**MAD3115**

**Final Project**

| Milestone Check-In  @10 | Presentation @10 | Specification @40 | Navigation @10 | Design @10 | Individual @20 | Total |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

Create a native application for iPhone or iPad (or Universal) that tests the user knowledge of a certain topic.

Your application should have at least 6 questions hard-coded and each time the user starts the application, three of these questions are selected randomly.

For every question, user should select one option out of four options presented.

When the user answers the third question, the application should display the user’s score (in percentage and out of 3) and one of the following messages depending on their score.

1. Score 0/3, 1/3, 2/3: Message => “Please try again!”
2. Score 1/3: Message => “Good job!”
3. Score 2/3 Message => “Very Good!”
4. Score 3/3 Message => “Excellent work!”

The topic for the questions should be chosen carefully and all the questions should be meaningful. All the questions should be based on the topic you have chosen.

If the score of the test is, Option 1 (see above), user should be given an option to retake the quiz.

View Controller Source Code:

**import** UIKit

**class** ViewController: UIViewController

{

**@IBOutlet** **weak** **var** questionCounter: UILabel!

**@IBOutlet** **weak** **var** scoreLabel: UILabel!

**@IBOutlet** **weak** **var** politicalImages: UIImageView!

**@IBOutlet** **weak** **var** questionLabel: UILabel!

**@IBOutlet** **weak** **var** optionA: UIButton!

**@IBOutlet** **weak** **var** optionB: UIButton!

**@IBOutlet** **weak** **var** optionC: UIButton!

**@IBOutlet** **weak** **var** optionD: UIButton!

**let** allQuestions = QuestionList()

**var** questionNumber:Int = 0

**var** score:Double = 0

**var** selectedAnswer : Int = 0

**var** list = [Question]()

**var** randomlist = [Question]()

**override** **func** viewDidLoad()

{

**super**.viewDidLoad()

// Do any additional setup after loading the view.

generateNewNumbers()

updateQuestion()

}

**func** generateNewNumbers()

{

randomlist.removeAll()

**let** uniqueNumbers = Int.getUniqueRandomNumbers(min: 0, max: 5, count: 3)

**for** i **in** 0...2

{

randomlist.append(allQuestions.list[uniqueNumbers[i]])

}

}

**@IBAction** **func** answerPressed(\_ sender: UIButton)

{

print(sender.tag)

**if** sender.tag == randomlist[questionNumber].correctAnswer

{

score+=1

}

**else**

{

}

questionNumber+=1

updateQuestion()

}

**func** updateQuestion()

{

**if** questionNumber < randomlist.count

{

politicalImages.image = UIImage(named:(randomlist[questionNumber].questionImage))

questionLabel.text = randomlist[questionNumber].question

optionA.setTitle(randomlist[questionNumber].optionA, for: UIControl.State.normal)

optionB.setTitle(randomlist[questionNumber].optionB, for: UIControl.State.normal)

optionC.setTitle(randomlist[questionNumber].optionC, for: UIControl.State.normal)

optionD.setTitle(randomlist[questionNumber].optionD, for: UIControl.State.normal)

selectedAnswer = randomlist[questionNumber].correctAnswer

updateUI()

}

**else**

{

**if** (score == 3)

{

**let** alert = UIAlertController(title: "Excellent Work! Your score is \(score) Out of \(randomlist.count).you have completed the quiz with an Outstanding \((score/3)\*100) percent", message: "End Of Quiz", preferredStyle: .alert)

present(alert , animated: **true**, completion: **nil**)

}

**else** **if** (score == 2)

{

**let** alert = UIAlertController(title: "Very Good! Your score is \(score) Out of \(randomlist.count).you have completed the quiz with a \((score/3)\*100) percent", message: "Please Try Again", preferredStyle: .alert)

**let** restartAction = UIAlertAction(title: "Retake The Quiz", style: .default, handler: {action **in** **self**.restartQuiz()})

alert.addAction(restartAction)

present(alert , animated:**true** , completion: **nil**)

}

**else** **if** (score == 1)

{

**let** alert = UIAlertController(title: "Good Job! Your score is \(score) Out of \(randomlist.count).you have completed the quiz with a \((score/3)\*100) percent", message: "Please Try Again", preferredStyle: .alert)

**let** restartAction = UIAlertAction(title: "Retake The Quiz", style: .default, handler: { action **in** **self**.restartQuiz()})

alert.addAction(restartAction)

present(alert , animated: **true**,completion: **nil**)

}

**else** **if** (score == 0)

{

**let** alert = UIAlertController(title: "Please Try Again", message: "Your score is \(score) Out of \(randomlist.count).you have completed the quiz with a \((score/3)\*100) percent", preferredStyle: .alert)

**let** restartAction = UIAlertAction(title: "Retake The Quiz", style: .default, handler: {action **in** **self**.restartQuiz()})

alert.addAction(restartAction)

present(alert , animated: **true**,completion: **nil**)

}

}

}

**func** updateUI()

{

scoreLabel.text = "Score: \(score)"

questionCounter.text = " Q : \(questionNumber+1)/\(randomlist.count)"

}

**func** restartQuiz()

{

generateNewNumbers()

score = 0

questionNumber = 0

updateQuestion()

}

}

**extension** Int

{

**static** **func** getUniqueRandomNumbers(min: Int, max: Int, count: Int) -> [Int]

{

**var** set = Set<Int>()

**while** set.count < count

{

set.insert(Int.random(in: min...max))

}

**return** Array(set)

}

}

questions swift file:

import Foundation

class Question

{

let questionImage : String

let question : String

let optionA : String

let optionB : String

let optionC : String

let optionD : String

let correctAnswer:Int

init(image:String,questionText:String,choiceA:String,choiceB:String,choiceC:String,choiceD:String,answer:Int){

questionImage = image

question = questionText

optionA = choiceA

optionB = choiceB

optionC = choiceC

optionD = choiceD

correctAnswer = answer

}

questionList swift file:

import Foundation

class QuestionList

{

var list = [Question]()

init()

{

list.append(Question(image: "rajyasabha", questionText: "What Is The Name Of The Upperhouse In India?", choiceA: "Lok Sabha", choiceB: "Rajya Sabha", choiceC: "Parliment", choiceD: "Assembly", answer: 2))

list.append(Question(image: "vote", questionText: "The minimum age of the voter in India is?", choiceA: "16", choiceB: "21", choiceC: "23", choiceD: "18", answer: 4))

list.append(Question(image: "ottawa", questionText: "What Is The Capital Of Canada?", choiceA: "san fransico", choiceB: "ottawa", choiceC: "Toronto", choiceD: "vancouver", answer: 2))

list.append(Question(image: "melaniejoly", questionText: "Who Is The Current Foreign Minister Of Canada?" , choiceA: "melanie joly", choiceB: "anitha anand", choiceC: "Omar Alghabra", choiceD: "sophie trudeau", answer: 1))

