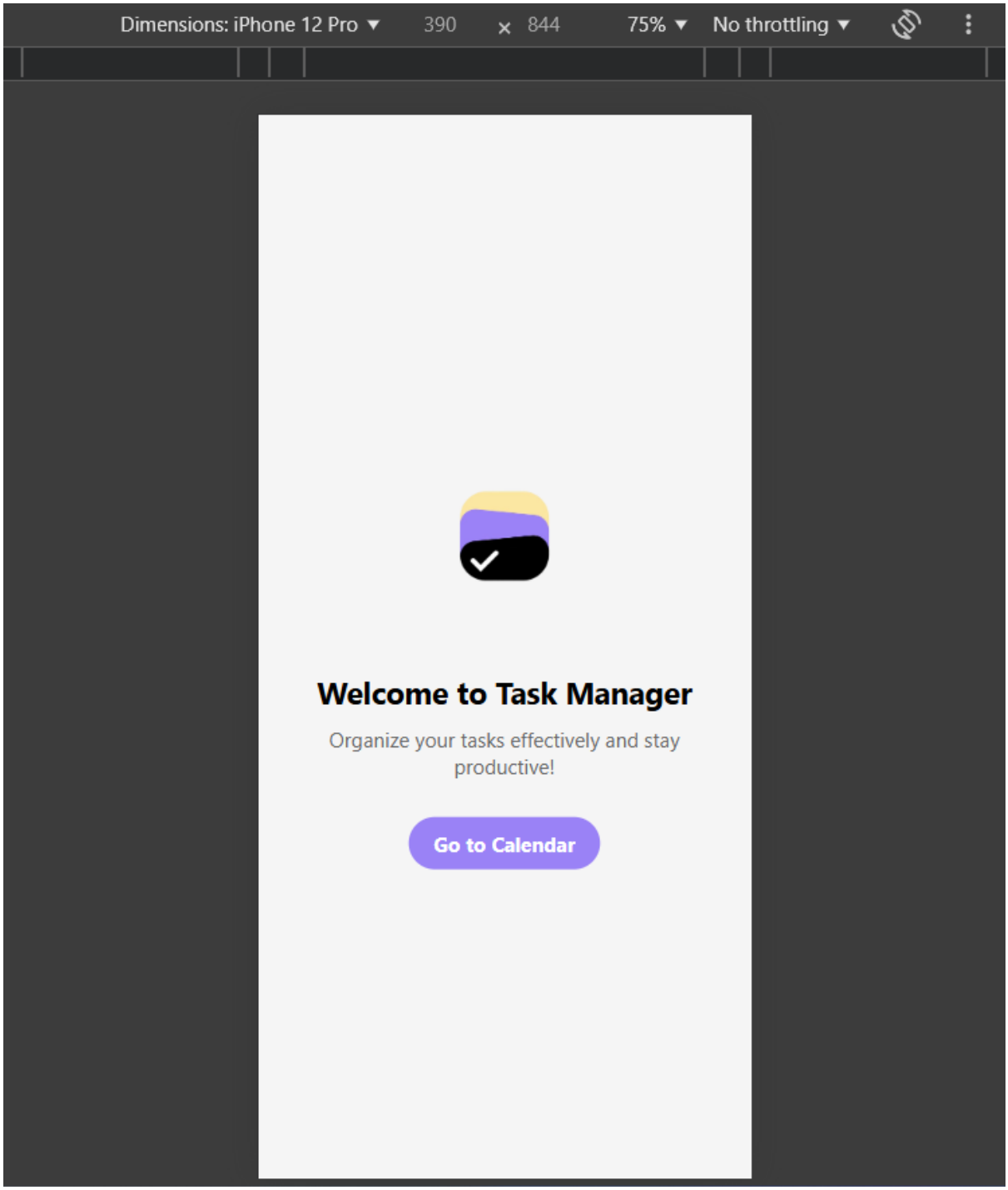
Calendar Screen

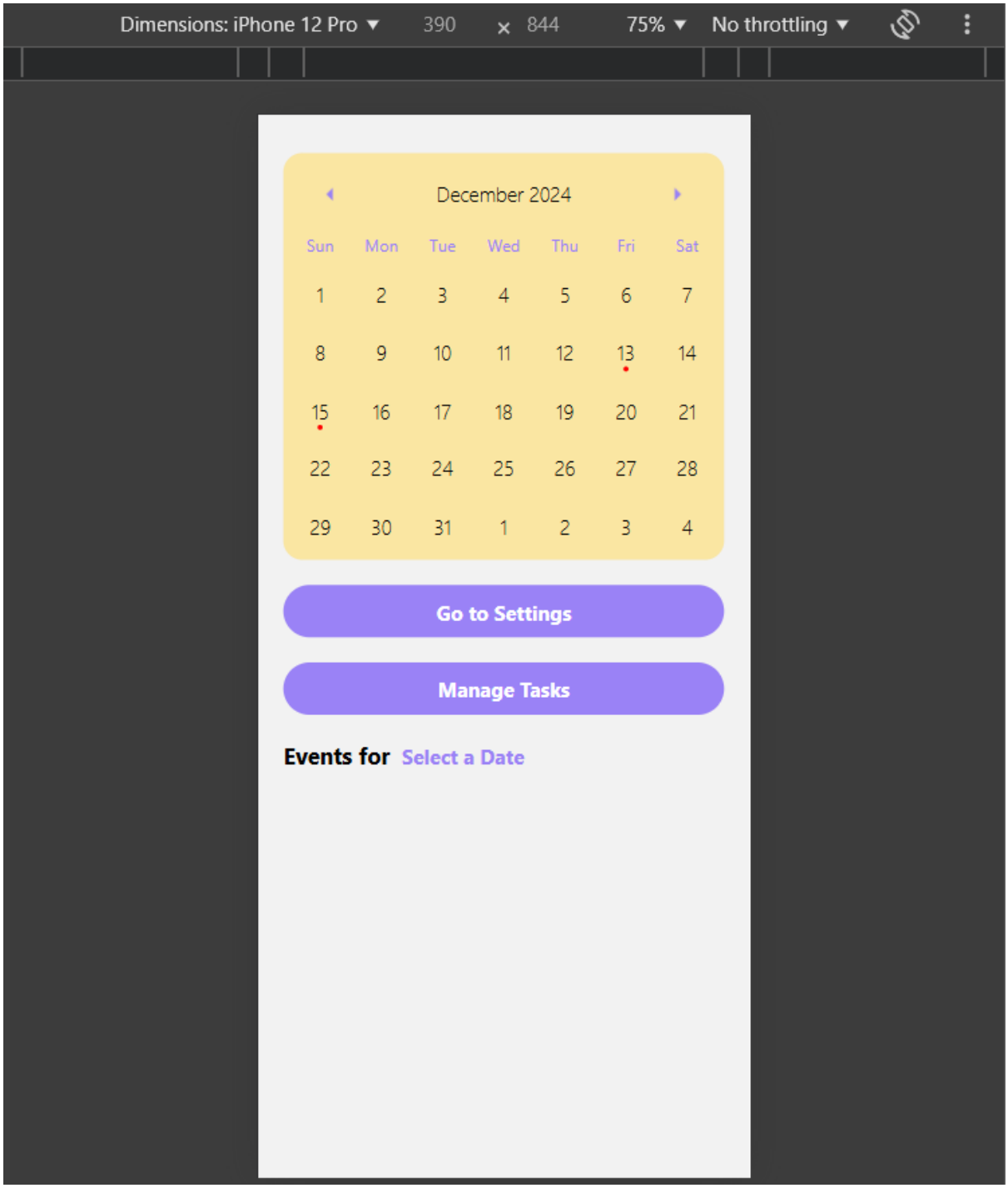
```
import React, {useState } from 'react';
import { View, Text, FlatList, Button, StyleSheet,TouchableOpacity } from
'react-native';
import { Calendar } from 'react-native-calendars';
import { useAppContext } from '../context/AppContext';

export default function CalendarScreen({navigation}) {
  const { data} = useAppContext();
  const [selectedDate, setSelectedDate] = useState('');
  // Filter data for the selected date
  const eventsForSelectedDate = data.filter((item) => item.date ===
selectedDate);
  return (
    <View style={styles.container}>
      <Calendar
        onDayPress={(day) => setSelectedDate(day.dateString)}
        markedDates={{
          [selectedDate]: { selected: true, selectedColor: 'blue' },
          ...data.reduce((acc, item) => {
            acc[item.date] = { marked: true, dotColor: 'red' };
            return acc;
          }, {}),
        }}
        theme={{
          backgroundColor: '#F9F4DC', // Overall calendar background
          calendarBackground: '#FAE6A2',
          textSectionTitleColor: '#9A82F6', // Month header text
          selectedDayBackgroundColor: '#000000', // Selected day's background
          selectedDayTextColor: 'white', // Selected day text
          todayTextColor: '#000000', // Today's text color
          dayTextColor: '#000000', // Default day text color
          textDisabledColor: '#000000', // Disabled days (e.g., past days)
text
          dotColor: '#9A82F6', // Dots for marked dates
          selectedDotColor: '#000000', // Dot color on selected day
          arrowColor: '#9A82F6', // Calendar navigation arrow color
          monthTextColor: '#000000', // Month name text color

        }}
        style={{
          borderRadius: 15,
          marginVertical: 10,
          padding: 5,
          elevation: 3, // Shadow effect
        }}
```

