MSM人員到公系統(ios版)

摘要

一.系統功能簡介--------------------------p.1

二.系統架構與環境-----------------------p.1

三.app畫面與功能演示------------------p.2

四.前端app程式碼------------------------p.8

五.資料庫路徑與後端程式碼------------p.32

六.下載與使用方式(附專案檔連結)----p.52

七.權限說明與職位帳號對照表---------p.53

八.本人/本開發者聯絡方式--------------p.58

一.系統功能簡介

MSM人員到公系統是一個打卡到公程式，能讓使用者提交或查詢今日到公情形，

其功能有：登入，註冊，查詢密碼，變更密碼，提交到公情形並記錄當下時間，依使用者權限查詢到公資料等，其中對資料的操作與更新都儲存於資料庫

二.系統架構與環境

使用xcode開發，前端使用swift，後端使用php連接mysql，

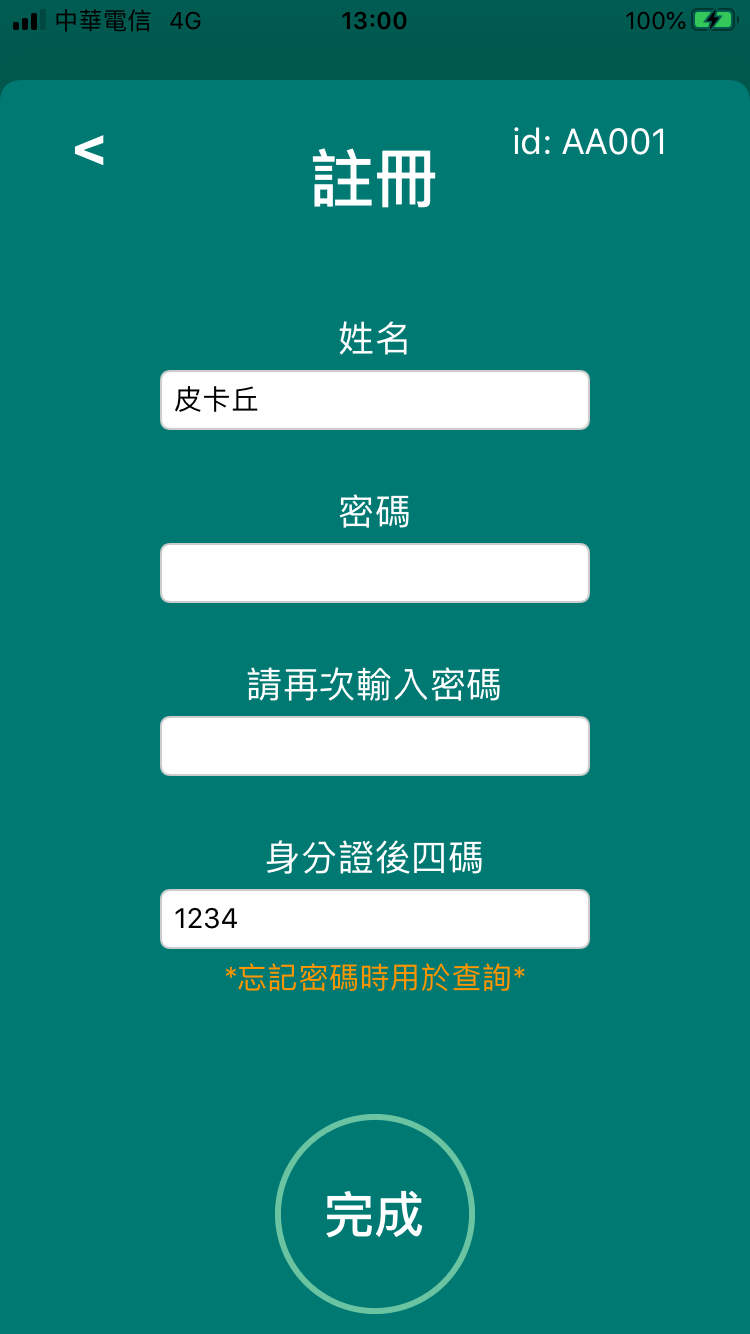
資料庫網址為msmiosappdb.online架設在hostinger.com