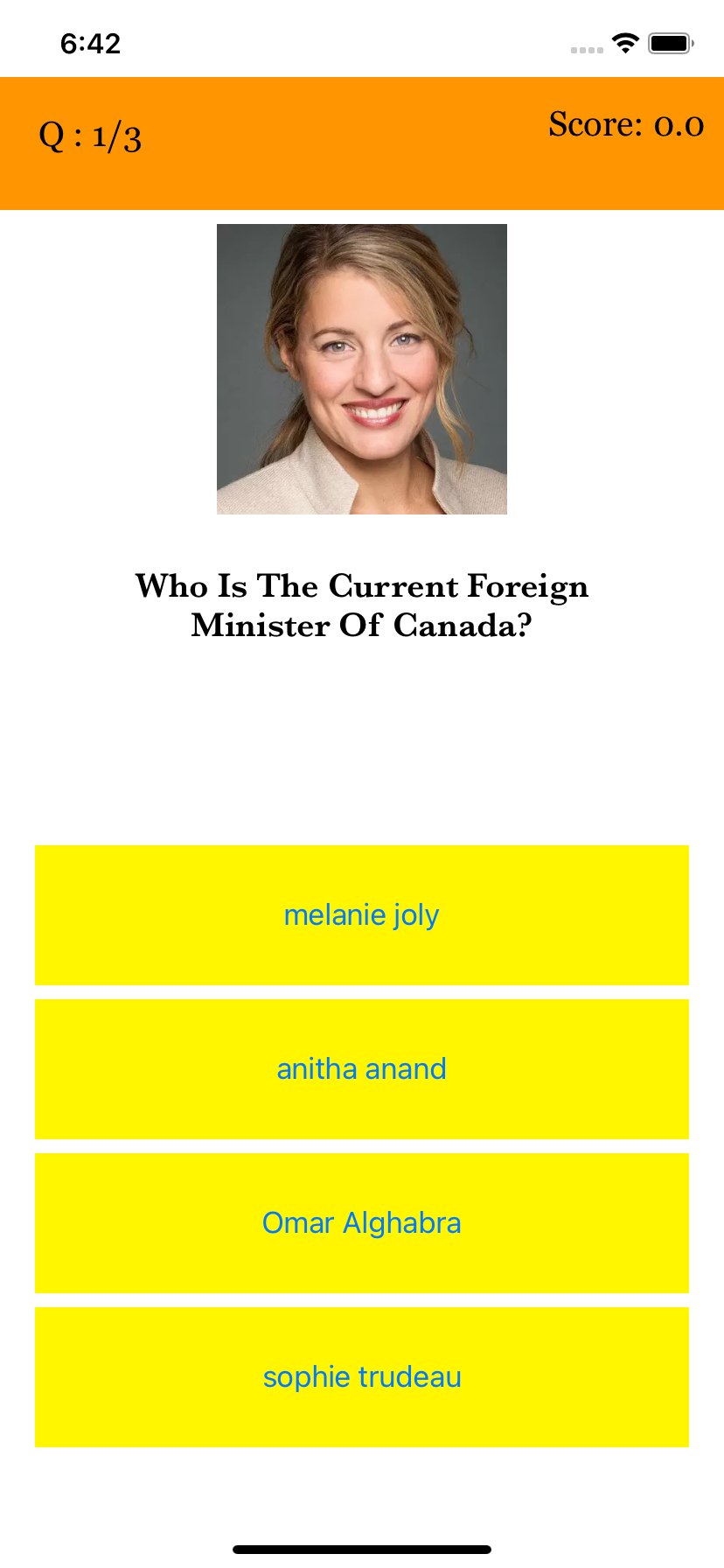
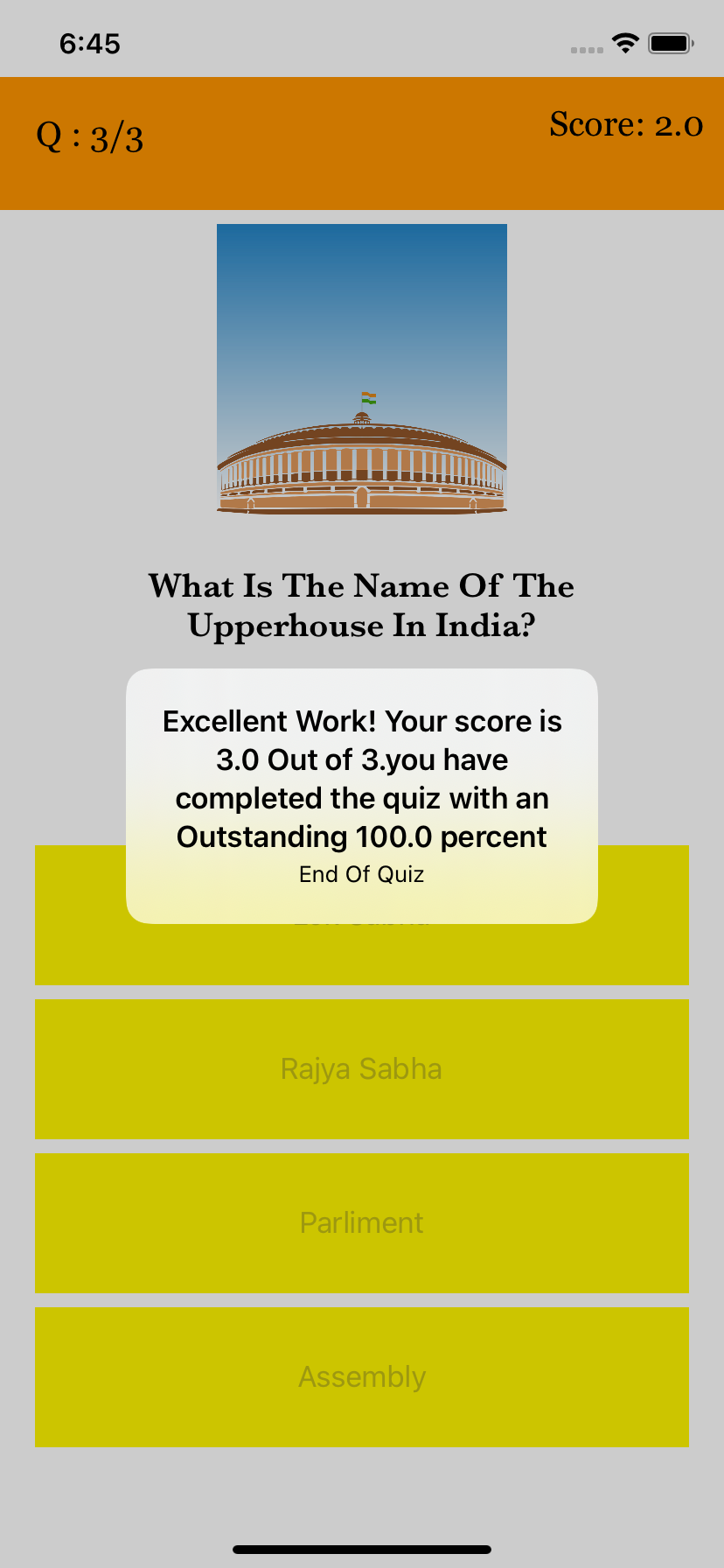
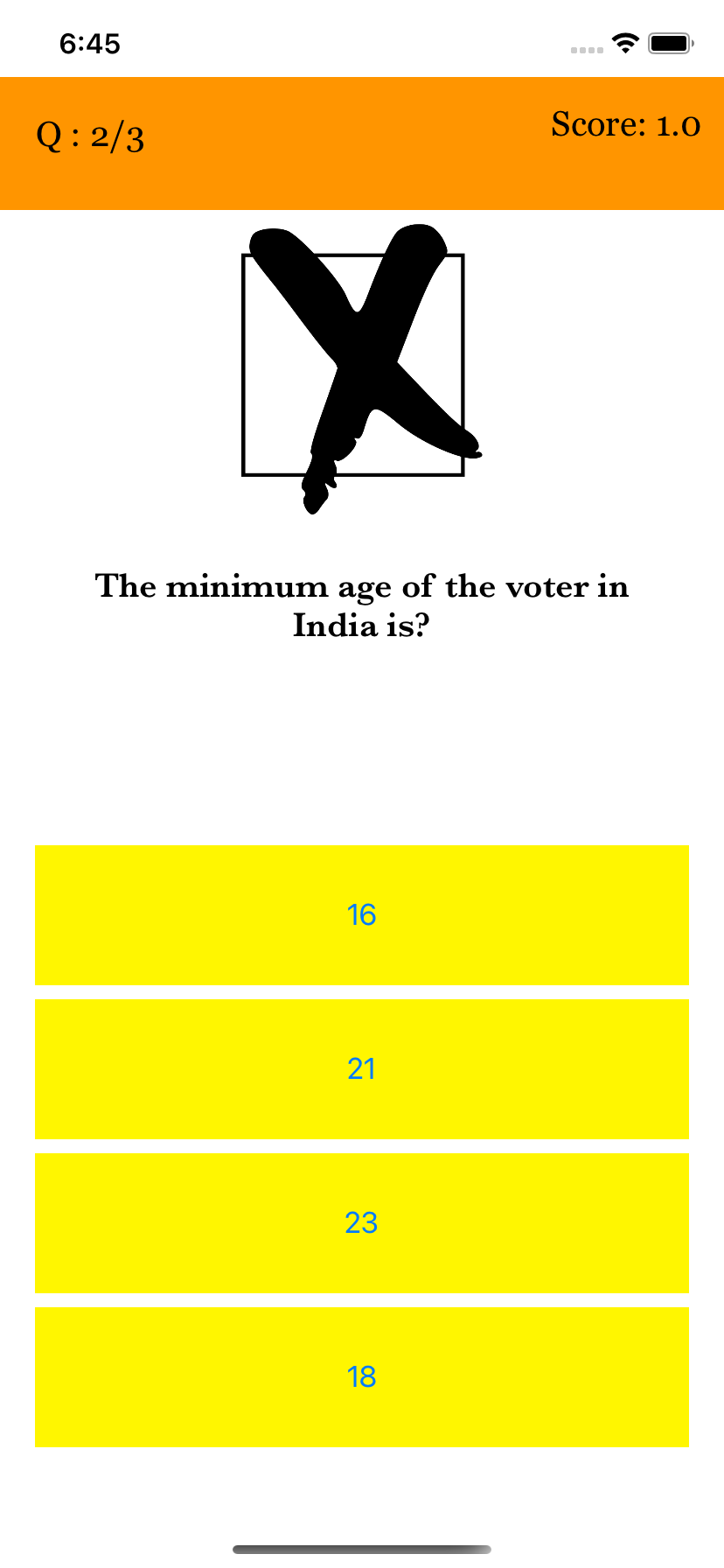
