Mobile Application Development

MAD QUIZ # 2

SUBMITTED TO :

Dr. Kanwal Yousaf

SUBMITTED BY :

Aleeza Zaman

21-SE-31

DATE:

25-April-2024

GitHub Repository Link:

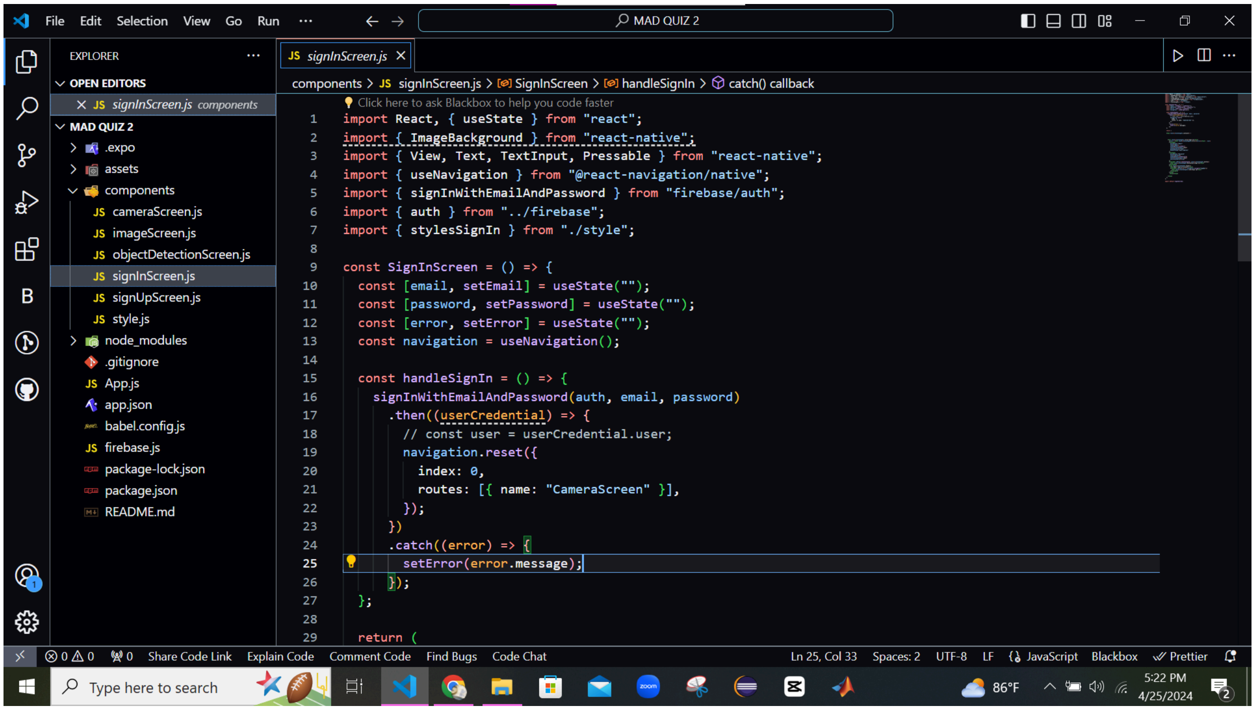
https://github.com/aleeza21/quiz2.git

UI Enhancement:

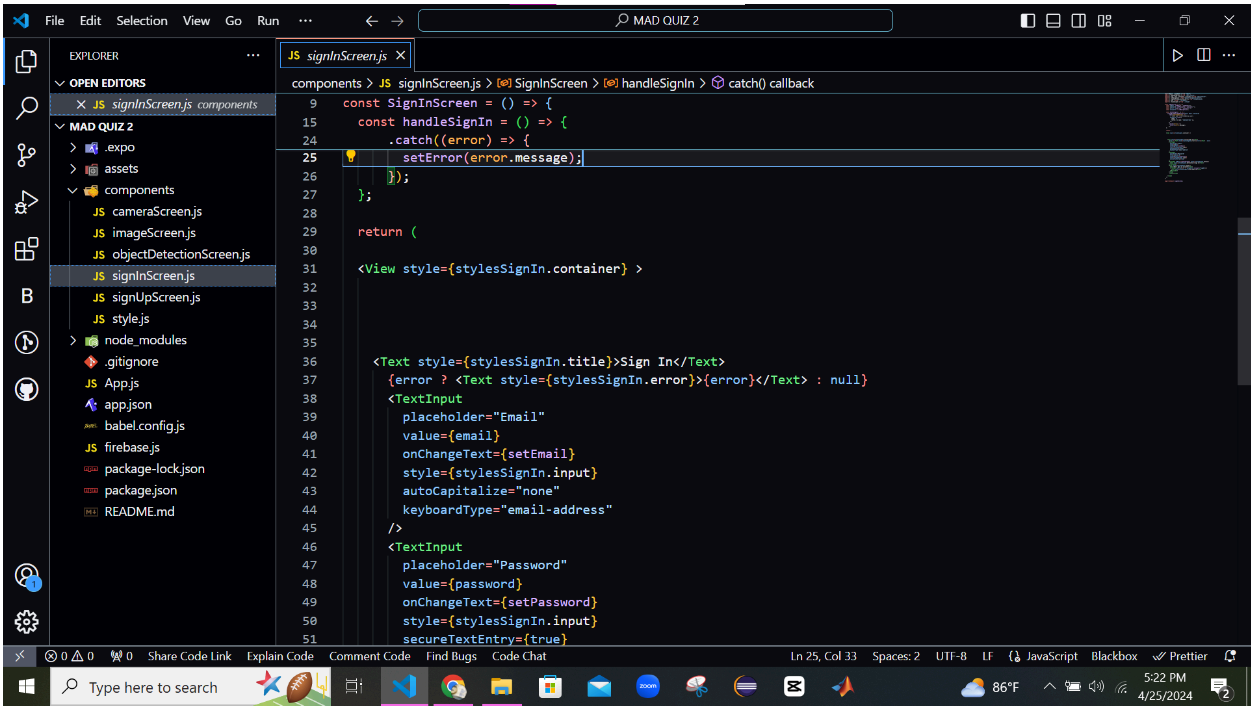
To enhance the user experience of the application, I've made several UI improvements, focusing on aesthetics, usability, and accessibility.

Firstly, I've revamped the color scheme to create a more visually appealing interface. I opted for a combination of soft, soothing colors to promote a sense of calm and clarity while using the application. The new colors not only look more modern but also contribute to reducing eye strain for users who spend extended periods interacting with the app.