三. app畫面與功能演示

1.登入畫面



2.註冊畫面



3.登入成功後



4.提交到公情形



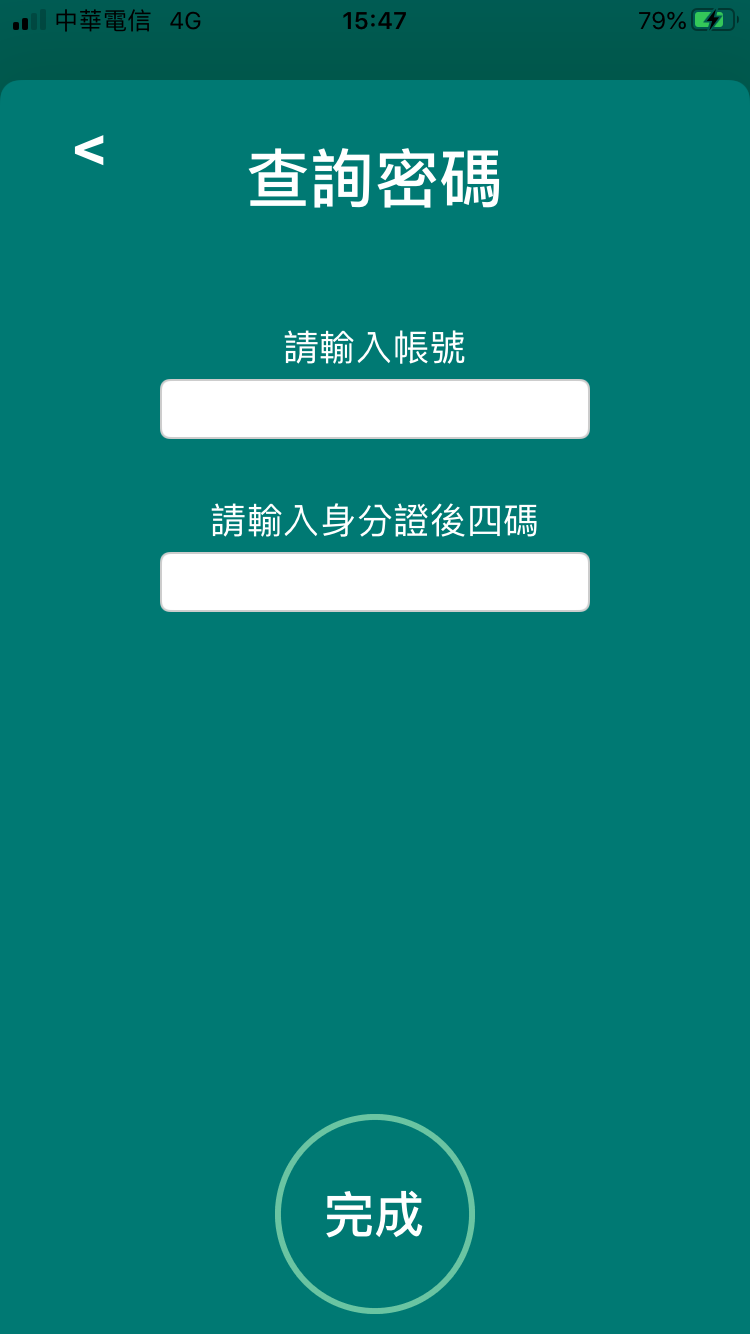
5.依帳號權限查詢到公情形



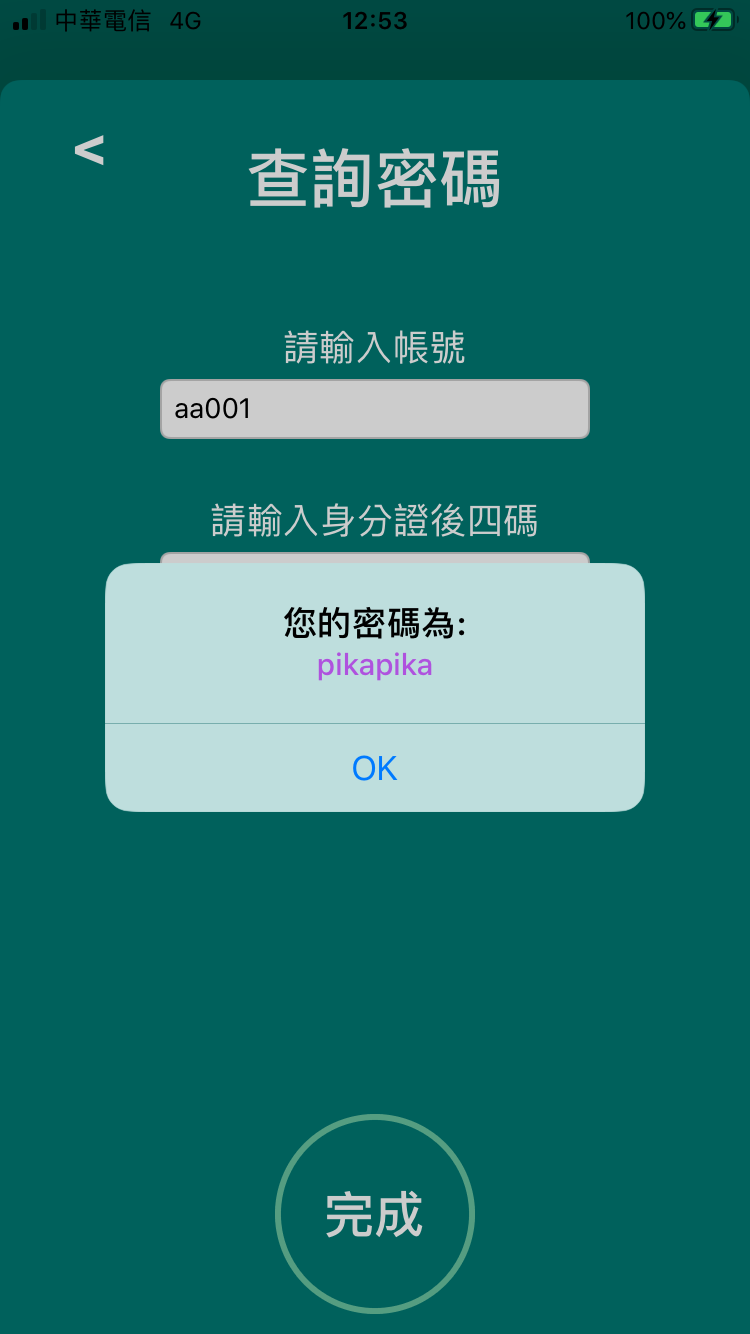
6.點擊按鈕會顯示事由



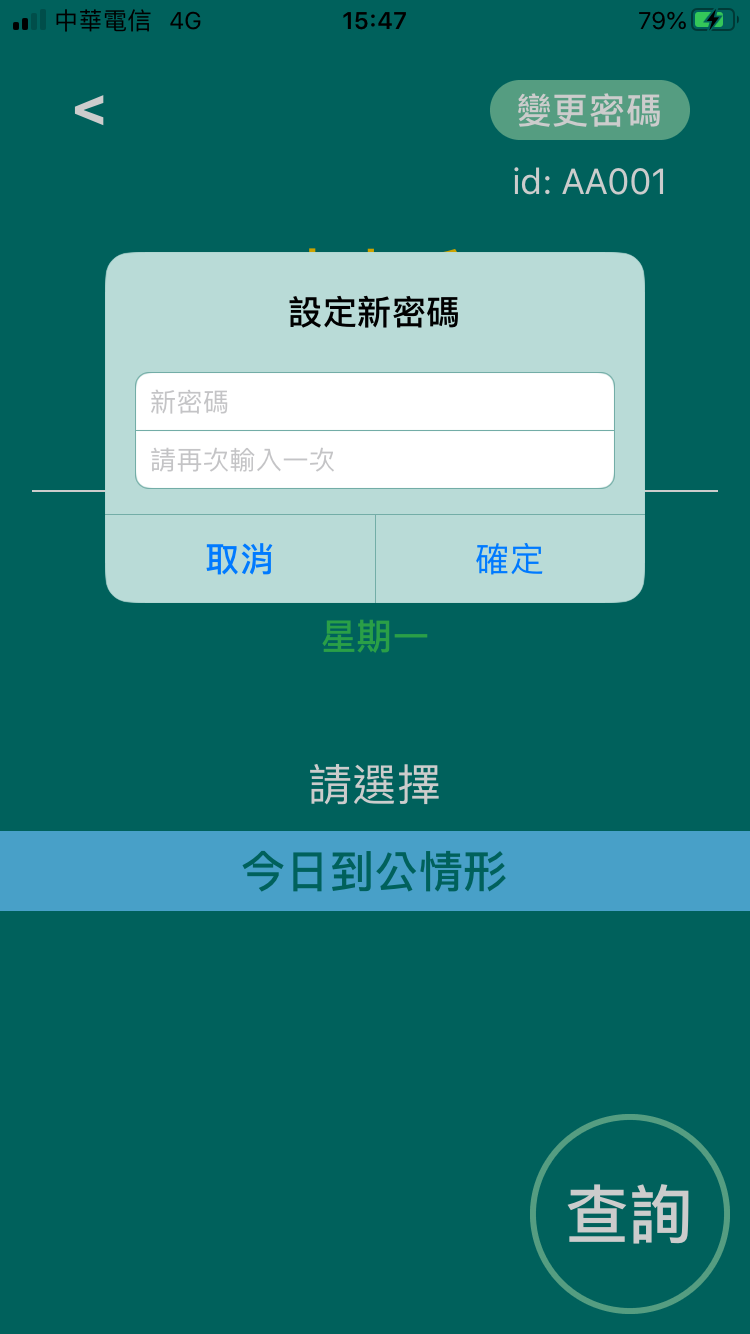
7.查詢密碼



8.查詢密碼成功



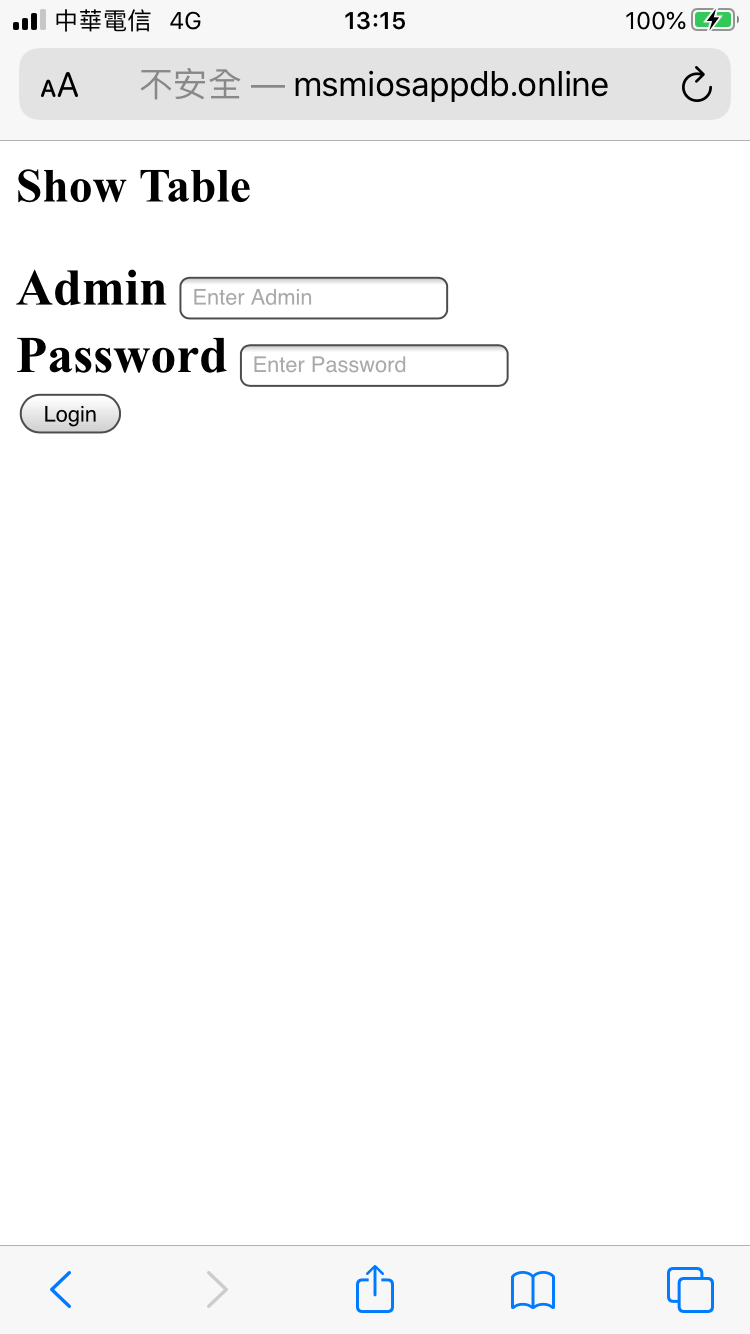
9.更改密碼

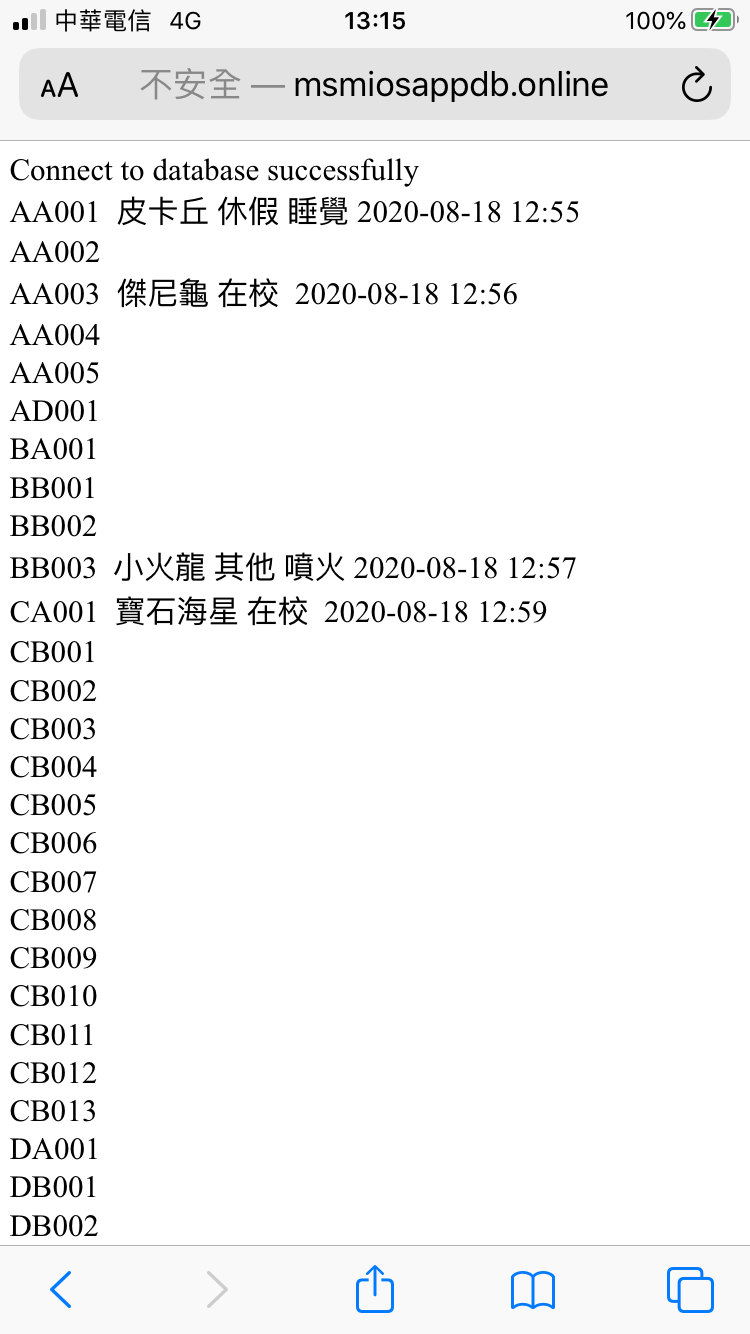


10.在網頁上msmisoappdb.online檢視資料表



11.輸入管理者帳號(見第五章資料庫)



12.顯示資料表

四.前端app程式碼

前端頁面結構



page1.swift

import UIKit

class page1: UIViewController{

@IBOutlet weak var account: UITextField!

@IBOutlet weak var password: UITextField!

@IBOutlet weak var login\_button: UIButton!

@IBOutlet weak var register\_button: UIButton!

@IBOutlet weak var forgot\_password: UIButton!

@IBOutlet weak var register\_reminder: UILabel!

@IBOutlet weak var register\_reminder2: UILabel!

var account\_word:String?

var account\_check:String?

var password\_word:String?

var account\_restore:String?

var password\_restore:String?

@IBAction func button\_event(\_ sender: UIButton) {

account\_word=account.text!

password\_word=password.text!

let whitespace = NSCharacterSet.whitespaces

if sender==login\_button{

if (account\_word == ""||password\_word == ""){

ok\_alert("請輸入帳號密碼")

}

else if account\_word!.rangeOfCharacter(from: whitespace) != nil || password\_word!.rangeOfCharacter(from: whitespace) != nil{

ok\_alert("不可輸入空白字元")

}

else{

self.loginAccountandPw()

}

}

if sender==register\_button{

setup\_alert(1)

}

if sender==forgot\_password{

performSegue(withIdentifier: "go\_forgot\_password", sender: self)

}

}

func change\_alert\_color(\_ controller: UIAlertController,\_ string: String?){

controller.setValue(NSAttributedString(

string: string!,attributes: [NSAttributedString.Key.font: UIFont.systemFont(ofSize: 15,weight: UIFont.Weight.medium),NSAttributedString.Key.foregroundColor : UIColor.systemPurple])

,forKey: "attributedMessage")

}

func ok\_alert(\_ text: String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

present(controller\_show\_ok,animated: true,completion: nil)

}

func setup\_alert(\_ condiction: Int){

if condiction==1{

var message\_text:String?

let text="請輸入帳號"

let controller\_check\_account = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_check\_account.addTextField{

(textField) in

//textField.placeholder="帳號格式：(例如)XX001"

textField.placeholder=""

}

let ok\_check\_account = UIAlertAction(title: "送出",style: .default){

action in

let whitespace = NSCharacterSet.whitespaces

if controller\_check\_account.textFields?[0].text! == ""{

message\_text="欄位不可留空"

controller\_check\_account.message=message\_text

self.change\_alert\_color(controller\_check\_account, message\_text)

self.present(controller\_check\_account, animated: true, completion: nil)

}else if controller\_check\_account.textFields?[0].text!.rangeOfCharacter(from: whitespace) != nil{

message\_text="不可輸入空白字元"

controller\_check\_account.message=message\_text

self.change\_alert\_color(controller\_check\_account, message\_text)

self.present(controller\_check\_account, animated: true, completion: nil)

}

else{

self.account\_check=controller\_check\_account.textFields?[0].text!

self.checkAccount()

}

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

controller\_check\_account.addAction(ok\_check\_account)

controller\_check\_account.addAction(noAction)

present(controller\_check\_account, animated: true, completion: nil)

}

if condiction == 2{

let submit\_or\_not = UIAlertController(title: "此帳號已經註冊，是否再次註冊?", message: nil, preferredStyle: .alert)

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

self.performSegue(withIdentifier: "go\_register", sender: self)

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

submit\_or\_not.addAction(yesAction)

submit\_or\_not.addAction(noAction)

present(submit\_or\_not,animated: true,completion: nil)

}

}

func secure\_password(\_ sender: UITextField){

sender.isSecureTextEntry=true

sender.textContentType=UITextContentType(rawValue: "")

}

func mod\_constraints(){

register\_reminder.centerXAnchor.constraint(equalTo: register\_button.centerXAnchor).isActive=true

register\_reminder2.centerXAnchor.constraint(equalTo: register\_button.centerXAnchor).isActive=true

}

func mod\_button(){

//106,196,162 藍綠色

//0,175,210 淺藍色

//0,121,115 深綠色

//195,152,78 棕橘色

register\_button.backgroundColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1)

register\_button.layer.cornerRadius=10.0

login\_button.backgroundColor=UIColor.systemBlue

login\_button.layer.cornerRadius=15.0

}

func loginAccountandPw(){

let login\_mode:String = "account\_pw"

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/login.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/login.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account.text!)&login\_mode\_=\(login\_mode)&pw\_=\(password.text!)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

self.parseJson(data!)

}

task.resume()

}

func checkAccount(){

let login\_mode:String = "account"

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/login.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/login.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_check!)&login\_mode\_=\(login\_mode)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

self.parseJson(data!)