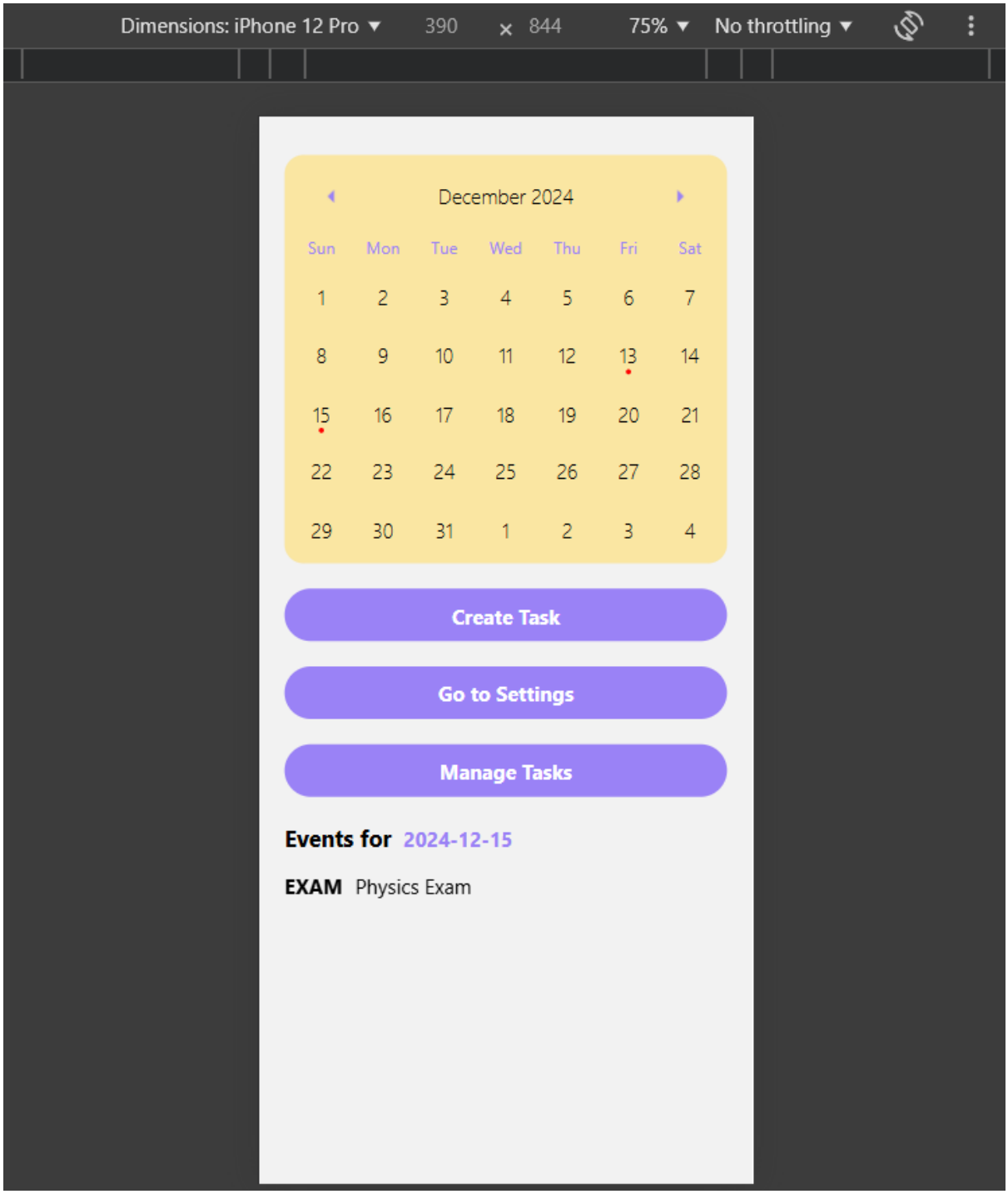
```
      />
      {selectedDate && (
        <View style={styles.btnwrap}>
            <TouchableOpacity
            style={styles.customButton}
            onPress={() => navigation.navigate("CreateTaskScreen", {
selectedDate })}
            >
            <Text style={styles.buttonText}>Create Task</Text>
            </TouchableOpacity>
        </View>
      )}
      <View style={styles.btnwrap}>
            <TouchableOpacity
            style={styles.customButton}
            onPress={() => navigation.navigate('SettingsScreen')}
            >
            <Text style={styles.buttonText}>Go to Settings</Text>
            </TouchableOpacity>
      </View>

      <View style={styles.btnwrap}>
            <TouchableOpacity
            style={styles.customButton}
            onPress={() => navigation.navigate('TaskManagementScreen')}
            >
            <Text style={styles.buttonText}>Manage Tasks</Text>
            </TouchableOpacity>
      </View>

      <Text style={styles.dateTitle}>
        Events for <Text style={styles.date}> {selectedDate || 'Select a
Date'}</Text>
      </Text>
      <FlatList
        data={eventsForSelectedDate}
        renderItem={({ item }) => (
          <View style={styles.eventContainer}>
            <Text style={styles.eventType}>{item.type}</Text>
            <Text style={styles.eventTitle}>{item.title}</Text>
          </View>
        )}
        keyExtractor={(item) => item.id.toString()}
      />
    </View>
  );
};
const styles = StyleSheet.create({
```

