**Tutorial 03**
**Registration Number:-** 2018/ICTS/47
**Name:-**Dinuka Dulanjana

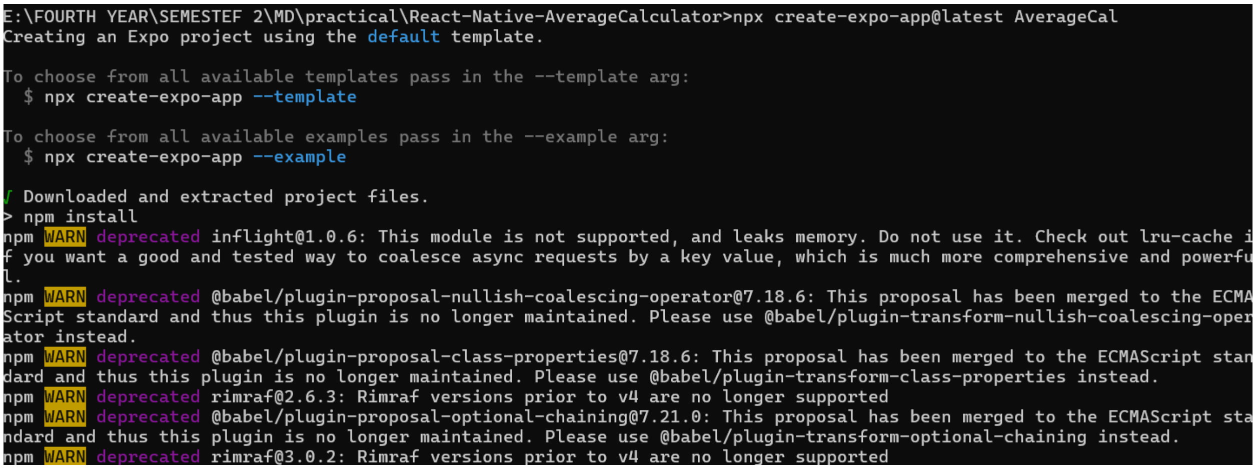
## **Project Title**: *Student Registration and Average Marks Calculator*

**Description**:
This project is a React Native mobile app designed to register a student’s details and calculate their average marks. Users can input their first name, last name, email, password, and marks for three subjects. Upon clicking the "REGISTER & CALCULATE AVERAGE" button, the app computes the average and displays the student’s full name along with their average marks on the screen. If invalid inputs are provided, an appropriate error message is displayed.

**Features**:

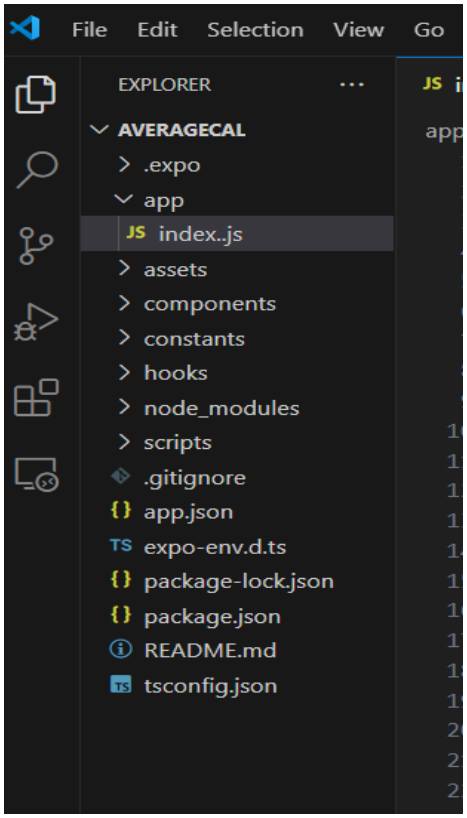
- User-friendly input form for student details.
- Validates marks to ensure correct input.
- Dynamically calculates and displays the average marks.
- Responsive UI with proper styling and layout.

