

```

//
// AppDelegate.swift
// FBLA-QuizME
//
// Created by Udit Garg on 11/6/18.
// Copyright © 2018 Udit Garg. All rights reserved.
//

import UIKit
import CoreData

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


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

    func applicationWillResignActive(_ application: UIApplication) {
        // Sent when the application is about to move from active to inactive state.
        This can occur for certain types of temporary interruptions (such as an incoming phone
        call or SMS message) or when the user quits the application and it begins the
        transition to the background state.
        // Use this method to pause ongoing tasks, disable timers, and invalidate
        graphics rendering callbacks. Games should use this method to pause the game.
    }

    func applicationDidEnterBackground(_ application: UIApplication) {
        // Use this method to release shared resources, save user data, invalidate
        timers, and store enough application state information to restore your application to
        its current state in case it is terminated later.
        // If your application supports background execution, this method is called
        instead of applicationWillTerminate: when the user quits.
    }

    func applicationWillEnterForeground(_ application: UIApplication) {
        // Called as part of the transition from the background to the active state;
        here you can undo many of the changes made on entering the background.
    }

    func applicationDidBecomeActive(_ application: UIApplication) {
        // Restart any tasks that were paused (or not yet started) while the
        application was inactive. If the application was previously in the background,
        optionally refresh the user interface.
    }

    func applicationWillTerminate(_ application: UIApplication) {
        // Called when the application is about to terminate. Save data if
        appropriate. See also applicationDidEnterBackground:.
        // Saves changes in the application's managed object context before the
        application terminates.
        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: "FBLA_QuizME")
        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 nserror = error as NSError
                fatalError("Unresolved error \(nserror), \(nserror.userInfo)")
            }
        }
    }
}

```