}

task.resume()

}

func parseJson(\_ data:Data){

DispatchQueue.main.async{

let string = String(data: data,encoding: String.Encoding.utf8)

if string == "new account"{

self.performSegue(withIdentifier: "go\_register", sender: self)

}

if string == "already registered"{

self.setup\_alert(2)

}

if (string == "account not exist" || string == "account not exist 2"){

self.ok\_alert("帳號錯誤")

}

if string == "new account 2"{

self.ok\_alert("此帳號尚未註冊")

}

if string == "login successed"{

self.performSegue(withIdentifier: "after\_login", sender: self)

}

if string == "password not exist"{

self.ok\_alert("密碼錯誤")

}

}

}

override func touchesEnded(\_ touch: Set<UITouch>,with event: UIEvent?){

account.resignFirstResponder()

password.resignFirstResponder()

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

}

override func viewDidLoad() {

super.viewDidLoad()

mod\_button()

mod\_constraints()

secure\_password(password)

account.text=account\_restore

password.text=password\_restore

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?){

if segue.identifier == "after\_login"{

let vc = segue.destination as! page2

vc.account\_logined=account\_word

vc.password\_logined=password\_word

}

if segue.identifier == "go\_register"{

let vc = segue.destination as! page\_register

vc.account\_checking=account\_check

}

}

}

page2.swift

import UIKit

class page2: UIViewController{

@IBOutlet weak var username: UILabel!

//@IBOutlet weak var position: UILabel!

@IBOutlet weak var back\_login\_button: UIButton!

@IBOutlet weak var currentTime: UILabel!

@IBOutlet weak var change\_password: UIButton!

@IBOutlet weak var show\_account: UILabel!

//@IBOutlet weak var name\_position: UIImageView!

@IBOutlet var reason\_buttons: [UIButton]!

@IBOutlet weak var select\_a\_reason: UIButton!

@IBOutlet weak var in\_school: UIButton!

@IBOutlet weak var vacation: UIButton!

@IBOutlet weak var other\_reason: UIButton!

@IBOutlet weak var current\_state: UILabel!

@IBOutlet weak var weekofday: UILabel!

@IBOutlet weak var reason\_message: UITextField!

@IBOutlet weak var submit: UIButton!

@IBOutlet weak var inquire: UIButton!

let date=Date()

let dateformatter = DateFormatter()

let dateComponents = Calendar.current.dateComponents(in: TimeZone.current, from: Date())

var whichday:String?

var account\_logined:String?

var password\_logined:String?

var today\_state:String?

let state\_array=["在校","休假","其他"]

var new\_password:String?

var changePW:Bool=false

@IBAction func button\_event(\_ sender: UIButton) {

if sender==back\_login\_button{

performSegue(withIdentifier: "back\_to\_login", sender: self)

}

if sender==change\_password{

setup\_alert(2)

}

if sender==select\_a\_reason{

reason\_buttons.forEach{ (button) in

UIView.animate(withDuration: 0.3 , animations: {

button.isHidden = !button.isHidden

self.view.layoutIfNeeded()

})

}

submit.setTitle("提交", for: .normal)

submit.titleLabel?.font=UIFont.boldSystemFont(ofSize: 32)

submit.setTitleColor(UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1), for: .normal)

submit.backgroundColor=UIColor.white

submit.layer.borderWidth=3

submit.layer.borderColor=UIColor.white.cgColor

submit.isHidden=true

//inquire.isHidden=true

reason\_message.isHidden=true

current\_state.isHidden=true

}

if sender==in\_school{

reason\_buttons.forEach{ (button) in

button.isHidden = !button.isHidden

self.view.layoutIfNeeded()

current\_state.text=state\_array[0]

today\_state=current\_state.text

current\_state.textColor=UIColor.systemGreen

reason\_message.isHidden=true

submit.isHidden=false

current\_state.isHidden=false

}

}

if sender==vacation{

reason\_buttons.forEach{ (button) in

button.isHidden = !button.isHidden

self.view.layoutIfNeeded()

current\_state.text=state\_array[1]

today\_state=current\_state.text

current\_state.textColor=UIColor.systemYellow

reason\_message.isHidden=false

submit.isHidden=false

current\_state.isHidden=false

}

}

if sender==other\_reason{

reason\_buttons.forEach{ (button) in

button.isHidden = !button.isHidden

self.view.layoutIfNeeded()

current\_state.text=state\_array[2]

today\_state=current\_state.text

current\_state.textColor=UIColor.init(red: 225/255, green: 110/255, blue: 121/255, alpha: 1)

reason\_message.isHidden=false

submit.isHidden=false

current\_state.isHidden=false

}

}

if sender==submit{

if submit.titleLabel?.text=="提交"{

if today\_state=="在校"{

setup\_alert(1)

}

if today\_state=="其他" || today\_state=="休假"{

if reason\_message.text==""{

ok\_alert("請輸入事由")

}else{

setup\_alert(1)

}

}

}

}

if sender==inquire{

performSegue(withIdentifier: "go\_calldb", sender: self)

}

}

func ok\_alert(\_ text: String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

present(controller\_show\_ok,animated: true,completion: nil)

}

func change\_alert\_color(\_ controller: UIAlertController,\_ string: String?){

controller.setValue(NSAttributedString(

string: string!,attributes: [NSAttributedString.Key.font: UIFont.systemFont(ofSize: 15,weight: UIFont.Weight.medium),NSAttributedString.Key.foregroundColor : UIColor.systemPurple])

,forKey: "attributedMessage")

}

func setup\_alert(\_ condiction: Int){

if condiction==1{

let submit\_or\_not = UIAlertController(title: "確定提交嗎?", message: nil, preferredStyle: .alert)

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

self.todaySubmit()

self.submit.setTitle("提交成功!", for: .normal)

self.submit.titleLabel?.font = UIFont.boldSystemFont(ofSize: 20)

self.submit.setTitleColor(UIColor.white, for: .normal)

self.submit.backgroundColor = UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

self.submit.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

//self.inquire.isHidden=false

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

submit\_or\_not.addAction(yesAction)

submit\_or\_not.addAction(noAction)

present(submit\_or\_not,animated: true,completion: nil)

}

if condiction==2{

let text="設定新密碼"

var message\_text:String?

let change\_password\_alert = UIAlertController(title: text, message: nil, preferredStyle: .alert)

change\_password\_alert.addTextField{

(textField) in

textField.placeholder="新密碼"

self.secure\_password(textField)

}

change\_password\_alert.addTextField{

(textField) in

textField.placeholder="請再次輸入一次"

self.secure\_password(textField)

}

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

var changePw\_ok:Bool = false

if (change\_password\_alert.textFields?[0].text! == "" || change\_password\_alert.textFields?[1].text! == ""){

message\_text="欄位不可留空"

}else{

let whitespace = NSCharacterSet.whitespaces

if(change\_password\_alert.textFields?[0].text!.rangeOfCharacter(from: whitespace) != nil || change\_password\_alert.textFields?[1].text!.rangeOfCharacter(from: whitespace) != nil){

message\_text="不可輸入空白字元"

}

else if(change\_password\_alert.textFields?[0].text! != change\_password\_alert.textFields?[1].text!){

message\_text="密碼不一致"

}

else if((change\_password\_alert.textFields?[0].text!.count)!<8 || (change\_password\_alert.textFields?[0].text!.count)!>20){

message\_text="密碼長度須介於8~20位之間"

}

else{

changePw\_ok = true

self.new\_password=change\_password\_alert.textFields?[0].text!

self.setup\_alert(3)

}

}

if changePw\_ok == false{

change\_password\_alert.message=message\_text

self.change\_alert\_color(change\_password\_alert,message\_text)

self.present(change\_password\_alert,animated: true,completion: nil)

}

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

change\_password\_alert.addAction(yesAction)

change\_password\_alert.addAction(noAction)

present(change\_password\_alert,animated: true,completion: nil)

}

if condiction==3{

let message\_text = new\_password

let changePW\_or\_not = UIAlertController(title: "是否將密碼變更為：", message: message\_text, preferredStyle: .alert)

let change\_password\_alert\_ok = UIAlertController(title: "密碼變更成功！", message: nil, preferredStyle: .alert)

self.change\_alert\_color(changePW\_or\_not,message\_text)

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

self.present(change\_password\_alert\_ok,animated: true,completion: nil)

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

let okAction = UIAlertAction(title: "OK", style: .default){

action in

self.changePw()

}

changePW\_or\_not.addAction(yesAction)

changePW\_or\_not.addAction(noAction)

change\_password\_alert\_ok.addAction(okAction)

present(changePW\_or\_not,animated: true,completion: nil)

}

if condiction==4{

let text="請輸入事由"

let controller\_reason = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_reason.addTextField{

(textField) in

//textField.placeholder="帳號格式：(例如)XX001"

textField.text=self.reason\_message.text

}

let okAction = UIAlertAction(title: "確定",style: .default){

action in

self.reason\_message.text=controller\_reason.textFields?[0].text!

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

controller\_reason.addAction(okAction)

controller\_reason.addAction(noAction)

present(controller\_reason, animated: true, completion: nil)

}

}

func printTime()->(date: String, week: String){

dateformatter.dateFormat="yyyy-MM-dd"+"\t"+" "+"HH:mm"

let weekday = dateComponents.weekday

let dateString = dateformatter.string(from: self.date)

switch weekday {

case 1:

whichday="星期日"

case 2:

whichday="星期一"

case 3:

whichday="星期二"

case 4:

whichday="星期三"

case 5:

whichday="星期四"

case 6:

whichday="星期五"

case 7:

whichday="星期六"

default:

break

}

return (dateString,whichday!)

}

func secure\_password(\_ sender: UITextField){

sender.isSecureTextEntry=true

sender.textContentType=UITextContentType(rawValue: "")

}

func mod\_button(){

submit.setTitleColor(UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1), for: .normal)

submit.backgroundColor=UIColor.white

submit.layer.borderWidth=3

submit.layer.borderColor=UIColor.white.cgColor

//submit.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

inquire.setTitleColor(UIColor.white, for: .normal)

inquire.backgroundColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

inquire.layer.borderWidth=3

inquire.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

select\_a\_reason.backgroundColor=UIColor.systemTeal

select\_a\_reason.setTitleColor(UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1), for: .normal)

change\_password.backgroundColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1)

change\_password.layer.cornerRadius=15.0

}

func mod\_label(){

//name\_position.alpha=0.2

//currentTime.textColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1)

//weekofday.textColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1)

currentTime.textColor=UIColor.systemGreen

weekofday.textColor=UIColor.systemGreen

username.textColor=UIColor.systemYellow

//position.textColor=UIColor.systemYellow

}

func circular\_button(\_ object1:AnyObject){

object1.layer.cornerRadius=object1.frame.width/2

object1.layer.masksToBounds=true

}

func todaySubmit(){

let submit\_time=self.printTime()

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/todaySubmit.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/todaySubmit.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_logined!)&pw\_=\(password\_logined!)&status\_=\(today\_state!)&reason\_=\(reason\_message.text!)&submit\_time\_=\(submit\_time.date)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

}

task.resume()

}

func changePw(){

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/changePw.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/changePw.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_logined!)&new\_pw\_=\(new\_password!)&old\_pw\_=\(password\_logined!)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

DispatchQueue.main.async{

self.performSegue(withIdentifier: "back\_to\_login", sender: self)

}

}

task.resume()

}

func showProfile(){

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/profile.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/profile.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_logined!)&pw\_=\(password\_logined!)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

self.parseJson(data!)

}

task.resume()

}

func parseJson(\_ data:Data){

DispatchQueue.main.async {

do{

let jsonArray = try JSONSerialization.jsonObject(with: data, options: []) as! [Any]

let jsonDict = jsonArray[0] as! [String:String]

self.username.text=jsonDict["name"]!

//self.position.text=jsonDict["position"]!

}catch{

print("There was an error")

}

}

}

@objc func ShowTextField(textField: UITextField){

setup\_alert(4)

}

override func touchesEnded(\_ touch: Set<UITouch>,with event: UIEvent?){

reason\_message.resignFirstResponder()

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

}

override func viewDidLoad() {

super.viewDidLoad()

showProfile()

current\_state.text=""

reason\_message.isHidden=true

submit.isHidden=true

//inquire.isHidden=true

circular\_button(submit)

circular\_button(inquire)

mod\_label()

mod\_button()

let time\_result=self.printTime()

currentTime.text=time\_result.date

weekofday.text=time\_result.week

show\_account.text="id: "+account\_logined!.uppercased()

reason\_message.addTarget(self, action: #selector(ShowTextField), for: .touchDown)

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

if segue.identifier == "back\_to\_login"{

let vc = segue.destination as! page1

vc.account\_restore = account\_logined!

vc.password\_restore = password\_logined!

