1. **No Connection Page Source Code**

* import UIKit
* class noConViewController: UIViewController {
* override func viewDidLoad() {
* super.viewDidLoad()
* }
* override func didReceiveMemoryWarning() {
* super.didReceiveMemoryWarning()
* }
* @IBAction func tryButton(\_ sender: Any) {
* print("ok")
* var status2 = String()
* let status = Reach().connectionStatus()
* switch status {
* case .unknown, .offline:
* status2 = ("Not connected")
* print("Not connected")
* case .online(.wwan):
* status2 = ("Connected via WWAN")
* print("Connected via WWAN")
* case .online(.wiFi):
* status2 = ("Connected via WiFi")
* print("Connected via WiFi")
* }
* if(status2 != "Not connected")
* {
* print("Connected OKKKK")
* OperationQueue.main.addOperation {
* self.performSegue(withIdentifier: "conOkSegue", sender: self)
* }
* }
* }
* }

1. View Controller Page Source Code

* import UIKit
* import CoreData
* class ViewController: UIViewController {
* @IBOutlet weak var emailLabel: UITextField!
* @IBOutlet weak var passLabel: UITextField!
* override func viewDidLoad() {
* super.viewDidLoad()
* var status2 = String()
* let status = Reach().connectionStatus()
* switch status {
* case .unknown, .offline:
* status2 = ("Not connected")
* print("Not connected")
* case .online(.wwan):
* status2 = ("Connected via WWAN")
* print("Connected via WWAN")
* case .online(.wiFi):
* status2 = ("Connected via WiFi")
* print("Connected via WiFi")
* }
* if(status2 == "Not connected")
* {
* print("No Connection")
* OperationQueue.main.addOperation {
* self.performSegue(withIdentifier: "noConSegue", sender: self)
* }
* }
* let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action: "dismissKeyboard")
* view.addGestureRecognizer(tap)
* }
* func dismissKeyboard() {
* view.endEditing(true)
* }
* override func didReceiveMemoryWarning() {
* super.didReceiveMemoryWarning()
* }
* func networkStatusChanged(\_ notification: Notification) {
* let userInfo = (notification as NSNotification).userInfo
* print(userInfo)
* }
* @IBAction func loginBtn(\_ sender: UIButton) {
* var userty:String!
* let urlString = "<http://localhost/api/login.php>"
* let urlRequest = URL(string: urlString)
* var request = URLRequest(url: urlRequest! as URL)
* request.httpMethod = "POST"
* let email = emailLabel.text
* let pass1 = passLabel.text
* let parameters = "email="+email!+"&password="+pass1!
* request.httpBody = parameters.data(using: String.Encoding.utf8)
* if((email?.isEmpty)! || (pass1?.isEmpty)!)
* {
* print("ERROR MISSING PARTS")
* }
* let task = URLSession.shared.dataTask(with: request as URLRequest) { (data, response, error) in
* if error != nil // Hata boş değilse, HATA VARSA
* {
* print(error!)
* }
* else
* {
* do
* {
* let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers) as? NSDictionary
* if let JsonParse = json{
* var usertype:String!
* var status:String!
* let userid = self.emailLabel.text
* usertype = JsonParse["type"] as! String?
* status = JsonParse["message"] as! String?
* print(usertype)
* print(status)
* if(status == "LoginOK")
* {
* userty=usertype
* switch usertype {
* case "student":
* print("student")
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "Login OK", message: "Login process completed succesfully", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier:"studentSegue", sender: userid)
* }))
* self.present(alert, animated: true, completion: nil)
* }
* case "admin":
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "Login OK", message: "Login process completed succesfully", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier:"adminSegue", sender: userid)
* }))
* self.present(alert, animated: true, completion: nil)
* }
* case "instructor":
* print("instructor")
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "Login OK", message: "Login process completed succesfully", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier:"instructorSegue", sender: userid)
* }))
* self.present(alert, animated: true, completion: nil)
* }
* default:
* print("scase error")
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "Login Error", message: "Please check id and password", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* }))
* self.present(alert, animated: true, completion: nil)
* }
* }
* }
* else
* {
* OperationQueue.main.addOperation {
* print("elsegeliyor")
* let alert = UIAlertController(title: "Login Error", message: "Please check id and password", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* }))
* self.present(alert, animated: true, completion: nil)
* }
* }
* }
* }
* catch
* {
* print(error)
* }
* print(userty)
* }
* }
* task.resume()
* }
* }

