

CPAN213 – Cross-Platform Mobile App Development

Assignment 2 (5%)

Copyright Policy

This assessment contains materials that are subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere such as CourseHero, OneClass, Chegg, etc. will be subject to the college's **Copyright policy and Academic Integrity policy**.

Academic Integrity

This assessment is to be done individually.

What is allowed:

- Looking up **syntax** related to JavaScript, React, React Native
- Accessing documents of any libraries required to use in assignment
- You can refer the code and classwork created in this course

What is **NOT** allowed:

- Searching for partial or full solutions of the main problem description
- Communication with others, either inside or outside the class
- Sharing of resources, including but not limited to code, links, computers, etc.

Submission Checklist:

1. **Screenshot(s)** of your mobile app showing the form and displayed information. You may add all the screenshots in word document and upload it. ***[Do not put screenshots in project/zip file]***.
2. **Delete the node_modules** folder as it has many files which are big in size and is not required for the submission. Create **a zip file** of your entire project folder and submit.
3. A **screen-recording** of your app execution which should demonstrate all the functionalities and/or errors if any. This is to ensure that the app works well on the device you tested.

Task Description:

Using **expo CLI**, create a new project named as **a2_your_name** (such as a2_john_doe)

You are required to create a multi-screen ReactNative application. The application will allow the user to

- sign in,
- sign out,
- visit dashboard,
- book appointment and
- access the booking information.

The app should present **SignIn screen** first. You must use “admin” and “admin” as username and password respectively. Once user successfully sign-in, navigate to Dashboard screen.

On the **Dashboard screen**, the tool bar should present logout button. When user presses “Logout” button, the app should navigate to the first screen they started navigation from (Login screen).

The **Dashboard screen** should allow the user to book a study room on campus. It should ask user to input student ID, name and number of people. Also, ask them to select a room number. You may choose a list of 4-5 room numbers such as A101, A102, A103, A104 and A105. Present “Check Availability” button on dashboard.

Validate the data on all screens as necessary.

Once user inputs all the details and presses “Check Availability” button, navigate to “Booking” screen.

The **Booking screen** should have an array with hard-coded room information. For example,

```
[
  {roomNumber : A101, capacity : 5, available: true},
  {roomNumber : A102, capacity : 10, available: false},
  {roomNumber : A103, capacity : 8, available: false},
  {roomNumber : A104, capacity : 10, available: true},
  {roomNumber : A105, capacity : 7, available: true}
]
```

Once the Booking screen receives data from Dashboard, it should check if the selected room is available and if is enough for selected number of people. Inform the user if the room is available or not.

Navigation and Routing:

- Use React Navigation library to implement multi-screen navigation within the application as practiced in the class.
- Configure stack navigation for navigating between different screens, ensuring a seamless user flow.

Styling and UI Design:

- Apply appropriate styling and design principles to create an intuitive and visually appealing user interface.
- Ensure consistency in design elements, typography, colors, and layout across all screens.

Implementation Guidelines

- Code must be written in JavaScript (not Typescript or any other language)
- You must use function-based components, not class-based components.
- Variables must be declared using let and const. No var declarations permitted.
- Functions should be declared using arrow function syntax, example: `const abc = () => {}`