}

if segue.identifier == "go\_calldb"{

let vc = segue.destination as! page\_calldb

vc.account\_logined\_ = account\_logined!

vc.password\_logined\_ = password\_logined!

vc.today\_state\_ = today\_state

vc.reason\_message\_ = reason\_message.text

}

}

}

page\_calldb.swift

import UIKit

let tableViewCell\_inst = tableViewCell()

class page\_calldb: UIViewController,UITableViewDataSource,UITableViewDelegate{

var account\_logined\_:String?

var password\_logined\_:String?

var today\_state\_:String?

var reason\_message\_:String?=""

var db\_struct\_data = [db\_struct]()

let date=Date()

let dateformatter = DateFormatter()

let dateComponents = Calendar.current.dateComponents(in: TimeZone.current, from: Date())

var whichday:String?

@IBOutlet weak var date\_stackview: UIStackView!

@IBOutlet weak var tableViewData: UITableView!

@IBOutlet weak var back\_page2\_button: UIButton!

@IBOutlet weak var show\_account: UILabel!

@IBAction func button\_event(\_ sender: UIButton){

if sender==back\_page2\_button{

performSegue(withIdentifier: "back\_calldb", sender: self)

}

}

@IBOutlet weak var current\_time: UILabel!

@IBOutlet weak var weekofday: UILabel!

func getItems(){

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/service.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/service.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_logined\_!)&authority\_=\(getCharcter(account\_logined\_,1))&location\_=\(getCharcter(account\_logined\_,0))&alter\_=\(getCharcter(account\_logined\_,2))"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

self.parseJson(data!)

}

task.resume()

}

func parseJson(\_ data:Data){

DispatchQueue.main.async {

var locArray = [db\_struct]()

do{

let jsonArray = try JSONSerialization.jsonObject(with: data, options: []) as! [Any]

for jsonResult in jsonArray{

let jsonDict = jsonResult as! [String:String]

let loc = db\_struct(account: jsonDict["account"]!,

//position: jsonDict["position"]!,

name: jsonDict["name"]!,

password: jsonDict["password"]!,

status: jsonDict["status"]!,

reason: jsonDict["reason"]!,

last\_4\_id: jsonDict["last\_4\_id"]!,

submit\_time: jsonDict["submit\_time"]!)

locArray.append(loc)

}

}catch{

print("There was an error")

}

self.db\_struct\_data=locArray

self.tableViewData.reloadData()

}

}

func mod\_constraints(){

let g = UILayoutGuide()

self.view.addLayoutGuide(g)

g.topAnchor.constraint(equalTo: tableViewData.bottomAnchor).isActive=true

g.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor).isActive=true

let f = UILayoutGuide()

self.view.addLayoutGuide(f)

f.topAnchor.constraint(equalTo: view.safeAreaLayoutGuide.topAnchor).isActive=true

f.bottomAnchor.constraint(equalTo: tableViewData.topAnchor).isActive=true

date\_stackview.translatesAutoresizingMaskIntoConstraints=false

self.view.addSubview(date\_stackview)

date\_stackview.topAnchor.constraint(equalTo: g.topAnchor).isActive=true

date\_stackview.bottomAnchor.constraint(equalTo: g.bottomAnchor).isActive=true

show\_account.translatesAutoresizingMaskIntoConstraints=false

self.view.addSubview(show\_account)

show\_account.topAnchor.constraint(equalTo: f.topAnchor).isActive=true

show\_account.bottomAnchor.constraint(equalTo: f.bottomAnchor).isActive=true

}

func printTime()->(date: String, week: String){

dateformatter.dateFormat="yyyy-MM-dd"+"\t"+"HH:mm"

let weekday = dateComponents.weekday

let dateString = dateformatter.string(from: self.date)

switch weekday {

case 1:

whichday="星期日"

case 2:

whichday="星期一"

case 3:

whichday="星期二"

case 4:

whichday="星期三"

case 5:

whichday="星期四"

case 6:

whichday="星期五"

case 7:

whichday="星期六"

default:

break

}

return (dateString,whichday!)

}

func ok\_alert(\_ text: String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

present(controller\_show\_ok,animated: true,completion: nil)

}

func getCharcter(\_ string: String?,\_ index: Int)->String{

return String(string![string!.index(string!.startIndex, offsetBy: index)])

}

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

return db\_struct\_data.count

}

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

let cell = tableView.dequeueReusableCell(withIdentifier: "BasicCell", for: indexPath) as! tableViewCell

//cell.textLabel?.text = dataformat(cellForRowAt: indexPath)

cell.reason\_button.isHidden=true

cell.submit\_date.isHidden=true

cell.submit\_time.isHidden=true

cell.reason\_button.setTitle("事由", for: .normal)

cell.reason\_button.setTitleColor(UIColor.systemTeal, for: .normal)

cell.reason\_button.layer.borderWidth=1

cell.reason\_button.layer.cornerRadius=10.0

cell.reason\_button.layer.borderColor=UIColor.systemTeal.cgColor

cell.backgroundColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

let selectColor = UIView()

selectColor.backgroundColor = UIColor.systemBlue

cell.selectedBackgroundView = selectColor

cell.submit\_time.widthAnchor.constraint(equalToConstant: cell.submit\_date.frame.width).isActive=true

cell.submit\_time.centerXAnchor.constraint(equalTo: cell.submit\_date.centerXAnchor).isActive=true

//cell.id\_label.text=db\_struct\_data[indexPath.row].account

cell.id\_label.attributedText=NSAttributedString(string: db\_struct\_data[indexPath.row].account, attributes: [.underlineStyle: NSUnderlineStyle.single.rawValue])

cell.id\_label.textColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1)

//cell.id\_label.layer.borderWidth=1

//cell.id\_label.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

//cell.position\_label.text=db\_struct\_data[indexPath.row].position

//cell.position\_label.textColor=UIColor.systemYellow

cell.name\_label.text=db\_struct\_data[indexPath.row].name

cell.name\_label.textColor=UIColor.white

cell.status\_label.text=db\_struct\_data[indexPath.row].status

if(db\_struct\_data[indexPath.row].status == "在校"){

cell.status\_label.textColor=UIColor.systemGreen

}

if(db\_struct\_data[indexPath.row].status == "休假"){

cell.status\_label.textColor=UIColor.systemYellow

}

if(db\_struct\_data[indexPath.row].status == "其他"){

cell.status\_label.textColor=UIColor.init(red: 225/255, green: 110/255, blue: 121/255, alpha: 1)

}

if(db\_struct\_data[indexPath.row].reason != ""){

cell.reason\_button.isHidden=false

cell.actionBlock = {

self.ok\_alert(self.db\_struct\_data[indexPath.row].reason)

}

}

if(db\_struct\_data[indexPath.row].submit\_time != ""){

let post\_time = db\_struct\_data[indexPath.row].submit\_time

cell.submit\_date.isHidden=false

cell.submit\_time.isHidden=false

cell.submit\_date.text = String(post\_time.prefix(10))

cell.submit\_time.text = String(post\_time.suffix(5))

cell.submit\_date.textColor = UIColor.white

cell.submit\_time.textColor = UIColor.white

}

return cell

}

func mod\_label(){

current\_time.textColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

weekofday.textColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

show\_account.text="id: "+account\_logined\_!.uppercased()

show\_account.textColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

}

func mod\_button(){

}

/\*

func dataformat(cellForRowAt indexPath: IndexPath)->String{

return db\_struct\_data[indexPath.row].account + " " + db\_struct\_data[indexPath.row].position + " " + db\_struct\_data[indexPath.row].name + " " + db\_struct\_data[indexPath.row].status + " " + db\_struct\_data[indexPath.row].reason

}\*/

override func viewDidLoad() {

super.viewDidLoad()

tableViewData.delegate = self

tableViewData.dataSource = self

tableViewData.backgroundColor = UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

getItems()

mod\_constraints()

let time\_result=self.printTime()

current\_time.text=time\_result.date

weekofday.text=time\_result.week

mod\_label()

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

if segue.identifier == "back\_calldb"{

let vc = segue.destination as! page2

vc.account\_logined = account\_logined\_!

vc.password\_logined = password\_logined\_!

}

}

}

page\_register.swift

import UIKit

class page\_register: UIViewController {

@IBOutlet weak var fin\_register: UIButton!

@IBOutlet weak var name: UITextField!

@IBOutlet weak var password: UITextField!

@IBOutlet weak var re\_password: UITextField!

@IBOutlet weak var last\_4\_id: UITextField!

@IBOutlet weak var register\_success: UILabel!

@IBOutlet weak var back\_to\_login: UIButton!

@IBOutlet weak var show\_account: UILabel!

var account\_checking:String?

@IBAction func button\_event(\_ sender: UIButton){

if sender == back\_to\_login{

performSegue(withIdentifier: "register\_to\_login", sender: self)

}

if sender == fin\_register{

if register\_success.isHidden==true{

let whitespace = NSCharacterSet.whitespaces

if (name.text == "" || password.text == "" || re\_password.text == "" || last\_4\_id.text == ""){

ok\_alert("欄位不可留空")

}else if(password.text!.rangeOfCharacter(from: whitespace) != nil || re\_password.text!.rangeOfCharacter(from: whitespace) != nil){

ok\_alert("不可輸入空白字元")

}

else{

if password.text != re\_password.text{

ok\_alert("密碼不一致")

}

else if password.text!.count<8 || password.text!.count>20{

ok\_alert("密碼長度須介於8~20位之間")

}

else if(!isStringAnInt(last\_4\_id.text!)){

ok\_alert("後四碼格式錯誤")

}

else if howlongint(last\_4\_id.text!) != 4{

ok\_alert("後四碼格式錯誤")

}

else{

setup\_alert(1)

}

}

}

}

}

func howlongint(\_ string: String) -> Int{

var a = Int(string)

var count:Int = 0

while a! > 0{

a = a!/10

count+=1

}

return count

}

func isStringAnInt(\_ string: String)->Bool{

return Int(string) != nil

}

func ok\_alert(\_ text: String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

present(controller\_show\_ok,animated: true,completion: nil)

}

func setup\_alert(\_ condiction: Int){

if condiction==1{

let submit\_or\_not = UIAlertController(title: "確定提交嗎?", message: nil, preferredStyle: .alert)

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

self.postItem()

self.register\_success.isHidden=false

self.fin\_register.isHidden=true

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

submit\_or\_not.addAction(yesAction)

submit\_or\_not.addAction(noAction)

present(submit\_or\_not,animated: true,completion: nil)

}

}

func mod\_constraints(){

register\_success.centerYAnchor.constraint(equalTo: fin\_register.centerYAnchor).isActive=true

}

func mod\_button(){

fin\_register.backgroundColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

fin\_register.layer.borderWidth=3

fin\_register.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

}

func secure\_password(\_ sender: UITextField){

sender.isSecureTextEntry=true

sender.textContentType=UITextContentType(rawValue: "")

}

func circular\_button(\_ object1:AnyObject){

object1.layer.cornerRadius=object1.frame.width/2

object1.layer.masksToBounds=true

}

func postItem(){

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/register.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/register.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account\_checking!)&name\_=\(name.text!)&pw\_=\(password.text!)&last\_4\_id\_=\(last\_4\_id.text!)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

}

task.resume()

}

override func touchesEnded(\_ touch: Set<UITouch>,with event: UIEvent?){

name.resignFirstResponder()

password.resignFirstResponder()

re\_password.resignFirstResponder()

last\_4\_id.resignFirstResponder()

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

}

override func viewDidLoad() {

super.viewDidLoad()

circular\_button(fin\_register)

mod\_constraints()

mod\_button()

show\_account.text="id: "+account\_checking!.uppercased()

secure\_password(password)

secure\_password(re\_password)

register\_success.isHidden=true

last\_4\_id.keyboardType = .numberPad

}

}

page\_forgot.swift

import UIKit

class page\_forgot: UIViewController {

@IBOutlet weak var last\_4\_id: UITextField!

@IBOutlet weak var account: UITextField!

@IBOutlet weak var back\_to\_login: UIButton!

@IBOutlet weak var submit: UIButton!

