

## Objective

A mobile application using **React Native Expo** named **UserVault**. The application collects and displays personal information while validating user inputs without using a database.

---

## App Requirements

### 1. Develop a Mobile Application

The application, **UserVault**, allows users to input and display the following personal information:

- **Name**
  - **Age**
  - **Address**
  - **Profile Picture** (Upload or Select)
  - **Family Details** (e.g., parents, siblings)
  - **Educational Details** (e.g., school, university, degrees)
- 

### 2. Form Validation

Validation techniques ensure that all data entered by the user is correct:

- **Name:** Required, must not contain numbers or special characters.
  - **Age:** Must be a positive integer between 1 and 120.
  - **Address:** Required, must contain at least 10 characters.
  - **Profile Picture:** Ensure a valid image file is uploaded or selected.
  - **Family Details:** Must include at least one family member's name and relation.
  - **Educational Details:** Must include at least one educational qualification.
- 

### 3. Local Data Storage

After submission, the entered personal details are displayed on the screen.

---

## 4. User Interface Requirements

- Provide a clean and intuitive **form-based UI**.
  - Include a **summary screen** to display saved personal information.
  - Allow the user to **update or reset** their data.
- 

# Step-by-Step Development Process

## Step 1: Project Setup

Install Expo CLI and initialize the React Native project:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.26100.2605]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pasan\Documents\4.2\Practical\TIC4242 Mobile Application Development\Mobile-Application-Development-Tutorials\Tutorial8>npx create-expo-app@latest
(node:18816) ExperimentalWarning: CommonJS module C:\Program Files\nodejs\node_modules\npm\node_modules\debug\src\node.js is loading ES Module C:\Program Files\nodejs\node_modules\npm\node_modules\supports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use 'node --trace-warnings ...' to show where the warning was created)
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

? What is your app named? ... User-Vault
/ Downloaded and extracted project files.
> npm install
(node:28328) ExperimentalWarning: CommonJS module C:\Program Files\nodejs\node_modules\npm\node_modules\debug\src\node.js is loading ES Module C:\Program Files\nodejs\node_modules\npm\node_modules\supports-color\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use 'node --trace-warnings ...' to show where the warning was created)
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm warn deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard 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 rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated sudo-prompt@9.1.1: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated domexception@2.0.0: Use your platform's native DOMException instead
npm warn deprecated xmldom@0.7.13: this version is no longer supported, please update to at least 0.8.*
npm warn deprecated sudo-prompt@8.2.5: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.

added 1178 packages, and audited 1179 packages in 1m

105 packages are looking for funding
  run 'npm fund' for details
```

```
C:\Windows\System32\cmd.exe
src\index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use 'node --trace-warnings ...' to show where the warning was created)
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much
more comprehensive and powerful.
npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-tr
ansform-nullish-coalescing-operator instead.
npm WARN deprecated @babel/plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-cla
ss-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 standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-op
tional-chaining instead.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm WARN deprecated sudo-prompt@9.1.1: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm WARN deprecated domexception@4.0.0: Use your platform's native DOMException instead
npm WARN deprecated @xmldom/xmldom@0.7.13: this version is no longer supported, please update to at least 0.8.*
npm WARN deprecated sudo-prompt@8.2.5: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.

added 1178 packages, and audited 1179 packages in 1m

105 packages are looking for funding
  run `npm fund` for details

5 low severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

🟢 Your project is ready!

