



جامعة الفيصل  
Alfaisal University

SE 328 L: Mobile Development Lab  
CoE – Software Engineering  
Alfaisal University

Lab 9: Machine Learning & Object Detection

Course Lecturer: Dr. George Violettas  
Spring Semester 2022

Lab Instructor: Eng. Hoda Elsayed

Expected Time: 2 hrs

Student Name: Mohammed AlAlem

### Objectives

The objective of this lab is to:

- Getting introduced to a MIT App Inventor Tool
- Merging Machine Learning concepts with mobile development
- Learn about object detection and Look extension for neural network processing

Learning Outcomes	Ex #
CLO3. Use Audio and other objects, other Android layout tools (SO1)	1

### Submission Style:

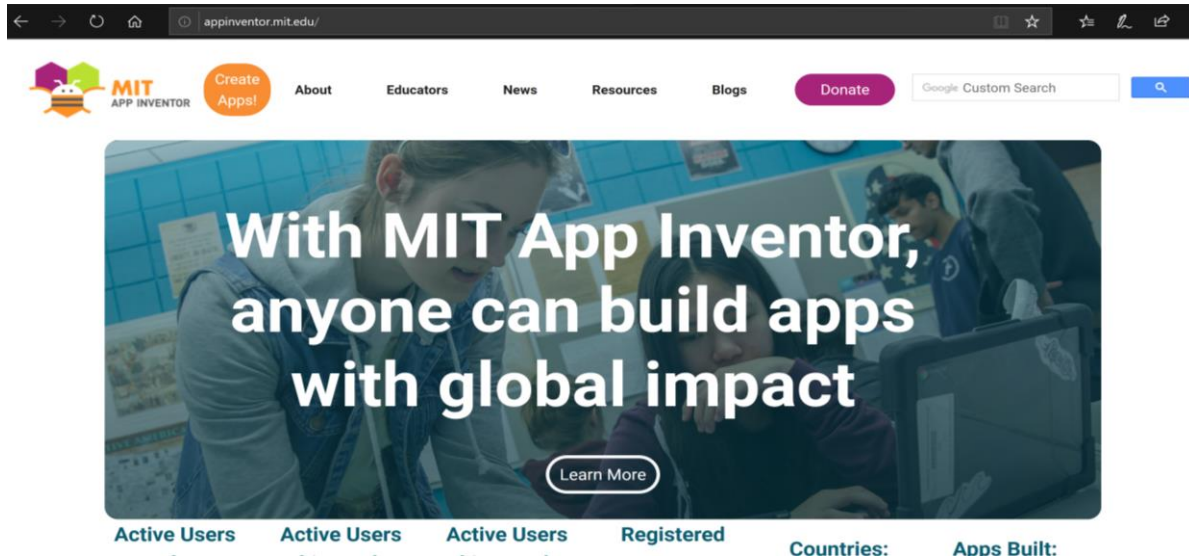
- Add screenshot of the final mobile app screen you built to this template
- Add screenshot of the blocks coding you wrote to this template



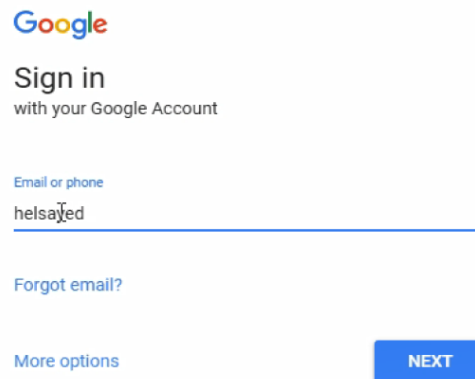
Ex#	Grade	Out of	Grader Comment(s)
Design (Run)		5	
Code		5	
Total		10	
Signature			

## Create a project on MIT App Inventor (After Setup)

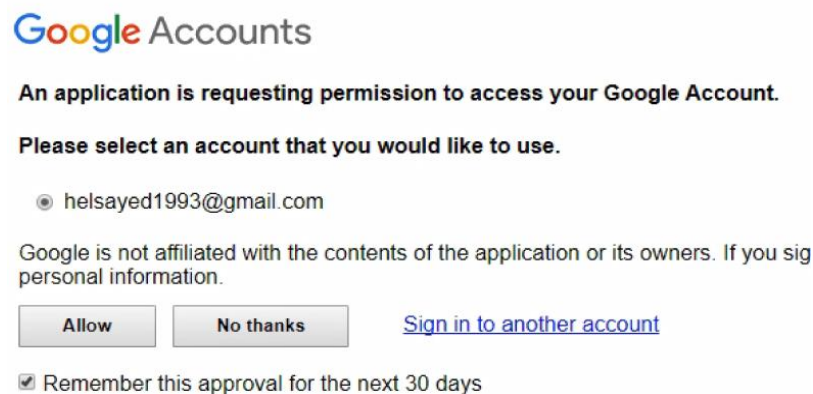
1. Open the following link on Chrome: <http://appinventor.mit.edu/explore/ai2/windows.html> then click on create apps button



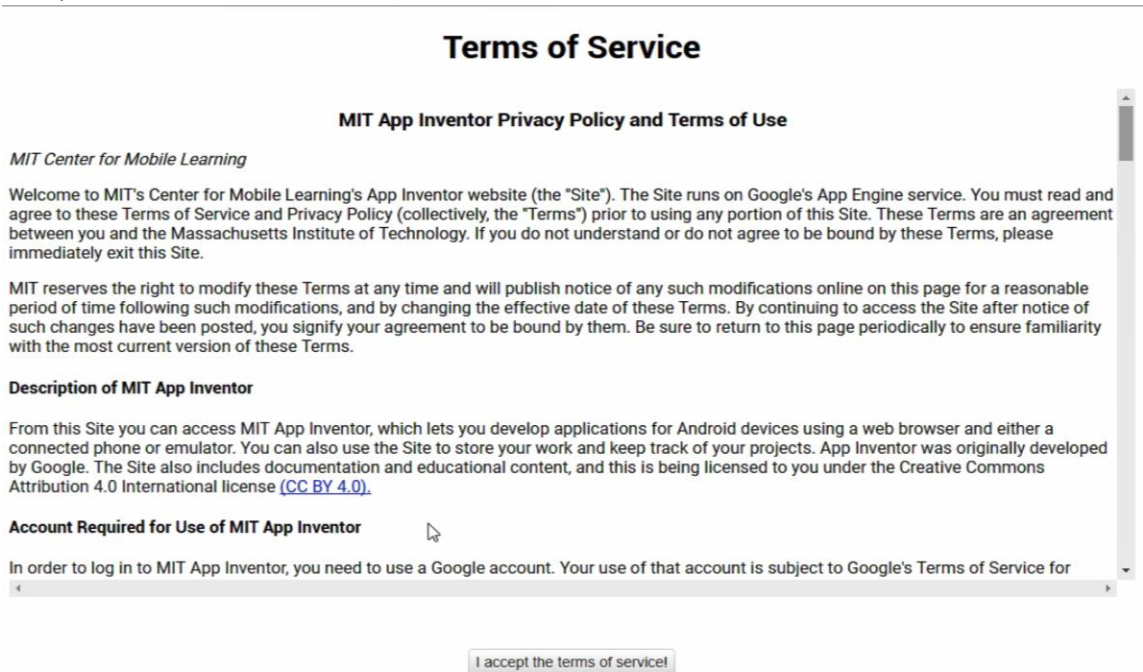
2. Sign in using your Gmail account:



3. Allow it a permission



4. Accept terms of services:



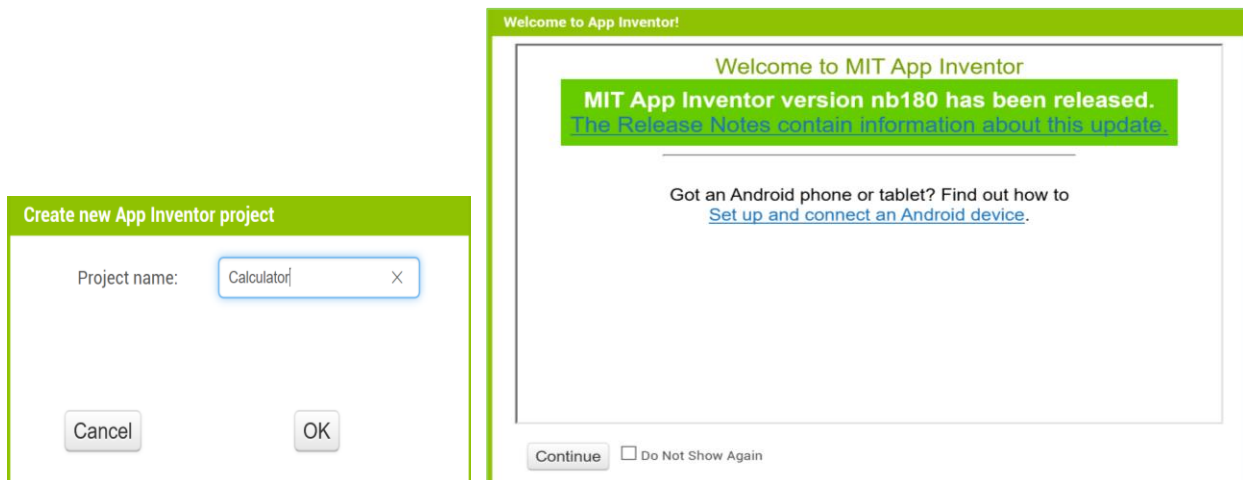
The screenshot shows the 'Terms of Service' page for MIT App Inventor. The title is 'Terms of Service' in bold. Below it is the subtitle 'MIT App Inventor Privacy Policy and Terms of Use'. The page is from the 'MIT Center for Mobile Learning'. The text explains that the site runs on Google's App Engine service and that users must agree to the terms. It also states that MIT reserves the right to modify the terms at any time. A section titled 'Description of MIT App Inventor' explains that the site allows users to develop applications for Android devices using a web browser and either a connected phone or emulator. It also mentions that the site includes documentation and educational content, and is licensed under the Creative Commons Attribution 4.0 International license. A section titled 'Account Required for Use of MIT App Inventor' states that users need a Google account to log in. At the bottom, there is a button that says 'I accept the terms of service!'.

5. Click on start new project



The screenshot shows the main interface of the MIT App Inventor website. The top navigation bar includes the MIT App Inventor logo and links for 'Projects', 'Connect', 'Build', 'Settings', and 'Help'. On the right side of the navigation bar, there are links for 'My Projects', 'Gallery', 'Guide', 'Report an Issue', 'English', and a user email address 'hag191993@gmail.com'. Below the navigation bar, there is a green bar with three buttons: 'Start new project', 'Delete Project', and 'Publish to Gallery'. Below the green bar, there is a section titled 'My Projects'.

6. Enter a valid project name and click continue



The screenshot shows two overlapping windows from the MIT App Inventor interface. The foreground window is titled 'Create new App Inventor project' and contains a text input field for 'Project name:' with the text 'Calculator' entered. There are 'Cancel' and 'OK' buttons at the bottom. The background window is titled 'Welcome to App Inventor!' and contains a message: 'Welcome to MIT App Inventor. MIT App Inventor version nb180 has been released. The Release Notes contain information about this update.' Below this message, there is a link: 'Got an Android phone or tablet? Find out how to Set up and connect an Android device.' At the bottom of the background window, there are 'Continue' and 'Do Not Show Again' buttons.

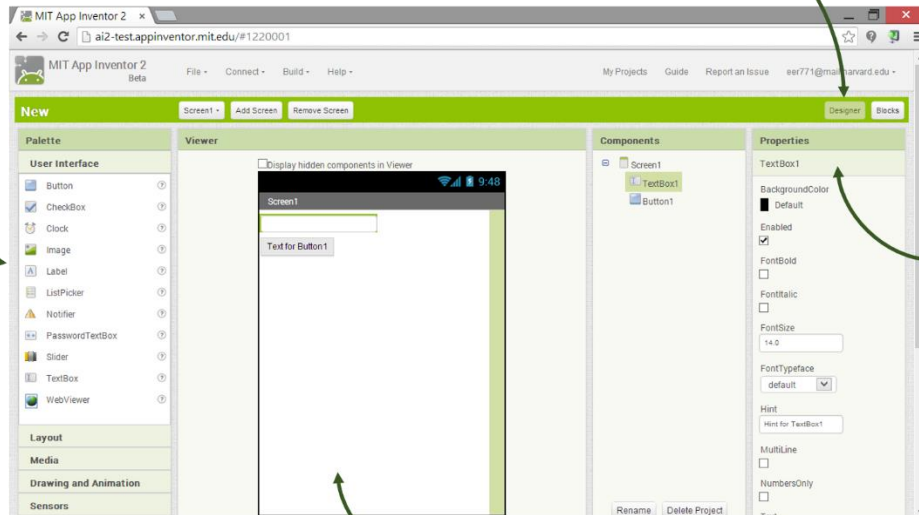
## Understand your IDE:

### 1. Design View

**Palette:** Find your components and drag them to the Viewer to add them to your app.

**Designer Button:**  
Click from any tab to go to the Designer tab.

**Properties:** Select a Component in the Components List to change its properties (color, size, behavior) here.



**Viewer:** Drag components from the Palette to the Viewer to see what your app will look like.

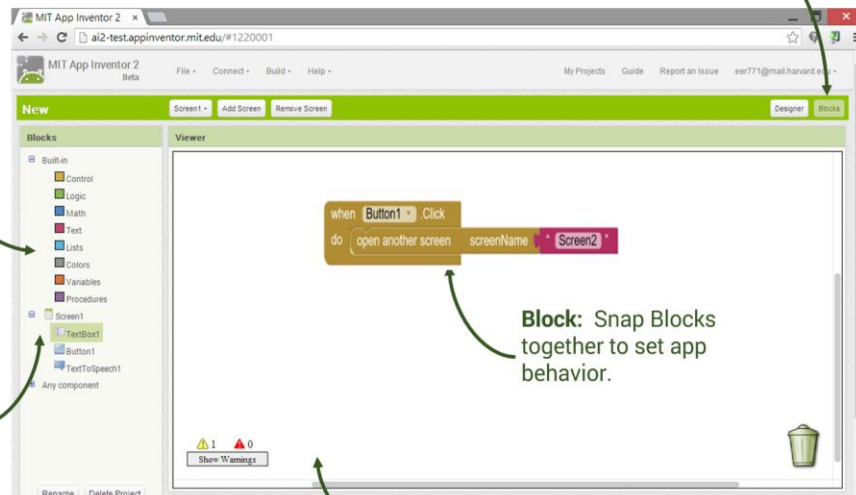
### 2. Blocks View

**Built-In Drawers:** Find Blocks for general behaviors you may want to add to your app and drag them to the Blocks Viewer.

**Blocks Button:** Click from any tab to go to the Blocks tab.

**Component-Specific Drawers:** Find Blocks for behaviors for specific Components and drag them to the Blocks Viewer.

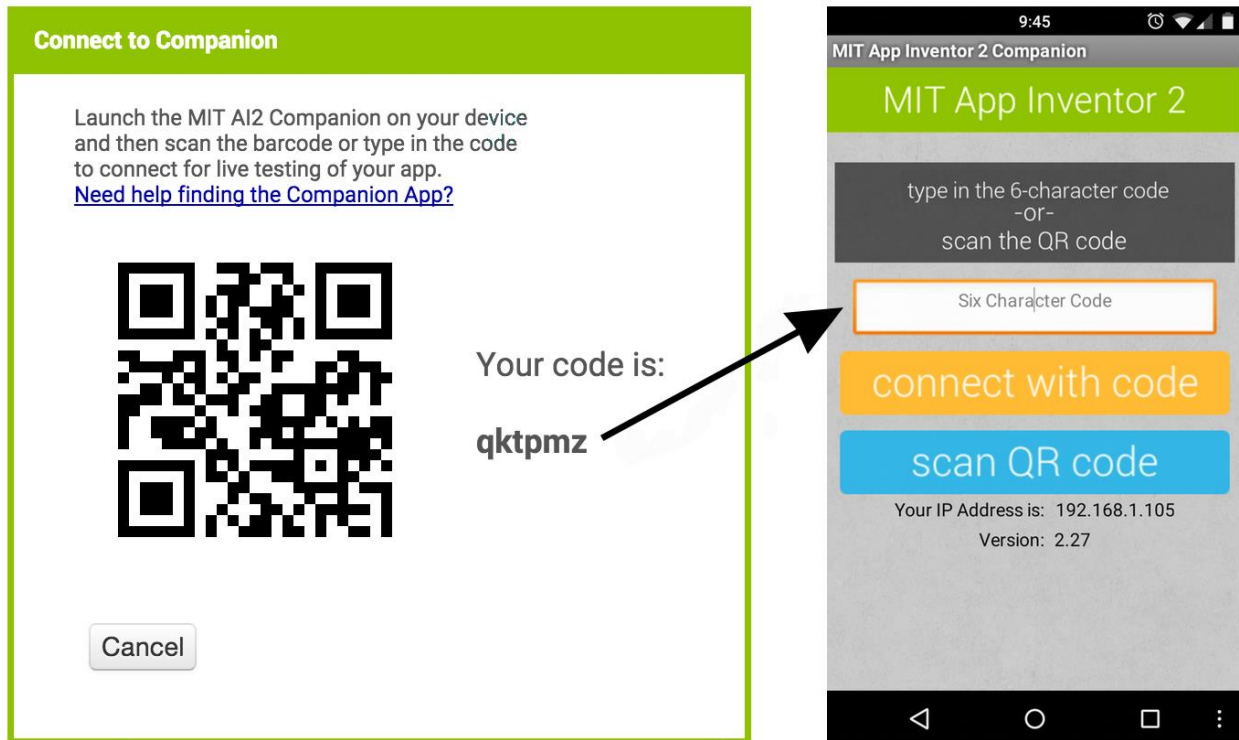
**Block:** Snap Blocks together to set app behavior.



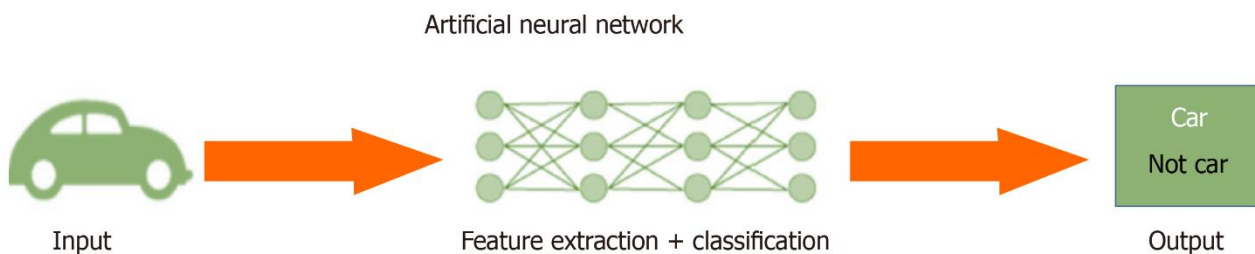
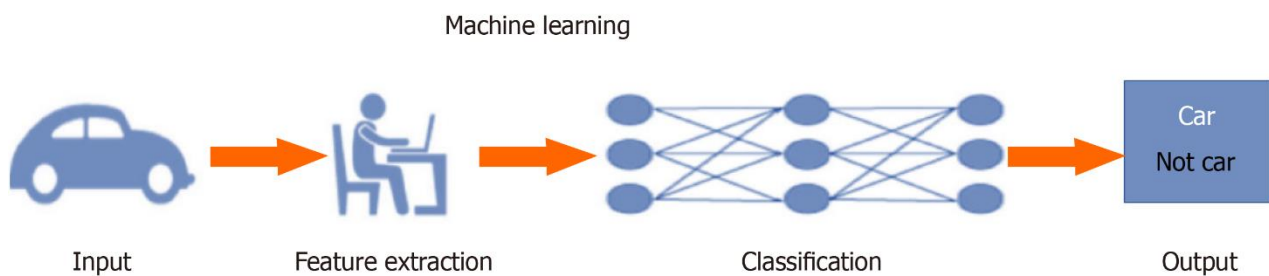
**Viewer:** Drag Blocks from the Drawers to the Blocks Viewer to build relationships and behavior.