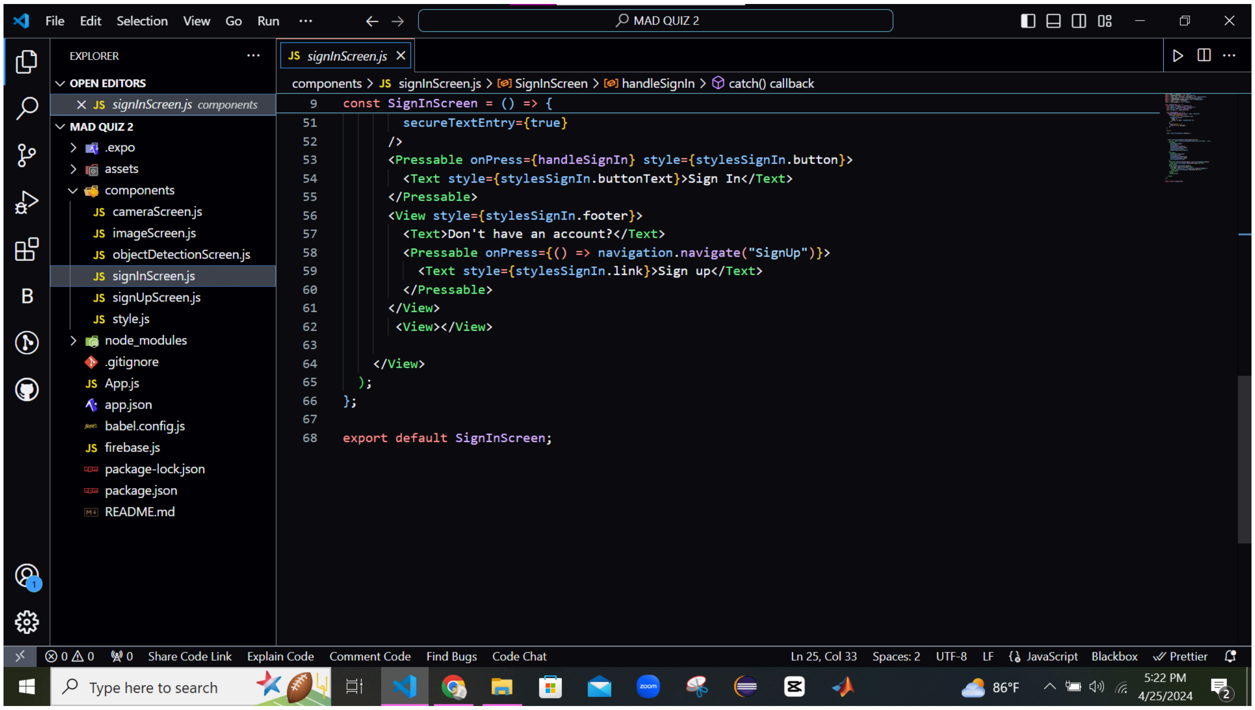
SignInScreen.js



```javascript
import React, { useState } from "react";
import { ImageBackground } from "react-native";
import { View, Text, TextInput, Pressable } from "react-native";
import { useNavigation } from "@react-navigation/native";
import { signInWithEmailAndPassword } from "firebase/auth";
import { auth } from "../firebase";
import { stylesSignIn } from "./style";

const SignInScreen = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [error, setError] = useState("");
  const navigation = useNavigation();

  const handleSignIn = () => {
    signInWithEmailAndPassword(auth, email, password)
      .then((userCredential) => {
        // const user = userCredential.user;
        navigation.reset({
          index: 0,
          routes: [{ name: "CameraScreen" }],
        });
      })
      .catch((error) => {
        setError(error.message);
      });
  };

  return (
```

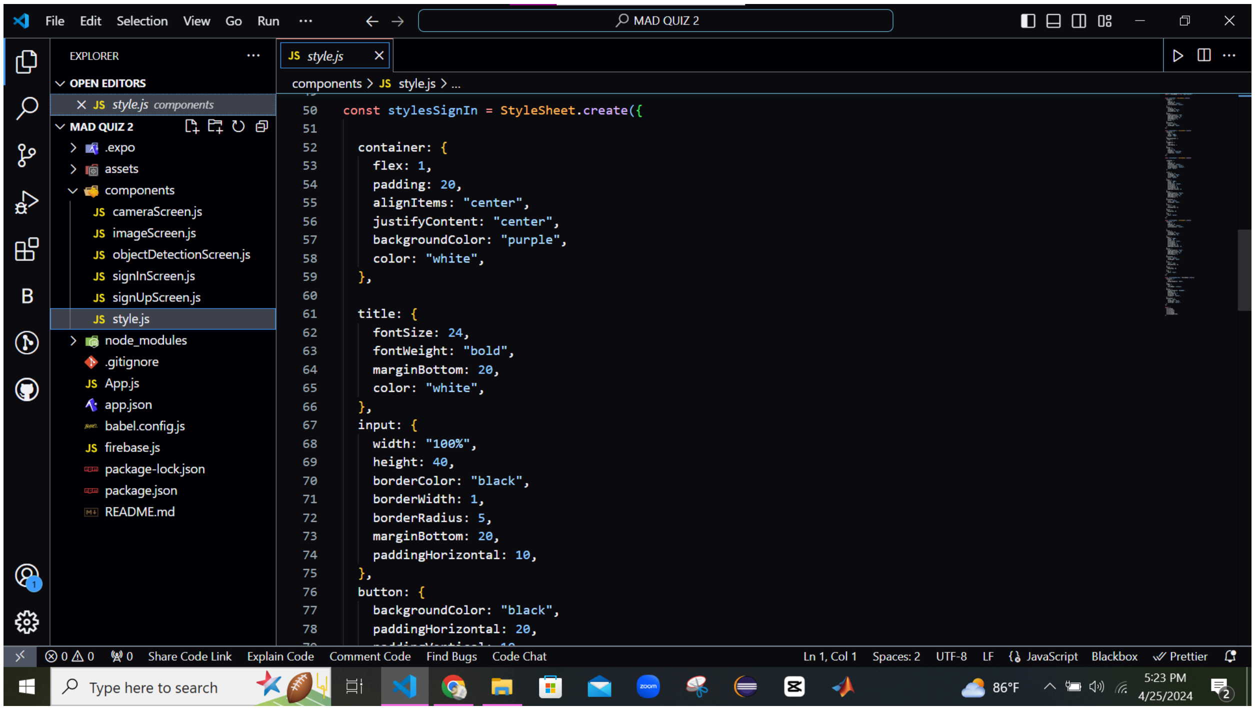


```
const SignInScreen = () => {
  const handleSignIn = () => {
      .catch((error) => {
        setError(error.message);
      });
  };

  return (

  <View style={stylesSignIn.container} >




    <Text style={stylesSignIn.title}>Sign In</Text>
      {error ? <Text style={stylesSignIn.error}>{error}</Text> : null}
      <TextInput
        placeholder="Email"
        value={email}
        onChangeText={setEmail}
        style={stylesSignIn.input}
        autoCapitalize="none"
        keyboardType="email-address"
      />
      <TextInput
        placeholder="Password"
        value={password}
        onChangeText={setPassword}
        style={stylesSignIn.input}
        secureTextEntry={true}
```