1. **Edit User View Controller Source Code**

* import UIKit
* class editUserViewController: UIViewController {
* @IBOutlet weak var userIdLabel: UILabel!
* @IBOutlet weak var userNameField: UITextField!
* @IBOutlet weak var userDeptField: UITextField!
* var userid = String()
* @IBOutlet weak var userTypeLabel: UILabel!
* var username = String()
* var userdept = String()
* var usertype = String()
* override func viewDidLoad() {
* super.viewDidLoad()
* userIdLabel.text = userid
* userNameField.placeholder = username
* userNameField.text = username
* userDeptField.placeholder = userdept
* userDeptField.text = userdept
* userTypeLabel.text = usertype
* let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action: "dismissKeyboard")
* view.addGestureRecognizer(tap)

* }

* func dismissKeyboard() {
* view.endEditing(true)
* }
* override func didReceiveMemoryWarning() {
* super.didReceiveMemoryWarning()

}

* @IBAction func editUserBtn(\_ sender: Any) {
* let urlString = "http://localhost/api/edituser.php"
* let urlRequest = URL(string: urlString)
* var request = URLRequest(url: urlRequest! as URL)
* request.httpMethod = "POST"
* let userid2 = userid
* let usern = userNameField.text
* let deptn = userDeptField.text
* let utype = usertype
* let parameters = "userid="+userid2+"&department="+deptn!+"&username="+usern!+"&usertype="+utype
* request.httpBody = parameters.data(using: String.Encoding.utf8)
* if(userNameField == nil || userDeptField == nil )
* {
* print("Hata! Lütfen boş alan bırakmayınız")
* }
* else{
* let task = URLSession.shared.dataTask(with: request as URLRequest) { (data, response, error) in
* if error != nil // Hata boş değilse, HATA VARSA
* {
* print(error!)
* }
* do
* {
* let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers) as? NSDictionary
* if let JsonParse = json{

var type:String!

* var status:String!
* type = JsonParse["type"] as! String?
* status = JsonParse["message"] as! String?
* print(type)
* print(status)
* if(status == "UserEdited")
* {

print("edited okey")

* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "User Updated", message: "User editing process completed succesfully", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier: "editOkSegue", sender: self)
* }))
* self.present(alert, animated: true, completion: nil)
* }
* }
* else
* {
* let alert = UIAlertController(title: "Error!", message: "Errors happened when trying to edit user!", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler: nil))
* self.present(alert, animated: true, completion: nil)
* }

* }
* }
* catch
* {
* print(error)
* }
* }
* task.resume()
* }
* }
* override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
* }
* }

1. **Add User View Controller Source Code**

* import UIKit
* import Alamofire
* import DLRadioButton
* class addUserViewController: UIViewController {
* @IBOutlet weak var userIdField: UITextField!
* @IBOutlet weak var userNameField: UITextField!
* @IBOutlet weak var deptNameField: UITextField!
* var typeselect: String!
* @IBAction func studentRadio(\_ sender: Any) {
* typeselect = "student"
* print(typeselect)
* }
* @IBAction func insRadio(\_ sender: Any) {
* typeselect = "instructor"
* print(typeselect)
* }
* @IBAction func adminRadio(\_ sender: Any) {
* typeselect = "admin"
* print(typeselect)
* }
* override func viewDidLoad() {
* super.viewDidLoad()
* let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action: "dismissKeyboard")
* view.addGestureRecognizer(tap)
* }
* func dismissKeyboard() {
* view.endEditing(true)
* }
* override func didReceiveMemoryWarning() {
* super.didReceiveMemoryWarning()
* }
* @IBAction func addUserButton(\_ sender: Any) {
* if (typeselect == nil || userIdField.text == "" || userNameField.text == "" || deptNameField.text == "")
* {
* print("type not selected - missing textfields")
* }
* else
* {



