

SettingsViewController:

```
import UIKit

class SettingsViewController: UIViewController {

    @IBOutlet var sideMenuBtn: UIBarButtonItem!

    override func viewDidLoad() {
        super.viewDidLoad()

        sideMenuBtn.target = self.revealViewController()
        sideMenuBtn.action = #selector(self.revealViewController()?.revealSideMenu)
    }
}
```

Country:

```
import Foundation
struct Country {
    var image: String = ""
    var name: String = ""
    var text: String = ""

    init(image: String, name: String, text: String){
        self.image = image
        self.name = name
        self.text = text
    }
}
```

CountryCollectionViewCell:

```
import Foundation
import UIKit

class CountryCollectionViewCell: UICollectionViewCell {
    @IBOutlet weak var countryImageView: UIImageView!
    @IBOutlet weak var countryNameLabel: UILabel!
}
```

CountryCollectionViewController:

```
import Foundation
import UIKit

private let reuseIdentifier = "Cell"

class CountryCollectionViewController: UICollectionViewController {

    @IBAction func unwindToMain(segue: UIStoryboardSegue){

    }

    private var countries : [Country] = [ Country(image: "America", name: "America", text: "1. What is the Official Languages for America? \n Answer:- English.\n\n 2. What is the Capital city for United States of America? \n Answer:- Washington, D.C.\n\n 3. What is the Currency for United States of America?? \n Answer:- American Doller."),
```

Country(image: "Australia", name: "Australia", text: "1. What is the Official Languages for Australia? \n Answer:- English.\n\n 2. What is the Capital city for United States of America? \n Answer:- Canberra.\n\n 3. What is the Currency for United States of America?? \n Answer:- Australian Doller."),

Country(image: "Austria", name: "Austria", text: "1. What is the Official Languages for Austria? \n Answer:- German.\n\n 2. What is the Capital city for Austria? \n Answer:- Vienna.\n\n 3. What is the Currency for Austria? \n Answer:- Euro."),

Country(image: "Bangladesh", name: "Bangladesh", text: "1. What is the Official Languages for Bangladesh? \n Answer:- Bengali.\n\n 2. What is the Capital city for Bangladesh? \n Answer:- Dhaka.\n\n 3. What is the Currency for Bangladesh? \n Answer:- Bangladeshi taka."),

Country(image: "Belgium", name: "Belgium", text: "1. What is the Official Languages for Belgium? \n Answer:- Dutch (north), French (south) and German.\n\n 2. What is the Capital city for Belgium? \n Answer:- Brussels.\n\n 3. What is the Currency for Belgium? \n Answer:- Euro."),

Country(image: "Canada", name: "Canada", text: "1. What is the Official Languages for Canada? \n Answer:- English and French.\n\n 2. What is the Capital city for Canada? \n Answer:- Ottawa.\n\n 3. What is the Currency for Canada? \n Answer:- Canadian Doller."),

Country(image: "China", name: "China", text: "1. What is the Official Languages for China? \n Answer:- Mandarin.\n\n 2. What is the Capital city for China? \n Answer:- Beijing.\n\n 3. What is the Currency for China? \n Answer:- Renminbi."),

Country(image: "Czech Republic", name: "Czech Republic", text: "1. What is the Official Languages for Czech Republic? \n Answer:- Czech.\n\n 2. What is the Capital city for Czech Republic? \n Answer:- Prague.\n\n 3. What is the Currency for United States of Czech Republic? \n Answer:- Czech koruna."),

Country(image: "France", name: "France", text: "1. What is the Official Languages for France? \n Answer:- French.\n\n 2. What is the Capital city for France? \n Answer:- Paris.\n\n 3. What is the Currency for France? \n Answer:- Euro, CFP franc."),

Country(image: "Germany", name: "Germany", text: "1. What is the Official Languages for Germany ? \n Answer:- German.\n\n 2. What is the Capital city for German? \n Answer:- Berlin.\n\n 3. What is the Currency for Germany ? \n Answer:- Euro."),

Country(image: "Hungary", name: "Hungary", text: "1. What is the Official Languages for Hungary? \n Answer:- Hungarian.\n\n 2. What is the Capital city for Hungary? \n Answer:- Budapest.\n\n 3. What is the Currency for Hungary? \n Answer:- Hungarian forint."),

Country(image: "India", name: "India", text: "1. What is the Official Languages for India? \n Answer:- Hindi and English.\n\n 2. What is the Capital city for India? \n Answer:- New Delhi.\n\n 3. What is the Currency for India? \n Answer:- Indian rupee."),

Country(image: "Korea", name: "Korea", text: "1. What is the Official Languages for Korea? \n Answer:- Korean.\n\n 2. What is the Capital city for Korea? \n Answer:- Seoul.\n\n 3. What is the Currency for Korea? \n Answer:- Korean won."),

Country(image: "Luxembourg", name: "Luxembourg", text: "1. What is the Official Languages for Luxembourg? \n Answer:- Luxembourgish, German and French.\n\n 2. What is the Capital city for Luxembourg? \n Answer:- Luxembourg.\n\n 3. What is the Currency for Luxembourg? \n Answer:- Euro."),

Country(image: "Netherlands", name: "Netherlands", text: "1. What is the Official Languages for Netherlands? \n Answer:- Dutch.\n\n 2. What is the Capital city for Netherlands? \n Answer:- Amsterdam.\n\n 3. What is the Currency for Netherlands? \n Answer:- Euro."),

Country(image: "Russia", name: "Russia", text: "1. What is the Official Languages for Russia? \n Answer:- Russian.\n\n 2. What is the Capital city for Russia? \n Answer:- Moscow.\n\n 3. What is the Currency for Russia? \n Answer:- Russian ruble."),

```
Country(image: "Slovakia", name: "Slovakia", text: "1. What is the Official Languages for Slovakia? \n Answer:- Slovak.\n\n 2. What is the Capital city for Slovakia? \n Answer:- Bratislava.\n\n 3. What is the Currency for Slovakia? \n Answer:- Euro."),
```

```
Country(image: "Switzerland", name: "Switzerland", text: "1. What is the Official Languages for Switzerland? \n Answer:- German, Romansh, French and Italian.\n\n 2. What is the Capital city for Switzerland? \n Answer:- Bern.\n\n 3. What is the Currency for Switzerland? \n Answer:- Swiss franc."),
```

```
Country(image: "taly", name: "Italy", text: "1. What is the Official Languages for Italy? \n Answer:- Italian.\n\n 2. What is the Capital city for Italy? \n Answer:- Rome.\n\n 3. What is the Currency for Italy? \n Answer:- Euro."),
```

```
Country(image: "UK", name: "United Kingdom", text: "1. What is the Official Languages for United Kingdom? \n Answer:- English.\n\n 2. What is the Capital city for United Kingdom? \n Answer:- London.\n\n 3. What is the Currency for United Kingdom? \n Answer:- Pound sterling."),
```

```
Country(image: "Vietnam", name: "Vietnam", text: "1. What is the Official Languages for Vietnam? \n Answer:- Vietnamese.\n\n 2. What is the Capital city for Vietnam? \n Answer:- Hanoi.\n\n 3. What is the Currency for Vietnam? \n Answer:- Vietnamese dong.")]
```

```
@IBOutlet var sideMenuBtn: UIButton!
```

```
override func viewDidLoad() {  
    super.viewDidLoad()  
  
    sideMenuBtn.target = self.revealViewController()  
    sideMenuBtn.action = #selector(self.revealViewController()?.revealSideMenu)  
}  
  
// MARK: UICollectionViewDataSource  
  
override func numberOfSections(in collectionView: UICollectionView) -> Int {  
    // #warning Incomplete implementation, return the number of sections  
    return 1  
}  
  
override func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {  
    // #warning Incomplete implementation, return the number of items  
    return countries.count  
}  
  
override func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {  
    let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "dataCell", for: indexPath) as! CountryCollectionViewCell  
  
    // Configure the cell  
  
    let country = countries[indexPath.row]  
    cell.countryImageView.image = UIImage(named: country.image)  
    cell.countryNameLabel.text = country.name  
  
    return cell  
}
```

```

override func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath)
{
    collectionView.deselectItem(at: indexPath, animated: false)
    performSegue(withIdentifier: "showDetail", sender: indexPath)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    guard segue.identifier == "showDetail" else {
        return
    }
    let indexPath = sender as! IndexPath
    let destinationController = segue.destination as! CountryDetailViewController
    destinationController.country = countries[indexPath.row]
    //collectionView.deselectItem(at: indexPaths[0], animated: false)
}
}

```

Country DetailViewController:

```

import Foundation
import UIKit

class CountryDetailViewController: UIViewController {

    @IBOutlet var countryImageView: UIImageView!
    @IBOutlet weak var countryTextView: UITextView!
    @IBOutlet weak var countryLabel: UILabel!

    var country: Country!

    override func viewDidLoad() {
        super.viewDidLoad()

        countryLabel.text = country.name
        countryTextView.text = country.text
        countryImageView.image = UIImage(named: country.image)
    }
}

```

ReceipesViewController:

```

import UIKit

class ReceipesViewController: UIViewController {

    @IBOutlet weak var sideMenuBtn: UIBarButtonItem!
    @IBOutlet weak var SouthIndianImg: UIImageView!
    @IBOutlet weak var NorthIndianImg: UIImageView!
    @IBOutlet weak var NoodlesImg: UIImageView!
    @IBOutlet weak var PizzaImg: UIImageView!
    @IBOutlet weak var CakeImg: UIImageView!
    @IBOutlet weak var BunImg: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

        self.sideMenuBtn.target = revealViewController()
    }
}