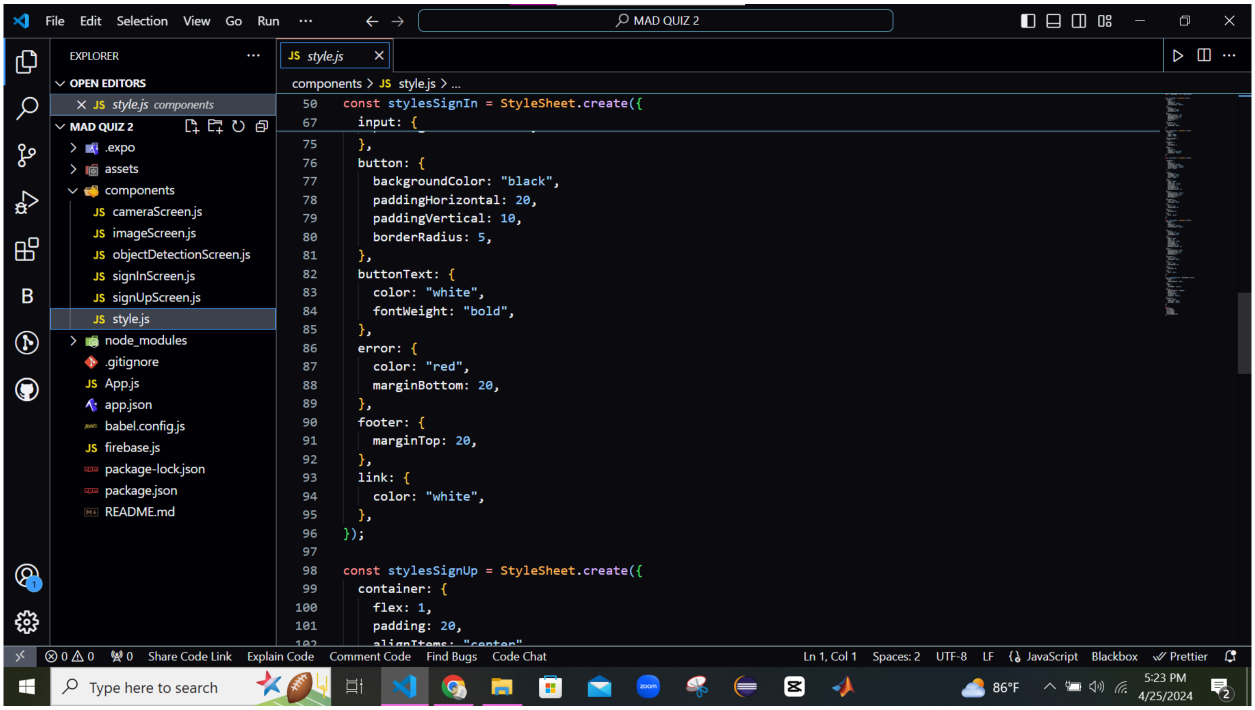


```
const SignInScreen = () => {
        secureTextEntry={true}
      />
      <Pressable onPress={handleSignIn} style={stylesSignIn.button}>
        <Text style={stylesSignIn.buttonText}>Sign In</Text>
      </Pressable>
      <View style={stylesSignIn.footer}>
        <Text>Don't have an account?</Text>
        <Pressable onPress={() => navigation.navigate("SignUp")}>
          <Text style={stylesSignIn.link}>Sign up</Text>
        </Pressable>
      </View>
       <View></View>

    </View>
  );
};

export default SignInScreen;
```

Style.js



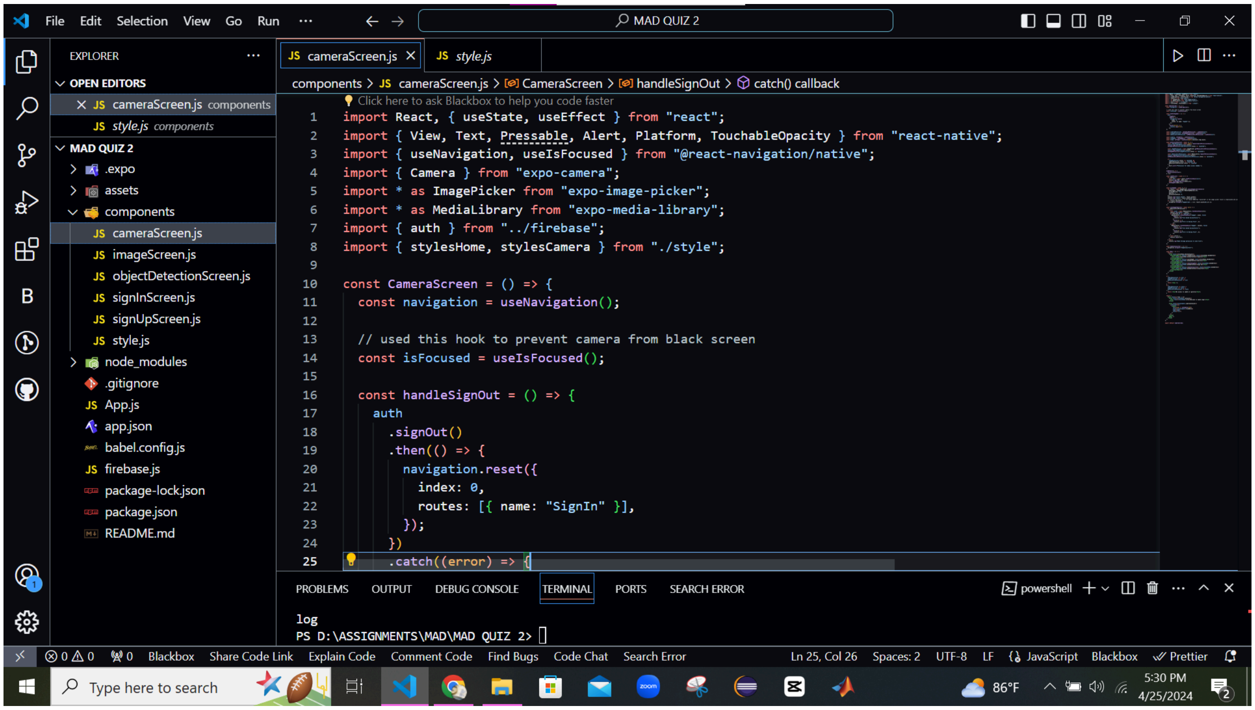
```
const stylesSignIn = StyleSheet.create({

  container: {
    flex: 1,
    padding: 20,
    alignItems: "center",
    justifyContent: "center",
    backgroundColor: "purple",
    color: "white",
  },

  title: {
    fontSize: 24,
    fontWeight: "bold",
    marginBottom: 20,
    color: "white",
  },
  input: {
    width: "100%",
    height: 40,
    borderColor: "black",
    borderWidth: 1,
    borderRadius: 5,
    marginBottom: 20,
    paddingHorizontal: 10,
  },
  button: {
    backgroundColor: "black",
    paddingHorizontal: 20,
    paddingVertical: 10,
```



```
const stylesSignIn = StyleSheet.create({
  input: {
  },
  button: {
    backgroundColor: "black",
    paddingHorizontal: 20,
    paddingVertical: 10,
    borderRadius: 5,
  },
  buttonText: {
    color: "white",
    fontWeight: "bold",
  },
  error: {
    color: "red",
    marginBottom: 20,
  },
  footer: {
    marginTop: 20,
  },
  link: {
    color: "white",
  },
});

const stylesSignUp = StyleSheet.create({
  container: {
    flex: 1,
    padding: 20,
    alignItems: "center"
```