### 01). Setup Screenshots:



```
E:\FOURTH YEAR\SEMESTEF 2\MD\practical\React-Native-AverageCalculator>npx create-expo-app@latest AverageCal
Creating an Expo project using the default template.

To choose from all available templates pass in the --template arg:
  $ npx create-expo-app --template

To choose from all available examples pass in the --example arg:
  $ npx create-expo-app --example

√ Downloaded and extracted project files.
> npm install
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMA
Script standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-oper
ator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript stan
dard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm WARN deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript sta
ndard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
```

- Capture the project folder structure in Visual Studio Code editor



```
AVERAGECAL
  > .expo
  v app
      index..js
  > assets
  > components
  > constants
  > hooks
  > node_modules
  > scripts
    .gitignore
    app.json
    expo-env.d.ts
    package-lock.json
    package.json
    README.md
    tsconfig.json
```

## 02) .Code Development Screenshots:

### Imports

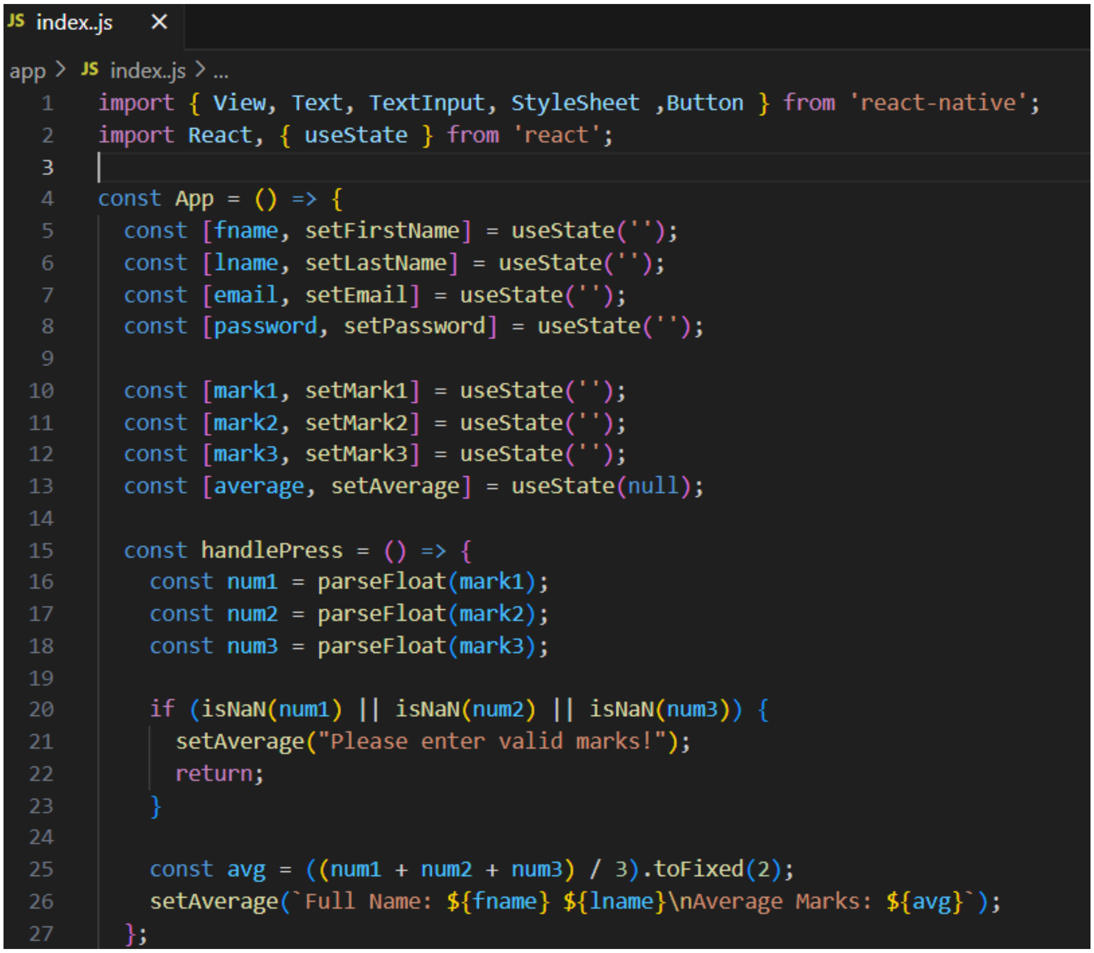
- **View**: A container component for layout.
- **Text**: Displays static or dynamic text.
- **TextInput**: Input fields for user data (e.g., name, email, marks).
- **Button**: A clickable button for actions.
- **useState**: Manages state (user inputs and calculated average).

