Habit Tracker App

Project Overview:

In this project you will be making your own multi-screen Android app to share your knowledge about a city you are very familiar with.

The goal is to create a Habit Tracker App which presents relevant information to a user who's visiting your city. The app can list top attractions, restaurants, public places, or events for the city. It can contain all the best known secrets that only locals know. It's up to you which categories you want to provide, as well as what information to provide in each category.

Why this project?

In the most recent portion of the Diploma program, you worked to build the Miwok language app, which taught you how to present information in a multi-screen app to your users. Now, you will practice that skillset by presenting information that you care about in your own life to a user. Your Habit Tracker App will be unique to your city, your life, and your interests.

What will I learn?

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Planning your app design and navigation before coding.
- Selecting proper data structures to store lists of information.
- Building layouts to display those lists of data.
- Navigating between those lists using intents and multiple Activities or a ViewPager.
- Creating your own custom classes.

- Properly handling images or audio (if applicable).

Build Your Project:

For this project, you'll be making a Habit Tracker App. This is an app with multiple screens, each of which lists a kind of attraction for a given city, national park, or historical monument. For instance, an app focused on Mountain View, CA might include a list of startup headquarters as well as a list of restaurants and a list of parks.

Once you've decided on the location you'd like to show off, you'll want to build the structure of your app. Using the skills from the Miwok app, you'll make multiple screens, each containing a list. You may choose to use explicit intents to move between activities, or you may choose to use a ViewPager combined with fragments.

Next, you'll fill in those lists with list items, populated using a custom adapter. It may make sense for some lists to include information such as a phone number or business hours, but for others it may make more sense to have a photograph. Design your adapter based on the information you would like to show.

Your project will be evaluated using the Tour Guide project rubric shown below.

Project Rubric:

Content Review

CRITERIA	MEETS SPECIFICATIONS
Overall Layout	App contains at least 4 lists of relevant attractions for a location
Navigation	User can navigate between lists using a central screen, a

	NavDrawer, or a View pager.
List Item contents	Each list item contains information about an event, restaurant, historical site, or similar.
Pictures	At least one list includes pictures of the location.
Layout Best practices	The code adheres to all of the following best practices: ● Text sizes are defined in sp ● Lengths are defined in dp ● Padding and margin is used appropriately, such tha the views are not crammed up against each other.

Functionality

CRITERIA	MEETS SPECIFICATIONS
Location Object	App contains a custom object for storing location informatio .
Custom Adapter	App uses a custom adapter to populate the layout with views based on instances of the custom class.
String Storage	All strings are stored in the strings.xml resource file.
Image Storage	All images are stored as drawables. All drawables are stored at multiple densities.
Errors	The code runs without errors.

Code Readability

CRITERIA	MEETS SPECIFICATIONS
Readability	Code is easily readable such that a fellow programmer can understand the purpose of the app.
Naming conventions	All variables, methods, and resource IDs are descriptively named such that another developer reading the code can easily understand their function.
Formatting	The code is properly formatted i.e. there are no unnecessary blank lines; there are no unused variables or methods; there is no commented out code. The code also has proper indentation when defining variables and methods.

Prepare for Submission:

Clean Your Build

Before submitting, please follow the instructions for cleaning your project files. This removes some temporary files and greatly decreases the size of your project.

Clean Your Project Files

Final Submission Checklist

Before submitting your project for evaluation, we recommend that you check that each of the following is true:

1. Your app compiles and runs as expected.
2. You are proud of your app and its output.
3. You completed this project according to instructions.
4. You cleaned the project using the instructions above.
5. You checked your project against the rubric.