**Install** MIT AI2 Companion on smart phone then click on connect (companion option) to scan generated QR Code:



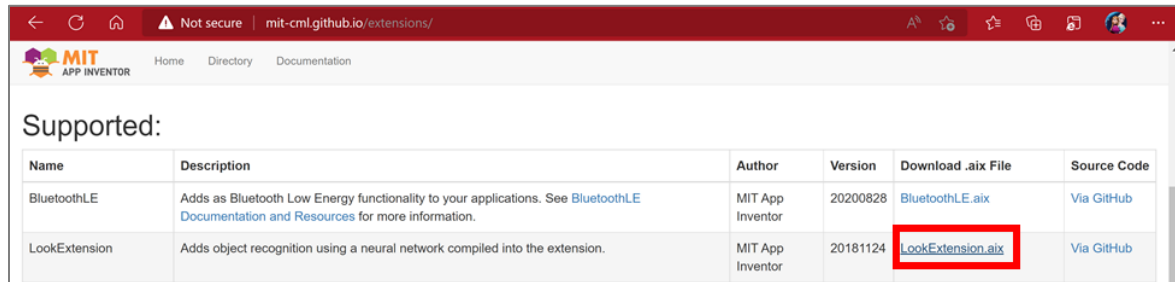
### What is Neural Networks classification:



## Exercise: Object Detector

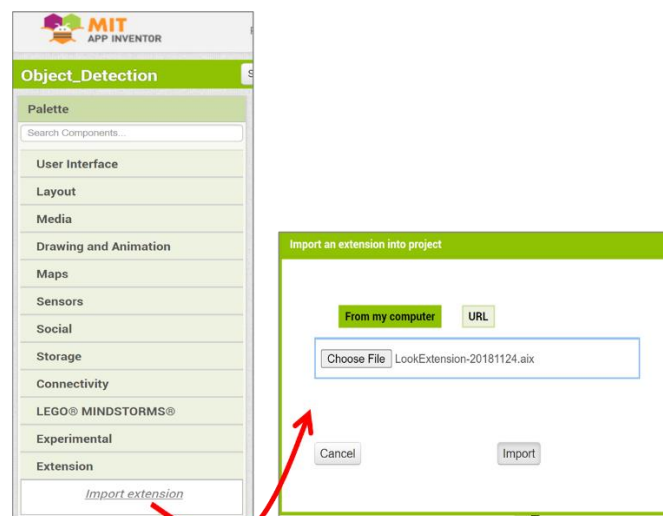
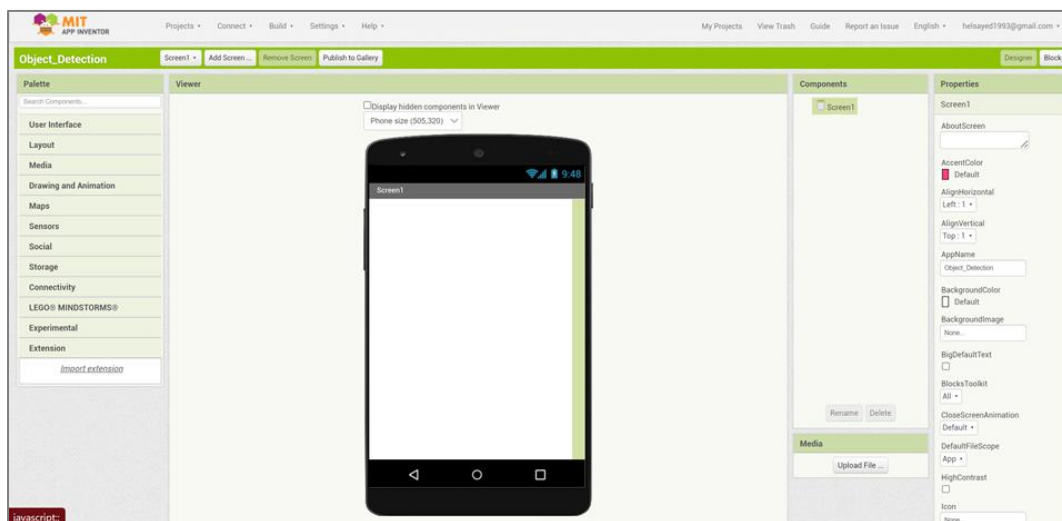
Follow these steps to build a mobile application that scans an object and scans it using Look library.

- 1- Open <http://mit-cml.github.io/extensions/> and click on LookExtension.aia extension link to download

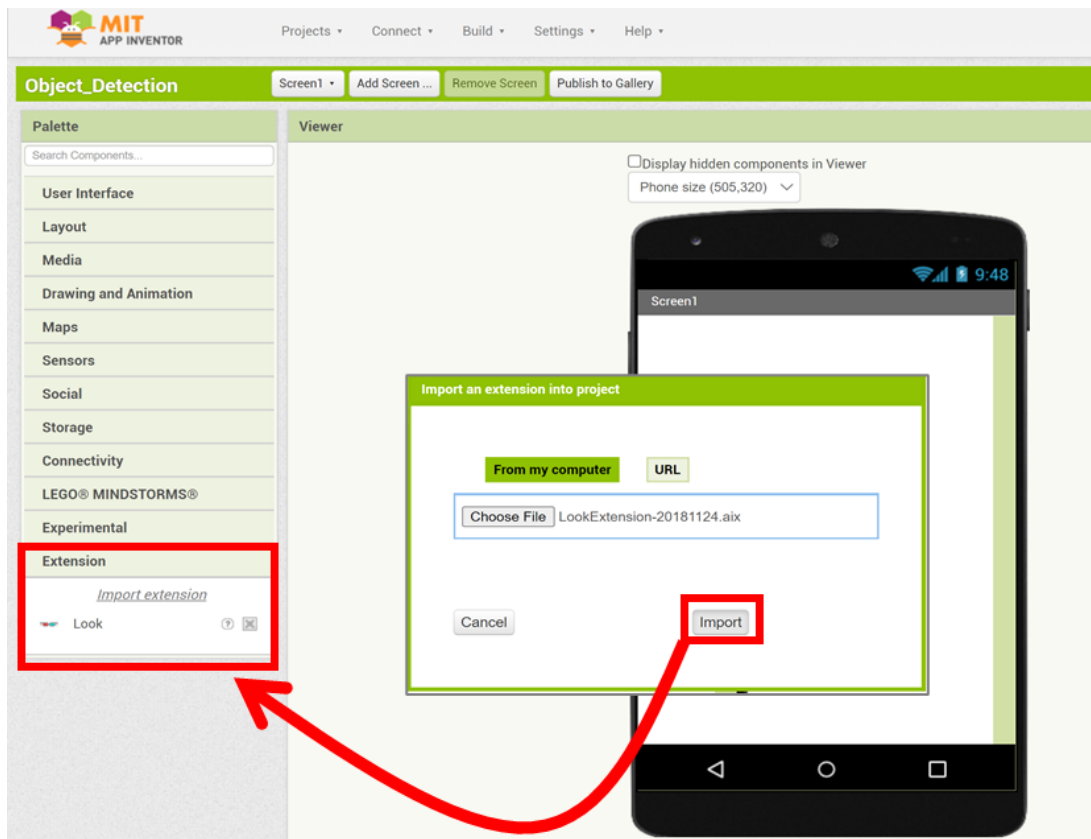


Name	Description	Author	Version	Download .aia File	Source Code
BluetoothLE	Adds as Bluetooth Low Energy functionality to your applications. See <a href="#">BluetoothLE Documentation and Resources</a> for more information.	MIT App Inventor	20200828	<a href="#">BluetoothLE.aia</a>	<a href="#">Via GitHub</a>
LookExtension	Adds object recognition using a neural network compiled into the extension.	MIT App Inventor	20181124	<a href="#">LookExtension.aia</a>	<a href="#">Via GitHub</a>