```

```

self.sideMenuBtn.action = #selector(self.revealViewController()?.revealSideMenu)

SouthIndianImg.isUserInteractionEnabled = true
NorthIndianImg.isUserInteractionEnabled = true
NoodlesImg.isUserInteractionEnabled = true
PizzaImg.isUserInteractionEnabled = true
CakeImg.isUserInteractionEnabled = true
BunImg.isUserInteractionEnabled = true

SouthIndianImg.layer.cornerRadius = 8
NorthIndianImg.layer.cornerRadius = 8
NoodlesImg.layer.cornerRadius = 8
PizzaImg.layer.cornerRadius = 8
CakeImg.layer.cornerRadius = 8
BunImg.layer.cornerRadius = 8

let southIndianRecognizer = UITapGestureRecognizer(target: self, action: #selector(openSouthIndian))
let northIndianRecognizer = UITapGestureRecognizer(target: self, action: #selector(openNorthIndian))
let noodlesRecognizer = UITapGestureRecognizer(target: self, action: #selector(openNoodles))
let pizzaRecognizer = UITapGestureRecognizer(target: self, action: #selector(openPizza))
let cakeRecognizer = UITapGestureRecognizer(target: self, action: #selector(openCake))
let bunRecognizer = UITapGestureRecognizer(target: self, action: #selector(openBun))

SouthIndianImg.addGestureRecognizer(southIndianRecognizer)
NorthIndianImg.addGestureRecognizer(northIndianRecognizer)
NoodlesImg.addGestureRecognizer(noodlesRecognizer)
PizzaImg.addGestureRecognizer(pizzaRecognizer)
CakeImg.addGestureRecognizer(cakeRecognizer)
BunImg.addGestureRecognizer(bunRecognizer)
}

@objc func openSouthIndian(){
    performSegue(withIdentifier: "mainPageToSouthIndia", sender: nil)
}

@objc func openNorthIndian(){
    performSegue(withIdentifier: "mainPageToNorthIndia", sender: nil)
}

@objc func openNoodles(){
    performSegue(withIdentifier: "mainPageToNoodles", sender: nil)
}

@objc func openPizza(){
    performSegue(withIdentifier: "mainPageToPizza", sender: nil)
}

@objc func openCake(){
    performSegue(withIdentifier: "mainPageToCake", sender: nil)
}

@objc func openBun(){
    performSegue(withIdentifier: "mainPageToBun", sender: nil)
}
}

```

SouthIndianKitchenViewController:

```
import UIKit
```

```
class SouthIndianKitchenViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
```

```
    var foodNames = [String]()  
    var foodImages = [UIImage]()  
    var chosenFoodName = ""  
    var chosenFoodImage = UIImage()
```

```
    @IBOutlet weak var worldTable: UITableView!
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        worldTable.delegate = self  
        worldTable.dataSource = self
```

```
        foodNames.append("Dosa")  
        foodNames.append("Idly")  
        foodNames.append("Puttu with Kadala Curry")  
        foodNames.append("Pulihora")
```

```
        foodImages.append(UIImage(named: "Dosa")!)  
        foodImages.append(UIImage(named: "Idly")!)  
        foodImages.append(UIImage(named: "Puttu with Kadala Curry")!)  
        foodImages.append(UIImage(named: "Pulihora")!)
```

```
        navigationItem.title = "SOUTH INDIAN"
```

```
    }
```

```
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
        return foodNames.count  
    }
```

```
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {  
        let cell = UITableViewCell()  
        cell.textLabel?.text = foodNames[indexPath.row]  
        return cell  
    }
```

```
    func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {  
        if editingStyle == .delete {  
            foodNames.remove(at: indexPath.row)  
            foodImages.remove(at: indexPath.row)  
            tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)  
        }  
    }
```

```
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
```

```
        chosenFoodName = foodNames[indexPath.row]  
        chosenFoodImage = foodImages[indexPath.row]
```

```

        performSegue(withIdentifier: "SouthIndianRecipes", sender: self)
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "SouthIndianRecipes" {
            let destinationVC = segue.destination as? SouthIndianRecipeViewController
            destinationVC?.selectedFoodName = chosenFoodName
            destinationVC?.selectedImage = chosenFoodImage
        }
    }
}

```

SouthIndianRecipeViewController:

```

import UIKit

class SouthIndianRecipeViewController: UIViewController {

    @IBOutlet weak var recipeName: UILabel!
    @IBOutlet weak var recipeImage: UIImageView!
    @IBOutlet weak var recipeText: UITextView!

    var selectedFoodName = ""
    var selectedImage = UIImage()
    var recipe = ""

    override func viewDidLoad() {
        super.viewDidLoad()

        recipeImage.layer.cornerRadius = 8

        recipeName.text = selectedFoodName
        recipeImage.image = selectedImage

        // self.navigationController?.popViewController(animated: true)
        // self.navigationController?.popToRootViewController(animated: true)

        // for obj in (self.navigationController?.viewControllers)! {
        //     if obj is SouthIndianKitchenViewController {
        //         self.navigationController?.popViewController(obj, animated: true)
        //         break
        //     }
        // }

        recipeText.setContentOffset(.zero, animated: true)

        if selectedFoodName == "Dosa" {
            recipe = "Dosa also called as dosai (in Tamil language) is a famous and popular South Indian breakfast or snack in India as well in the rest of the world. Dosa are basically crispy or soft crepes made with ground and fermented lentil and rice batter. To make the batter, first the lentils and rice are soaked in water for 4 to 5 hours. They are then ground separately to a fine consistency. Then both the lentil batter and the rice batter are mixed with salt in a wide bowl or pan. This batter is then allowed to ferment overnight or for 8 to 9 hours. After the batter is fermented, it increases in volume, has a pleasant fermented aroma and a slightly sour taste. A well-

```

fermented batter makes the best dosa. You will also see many tiny air-pockets in the batter. Making the batter and then fermenting it is such a joy that I always look forward to making fermented foods like dosa and Idli. The fermented batter is poured on a seasoned cast-iron pan or skillet (tawa) and spread like a pancake and cooked till crispy and golden with a drizzle of oil or ghee."

```
    recipeText.text = recipe
  }
  else if selectedFoodName == "Idly"{
    recipe = "Idly is one of the most healthiest and popular South Indian breakfast dish. These are soft, light, fluffy steamed round cake made with a ground, fermented rice and lentil batter. Here I share my foolproof recipe with video and step-by-step photos that will help you in making the best idli. This Idli recipe is one of the earliest recipe from the blog which has been tried and tested with great results by many of our readers. Idli is one of the most popular naturally fermented breakfast food. Mustard essential oil was found to be the best bio-preservative which showed a biocidal effect at 80 ppm against LAB strains and at 40 ppm against Candida versatilis. The efficacy of mustard essential oil incorporated in the idli batter at 0.1% (w/w) was evaluated by measuring the titratable acidity, pH, viscosity, batter volume, microbial count of idli batter as well as sensory parameters of idli prepared from preserved batter stored at 4 and 30 °C. Unfavorable changes in acidity, batter volume and whey separation of idli batter containing mustard oil were significantly reduced, which resulted in a reduction of sour taste and improved texture of idli. The growth of yeast and LAB was retarded evidenced by decreased microbial counts than control batter, which delayed the deterioration of the batter under both storage conditions. The addition of 0.1% mustard essential oil in the optimally fermented idli batter extended its shelf life to 5 days when stored at 30 °C and 30 days at 4 °C."
    recipeText.text = recipe
  }
  else if selectedFoodName == "Puttu with Kadala Curry"{
    recipe = "Ingredients for Puttu Puttu mavu (rice flour) – 2 cups, Salt, Hot water, Coconut (grated) Puttu – method of preparation Just add sufficient hot water and salt to the rice flour. Mix the flour well with fingers. For making the Puttu, you may use the traditional Puttu Kudam. You may begin by adding a spoonful of grated coconut into the cylindrical container, followed by the well mixed rice flour. As you add the rice flour into the container, in between, you may also add small quantities of grated coconut. Make sure that the cylinder is just fully filled and now you may place it on the Puttu Kudam after covering the top portion of the cylinder. Allow the steam to take care of the cooking. Your Puttu will be ready in three minutes. Have it with Kadala curry for a great combination. Ingredients for Kadala Curry, Black Channa (black chickpea / bengal gram) – 1 cup, Small onion (chopped) – 1 cup, Coconut (grated) – 1 cup, Coriander powder – 2 tsp, Red chilli powder – 1 tsp, Turmeric powder – ½ tsp, Garam masala – ½ tsp, Coconut (diced) – ¼ cup, Tomato (chopped) – ¼ cup, Curry leaves, Coriander leaves, Salt, Oil, Mustard seeds, Red chillies, Ginger-Garlic paste. Kadala Curry – method of preparation. First, soak the black channa (black chickpea / bengal gram) for at least ten hours by adding salt. Cook the black channa well and keep it aside. Heat one teaspoon of oil in a pan. Put the grated coconut and fry until it turns brown. To this, add coriander powder, chilli powder and garam masala. Stir all the ingredients until it becomes brown. Make a paste of this and keep it aside. Take a pan and heat some coconut oil. Put some mustard and add red chillies. When it begins to splutter, add the diced coconut. Stir a bit. Now add the ginger-garlic paste, followed by onions and tomatoes. Add some curry leaves and stir gently. You may now add the black channa (black chickpea). Pour sufficient water and add salt. Stir well. Now add the coconut paste kept aside to the cooked black channa. Boil the curry well. You may now garnish the curry with chopped coriander leaves and serve it ideally with Puttu. It is a great combination to relish."
    recipeText.text = recipe
  }
  else if selectedFoodName == "Pulihora"{
    recipe = "Pulihora recipe | chintapandu pulihora | how to make andhra style tamarind rice with step by step photo and video recipe. tamarind rice is a known south indian staple food which is hugely popular across indian states. each state in south india has its own variation of preparing it, with a different set of spices used in it and a different local name. one such easy and simple variation is pulihora recipe from the famous andhra cuisine. In south indian, there are 3 major variations to the tamarind rice variations. out of these, i have already posted famous puliyogare from karnataka and puliyodharai from tamilnadu. even though in english it is commonly referred to as tamarind rice, but has unique and different local names. apart from its names, it also differs in its preparation and also in its taste. out of these 3, i personally feel the andhra version has less use of spices in it. it focuses more on tamarind sauce rather than spices. whereas in the other 2 variations it is a combination of sweetness, spiciness and sourness from tamarind sauce. consequently, due to the less use of spices in this recipe, it has a light colour compared to the other 2 variations. it can be pale yellow in colour whereas, the other 2 offers you a dark red in colour. Furthermore, some of the easy and important tips,
```


suggestions and variations for pulihora recipe. firstly, i would heavily recommend preparing this recipe with short-grained rice like sona masuri or jasmine rice. avoid using basmati rice which may not perfectly gel with spices and tamarind sauce. secondly, avoid using freshly cooked rice for this recipe. use rested or cooled down rice which should not have moisture in it. the freshly cooked rice would have moisture in it and when mixed with spices and sauce, it would lose its concentration. lastly, you can always pre-make the spices and sauce concentration and use it whenever you wish to make this rice recipe."

```
        recipeText.text = recipe
    }
}
```

```
}
```

