Many different platforms are available for developing mobile apps that require limited or no coding. This is usually done using pre-developed objects known as components, coupled with some block programming.

The Platform used for this activity is **Thunkable**. This service has a free (public) app development section and a paid (private) app development section. We will be utilizing the free (public) app development section. If you have an Android or iOS mobile device you can download their live app viewer app from the respective app store. If you do not have a mobile device do not worry as you can live preview your app directly in the web browser for most functions. The Map component you will be adding to this app however, will only display the items other than the Map as the Map requires either the Android or iOS environment.

We will be developing a mobile application keeping with our camp theme to work as a Tourist App. The app will consist of a background image, and several buttons on the main screen. The buttons will trigger points of interest to show up on a map contained on the second screen.

This app demonstrates the concept of **Availability**. The layout should be easy to use for everyone utilizing the concept of **Keep It Simple**.

**Section: Account Creation / Access:**

If you have not created an account yet please go to: <https://x.thunkable.com/signup> , if you have created an account you can skip this section.

You will need to enter your email address into the box with the placeholder text “Enter your email” and then click the button “Email me the link”.

Check your email and you should receive a link to click which will log you into the site on your device. If you do not click the logout button afterwards it should keep you logged in for 30 days and you can just revisit the site to continue working.

**Section: Account Access:**

If when you visit the site again you find yourself logged out you will need to go to <https://x.thunkable.com/login> and once again provide your email address and click “Email me the link”. Then log into your email and click the link to be automatically logged into the site.

**Section: Getting Started:**

If you have never used Thunkable, a button such as the one below will be visible:

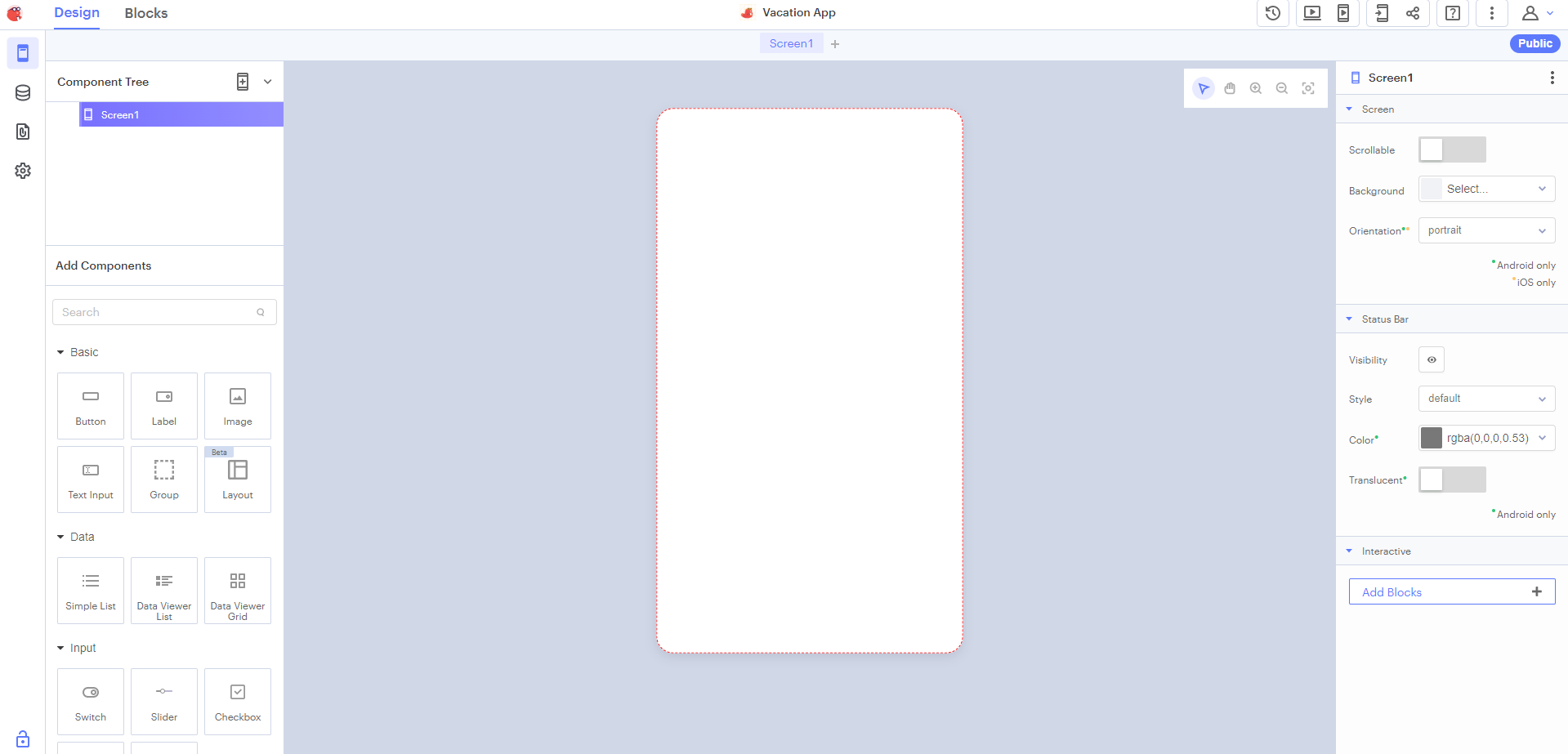
A picture containing text, screenshot, line, font