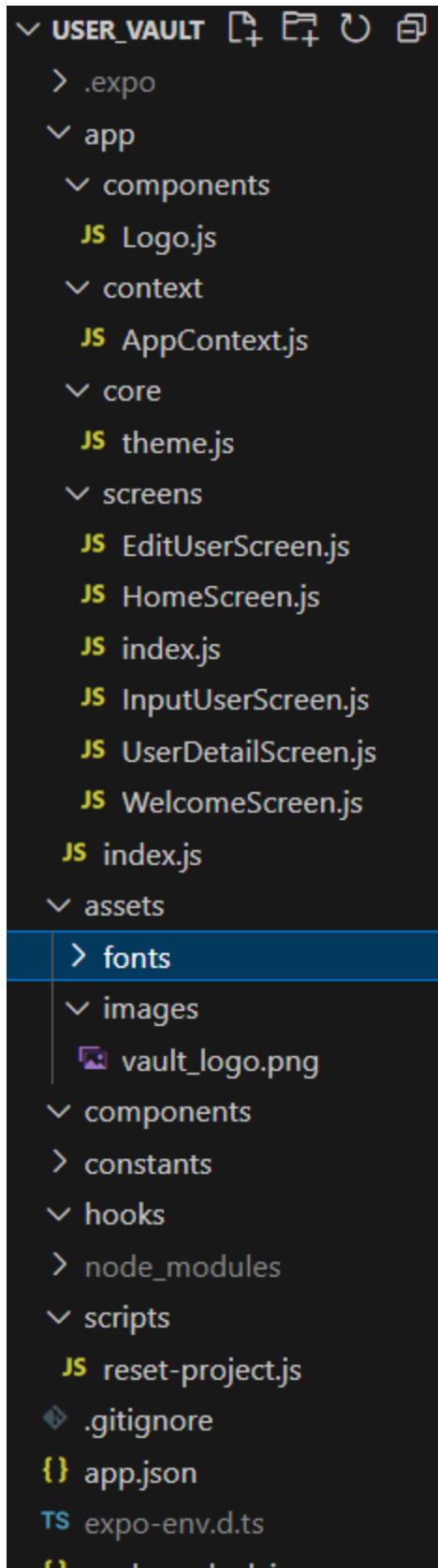
To run your project, navigate to the directory and run one of the following npm commands.

- cd User_Vault
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web

C:\Users\pasan\Documents\4.2\Practical\TICT4242 Mobile Application Development\Mobile-Application-Development-Tutorials\Tutorial8>cd User_Vault
C:\Users\pasan\Documents\4.2\Practical\TICT4242 Mobile Application Development\Mobile-Application-Development-Tutorials\Tutorial8\User_Vault>code .
```

## Step 2: File Structure Setup

- Create the necessary folders and files:



---

## Step 3: Code Implementation

### App Context Setup

Create a custom context to manage user data.

```
import React, { createContext, useContext, useState } from 'react';

// Create the context
const UserContext = createContext();

// Create a provider component
export const UserProvider = ({ children }) => {

  const [user, setUser] = useState({

    name: "",

    age: "",

    address: "",

    profilePicture: null,

    familyDetails: [],

    educationDetails: [],

  });

  const updateUser = (newData) => {

    setUser((prevUser) => ({ ...prevUser, ...newData }));

  };
}
```

```
const resetUser = () => {

  setUser({

    name: "",

    age: "",

    address: "",

    profilePicture: null,

    familyDetails: [],

    educationDetails: [],

  });

};

return (

  <UserContext.Provider value={{ user, updateUser, resetUser }}>

    {children}

  </UserContext.Provider>

);

};

// Hook to use the context

export const useUser = () => useContext(UserContext);
```

---

## Screen Development

### HomeScreen

```
import React from "react";

import { StyleSheet, Text, View, Image, Button, ScrollView,
TouchableOpacity } from "react-native";

import { useUser } from "../context/AppContext";

export default function HomeScreen({ navigation }) {

  const { user } = useUser();

  return (

    <ScrollView contentContainerStyle={styles.container}>

      <View style={styles.profileSection}>

        {user.profilePicture ? (

          <Image source={{ uri: user.profilePicture }}
style={styles.profileImage} />

        ) : (

          <View style={styles.placeholderImage}>

            <Text style={styles.placeholderText}>No Image</Text>

          </View>

        )}

        <Text style={styles.userName}>{user.name || "Name not
set"}</Text>
```

```
        <Text style={styles.userDetails}>Age: {user.age ||  
"N/A"}</Text>  
  
        <Text style={styles.userDetails}>Address: {user.address ||  
"N/A"}</Text>  
  
    </View>  
  
    <View style={styles.detailsSection}>  
  
        <Text style={styles.sectionTitle}>Family Details:</Text>  
  
        {user.familyDetails.length > 0 ? (  
  
            user.familyDetails.map((detail, index) => (  
  
                <Text key={index} style={styles.detailText}>  
  
                    - {detail}  
  
                </Text>  
  
            ))  
  
        ) : (  
  
            <Text style={styles.detailText}>No family details  
available.</Text>  
  
        )}  
  
        <Text style={styles.sectionTitle}>Education Details:</Text>  
  
        {user.educationDetails.length > 0 ? (  
  
            user.educationDetails.map((detail, index) => (  
  
                <Text key={index} style={styles.detailText}>  
  
                    - {detail}
```



```

        </Text>

        ))

