**Assignment 4**

**Points 20**

SearchApp is a simple application where you will get images of the topic based on your search.

**Please follow the following instructions to complete this assignment.**

1. Open Xcode from the launchpad of your mac.
2. Click on create a new Xcode project. Select the iOS template and click on the App application.
3. Click on next which will prompt you to choose options for the project.
4. Provide product name as **Lastname\_SearchApp** , “**nwmsu**” for organization identifier, “**storyboard**” as interface and swift as language.
5. Click on next and select an appropriate location to save your app and click on create. A project directory will be loaded.
6. When you are done with selecting the location of the project on your local storage, moving further you can see project settings screen. Set your **Project Format** to “Xcode 13.0-compatible” in your Project Document settings.
7. Now look into **Deployment Info** settings on the same project settings and set it to “ios 15.0”

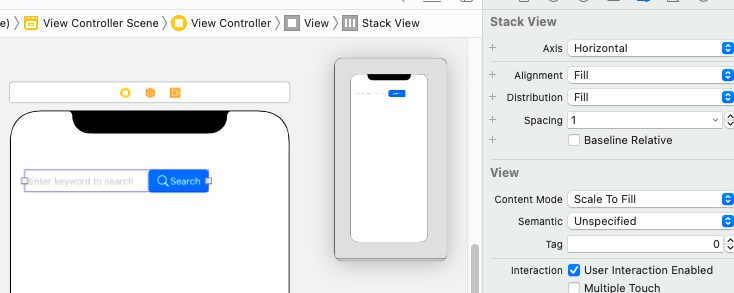
and make sure to check iPhone and iPad. Check below image and highlighted portion

with red and follow accordingly.

Graphical user interface, application

Description automatically generated

1. **Note:** Follow the exact instructions given and don’t modify any other settings as it causes application not to run in different machines.
2. From the project navigator click on “Main.storyboard” file, a blank mobile screen will be loaded where the required fields for an app need to be added.
3. Add a Text Field from the component library to the screen and name it as “**searchTextField**” while creating outlet and add a descriptive text to the placeholder from the attribute inspector. Ex: Enter a keyword to search.
4. Add a **filled** button or just button from library and name it as **searchButtonAction** for the IBAction. Add an image to the button from the attribute inspector by selecting image attribute and select *magnifyingglass* icon from the dropdown. Use your creativity to color the button.
5. Add the searchTextField and search button to a stack view. Select the stack view and in the attribute inspector make sure the distribution is fill and spacing is 1.



1. Now we need an image to the layout. Drag an Image View from the library and add it below the stack view. Name it as “**resultImage**” while making outlet connection. Give content mode as Aspect fit from the attribute inspector.
2. Add a new filled button or just button from the library and change the text as ***Next***, create another button and change the test to **Prev*,*** Similarly, create another button and change the test to **Reset,**Use your creativity for styling the buttons. Give a connection for action and name it as **ShowNextImagesBtn** ,**ShowPrevImagesBtn** and **ResetBtn.**
3. Give nice background to the buttons.
4. Now add a Text View from the library and name it as “**topicInfoText**” while making the outlet connection.
5. Add appropriate stack views to align the controls.
6. The layout will be misaligned when added to stack. Add constraints to the main stack view (left-20, top-30/40, right-20,bottom-20) from the constraint’s inspector. Give a constraint for height of 200 for the **resultImage**. Give a height of 150 for the ***topicInfoText.*** Use Distribution as Equal Spacing/Fill proportionally and spacing is 10 or as necessary. After applying all the constraints, the final view should be similar to the below image:

Graphical user interface, application

Description automatically generated

1. Now we are done with the UI part. Make sure all the connections for the controls are made correctly.
2. This app is about displaying the images based on the keyword we give. For example, if we search for **actors** all the images that are related to actors should be displayed. Any information related to actors or topic is shown in the text view. Text should be different for different topic images.
3. When we click on the next and previous images button the other actor or topic images should be displayed.
4. For the above functionality to work, choose any 3 topics of your choice like animals, books, flowers, actors etc. You can use the above topics. Download at least 5 images for each topic. Add the downloaded images to the assets folder.
5. Now in the controller file create an array with 5 values where each value is an array of image names of the respective topic.

Ex: var arr = [

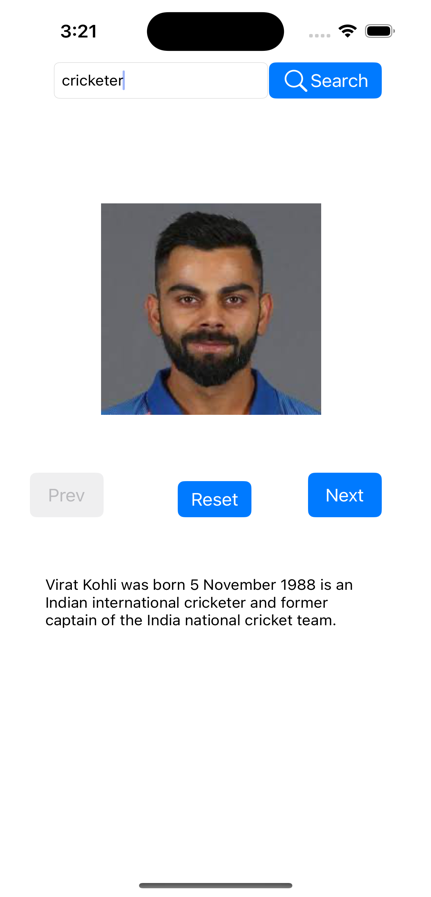
[images of topic1]

, [images of topic2],

[images of topic3]]

1. Now create 3 arrays with different keywords for each topic or nested array of arrays for the keywords, if we want to create a keywords array for actors’ topic, then the keywords would be: “actor”,” movie”,” hero”,” film” etc. You can give your own keywords for the selected topics. Name the keywords arrays as topicname\_keywords. Ex: actors\_keywords, flowers\_keywords etc.
2. For topics create a new array called topics\_array and give a brief explanation about the topic image that is displayed.
3. So, whenever user enters a text and click on search button then if the text matches with the keywords of any topic, then display only that topic images. You can use **actors\_keywords.contains()** for checking whether the array contains that element are not.
4. Every time user clicks on search button if keyword matches with any of the topic, then assign a variable called topic with values 1 or 2 or 3 appropriately. So, when user clicks the more images button use necessary control structures like if/switch to view all the images. If **topic=1**, show images of topic1 etc. Hint: Make use of **next and Previous** functionality from the **coursesDisplay** App that is explained in class.
5. Disable the button accordingly when user reaches the first and last images.
6. Disable the search button if the **searchTextField** is empty and enable if some text is entered
7. Make use of **editing changed** Action for the text field.
8. Search for “search not found image” and “welcome” images in any browser. Download one image you like and display “welcome” image whenever you launch the app. If user searches anything that is not matched with any keywords display the “search not found image” image.
9. when user clicks Reset button hide all the controls except the image view. Also, clear the variable values if created and assign “welcome” image to image view.

The final app will be like this:

 Graphical user interface, text, application, chat or text message

Description automatically generated Application

Description automatically generated with low confidence

**View for not matched search.**

**Logo, company name

Description automatically generated**

**View after clicking Reset Button.**

**Please submit your app as compressed file, your compressed files should contain Lastname\_SearchApp folder and Lastname\_SearchApp.xcodeproj file. Please check your submission by downloading the submitted file and rechecking in xcode.**