



مسابقة الذكاء الاصطناعي لطلاب المدارس الثانوية بدولة قطر

د. صالح الحازبي

قسم علوم وهندسة الحاسب

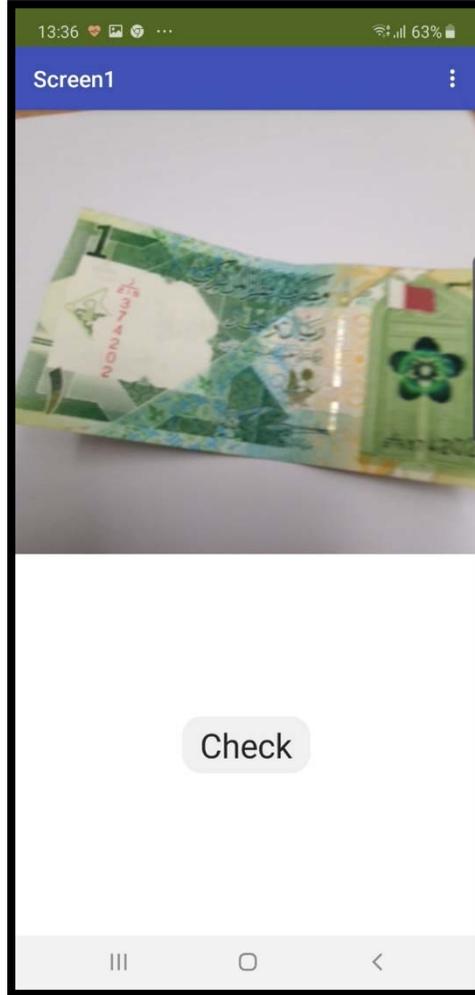
كلية الهندسة - جامعة قطر



الهدف من المسابقة

- رفع مستوى الوعي لدى طلاب المدارس بأهمية ودور الذكاء الاصطناعي في حل المشكلات الواقعية.
- تحفيز الشغف لدى الطلاب للاهتمام بهذا المجال
- تشجيع الطلاب على الإبداع والابتكار لتطوير حلول معتمدة على الذكاء الاصطناعي

الطلاب المستهدفون: طلاب المدارس الثانوية في دولة قطر.



مثال : بناء تطبيق لمساعدة الأشخاص الذين لديهم مشاكل بصرية في التعرف على اوراق العملة القطرية.

بالضغط على الزر "Check", يقوم التطبيق بالتعرف على الورقة النقدية و يصدر صوت بقيمة الورقة

الخطوات:

1- بناء مجموعة البيانات وتدريب الآلة للتعرف على الأوراق النقدية القطرية.

الوسيلة : استخدام Google Teachable Machine
<https://teachablemachine.withgoogle.com/>

2- بناء واجهة تطبيق الجوال باستخدام App Inventor
<https://appinventor.mit.edu/>

3- ربط التطبيق بنموذج التدريب.
الوسيلة : استخدام ملحق TMIC والذي يمكن تنزيله من الموقع التالي :
<https://codigos.ufsc.br/gqs/tmic/-/blob/master/dist/br.ufsc.gqs.teachablemachineimageclassifier.aix>

4- برمجة التطبيق باستخدام اللبنة.

الخطوة الأولى

بناء مجموعة البيانات وتدريب الآلة للتعرف على الأوراق النقدية القطرية.

الوسيلة : استخدام Google Teachable Machine
<https://teachablemachine.withgoogle.com/>

أ- بناء مجموعة البيانات : صور العملات الورقية (العملات)

Teachable Machine

تحميل الصور أو استخدام كاميرا الجهاز

The screenshot displays the Teachable Machine interface. On the left, there are five classes, each with a set of image samples and 'Webcam' and 'Upload' buttons. Class 1 has 42 samples, Class 2 has 29, Class 3 has 6, Class 4 has 47, and Class 5 has 18. In the center, there is a 'Training' panel with a 'Train Model' button and an 'Advanced' dropdown. On the right, there is a 'Preview' panel with an 'Export Model' button, an 'Input' section with a 'Webcam' dropdown and a 'Webcam' icon, and an 'Output' section with five colored bars representing Class 1 (orange), Class 2 (pink), Class 3 (purple), Class 4 (blue), and Class 5 (dark orange) with a '100%' label.