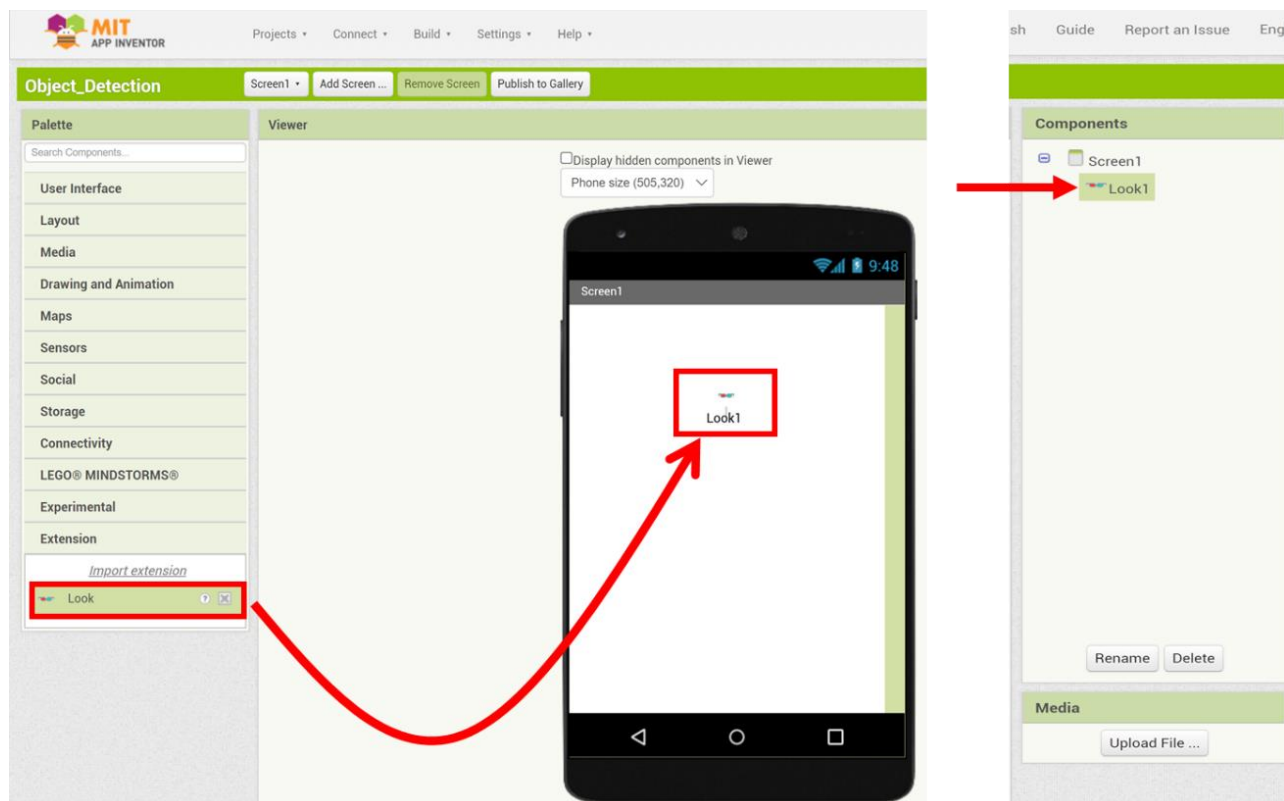
- 2- Open MIT project and scroll down to extensions then import it from the location on your PC