@IBAction func button\_event(\_ sender: UIButton){

if sender==submit{

if last\_4\_id.text==""{

ok\_alert("欄位不可留空")

}else{

if(!isStringAnInt(last\_4\_id.text!)){

ok\_alert("後四碼格式錯誤")

}else if howlongint(last\_4\_id.text!) != 4{

ok\_alert("後四碼格式錯誤")

}else{

//setup\_alert(1)

self.forgotPw()

}

}

}

if sender==back\_to\_login{

performSegue(withIdentifier: "forgot\_password\_to\_login", sender: self)

}

}

func mod\_button(){

submit.backgroundColor=UIColor.init(red: 0/255, green: 121/255, blue: 115/255, alpha: 1)

submit.layer.borderWidth=3

submit.layer.borderColor=UIColor.init(red: 106/255, green: 196/255, blue: 162/255, alpha: 1).cgColor

}

func circular\_button(\_ object1:AnyObject){

object1.layer.cornerRadius=object1.frame.width/2

object1.layer.masksToBounds=true

}

func howlongint(\_ string: String) -> Int{

var a = Int(string)

var count:Int = 0

while a! > 0{

a = a!/10

count+=1

}

return count

}

func isStringAnInt(\_ string: String)->Bool{

return Int(string) != nil

}

func change\_alert\_color(\_ controller: UIAlertController,\_ string: String?){

controller.setValue(NSAttributedString(

string: string!,attributes: [NSAttributedString.Key.font: UIFont.systemFont(ofSize: 15,weight: UIFont.Weight.medium),NSAttributedString.Key.foregroundColor : UIColor.systemPurple])

,forKey: "attributedMessage")

}

func ok\_alert(\_ text: String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: nil, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

present(controller\_show\_ok,animated: true,completion: nil)

}

func ok\_alert\_showPw(\_ text: String,\_ message:String){

let okAction = UIAlertAction(title: "OK", style: .default, handler: nil)

let controller\_show\_ok = UIAlertController(title: text, message: message, preferredStyle: .alert)

controller\_show\_ok.addAction(okAction)

self.change\_alert\_color(controller\_show\_ok,message)

present(controller\_show\_ok,animated: true,completion: nil)

}

func setup\_alert(\_ condiction: Int){

if condiction==1{

let submit\_or\_not = UIAlertController(title: "確定提交嗎?", message: nil, preferredStyle: .alert)

let yesAction = UIAlertAction(title: "確定", style: .default){

action in

self.forgotPw()

}

let noAction = UIAlertAction(title: "取消", style: .cancel, handler: nil)

submit\_or\_not.addAction(yesAction)

submit\_or\_not.addAction(noAction)

present(submit\_or\_not,animated: true,completion: nil)

}

}

func forgotPw(){

//let request = NSMutableURLRequest(url: NSURL(string: "https://iosappdb.000webhostapp.com/usage/forgotPw.php")! as URL)

let request = NSMutableURLRequest(url: NSURL(string: "http://msmiosappdb.online/usage/forgotPw.php")! as URL)

request.httpMethod = "POST"

let postString = "account\_=\(account.text!)&last\_4\_id\_=\(last\_4\_id.text!)"

request.httpBody=postString.data(using: String.Encoding.utf8)

let task = URLSession.shared.dataTask(with: request as URLRequest){

(data, response, error) in

if error != nil{

print("error = \(String(describing: error))")

}

print("response = \(String(describing: response))")

let responseString = NSString(data: data!, encoding: String.Encoding.utf8.rawValue)

print("responseString = \(String(describing: responseString))")

self.parseJson(data!)

}

task.resume()

}

func parseJson(\_ data:Data){

DispatchQueue.main.async{

let string = String(data: data,encoding: String.Encoding.utf8)

if string == "account not exist"{

self.ok\_alert("帳號錯誤")

}

else if string == "not registered yet"{

self.ok\_alert("此帳號尚未註冊")

}

else if string == "last\_4\_id wrong"{

self.ok\_alert("後四碼錯誤")

}else{

self.ok\_alert\_showPw("您的密碼為:",string!)

}

}

}

override func touchesEnded(\_ touch: Set<UITouch>,with event: UIEvent?){

last\_4\_id.resignFirstResponder()

account.resignFirstResponder()

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

}

override func viewDidLoad() {

super.viewDidLoad()

mod\_button()

circular\_button(submit)

last\_4\_id.keyboardType = .numberPad

}

}

db\_struct.swift

import Foundation

struct db\_struct{

var account = ""

//var position = ""

var name = ""

var password = ""

var status = ""

var reason = ""

var last\_4\_id = ""

var submit\_time = ""

}

tableViewCell.swift

import UIKit

class tableViewCell: UITableViewCell{

var actionBlock: (()->Void)? = nil

@IBOutlet weak var id\_label: UILabel!

//@IBOutlet weak var position\_label: UILabel!

@IBOutlet weak var name\_label: UILabel!

@IBOutlet weak var status\_label: UILabel!

@IBOutlet weak var reason\_button: UIButton!

@IBOutlet weak var submit\_date: UILabel!

@IBOutlet weak var submit\_time: UILabel!

@IBAction func didTapButton(\_ sender: Any) {

actionBlock?()

}

}

五.資料庫路徑與後端程式碼

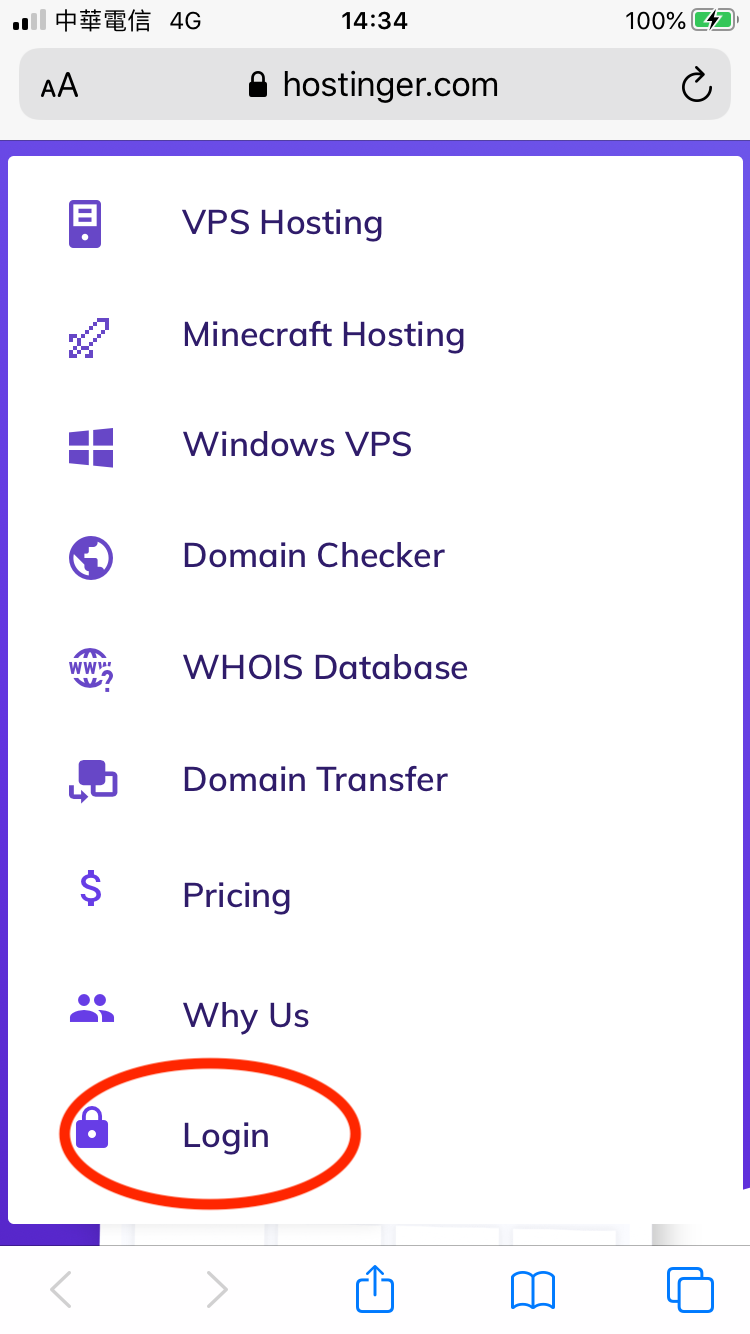
A.資料庫路徑：

本系統的資料庫架設於hostinger.com此網站，以下使用手機版圖示說明資料庫位置，以及本系統php檔案的存放位置，除非有必要，例如要轉移資料庫(建議複製php檔案，並更改swift的url連結就好)，盡量不要更改，以免app程式找不到php檔案正確的回傳值而出錯。