### State Variables

- **fname, lname:** Store the user's first and last names.
- **email:** Stores the user's email address.
- **password:** Stores the password input (hidden with stars).
- **mark1, mark2, mark3:** Store marks for three subjects.
- **average:** Holds the calculated average and full name result.

### Calculation Logic

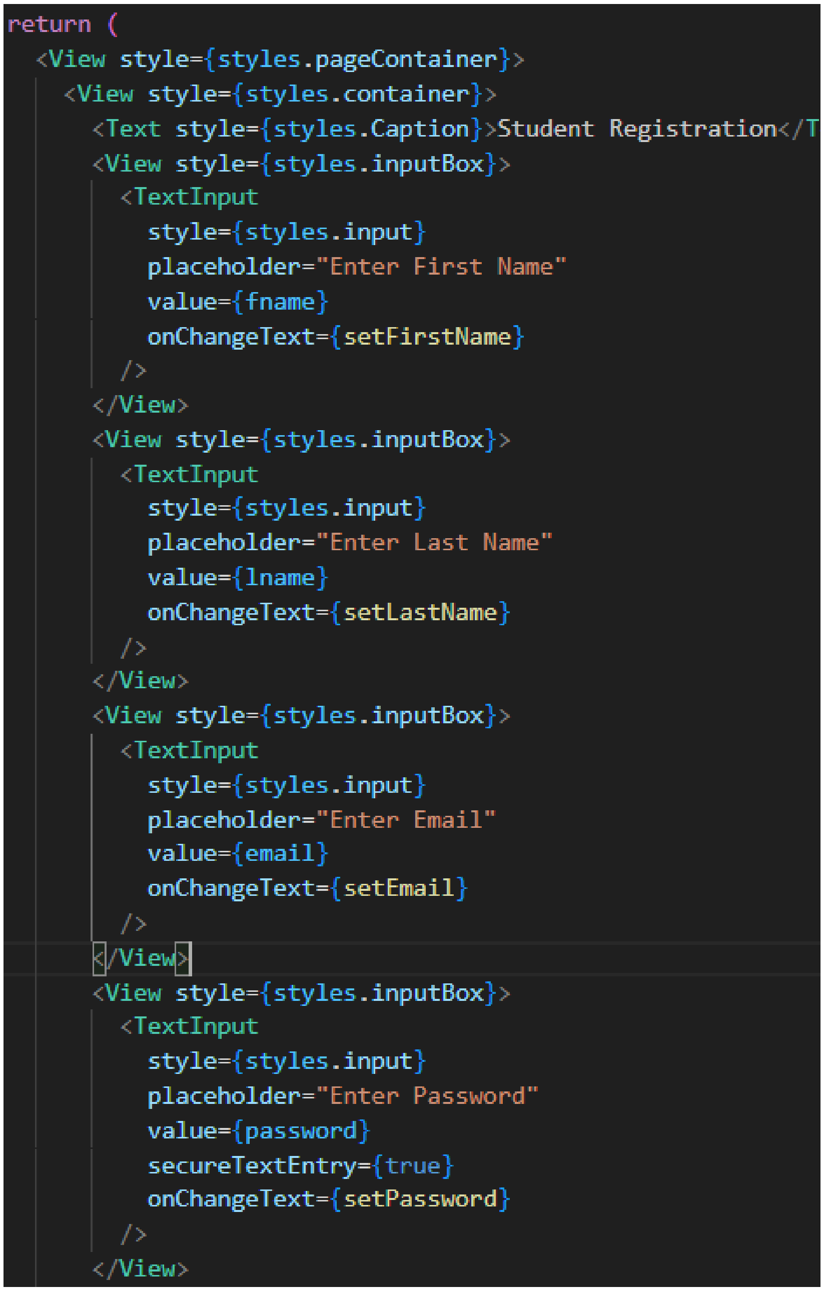
- **parseFloat:** Converts string input to a number for calculations.
- **Validation**: Checks if all inputs are valid numbers.
- **Average Calculation**: Sums the marks, divides by 3, and rounds to 2 decimal places.
- **Result**: Updates the average state to display the student's name and average marks.



```
import { View, Text, TextInput, StyleSheet ,Button } from 'react-native';
import React, { useState } from 'react';

const App = () => {
  const [fname, setFirstName] = useState('');
  const [lname, setLastName] = useState('');
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const [mark1, setMark1] = useState('');
  const [mark2, setMark2] = useState('');
  const [mark3, setMark3] = useState('');
  const [average, setAverage] = useState(null);

  const handlePress = () => {
    const num1 = parseFloat(mark1);
    const num2 = parseFloat(mark2);
    const num3 = parseFloat(mark3);

    if (isNaN(num1) || isNaN(num2) || isNaN(num3)) {
      setAverage("Please enter valid marks!");
      return;
    }

    const avg = ((num1 + num2 + num3) / 3).toFixed(2);
    setAverage(`Full Name: ${fname} ${lname}\nAverage Marks: ${avg}`);
  };
```