NorthIndianKitchenViewController:

```
import UIKit
```

```
class NorthIndianKitchenViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
```

```
    @IBOutlet weak var tableView: UITableView!
```

```
    var foodNames = [String]()
    var foodImages = [UIImage]()
    var chosenFoodName = ""
    var chosenFoodImage = UIImage()
```

```
    override func viewDidLoad() {
        super.viewDidLoad()
```

```
        tableView.delegate = self
        tableView.dataSource = self
```

```
        foodNames.append("Naan")
        foodNames.append("Butter Chicken")
        foodNames.append("Briyani")
        foodNames.append("Chole Bhature")
```

```
        foodImages.append(UIImage(named: "Naan")!)
        foodImages.append(UIImage(named: "Butter Chicken")!)
        foodImages.append(UIImage(named: "Briyani")!)
        foodImages.append(UIImage(named: "Chole Bhature")!)
```

```
        navigationItem.title = "NORTH INDIAN"
    }
```

```
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        foodNames.count
    }
```

```
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = UITableViewCell()
        cell.textLabel?.text = foodNames[indexPath.row]
        return cell
    }
```

```

func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt
indexPath: IndexPath) {
    if editingStyle == .delete {
        foodNames.remove(at: indexPath.row)
        foodImages.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)
    }
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    chosenFoodName = foodNames[indexPath.row]
    chosenFoodImage = foodImages[indexPath.row]

    performSegue(withIdentifier: "NorthIndianRecipes", sender: nil)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "NorthIndianRecipes" {
        let destinationVC = segue.destination as? NorthIndianRecipeViewController
        destinationVC?.selectedFoodName = chosenFoodName
        destinationVC?.selectedImage = chosenFoodImage
    }
}
}

```

NorthIndianRecipeViewController:

```

import UIKit

class NorthIndianRecipeViewController: UIViewController {
    @IBOutlet weak var recipeName: UILabel!
    @IBOutlet weak var recipeImage: UIImageView!
    @IBOutlet weak var recipeText: UITextView!

    var selectedFoodName = ""
    var selectedImage = UIImage()
    var recipe = ""

    override func viewDidLoad() {
        super.viewDidLoad()

        recipeImage.layer.cornerRadius = 8

        recipeName.text = selectedFoodName
        recipeImage.image = selectedImage

        if selectedFoodName == "Naan" {
            recipe = "Naan is an Indian recipe; it's a type of flat bread. Traditionally they are baked inside a very hot
            clay tandoor oven, with charcoal or wood fire. The naan bread are stuck to the inside of the oven just like the
            picture below. The tandoor oven retains very high heat, up to 900°F and hence fills the inside of the flat bread
            with air pockets and bakes them to perfection. In a modern day kitchen, we do not have a tandoor oven, but you
            can make naan at home with a hot cast-iron skillet on the stove top, or bake in the oven. Naan recipe is made on
            a stove top, using a very hot cast-iron skillet. The naan turn out extremely delicious, soft, puffy, airy, with the
            beautiful golden brown spots on the surface. They taste just like the best Indian buffet restaurants!"
            recipeText.text = recipe
        }
    }
}

```

```

    else if selectedFoodName == "Butter Chicken" {
        recipe = "First, mix your yogurt marinade together. Marinate chicken for 20 minutes, or overnight if time allows. (The longer it marinates, the more tender your chicken will be. Sear chicken in a skillet or pan. Keep those charred bits on the pan to make your sauce. Browned bits = flavour. If you have too many burnt pieces stuck to the pan, you can discard some of them if you wish. Finally, make your curry sauce in the same skillet."
        recipeText.text = recipe
    }
    else if selectedFoodName == "Briyani" {
        recipe = "Soak 300g basmati rice in warm water, then wash in cold until the water runs clear. Heat 25g butter in a saucepan and cook 1 finely sliced large onion with 1 bay leaf, 3 cardamom pods and 1 small cinnamon stick for 10 mins. Sprinkle in 1 tsp turmeric, then add 4 chicken breasts, cut into large chunks, and 4 tbsp curry paste. Cook until aromatic. Stir the rice into the pan with 85g raisins, then pour over 850ml chicken stock. Place a tight-fitting lid on the pan and bring to a hard boil, then lower the heat to a minimum and cook the rice for another 5 mins. Turn off the heat and leave for 10 mins. Stir well, mixing through 15g chopped coriander. To serve, scatter over the leaves of the remaining 15g coriander and 2 tbsp toasted almonds."
        recipeText.text = recipe
    }
    else if selectedFoodName == "Chole Bhature" {
        recipe = "Less all-purpose flour (maida, refined white flour) and more whole wheat flour (atta) – The ratio of all purpose flour to whole wheat flour used in this recipe is approximate 1:3 respectively. Some of the other bhature recipes are made of only all purpose flour. Though you can make this recipe completely with either all purpose flour or whole wheat flour. Made without yeast – The leavening is done naturally and similar to a sourdough starter. But here the starter has curd (yogurt) and baking powder or baking soda. This technique is the traditional Indian method to draw natural yeast from the air which helps in the leavening of the dough. In Hindi language it is also called as making “khameer” – where the word khameer refers to yeast. The bacteria culture in the curd (yogurt) also helps in fermentation. Fermented dough – The flour starter is allowed to leaven and ferment overnight. So the dough has the typical aroma and sour notes of fermented food. The rolling and frying of the bread takes around 20 to 30 minutes. But the leavening of the flour mixture, takes 8 hours (overnight) plus 2 hours. The time also depends on the temperature conditions. So keep the starter and the dough in a warm place. Fluffy and crisp – Since the dough is leavened and fermented, the bhature has a nice fluffy texture from within and puffs up really well in oil. Even though whole wheat flour is used in the recipe, you won't feel the difference – the overnight fermentation help in giving a good soft texture to the bhatura. Stays soft even after cooling – Usually, bhature becomes chewy after cooling at room temperature. But these bhature stay soft."
        recipeText.text = recipe
    }
}
}

```

NoodlesKitchenViewController:

```

import UIKit

class NoodlesKitchenViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    @IBOutlet weak var hamurTableView: UITableView!

    var foodNames = [String]()
    var foodImages = [UIImage]()

    var chosenFood = ""
    var chosenImage = UIImage()

    override func viewDidLoad() {
        super.viewDidLoad()

        hamurTableView.delegate = self
        hamurTableView.dataSource = self
    }
}

```

```

    foodNames.append("Indian Style Noodles")
    foodNames.append("Japanese Style Noodles")
    foodNames.append("Korean Style Noodles")
    foodNames.append("Chinese Style Noodles")

    foodImages.append(UIColor(named: "Main Noodles")!)
    foodImages.append(UIColor(named: "Japanese Noodles")!)
    foodImages.append(UIColor(named: "Korean Noodles")!)
    foodImages.append(UIColor(named: "Chinese Noodles")!)

    navigationItem.title = "Noodles"
}
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return foodNames.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = UITableViewCell()
    cell.textLabel?.text = foodNames[indexPath.row]
    return cell
}

func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        foodNames.remove(at: indexPath.row)
        foodImages.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)
    }
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    chosenFood = foodNames[indexPath.row]
    chosenImage = foodImages[indexPath.row]

    performSegue(withIdentifier: "NoodlesRecipes", sender: nil)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "NoodlesRecipes" {
        let destinationVC = segue.destination as? NoodlesRecipeViewController
        destinationVC?.selectedFood = chosenFood
        destinationVC?.selectedImage = chosenImage
    }
}
}

```

NoodlesRecipeViewController:

```

import UIKit

class NoodlesRecipeViewController: UIViewController {

    @IBOutlet weak var recipeName: UILabel!
    @IBOutlet weak var recipeImage: UIImageView!
    @IBOutlet weak var recipeText: UITextView!

    var selectedFood = ""
}

```

```

var selectedImage = UIImage()
var recipe = ""

override func viewDidLoad() {
    super.viewDidLoad()

    recipeImage.layer.cornerRadius = 8
    recipeImage.inputView?.contentMode = UIView.ContentMode.scaleAspectFill

    recipeName.text = selectedFood
    recipeImage.image = selectedImage

    if selectedFood == "Indian Style Noodles"{
        recipe = "Bring a pot of water to a boil. Add two slabs of Hakka noodles. Boil till the noodles are 90% done. This will approximately take 3 to 4 minutes. The noodles should break easily but not be pasty. Strain the noodles. Drizzle 2 tablespoons of oil. Coat the noodles with oil so that they don't stick to each other. Add 2 tablespoons oil to a pan or wok set on high heat. Add onions and give it a stir. Add ginger, garlic and stir fry for 30 seconds. Add carrots and French green beans and stir fry for one minute. Add bell pepper (capsicum) and stir fry for 10 seconds. Add cabbage and stir fry for 10 seconds. Add the pre-boiled noodles and mix well. Drizzle the sauce mix and combine evenly. Add salt to taste and stir fry for five minutes approximately. Add powder black pepper and stir fry for five more minutes. Garnish with spring onion greens."
        recipeText.text = recipe
    }
    else if selectedFood == "Japanese Style Noodles"{
        recipe = "Bring a large pot of lightly salted water to a boil. Cook udon in boiling water, stirring occasionally, until noodles are tender yet firm to the bite, 10 to 12 minutes. Drain and rinse with cold water. Stir in a few drops of sesame oil. Heat the remaining sesame oil in a large skillet over medium heat. Cook broccoli until bright green and still crunchy, about 5 minutes. Add green bell pepper and carrots; cook and stir until slightly softened, about 2 minutes. Add zucchini; cook until slightly softened, about 2 minutes more. Add soy sauce, mirin, chili-garlic sauce, and ginger; stir to combine. Mix in the noodles; cook and stir until noodles absorb some of the sauce, 1 to 2 minutes more."
        recipeText.text = recipe
    }
    else if selectedFood == "Korean Style Noodles"{
        recipe = "Slice the Beef & Veggies: Using a sharp knife, slice the beef into thin strips and chop your veggies. Prepare the Korean noodles according to the instructions on the package. Saute the Veggies: In a wok, saute the mushrooms. Set them aside. Next, saute the carrots. Set them aside. Prepare the Sauce: Whisk together the sauce ingredients in a small bowl and set aside. Saute the Beef: In the same wok, saute the beef. After the beef forms a crust, pour the sauce over the top and let simmer for a couple of minutes. Stir Fry the Ingredients: Add the noodles and veggies back into the wok. Toss in some green onion and stir to combine. Add in the spinach and stir fry the mixture for 5 minutes. "
        recipeText.text = recipe
    }
    else if selectedFood == "Chinese Style Noodles"{
        recipe = "Cook the noodles a little less than stated on the package instructions (until just underdone). Freshly cooked Chinese noodles being lifted from a pot. Drain and rinse with cold water and set aside. Heat the oil in the wok over a high heat. Once hot, add the garlic and ginger. The ginger added to the garlic and oil in the wok. Stir-fry for 30 seconds before adding the green onions and stir-fry vegetables. stir fry vegetables added to the wok with the onions garlic and spring onions Stir-fry for 2-3 minutes. Add the sherry and allow to almost evaporate before adding any meat you are using, and cooked noodles. Toss so that all of the ingredients are combined. Perfect Chinese Noodles cooking in the wok Turn the heat down to medium and add both of the soy sauces, oyster sauce, and pepper. Stir-fry for a further 1-2 mins before adding the remaining green onions, then toss to combine and serve."
        recipeText.text = recipe
    }
}
}