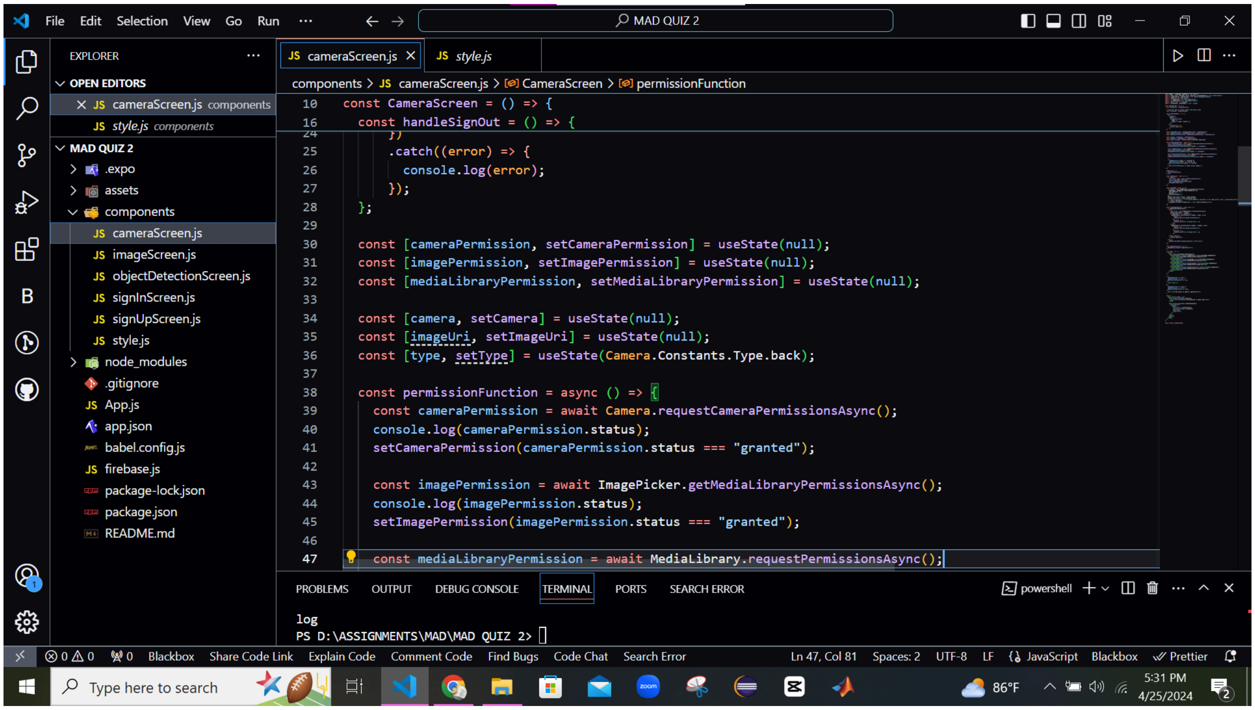
Additionally, menus are introduces to streamline navigation within the application. By organizing features and options into intuitive menus, users can easily find what they need without getting lost in cluttered interfaces. These menus are designed to be easily accessible, either through traditional dropdowns or sleek, modern slide-in panels, depending on the context.

CameraScreen.js


File Edit Selection View Go Run ···
MAD QUIZ 2
EXPLORER
OPEN EDITORS
cameraScreen.js components
style.js components
MAD QUIZ 2
.expo
assets
components
cameraScreen.js
imageScreen.js
objectDetectionScreen.js
signInScreen.js
signUpScreen.js
style.js
node_modules
.gitignore
App.js
app.json
babel.config.js
firebase.js
package-lock.json
package.json
README.md
components > cameraScreen.js > CameraScreen > handleSignOut > catch() callback
Click here to ask Blackbox to help you code faster
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
powershell
Blackbox Share Code Link Explain Code Comment Code Find Bugs Code Chat Search Error
Ln 25, Col 26 Spaces: 2 UTF-8 LF JavaScript Blackbox Prettier
Type here to search
86°F
5:30 PM
4/25/2024