ب- تدريب النموذج

Teachable Machine

Class 1  

42 Image Samples

Webcam Upload 

Class 2  

29 Image Samples

Webcam Upload 

Class 3  

6 Image Samples

Webcam Upload 

Class 4  

47 Image Samples

Webcam Upload 

Class 5  

18 Image Samples

Webcam Upload 

Training

Train Model

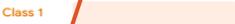
Advanced 

Preview  Export Model

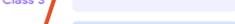
Input ON Webcam 



Output

Class 1 

Class 2 

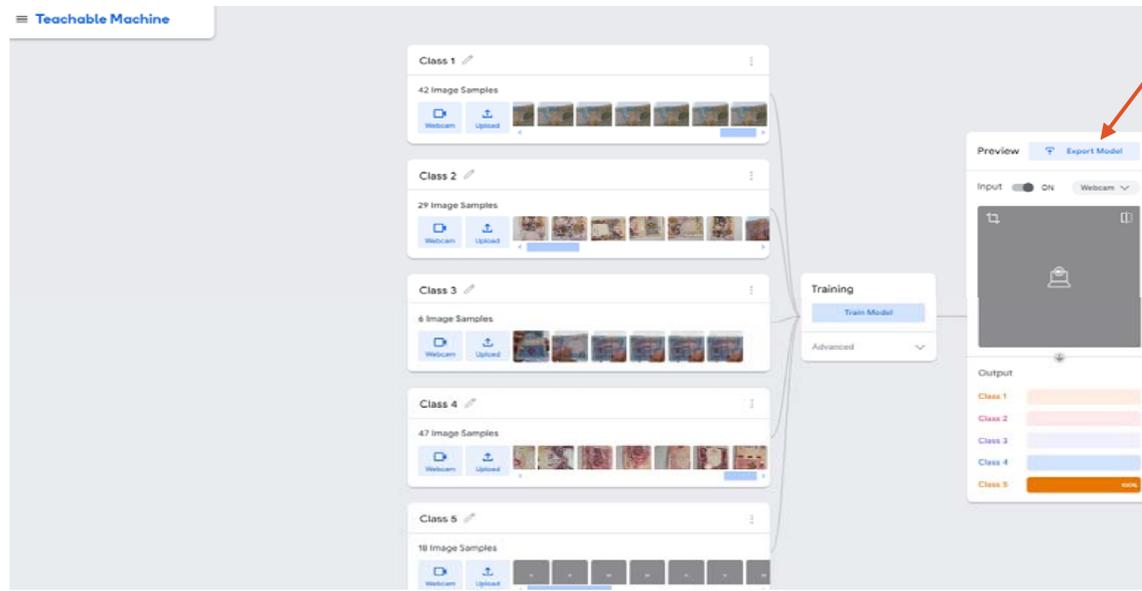
Class 3 

Class 4 

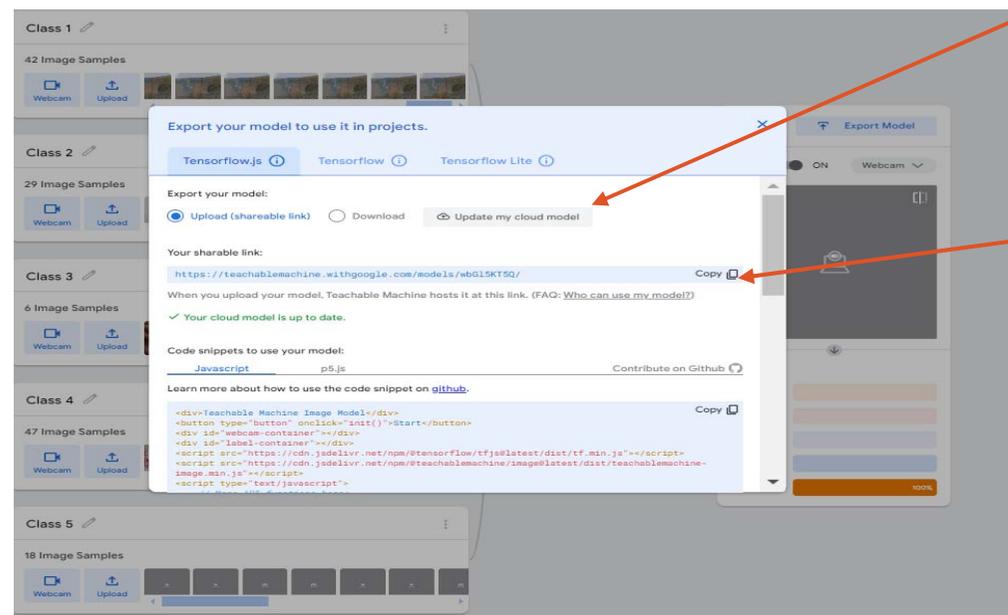
Class 5  100%

ج- اختبار النموذج

د- تصدير النموذج



ه- تحميل النموذج إلى Cloud



و- نسخ رابط النموذج

الخطوة 2: بناء تطبيق الجوال

الوسيلة : استخدام App Inventor

<https://appinventor.mit.edu>

The screenshot shows the MIT App Inventor website. The browser address bar displays <https://appinventor.mit.edu>. The navigation menu includes 'Create Apps!' (circled in red), 'About', 'For Educators', 'News & Events', 'Get Involved', 'Resources', 'Donate', and a search icon. The main banner features the MIT App Inventor logo and the text 'SUMMER APPATHON Congratulations Winners!'. Below the banner, there are four buttons: 'USERS TODAY:', 'USERS THIS MONTH:', 'ALL-TIME USERS:', and 'APPS BUILT:'. The URL in the address bar is <https://appathon.appinventor.mit.edu/results>.

Create new App Inventor project

Project name:

Toolkit: ?

Theme: ?

أ- إنشاء مشروع جديد

1- Drag/drop HorizontalArrangement

2- Change Height to 30 percent. Width to "Fill parent".

ب- بناء الواجهة