### Input Fields

- **TextInput**: Collects user input for each field (e.g., name, marks).
- **onChangeText**: Updates the corresponding state variable.
- **secureTextEntry**: Masks password input with stars (*).



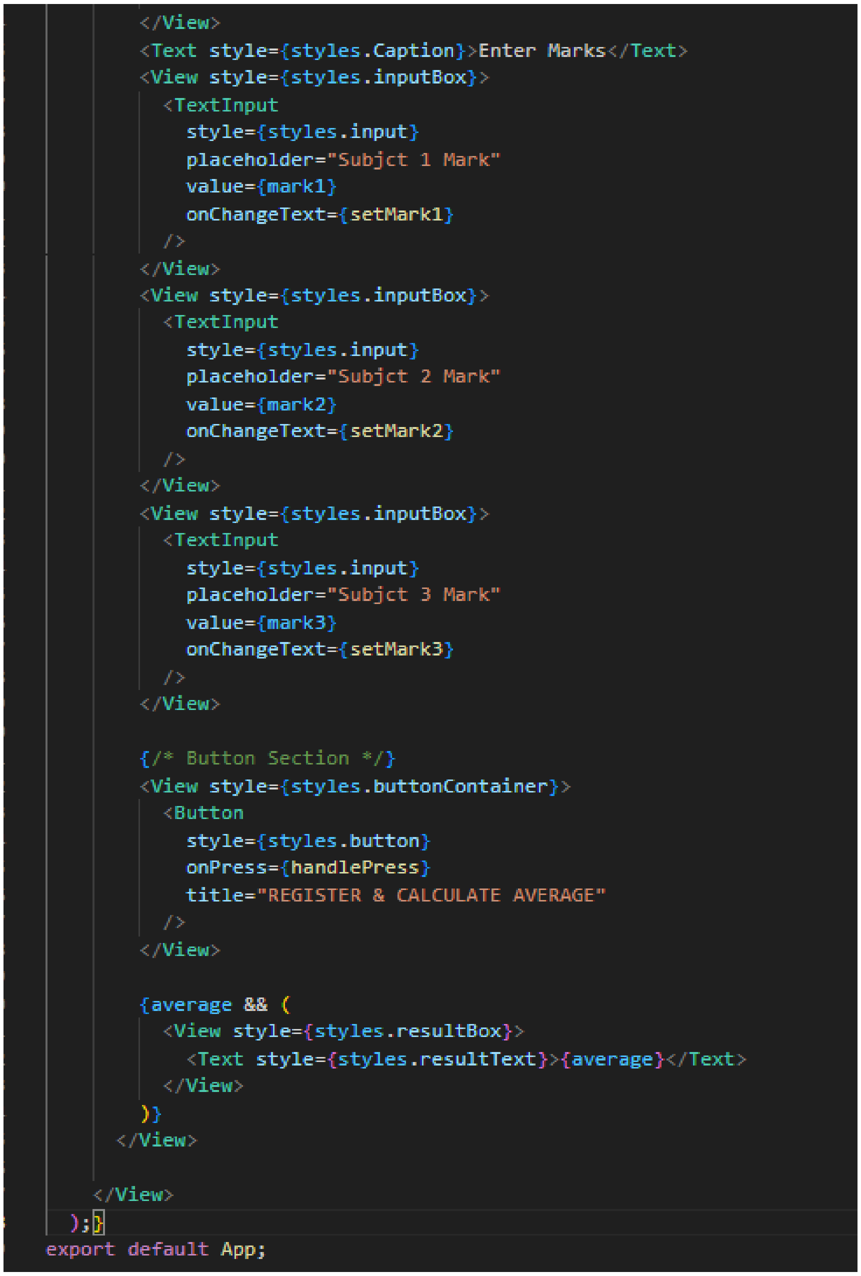
```
return (
  <View style={styles.pageContainer}>
    <View style={styles.container}>
      <Text style={styles.Caption}>Student Registration</T
      <View style={styles.inputBox}>
        <TextInput
          style={styles.input}
          placeholder="Enter First Name"
          value={fname}
          onChangeText={setFirstName}
        />
      </View>
      <View style={styles.inputBox}>
        <TextInput
          style={styles.input}
          placeholder="Enter Last Name"
          value={lname}
          onChangeText={setLastName}
        />
      </View>
      <View style={styles.inputBox}>
        <TextInput
          style={styles.input}
          placeholder="Enter Email"
          value={email}
          onChangeText={setEmail}
        />
      </View>
      <View style={styles.inputBox}>
        <TextInput
          style={styles.input}
          placeholder="Enter Password"
          value={password}
          secureTextEntry={true}
          onChangeText={setPassword}
        />
      </View>
```