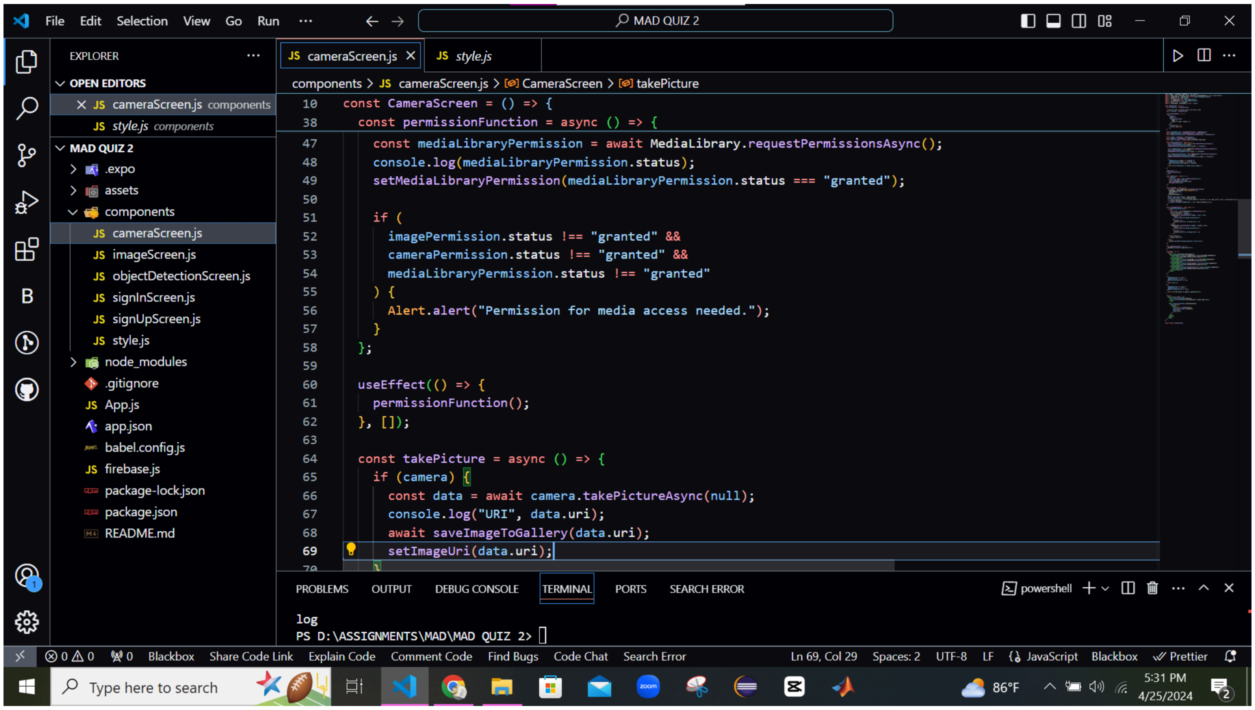
```javascript
import React, { useState, useEffect } from "react";
import { View, Text, Pressable, Alert, Platform, TouchableOpacity } from "react-native";
import { useNavigation, useIsFocused } from "@react-navigation/native";
import { Camera } from "expo-camera";
import * as ImagePicker from "expo-image-picker";
import * as MediaLibrary from "expo-media-library";
import { auth } from "../firebase";
import { stylesHome, stylesCamera } from "./style";

const CameraScreen = () => {
  const navigation = useNavigation();

  // used this hook to prevent camera from black screen
  const isFocused = useIsFocused();

  const handleSignOut = () => {
    auth
      .signOut()
      .then(() => {
        navigation.reset({
          index: 0,
          routes: [{ name: "SignIn" }],
        });
      })
      .catch((error) => {
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```


File Edit Selection View Go Run ···
MAD QUIZ 2
EXPLORER
OPEN EDITORS
cameraScreen.js components
style.js components
MAD QUIZ 2
.expo
assets
components
cameraScreen.js
imageScreen.js
objectDetectionScreen.js
signInScreen.js
signUpScreen.js
style.js
node_modules
.gitignore
App.js
app.json
babel.config.js
firebase.js
package-lock.json
package.json
README.md
components > cameraScreen.js > CameraScreen > permissionFunction
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
powershell
Blackbox Share Code Link Explain Code Comment Code Find Bugs Code Chat Search Error
Ln 47, Col 81 Spaces: 2 UTF-8 LF JavaScript Blackbox Prettier
Type here to search
86°F
5:31 PM
4/25/2024

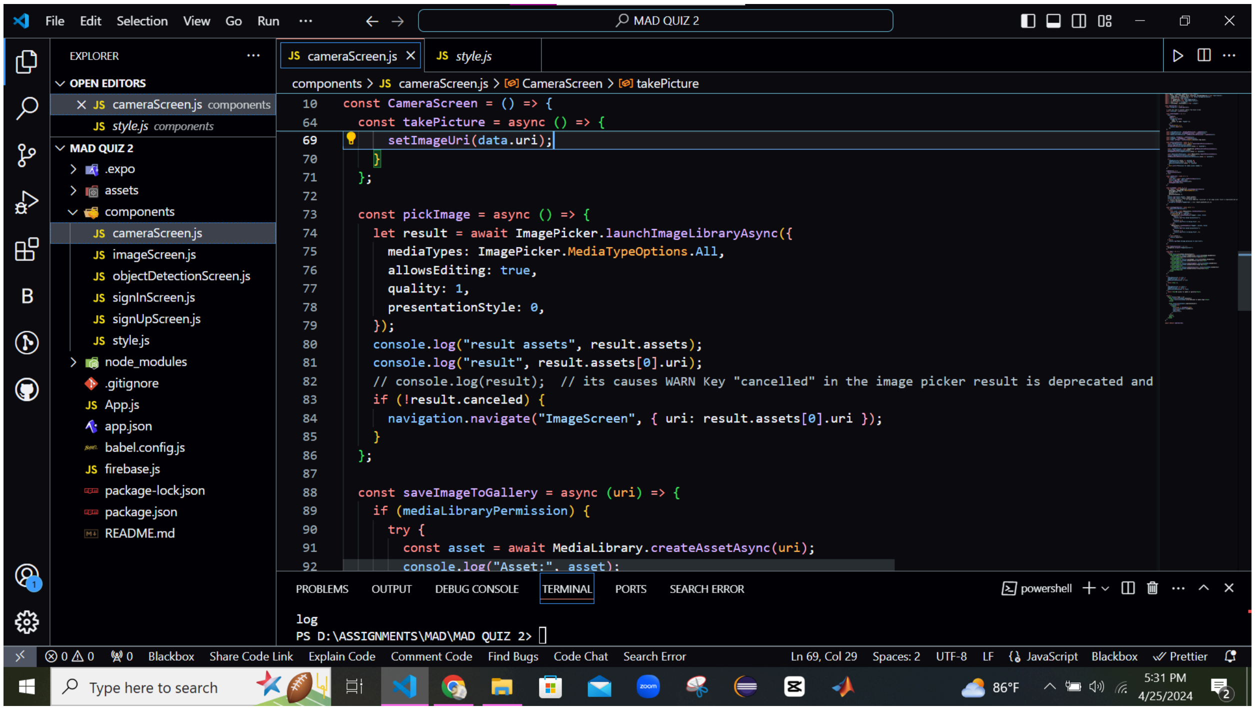
```javascript
const CameraScreen = () => {
  const handleSignOut = () => {
      })
      .catch((error) => {
        console.log(error);
      });
  };

  const [cameraPermission, setCameraPermission] = useState(null);
  const [imagePermission, setImagePermission] = useState(null);
  const [mediaLibraryPermission, setMediaLibraryPermission] = useState(null);

  const [camera, setCamera] = useState(null);
  const [imageUri, setImageUri] = useState(null);
  const [type, setType] = useState(Camera.Constants.Type.back);

  const permissionFunction = async () => {
    const cameraPermission = await Camera.requestCameraPermissionsAsync();
    console.log(cameraPermission.status);
    setCameraPermission(cameraPermission.status === "granted");

    const imagePermission = await ImagePicker.getMediaLibraryPermissionsAsync();
    console.log(imagePermission.status);
    setImagePermission(imagePermission.status === "granted");

    const mediaLibraryPermission = await MediaLibrary.requestPermissionsAsync();
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```