Description automatically generated

|  |  |  |
| --- | --- | --- |
| 1. Otherwise the button to the right will be visible. Click the one you have. 2. Fill in the Form with the Project name being Vacation App and the Category being Entertainment. 3. Then Click the “Create” button 4. If Visible, Hide the tutorials section by clicking the purple tab with the left arrow. |  | |
|  | |  |

**Section: The Design Process:**

1. At this point, your screen should look like the one below:



On the left at the top you will see 2 tabs, named “Design” and “Blocks”. To begin, we will be working on the design of the app. Make sure the “Design” tab is selected. Below that is a section for components that are currently part of your app and further below that are components you can drag onto your app.

On the right will be the individual properties for the current selected component. Initially it is showing the properties of the Screen1 component.

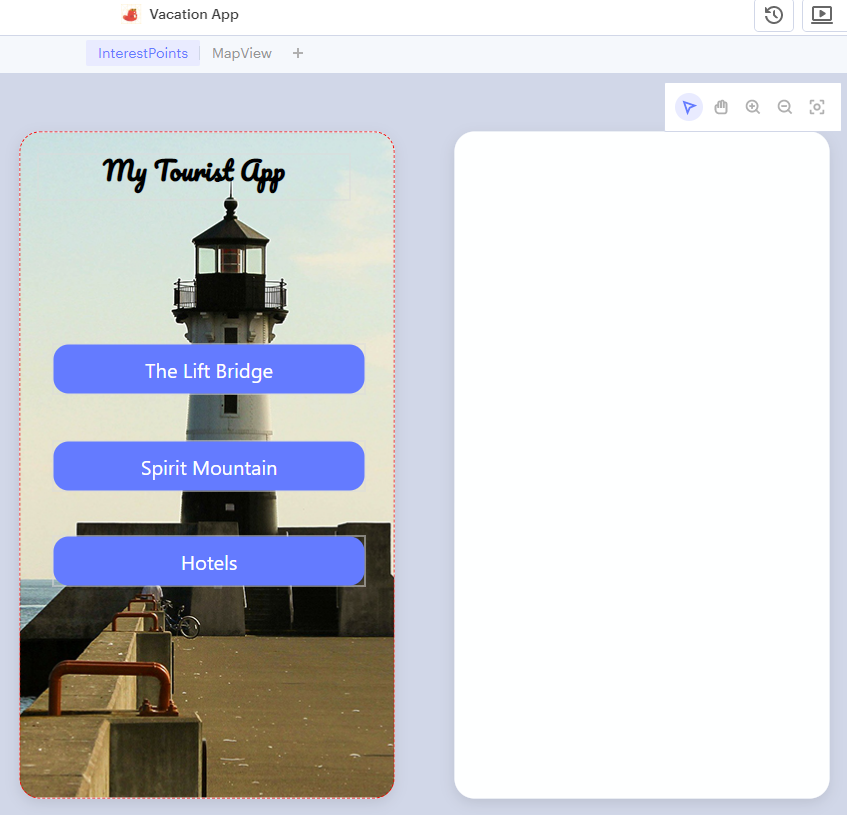
1. This program will consist of 2 main Screens. The first will contain an image for a background and 3 buttons. These buttons will represent points of interest in the town. You will find the image in this walkthrough supplied on your flash drive. (lighthouse.jpg)
2. First you will work on Screen1. Change the name of the component to something more unique such as “InterestPoints”. To do this make sure the Screen1 component is selected, then on the right-side pane there should be a pencil icon or 3 stacked dots. Click either the pencil and apply the name change, or If the pencil is missing, click the 3 dots and then click Rename.