* let urlString = "<http://localhost/api/adduser.php>"
* let urlRequest = URL(string: urlString)
* var request = URLRequest(url: urlRequest! as URL)
* request.httpMethod = "POST"
* let userid = userIdField.text
* let usern = userNameField.text
* let deptn = deptNameField.text
* let pass = userIdField.text
* let parameters = "userid="+userid!+"&password="+pass!+"&department="+deptn!+"&username="+usern!+"&usertype="+typeselect!
* request.httpBody = parameters.data(using: String.Encoding.utf8)
* if(userIdField == nil || userNameField == nil || deptNameField == nil)
* {
* print("Hata! Lütfen boş alan bırakmayınız")
* }
* else{
* let task = URLSession.shared.dataTask(with: request as URLRequest) { (data, response, error) in
* if error != nil // Hata boş değilse, HATA VARSA
* {
* print(error!)
* }
* do
* {
* let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers) as? NSDictionary
* if let JsonParse = json{
* var type:String!
* var status:String!
* type = JsonParse["type"] as! String?
* status = JsonParse["message"] as! String?
* print(type)
* print(status)
* if(status == "UserAddedSuccess")
* {
* print("added okey")
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "User Added", message: "User adding process completed succesfully", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier: "goUsersSegue", sender: userid)
* }))
* self.present(alert, animated: true, completion: nil)
* }
* }
* else
* {
* print("elsegeliyor")
* OperationQueue.main.addOperation {
* let alert = UIAlertController(title: "User Adding Failed", message: "User adding process failed", preferredStyle: .alert)
* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* }))
* self.present(alert, animated: true, completion: nil)
* }
* }

* }
* }
* catch
* {
* print(error)
* }
* print(userid)
* }
* task.resume()
* }
* }
* }
* }

1. **Delete User View Controller Source Code**

* import UIKit
* class deleteUserViewController: UIViewController {
* @IBOutlet weak var useridLabel: UILabel!
* @IBOutlet weak var userNameLabel: UILabel!
* @IBOutlet weak var userDepartmentLabel: UILabel!
* @IBOutlet weak var userTypeLabel: UILabel!
* var userid = String()
* var username = String()
* var usertype = String()
* var deptname = String()
* override func viewDidLoad() {

super.viewDidLoad()

print(userid)

print(username)

print(usertype)

print(deptname)

useridLabel.text = userid

userNameLabel.text = username

userDepartmentLabel.text = deptname

userTypeLabel.text = usertype

* }
* override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

* }
* @IBAction func editBtn(\_ sender: Any) {

DispatchQueue.main.async {

* + - let storyboard:UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
    - let editVC = storyboard.instantiateViewController(withIdentifier: "editPage") as! editUserViewController
    - editVC.userid = self.userid
    - editVC.username = self.username
    - editVC.userdept = self.deptname
    - editVC.usertype = self.usertype
    - self.present(editVC, animated:true, completion: nil)

}

* }
* @IBAction func deleteBtn(\_ sender: Any) {

let refreshAlert = UIAlertController(title: "Delete User", message: "User will be deleted. Are you sure?", preferredStyle: UIAlertControllerStyle.alert)

refreshAlert.addAction(UIAlertAction(title: "Ok", style: .default, handler: { (action: UIAlertAction!) in

* + - let urlString = "http://localhost/api/deleteuser.php"
    - let urlRequest = URL(string: urlString)
    - var request = URLRequest(url: urlRequest! as URL)
    - request.httpMethod = "POST"
    - let parameters = "userid="+self.userid
    - request.httpBody = parameters.data(using: String.Encoding.utf8)
      * let task = URLSession.shared.dataTask(with: request as URLRequest) { (data, response, error) in

do

{

let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers) as? NSDictionary

if let JsonParse = json{

var type:String!

var status:String!

type = JsonParse["type"] as! String?

status = JsonParse["message"] as! String?

print(type)

print(status)

if(status == "UserDeleted")

{

print("delete okey")

OperationQueue.main.addOperation {

let alert = UIAlertController(title: "User Deleted", message: "User deleting process completed succesfully", preferredStyle: .alert)

* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier: "deleteBackSegue", sender: self)

}))

self.present(alert, animated: true, completion: nil)

}

}

}

else

{

print("elsegeliyor")

}

}

catch

{

print(error)

}

* + - * }
      * task.resume()

}))

refreshAlert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: { (action: UIAlertAction!) in

* + - print("Handle Cancel Logic here")

}))

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

* }
* }

1. **Users Table View Cell**

* import UIKit
* class usersTableViewCell: UITableViewCell {
* @IBOutlet weak var userTypeLabel1: UILabel!
* @IBOutlet weak var userTypeLabel2: UILabel!
* @IBOutlet weak var userIdLabel: UILabel!
* @IBOutlet weak var userNameLabel: UILabel!
* @IBOutlet weak var deptNameLabel: UILabel!
* @IBOutlet weak var userTypeLabel3: UILabel!
* override func awakeFromNib() {

super.awakeFromNib()