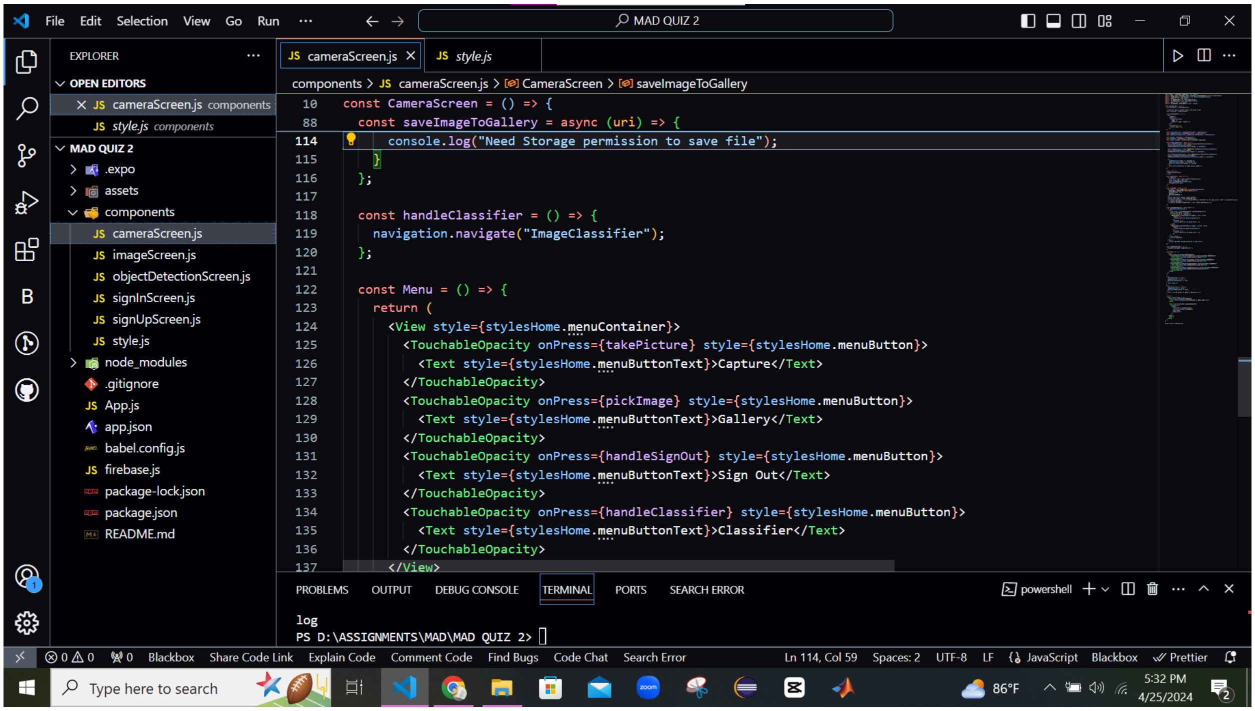
```
const CameraScreen = () => {
  const permissionFunction = async () => {
    const mediaLibraryPermission = await MediaLibrary.requestPermissionsAsync();
    console.log(mediaLibraryPermission.status);
    setMediaLibraryPermission(mediaLibraryPermission.status === "granted");

    if (
      imagePermission.status !== "granted" &&
      cameraPermission.status !== "granted" &&
      mediaLibraryPermission.status !== "granted"
    ) {
      Alert.alert("Permission for media access needed.");
    }
  };

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

  const takePicture = async () => {
    if (camera) {
      const data = await camera.takePictureAsync(null);
      console.log("URI", data.uri);
      await saveImageToGallery(data.uri);
      setImageUri(data.uri);
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```



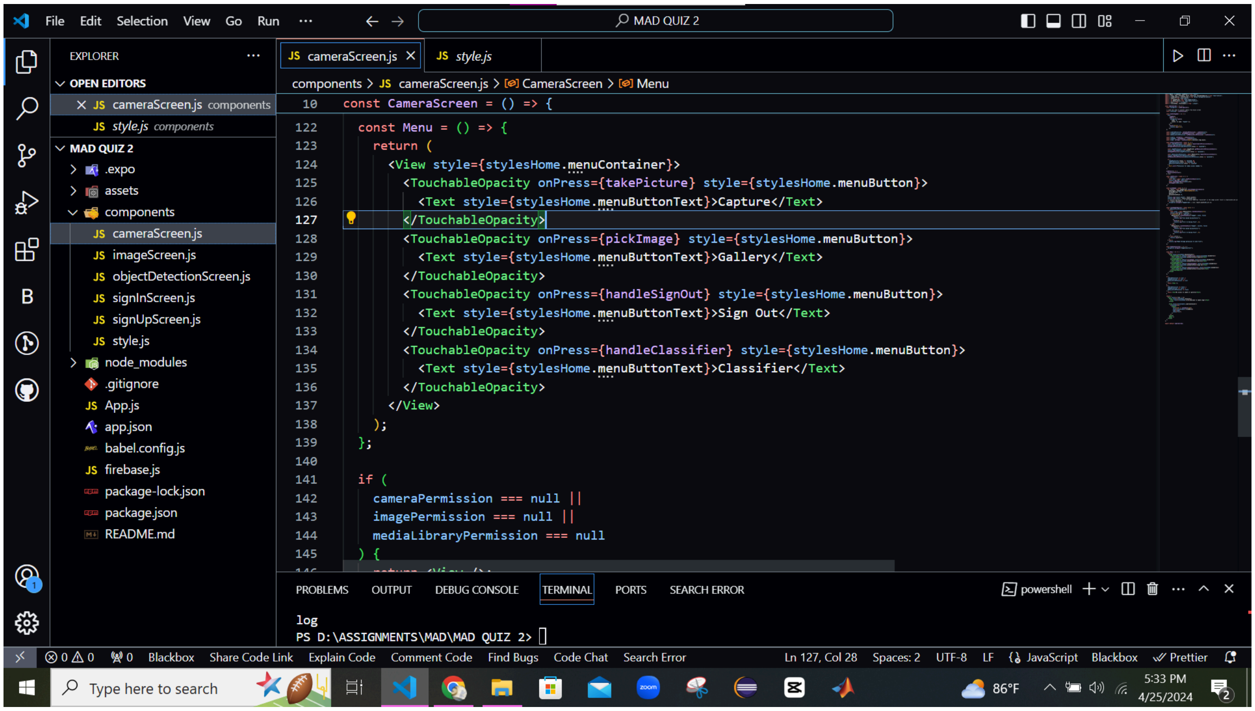
```
const CameraScreen = () => {
  const takePicture = async () => {
      setImageUri(data.uri);
    }
  };

  const pickImage = async () => {
    let result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.All,
      allowsEditing: true,
      quality: 1,
      presentationStyle: 0,
    });
    console.log("result assets", result.assets);
    console.log("result", result.assets[0].uri);
    // console.log(result);  // its causes WARN Key "cancelled" in the image picker result is deprecated and
    if (!result.canceled) {
      navigation.navigate("ImageScreen", { uri: result.assets[0].uri });
    }
  };

  const saveImageToGallery = async (uri) => {
    if (mediaLibraryPermission) {
      try {
        const asset = await MediaLibrary.createAssetAsync(uri);
        console.log("Asset:", asset);
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```



