

Assignment 3

Instructions and Grading Criteria

- This is an **individual** assessment. Please review the college's **Academic Integrity Policy** to ensure that you are completing your work in an academically honest manner.
- Your application will be tested using an iPhone 14. It is your responsibility to ensure that your application runs properly on this simulator.
- In addition to the required functionality, learners are expected to use the coding conventions demonstrated in class, meaningful variable naming, and clearly organized code. Comments are helpful but not required.
- This assignment is due on **24th March 2023**. Late submissions will not be accepted.

Submission Checklist

For your submission to be graded, provide a **zip** file of your project, and a **screen recording** demonstrating the functionality you implemented.

1. Create a zip file of your project

- Name the zip file **A3_firstname_lastname.zip**. **.7zip or .rar files will not be accepted.**

3. Creating Your Screen Recording

- In the screen recording, demonstrate the app running in the console, and show the relevant output.
- Max 7 mins.

3. In the assignment:

1. Upload your screen recording to **Microsoft OneDrive** and ensure that the link is set to: "Anyone with the link can view". Paste a link to the recording in the **submission comments**.
2. Submit your zip file containing the project

Academic Integrity

- This is an individual assessment.
- Permitted activities: Usage of Internet to search for syntax only; usage of course materials
- Not permitted:
 - Communication with others (both inside and outside the class)
 - Discussion of solution or approaches with others; sharing/using a "reference" from someone
 - Searching the internet for full or partial solutions
 - Sharing of resources, including links, computers, accounts

Problem Description

Create two apps:

1. Map App
2. Movie App

1. Map App:

Here is the data containing the title, latitude, and longitude of the 7 wonders:

```
[  
  ["title": "The Great Wall of China", "latitude": "40.431908", "longitude": "116.570374"],  
  ["title": "Chichén-Itzá, Mexico", "latitude": "20.682985", "longitude": "-88.568649"],  
  ["title": "Petra, Jordan", "latitude": "30.328960", "longitude": "35.444832"],  
  ["title": "Machu Picchu, Peru", "latitude": "-13.163068", "longitude": "-72.545128"],  
  ["title": "Christ the Redeemer, Rio de Janeiro", "latitude": "-22.908333", "longitude": "-  
43.196388"],  
  ["title": "Colosseum, Rome", "latitude": "41.890251", "longitude": "12.492373"],  
  ["title": "Taj Mahal, India", "latitude": "27.173891", "longitude": "78.042068"],  
]
```

Instructions:

- Use this data to plot annotations on the map.
- When an annotation is selected:
 - o Make a direction request (`MKDirections(request:)`) from user's location to that annotation's location
 - o Draw the direction and render it on the map (`MKPolylineRenderer(overlay:)`)
- If no directions are found, then alert the user accordingly.
- The app must have a text field where you can enter an address and a search button which will geocode the address to a Placemark.
- From that placemark, extract required information (title, latitude, longitude) and add it to the existing array of 7 wonders, then plot it as an annotation on the map.

Marks Weightage:

- | | |
|----------------------------|----------------|
| - Plotting annotations: | 1 mark |
| - Direction Request: | 2 marks |
| - Polyline overlay render: | 2 marks |
| - Geocoding: | 1 mark |
| - Alert: | 1 mark |
| - Total: | 7 marks |

2. Movie App:

Here are a few movies related APIs:

<https://www.omdbapi.com/>

<https://developers.themoviedb.org/3/movies/get-movie-details>

PS: You might have to register (for free) to get an API key for the API to work

You can get a free **OMDb API** key here: <https://www.omdbapi.com/apikey.aspx>

Instructions:

- Create an iOS app that has a search bar, search button and a table view.
- User can type a movie name in the search bar and press the search button.
- Once the search button is pressed, make an API call to get all movies related to that search keyword. (Hint: <https://www.omdbapi.com/?apikey=<key>&s=Batman>)
- Populate the table view with the result array of movies.
- For OMDb API, the structure looks like this:

```
{
  "Search": [
    {
      "Title": "Batman Begins",
      "Year": "2005",
      "imdbID": "tt0372784",
      "Type": "movie",
      "Poster": "..."
    },
    ...
  ]
}
```

- User can select any row from the table view which should take the user to movie description screen.
 - o Hint: To get details of movie, you will need its IMDbId which you can get from the movie array in **didSelect row** function of table view
 - o Once you are on the Movie Details screen with the IMDbId, you can call this API: <https://omdbapi.com/?apikey=<key>&i=<IMDbId>>
- Take design inspiration from:
 - o <https://dribbble.com/shots/4375313-Film>
 - o <https://dribbble.com/shots/7879826-Movie-and-TV-shows-App>
 - o <https://dribbble.com/shots/8860423-Movie-App>
 - o Other sources

Marks Weightage:

- | | |
|--|----------------|
| - Search, table view: | 2 marks |
| - List of movies API call and display: | 2 marks |
| - Movie details API call and display: | 2 marks |
| - Overall UI & UX: | 1 mark |
| - MVC pattern: | 1 mark |
| - Total: | 8 marks |

1. Map App:	7 marks
2. Movie App:	8 marks
Total:	15 marks

END OF ASSESSMENT