* }
* override func setSelected(\_ selected: Bool, animated: Bool) {

super.setSelected(selected, animated: animated)

* }
* }

1. **Users Table View Controller Source Code**

* import UIKit
* import Alamofire
* class usersTableViewController: UIViewController , UITableViewDelegate, UITableViewDataSource {
* @IBOutlet weak var tableView2: UITableView!
* var userid = [String]()
* var username = [String]()
* var deptname = [String]()
* var usertype = [String]()
* override func viewDidLoad() {

super.viewDidLoad()

tableView2.rowHeight = UITableViewAutomaticDimension

tableView2.estimatedRowHeight = 150

self.tableView2.rowHeight = 145.0

* Alamofire.request("<http://localhost/api/users.php>").responseJSON { response in
  + - self.retrieveData(JSONData: response.data!)

}

* }
* func retrieveData(JSONData: Data)
* {

do

{

* + - let resultJSON = try JSONSerialization.jsonObject(with: JSONData, options: .mutableContainers)
    - if let usrs = resultJSON as? NSArray {
      * for i in 0..<usrs.count
      * {

if let usrall = usrs[i] as? NSDictionary

{

if let uid = usrall["userid"] as? String

{

self.userid.append(uid)

}

* + - * + if let uname = usrall["name"] as? String

{

self.username.append(uname)

}

if let dname = usrall["deptname"] as? String

{

self.deptname.append(dname)

}

if let type = usrall["utype"] as? String

{

self.usertype.append(type)

}

}

* + - * }
      * self.tableView2.reloadData()
      * print(username)
    - }

}

catch

{

* + - print(error)

}

* }
* override func didReceiveMemoryWarning() {
  + super.didReceiveMemoryWarning()
* }
* func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return userid.count

* }
* public func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell
* {

let cell = tableView2.dequeueReusableCell(withIdentifier: "userCell", for: indexPath) as! usersTableViewCell

cell.userIdLabel.text = userid[indexPath.row]

cell.userTypeLabel1.text = usertype[indexPath.row] + " ID:"

cell.userTypeLabel2.text = usertype[indexPath.row] + " Name:"

cell.userTypeLabel3.text = usertype[indexPath.row]

cell.userNameLabel.text = username[indexPath.row]

cell.deptNameLabel.text = deptname[indexPath.row]

return cell

* }
* func tableView(\_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) { NSLog("You selected cell number: \(indexPath.row)!")

let suserid = userid[indexPath.row]

let susertype = usertype[indexPath.row]

let susername = username[indexPath.row]

let sdept = deptname[indexPath.row]

* + DispatchQueue.main.async {
    - let storyboard:UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
    - let deleteVC = storyboard.instantiateViewController(withIdentifier: "editDeletePage") as! deleteUserViewController
    - deleteVC.userid = suserid
    - deleteVC.username = susername
    - deleteVC.deptname = sdept
    - deleteVC.usertype = susertype
    - self.present(deleteVC, animated:true, completion: nil)

}

* }
* }

1. **Courses Table View Controller Source Code**

import UIKit

import Alamofire

class coursesTableViewController: UITableViewController {

@IBOutlet weak var tableView3: UITableView!

var courseid = [String]()

var coursename = [String]()

var deptname = [String]()

var insid = [String]()

var gid = [String]()

override func viewDidLoad() {

super.viewDidLoad()

print("ok")

tableView3.rowHeight = UITableViewAutomaticDimension

tableView3.estimatedRowHeight = 150

self.tableView3.rowHeight = 145.0

Alamofire.request("<http://localhost/api/courses.php>").responseJSON { response in

self.retrieveData(JSONData: response.data!)