1. Now, add the image to the background. While InterestPoints is still selected, select “background” then “Image”, then click inside of the text box that says: “No file source”. Then upload the supplied lighthouse.jpg image.
2. Next, drag a label to the screen. Position it at the top of the application. Replace the Text with “My Tourist App”, change the font size to 26, font-weight to bold, and text align of center with a font of your choice.

|  |  |
| --- | --- |
| 6. Drag 3 button components to the screen and arrange them to match this example. | 7. Select the first button and change the text to “The Lift Bridge”, the font-size to 20 and then choose a background color of your choice. Apply the same changes to the other two buttons with the text as “Spirit Mountain” and then “Hotels”. Your screen should look like the one below: |

8. Before we start the Code Blocks for this screen, we need to create another screen to the designer. This screen will be what the program navigates to after a button is clicked. Click the + sign next to your InterestPoints Screen name in the center of the designer window. Rename this new screen to MapView and then click the InterestPoints Screen to return to it.



9. Time for the Code Blocks. We need to initialize 5 variables and provide some default values. To open the block coding space, clikc the “Blocks” tab at the top of the project window. In the left pane select the “Variables” category and then drag 5 of the top blocks to the code space. Change the names of the variables to the following variable names: **longg**, **latt**, hotels, desc, and title. Under the Math category drag the default number block (first block) to the first 3 variables. Leave hotels set to 0 but change the value for **latt** to 46.7841038 and **longg** to -92.1026938. Under the Text category, drag 2 empty text blocks to the remaining variables. These will be the variables that our buttons will change, which will then be used on the MapView Screen to show the locations.

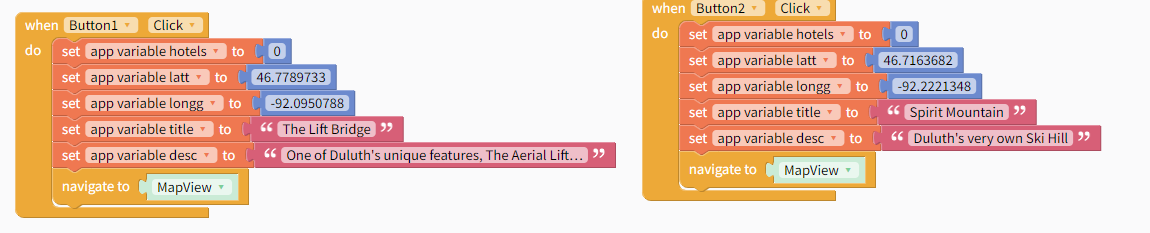


1. Next, we will add the 3 event listeners for our button clicks. Select Button1 from the left pane and then drag 3 of the “When Button1 Click, do” blocks to the screen. We will just change the Button name by using the drop down. Change the 2nd and 3rd button click names to Button2 and Button3. In the Variables category, drag one of the “set app (variable) to” blocks to EACH one of the When/Do blocks. Change the variable names all to “hotels”. Then using CTRL-C and CTRL-V, copy the math block containing the value 0 from where the hotels variable was initialized and drag them to each of the set blocks. Change the value of Button3’s hotels variable to 1. Finally, for this step, select the Controls category and drag a navigate to block to each one of the When/Do blocks and make sure MapView is selected in the dropdown. Your blocks should now look like the ones below:

|  |  |  |
| --- | --- | --- |
| A picture containing text, font, screenshot, number  Description automatically generated | A picture containing text, font, screenshot, number  Description automatically generated |  |

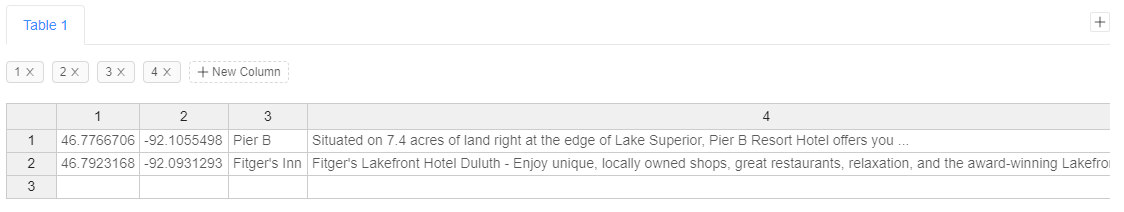
1. Now we need to have Button1 and Button2 change the values of our variables BEFORE navigating to MapView. In the same way you dragged the “Set app (variable) to” Block above for hotels, drag 5 more to the Button1 and Button2 click blocks changing the variables to the following values:

|  |  |
| --- | --- |
| Button1 | Button2 |
| latt = 46.7789733 | latt = 46.7163682 |
| longg = -92.0950788 | longg = -92.2221348 |
| title = The Lift Bridge! | title = Spirit Mountain |
| desc = One of Duluth's unique features, the Aerial lift bridge | desc = Duluth’s very own Ski Hill |