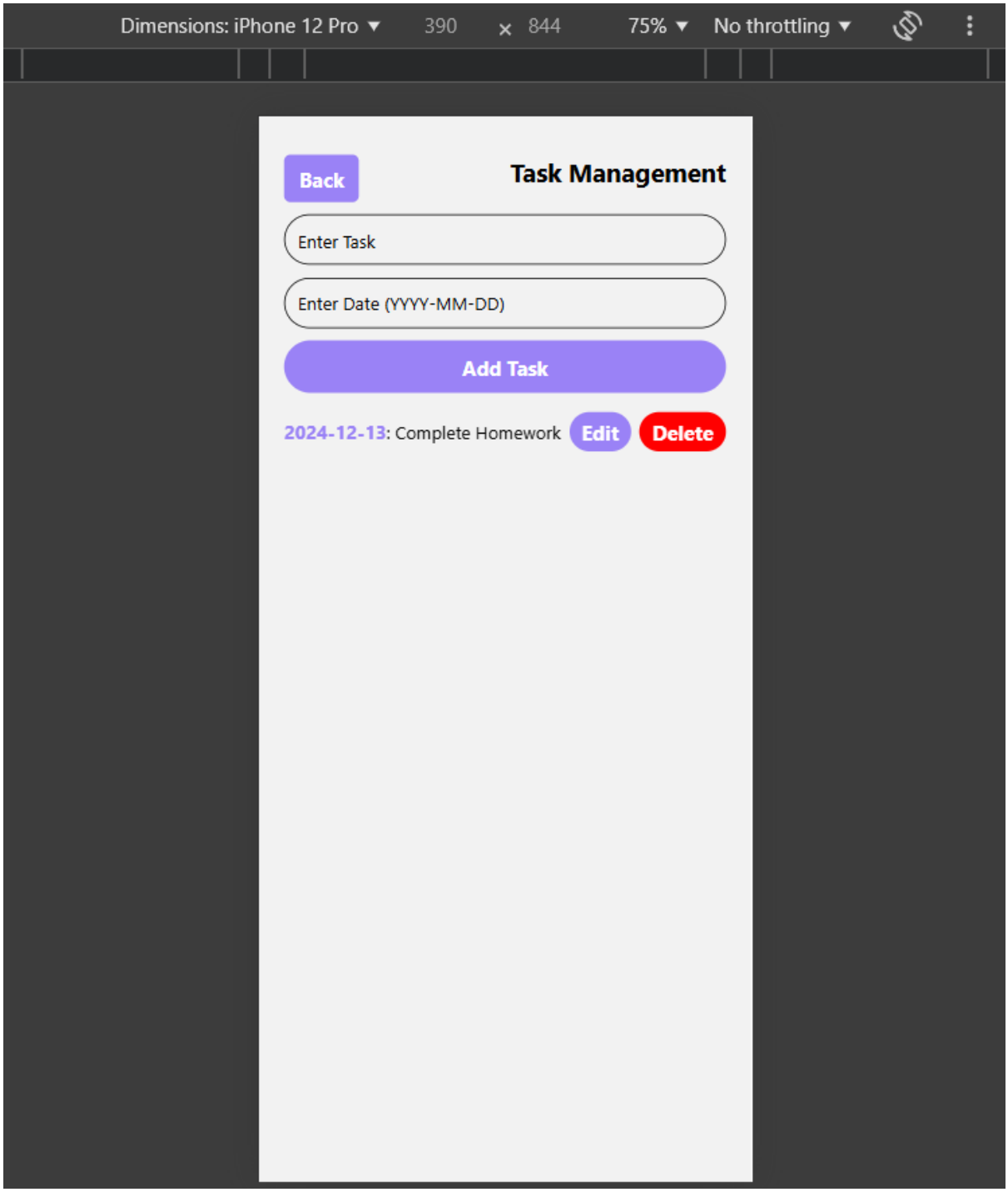
```
  container: {
    flex: 1,
    padding: 20,
  },
  dateTitle: {
    fontSize: 18,
    fontWeight: 'bold',
    marginVertical: 10,
  },
  eventContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    marginVertical: 5,
  },
  eventType: {
    fontWeight: 'bold',
    marginRight: 10,
    fontSize: 16,
    textTransform: 'uppercase',
  },
  eventTitle: {
    fontSize: 16,
  },
  btnwrap: {marginBottom: 10,marginTop: 10,},
  buttonText: {
    color: "white",
    fontSize: 16,
    fontWeight: "bold",
  },
  date: {
    color: "#9A82F6",
    fontSize: 16,
    fontWeight: "bold",
  },
  customButton: {
    backgroundColor: "#9A82F6",
    paddingVertical: 10,
    paddingHorizontal: 20,
    borderRadius: 30,
    alignItems: "center",
  },
});
```