    ) : (

        <Text style={styles.detailText}>No education details
available.</Text>

    )}

</View>

<View style={styles.navigationSection}>

    <View style={styles.btnwrap}>

        <TouchableOpacity

            style={styles.Button}

            onPress={() => navigation.navigate('EditUserScreen')}

        >

        <Text style={styles.buttonText}>Create User Details</Text>

        </TouchableOpacity>

    </View>

    <View style={styles.btnwrap}>

        <TouchableOpacity

            style={styles.Button}

            onPress={() => navigation.navigate('UserDetailScreen')}

        >

        <Text style={styles.buttonText}>View User Details</Text>

```

```
        </TouchableOpacity>

        </View>

    </View>

</ScrollView>

);
}

const styles = StyleSheet.create({

  container: {

    flexGrow: 1,

    padding: 20,

    backgroundColor: "#f5f5f5",

  },

  profileSection: {

    alignItems: "center",

    marginBottom: 20,

  },

  profileImage: {

    width: 100,

    height: 100,

    borderRadius: 50,

    marginBottom: 10,

  },
```

```
placeholderImage: {  
  
  width: 100,  
  
  height: 100,  
  
  borderRadius: 50,  
  
  backgroundColor: "#ddd",  
  
  justifyContent: "center",  
  
  alignItems: "center",  
  
  marginBottom: 10,  
  
},  
  
placeholderText: {  
  
  color: "#2B2C4E",  
  
},  
  
userName: {  
  
  fontSize: 20,  
  
  fontWeight: "bold",  
  
  color: "#2B2C4E",  
  
},  
  
userDetails: {  
  
  fontSize: 16,  
  
  color: "#555",  
  
},  
  
detailsSection: {  
  
  marginVertical: 20,
```

```
    },

    sectionTitle: {

      fontSize: 18,

      fontWeight: "bold",

      marginBottom: 5,

      color: "#2B2C4E",

    },

    detailText: {

      fontSize: 16,

      color: "#555",

      marginBottom: 5,

    },

    navigationSection: {

      marginTop: 20,

    },

    buttonText: {

      color: "white",

      fontSize: 16,

      fontWeight: "bold",

    },

    Button: {

      marginTop: 10,

      backgroundColor: "#2B2C4E",
```

```
paddingVertical: 10,  
  
paddingHorizontal: 10,  
  
borderRadius: 30,  
  
alignItems: "center",  
  
},  
  
));
```

## EditUserScreen

```
import React, { useState } from "react";  
  
import {  
  
  View,  
  
  Text,  
  
  TextInput,  
  
  Button,  
  
  StyleSheet,  
  
  FlatList,  
  
  TouchableOpacity,  
  
  Image,  
  
  Alert,  
  
  ScrollView,  
  
} from "react-native";  
  
import * as ImagePicker from "expo-image-picker"; // For profile  
picture selection
```

```
import { useUser } from "../context/AppContext";

export default function EditUserScreen({ navigation }) {

  const { user, updateUser } = useUser();

  const [formData, setFormData] = useState(user);

  const [newFamilyDetail, setNewFamilyDetail] = useState("");

  const [newEducationDetail, setNewEducationDetail] = useState("");

  // Profile picture picker

  const pickImage = async () => {

    let result = await ImagePicker.launchImageLibraryAsync({

      mediaTypes: ["image"],

      allowsEditing: true,

      aspect: [1, 1],

      quality: 0.5,

    });

    if (!result.canceled) {

      setFormData({ ...formData, profilePicture:
result.assets[0].uri });

    }

  };

  // Handle form submission
```

```
const handleSubmit = () => {

  if (validateForm()) {

    updateUser(formData);

    navigation.navigate("HomeScreen");

  }

};

// Form validation

const validateForm = () => {

  if (!formData.name.trim() ||
  !/^([a-zA-Z\s])+$/ .test(formData.name)) {

    Alert.alert("Validation Error", "Name must only contain
letters.");

    return false;

  }

  if (!formData.age || isNaN(formData.age) || formData.age < 1 ||
formData.age > 120) {

    Alert.alert("Validation Error", "Age must be a number between
1 and 120.");

    return false;

  }

  if (!formData.address || formData.address.length < 10) {

    Alert.alert("Validation Error", "Address must be at least 10
characters.");

    return false;

  }

}
```

```
    }

    return true;

  };