1.進入hostinger.com, 點右上角選單



2.點選底下的Login登入



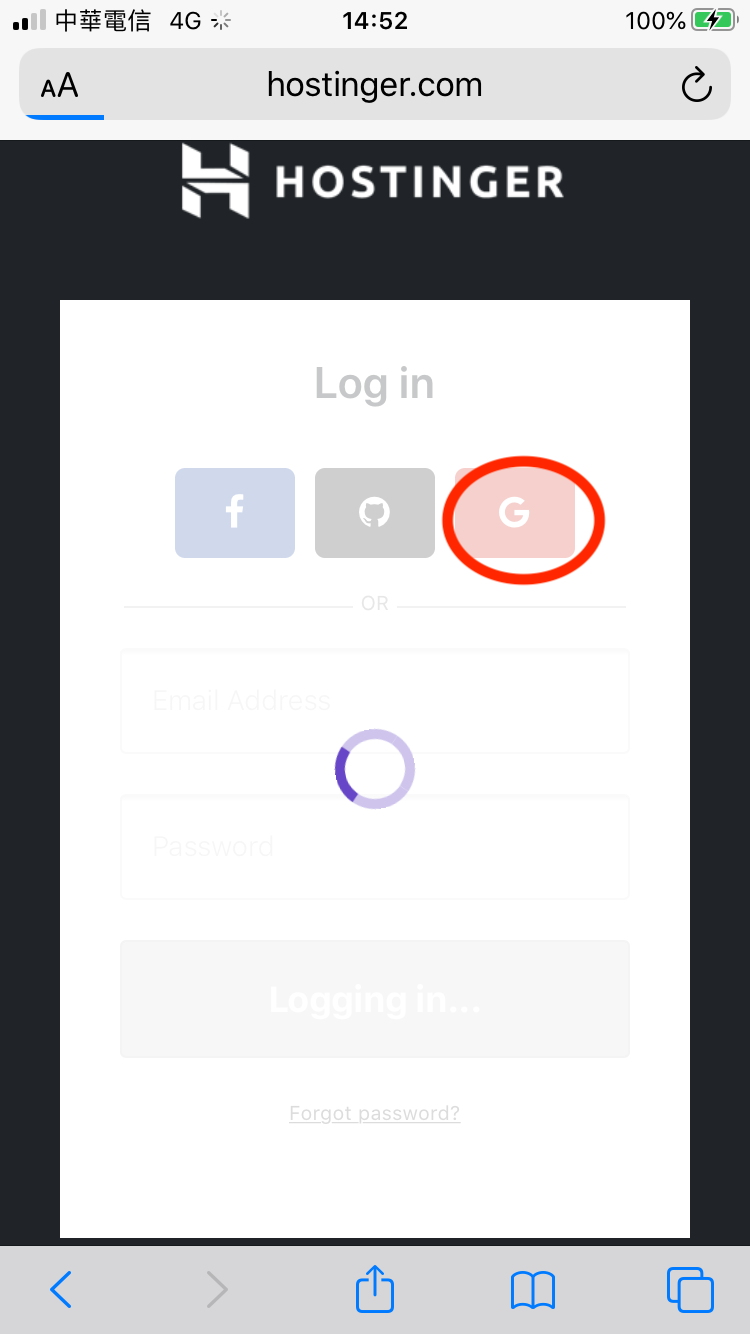
3.選擇以google登入

本人/本開發者的帳號為danny102575@gmail.com

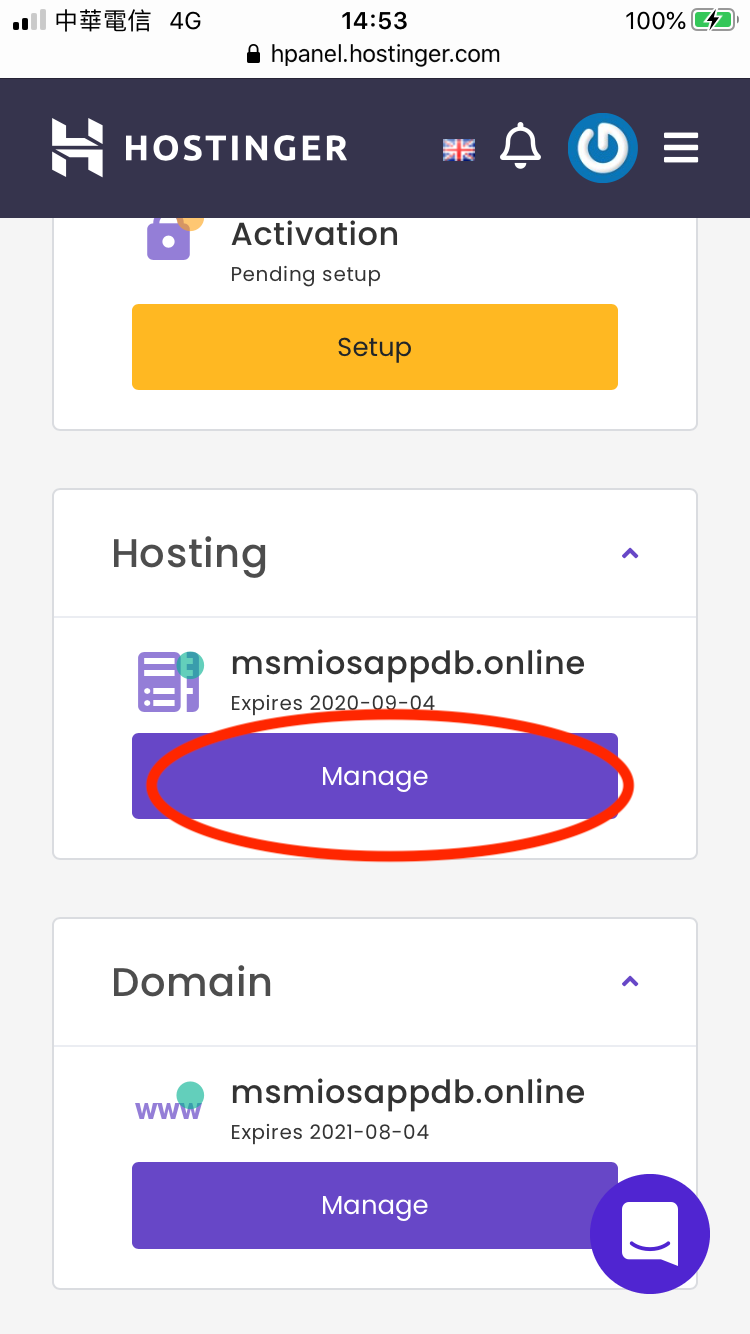
密碼為Qazwsxqwe!

\*此為本人徐梓修的google帳號，

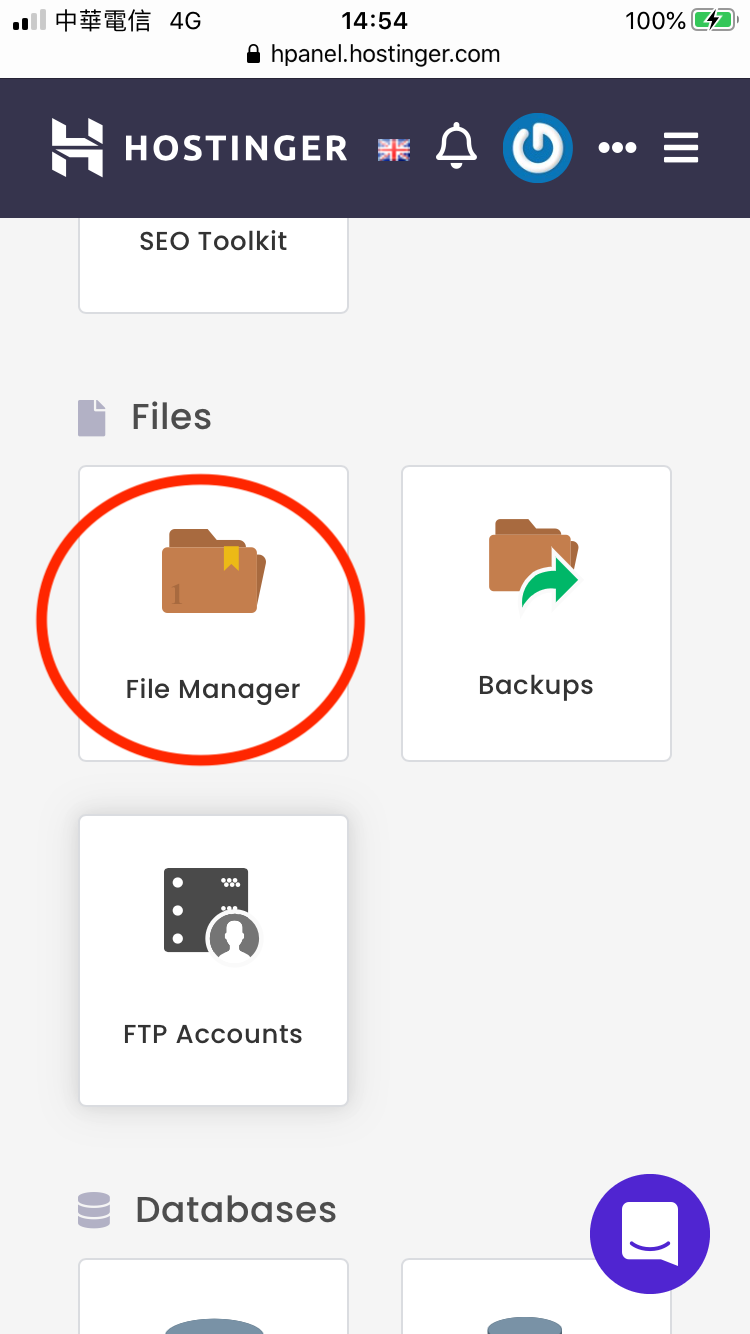
切勿使用於公務外事務，並注意個人隱私權\*



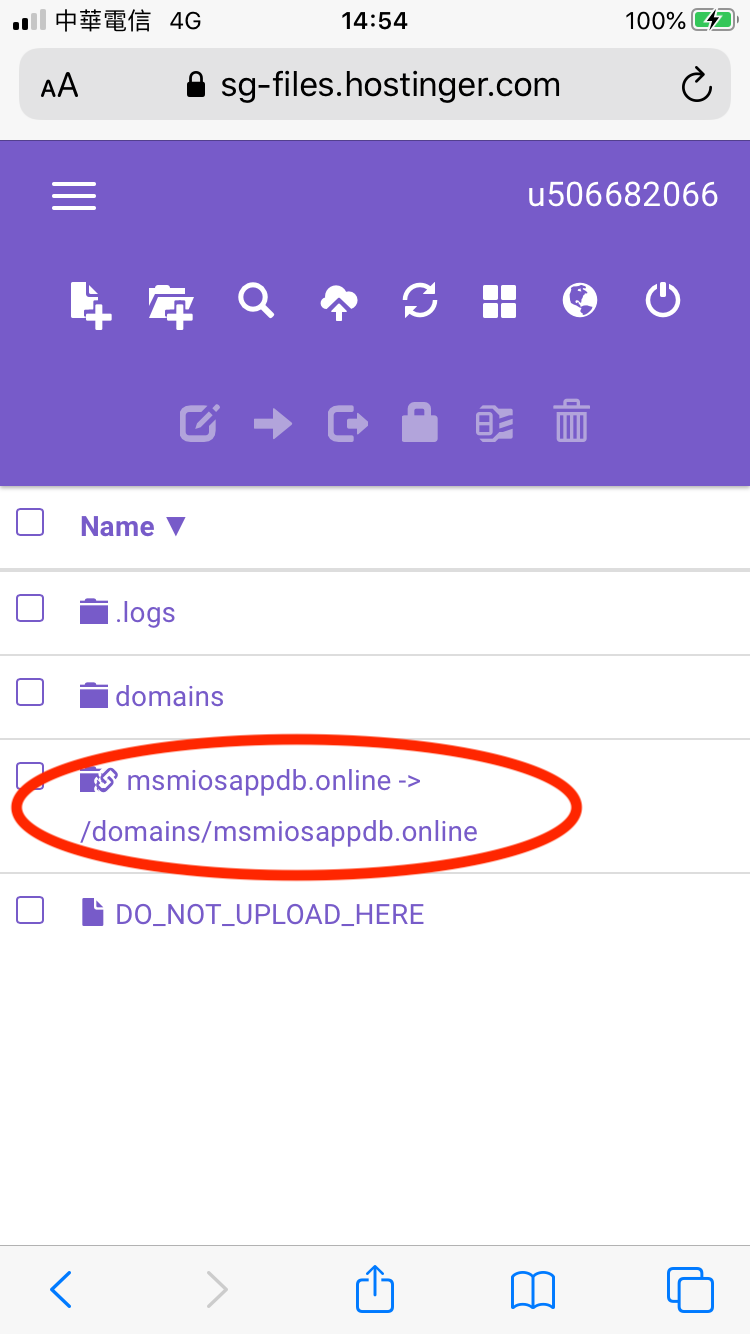
4.登入後，點選hosting欄位的管理(不要錯點到domain)



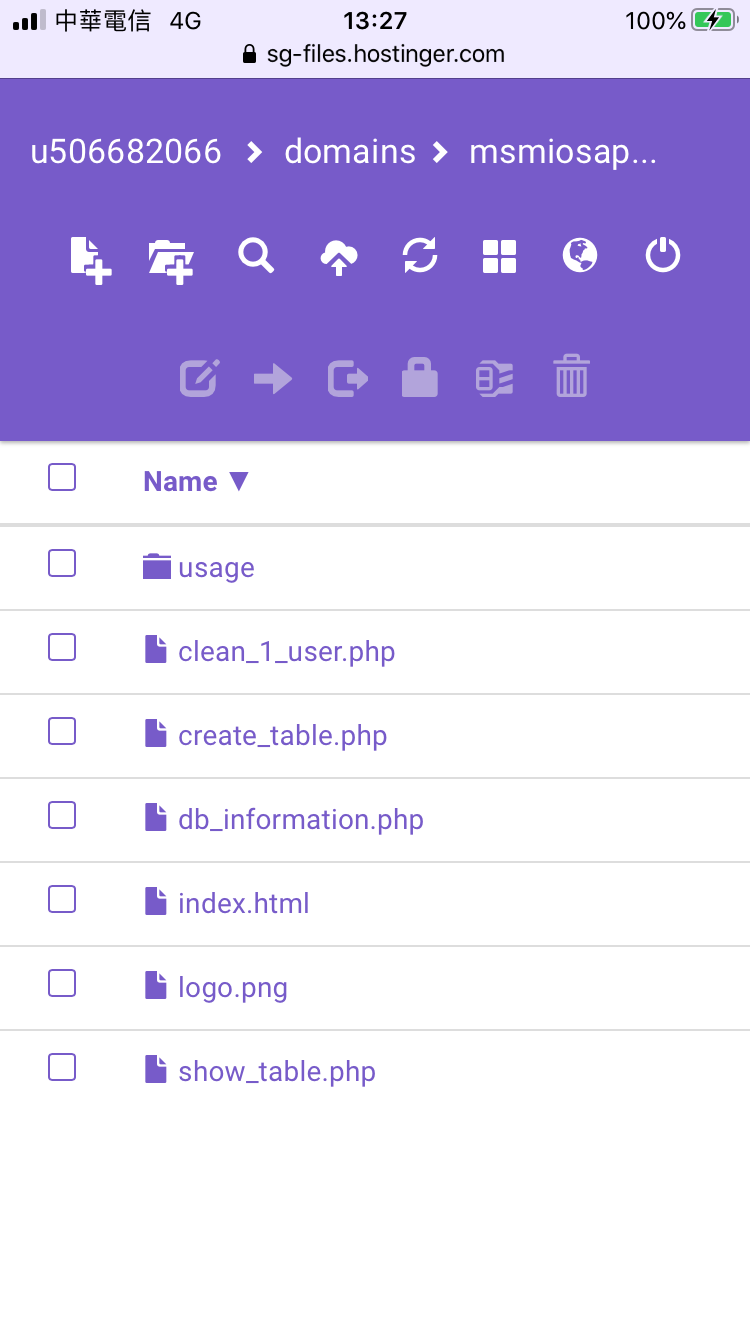
5.往下滑找到File Manager



6.點選此目錄



7.資料庫的php檔案都存放於此



B.網頁端程式碼

index.html

//首頁

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<style>

#container,h1{

display: table;

margin: 0 auto;