}

}

func retrieveData(JSONData: Data)

{

do

{

let resultJSON = try JSONSerialization.jsonObject(with: JSONData, options: .mutableContainers)

if let crs = resultJSON as? NSArray

{

for i in 0..<crs.count

{

if let courseall = crs[i] as? NSDictionary

{

if let cid = courseall["course\_id"] as? String

{

self.courseid.append(cid)

print(courseid)

}

if let cname = courseall["course\_name"] as? String

{

self.coursename.append(cname)

}

if let dname = courseall["dept\_name"] as? String

{

self.deptname.append(dname)

}

if let insid = courseall["instructor\_id"] as? String

{

self.insid.append(insid)

}

if let gidd = courseall["id"] as? String

{

self.gid.append(gidd)

}

}

}

self.tableView3.reloadData()

print(coursename)

}

}

catch

{

print(error)

}

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

}

override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return courseid.count

}

public override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell

{

let cell = tableView3.dequeueReusableCell(withIdentifier: "coursesCell", for: indexPath) as! coursesTableViewCell

cell.courseIdLabel.text = self.courseid[indexPath.row]

cell.courseNameLabel.text = self.coursename[indexPath.row]

cell.departmentLabel.text = self.deptname[indexPath.row]

cell.instructorIdLabel.text = self.insid[indexPath.row]

return cell

}

}

1. **Add Courses View Controller Source Code**

* import UIKit
* class addCourseViewController: UIViewController {
* @IBOutlet weak var insIdField: UITextField!
* @IBOutlet weak var deptField: UITextField!
* @IBOutlet weak var courseNameField: UITextField!
* @IBOutlet weak var courseIdField: UITextField!
* override func viewDidLoad() {

super.viewDidLoad()

let tap: UITapGestureRecognizer = UITapGestureRecognizer(target: self, action: "dismissKeyboard")

view.addGestureRecognizer(tap)

* }
* func dismissKeyboard() {
  + view.endEditing(true)
* }
* override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

* }
* @IBAction func addCourseBtn(\_ sender: Any)
* {
  + if (false)
* {
* }
* else
* {
  + let urlString = "[http://localhost/api/addcourse](http://localhost/api/adduser.php)"
  + let urlRequest = URL(string: urlString)
  + var request = URLRequest(url: urlRequest! as URL)
  + request.httpMethod = "POST"
  + let courseid = courseIdField.text
  + let coursename = courseNameField.text
  + let deptn = deptField.text
  + let insid = insIdField.text
  + let parameters = "course\_id="+courseid!+"&course\_name="+coursename!+"&dept\_name="+deptn!+"&instructor\_id="+insid!
  + request.httpBody = parameters.data(using: String.Encoding.utf8)
  + if(courseIdField == nil || courseNameField == nil || insIdField == nil || courseIdField == nil)
  + {
    - print("Hata! Lütfen boş alan bırakmayınız")
    - let alert = UIAlertController(title: "Error", message: "Please fill the textfields completely!!", preferredStyle: .alert)
    - alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
    - }))
    - self.present(alert, animated: true, completion: nil)
  + }
  + else{
    - let task = URLSession.shared.dataTask(with: request as URLRequest) { (data, response, error) in
      * if error != nil
      * {
        + print(error!)
      * }
      * do
      * {
        + let json = try JSONSerialization.jsonObject(with: data!, options: .mutableContainers) as? NSDictionary

if let JsonParse = json{

var type:String!

var status:String!

type = JsonParse["type"] as! String?

status = JsonParse["message"] as! String?

print(type)

print(status)

if(status == "CourseAddedSuccess")

{

print("added okey")

OperationQueue.main.addOperation {

let alert = UIAlertController(title: "Course Added", message: "Course adding process completed succesfully", preferredStyle: .alert)

* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in
* self.performSegue(withIdentifier: "goCoursesSegue", sender: self)

}))

self.present(alert, animated: true, completion: nil)

}

}

else

{

OperationQueue.main.addOperation {

let alert = UIAlertController(title: "Course Adding Error", message: "Course adding process failed!!", preferredStyle: .alert)

* alert.addAction(UIAlertAction(title:"OK!", style:UIAlertActionStyle.default, handler:{ action in

}))

self.present(alert, animated: true, completion: nil)

print("elsegeliyor")

}

* + - * + }
        + }
      * }
      * catch
      * {
        + print(error)
      * }
      * print(courseid)
    - }
    - task.resume()

}

}

* }
* }

1. **Courses Table View Cell Source Code**

import UIKit

class coursesTableViewCell: UITableViewCell {

@IBOutlet weak var courseIdLabel: UILabel!

@IBOutlet weak var courseNameLabel: UILabel!

@IBOutlet weak var departmentLabel: UILabel!

@IBOutlet weak var instructorIdLabel: UILabel!

override func awakeFromNib() {

super.awakeFromNib()

}

override func setSelected(\_ selected: Bool, animated: Bool) {

super.setSelected(selected, animated: animated)

}

}