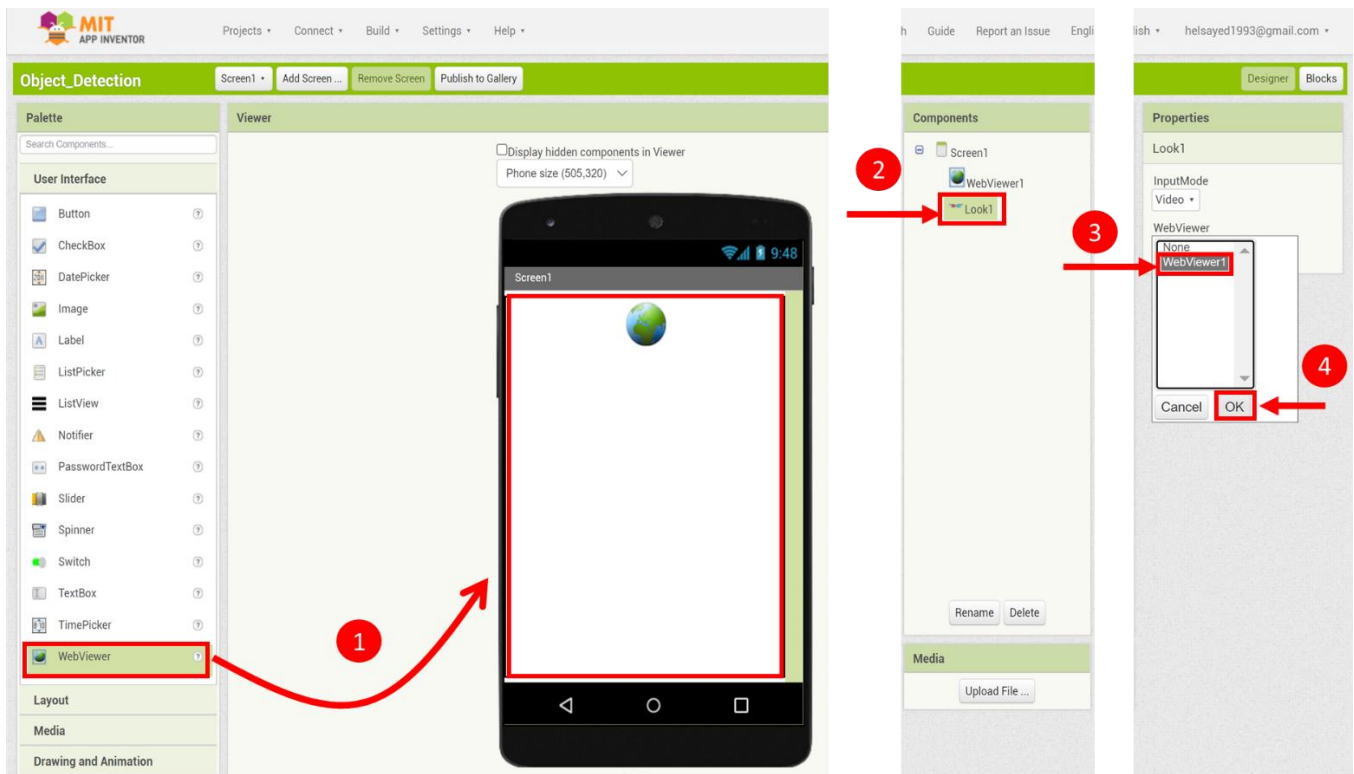




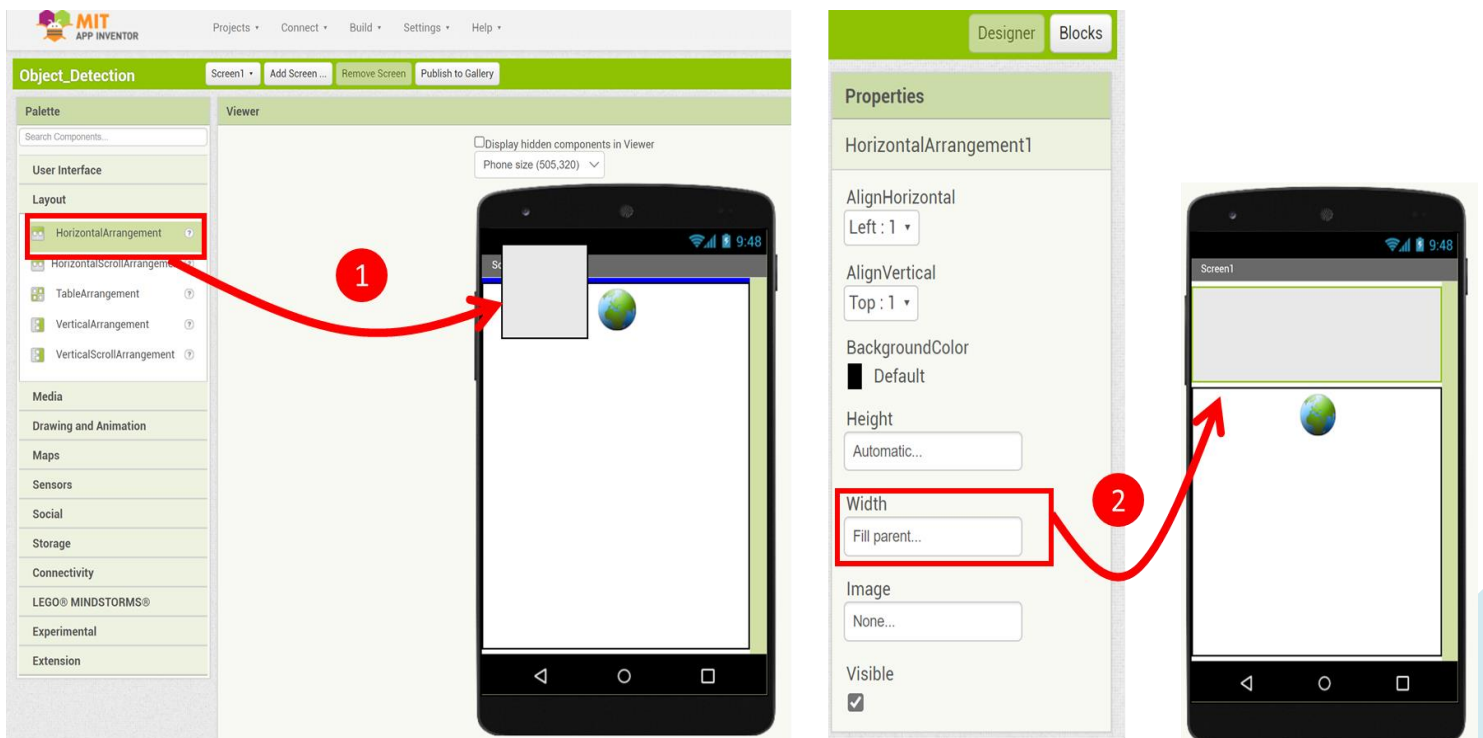
3- Drag and drop the extension into design area



- 4- Drag and drop a web viewer into design area (from user interface palette) then change webviewer property of look component to be the web viewer you just created

