**//CoreData**

//array of type CoreData Entity

var sportsList = [Sports]()

//needed objects

let context = (UIApplication.shared.delegate as! AppDelegate).persistentContainer.viewContext

let save = (UIApplication.shared.delegate as! AppDelegate).saveContext

//fetching data

func fetchingSportsData() {

let result: NSFetchRequest<Sports> = Sports.fetchRequest()

do {

sportsList = try context.fetch(result)

}catch {

print(error)

}

tableView.reloadData()

}

//add new sport

@IBAction func addNewSport(\_ sender: UIBarButtonItem) {

let alert = UIAlertController(title: "Add New Sport", message: nil, preferredStyle: .alert)

alert.addTextField { textField in

textField.placeholder = "Enter Sport Name"

}

alert.addAction(UIAlertAction(title: "Save", style: .default, handler: { \_ in

let name = alert.textFields![0].text!

if name.isEmpty {

return

}

let newSport = Sports(context: self.context)

newSport.name = name

self.save()

self.fetchingSportsData()

}))

alert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: .none))

present(alert, animated: true)

}

//delete

override func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {

context.delete(sportsList[indexPath.row])

save()

fetchingSportsData()

}

**//image picker**

// variable

var imageView = UIImageView()

//Picker calling variable

let picker = UIImagePickerController()

picker.delegate = self

picker.sourceType = .photoLibrary

present(picker, animated: true)

//extension

extension ……: UIImagePickerControllerDelegate, UINavigationControllerDelegate {

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

guard let userPickedImage = info[.originalImage] as? UIImage else { return }

imageView.image = userPickedImage

picker.dismiss(animated: true, completion: nil)

}

}