### Button

- **onPress**: Triggers the calculation when clicked.
- **handlePress**: Called to calculate the average and update the result.

### Displaying Results

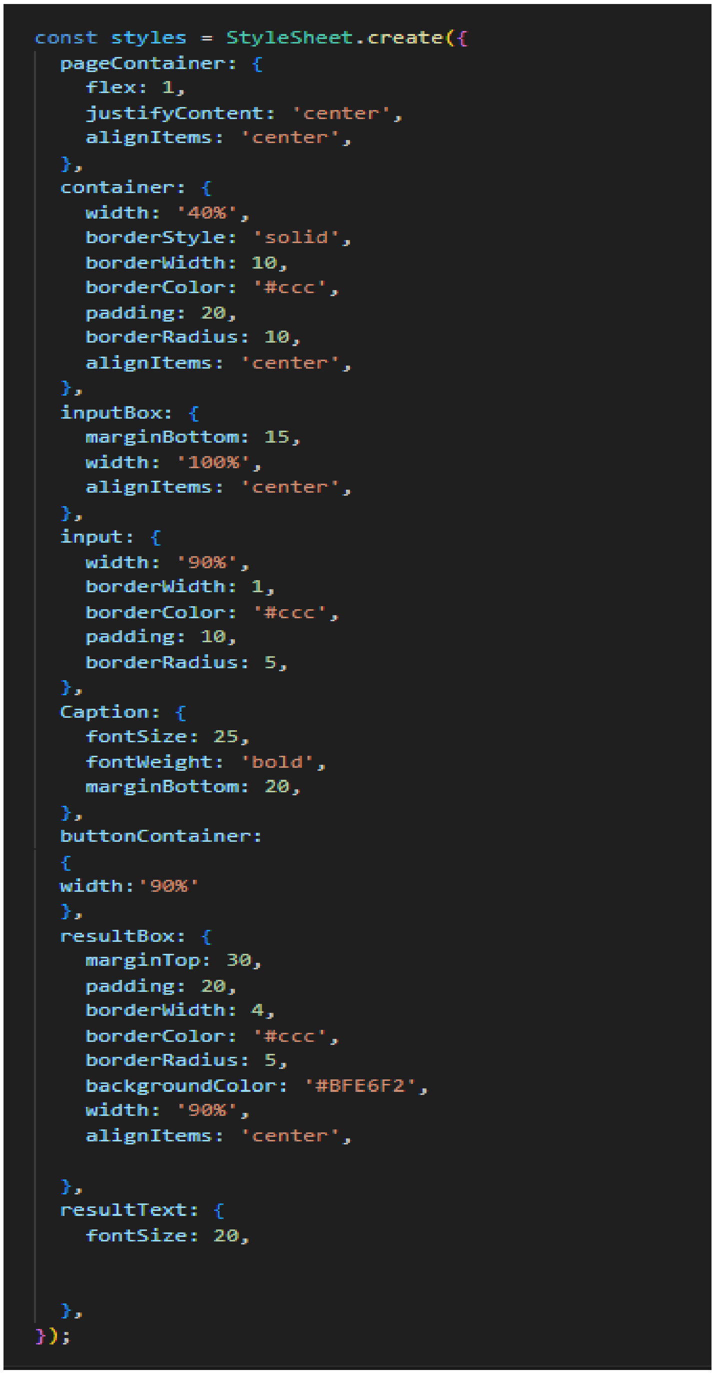
- **Conditional Rendering**: Displays the result only when average is not null.
- **Dynamic Content**: Shows the full name and average marks.



```
        </View>
        <Text style={styles.Caption}>Enter Marks</Text>
        <View style={styles.inputBox}>
          <TextInput
            style={styles.input}
            placeholder="Subjct 1 Mark"
            value={mark1}
            onChangeText={setMark1}
          />
        </View>
        <View style={styles.inputBox}>
          <TextInput
            style={styles.input}
            placeholder="Subjct 2 Mark"
            value={mark2}
            onChangeText={setMark2}
          />
        </View>
        <View style={styles.inputBox}>
          <TextInput
            style={styles.input}
            placeholder="Subjct 3 Mark"
            value={mark3}
            onChangeText={setMark3}
          />
        </View>

        {/* Button Section */}
        <View style={styles.buttonContainer}>
          <Button
            style={styles.button}
            onPress={handlePress}
            title="REGISTER & CALCULATE AVERAGE"
          />
        </View>

        {average && (
          <View style={styles.resultBox}>
            <Text style={styles.resultText}>{average}</Text>
          </View>
        )}
      </View>

    </View>
  );}
export default App;
```

