

# Mobile App Development Practice

## Project: “Simple Profile App”

This practice will help a beginner understand:

- Project setup
- Folder structure
- Creating screens
- Using JSX (HTML-like syntax)
- Styling components
- Navigation
- Handling basic state

## Prerequisites

- Node.js installed
- VS Code installed
- React Native CLI or Expo (we will use **Expo for beginners**)

## STEP 1 — Create the Project

```
npm install -g expo-cli  
expo init ProfileApp  
cd ProfileApp  
expo start
```

Choose **blank template**.

## STEP 2 — Understand the HTML-like JSX Structure

React Native doesn't use `<div>` or `<span>` — instead, it uses similar components:

HTML	React Native
<code>&lt;div&gt;</code>	<code>&lt;View&gt;</code>
<code>&lt;p&gt;</code>	<code>&lt;Text&gt;</code>
<code>&lt;img&gt;</code>	<code>&lt;Image&gt;</code>
<code>&lt;button&gt;</code>	<code>&lt;Button&gt;</code>

Create a file called **ProfileCard.js** inside `components/`.

### **components/ProfileCard.js**

```
import React from 'react';
```

```

import { View, Text, Image, StyleSheet } from 'react-native';

export default function ProfileCard() {
  return (
    <View style={styles.card}>
      <Image
        source={{ uri: 'https://placekitten.com/200/200' }}
        style={styles.avatar}
      />
      <Text style={styles.name}>John Doe</Text>
      <Text style={styles.bio}>React Native Beginner Developer</Text>
    </View>
  );
}

const styles = StyleSheet.create({
  card: {
    alignItems: 'center',
    padding: 20,
    backgroundColor: '#fff',
    borderRadius: 10,
    elevation: 3
  },
  avatar: {
    width: 100,
    height: 100,
    borderRadius: 50
  },
  name: {
    fontSize: 22,
    fontWeight: 'bold',
    marginTop: 10
  },
  bio: {
    fontSize: 16,
    color: '#555',
    marginTop: 5
  }
});

```

## STEP 3 — Use Your Component in App.js

Open **App.js** and replace content:

```

import React from 'react';
import { View, StyleSheet } from 'react-native';
import ProfileCard from './components/ProfileCard';

export default function App() {
  return (
    <View style={styles.container}>

```

```

        <ProfileCard />
    </View>
);
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#f1f1f1'
  }
});

```

This acts like an HTML page including your “profile card”.

## STEP 4 — Add a Button to Change the Bio (Simple State)

Modify **ProfileCard.js**:

```

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

export default function ProfileCard() {
  const [bio, setBio] = useState("React Native Beginner Developer");

  return (
    <View style={styles.card}>
      <Image
        source={{ uri: 'https://placekitten.com/200/200' }}
        style={styles.avatar}
      />
      <Text style={styles.name}>John Doe</Text>
      <Text style={styles.bio}>{bio}</Text>

      <Button
        title="Change Bio"
        onPress={() => setBio("Learning React Native Step by Step!" )}
      />
    </View>
  );
}

```

Now the user can press a button, and the text updates.

## STEP 5 — Add Multiple Screens (Optional Advanced Practice)

Install navigation:

```
npm install @react-navigation/native  
npm install @react-navigation/native-stack  
expo install react-native-screens react-native-safe-area-context
```

Create:

**screens/HomeScreen.js**  
**screens/ProfileScreen.js**

Then set up navigation in **App.js**.

## STEP 6 — Practice Tasks (For the Learner)

Give the beginner these tasks:

### Easy Tasks

- Change the profile picture
- Change the name
- Add background color to the card

### Intermediate Tasks

- Add an “Edit Profile” screen
- Add input fields for name and bio
- Add a button to toggle dark mode

### Advanced Tasks

- Store profile data in AsyncStorage
- Fetch profile data from an API
- Build a list of multiple profile cards

## STEP 7 — Create an Expo Account

1. Go to **expo.dev**
2. Click **Sign Up**
3. Create an account using Google, GitHub, or email
4. Verify email (optional but recommended)

**Sign in from your terminal:**

```
expo login
```

You will be asked for:

- Username
- Password

Once logged in, your terminal will show:

```
Logged in as <your-username>
```

## STEP 8 — Connect Your Project to Expo Account

In your project folder run:

```
expo whoami
```

If you see your username → good.

If not logged in:

```
expo login
```

## STEP 9 — Prepare the Project for Build

Expo requires an **app.json** file.

Open app.json and update:

```
{
  "expo": {
    "name": "ProfileApp",
    "slug": "profile-app",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "android": {
      "package": "com.yourusername.profileapp"
    }
  }
}
```

**Important:**

Change "com.yourusername.profileapp" to your own bundle ID.

## STEP 10 — Generate an APK File (Android)

Expo uses **EAS Build**.

Install EAS CLI:

```
npm install -g eas-cli
```

Login:

```
eas login
```

Initialize build:

```
eas build:configure
```

Now build APK:

```
eas build -p android --profile preview
```

To generate a "universal APK" run:

```
eas build -p android --profile production
```

**You will get a link after the build completes:**

Example:

Build completed: <https://expo.dev/accounts/<yourname>/projects/<project>/builds/<id>>

Click the link → download APK.

## STEP 11 — Install the APK on an Android Phone

3 methods:

### Method 1 — Direct download

1. Send the APK file to your phone (email, WhatsApp, Google Drive).
2. Open and install.
3. If you get “unknown apps not allowed”:
  - o Settings → Security → Install unknown apps → Allow for your file manager.

## **Method 2 — USB install**

1. Connect phone to PC
2. Enable "Install via USB"
3. Drag APK to phone and open

## **Method 3 — Using ADB**

Requires Android SDK:

```
adb install your-app.apk
```

## **STEP 12 — Publish Your App to Expo**

(Optional but useful for beginners)

Run:

```
expo publish
```

This generates a URL like:

```
https://expo.dev/@username/profile-app
```

You can open this link and scan QR code to run your app instantly.

## **STEP 13 — Best Practice for Beginners**

### **✓ Use Expo Go app (fastest)**

- Install Expo Go from Play Store / App Store
- Run:

```
expo start
```

- Scan QR code
- Your app appears instantly

### **✓ When to build APK?**

Only when:

- You want to share an installable file
- You want to upload to Play Store

## Full Flow Summary

Step	Action
1	Create Expo account
2	Login in terminal
3	Prepare app.json
4	Install EAS CLI
5	Run eas build
6	Download APK
7	Install on Android