## About the course

You'll see that this course is a little different from the previous courses. Over the last three courses you've developed skills and strategies for creating interactive apps through videos, quizzes and peer assessments.

At this level we hope you can take what you've learned and - using the starter code - create your own unique responsive, interactive app.

**You have a choice of the kind of app you will create.** There are two different project choices: **Fun With Selfies**, and **Fun With Web Services**. You can choose either type of project depending on your preference; we have provided starter code for each one. In Fun With Selfies, you will create an animated photo app that uses a face tracking module that we have provided for you. In Fun With Web Services, you will create an app that connects to a web service "in the cloud", which is a common scenario for apps developed on a professional level.

Some of the content will be only applicable to one type of project or the other. If you see content titled with the project name (as a prefix), then that content only applies if you are doing that particular project.

During the run of this course we will have less video and put more of a focus on the final project. To help prepare for this final project you will see 1 peer assessment assignment worth 10% of your total grade. This assignment is meant to encourage participation and have everyone provide constructive feedback on your project. Please use this as an opportunity to experiment with your own work and give helpful feedback to your peers. The practice and feedback will help prepare you for your final assignment.

This final assignment is worth 90% of your total grade as it brings together all you've learned in this, and previous, courses. Your grade will be based on this final functioning app.

We look forward to seeing what you've created.

## 

## Fun with Selfies: Using the FaceTracker framework

Please note: The latest version of the FaceTracker module does not require use of a specific app ID.

If you cannot build it with the included app ID, change the ID in the project settings to a unique one.

In this version you must also enable ‘Allow Arbitrary Loads’ in your project Info.plist file. See instructions below for how to include the framework in your own original project.

# Adding the framework to your project

1. Open your project in Xcode.

2. Drag and drop the \*FaceTracker.framework\* file into the project navigator.

3. In the popup that appears, ensure \*Copy items if needed\* is checked and click \*Finish\*.

4. Select the project in the project navigator to open up the project settings. Select the main target and drag and drop the \*FaceTracker.framework\* file from the project navigator to the \*Embedded Binaries\* section.

5. Go to the \*Build Settings\* tab and set \*Enable Bitcode\* to \*No\*.

6. Open your project's Info.plist file. Add the following row: "App Transport Security". (To add a row, right-click an existing row and select 'Add Row', then use the drop-down to select the desired key.)

7. Click to open the disclosure triangle next to App Transport Security and add the sub-key "Allow Arbitrary Loads". Make sure to set the value for this row to YES.

# Using the framework

\* A view controller called \*FaceTrackerViewController\* has been included which handles the face tracking and rendering. This view controller can be instantiated using code or through interface builder.

\* The view controller provides a protocol called \*FaceTrackerViewControllerDelegate\* which notifies the delegate whenever the face points have changed.

\* The delegate can use the returned face points to adjust the positioning of the views in the app.

# Demo App

A demo app has been included that shows an example of how to use the face tracker framework.

Fun with Web services

<https://www.coursera.org/learn/build-app/supplement/V4kGg/fun-with-web-services>

# Yelp

## Getting Started

Before you can use the Yelp API or run the sample app, you need a Yelp account and your own Client ID and Secret. Getting these tokens is easy and free, and only takes a few minutes.

Here are the steps:

* Visit the Getting Started Page:
* <https://www.yelp.ca/developers/documentation/v3/get_started>
* Click **Create App**. (It’s under **Developer Settings** on the left-hand side menu.
* Log in with a Yelp account, or sign up for a new account.
* Submit the **Create New App** form. You’ll be asked for some basic information about your app.
* You will be provided with your Client ID and Secret.

If you are using our YelpAPI Swift class to communicate with Yelp, you can use these codes to initialize an instance of YelpAPI.

If you share your code with someone else or post it online, make sure to delete your client ID and secret from your code first. You must keep them private – like a username and password.

If you would like to run the sample project we have provided, copy your ClientID and Secret into the noted location in the ViewController class.

# Microsoft Cognitive Services

## Getting Started

Before you can use the Microsoft Cognitive Services API, or run the sample app, you need a Microsoft account and a subscription key for the API(s) you wish to use. Setting this up is easy and free and it only takes a few moments.

Here are the steps:

* Visit the Microsoft Cognitive Services home page:
* <https://www.microsoft.com/cognitive-services>
* Click **Get started for free**
* Subscribe using one of the available accounts and confirm your account by following Microsoft’s instructions.
* Click **Request new trials** and select the products you would like to use.
* If you would like to try the sample code we have provided, you will need to request a subscription key for the **Computer Vision** service.
* You will be provided with a subscription key.

If you are using our MSCSAPIClient class to communicate with the MSCS API, make you can use this subscription key to initialize an instance of MSCSAPIClient.

If you share your code with someone else or post it online, make sure to delete subscription key from your code first. You must keep it private – like a username and password.

If you would like to run the sample project we have provided, copy your Computer Vision Subscription key into the noted location in the ViewController class.

## Yelp

<https://www.yelp.com/developers/documentation/v3/get_started>

## Microsoft Cognitive Services

### All Cognitive Services APIs

https://www.microsoft.com/cognitive-services/en-us/documentation

### Computer Vision API

<https://dev.projectoxford.ai/docs/services/56f91f2d778daf23d8ec6739/operations/56f91f2e778daf14a499e1fa>

## Final Project

Congratulations for working through the course (and likely the entire specialization)!

At this point you have had some feedback on your plans for your final project. In this section you will submit your final app for review by your peers (read full details in the submission section).

Your final grade is based on this project. We look forward to seeing what you have created!

NOTE: A few have expressed concerns about privacy with the final assignment as we ask to submit your app in action on YouTube. If your app uses the face tracker framework, you need not use your or a friend's face. Feel free to use the facetracker on a magazine picture or television image. That is just fine!