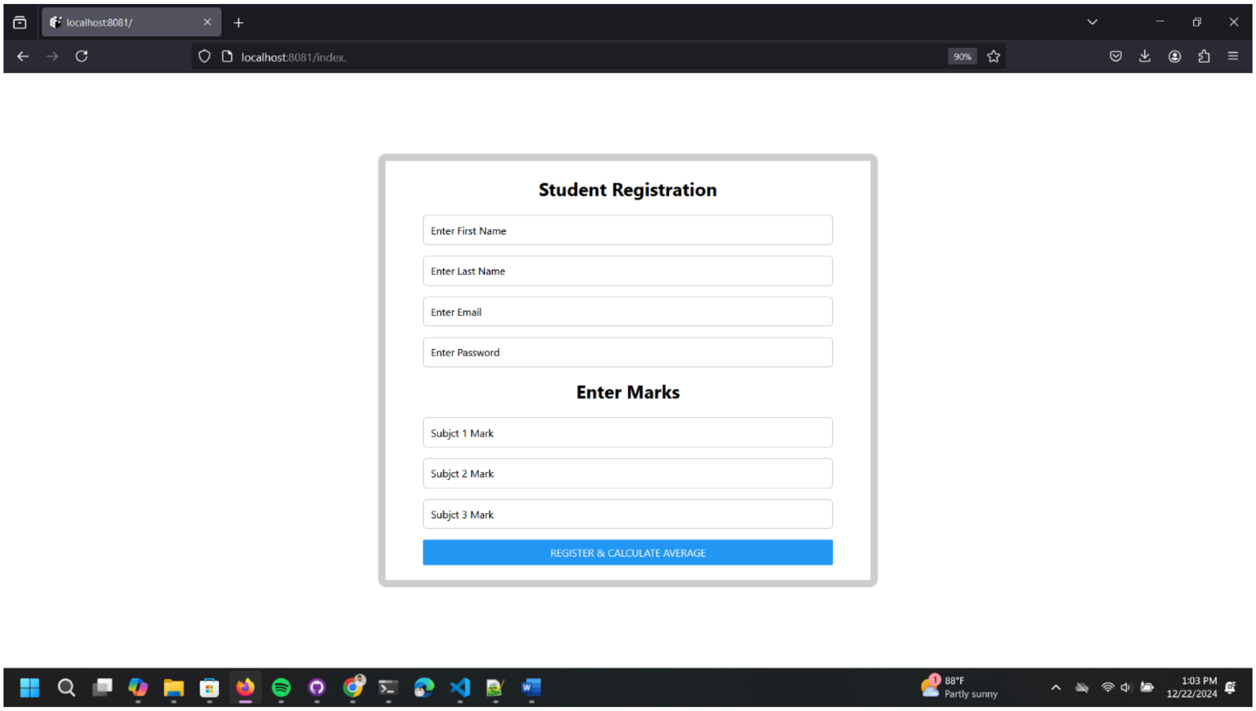
### Styling

- **container**: Centers and styles the main form.
- **input**: Adds padding, borders, and rounded corners to input fields.
- **resultBox**: Styles the result display with a background color and borders.

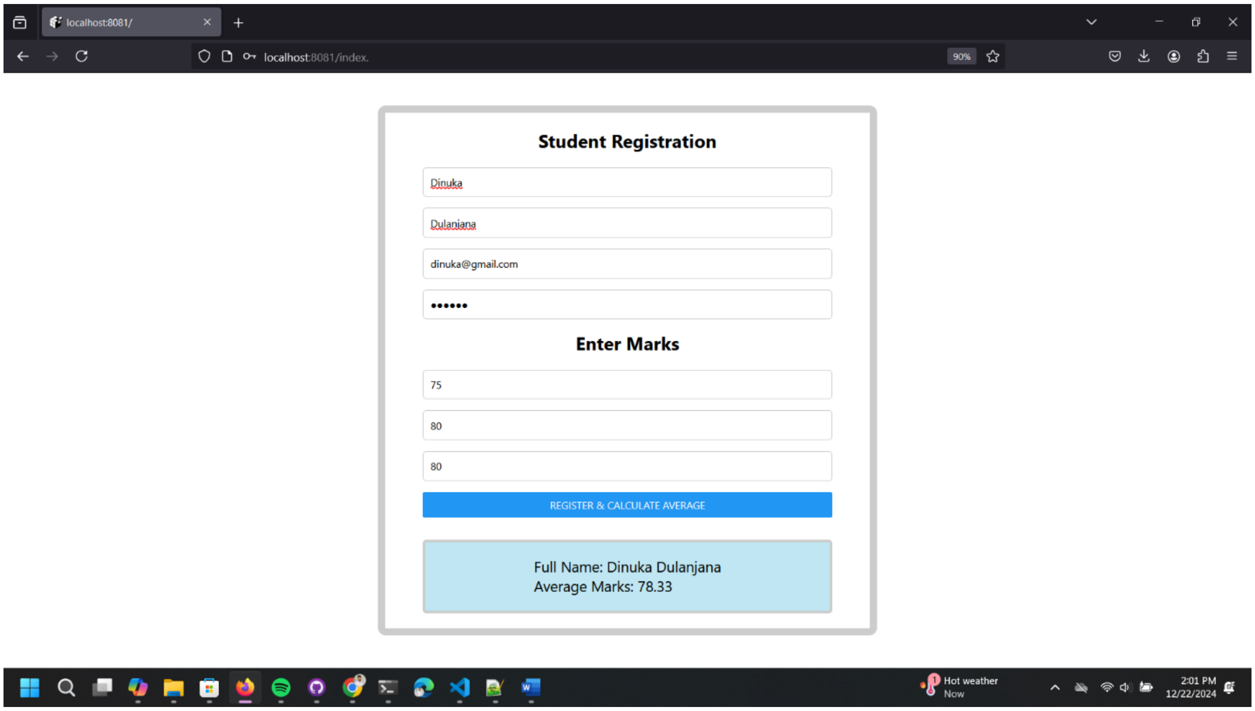


```
const styles = StyleSheet.create({
  pageContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  container: {
    width: '40%',
    borderStyle: 'solid',
    borderWidth: 10,
    borderColor: '#ccc',
    padding: 20,
    borderRadius: 10,
    alignItems: 'center',
  },
  inputBox: {
    marginBottom: 15,
    width: '100%',
    alignItems: 'center',
  },
  input: {
    width: '90%',
    borderWidth: 1,
    borderColor: '#ccc',
    padding: 10,
    borderRadius: 5,
  },
  Caption: {
    fontSize: 25,
    fontWeight: 'bold',
    marginBottom: 20,
  },
  buttonContainer:
  {
  width:'90%'
  },
  resultBox: {
    marginTop: 30,
    padding: 20,
    borderWidth: 4,
    borderColor: '#ccc',
    borderRadius: 5,
    backgroundColor: '#BFE6F2',
    width: '90%',
    alignItems: 'center',

  },
  resultText: {
    fontSize: 20,


  },
});
```