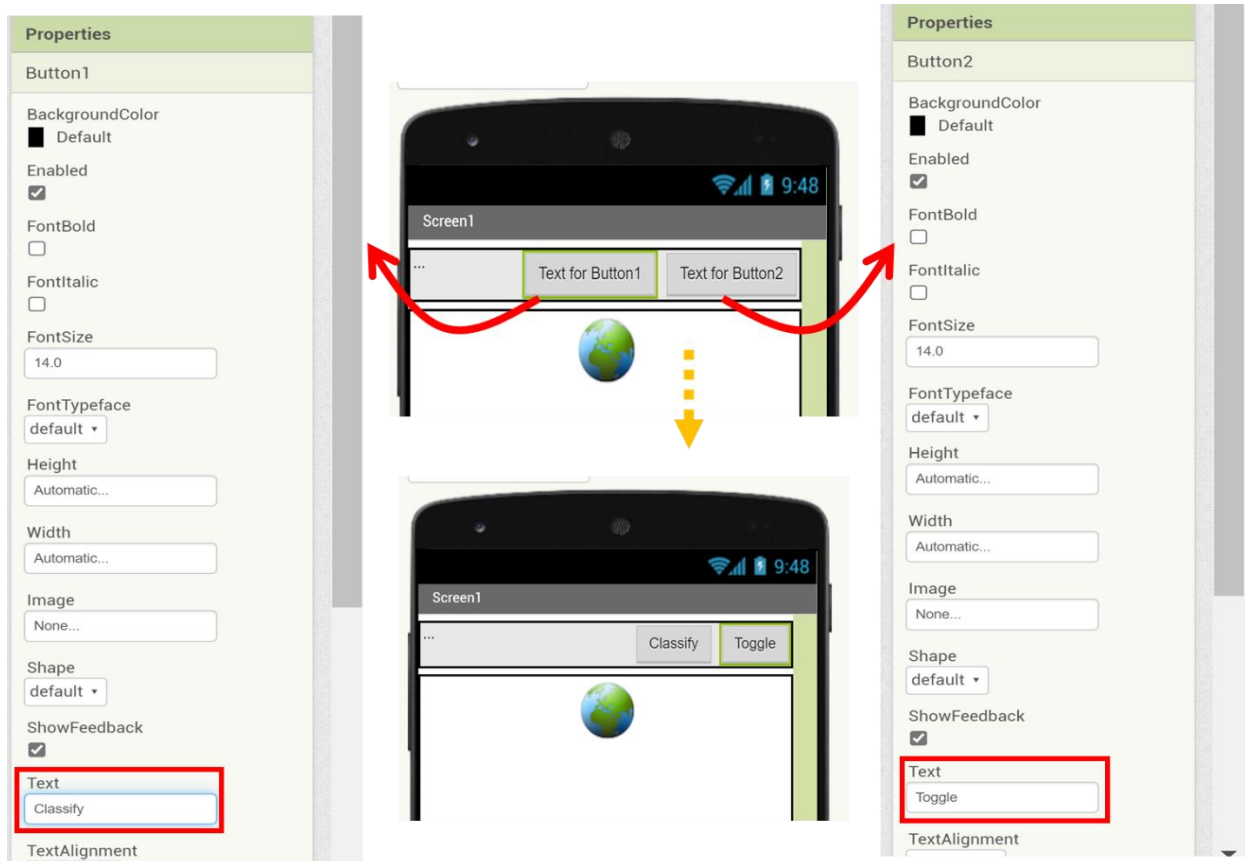
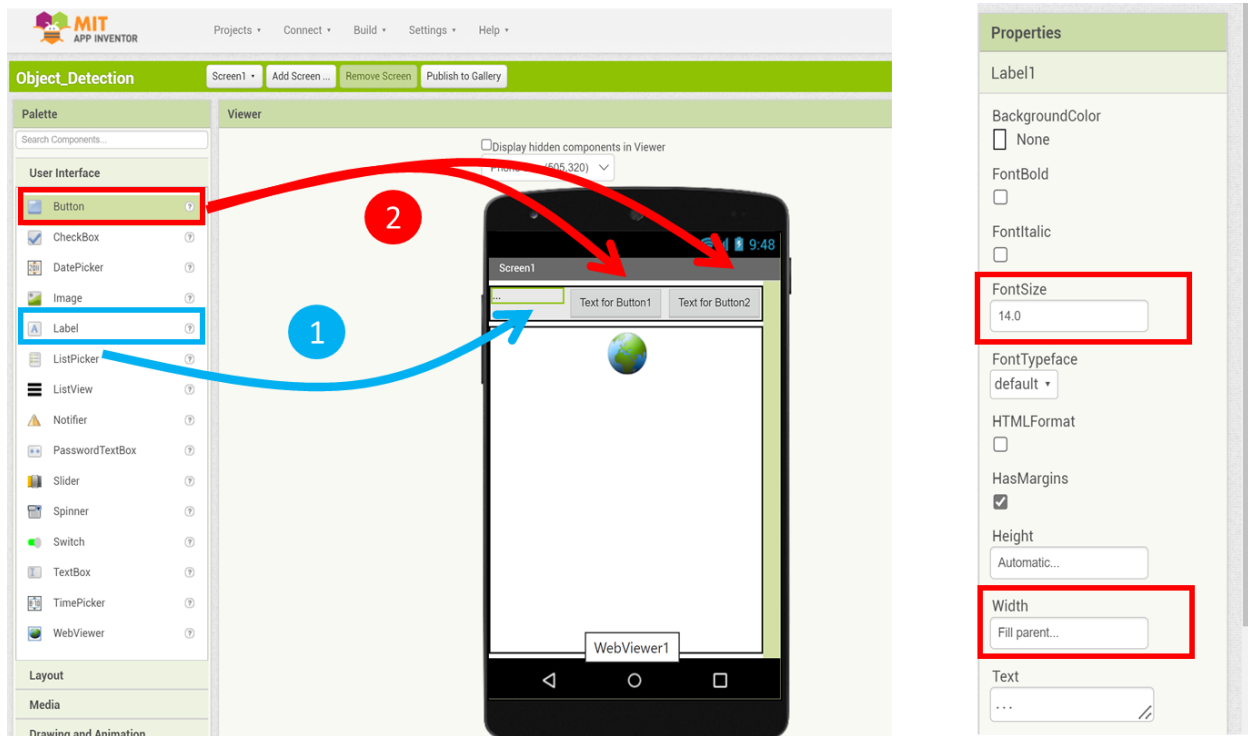


- 5- Drag a horizontal arrangement to the layout above the webviewer then set width property to fill parent