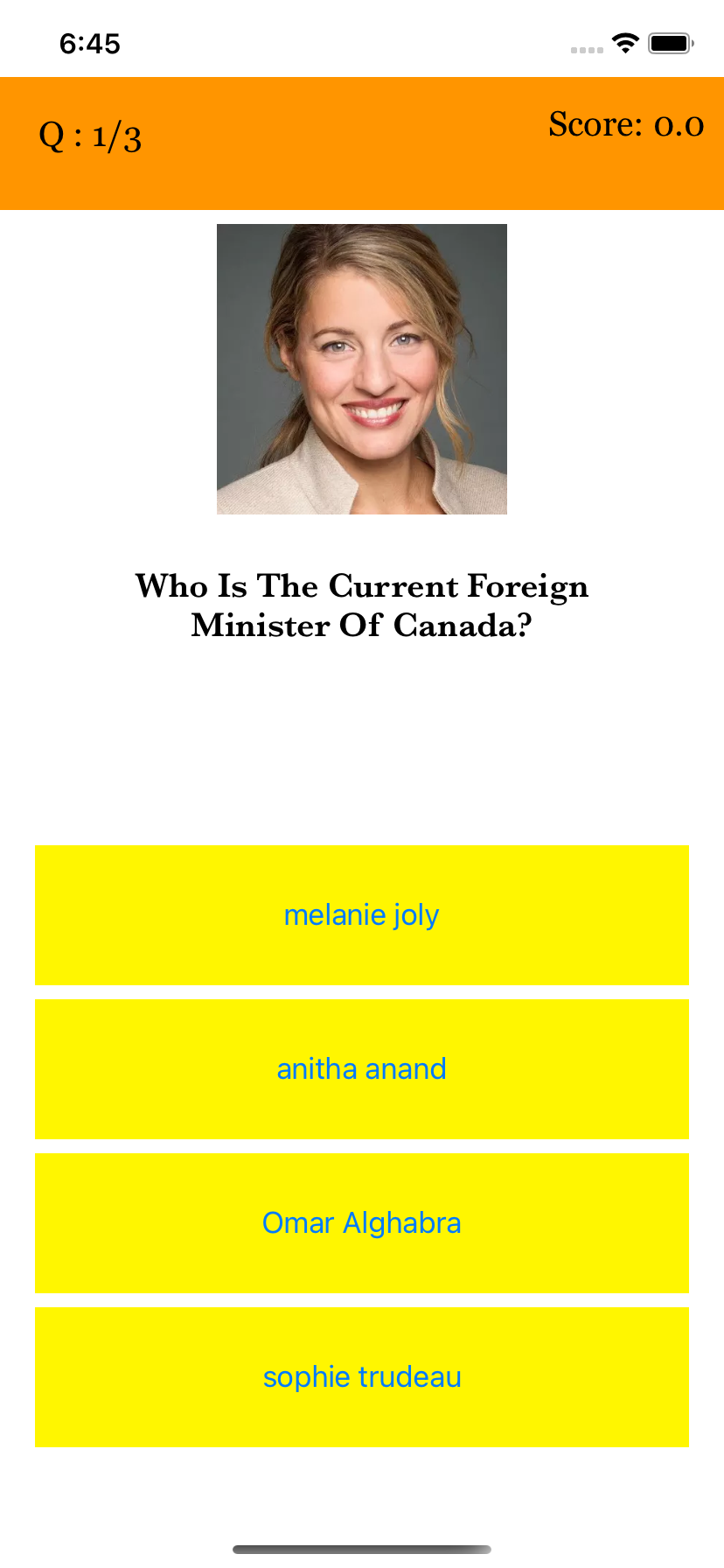
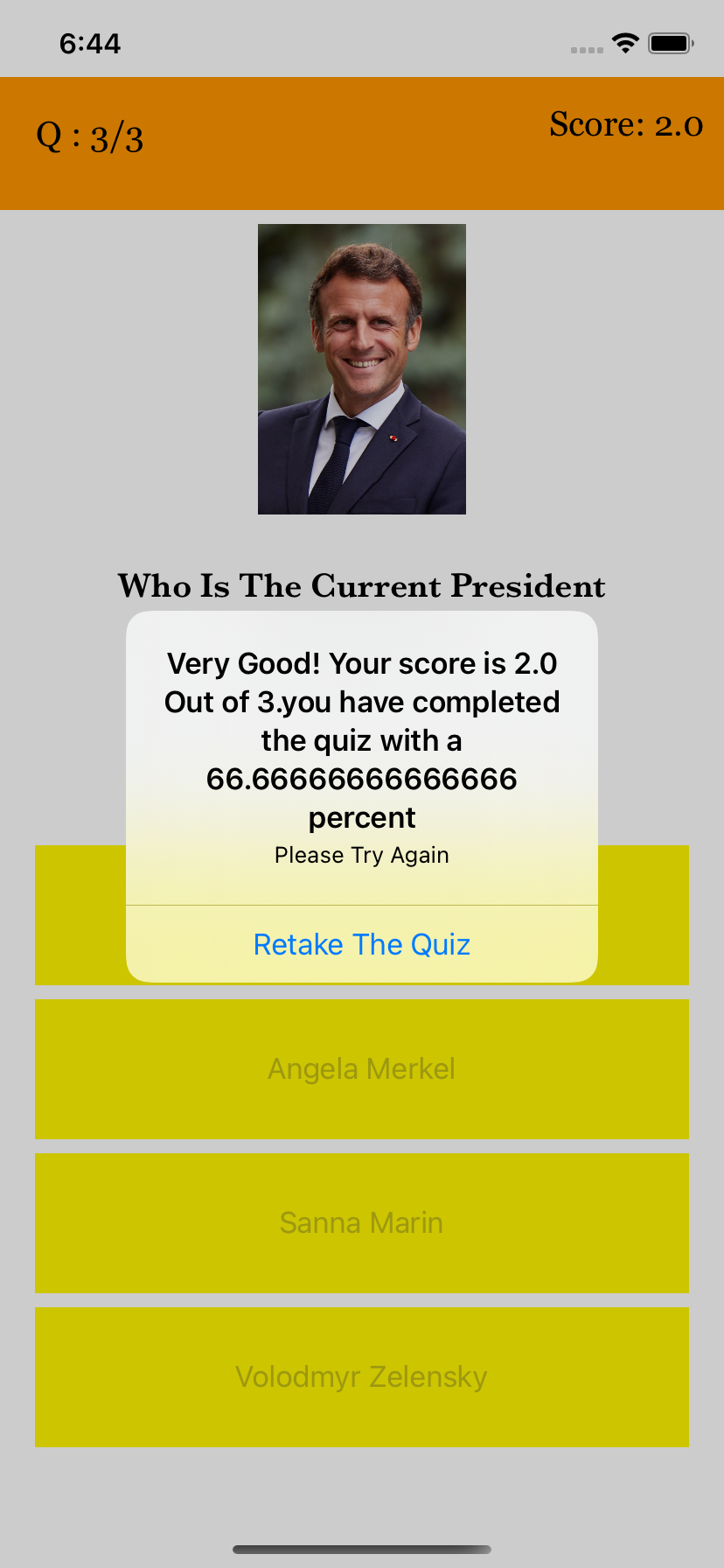
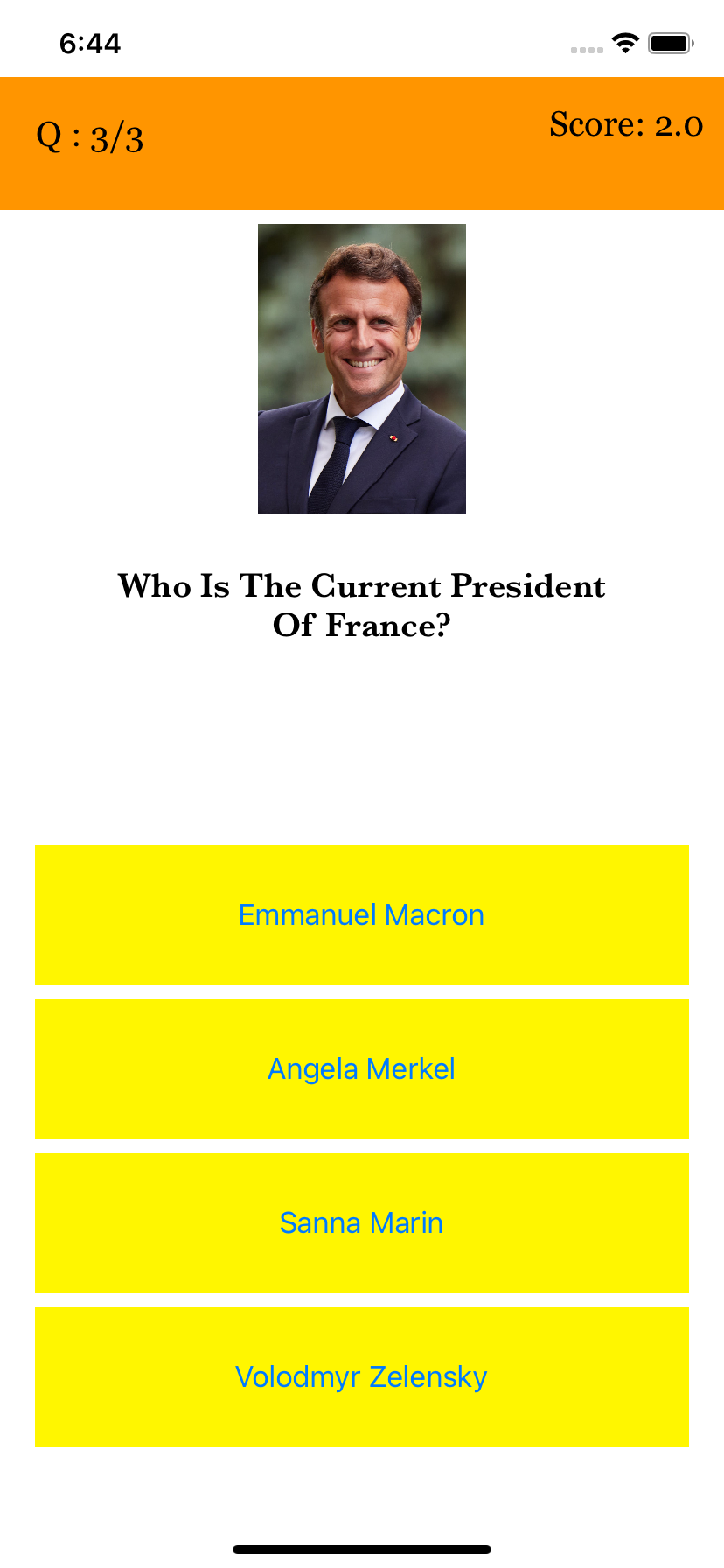