```

PizzaKitchenViewController:

```

import UIKit

class PizzaKitchenViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    @IBOutlet weak var mezeTableView: UITableView!

    var chosenFood = ""
    var chosenImage = UIImage()

    var foodNames = [String]()
    var foodImages = [UIImage]()

    override func viewDidLoad() {
        super.viewDidLoad()

        mezeTableView.delegate = self
        mezeTableView.dataSource = self

        foodNames.append("Home Made Pizza")
        foodNames.append("Margherita")
        foodNames.append("BBQ Chicken Pizza")
        foodNames.append("Pepperoni Pizza")

        foodImages.append(UIImage(named:"Home Made Pizza")!)
        foodImages.append(UIImage(named:"Margherita")!)
        foodImages.append(UIImage(named:"BBQ Chicken Pizza")!)
        foodImages.append(UIImage(named:"Pepperoni")!)

        navigationItem.title = "Pizza"
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return foodNames.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = UITableViewCell()
        cell.textLabel?.text = foodNames[indexPath.row]
        return cell
    }

    func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
        if editingStyle == .delete {
            foodNames.remove(at: indexPath.row)
            foodImages.remove(at: indexPath.row)
            tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)
        }
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        chosenFood = foodNames[indexPath.row]
        chosenImage = foodImages[indexPath.row]

        performSegue(withIdentifier: "PizzaRecipes", sender: nil)
    }
}

```

```

    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "PizzaRecipes" {
            let destinationVC = segue.destination as? PizzaRecipeViewController
            destinationVC?.selectedFood = chosenFood
            destinationVC?.selectedImage = chosenImage
        }
    }
}

```

PizzaRecipeViewController:

```
import UIKit
```

```
class PizzaRecipeViewController: UIViewController {
```

```

    @IBOutlet weak var recipeName: UILabel!
    @IBOutlet weak var recipeImage: UIImageView!
    @IBOutlet weak var recipeText: UITextView!

```

```

    var selectedFood = ""
    var selectedImage = UIImage()
    var recipe = ""

```

```

    override func viewDidLoad() {
        super.viewDidLoad()

```

```

        recipeImage.layer.cornerRadius = 8
        recipeImage.inputView?.contentMode = UIView.ContentMode.scaleAspectFill

```

```

        recipeName.text = selectedFood
        recipeImage.image = selectedImage

```

```

        if selectedFood == "Home Made Pizza" {
            recipe = "Place the sugar, salt, olive oil, water, 1 cup of flour, yeast, and remaining cup of flour into a standing mixer's work bowl. Using the paddle attachment, start the mixer on low and mix until the dough just comes together, forming a ball. Lube the hook attachment with cooking spray. Attach the hook to the mixer and knead for 15 minutes on medium speed. Tear off a small piece of dough and flatten into a disc. Stretch the dough until thin. Hold it up to the light and look to see if the baker's windowpane, or taut membrane, has formed. If the dough tears before it forms, knead the dough for an additional 5 to 10 minutes. Roll the pizza dough into a smooth ball on the countertop. Place into a stainless steel or glass bowl. Add 2 teaspoons of olive oil to the bowl and toss to coat. Cover with plastic wrap and refrigerate for 18 to 24 hours. Place the pizza stone or tile onto the bottom of a cold oven and turn the oven to its highest temperature, about 500 degrees F. If the oven has coils on the oven floor, place the tile onto the lowest rack of the oven. Split the pizza dough into 2 equal parts using a knife or a dough scraper. Flatten into a disk onto the countertop and then fold the dough into a ball. Wet hands barely with water and rub them onto the countertop to dampen the surface. Roll the dough on the surface until it tightens. Cover one ball with a tea towel and rest for 30 minutes. Repeat the steps with the other piece of dough. If not baking the remaining pizza immediately, spray the inside of a zip top bag with cooking spray and place the dough ball into the bag. Refrigerate for up to 6 days. Sprinkle the flour onto the peel and place the dough onto the peel. Using your hands, form a lip around the edges of the pizza. Stretch the dough into a round disc, rotating after each stretch. Toss the dough in the air if you dare. Shake the pizza on the peel to be sure that it will slide onto the pizza stone or tile. (Dress and bake the pizza immediately for a crisp crust or rest the dough for 30 minutes if you want a chewy texture. Brush the rim of the pizza with olive oil. Spread the pizza sauce evenly onto the pizza. Sprinkle the herbs onto the pizza and top with the cheese. Slide the pizza onto the tile and bake for 7 minutes, or until bubbly and golden brown. Rest for 3 minutes before slicing."

```

```

            recipeText.text = recipe
        }
        else if selectedFood == "Margherita" {

```



```

var foodImages = [UIImage]()
var chosenFood = ""
var chosenImage = UIImage()

override func viewDidLoad() {
    super.viewDidLoad()

    tatliTableView.delegate = self
    tatliTableView.dataSource = self

    foodNames.append("Butter Cake")
    foodNames.append("Pound Cake")
    foodNames.append("Sponge Cake")
    foodNames.append("Biscuit Cake")

    foodImages.append(UIImage(named: "Butter Cake")!)
    foodImages.append(UIImage(named: "Pound Cake")!)
    foodImages.append(UIImage(named: "Sponge Cake")!)
    foodImages.append(UIImage(named: "Biscuit Cake")!)

    navigationItem.title = "Cake"
}
@objc func addButtonClicked(){
}

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return foodNames.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = UITableViewCell()
    cell.textLabel?.text = foodNames[indexPath.row]
    return cell
}

func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        foodNames.remove(at: indexPath.row)
        foodImages.remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)
    }
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    chosenFood = foodNames[indexPath.row]
    chosenImage = foodImages[indexPath.row]
    performSegue(withIdentifier: "CakeRecipes", sender: nil)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "CakeRecipes" {
        let destinationVC = segue.destination as? CakeRecipeViewController
        destinationVC?.selectedFood = chosenFood
        destinationVC?.selectedImage = chosenImage
    }
}

```

CakeRecipeViewController:

```
import UIKit
```

```
class CakeRecipeViewController: UIViewController {
```

```
    @IBOutlet weak var recipeName: UILabel!  
    @IBOutlet weak var recipeImage: UIImageView!  
    @IBOutlet weak var recipeText: UITextView!
```

```
    var selectedFood = ""  
    var selectedImage = UIImage()  
    var recipe = ""
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        recipeImage.layer.cornerRadius = 8  
        recipeImage.inputView?.contentMode = UIView.ContentMode.scaleAspectFill
```

```
        recipeName.text = selectedFood  
        recipeImage.image = selectedImage
```

```
        if selectedFood == "Butter Cake" {  
            recipe = "Unsalted butter, All purpose flour, Sugar, Eggs, Yogurt or Milk, Baking powder, Vanilla  
extract. Use a high quality butter. I like Challenge brand unsalted butter. Use a hand mixer or stand mixer to  
make the cake batter. Sift the flour for a smooth cake batter. Yogurt is preferred over milk and will make the  
cake extra soft, fluffy and moist. To avoid over browning of the top of the cake, cover it halfway while baking  
with a sheet of aluminum foil. DO NOT COVER when the cake goes into the oven or it won't rise properly. If  
you cover the top of the cake with aluminum foil, make sure the cake batter doesn't stick to it."  
            recipeText.text = recipe  
        }  
        else if selectedFood == "Pound Cake" {  
            recipe = "Set oven to 350F and butter and flour a 9"x5" baking tin. Whisk together the flour, baking  
powder, and salt then set aside. Cream the room temperature butter and sugar together with a stand mixer or  
hand mixer until light and fluffy. eggs and sour cream getting added to whipped butter and sugar. Add the eggs  
one at a time. Scrape the mixing bowl down and whisk once more. Add the sour cream and vanilla then beat to  
combine. Flour getting added to pound cake batter, which is then transferred to a loaf pan. Add flour the mixture  
and mix until just combined. Scrape the bowl down and use the spatula to finish mixing the batter. Transfer  
batter to your prepared tin and bake until center is set."  
            recipeText.text = recipe  
        }  
        else if selectedFood == "Sponge Cake" {  
            recipe = "Heat oven to 180C/160C fan/gas 4, butter and line the base of two 20cm spring-form cake tins  
with baking parchment. Using an electric whisk beat the butter and sugar together until pale and fluffy. Crack  
the eggs in one at a time and whisk well, scraping down the sides of the bowl after each addition. Add the lemon  
zest, vanilla, flour, milk and a pinch of salt. Whisk until just combined then divide the mixture between the two  
tins. Bake in the centre of the oven for 25-30 mins until a skewer inserted into the middle of each cake comes  
out clean. After 10 mins remove the cakes from their tins and leave to cool completely on a wire rack. Fill how  
you like. My personal favourite is a good dollop of lemon curd and some fresh cream, then dust the top with  
icing sugar. Will keep for 3 days."  
            recipeText.text = recipe  
        }  
        else if selectedFood == "Biscuit Cake" {  
            recipe = "Line a 2lb loaf tin with a double layer of clingfilm, and for a smooth finish, parchment paper.  
Set aside. Combine the butter with both of the chocolates and condense milk in a medium sized heat proof bowl  
set over a saucepan of barely simmering water and leave to melt. Stir occasionally then remove from heat once  
completely smooth. Leave to cool, about 10 minutes. Break biscuits into medium and small pieces into a large  
bowl. Pour about 3/4 of the chocolate mixture over the broken biscuits, mix until pieces are coated with the
```

mixture. Add the Maltesers and Crunchie pieces and mix to distribute. Pour half of the remaining chocolate into the bottom of the prepared tin and spread it out evenly with a spatula, making sure to get the mixture into the corners. Tip about half of biscuit mixture into the tin and spread out evenly, pressing the surface down firmly with a spatula or large metal spoon to compress the mixture and remove any air pockets. Spoon in the remaining biscuit mixture and repeat. Smooth and flatten out the top. Pour the remaining chocolate mixture over the top and smooth with a spatula. Refrigerate until firm, about 3 hours. Once set, remove from tin and peel away clingfilm and parchment paper. Cut into slices and enjoy. Will keep for up to 3 weeks in an airtight container in the fridge."

```
        recipeText.text = recipe
    }
}
}
```