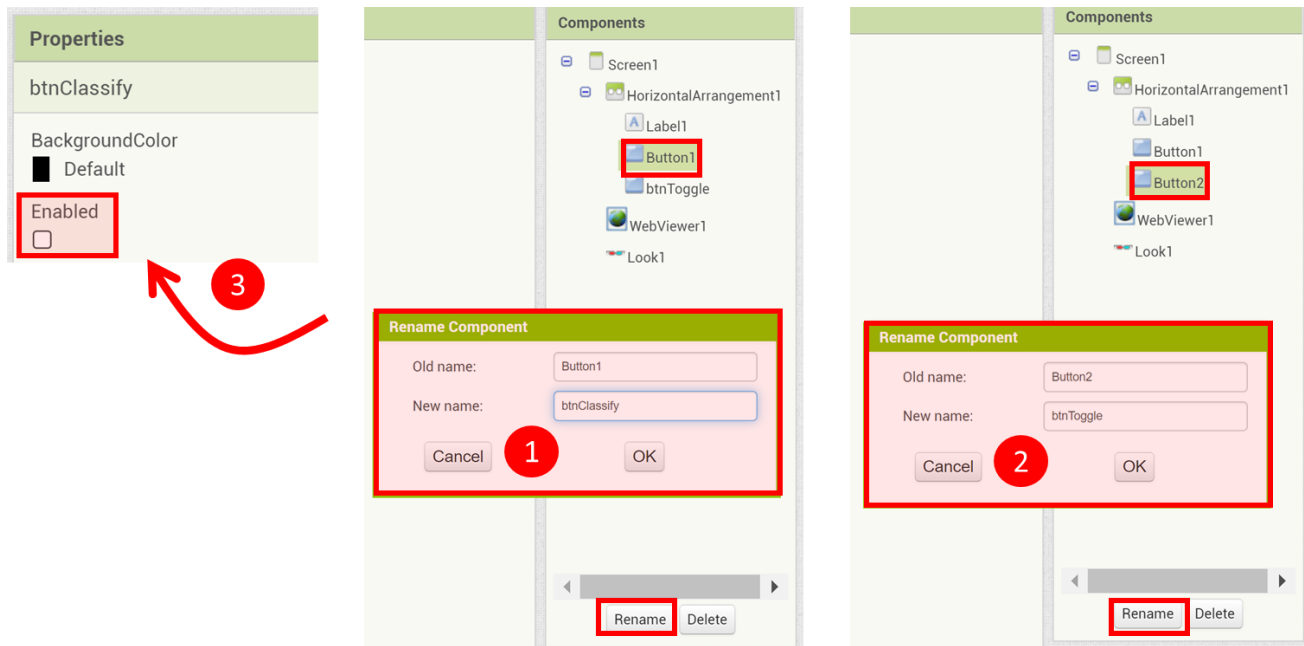




6- Drag a label and 2 buttons inside it with these properties



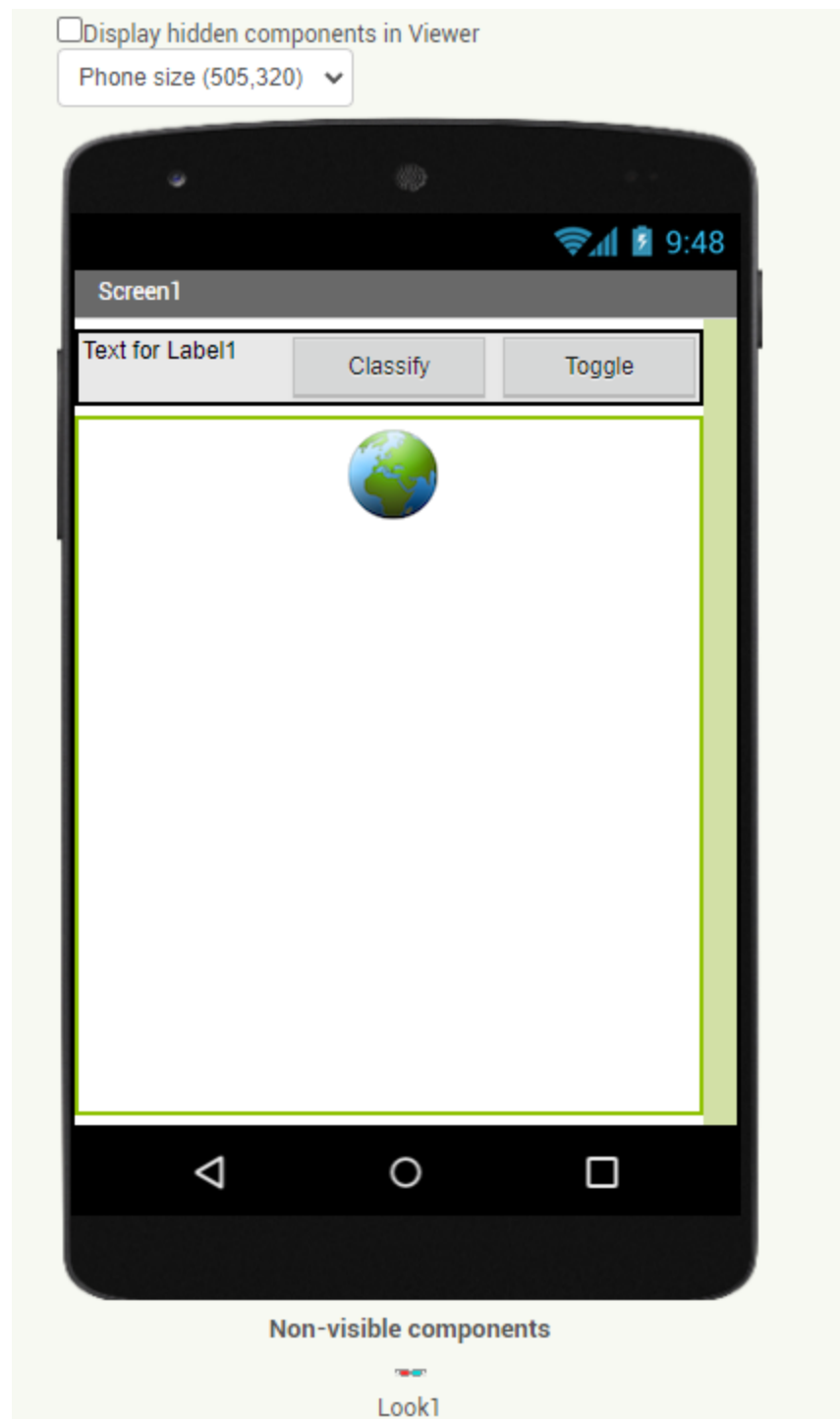
7- Rename buttons if needed and disable classify button by default



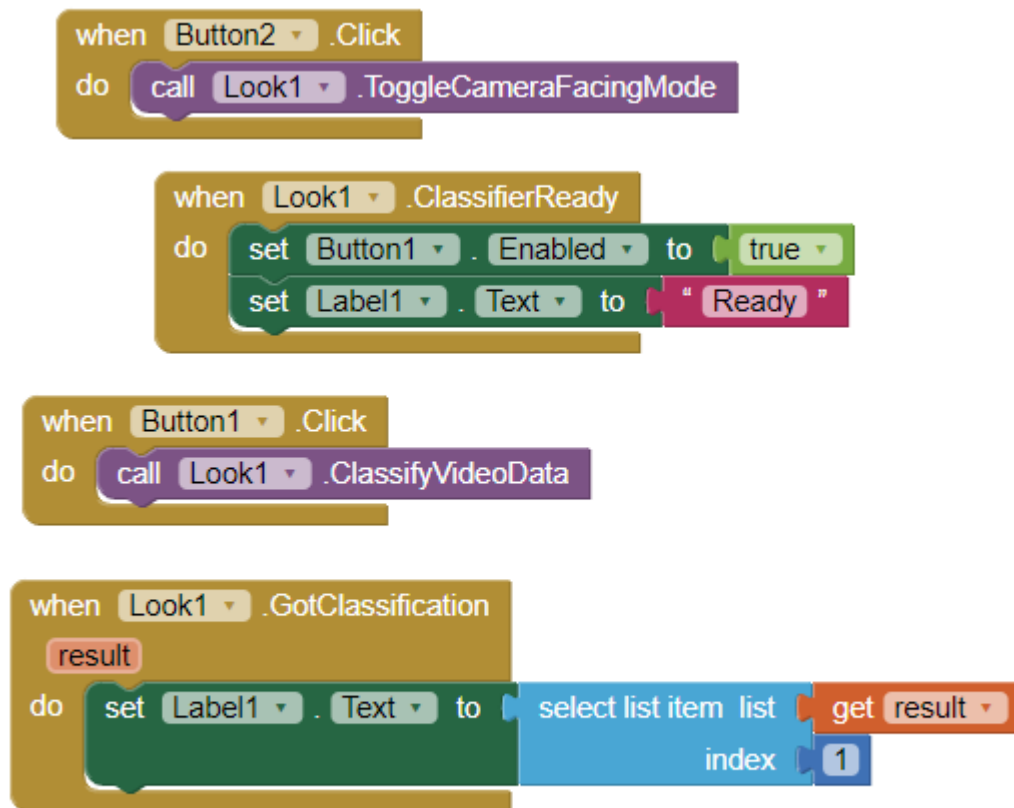
8- Play around with the properties for a better look and add Backend Code

Answer:

*Layout*

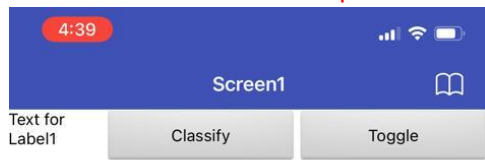


Code



*Object detection runs (2 exs)*

Look Extension are not compatible with IOS there is a demo video showing the error



invoke: unable to invoke method  
`ToggleCameraFacingMode` in object of type  
boolean. Irritants: ()