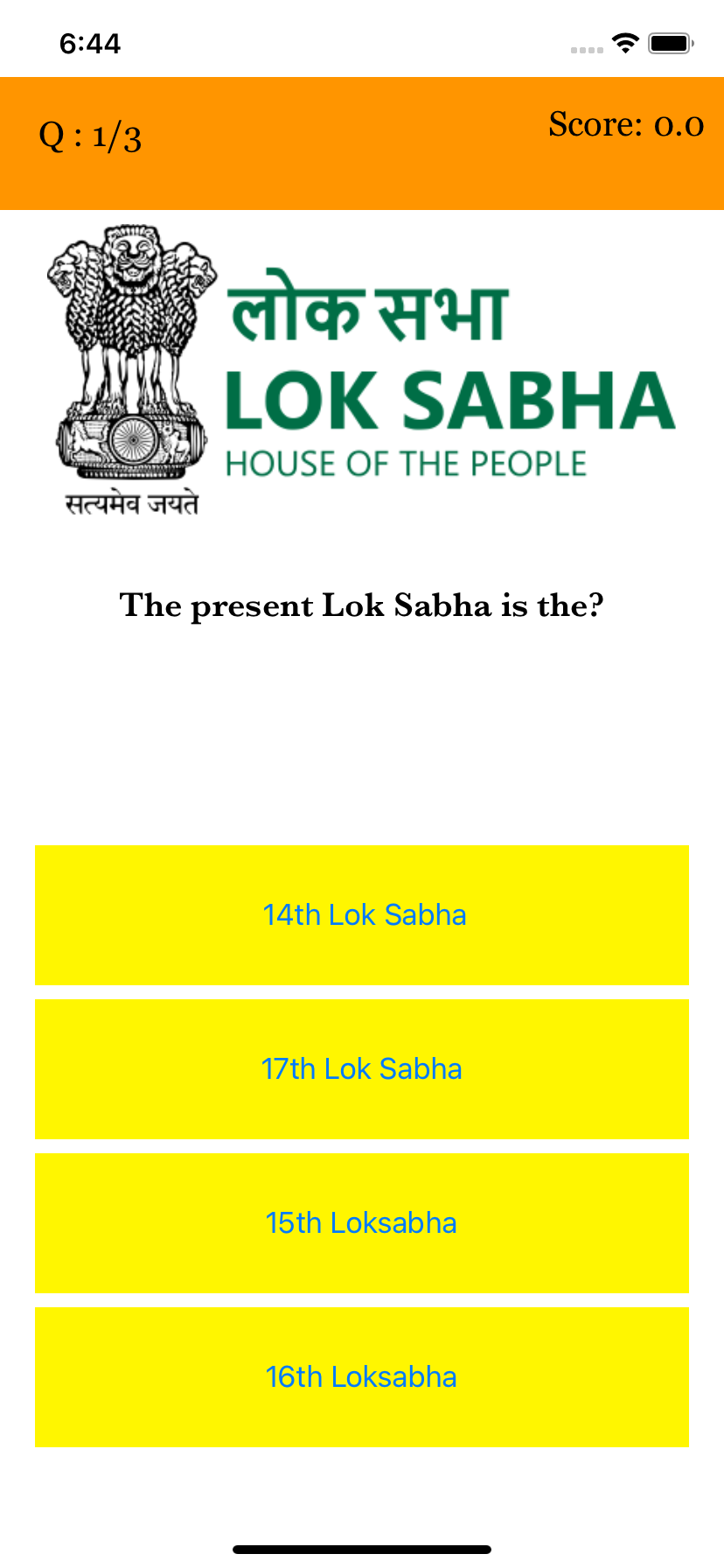
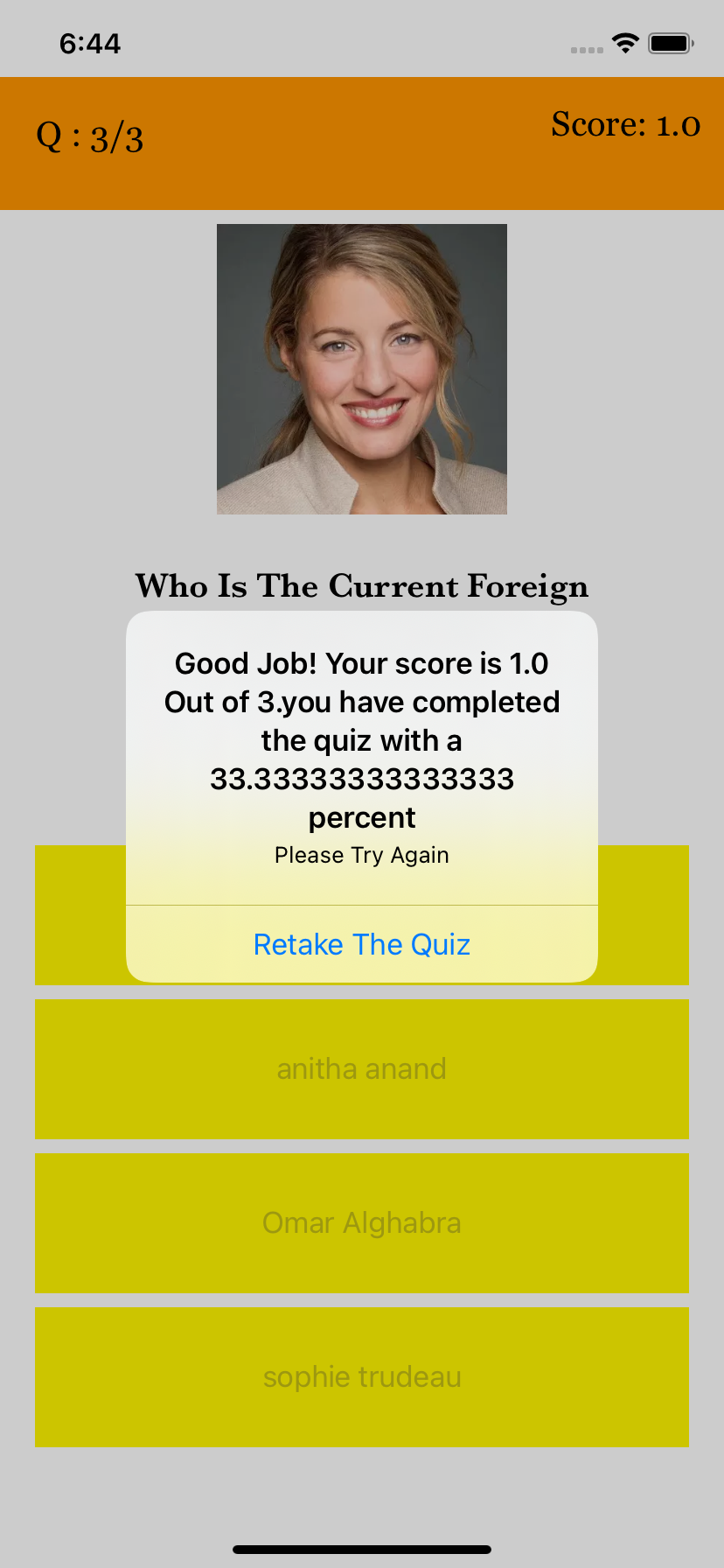
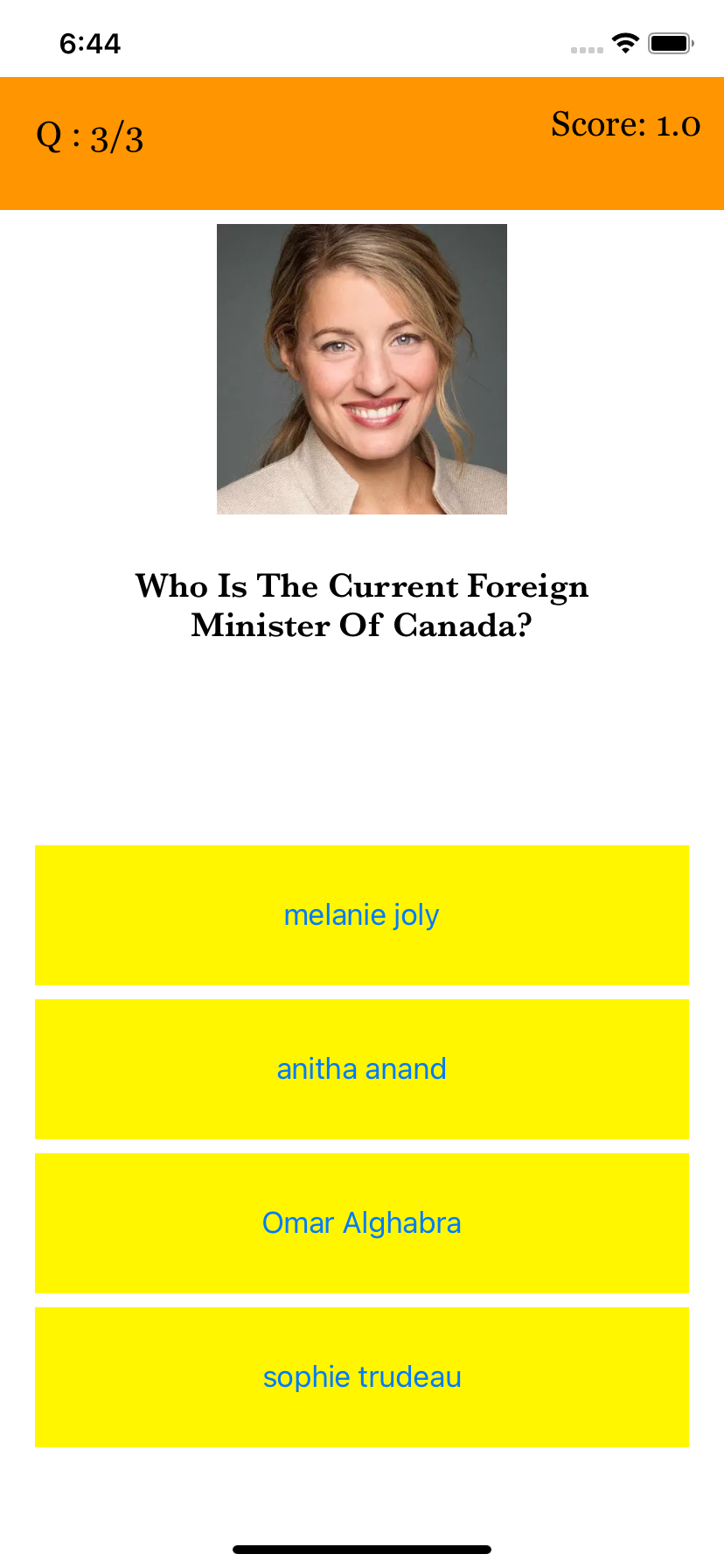