BunKitchenViewController:

```
import UIKit
```

```
class BunKitchenViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
```

```
    @IBOutlet weak var yoreselTableView: UITableView!
```

```
    var chosenFood = ""
    var chosenImage = UIImage()
```

```
    var foodNames = [String]()
    var foodImages = [UIImage]()
```

```
    override func viewDidLoad() {
        super.viewDidLoad()
```

```
        yoreselTableView.delegate = self
        yoreselTableView.dataSource = self
```

```
        foodNames.append("Plain Bun")
        foodNames.append("Sesame Seeds")
        foodNames.append("English Muffins")
        foodNames.append("Sliced Bread")
```

```
        foodImages.append(UIImage(named: "Bun")!)
        foodImages.append(UIImage(named: "Sesame Seeds")!)
        foodImages.append(UIImage(named: "English Muffins")!)
        foodImages.append(UIImage(named: "Sliced Bread")!)
```

```
    }
```

```
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return foodNames.count
    }
```

```
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let cell = UITableViewCell()
        cell.textLabel?.text = foodNames[indexPath.row]
        return cell
    }
```

```
    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        chosenFood = foodNames[indexPath.row]
        chosenImage = foodImages[indexPath.row]
        performSegue(withIdentifier: "BunRecipes", sender: nil)
```

```

    }

    func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
        if editingStyle == .delete {
            foodNames.remove(at: indexPath.row)
            foodImages.remove(at: indexPath.row)
            tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.fade)
        }
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "BunRecipes" {
            let destinationVC = segue.destination as? BunRecipeViewController
            destinationVC?.selectedFood = chosenFood
            destinationVC?.selectedImage = chosenImage
        }
    }
}

```

BunRecipeViewController:

```

import UIKit

class BunRecipeViewController: UIViewController {

    @IBOutlet weak var recipeName: UILabel!
    @IBOutlet weak var recipeImage: UIImageView!
    @IBOutlet weak var recipeText: UITextView!

    var selectedFood = ""
    var selectedImage = UIImage()
    var recipe = ""

    override func viewDidLoad() {
        super.viewDidLoad()

        recipeImage.inputView?.contentMode = UIView.ContentMode.scaleAspectFill
        recipeImage.layer.cornerRadius = 8

        recipeName.text = selectedFood
        recipeImage.image = selectedImage

        if selectedFood == "Plain Bun" {
            recipe = "Stir together 2 cups flour and the yeast. In a separate bowl, heat milk, water, oil, sugar and salt to lukewarm in microwave. Add all at once to the flour mixture, and beat until smooth, about 3 minutes. Mix in enough flour to make a soft dough, 2 to 3 cups. Mix well. Dust a flat surface with flour, turn dough out onto floured surface, and let rest under bowl for about 10 minutes. Shape dough into 12 slightly flat balls, and place on greased baking sheet to rise until doubled in size. Bake in a preheated 400 degrees F (200 degrees C) oven for 12 to 15 minutes."
            recipeText.text = recipe
        }
        else if selectedFood == "Sesame Seeds" {
            recipe = "Place flour, yeast, sugar & salt in a large mixing bowl. Stir in milk, followed by oil and egg. Beat on high for 2 minutes. On low speed add about 1/4 cup flour until dough forms a mass. Place dough on floured surface and knead 50 turns. Cover and let rest 10 minutes. Meantime, line a baking sheet with parchment paper. Cut dough into 8 pieces. Shape each into a ball and place on baking sheet. Flatten each ball to about 1/2-

```

inch thickness / about 3 1/2-inches across. Cover with a towel and let rise in a warm spot until they double in size, about 30-35 minutes. Meantime, preheat oven to 375° F. Brush each bun with beaten egg and sprinkle with sesame seeds. Bake for 12 minutes."

```
recipeText.text = recipe
```

```
}
```

```
else if selectedFood == "English Muffins"{
```

```
recipe = "Make the dough starter: Mix the flour, water, and yeast for the starter in a small mixing bowl.
```

```
Beat until the batter is smooth and glossy, about 100 strokes. Let the starter sit 1 to 12 hours: Cover the starter and place it out of the way for at least 1 or up to 12 hours. The starter will become increasingly bubbly the longer it sits and will double in bulk. The longer you can let the starter ferment, the better the flavor and structure of your finished English muffins. Whisk together the milk, yeast, and starter: In the bowl of a stand mixer or large mixing bowl, combine the milk and yeast for the dough. Scrape the starter into the bowl and use a whisk to break it up and dissolve it into the milk. It should become quite frothy. Mix the dough together: Add the sugar, butter, and salt to the bowl and whisk to combine. Add 3 cups of the flour and stir with a stiff spatula until you form a shaggy, floury dough. Knead the dough: With a dough hook on a stand mixer, knead the dough until it comes together in a smooth ball, 5 to 8 minutes. Alternatively, knead by hand against the counter. If the dough is very sticky like bubble gum, add extra flour as needed, but err on the side of caution. The dough is ready when it forms into a smooth ball and springs back when poked; it will feel slightly tacky to the touch, but shouldn't stick to the bowl or your hands. Let the dough rise overnight in the fridge: Transfer the dough to a large bowl lightly filmed with oil. Cover and place in the fridge overnight or for up to 3 days.→ Quicker English Muffins: You can also let the dough rise at room temperature until doubled in bulk, 1 1/2 to 2 hours, and then make the muffins immediately. These muffins will have a milder flavor. Divide and shape the muffins: Turn the risen dough out onto a lightly floured work surface. Use a pastry scraper to divide the dough into 12 equal pieces. Roll each piece gently against the counter to shape into smooth, round balls. Transfer the muffins to a baking sheet to rise: Scatter cornmeal generously over a baking sheet and arrange the balls on top, spaced a little apart. If you have muffin rings, place them around the balls at this point. Sprinkle the tops of the balls with more cornmeal. Let the muffins rise until puffy: For dough that was refrigerated, this will take 1 1/2 to 2 hours; for room temperature dough, this will take about 1 hour. Depending on the size of your muffin rings, the muffins may not totally fill the rings — that's okay. Warm a skillet: When ready to cook the muffins, warm a large skillet over medium heat. Melt a small pat of butter — enough to just coat the bottom of the pan and prevent sticking. Cook the muffins 5 to 6 minutes on one side: Working in batches, transfer a few of the muffins to the skillet — allow an inch or so of space between muffins and do not crowd the pan. If using rings, transfer the muffins with their rings to the pan. Cook until the bottoms of the muffins are golden brown, 5 to 6 minutes. Flip and cook 5 to 6 minutes on the other side: Flip the muffins and cook the other side until golden brown, 5 to 6 minutes. If you prefer thinner, less puffy English muffins, you can gently press the tops with the spatula to prevent them from rising too much. Adjust the heat as needed: If your muffins seem to be browning too quickly on the bottoms (or not quickly enough), adjust the heat as needed. (If you find that your muffins are browning too quickly, throw them in the oven at 350°F to finish baking through. Finish cooking all of the muffins: Transfer cooked muffins to a cooling rack. Continue working in batches until all the muffins have been cooked. Add a small pat of butter to the pan between batches to prevent sticking. Split and serve! Split the English muffins with a fork, spread with butter or jam (or both!), and eat. English muffins will keep for several days in an airtight container on the counter and are fantastic warmed in the toaster oven. Fresh English muffins can also be wrapped tightly in plastic wrap or aluminum foil, and kept frozen for up to 3 months."
```

```
recipeText.text = recipe
```

```
}
```

```
else if selectedFood == "Sliced Bread"{
```

```
recipe = "Mix 500g strong Wheat flour, 2 tsp salt and a 7g sachet of fast-action yeast in a large bowl.
```

```
Make a well in the centre, then add 3 tbsp olive oil and 300ml water, and mix well. If the dough seems a little stiff, add another 1-2 tbsp water and mix well. Tip onto a lightly floured work surface and knead for around 10 mins. Once the dough is satin-smooth, place it in a lightly oiled bowl and cover with cling film. Leave to rise for 1 hour until doubled in size or place in the fridge overnight. Line a baking tray with baking parchment. Knock back the dough (punch the air out and pull the dough in on itself) then gently mould the dough into a ball. Place it on the baking parchment to prove for a further hour until doubled in size. Heat oven to 220C/fan 200C/gas.Dust the loaf with some extra flour and cut a cross about 6cm long into the top of the loaf with a sharp knife.Bake for 25-30 mins until golden brown and the loaf sounds hollow when tapped underneath. Cool on a wire rack."
```

```
recipeText.text = recipe
```

```
}
```

```
}
```

```
}
```