The image shows a screenshot of the MIT App Inventor web interface. The browser address bar displays `https://ai2.appinventor.mit.edu/#4519639107108864`. The interface includes a top navigation bar with 'MyProject', 'Screen1', 'Add Screen...', 'Remove Screen', 'Project Properties', and 'Publish to Gallery'. Below this is a 'Palette' on the left with categories like 'User Interface', 'Layout', 'Media', etc. The 'Layout' section contains 'HorizontalArrangement', 'HorizontalScrollArrangement', 'TableArrangement', 'VerticalArrangement', and 'VerticalScrollArrangement'. A red arrow points from the 'TableArrangement' component in the palette to a mobile screen in the 'Viewer' area. The mobile screen shows a blue header 'Screen1' and a grey rectangular area. A second red arrow points from the 'TableArrangement' component in the 'All Components' panel to the 'TableArrangement1' component on the mobile screen. The 'Properties' panel on the right shows the 'Appearance' section for 'TableArrangement1 (TableArrangement)', with 'Columns' set to 3 and 'Rows' set to 1. A red arrow points from the text 'Make it 3 columns and 1 row.' to the 'Columns' input field, and another red arrow points from the same text to the 'Rows' input field.

3-Drag/drop a TableArrangement

Make it 3 columns and 1 row.

← → ↻ <https://ai2.appinventor.mit.edu/#4519639107108864> 70% ☆

MIT APP INVENTOR Projects Connect Build Settings Help Loading ... My Projects View Trash Guide Report an Issue English salehalhazbi@gmail.com

MyProject Screen1 Add Screen ... Remove Screen Project Properties Publish to Gallery Designer Blocks

Palette Search Components...