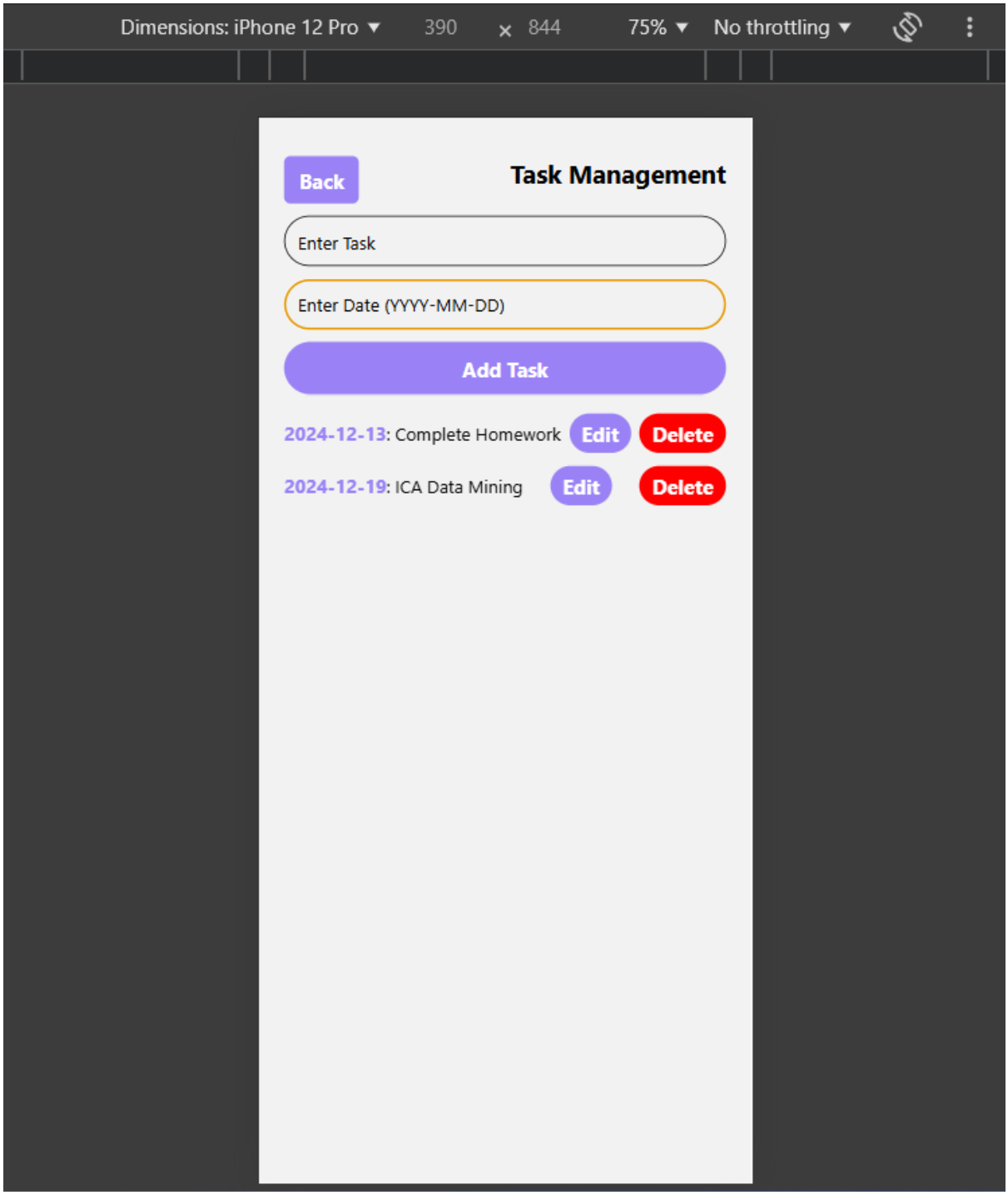
Preview

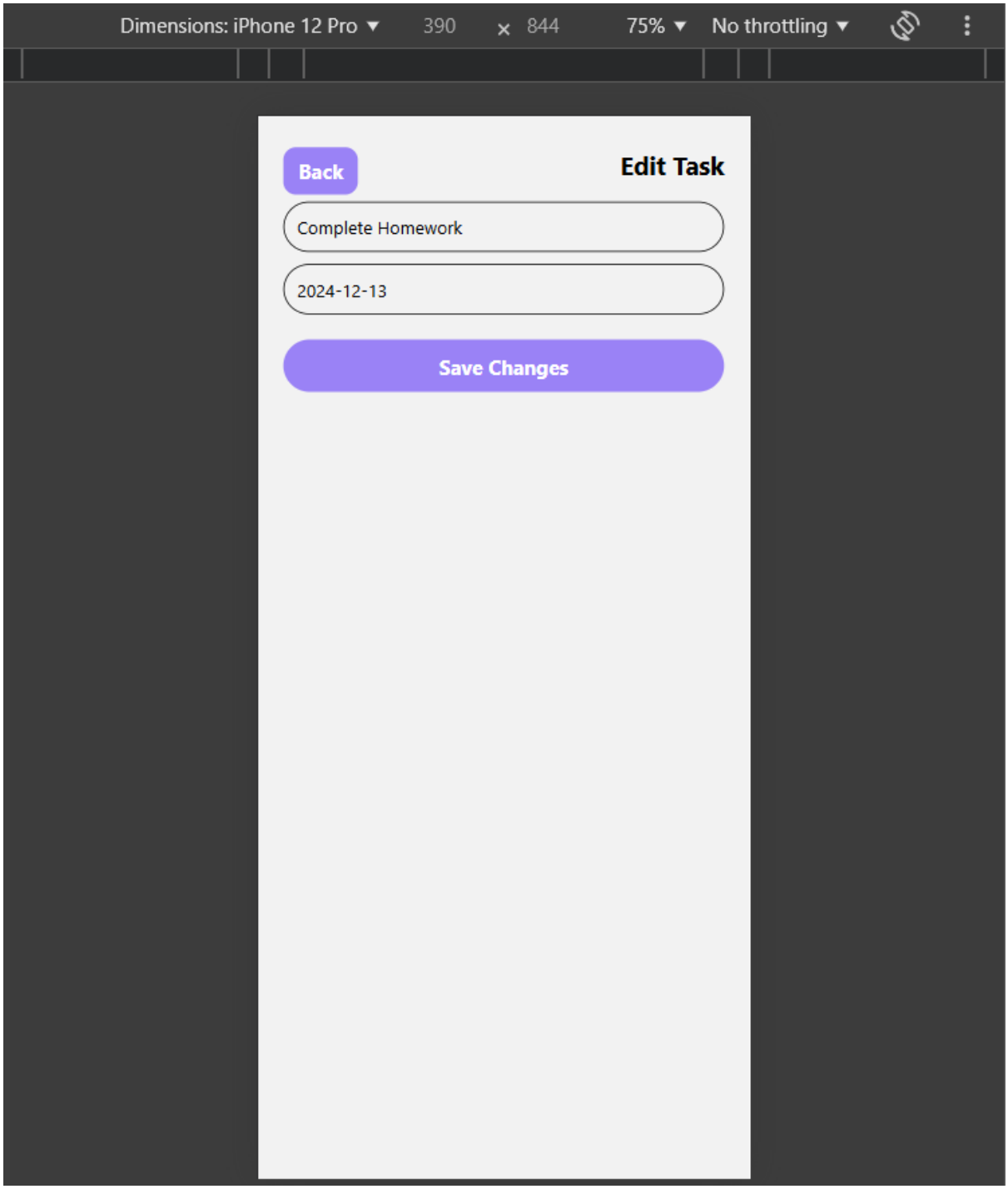

Dimensions: iPhone 12 Pro
390
844
75%
No throttling
Welcome to Task Manager
Organize your tasks effectively and stay productive!
Go to Calendar