ProfileViewController:

```
import UIKit
import CoreData

class ProfileViewController: UIViewController, UITableViewDelegate {

    @IBOutlet weak var tableView: UITableView!
    @IBOutlet weak var sideMenuBtn: UIBarButtonItem!

    // var names: [String] = []
    var people: [NSManagedObject] = []

    override func viewDidLoad() {
        super.viewDidLoad()

        self.sideMenuBtn.target = revealViewController()
        self.sideMenuBtn.action = #selector(self.revealViewController()?.revealSideMenu)

        title = "The List"
        tableView.register(UITableViewCell.self, forCellReuseIdentifier: "Cell")
        // self.navigationItem.leftBarButtonItem = self.editButtonItem
        self.tableView.reloadData()
    }

    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)

        guard let appDelegate = UIApplication.shared.delegate as? AppDelegate else {
            return
        }
        let managedContext = appDelegate.persistentContainer.viewContext
        let fetchRequest = NSFetchRequest<NSManagedObject>(entityName: "Person")

        do {
            people = try managedContext.fetch(fetchRequest)
        } catch let error as NSError {
            print("Could not fetch. \(error), \(error.userInfo)")
        }
    }

    @IBAction func addName(_ sender: UIBarButtonItem) {
        let alert = UIAlertController(title: "New Product Name", message: "Add a new Product name",
        preferredStyle: .alert)

        let saveAction = UIAlertAction(title: "Save", style: .default) { [unowned self] action in

            guard let textField = alert.textFields?.first,
            let nameToSave = textField.text else {
                return
            }

            self.save(name: nameToSave)
            self.tableView.reloadData()
        }
    }
}
```

```

let cancelAction = UIAlertAction(title: "Cancel", style: .cancel)
alert.addTextField()
alert.addAction(saveAction)
alert.addAction(cancelAction)

present(alert, animated: true)
}

func save(name: String) {
guard let appDelegate = UIApplication.shared.delegate as? AppDelegate else {
return
}

let managedContext = appDelegate.persistentContainer.viewContext
let entity = NSEntityDescription.entity(forEntityName: "Person", in: managedContext)!
let person = NSManagedObject(entity: entity, insertInto: managedContext)
person.setValue(name, forKeyPath: "name")

do {
try managedContext.save()
people.append(person)
} catch let error as NSError {
print("Could not save. \(error), \(error.userInfo)")
}
}

// MARK: - UITableViewDataSource
extension ProfileViewController: UITableViewDataSource {

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
return people.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let person = people[indexPath.row]
let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
cell.textLabel?.text = person.value(forKeyPath: "name") as? String
return cell
}

func tableView(_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {
return true
}

func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
if editingStyle == .delete {
print("Deleted")

guard let appDelegate =
UIApplication.shared.delegate as? AppDelegate else {
return
}

let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext
context.delete(people[indexPath.row] as NSManagedObject)
//context.save(nil)
}
}
}

```

```

    people.remove(at: indexPath.row)
    self.tableView.beginUpdates()
    //self.tableView.deleteRows(at: [indexPath], with: .fade)
    self.tableView.deleteRows(at: [indexPath], with: UITableView.RowAnimation.automatic)
    self.tableView.endUpdates()

    appDelegate.saveContext()

    //people.remove(at: indexPath.row)
    //tableView.reloadData()

}
}

```

```

func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.

    let appDelegate = UIApplication.shared.delegate as! AppDelegate
    let context: NSManagedObjectContext = appDelegate.persistentContainer.viewContext

    return true
}
}

```

MainViewController:

```

import UIKit
import SafariServices

class MainViewController: UIViewController {
    private var sideMenuViewController: SideMenuViewController!
    private var sideMenuShadowView: UIView!
    private var sideMenuRevealWidth: CGFloat = 260
    private let paddingForRotation: CGFloat = 150
    private var isExpanded: Bool = false
    private var draggingIsEnabled: Bool = false
    private var panBaseLocation: CGFloat = 0.0

    // Expand/Collapse the side menu by changing trailing's constant
    private var sideMenuTrailingConstraint: NSLayoutConstraint!

    private var revealSideMenuOnTop: Bool = true

    var gestureEnabled: Bool = true

    override func viewDidLoad() {
        super.viewDidLoad()
        self.view.backgroundColor = colorLiteral(red: 1, green: 0.1607843137, blue: 0.2784313725, alpha: 1)

        // Shadow Background View
        self.sideMenuShadowView = UIView(frame: self.view.bounds)
        self.sideMenuShadowView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
        self.sideMenuShadowView.backgroundColor = .black
        self.sideMenuShadowView.alpha = 0.0
    }
}

```



```

    let tapGestureRecognizer = UITapGestureRecognizer(target: self, action:
#selector(TapGestureRecognizer))
    tapGestureRecognizer.numberOfTapsRequired = 1
    tapGestureRecognizer.delegate = self
    view.addGestureRecognizer(tapGestureRecognizer)
    if self.revealSideMenuOnTop {
        view.insertSubview(self.sideMenuShadowView, at: 1)
    }

    // Side Menu
    let storyboard = UIStoryboard(name: "Main", bundle: Bundle.main)
    self.sideMenuViewController = storyboard.instantiateViewController(withIdentifier: "SideMenuID") as?
SideMenuViewController
    self.sideMenuViewController.defaultHighlightedCell = 0 // Default Highlighted Cell
    self.sideMenuViewController.delegate = self
    view.insertSubview(self.sideMenuViewController!.view, at: self.revealSideMenuOnTop ? 2 : 0)
    addChild(self.sideMenuViewController!)
    self.sideMenuViewController!.didMove(toParent: self)

    // Side Menu AutoLayout

    self.sideMenuViewController.view.translatesAutoresizingMaskIntoConstraints = false

    if self.revealSideMenuOnTop {
        self.sideMenuTrailingConstraint =
self.sideMenuViewController.view.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: -
self.sideMenuRevealWidth - self.paddingForRotation)
        self.sideMenuTrailingConstraint.isActive = true
    }
    NSLayoutConstraint.activate([
        self.sideMenuViewController.view.widthAnchor.constraint(equalToConstant:
self.sideMenuRevealWidth),
        self.sideMenuViewController.view.bottomAnchor.constraint(equalTo: view.bottomAnchor),
        self.sideMenuViewController.view.topAnchor.constraint(equalTo: view.topAnchor)
    ])

    // Side Menu Gestures
    let panGestureRecognizer = UIPanGestureRecognizer(target: self, action: #selector(handlePanGesture))
    panGestureRecognizer.delegate = self
    view.addGestureRecognizer(panGestureRecognizer)

    // Default Main View Controller
    showViewController(viewController: UINavigationController.self, storyboardId: "HomeNavID")
}

// Keep the state of the side menu (expanded or collapse) in rotation
override func viewWillTransition(to size: CGSize, with coordinator:
UIViewControllerTransitionCoordinator) {
    super.viewWillTransition(to: size, with: coordinator)
    coordinator.animate { _ in
        if self.revealSideMenuOnTop {
            self.sideMenuTrailingConstraint.constant = self.isExpanded ? 0 : (-self.sideMenuRevealWidth -
self.paddingForRotation)
        }
    }
}

func animateShadow(targetPosition: CGFloat) {
    UIView.animate(withDuration: 0.5) {
        // When targetPosition is 0, which means side menu is expanded, the shadow opacity is 0.6

```

```

        self.sideMenuShadowView.alpha = (targetPosition == 0) ? 0.6 : 0.0
    }
}

// Call this Button Action from the View Controller you want to Expand/Collapse when you tap a button
@IBAction open func revealSideMenu() {
    self.sideMenuState(expanded: self.isExpanded ? false : true)
}

func sideMenuState(expanded: Bool) {
    if expanded {
        self.animateSideMenu(targetPosition: self.revealSideMenuOnTop ? 0 : self.sideMenuRevealWidth) { _
in
            self.isExpanded = true
        }
        // Animate Shadow (Fade In)
        UIView.animate(withDuration: 0.5) { self.sideMenuShadowView.alpha = 0.6 }
    }
    else {
        self.animateSideMenu(targetPosition: self.revealSideMenuOnTop ? (-self.sideMenuRevealWidth -
self.paddingForRotation) : 0) { _ in
            self.isExpanded = false
        }
        // Animate Shadow (Fade Out)
        UIView.animate(withDuration: 0.5) { self.sideMenuShadowView.alpha = 0.0 }
    }
}

func animateSideMenu(targetPosition: CGFloat, completion: @escaping (Bool) -> ()) {
    UIView.animate(withDuration: 0.5, delay: 0, usingSpringWithDamping: 1.0, initialSpringVelocity: 0,
options: .layoutSubviews, animations: {
        if self.revealSideMenuOnTop {
            self.sideMenuTrailingConstraint.constant = targetPosition
            self.view.layoutIfNeeded()
        }
        else {
            self.view.subviews[1].frame.origin.x = targetPosition
        }
    }, completion: completion)
}

override var preferredStatusBarStyle: UIStatusBarStyle {
    return .lightContent
}

extension MainViewController: SideMenuViewControllerDelegate {
    func selectedCell(_ row: Int) {
        switch row {
        case 0:
            // Home
            self.showViewController(viewController: UINavigationController.self, storyboardId: "HomeNavID")
        case 1:
            // Music
            self.showViewController(viewController: UINavigationController.self, storyboardId: "MusicNavID")
        case 2:
            // Maps
            self.showViewController(viewController: UINavigationController.self, storyboardId: "MapsNavID")
        case 3:
            // Receipe

```

```

        self.showViewController(viewController: UINavigationController.self, storyboardId: "ReceipeVCID")
    case 4:
        // Profile
        self.showViewController(viewController: UINavigationController.self, storyboardId:
"ShoppinglistNavID")
//        let storyboard = UIStoryboard(name: "Main", bundle: Bundle.main)
//        let profileModalVC = storyboard.instantiateViewController(withIdentifier: "ProfileModalID") as?
ProfileViewController
        //present(profileModalVC!, animated: true, completion: nil)
    case 5:
        // Settings
        self.showViewController(viewController: UINavigationController.self, storyboardId: "SettingsNavID")
    case 6:
        // Like us on facebook
        let safariVC = SFSafariViewController(url: URL(string:
"https://instagram.com/saro_sarvana?r=nametag")!)
        present(safariVC, animated: true)
    default:
        break
    }

    // Collapse side menu with animation
    DispatchQueue.main.async { self.sideMenuState(expanded: false) }
}

func showViewController<T: UIViewController>(viewController: T.Type, storyboardId: String) -> () {
    // Remove the previous View
    for subview in view.subviews {
        if subview.tag == 99 {
            subview.removeFromSuperview()
        }
    }
    let storyboard = UIStoryboard(name: "Main", bundle: nil)
    let vc = storyboard.instantiateViewController(withIdentifier: storyboardId) as! T
    vc.view.tag = 99
    view.insertSubview(vc.view, at: self.revealSideMenuOnTop ? 0 : 1)
    addChild(vc)
    if !self.revealSideMenuOnTop {
        if isExpanded {
            vc.view.frame.origin.x = self.sideMenuRevealWidth
        }
        if self.sideMenuShadowView != nil {
            vc.view.addSubview(self.sideMenuShadowView)
        }
    }
    vc.didMove(toParent: self)
}

extension MainViewController: UIGestureRecognizerDelegate {
    @objc func TapGestureRecognizer(sender: UITapGestureRecognizer) {
        if sender.state == .ended {
            if self.isExpanded {
                self.sideMenuState(expanded: false)
            }
        }
    }
}

// Close side menu when you tap on the shadow background view