User Interface

- Button
- CheckBox
- CircularProgress
- DatePicker
- Image
- Label
- LinearProgress
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Layout

Media

Viewer Display hidden components in Viewer
Phone size (505,320)
Android 5+ Devices (Android Material)

Screen1

Check

All Components

- Screen1
 - HorizontalArrangement1
 - TableArrangement1
 - Button1

Properties

Button1 (Button)

Appearance

- BackgroundColor: Default
- FontBold:
- FontItalic:
- FontSize: 14.0
- FontTypeface: default...
- Height: Automatic...
- Width: Automatic...
- Image: None...
- Shape: default
- ShowFeedback:
- Text: Check
- TextAlignment: center : 1
- TextColor

Rename Delete

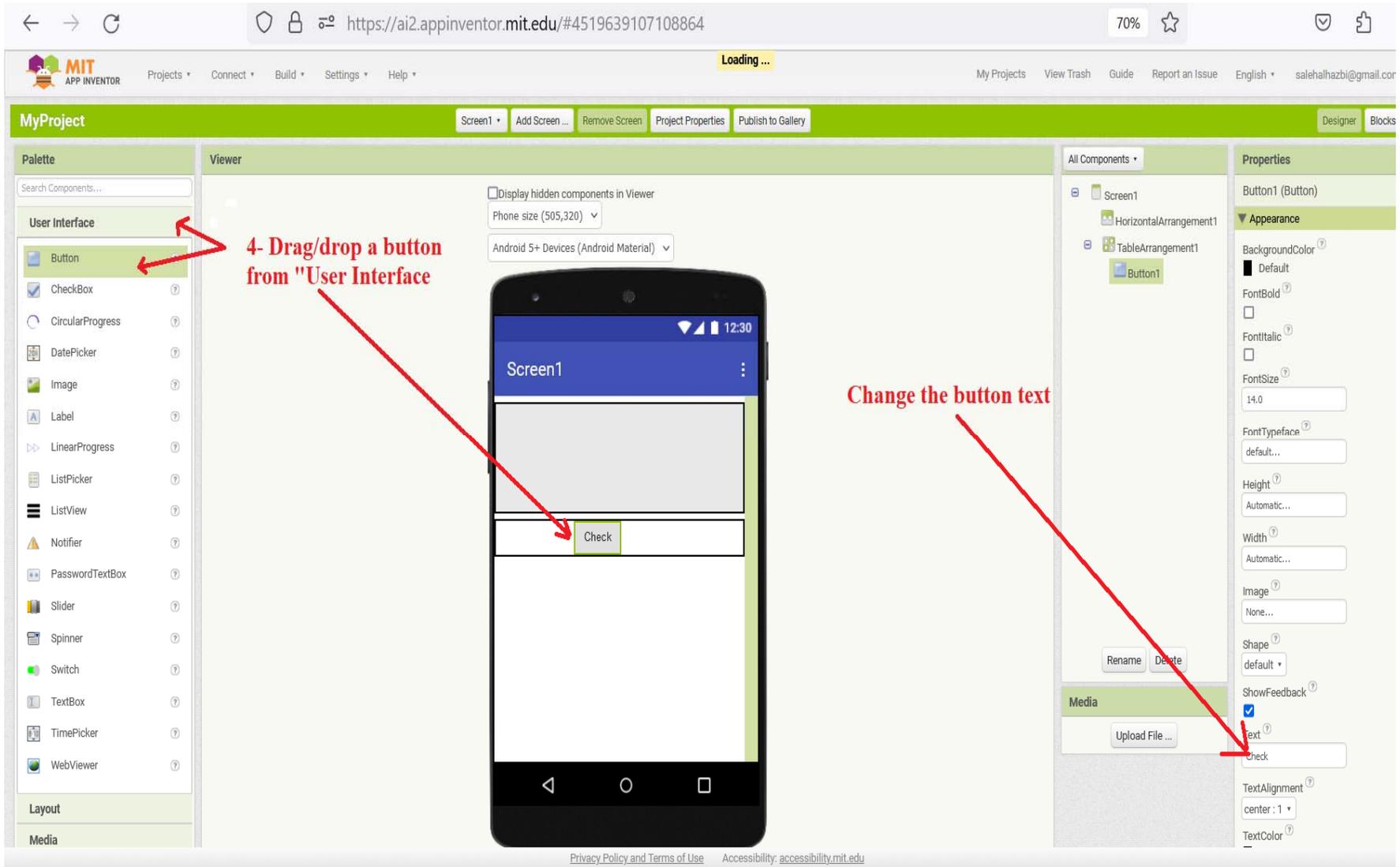
Media

Upload File ...