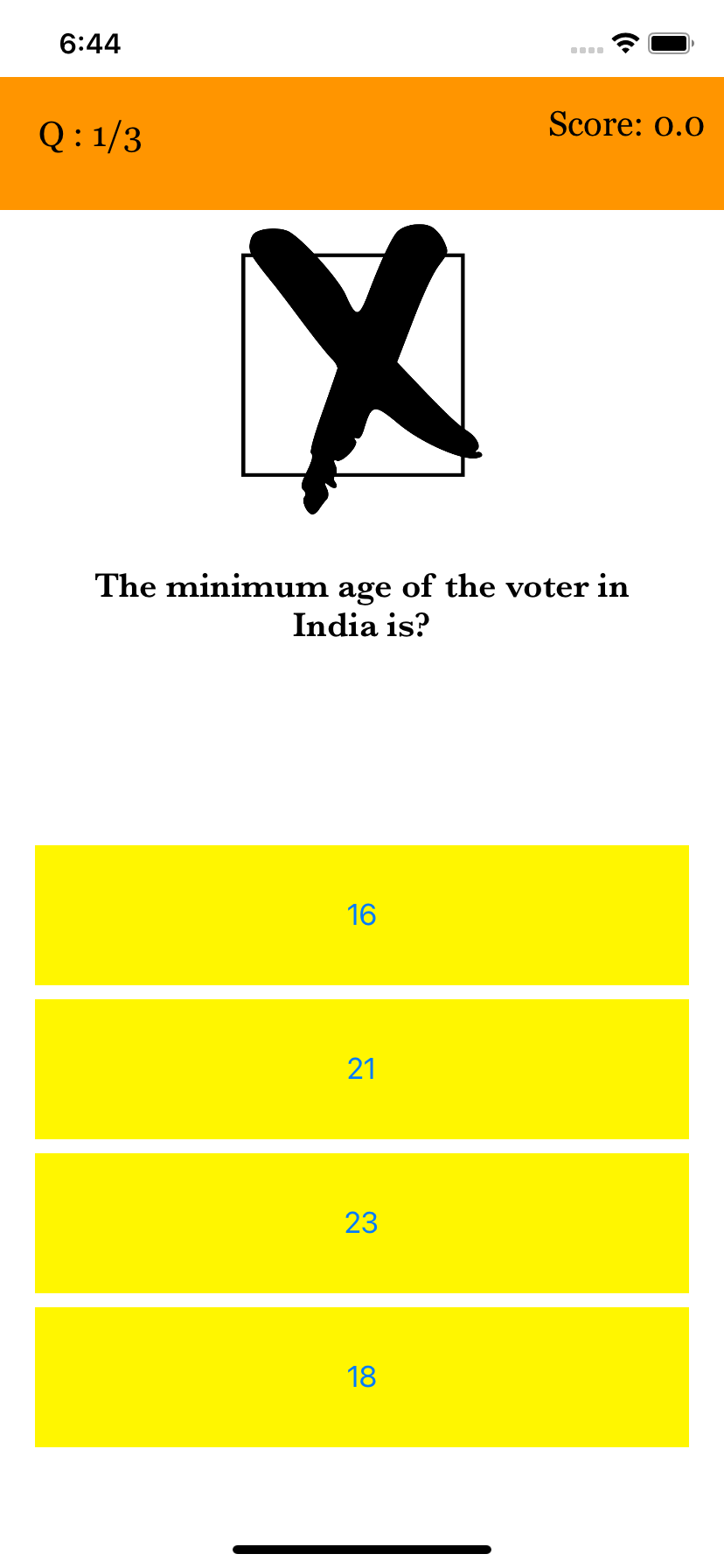
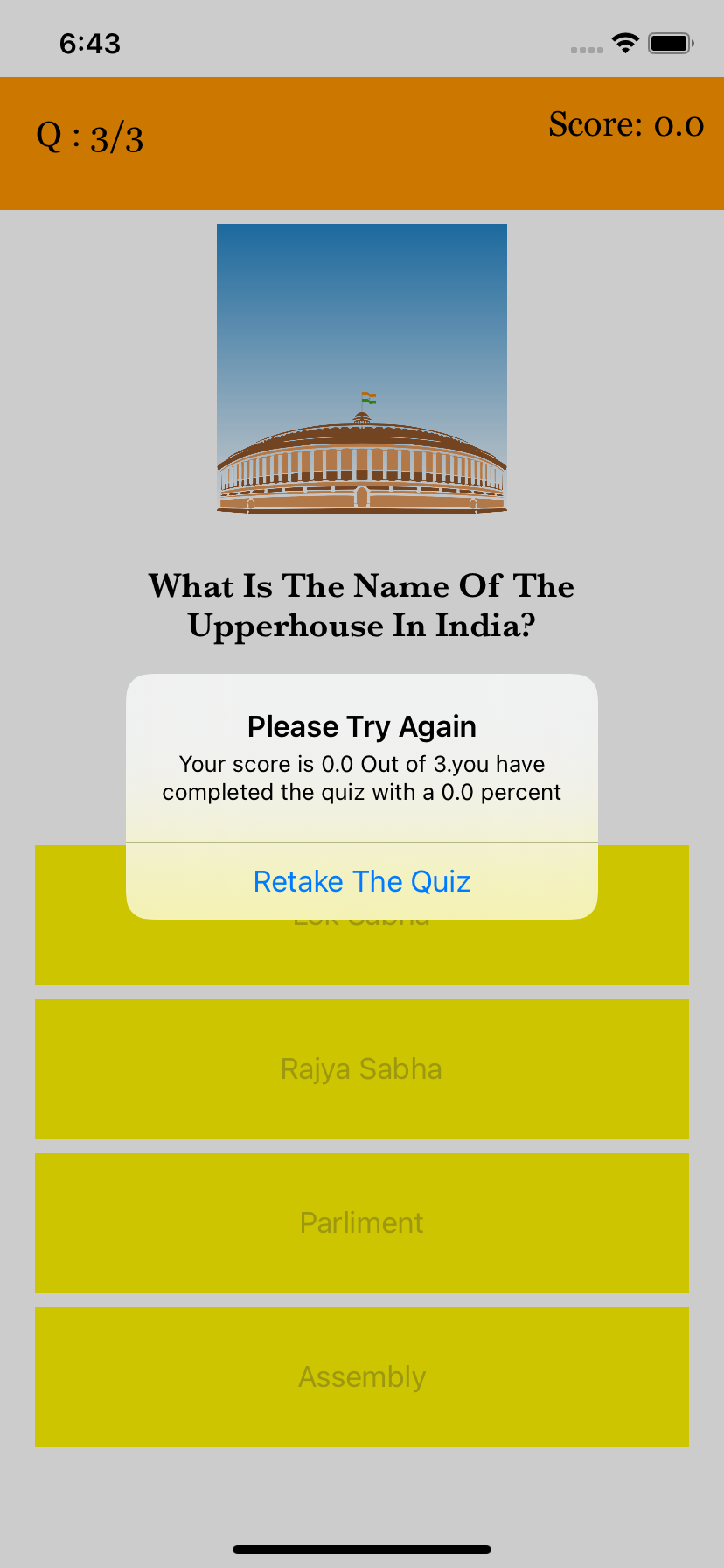