}

#background{

background-color: rgb(106,196,162);

}

hr{

border: 1px solid rgb(0,121,115);

}

</style>

</head>

<body>

<div id="background">

<h1>MSM人員到公系統</h1><hr>

<div id="container">

<p><a href="show\_table.php">顯示資料表</a></p>

<p><a href="clean\_1\_user.php">清除單一使用者資料(保留帳號)</a></p>

<p><a href="create\_table.php">清除所有使用者資料(保留帳號)</a></p>

<img src="logo.png" alt="marine logo">

</div>

</div>

</body>

</html>

show\_table.php

//在網頁上檢視資料表

<form action="usage/show\_table\_usage.php" method="post">

<div class="container">

<h2>Show Table</h2>

<label for="admin"><b>Admin</b></label>

<input type="text" placeholder="Enter Admin" name="admin" required><br>

<label for="admin\_pw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="admin\_pw" required><br>

<button type="submit">Login</button>

</div>

</form>

create\_table.php

//在網頁上將整張資料表marines清空，會保留帳號

<form action="usage/create\_table\_usage.php" method="post">

<div class="container">

<h2>Create Table</h2>

<label for="admin"><b>Admin</b></label>

<input type="text" placeholder="Enter Admin" name="admin" required><br>

<label for="admin\_pw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="admin\_pw" required><br>

<button type="submit">Login</button>

</div>

</form>

clean\_1\_user.php

//在網頁上清除單一使用者的資料，會保留帳號

<form action="usage/clean\_1\_user\_usage.php" method="post">

<div class="container">

<h2>Clean 1 User's Data</h2>

<label for="admin"><b>Admin</b></label>

<input type="text" placeholder="Enter Admin" name="admin" required><br>

<label for="admin\_pw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="admin\_pw" required><br>

<label for="account"><b>Account</b></label>

<input type="text" placeholder="Enter Account" name="account" required><br>

<button type="submit">Login</button>

</div>

</form>

db\_information.php

//儲存資料庫登入之帳號密碼

<?php

$requustType = $\_SERVER['REQUEST\_METHOD'];

if ($requustType == 'GET'){

die("Prohibit From Direct Access");

}

else if ($requustType == 'POST'){

define('db\_servername' , 'localhost');

define('db\_username' , 'u506682066\_msm\_user');

define('db\_password' , 'katcud-Cydwa4-dohzab');

define('db\_name' , 'u506682066\_msm');

define('db\_admin','danny102575@gmail.com');

define('db\_admin\_pw','Qazwsxqwe!');

}

?>

usage/show\_table\_usage.php

//為 show\_table.php所使用

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$admin = $\_POST['admin'];

$admin\_pw = $\_POST['admin\_pw'];

if ($admin == db\_admin && $admin\_pw == db\_admin\_pw){

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

echo "Connect to database successfully"."<br>";

$sql = "SELECT \* FROM marines";

$result = $conn->query($sql);

$display\_pw = False;

if ($result->num\_rows > 0) {

// output data of each row

while($row = $result->fetch\_assoc()) {

if ($display\_pw == False){

echo $row["account"].'&nbsp'.

//$row["position"].'&nbsp'.

$row["name"].'&nbsp'.

$row["status"].'&nbsp'.

$row["reason"].'&nbsp'.

$row["submit\_time"].'<br>';

}else{

echo $row["account"].'&nbsp'.

//$row["position"].'&nbsp'.

$row["name"].'&nbsp'.

$row["password"].'&nbsp'.

$row["status"].'&nbsp'.

$row["reason"].'&nbsp'.

$row["last\_4\_id"].'&nbsp'.

$row["submit\_time"].'<br>';

}

}

}else{

echo "0 results";

}

$conn->close();

}else{

echo "Login Failed";

}

?>

usage/create\_table\_usage.php

//為 create\_table.php所使用

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$admin = $\_POST['admin'];

$admin\_pw = $\_POST['admin\_pw'];

if ($admin == db\_admin && $admin\_pw == db\_admin\_pw){

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

echo "Connect to database successfully"."<br>";

$sql = "DROP TABLE marines";

if($conn->query($sql) === TRUE){

echo "Drop table successfully"."<br>";

}else{

echo "Drop table failed"."<br>";

}

$sql = "CREATE TABLE marines (

account VARCHAR(30) PRIMARY KEY DEFAULT '',

name VARCHAR(30) NOT NULL DEFAULT '',

password VARCHAR(30) NOT NULL DEFAULT '',

status VARCHAR(10) NOT NULL DEFAULT '',

reason TEXT DEFAULT '',

last\_4\_id VARCHAR(10) NOT NULL DEFAULT '',

submit\_time VARCHAR(30) NOT NULL DEFAULT ''

)";

if($conn->query($sql) === TRUE){

echo "Table marines created successfully"."<br>";

}else{

echo "Error creating table: "."<br>" . $conn->error;

}

$sql = "INSERT INTO marines (account)

VALUES

('AA001'), #校長室

('AA002'),

('AA003'),

('AA004'),

('AA005'), #士官督導長室

('AD001'),

('BA001'), #學員生事務處

('BB001'),

('BB002'),

('BB003'),

('CA001'), #總務處

('CB001'),

('CB002'),

('CB003'),

('CB004'),

('CB005'),

('CB006'),

('CB007'),

('CB008'),

('CB009'),

('CB010'),

('CB011'),

('CB012'),

('CB013'),

('DA001'), #教務處

('DB001'),

('DB002'),

('DB003'),

('DB004'),

('DB005'),

('DB006'),

('DB007'),

('DB008'),

('DB009'),

('DB010'),

('DB011'),

('DB012'),

('DB013'),

('DB014'),

('DB015'),

('DB016'),

('EA001'), #資訊模擬中心

('EB001'),

('EB002'),

('EB003'),

('EB004'),

('EB005'),

('EB006'),

('FA001'), #主計室

('FB001'),

('FB002'),

('FB003'),

('FB004'),

('GA001'), #醫務所

('GB001'),

('GB002'),

('GB003'),

('GB004'),

('GB005'),

('GB006'),

('GB007'),

('GB008'),

('GB009'),

('GB010'),

('HA001'), #總教官室

('IA001'), #兩棲戰術組

('IB001'),

('IB002'),

('IB003'),

('IB004'),

('JA001'), #小部隊組

('JB001'),

('JB002'),

('JB003'),

('JB004'),

('JB005'),

('JB006'),

('JB007'),

('JB008'),

('KA001'), #通識教學組

('KB001'),

('KB002'),

('KB003'),

('KB004'),

('KB005'),

('LA001'), #學員生總隊部

('LC001'),

('LD001'),

('LC101'), #學員生一大隊

('LC102'),

('LC103'),

('LC111'), #第一中隊

('LC112'),

('LC113'),

('LC114'), #第二中隊

('LC115'),

('LC116'),

('LC117'),

('LC118'),

('LC119'),

('LC120'),

('LC121'),

('LC122'), #第三中隊

('LC123'),

('LC124'),

('LC125'),

('LC126'),

('LC127'),

('LC128'),

('LC129'),

('LC130'), #第四中隊

('LC131'),

('LC132'),

('LC133'),

('LC134'),

('LC135'),

('LC136'),

('LC137'),

('LC201'), #學員生二大隊

('LC202'),

('LC203'),

('LC211'), #第五中隊

('LC212'),

('LC213'),

('LC214'),

('LC215'),

('LC216'),

('LC217'),

('LC218'),

('LC219'),

('LC220'),

('LC221'), #第六中隊

('LC222'),

('LC223'),

('LC224'),

('LC225'),

('LC226'),

('LC227'),

('LC228'),

('LC229'),

('LC230'),

('LC231'), #第七中隊

('LC232'),

('LC233'),

('LC234'),

('LC235'),

('LC236'),

('LC237'),

('LC238'),

('LC239'),

('LC240'),

('LC241'), #第八中隊

('LC242'),

('LC243'),

('LC244'),

('LC245'),

('LC246'),

('LC247'),

('LC248'),

('LC249'),

('LC250'),

('MB001'), #莒拳隊

('MB002'),

('MB003'),

('MB004'),

('MB005'),

('MB006'),

('MB007'),

('MB008'),

('MB009'),

('MB010'),

('MB011'),

('MB012'),

('NB001'), #勤務隊

('NB002'),

('NB003'),

('NB004'), #支援班

('NB005'),

('NB006'),

('NB007'),

('NB008'),

('NB009'),

('NB010'),

('NB011'),

('NB012'), #警衛班

('NB013'),

('NB014'),

('NB015'),

('NB016'),

('NB017'),

('NB018'),

('NB019'),

('NB020'),

('NB021'),

('NB022'),

('NB023'),

('NB024'), #修護運輸班

('NB025'),

('NB026'),

('NB027'),

('NB028'),

('NB029'),

('NB030'),

('NB031'),

('NB032'),

('NB033')

";

if ($conn->query($sql) === TRUE) {

echo "New records created successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

}else{

echo "Login Failed";

}

?>

usage/clean\_1\_user\_usage.php

//為 clean\_1\_user.php所使用

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$admin = $\_POST['admin'];

$admin\_pw = $\_POST['admin\_pw'];

$account = $\_POST['account'];

if ($admin == db\_admin && $admin\_pw == db\_admin\_pw)

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

echo "Connect to database successfully"."<br>";

$sql ="SELECT \* FROM marines WHERE account = '$account'";

$result = $conn->query($sql);

if($result->num\_rows > 0){

$sql = "UPDATE marines SET name = '', password = '', status = '',reason = '', last\_4\_id = '',submit\_time = '' WHERE account = '$account'";

if ($conn->query($sql) === TRUE){

echo "1 record clean successfully";

}else{

echo "Error: " . $sql . "<br>" . $conn->error;

}

}else{

echo "account not exist";

}

$conn->close();

?>

C.手機端呼叫之程式碼

usage/login.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$pw = $\_POST['pw\_'];

$login\_mode = $\_POST['login\_mode\_'];

if ($login\_mode == "account"){

$sql ="SELECT \* FROM marines WHERE account = '$account'";

$result = $conn->query($sql);

if($result->num\_rows > 0){

$sql\_ ="SELECT password FROM marines WHERE account = '$account'";

$result\_ = $conn->query($sql\_)->fetch\_object()->password;

if($result\_ == ""){

echo "new account";

}else{

echo "already registered";

}

}else{

echo "account not exist";

}

}

if ($login\_mode == "account\_pw"){

$sql ="SELECT \* FROM marines WHERE account = '$account'";

$result = $conn->query($sql);

if($result->num\_rows > 0){

$sql\_ ="SELECT password FROM marines WHERE account = '$account'";

$result\_ = $conn->query($sql\_)->fetch\_object()->password;

if($result\_ == ""){

echo "new account 2";

}else{

$sql ="SELECT \* FROM marines WHERE account = '$account' AND BINARY password = '$pw'";

$result = $conn->query($sql);

if($result->num\_rows > 0){

echo "login successed";

}else{

echo "password not exist";

}

}

}else{

echo "account not exist 2";

}

}

$conn->close();

?>

usage/register.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$name = $\_POST['name\_'];

$pw = $\_POST['pw\_'];

$last\_4\_id = $\_POST['last\_4\_id\_'];

$sql = "UPDATE marines SET name = '$name', password = '$pw', last\_4\_id = '$last\_4\_id' WHERE account = '$account'";

if ($conn->query($sql) === TRUE){

echo "New records created successfully";

}else{

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

usage/forgotPw.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$last\_4\_id = $\_POST['last\_4\_id\_'];

$sql ="SELECT \* FROM marines WHERE account = '$account'";

$result = $conn->query($sql);

if($result->num\_rows > 0){

$sql ="SELECT last\_4\_id FROM marines WHERE account = '$account'";

$result = $conn->query($sql)->fetch\_object()->last\_4\_id;

if ($result == ""){

echo "not registered yet";

}else{

$sql ="SELECT password FROM marines WHERE account = '$account' AND last\_4\_id = '$last\_4\_id'";

$result = $conn->query($sql)->fetch\_object()->password;

if ($result == ""){

echo "last\_4\_id wrong";

}else{

echo $result;

}

}

}else{

echo "account not exist";

}

$conn->close();

?>

usage/profile.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$pw = $\_POST['pw\_'];

$sql = "SELECT name FROM marines WHERE account = '$account' AND BINARY password = '$pw'";

$result = $conn->query($sql)->fetch\_object();