That’s it for the **InterestPoints** Code Blocks, Click the Design tab to go back to the Design View.

1. The final component we will add to our **InterestPoints** screen will be a Local DB (database) component. To create one, you must first click the icon on the left-hand side of the screen that looks like a bunch of stacked pancakes. A new pane shows with the word Data Sources and a plus sign. Click the plus sign, then click the “Create New” button. Select the first option which should be the Table. Name the Table, **appData**, then click create. Make sure you have 4 columns and then enter the data as seen in the image below (Note you can double click in a table cell to enter the data:



The last column doesn’t have to be exact, it is just an example of how the description will be used. If the image is too small, the text below is what is displayed there:

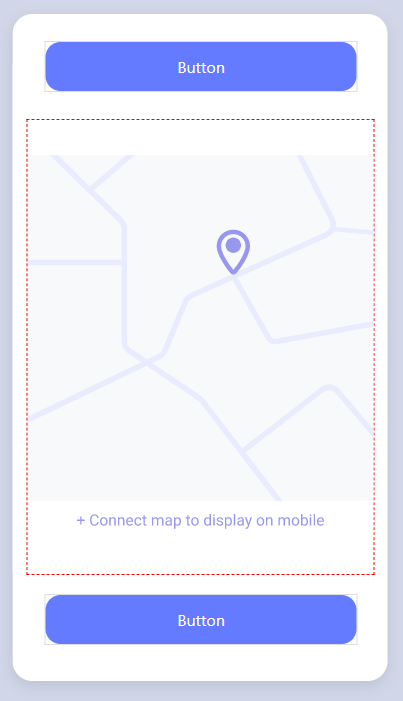
* Situated on 7.4 acres of land right at the edge of Lake Superior, Pier B Resort Hotel offers you ...
* Fitger's Lakefront Hotel Duluth - Enjoy unique, locally owned shops, great restaurants, relaxation, and the award-winning Lakefront Hotel in Duluth.

After the data is entered, you are now complete with the InterestPoints screen.

\*\* Take a break \*\*

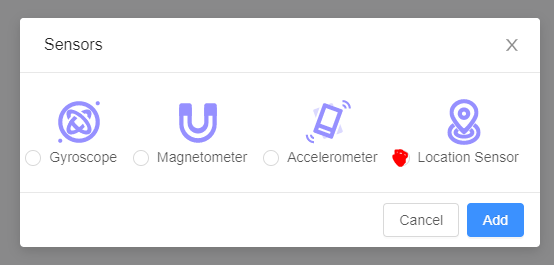
**Section: MapView:**

1. The MapView screen design is much simpler. This will be where markers are displayed on a map. The screen will also have a “Find Me” button and a “Go Back” button to return to the previous screen. Drag 2 **Buttons** as well as a **Map** component to the MapView Screen. Similar to this:

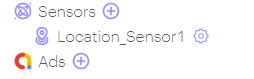


|  |  |
| --- | --- |
| 2. Change the following properties of the components you just dragged onto the screen.   * Rename the top button to **btnBack** * Change the Text to “**Go Back**” * Set font to 20 and the B icon for Bold. | 3. For the Bottom Button:   * Change the name to **btnFindMe** * Change Text to “**Find Me**”. * Set the font to 20 the B icon for Bold |

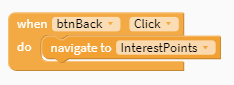
1. The Last component we need to add to this screen is a Location Sensor. Click the **Blocks** tab, then select the **plus-sign** next to the word **Sensors** which will be located under App Features towards the bottom on the left-hand pane. Then in the window that pops up, click location sensor.