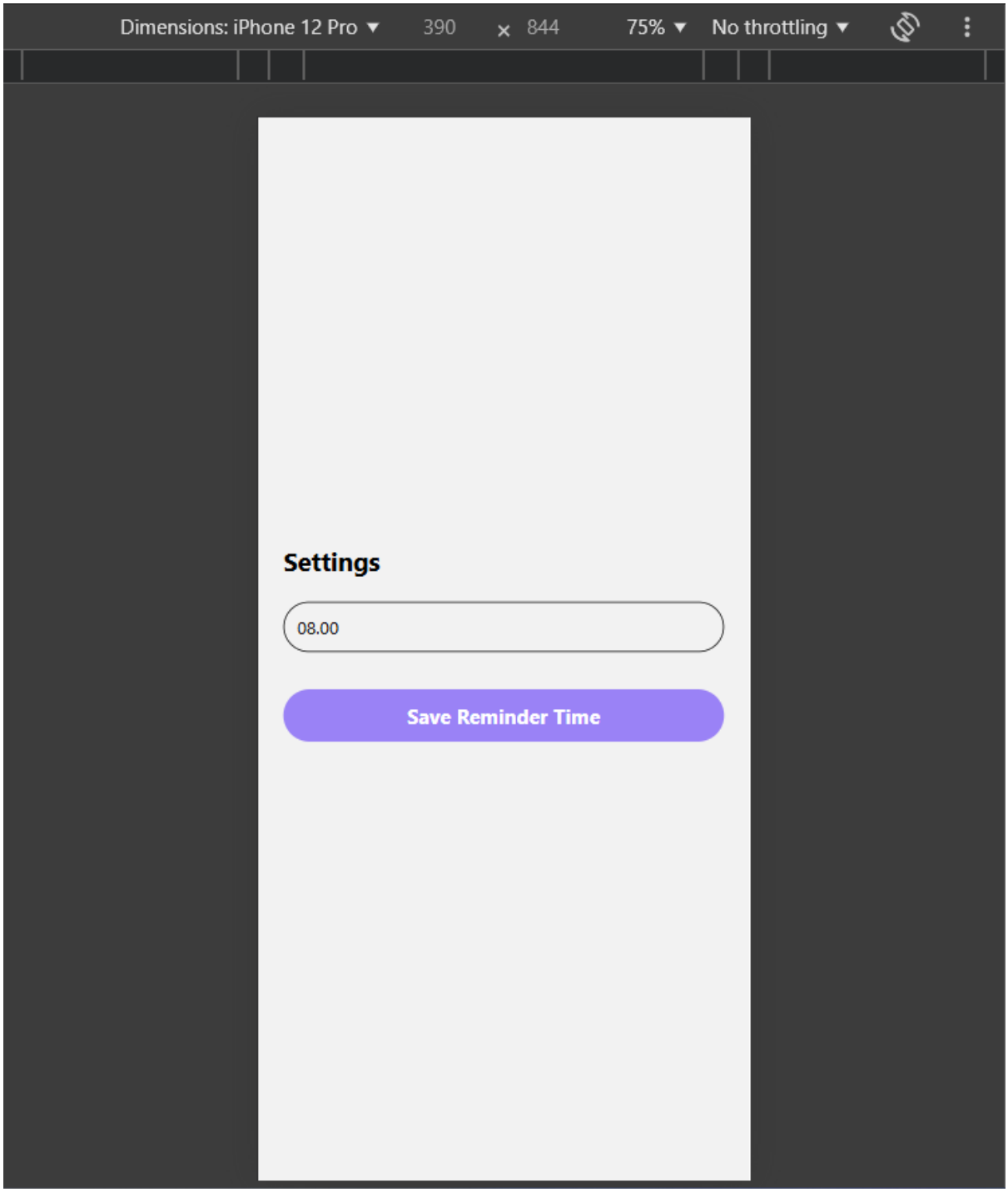

Dimensions: iPhone 12 Pro
390
844
75%
No throttling
December 2024
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
Go to Settings
Manage Tasks
Events for Select a Date