$resultArray = array();

array\_push($resultArray,$result);

echo json\_encode($resultArray);

$conn->close();

?>

usage/changePw.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$new\_pw = $\_POST['new\_pw\_'];

$old\_pw = $\_POST['old\_pw\_'];

$sql = "UPDATE marines SET password = '$new\_pw' WHERE account = '$account' AND BINARY password = '$old\_pw'";

if ($conn->query($sql) === TRUE){

echo "Change password successfully";

}else{

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

usage/todaySubmit.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$pw = $\_POST['pw\_'];

$status = $\_POST['status\_'];

$reason = $\_POST['reason\_'];

$submit\_time = $\_POST['submit\_time\_'];

$sql = "UPDATE marines SET status = '$status', reason = '$reason', submit\_time = '$submit\_time' WHERE account = '$account' AND BINARY password = '$pw'";

if ($conn->query($sql) === TRUE){

echo "today submit successfully";

}else{

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

usage/service.php

<?php

require\_once \_\_DIR\_\_ . '/../db\_information.php';

$conn = new mysqli(db\_servername, db\_username, db\_password, db\_name);

if ($conn->connect\_error){

die("Connection failed: " . $conn->connect\_error);

}

$account = $\_POST['account\_'];

$authority = $\_POST['authority\_'];

$location = $\_POST['location\_'];

$alter = $\_POST['alter\_'];

if ($authority == 'A' || $authority == 'a'){

$sql = "SELECT \* FROM marines";

}

if ($authority == 'B' || $authority == 'b'){

$sql = "SELECT \* FROM marines WHERE account LIKE '$location%'";

}

if ($authority == 'C' || $authority == 'c'){

if ($alter == '0'){

$sql = "SELECT \* FROM marines WHERE account LIKE '$location%'";

}

if ($alter == '1'){

$sql = "SELECT \* FROM marines WHERE account LIKE CONCAT('$location','$authority','1','%')";

}

if ($alter == '2'){

$sql = "SELECT \* FROM marines WHERE account LIKE CONCAT('$location','$authority','2','%')";

}

}

if ($authority == 'D' || $authority == 'd'){

$sql = "SELECT \* FROM marines WHERE account = '$account'";

}

//$sql = "SELECT \* FROM marines WHERE account LIKE 'A%'";

if($result = $conn->query($sql)){

$resultArray = array();

$tempArray = array();

while($row = $result->fetch\_object()){

$tempArray = $row;

array\_push($resultArray,$tempArray);

}

echo json\_encode($resultArray);

}

$conn->close();

?>

六.下載與使用方式(附專案檔連結)

本人已將此程式(msm人員到公系統)上傳至自己的apple developer帳號，透過testflight可以讓我們在不上架至apple store的情況下，使用app的beta測試版，以下將提供本系統的beta版下載連結，以及xcode的專案檔(附後端php程式)，若有問題如無法下載再聯絡本人

app下載連結：<https://testflight.apple.com/join/4C1mL4FY>

( 請依連結指示先下載testflight，再下載MSM，目前app版本為1.0(5) , 需要ios 13.5以上)



專案檔下載連結：https://drive.google.com/file/d/1t7KMQSQyCs4tW0dVqxWOKTiiLk5viirl/view?usp=drivesdk

(請用mac系統開啟，需安裝xcode)

php檔下載連結：https://drive.google.com/file/d/1h25-f1e5PCF9ZBy0sqGU2A\_f5687NL31/view?usp=drivesdk

七.權限說明與職位帳號對照表

第一碼為處所區域：

A為校長室與士官督導室

B為學員生事務處

C為總務處

D為教務處

E為資訊模擬中心

F為主計處

G為醫務所

H為總教官室

I為兩棲戰術組

J為小部隊組

K為通識教學組

L為學員生隊

M為莒拳隊

N為勤務隊

第二碼為權限：

A為看全校

B為看所在處所

D為只看自己

C為學員生隊：{

LC001看一加二大隊

LC1xx為一大隊，看一大隊

LC2xx為二大隊，看二大隊

}

('AA001' , '少將校長'), #校長室

('AA002' , '上校教育長'),

('AA003' , '上校政戰主任'),

('AA004' , '少校監察官'),

('AA005' , '一等長士督長'), #士官督導長室

('AD001' , '上士行政士'),

('BA001' , '中校處長'), #學員生事務處

('BB001' , '少校政戰官'),

('BB002' , '上尉心輔官'),

('BB003' , '上尉保防官'),

('CA001' , '中校處長'), #總務處

('CB001' , '少校副處長'),

('CB002' , '中尉補給官'),

('CB003' , '中尉工程官'),

('CB004' , '三等長文書士'),

('CB005' , '一等長後勤士'),

('CB006' , '三等長兵修士'),

('CB007' , '上士人事士'),

('CB008' , '中士人事士'),

('CB009' , '中士文書士'),

('CB010' , '下士人事士'),

('CB011' , '上兵人事士'),

('CB012' , '雇二人事士'),

('CB013' , '雇二檔管員'),

('DA001' , '中校處長'), #教務處

('DB001' , '少校副處長'),

('DB002' , '少校動官員'),

('DB003' , '少校作戰官'),

('DB004' , '中尉動員官'),

('DB005' , '中尉教育官'),

('DB006' , '三等長教育士'),

('DB007' , '一等長教育士'),

('DB008' , '一等長體育士'),

('DB009' , '上士情報士'),

('DB010' , '上士教育士'),

('DB011' , '中士教育士'),

('DB012' , '中士教育士'),

('DB013' , '中士體育士'),

('DB014' , '中士體育士'),

('DB015' , '下士體育士'),

('DB016' , '下士體育士'),

('EA001' , '中校主任'), #資訊模擬中心

('EB001' , '少校資參官'),

('EB002' , '上尉教官'),

('EB003' , '中尉系統官'),

('EB004' , '上士通信士'),

('EB005' , '中士助教'),

('EB006' , '下士助教'),

('FA001' , '中校主任'), #主計室

('FB001' , '少校預財官'),

('FB002' , '上尉預財官'),

('FB003' , '上士預財官'),

('FB004' , '下士預財官'),

('GA001' , '少校主任'), #醫務所

('GB001' , '上尉醫官'),

('GB002' , '中尉醫官'),

('GB003' , '中尉醫官'),

('GB004' , '二等長醫務士'),

('GB005' , '上士醫務士'),

('GB006' , '中士醫務士'),

('GB007' , '中士醫務士'),

('GB008' , '下士醫務兵'),

('GB009' , '上兵醫務兵'),

('GB010' , '上兵醫務兵'),

('HA001' , '上校總教官'), #總教官室

('IA001' , '中校組長'), #兩棲戰術組

('IB001' , '中校戰術教官'),

('IB002' , '中校情報教官'),

('IB003' , '中校砲兵教官'),

('IB004' , '上尉戰術教官'),

('JA001' , '上校組長'), #小部隊組

('JB001' , '中校戰術教官'),

('JB002' , '少校戰術教官'),

('JB003' , '二等長教官'),

('JB004' , '一等長教官'),

('JB005' , '一等長教官'),

('JB006' , '三等長教官'),

('JB007' , '一等長教官'),

('JB008' , '二等長教官'),

('KA001' , '中校組長'), #通識教學組

('KB001' , '中校政治教官'),

('KB002' , '少校體育教官'),

('KB003' , '少校政治教官'),

('KB004' , '少校後勤教官'),

('KB005' , '中尉教官'),

('LA001' , '上校總隊長'), #學員生總隊部

('LC001' , '中校政戰處長'),

('LD001' , '中士行政士'),

('LC101' , '中校大隊長'), #學員生一大隊

('LC102' , '少校輔導長'),

('LC103' , '中士行政士'),

('LC111' , '少校中隊長'), #第一中隊

('LC112' , '中尉輔導長'),

('LC113' , '下士行政士'),

('LC114' , '一等長中隊長'), #第二中隊

('LC115' , '三等長輔導長'),

('LC116' , '上士區隊長'),

('LC117' , '上士區隊長'),

('LC118' , '下士區隊長'),

('LC119' , '下士區隊長'),

('LC120' , '下士區隊長'),

('LC121' , '下士行政士'),

('LC122' , '一等長中隊長'), #第三中隊

('LC123' , '二等長輔導長'),

('LC124' , '上士區隊長'),

('LC125' , '上士區隊長'),

('LC126' , '下士區隊長'),

('LC127' , '下士區隊長'),

('LC128' , '下士區隊長'),

('LC129' , '下士行政士'),

('LC130' , '一等長中隊長'), #第四中隊

('LC131' , '二等長輔導長'),

('LC132' , '上士區隊長'),

('LC133' , '上士區隊長'),

('LC134' , '下士區隊長'),

('LC135' , '下士區隊長'),

('LC136' , '下士區隊長'),

('LC137' , '下士行政士'),

('LC201' , '中校大隊長'), #學員生二大隊

('LC202' , '少校輔導長'),

('LC203' , '中士行政士'),

('LC211' , '上尉中隊長'), #第五中隊

('LC212' , '少尉輔導長'),

('LC213' , '上士助教領導'),

('LC214' , '中士助教'),

('LC215' , '下士助教'),

('LC216' , '下士助教'),

('LC217' , '上兵助教'),

('LC218' , '上兵助教'),

('LC219' , '下士助教'),

('LC220' , '下士行政士'),

('LC221' , '上尉中隊長'), #第六中隊

('LC222' , '少尉輔導長'),

('LC223' , '上士助教'),

('LC224' , '中士助教'),

('LC225' , '下士助教'),

('LC226' , '下士助教'),

('LC227' , '下士助教'),

('LC228' , '下士助教'),

('LC229' , '下士助教'),

('LC230' , '下士行政士'),

('LC231' , '上尉中隊長'), #第七中隊

('LC232' , '少尉輔導長'),

('LC233' , '上士助教'),

('LC234' , '中士助教'),

('LC235' , '下士助教'),

('LC236' , '下士助教'),

('LC237' , '下士助教'),

('LC238' , '上兵助教'),

('LC239' , '上兵助教'),

('LC240' , '下士行政士'),

('LC241' , '上尉隊長'), #第八中隊

('LC242' , '少尉輔導長'),

('LC243' , '上士助教'),

('LC244' , '中士助教'),

('LC245' , '下士助教'),

('LC246' , '上兵助教'),

('LC247' , '上兵助教'),

('LC248' , '下士隊長'),

('LC249' , '下士隊長'),

('LC250' , '上兵行政士'),

('MB001' , '中校隊長'), #莒拳隊

('MB002' , '少校輔導長'),

('MB003' , '三等長助教'),

('MB004' , '一等長助教'),

('MB005' , '一等長助教'),

('MB006' , '三等長助教'),

('MB007' , '上士助教'),

('MB008' , '上士助教'),

('MB009' , '中士助教'),

('MB010' , '中士助教'),

('MB011' , '中士助教'),

('MB012' , '中士行政士'),

('NB001' , '上尉隊長'), #勤務隊

('NB002' , '三等長副隊長'),

('NB003' , '下士行政士'),

('NB004' , '中士班長'), #支援班

('NB005' , '中士補給士'),

('NB006' , '中士補給士'),

('NB007' , '下士補給士'),

('NB008' , '下士營產士'),

('NB009' , '一兵補給兵'),

('NB010' , '上兵補給兵'),

('NB011' , '上兵補給兵'),

('NB012' , '上士班長'), #警衛班

('NB013' , '中士副班長'),

('NB014' , '下士作戰士'),

('NB015' , '下士作戰士'),

('NB016' , '上兵步槍兵'),

('NB017' , '上兵步槍兵'),

('NB018' , '二兵步槍兵'),

('NB019' , '一兵步槍兵'),

('NB020' , '二兵步槍兵'),

('NB021' , '上兵步槍兵'),

('NB022' , '一兵機槍兵'),

('NB023' , '一兵機槍兵'),

('NB024' , '上士班長'), #修護運輸班

('NB025' , '中士副班長'),

('NB026' , '下士機工士'),

('NB027' , '下士駕駛士'),

('NB028' , '上兵機工士'),

('NB029' , '上兵機工士'),

('NB030' , '上兵駕駛兵'),

('NB031' , '一兵駕駛兵'),

('NB032' , '一兵駕駛兵'),

('NB033' , '二兵駕駛兵')

八.本人/本開發者聯絡方式

姓名：徐梓修

行動電話：0963516090

Email：[danny102575@gmail.com](mailto:danny102575@gmail.com)

Line: hugh102575