  // Handle family detail addition

  const handleAddFamilyDetail = () => {

    if (newFamilyDetail.trim()) {

      setFormData((prev) => ({

        ...prev,

        familyDetails: [...prev.familyDetails,
newFamilyDetail.trim()],

      }));

      setNewFamilyDetail("");

    }

  };

  // Handle education detail addition

  const handleAddEducationDetail = () => {

    if (newEducationDetail.trim()) {

      setFormData((prev) => ({

        ...prev,

        educationDetails: [...prev.educationDetails,
newEducationDetail.trim()],

      }));

    }

  };

}
```



```

        setNewEducationDetail("");
    }

};

return (
    <ScrollView contentContainerStyle={styles.container}>
        <View style={styles.headerContainer}>
            <TouchableOpacity
                style={styles.BackButton}
                onPress={() => navigation.navigate('HomeScreen')}
            >
                <Text style={styles.buttonText}>Back</Text>
            </TouchableOpacity>

            <Text style={styles.title}>Home</Text>
        </View>

        <Text style={styles.label}>Name:</Text>

        <TextInput
            style={styles.input}
            value={formData.name}
            placeholder="Enter your name"
            onChangeText={(text) => setFormData({ ...formData, name:
text })}
        />
    </ScrollView>
);

```

```

    { /* Age */}

    <Text style={styles.label}>Age:</Text>

    <TextInput

      style={styles.input}

      keyboardType="numeric"

      value={formData.age}

      placeholder="Enter your age"

      onChangeText={ (text) => setFormData({ ...formData, age: text
    }}}

  />

  { /* Address */}

  <Text style={styles.label}>Address:</Text>

  <TextInput

    style={styles.input}

    value={formData.address}

    placeholder="Enter your address"

    multiline={true}

    onChangeText={ (text) => setFormData({ ...formData, address:
text })}

  />

  { /* Profile Picture */}

```

```

<Text style={styles.label}>Profile Picture:</Text>

    {formData.profilePicture ? (

        <Image source={{ uri: formData.profilePicture }}
style={styles.image} />

    ) : (

        <Text style={styles.placeholder}>No profile picture
selected.</Text>

    )}

<View style={styles.btnwrap}>

    <TouchableOpacity

        style={styles.Button}

        onPress={pickImage}

    >

        <Text style={styles.buttonText}>Select Profile
Picture</Text>

    </TouchableOpacity>

</View>

{/* Family Details */}

<Text style={styles.label}>Family Details:</Text>

<FlatList

    data={formData.familyDetails}

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

    renderItem={({ item }) => <Text style={styles.listItem}>-
{item}</Text>}

```

```
/>

<View style={styles.row}>

  <TextInput

    style={styles.input1}

    value={newFamilyDetail}

    placeholder="Add Family Detail"

    onChangeText={setNewFamilyDetail}

  />

  <View style={styles.btnwrap}>

    <TouchableOpacity

      style={styles.Button1}

      onPress={handleAddFamilyDetail}

    >

      <Text style={styles.buttonText}>Add</Text>

    </TouchableOpacity>

  </View>

</View>

{/* Education Details */}

<Text style={styles.label}>Education Details:</Text>

<FlatList

  data={formData.educationDetails}

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


```

```
        renderItem={({ item }) => <Text style={styles.listItem}>-
{item}</Text>}

        />

    <View style={styles.row}>

        <TextInput

            style={styles.input1}

            value={newEducationDetail}

            placeholder="Add Education Detail"

            onChangeText={setNewEducationDetail}

        />

        <View style={styles.btnwrap}>

            <TouchableOpacity

                style={styles.Button1}

                onPress={handleAddEducationDetail}

            >

                <Text style={styles.buttonText}>Add</Text>

            </TouchableOpacity>

        </View>

    </View>

    <View style={styles.btnwrap}>

        <TouchableOpacity

            style={styles.Button1}

            onPress={handleSubmit}
```

```
        >

        <Text style={styles.buttonText}>Update</Text>

    </TouchableOpacity>

</View>

</ScrollView>

);
}

const styles = StyleSheet.create({
  container: {
    flexGrow: 1,
    padding: 20,
    backgroundColor: "#fff",
  },
  label: {
    fontSize: 16,
    fontWeight: "bold",
    marginBottom: 5,
    color: "#2B2C4E"
  },
  input: {
    borderWidth: 1,
    borderColor: "#ccc",
```