## Initial Screen:



## After Input:



Dinuka

Dulanjana

dinuka@gmail.com

••••••

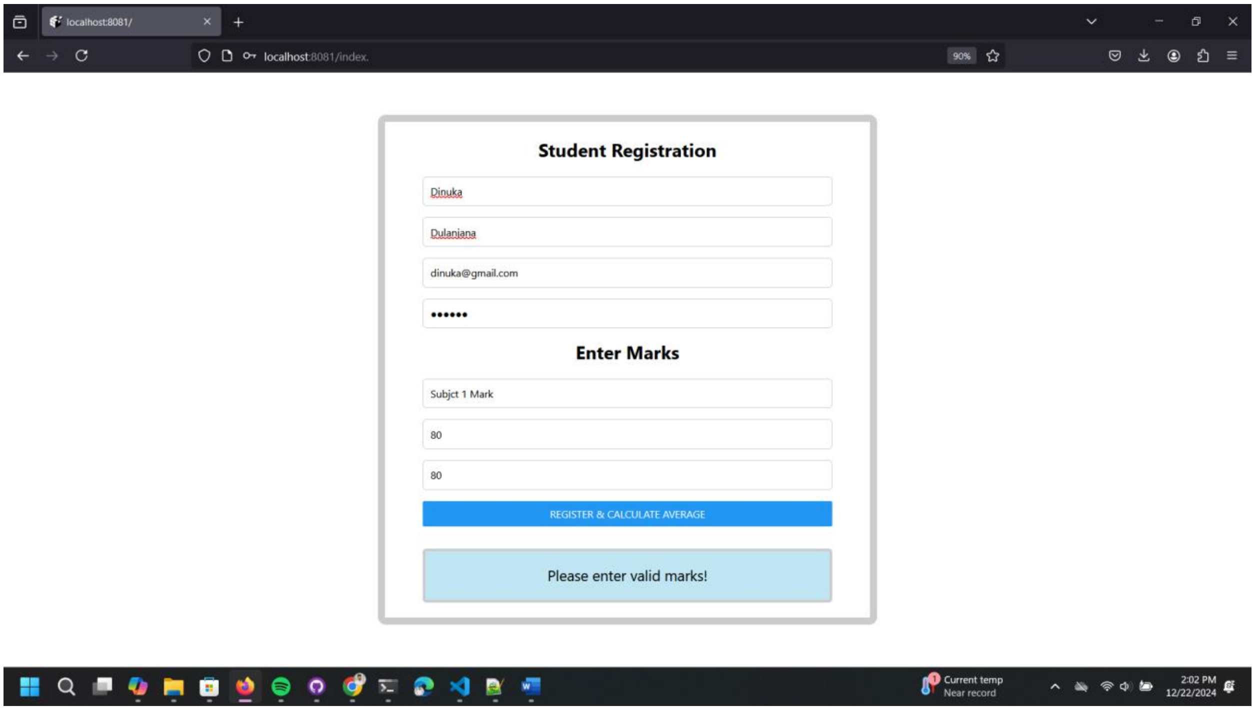
75

80

80

Full Name: Dinuka Dulanjana
Average Marks: 78.33

**Invalid Input**:



Student Registration

Dinuka

Dulanjana

dinuka@gmail.com

••••••

Enter Marks

Subjct 1 Mark

80

80

Please enter valid marks!