4- Drag/drop a button from "User Interface"

Change the button text

Privacy Policy and Terms of Use Accessibility: accessibility.mit.edu



MIT APP INVENTOR

Projects Connect Build Settings Help

My Projects View Trash Guide Report an Issue English salehalhazbi@gmail.com

MyProject Screen1 Add Screen Remove Screen Project Properties Publish to Gallery Designer Blocks

Search Components...

User Interface

- Button
- CheckBox
- CircularProgress
- DatePicker
- Image
- Label
- LinearProgress
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Layout

Media

Display hidden components in Viewer

Phone size (505,320)

Android 5+ Devices (Android Material)

Screen1

Screen1

HorizontalArrangement1

WebViewer1

TableArrangement1

Button1

Properties

WebViewer1 (WebViewer)

Appearance

Height

Fill parent...

Width

Fill parent...

Visible

Behavior

FollowLinks

HomeUri

IgnoreSslErrors

PromptforPermission

UsesLocation

Rename Delete

Media

Upload File ...

5-Drag and drop WebViewer

Make it "Fill parent"

MIT App Inventor

MIT App Inventor

https://ai2.appinventor.mit.edu/#4519639107108864

70%

MIT APP INVENTOR

Projects Connect Build Settings Help

My Projects View Trash Guide Report an Issue English salehalhazbi@gmail.com

Search Components...

User Interface

Layout

Media

Camcorder

Camera

FilePicker

ImagePicker

Player

Sound

SoundRecorder

SpeechRecognizer

TextToSpeech

Translator

VideoPlayer

Drawing and Animation

Maps

Charts

Data Science

Sensors

Social

Storage

Connectivity

Display hidden components in Viewer

Phone size (505,320)

Android 5+ Devices (Android Material)

Screen1

HorizontalArrangement1

WebView1

TableArrangement1

Button1

TextToSpeech1

TextToSpeech1 (TextToSpeech)

Behavior

Country

Default

Language

Default

Pitch

1.0

SpeechRate

1.0

Check

Non-visible components

TextToSpeech1

Upload File ...

6-Add TextToSpeech to the interface

الخطوة 3: ربط التطبيق بنموذج التدريب باستخدام ملحق TMIC

أ- قم بتنزيل الملحق من الموقع التالي

<https://codigos.ufsc.br/gqs/tmic/-/blob/master/dist/br.ufsc.gqs.teachablemachineimageclassifier.aix>

The screenshot shows the GitLab interface for the repository 'TMIC - Teachable Machine Image Classifier'. The file 'br.ufsc.gqs.teachablemachineimageclassifier.aix' is displayed with a size of 437 KB. A red circle highlights the download icon in the top right corner of the file entry. Below the file entry, a large red circle highlights the 'Download (437 KB)' button.

ب- إضافة الملحق إلى التطبيق

The screenshot displays the MIT App Inventor web interface. On the left sidebar, the 'Extension' category is selected, and the 'Import extension' button is highlighted with a red arrow. The main workspace shows a mobile device preview with a 'TextToSpeech1' component added to the 'Non-visible components' area. A dialog box titled 'Import an extension into project' is open, with a red arrow pointing to the 'From my computer' button. The dialog box also features a 'Browse...' button, a 'URL' input field, and 'Cancel' and 'Import' buttons. The right sidebar shows the 'TextToSpeech1 (TextToSpeech)' component's properties, including 'Country', 'Language', 'Pitch', and 'SpeechRate'.