```
        borderRadius: 30,

        padding: 10,

        marginBottom: 15,

        fontSize: 14,
    },

    input1: {

        borderWidth: 1,

        borderColor: "#ccc",

        borderRadius: 30,

        padding: 10,

        marginBottom: 15,

        fontSize: 14,

        marginRight: 5
    },

    image: {

        width: 100,

        height: 100,

        borderRadius: 50,

        marginVertical: 10,

        alignSelf: "center",
    },

    placeholder: {

        fontSize: 14,
```

```
        fontStyle: "italic",

        color: "#777",

        textAlign: "center",

        marginBottom: 10,

    },

    row: {

        flexDirection: "row",

        alignItems: "center",

    },

    listItem: {

        fontSize: 14,

        marginVertical: 5,

    },

    headerContainer: {

        flexDirection: 'row',

        alignItems: 'center',

        justifyContent: 'space-between',

        paddingVertical: 10,

    },

    buttonText: {

        color: "white",

        fontSize: 16,

        fontWeight: "bold",
```



```
},

title: {

  color: "#2B2C4E",

  fontSize: 16,

  fontWeight: "bold",

},

Button: {

  backgroundColor: "#2B2C4E",

  paddingVertical: 10,

  paddingHorizontal: 10,

  borderRadius: 30,

  alignItems: "center",

},

Button1: {

  backgroundColor: "#2B2C4E",

  paddingVertical: 10,

  paddingHorizontal: 10,

  borderRadius: 30,

  alignItems: "center",

  marginBottom: 15,

},

BackButton: {

  backgroundColor: '#2B2C4E',
```

```
paddingVertical: 8,  
  
paddingHorizontal: 12,  
  
borderRadius: 5,  
  
},  
  
));
```

## UserDetailScreen

```
import React from 'react';  
  
import { View, Text, StyleSheet, Button, Image, TouchableOpacity }  
from 'react-native';  
  
import { useUser } from '../context/AppContext';  
  
export default function UserDetailScreen ({ navigation }) {  
  
  const { user, resetUser } = useUser();  
  
  return (  
  
    <View style={styles.container}>  
  
      <View style={styles.headerContainer}>  
  
        <TouchableOpacity  
  
          style={styles.BackButton}  
  
          onPress={() => navigation.navigate('HomeScreen')}  
  
        >  
  
        <Text style={styles.buttonText}>Back</Text>  
  
      </View>  
  
    </View>  
  
  );  
  
}
```

```

</TouchableOpacity>

<Text style={styles.title}>Home</Text>

</View>

<Text style={styles.title}>User Details</Text>

<Text>Name: {user.name}</Text>

<Text>Age: {user.age}</Text>

<Text>Address: {user.address}</Text>

{user.profilePicture && (
  <Image source={{ uri: user.profilePicture }}
style={styles.image} />
)}

<Text>Family Details:</Text>

{user.familyDetails.length > 0 ? (
  user.familyDetails.map((member, index) => (
    <Text key={index}>{member.name} -
{member.relation}</Text>
  ))
) : (
  <Text>No family details added.</Text>
)}

<Text>Educational Details:</Text>

{user.educationDetails.length > 0 ? (
  user.educationDetails.map((edu, index) => (

```

```

        <Text key={index}>{edu.institution} -
{edu.degree}</Text>

      ))

    ) : (

      <Text>No education details added.</Text>

    )}

    <View style={styles.btnwrap}>

      <TouchableOpacity

        style={styles.Button1}

        onPress={() =>
navigation.navigate("EditUserScreen")}

      >

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

      </TouchableOpacity>

    </View>

    <View style={styles.btnwrap}>

      <TouchableOpacity

        style={styles.Button1}

        onPress={resetUser}

      >

        <Text style={styles.buttonText}>Reset</Text>

      </TouchableOpacity>

    </View>

```

```
    </View>

  );
};

const styles = StyleSheet.create({
  container: { padding: 20, },
  title: { fontSize: 18, fontWeight: "bold", marginBottom: 10 },
  image: { width: 100, height: 100, marginVertical: 10 },
  headerContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between',
    paddingVertical: 10,
  },
  BackButton: {
    backgroundColor: '#2B2C4E',
    paddingVertical: 8,
    paddingHorizontal: 12,
    borderRadius: 5,
  },
  buttonText: {
    color: "white",
```