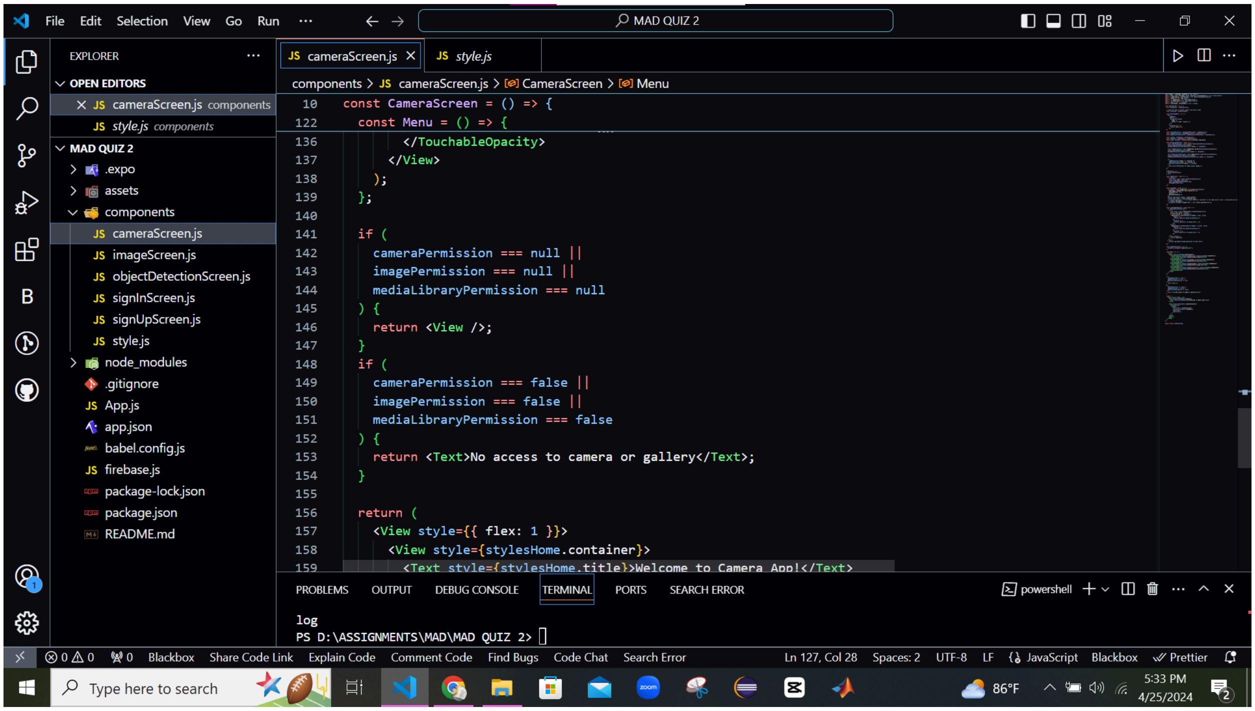
```javascript
const CameraScreen = () => {
  const saveImageToGallery = async (uri) => {
      console.log("Need Storage permission to save file");
    }
  };

  const handleClassifier = () => {
    navigation.navigate("ImageClassifier");
  };

  const Menu = () => {
    return (
      <View style={stylesHome.menuContainer}>
        <TouchableOpacity onPress={takePicture} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Capture</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={pickImage} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Gallery</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={handleSignOut} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Sign Out</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={handleClassifier} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Classifier</Text>
        </TouchableOpacity>
      </View>
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2>
```



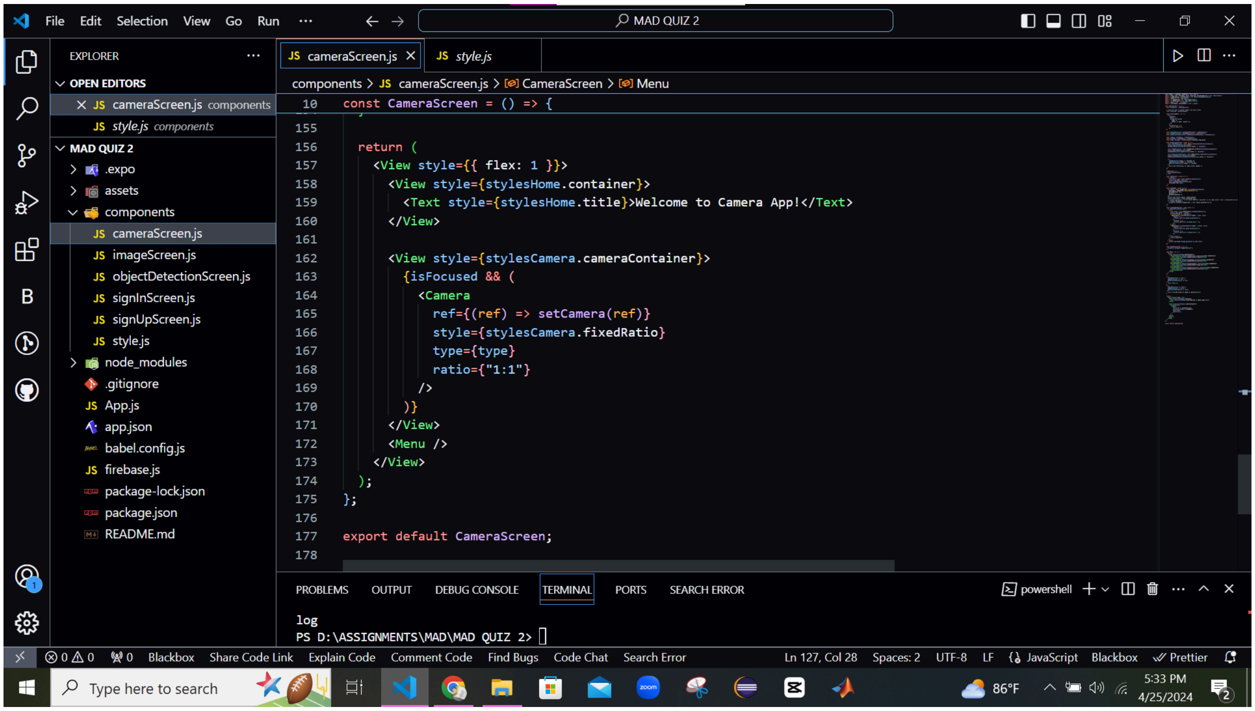
```javascript
const CameraScreen = () => {
  const Menu = () => {
    return (
      <View style={stylesHome.menuContainer}>
        <TouchableOpacity onPress={takePicture} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Capture</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={pickImage} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Gallery</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={handleSignOut} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Sign Out</Text>
        </TouchableOpacity>
        <TouchableOpacity onPress={handleClassifier} style={stylesHome.menuButton}>
          <Text style={stylesHome.menuButtonText}>Classifier</Text>
        </TouchableOpacity>
      </View>
    );
  };

  if (
    cameraPermission === null ||
    imagePermission === null ||
    mediaLibraryPermission === null
  ) {
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2>
```



```
const CameraScreen = () => {
  const Menu = () => {
        </TouchableOpacity>
      </View>
    );
  };

  if (
    cameraPermission === null ||
    imagePermission === null ||
    mediaLibraryPermission === null
  ) {
    return <View />;
  }
  if (
    cameraPermission === false ||
    imagePermission === false ||
    mediaLibraryPermission === false
  ) {
    return <Text>No access to camera or gallery</Text>;
  }

  return (
    <View style={{ flex: 1 }}>
      <View style={stylesHome.container}>
        <Text style={stylesHome.title}>Welcome to Camera App!</Text>
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```