```

```

func gestureRecognizer(_ gestureRecognizer: UIGestureRecognizer, shouldReceive touch: UITouch) -> Bool
{
    if (touch.view?.isDescendant(of: self.sideMenuViewController.view))! {
        return false
    }
    return true
}

// Dragging Side Menu
@objc private func handlePanGesture(sender: UIPanGestureRecognizer) {

    guard gestureEnabled == true else { return }

    let position: CGFloat = sender.translation(in: self.view).x
    let velocity: CGFloat = sender.velocity(in: self.view).x

    switch sender.state {
    case .began:

        // If the user tries to expand the menu more than the reveal width, then cancel the pan gesture
        if velocity > 0, self.isExpanded {
            sender.state = .cancelled
        }

        // If the user swipes right but the side menu hasn't expanded yet, enable dragging
        if velocity > 0, !self.isExpanded {
            self.draggingIsEnabled = true
        }
        // If user swipes left and the side menu is already expanded, enable dragging
        else if velocity < 0, self.isExpanded {
            self.draggingIsEnabled = true
        }
        }

    if self.draggingIsEnabled {
        // If swipe is fast, Expand/Collapse the side menu with animation instead of dragging
        let velocityThreshold: CGFloat = 550
        if abs(velocity) > velocityThreshold {
            self.sideMenuState(expanded: self.isExpanded ? false : true)
            self.draggingIsEnabled = false
            return
        }

        if self.revealSideMenuOnTop {
            self.panBaseLocation = 0.0
            if self.isExpanded {
                self.panBaseLocation = self.sideMenuRevealWidth
            }
        }
    }

    case .changed:

        // Expand/Collapse side menu while dragging
        if self.draggingIsEnabled {
            if self.revealSideMenuOnTop {
                // Show/Hide shadow background view while dragging
                let xLocation: CGFloat = self.panBaseLocation + position
                let percentage = (xLocation * 150 / self.sideMenuRevealWidth) / self.sideMenuRevealWidth

                let alpha = percentage >= 0.6 ? 0.6 : percentage
            }
        }
    }
}

```



```

songs.append(Song(name: "Beggin",
    albumName: "Film: Maneskin",
    artistName: "Music by: Maneskin",
    imageName: "Beggin",
    trackName: "Song3"))

songs.append(Song(name: "Thalaivar",
    albumName: "Film: Darbar",
    artistName: "Music by: Aniruth",
    imageName: "Darbar",
    trackName: "Song4"))

songs.append(Song(name: "Neeye Oli",
    albumName: "Film: Sarpatta Parambarai",
    artistName: "Music by: Santhosh",
    imageName: "Sarpatta Parambarai",
    trackName: "Song1"))

songs.append(Song(name: "Chammak",
    albumName: "Film: Ra One",
    artistName: "Music by: Arjith Singh",
    imageName: "RaOne",
    trackName: "Song2"))

songs.append(Song(name: "Beggin",
    albumName: "Film: Maneskin",
    artistName: "Music by: Maneskin",
    imageName: "Beggin",
    trackName: "Song3"))

songs.append(Song(name: "Thalaivar",
    albumName: "Film: Darbar",
    artistName: "Music by: Aniruth",
    imageName: "Darbar",
    trackName: "Song4"))

songs.append(Song(name: "Neeye Oli",
    albumName: "Film: Sarpatta Parambarai",
    artistName: "Music by: Santhosh",
    imageName: "Sarpatta Parambarai",
    trackName: "Song1"))
}

//Table
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return songs.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
    let song = songs[indexPath.row]
    // configure
    cell.textLabel?.text = song.name
    cell.detailTextLabel?.text = song.albumName
    cell.accessoryType = .disclosureIndicator
    cell.imageView?.image = UIImage(named: song.imageName)
    cell.textLabel?.font = UIFont(name: "Helvetica-Bold", size: 18)
    cell.detailTextLabel?.font = UIFont(name: "Helvetica", size: 17)

    return cell
}

```

```

}

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 100
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    tableView.deselectRow(at: indexPath, animated: true)

    // present the player
    let position = indexPath.row
    guard let vc = storyboard?.instantiateViewController(identifier: "player") as? PlayerViewController else {
        return
    }
    vc.songs = songs
    vc.position = position
    present(vc, animated: true)
}

}

struct Song {
    let name: String
    let albumName: String
    let artistName: String
    let imageName: String
    let trackName: String
}

```

PlayerViewController:

```

import Foundation
import AVFoundation
import UIKit

class PlayerViewController: UIViewController {

    public var position: Int = 0
    public var songs: [Song] = []

    @IBOutlet weak var holder: UIView!

    var player: AVAudioPlayer?

    // User Interface elements

    private let albumImageView: UIImageView = {
        let imageView = UIImageView()
        imageView.contentMode = .scaleAspectFill
        return imageView
    }()

    private let songNameLabel: UILabel = {
        let label = UILabel()
        label.textAlignment = .center
        label.numberOfLines = 0 // line wrap
        return label
    }()

```



```

}()

private let artistNameLabel: UILabel = {
    let label = UILabel()
    label.textAlignment = .center
    label.numberOfLines = 0 // line wrap
    return label
}()

private let albumNameLabel: UILabel = {
    let label = UILabel()
    label.textAlignment = .center
    label.numberOfLines = 0 // line wrap
    return label
}()

let playPauseButton = UIButton()

override func viewDidLoad() {
    super.viewDidLoad()
}

override func viewDidLoadSubviews() {
    super.viewDidLoadSubviews()
    if holder.subviews.count == 0 {
        configure()
    }
}

func configure() {
    // set up player
    let song = songs[position]

    let urlString = Bundle.main.path(forResource: song.trackName, ofType: "mp3")

    do {
        try AVAudioSession.sharedInstance().setMode(.default)
        try AVAudioSession.sharedInstance().setActive(true, options: .notifyOthersOnDeactivation)

        guard let urlString = urlString else {
            print("urlstring is nil")
            return
        }

        player = try AVAudioPlayer(contentsOf: URL(string: urlString)!)

        guard let player = player else {
            print("player is nil")
            return
        }
        player.volume = 0.5

        player.play()
    }
    catch {
        print("error occurred")
    }

    // set up user interface elements

```

```

// album cover
albumImageView.frame = CGRect(x: 10,
                               y: 10,
                               width: holder.frame.size.width-20,
                               height: holder.frame.size.width-20)
albumImageView.image = UIImage(named: song.imageName)
holder.addSubview(albumImageView)

// Labels: Song name, album, artist
songNameLabel.frame = CGRect(x: 10,
                              y: albumImageView.frame.size.height + 10,
                              width: holder.frame.size.width-20,
                              height: 70)
albumNameLabel.frame = CGRect(x: 10,
                              y: albumImageView.frame.size.height + 10 + 70,
                              width: holder.frame.size.width-20,
                              height: 70)
artistNameLabel.frame = CGRect(x: 10,
                               y: albumImageView.frame.size.height + 10 + 140,
                               width: holder.frame.size.width-20,
                               height: 70)

songNameLabel.text = song.name
albumNameLabel.text = song.albumName
artistNameLabel.text = song.artistName

holder.addSubview(songNameLabel)
holder.addSubview(albumNameLabel)
holder.addSubview(artistNameLabel)

// Player controls
let nextButton = UIButton()
let backButton = UIButton()

// Frame
let yPosition = artistNameLabel.frame.origin.y + 70 + 20
let size: CGFloat = 70

playPauseButton.frame = CGRect(x: (holder.frame.size.width - size) / 2.0,
                               y: yPosition,
                               width: size,
                               height: size)

nextButton.frame = CGRect(x: holder.frame.size.width - size - 20,
                          y: yPosition,
                          width: size,
                          height: size)

backButton.frame = CGRect(x: 20,
                          y: yPosition,
                          width: size,
                          height: size)

// Add actions
playPauseButton.addTarget(self, action: #selector(didTapPlayPauseButton), for: .touchUpInside)
nextButton.addTarget(self, action: #selector(didTapNextButton), for: .touchUpInside)
backButton.addTarget(self, action: #selector(didTapBackButton), for: .touchUpInside)

// Styling

```

```

playPauseButton.setBackgroundImage(UIColor(systemName: "pause.fill"), for: .normal)
backButton.setBackgroundImage(UIColor(systemName: "backward.fill"), for: .normal)
nextButton.setBackgroundImage(UIColor(systemName: "forward.fill"), for: .normal)

playPauseButton.tintColor = .black
backButton.tintColor = .black
nextButton.tintColor = .black

holder.addSubview(playPauseButton)
holder.addSubview(nextButton)
holder.addSubview(backButton)

// slider
let slider = UISlider(frame: CGRect(x: 20,
    y: holder.frame.size.height-60,
    width: holder.frame.size.width-40,
    height: 50))

slider.value = 0.5
slider.addTarget(self, action: #selector(didSlideSlider(_)), for: .valueChanged)
holder.addSubview(slider)
}

@objc func didTapBackButton() {
    if position > 0 {
        position = position - 1
        player?.stop()
        for subview in holder.subviews {
            subview.removeFromSuperview()
        }
        configure()
    }
}

@objc func didTapNextButton() {
    if position < (songs.count - 1) {
        position = position + 1
        player?.stop()
        for subview in holder.subviews {
            subview.removeFromSuperview()
        }
        configure()
    }
}

@objc func didTapPlayPauseButton() {
    if player?.isPlaying == true {
        // pause
        player?.pause()
        // show play button
        playPauseButton.setBackgroundImage(UIColor(systemName: "play.fill"), for: .normal)

        // shrink image
        UIView.animate(withDuration: 0.2, animations: {
            self.albumImageView.frame = CGRect(x: 30,
                y: 30,
                width: self.holder.frame.size.width-60,
                height: self.holder.frame.size.width-60)
        })
    }
    else {