Select Extension->import extension

upload the downloaded extension from your computer



- Search Components...
- User Interface
 - Layout
 - Media
 - Drawing and Animation
 - Maps
 - Charts
 - Data Science
 - Sensors
 - Social
 - Storage
 - Connectivity
 - LEGO® MINDSTORMS®
 - Experimental
 - Extension
- [Import extension](#)
- TeachableMachineImageClassifier

Display hidden components in Viewer

Phone size (505,320) ▾

Android 5+ Devices (Android Material) ▾

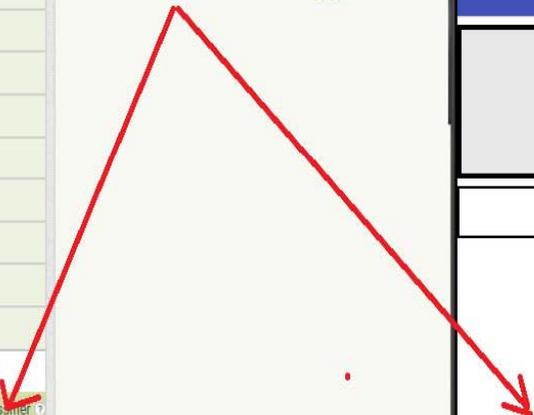
Screen1

Check

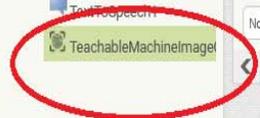
Non-visible components

- TextToSpeech1
- TeachableMachineImageClassifier1

Drag/drop the extension to the app



- Screen1
 - HorizontalArrangement1
 - WebView1
 - TableArrangement1
 - Button1
 - TextToSpeech1
 - TeachableMachineImageClassifier1



ج- ربط التطبيق بالنموذج : قم باختيار الملحق

The screenshot displays the MIT App Inventor web interface. The browser address bar shows the URL: <https://ai2.appinventor.mit.edu/#4519639107108864>. The interface includes a top navigation bar with 'MyProject', 'Screen1', 'Add Screen...', 'Remove Screen', 'Project Properties', and 'Publish to Gallery'. Below this is a 'Designer' tab and a 'Blocks' button. The main workspace is divided into several panels:

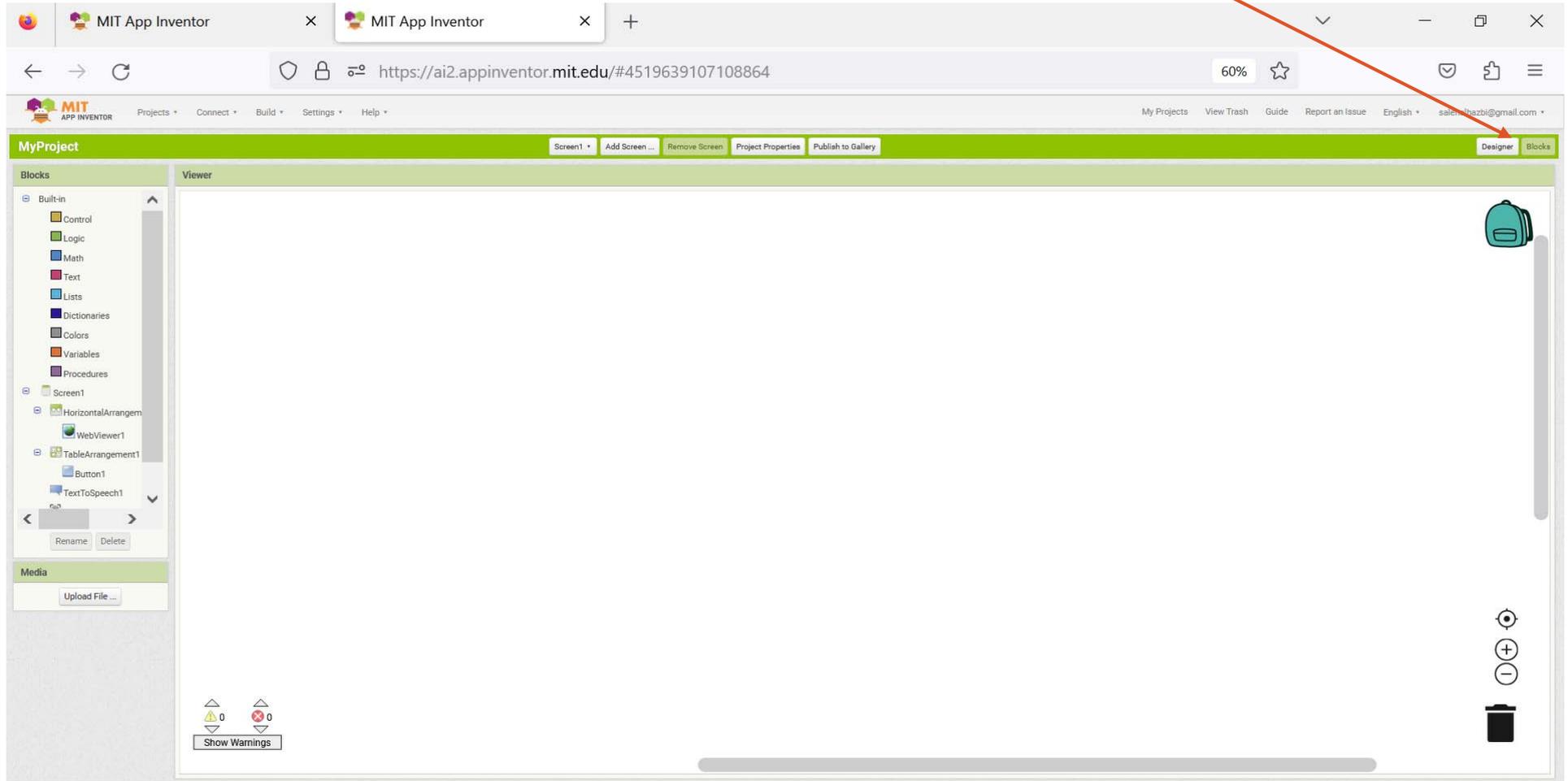
- Palette:** A sidebar on the left containing various UI components categorized into 'User Interface', 'Layout', 'Media', 'Drawing and Animation', 'Maps', 'Charts', and 'Data Science'. The 'WebViewer' component is visible in the 'User Interface' section.
- Viewer:** A central area showing a mobile device simulation. The screen displays 'Screen1' with a globe icon and a 'Check' button. Below the device, there is a 'Non-visible components' section showing 'TextToSpeech1' and 'TeachableMachineImageClassifier1'.
- All Components:** A panel on the right showing a tree view of the project's components, including 'Screen1', 'HorizontalArrangement1', 'WebViewer1', 'TableArrangement1', 'Button1', 'TextToSpeech1', and 'TeachableMachineImageClassifier1'.
- Properties:** A panel on the right showing the properties of the selected 'TeachableMachineImageClassifier1' component. The 'UriModel' property is set to 'https://teachablemachine.'. The 'WebViewer' dropdown menu is open, showing 'None' and 'WebViewer1' as options.

Two red arrows point from the text annotations to the 'UriModel' property and the 'WebViewer1' option in the dropdown menu.

Add the link of the model that was copied from Google Teachable Machine

set the webViewer to Webviewer 1

الخطوة 4: برمجة التطبيق باستخدام اللبانات. غير الواجهة إلى اللبانات لبرمجة التطبيق بالضغط على "Blocks".



عند الضغط على الزر يتم استدعاء الملحق والذي يقوم باستدعاء النموذج الذي تم تدريبه وارجاع النتائج على شكل Dictionary

ينبغي قراءة هذه النتائج ومن ثم تحديد أي من الأصناف الذي حصل على نسبة أعلى واعتباره هو الصنف الذي تم التعرف عليه ومن ثم استدعاء خدمة TextToSpeech لكي يحول النص إلى صوت.