```
const CameraScreen = () => {

  return (
    <View style={{ flex: 1 }}>
      <View style={stylesHome.container}>
        <Text style={stylesHome.title}>Welcome to Camera App!</Text>
      </View>

      <View style={stylesCamera.cameraContainer}>
        {isFocused && (
          <Camera
            ref={(ref) => setCamera(ref)}
            style={stylesCamera.fixedRatio}
            type={type}
            ratio={"1:1"}
          />
        )}
      </View>
      <Menu />
    </View>
  );
};

export default CameraScreen;
```

```
log
PS D:\ASSIGNMENTS\MAD\MAD QUIZ 2> 
```

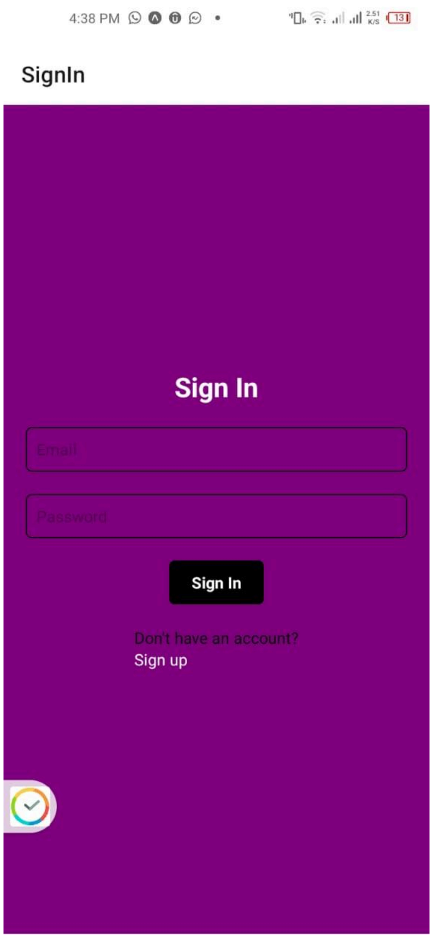
These changes were made with careful consideration of accessibility principles. For instance, the color palette now adheres to accessibility guidelines, ensuring sufficient contrast between text and background elements for users with visual impairments. Moreover, the menu structures were designed with keyboard navigation and screen reader compatibility in mind, ensuring that all users can navigate the application effortlessly.

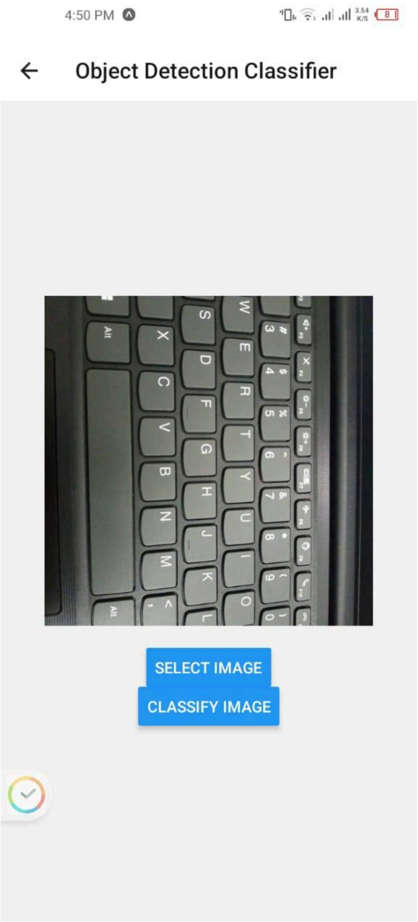
Overall, these UI enhancements aim to elevate the user experience by improving aesthetics, usability, and accessibility. By incorporating these changes, I believe the application will become more inviting and intuitive, ultimately enhancing user satisfaction and engagement.

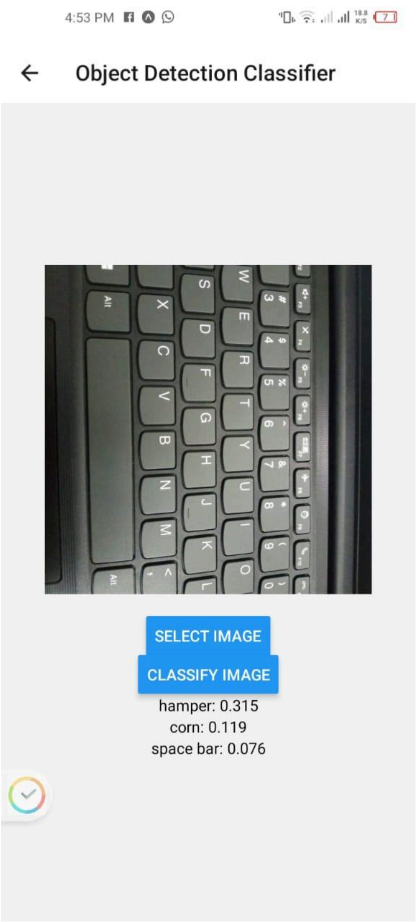
AI model Integration

The model is an object classifier it detects an object and tells about the match the three possible matches.

Output:







```
hamper: 0.315
corn: 0.119
space bar: 0.076
```