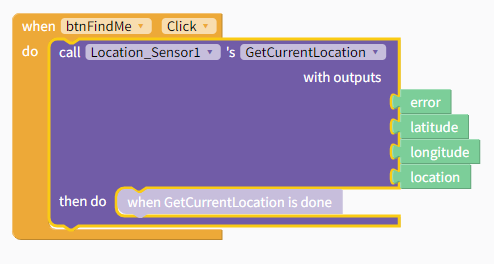
After clicking Add, just use the default settings provided. It should now be added to your sensors list on the left-hand pane.



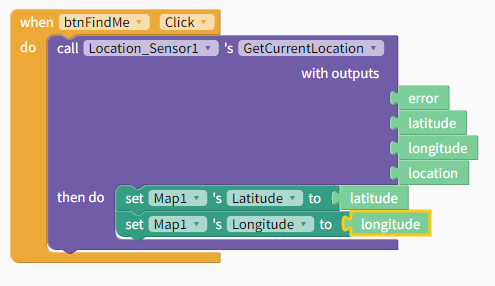
1. Select the **Blocks** tab and click **btnBack** under UI Components. Drag the “When btnBack Click do” block to the screen. Then drag a “navigate to” block from the Control category and make sure it says to navigate to InterestPoints.



1. Drag another “**When/Do**” block for another click event, but this time it will be attached to the **btnFindMe** Button component. Click on **Location\_Sensor1** on the left-hand panel and drag a “call Location\_Sensor1’s GetCurrentLocation” block to the DO section of the When/Do block.

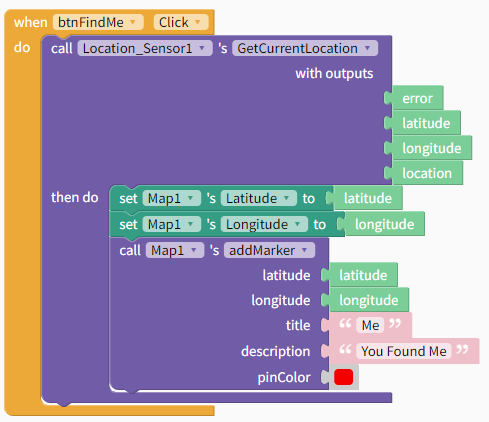


1. Select Map1 on the left panel, then drag 2 of the “set Map1’s (property) to” blocks into the “then do” spot of the Location Sensor. Make sure the top one is set to **Latitude** and the bottom one is set to **Longitude**. Then drag the corresponding green blocks from the Location Sensor block to the appropriate areas.



Now the button will find your location and center the map on it when clicked. Let’s also add a Marker to that position designating our location.

1. Select Map1 from the left and drag out a “call Map1’s addMarker” block. Place it directly below the 2 “set” Map1 blocks we added. Again, drag the latitude and longitude blocks to the corresponding spots. Set the title to “Me” and the description to “You Found Me”. Now when you click on this marker you will get that data.

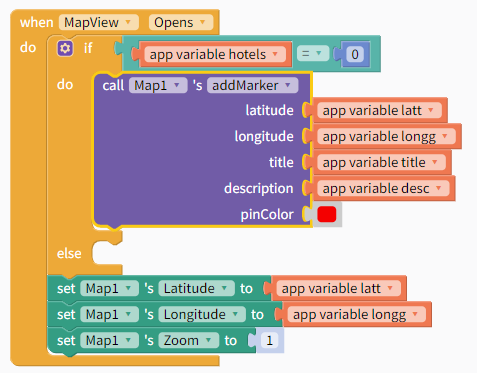


1. One last code block to write, and then this app is finished. This block however, is much longer than the rest. Start out by clicking MapView on the left, then drag a “When MapView Opens do” block and 2 “set Map1’s (property) to” blocks to the **do** section of the **When/Do** block. This time, to get the latitude and longitude, you will need to click Variables on the left and drag the “app latt” variable and the “app longg” variable to the correct spots. Next drag a “set Map1’s Zoom to” block and set the value to 1. Under the Control category, drag an if-do-else block and place it above the lattitude block. Under the Logic category, drag the first block (the equals block) to the IF spot of the if-do-else block. Grab the “app hotels” variable block under Variables and place it in the first spot of the equals block, while placing an integer block with value of 0 in the second open spot.