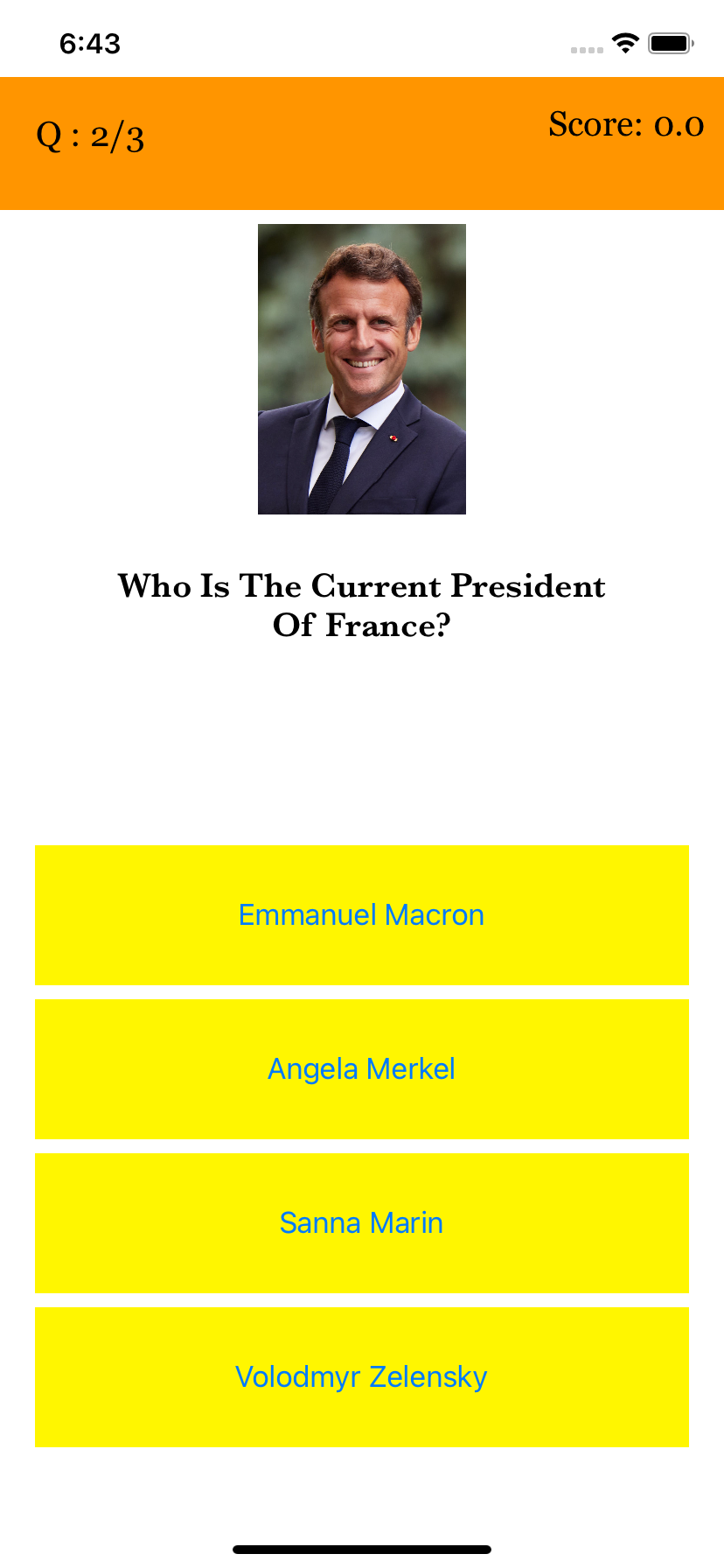
list.append(Question(image: "loksabha", questionText: "The present Lok Sabha is the?" , choiceA: " 14th Lok Sabha", choiceB: "17th Lok Sabha", choiceC: "15th Loksabha", choiceD: "16th Loksabha", answer: 2))