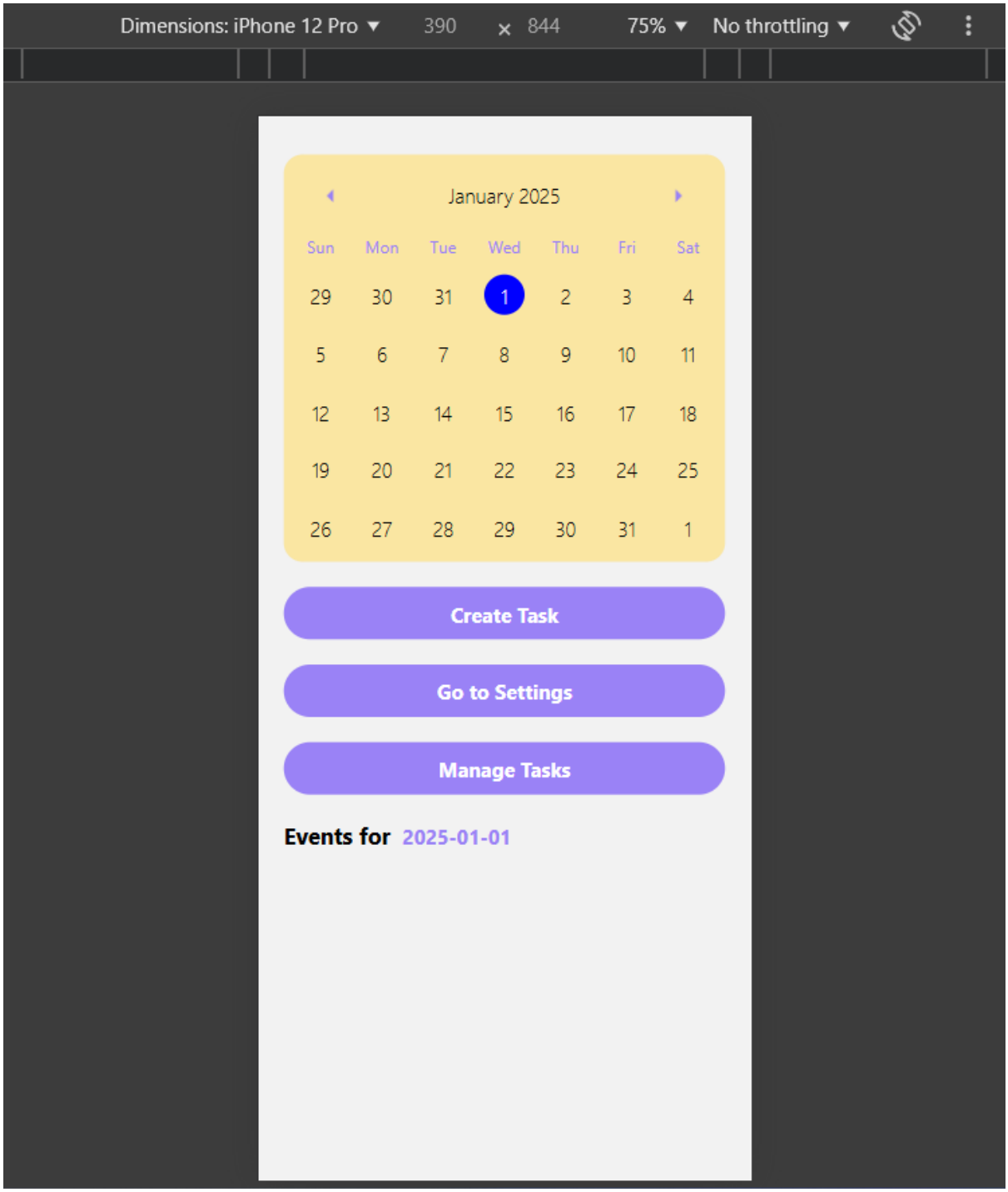

Dimensions: iPhone 12 Pro
390
844
75%
No throttling
December 2024
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31 1 2 3 4
Create Task
Go to Settings
Manage Tasks
Events for 2024-12-15
EXAM Physics Exam


Dimensions: iPhone 12 Pro
390
844
75%
No throttling
Back
Task Management
Enter Task
Enter Date (YYYY-MM-DD)
Add Task
2024-12-13: Complete Homework
Edit
Delete


Dimensions: iPhone 12 Pro
390
844
75%
No throttling
Back
Task Management
Enter Task
Enter Date (YYYY-MM-DD)
Add Task
2024-12-13: Complete Homework
Edit
Delete
2024-12-19: ICA Data Mining
Edit
Delete


Dimensions: iPhone 12 Pro
390
844
75%
No throttling
Back
Edit Task
Complete Homework
2024-12-13
Save Changes


Dimensions: iPhone 12 Pro
390
844
75%
No throttling
Settings
08.00
Save Reminder Time


Dimensions: iPhone 12 Pro
390
844
75%
No throttling
January 2025
Sun Mon Tue Wed Thu Fri Sat
29 30 31 1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31 1
Create Task
Go to Settings
Manage Tasks
Events for 2025-01-01