```
      fontSize: 16,

      fontWeight: "bold",

    },

    title: {

      color: "#2B2C4E",

      fontSize: 16,

      fontWeight: "bold",

    },

    Button1: {

      backgroundColor: "#2B2C4E",

      paddingVertical: 10,

      paddingHorizontal: 10,

      borderRadius: 30,

      alignItems: "center",

      marginBottom: 15,

    },

  });
```

---

## Step 4: Form Validation

```
// Form validation

const validateForm = () => {
```

```

        if (!formData.name.trim() ||
!/^[a-zA-Z\s]+$/.test(formData.name)) {

            Alert.alert("Validation Error", "Name must only contain
letters.");

            return false;

        }

        if (!formData.age || isNaN(formData.age) || formData.age < 1 ||
formData.age > 120) {

            Alert.alert("Validation Error", "Age must be a number between
1 and 120.");

            return false;

        }

        if (!formData.address || formData.address.length < 10) {

            Alert.alert("Validation Error", "Address must be at least 10
characters.");

            return false;

        }

        return true;

    };

```

---

## Step 5: Local Data Storage and State Management

```

import React, { createContext, useContext, useState } from 'react';

// Create the context

```

```
const UserContext = createContext();

// Create a provider component

export const UserProvider = ({ children }) => {

  const [user, setUser] = useState({

    name: "",

    age: "",

    address: "",

    profilePicture: null,

    familyDetails: [],

    educationDetails: [],

  });

  const updateUser = (newData) => {

    setUser((prevUser) => ({ ...prevUser, ...newData }));

  };

  const resetUser = () => {

    setUser({

      name: "",

      age: "",

      address: "",

      profilePicture: null,
```



```

        familyDetails: [],

        educationDetails: [],

    });

};

return (

    <UserContext.Provider value={{ user, updateUser, resetUser }}>

        {children}

    </UserContext.Provider>

);

};

// Hook to use the context

export const useUser = () => useContext(UserContext);

```

```

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'

```

```
import React from "react";

import { Provider } from "react-native-paper";

import { NavigationContainer } from "@react-navigation/native";

import { createStackNavigator } from "@react-navigation/stack";

import { theme } from "../core/theme";

import { UserProvider } from "../context/AppContext";

import { WelcomeScreen , HomeScreen , EditUserScreen , UserDetailsScreen ,
InputUserScreen} from "../screens";

const Stack = createStackNavigator();

export default function App() {

  return (

    <UserProvider>

      <Provider theme={theme}>

        <Stack.Navigator

          initialRouteName="WelcomeScreen"

          options={{ headerShown: false}}

        >

          <Stack.Screen name="WelcomeScreen" component={WelcomeScreen}
options={{ headerShown: false }}/>
```

```

        <Stack.Screen name="HomeScreen" component={HomeScreen} options={{
headerShown: false }}/>

        <Stack.Screen name="EditUserScreen" component={EditUserScreen}
options={{ headerShown: false }}/>

        <Stack.Screen name="UserDetailScreen" component={UserDetailScreen}
options={{ headerShown: false }}/>

        { /* <Stack.Screen name="InputUserScreen"
component={InputUserScreen} options={{ headerShown: false }}/> */ }