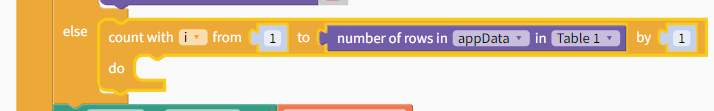
A picture containing text, screenshot, font, number

Description automatically generated

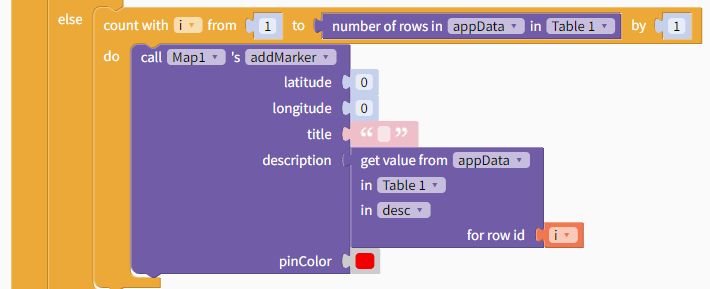
1. For the Do spot, drag an addMarker block to the screen and fill in the latitude, longitude, title and description spots using the corresponding blocks found in the Variables category.

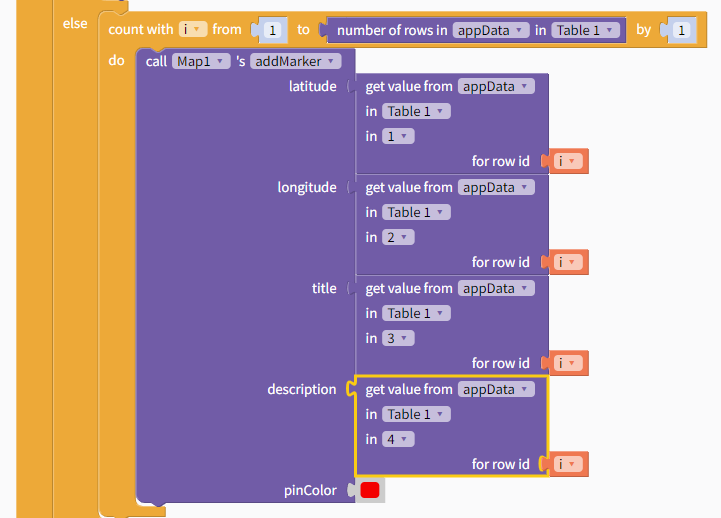


1. Now we need to handle the IF app hotels equals 1, which means it is from the Hotel link. The reason we are doing this, is multiple items will be set from this link. Remember the data we entered in the data table component? That is what we will be using to populate this next block. To do this, we must iterate through EVERY item in the table. For that we will need to click the Control category and drag the block that says: “count with I from 1 to 10 by 1 do”. Replace the number 10 with a block from the Data Sources category on the left side. The block we want is “number of rows in appData in Table 1”.

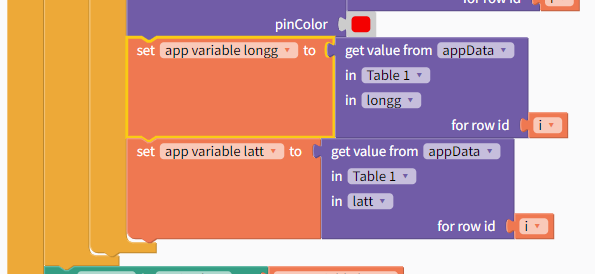


1. Now drag an “addMarker” block to the Do spot of the Count block.



1. Next, we must replace the properties in the addMarker block. To do this, click on Data Sources and select the first block which should be “get value from appData in Table 1 in 1…” the “in 1” spot is the column you are looking in. Had we made this more descriptive when creating our table this would make more sense. For now, 1 is latitude, 2 is longitude, 3 is title and 4 will be the description. You will find the counter variable “i” under the variable’s category. Make sure you finished block looks like this:

14. Lastly, at the end of the loop, set the variables long and latt to values of the item from the table. In this visual example, the numbers of the columns have been replaced with the strings “latt” and “longg” but either way will work (1, 2 or latt, long).



Now you can click the “Live Test” icon at the top of the screen. The Map portion will not be displayed in the web view so you must have the Thunkable live test app on a mobile device installed in order to get the full feeling of the app.

**Section: Challenges:**

1. Using google maps, find 4 more points of interest in the Duluth area and add them to the app. You will need the Latitude and Longitude of these places which can be found in the URL when you look something up on google maps.



1. Add another Table component. Look up 4 different restaurants using the method above and just like what was done for the Hotels, add the data for the restaurants to the new DB. Then figure out how to get a “Restaurants” button to display all 4 locations.

**Section: Additional Information:**

Thunkable Docs: <https://docs.thunkable.com/get-started/>