list.append(Question(image: "emmanuel", questionText: "Who Is The Current President Of France?" , choiceA: "Emmanuel Macron" , choiceB: "Angela Merkel", choiceC: "Sanna Marin", choiceD: "Volodmyr Zelensky", answer: 1))

**}**

**}**

**output screenshots:**

** **

**Marking Scheme:**

Marking of the assignment will be done according to the following scheme:

**Milestone Check-In [10%]:** This includes the short presentation halfway through the project to check how the group is performing and the progress of the project.

**Presentation [15%]:** This includes the group presentation of the assignment. Individual marks will be awarded for this rubric. Presentation should be professional and without any glitches.

**Specification [35%]:** This includes all the requirements specified above. Assignment will be considered for complete marks only if all the specifications are properly implemented. Partial implementation does not get awarded any marks.

**Navigation [10%]:** This rubric includes marks for user interface interaction component. The easier it is for the user to navigate the app the higher the mark is. This is somewhat subjective aspect of awarding a mark but for this assignment, good design practices covered in class will be used.

**Design [10%]:** This rubric focuses on overall design of the app which includes graphical user interface, images, polished look and proper file organization used for the app.

**Individual [20%]** – This mark is awarded to each member of the group individually based on their presentation and involvement in the project. Students should be prepared to answer questions by the professor.