    </Stack.Navigator>

    </Provider>

    </UserProvider>

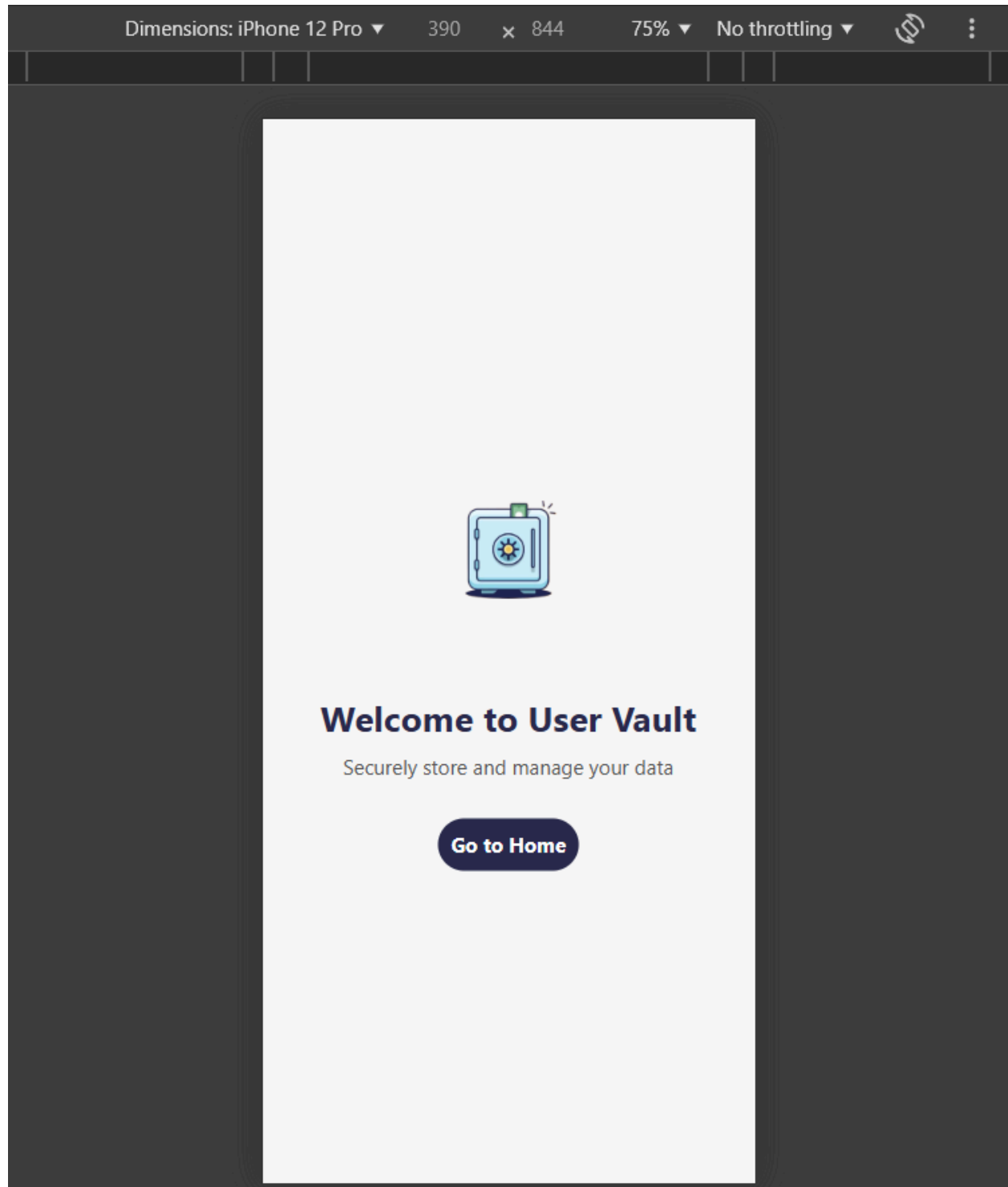
);
}

```

## Step 6: Testing the App

- Input valid details for all fields and take a screenshot of the final UI.
  - Input invalid details (e.g., empty fields, invalid name/age) and screenshot the validation alerts.
-

# App UI



Dimensions: iPhone 12 Pro ▼

390

× 844

75% ▼

No throttling ▼



No Image

### **Name not set**

Age: N/A

Address: N/A

#### **Family Details:**

No family details available.

#### **Education Details:**

No education details available.

**Create User Details**

**View User Details**

[Back](#)[Home](#)**Name:****Age:****Address:****Profile Picture:***No profile picture selected.*[Select Profile Picture](#)**Family Details:**[Add](#)**Education Details:**[Add](#)[Update](#)

Dimensions: iPhone 12 Pro ▼

390



844

75% ▼

No throttling ▼



**Back**

**Home**

### **User Details**

Name:

Age:

Address:

Family Details:

No family details added.

Educational Details:

No education details added.

**Edit**

**Reset**

[Back](#)[Home](#)**Name:**

Pasan Madhuranga

**Age:**

26

**Address:**

No.24/17, Temple Road, Weralupa, Rathnapura.

**Profile Picture:**[Select Profile Picture](#)**Family Details:**

- Mother

- Father

[Add Family Detail](#)[Add](#)**Education Details:**

- GCE A/L &amp; O/L

- BICT(Hons)

[Add Education Detail](#)[Add](#)[Update](#)





## Pasan Madhuranga

Age: 26

Address: No.24/17, Temple Road, Weralupa,  
Rathnapura.

### Family Details:

- Mother
- Father

### Education Details:

- GCE A/L & O/L
- BICT(Hons)

Create User Details

View User Details