```

```

// play
player?.play()
playPauseButton.setBackgroundImage(UIImage(systemName: "pause.fill"), for: .normal)

// increase image size
UIView.animate(withDuration: 0.2, animations: {
    self.albumImageView.frame = CGRect(x: 10,
        y: 10,
        width: self.holder.frame.size.width-20,
        height: self.holder.frame.size.width-20)
})
}
}

@objc func didSlideSlider(_ slider: UISlider) {
    let value = slider.value
    player?.volume = value
}

override func viewWillDisappear(_ animated: Bool) {
    super.viewWillDisappear(animated)
    if let player = player {
        player.stop()
    }
}
}
}

```

GoogleMapViewController:

```

import Foundation
import GoogleMaps
import GooglePlaces
import SideMenu
import UIKit
import CoreLocation

class GoogleMapViewController: UIViewController, CLLocationManagerDelegate {

    @IBOutlet weak var sideMenuBtn: UIBarButtonItem!
    @IBOutlet private weak var pinImageVerticalConstraint: NSLayoutConstraint!

    var locationManager = CLLocationManager()
    lazy var mapView = GMSMapView()

    let didFindMyLocation = false

    override func viewDidLoad() {
        self.sideMenuBtn.target = revealViewController()
        self.sideMenuBtn.action = #selector(self.revealViewController()?.revealSideMenu)

        let camera = GMSCameraPosition.camera(withLatitude: 50.1300, longitude: 14.3736, zoom: 6.0)
        //let mapView = GMSMapView.map(withFrame: CGRect.zero, camera: camera)
        mapView = GMSMapView.map(withFrame: CGRect.zero, camera: camera)
        mapView.isMyLocationEnabled = true
        self.view = mapView
    }
}

```

```

//Google Maps SDK: Border
let mapInsets = UIEdgeInsets(top: 10.0, left: 0.0, bottom: 10.0, right: 0.0)
mapView.padding = mapInsets

locationManager.distanceFilter = 100
locationManager.delegate = self
locationManager.requestWhenInUseAuthorization()
locationManager.desiredAccuracy = kCLLocationAccuracyBest
locationManager.startUpdatingLocation()

//Google Maps SDK: Compass
mapView.settings.compassButton = true

//Google Maps SDK: User's Location
mapView.isMyLocationEnabled = true
mapView.settings.myLocationButton = true

let marker = GMSMarker()
marker.position = CLLocationCoordinate2D(latitude: 50.1300, longitude: 14.3736)
marker.title = "Ceska Zemedelska Univerzita v Praze"
marker.snippet = "Prague, Czech Republic"
marker.map = mapView

}

```

SideMenuCell:

```

import UIKit

class SideMenuCell: UITableViewCell {

    class var identifier: String { return String(describing: self) }
    class var nib: UINib { return UINib(nibName: identifier, bundle: nil) }

    @IBOutlet var iconImageView: UIImageView!
    @IBOutlet var titleLabel: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()

        // Background
        self.backgroundColor = .clear

        // Icon
        self.iconImageView.tintColor = .white

        // Title
        self.titleLabel.textColor = .white
    }
}

```

SideMenuModel:

```

import UIKit

struct SideMenuModel {
    var icon: UIImage
}

```

```

    var title: String
}

SideMenuViewController:
import UIKit

protocol SideMenuViewControllerDelegate {
    func selectedCell(_ row: Int)
}

class SideMenuViewController: UIViewController {
    @IBOutlet var headerImageView: UIImageView!
    @IBOutlet var sideMenuTableView: UITableView!
    @IBOutlet var footerLabel: UILabel!

    var delegate: SideMenuViewControllerDelegate?

    var defaultHighlightedCell: Int = 0

    var menu: [SideMenuModel] = [
        SideMenuModel(icon: UIImage(systemName: "house.fill")!, title: "Home"),
        SideMenuModel(icon: UIImage(systemName: "music.note")!, title: "Music"),
        SideMenuModel(icon: UIImage(systemName: "map.fill")!, title: "Maps"),
        SideMenuModel(icon: UIImage(systemName: "book.fill")!, title: "Recipes"),
        SideMenuModel(icon: UIImage(systemName: "person.fill")!, title: "Shopping List"),
        SideMenuModel(icon: UIImage(systemName: "slider.horizontal.3")!, title: "General Knowledge"),
        SideMenuModel(icon: UIImage(systemName: "hand.thumbsup.fill")!, title: "Follow us on instagram")
    ]

    override func viewDidLoad() {
        super.viewDidLoad()

        // TableView
        self.sideMenuTableView.delegate = self
        self.sideMenuTableView.dataSource = self
        self.sideMenuTableView.backgroundColor = colorLiteral(red: 0.8078, green: 0.0941, blue: 0, alpha: 1)
        self.sideMenuTableView.separatorStyle = .none

        // Set Highlighted Cell
        DispatchQueue.main.async {
            let defaultRow = IndexPath(row: self.defaultHighlightedCell, section: 0)
            self.sideMenuTableView.selectRow(at: defaultRow, animated: false, scrollPosition: .none)
        }

        // Footer
        self.footerLabel.textColor = UIColor.white
        self.footerLabel.font = UIFont.systemFont(ofSize: 12, weight: .bold)
        self.footerLabel.text = "by Saravana Kumar"

        // Register TableView Cell
        self.sideMenuTableView.register(SideMenuCell.nib, forCellReuseIdentifier: SideMenuCell.identifier)

        // Update TableView with the data
        self.sideMenuTableView.reloadData()
    }
}

// MARK: - UITableViewDelegate

extension SideMenuViewController: UITableViewDelegate {

```

```

func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
    return 44
}
}

// MARK: - UITableViewDataSource

extension SideMenuViewController: UITableViewDataSource {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return self.menu.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        guard let cell = tableView.dequeueReusableCell(withIdentifier: SideMenuCell.identifier, for: indexPath)
as? SideMenuCell else { fatalError("xib doesn't exist") }

        cell.imageView.image = self.menu[indexPath.row].icon
        cell.titleLabel.text = self.menu[indexPath.row].title

        // Highlighted color
        let myCustomSelectionColorView = UIView()
        myCustomSelectionColorView.backgroundColor = colorLiteral(red: 0.8078, green: 0.0941, blue: 0, alpha:
1)
        cell.selectedBackgroundView = myCustomSelectionColorView
        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        self.delegate?.selectedCell(indexPath.row)
        // Remove highlighted color when you press the 'Profile' and 'Like us on facebook' cell
        if indexPath.row == 4 || indexPath.row == 6 {
            tableView.deselectRow(at: indexPath, animated: true)
        }
    }
}
}

```

AppDelegate:

```

import UIKit
import CoreData
import SwiftUI
import GoogleMaps
import GooglePlaces
import AVFAudio

let googleApiKey = "AIzaSyANuAR-3bcvxxl1qk7DfPrFd5z3BchDOC0"

@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {

        GMSServices.provideAPIKey("AIzaSyANuAR-3bcvxxl1qk7DfPrFd5z3BchDOC0")
        GMSPPlacesClient.provideAPIKey("AIzaSyANuAR-3bcvxxl1qk7DfPrFd5z3BchDOC0")
        //GMSServices.provideAPIKey(googleApiKey)
    }
}

```

```

do {
    try AVAudioSession.sharedInstance().setCategory(AVAudioSession.Category.playback, mode:
AVAudioSession.Mode.default, options: [.mixWithOthers, .allowAirPlay])
    print("Playback OK")
    try AVAudioSession.sharedInstance().setActive(true)
    print("Session is Active")
} catch {
    print(error)
}

return true
}

```

```

func applicationWillTerminate(_ application: UIApplication) {
    self.saveContext()
}

```

// MARK: - Core Data stack

```

lazy var persistentContainer: NSPersistentContainer = {
    // The persistent container for the application. This implementation
    // creates and returns a container, having loaded the store for the
    // application to it. This property is optional since there are legitimate
    // error conditions that could cause the creation of the store to fail.
    let container = NSPersistentContainer(name: "DataModel")
    container.loadPersistentStores(completionHandler: { (storeDescription, error) in
        if let error = error as NSError? {
            // Replace this implementation with code to handle the error appropriately.
            // fatalError() causes the application to generate a crash log and terminate.
            // You should not use this function in a shipping application, although it may be useful during
            development.

            /*
            Typical reasons for an error here include:
            * The parent directory does not exist, cannot be created, or disallows writing.
            * The persistent store is not accessible, due to permissions or data protection when the device is locked.
            * The device is out of space.
            * The store could not be migrated to the current model version.
            Check the error message to determine what the actual problem was.
            */
            fatalError("Unresolved error \(error), \(error.userInfo)")
        }
    })
    return container
}()

```

// MARK: - Core Data Saving support

```

func saveContext () {
    let context = persistentContainer.viewContext
    if context.hasChanges {
        do {
            try context.save()
        } catch {
            // Replace this implementation with code to handle the error appropriately.
            // fatalError() causes the application to generate a crash log and terminate.
            // You should not use this function in a shipping application, although it may be useful during
            development.
            let nerror = error as NSError
            fatalError("Unresolved error \(nerror), \(nerror.userInfo)")
        }
    }
}

```



```
}  
}  
}  
}
```

// MARK: UISceneSession Lifecycle

```
func application(_ application: UIApplication, configurationForConnecting connectingSceneSession:  
UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {  
    // Called when a new scene session is being created.  
    // Use this method to select a configuration to create the new scene with.  
    return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)  
}  
  
func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions:  
Set<UISceneSession>) {  
  
}
```

PodFile:

```
'SideMenu'  
'Google-Maps-iOS-Utills' '~> 4.0.0'  
'GoogleMaps' '6.0.1.0'  
'GooglePlaces'  
  
'MultiTaskerTests' do  
  
# Pods for testing  
end  
  
'MultiTaskerUITests' do  
# Pods for testing  
end  
  
end
```