viewController.6

import UIKit

class MealViewController: UIViewController, UITextFieldDelegate, UIImagePickerControllerDelegate, UINavigationControllerDelegate {

//MARK: Properties

@IBOutlet weak var nameTextField: UITextField!

@IBOutlet weak var photoImageView: UIImageView!

@IBOutlet weak var ratingControl: RatingControl!

override func viewDidLoad() {

super.viewDidLoad()

// Handle the text field’s user input through delegate callbacks.

nameTextField.delegate = self

}

//MARK: UITextFieldDelegate

func textFieldShouldReturn(\_ textField: UITextField) -> Bool {

// Hide the keyboard.

textField.resignFirstResponder()

return true

}

func textFieldDidEndEditing(\_ textField: UITextField) {

}

//MARK: UIImagePickerControllerDelegate

func imagePickerControllerDidCancel(\_ picker: UIImagePickerController) {

// Dismiss the picker if the user canceled.

dismiss(animated: true, completion: nil)

}

func imagePickerController(\_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : Any]) {

// The info dictionary may contain multiple representations of the image. You want to use the original.

guard let selectedImage = info[UIImagePickerControllerOriginalImage] as? UIImage else {

fatalError("Expected a dictionary containing an image, but was provided the following: \(info)")

}

// Set photoImageView to display the selected image.

photoImageView.image = selectedImage

// Dismiss the picker.

dismiss(animated: true, completion: nil)

}

//MARK: Actions

@IBAction func selectImageFromPhotoLibrary(\_ sender: UITapGestureRecognizer) {

// Hide the keyboard.

nameTextField.resignFirstResponder()

// UIImagePickerController is a view controller that lets a user pick media from their photo library.

let imagePickerController = UIImagePickerController()

// Only allow photos to be picked, not taken.

imagePickerController.sourceType = .photoLibrary

// Make sure ViewController is notified when the user picks an image.

imagePickerController.delegate = self

present(imagePickerController, animated: true, completion: nil)

}

}