نحتاج بداية لتعريف متغيرات التي سيتم حفظ نسبة الثقة في التعرف على كل صنف

The screenshot displays the MIT App Inventor web interface. The browser address bar shows the URL <https://ai2.appinventor.mit.edu/#5734939563524096>. The interface includes a top navigation bar with 'MIT APP INVENTOR' and menu options like 'Projects', 'Connect', 'Build', 'Settings', and 'Help'. Below this is a project-specific bar for 'Proj1' with buttons for 'Screen1', 'Add Screen...', 'Remove Screen', 'Project Properties', and 'Publish to Gallery'. The main workspace is divided into 'Blocks' and 'Viewer' panes. The 'Blocks' pane on the left lists categories: Built-in (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures), Screen1, HorizontalArrangem, WebView1, HorizontalArrangem, Button1, and TeachableMachinelr. A red arrow points from the 'Variables' category in the 'Blocks' pane to a stack of four 'initialize global' blocks in the 'Viewer' pane. These blocks are: 'initialize global one to 0', 'initialize global five to 0', 'initialize global Ten to 0', and 'initialize global fifty to 0'.

تحميل الملحق عند الضغط على الزر

The screenshot displays the MIT App Inventor web interface. The top navigation bar includes the MIT App Inventor logo and menu items: Projects, Connect, Build, Settings, and Help. Below this, a project header for 'Proj1' contains buttons for 'Screen1', 'Add Screen...', 'Remove Screen', 'Project Properties', and 'Publish to Gallery'. The main workspace is divided into 'Blocks' and 'Viewer' sections. The 'Blocks' panel on the left lists categories: Built-in, Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under 'Screen1', components like 'HorizontalArrangem', 'WebView1', 'Button1', and 'TeachableMachin...' are visible. The 'Viewer' section shows a code block for a 'when Button1 .Click' event. This block contains four 'initialize global' blocks (one, five, Ten, fifty) each set to 0, followed by a 'do' block that calls 'TeachableMachineImageClassifier1 .ClassifyVideoData'. Red arrows point from the 'Control' category in the 'Blocks' panel to the 'initialize global' blocks, and from the 'TeachableMachin...' component in the 'Screen1' list to the 'call' block in the code.

استدعاء الملحق وقراءة النتائج

The image shows a programming environment with a 'BLOCKS' panel on the left and a 'viewer' area on the right. The 'BLOCKS' panel lists categories like Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under 'Screen1', there are blocks for HorizontalArrangem, WebView1, Button1, and TeachableMachine. A red arrow points from the 'TeachableMachine' block in the 'BLOCKS' panel to the 'result' block in the script.

The script in the viewer area consists of the following blocks:

- initialize global `titty` to `0`
- when `Button1` .Click
- do call `TeachableMachineImageClassifier1` .ClassifyVideoData
- when `TeachableMachineImageClassifier1` .GotClassification
- do `result`
- set global `one` to `get value for key "Class 1" in dictionary get result or if not found "not found"`
- set global `five` to `get value for key "Class 2" in dictionary get result or if not found "not found"`
- set global `Ten` to `get value for key "Class 3" in dictionary get result or if not found "not found"`
- set global `fifty` to `get value for key "Class 4" in dictionary get result or if not found "not found"`
- if `get global one` \geq `50`

At the bottom of the viewer area, there are warning icons (0) and a 'Show Warnings' button.

تحديد الصنف وتحويل النص إلى صوت

The screenshot displays the App Inventor interface for a project named "Proj1". The interface is divided into three main sections: "Blocks", "Viewer", and "Media".

- Blocks:** A vertical sidebar on the left contains various block categories: Logic, Math, Text, Lists, Dictionaries, Colors, Variables, and Procedures. Under the "Dictionaries" category, a "TextToSpeech1" block is visible. A red arrow points from this block to its usage in the script.
- Viewer:** The main workspace on the right contains a Scratch-style script. The script begins with a "set global fifty to" block, followed by a "get value for key in dictionary or if not found" block. The key is "Class 4" and the default value is "not found".
- Script Logic:** The script uses a series of "if-then-else if" blocks to check the value of global variables against a threshold of 50. Each check is followed by a "call TextToSpeech1 .Speak" block with a specific message:
 - If "global one" is ≥ 50 , the message is "One Rial".
 - If "global five" is ≥ 50 , the message is "five Rials".
 - If "global Ten" is ≥ 50 , the message is "Ten Rials".
 - If "global fifty" is ≥ 50 , the message is "Fifty Rials".
 - If none of the above conditions are met, the message is "Not recognized".

شُكْرًا لَكُمْ