```swift
//
//  ViewController.swift
//  AttendanceApplication
//


import UIKit
import Alamofire

class LoginViewController: UIViewController {

    // MARK: IBOutlets

    @IBOutlet weak var _username: UITextField!
    @IBOutlet weak var _password: UITextField!
    @IBOutlet weak var _loginButton: UIButton!
    @IBOutlet weak var errorLabel: UILabel!

    // MARK: IBOutlets (Error Labels)

    @IBOutlet weak var wrongPasswordLabel: UILabel!
    @IBOutlet weak var connectionErrorLabel: UILabel!

    // MARK: UIViewController Methods

    override func viewWillAppear(_ animated: Bool) {
        resetLabels()
    }

    // MARK: IBAction methods

    @IBAction func loginButton(_ sender: Any) {
        // Firstly, ensure that values have been entered.
        let username = _username.text
        let password = _password.text
        if username == "" || password == "" { return }

        // Then pass the strings to perform the login
        performLogin(username: username!, password: password!)
    }

    // MARK: Internal Methods

    internal func performLogin(username:String, password:String){
        let parameters: Parameters = [
            "type": "universal.login",
            "args": [
                "username": username,
                "password": password
            ]
        ]

        Alamofire.request(HTTPHelper.url, method: .post, parameters: parameters,
         encoding: JSONEncoding.default).responseJSON {
```

```
        response in

        switch response.result {
        case .failure( _):
            // In case of total failure to send request, give a connection error
            self.setConnectionError()
            return

        case .success(let data):
            // First make sure a dictionary is recieved: Data validation
            guard let json = data as? [String : AnyObject] else {
                // Print statement for debugging purposes, not seen by users.
                 Users are given a connection error.
                self.setConnectionError()
                print("Failed to get expected dictionary from webserver.")
                return
            }

            // Make sure this is the actual key/value types that are expected
            guard let success = json["successful"] as? Int, let reason =
             json["reason"] as? String, let classification =
             json["classification"] as? String else {
                // Print statement for debugging purposes, not seen by users.
                 Users are given a connection error.
                self.setConnectionError()
                print("Failed to get data from webserver")
                return
            }

            if success == 1 {
                // If it succesfully connected, check if this is an admin or a
                 student and send them to their respective view using a segue.
                if classification == "admin" {
                    self.performSegue(withIdentifier: "segueToAdminView",
                     sender: self)
                } else {
                    self.performSegue(withIdentifier: "segueToStudentView",
                     sender: self)
                }
            } else if reason == "inc_login" {
                // If it was a login error
                self.setIncorrectLoginLabels()
            } else {
                // If it was a database connection errorx
                self.setConnectionError()
            }
        }
    }
}

internal func setIncorrectLoginLabels() {
    // Function for displaying the incorrect login error
    wrongPasswordLabel.isHidden = false
    _password.layer.borderColor = UIColor.red.cgColor
```

```
        _password.layer.borderWidth = 1
    }

    internal func setConnectionError() {
        // Function for displaying the connection error
        connectionErrorLabel.isHidden = false
    }

    internal func resetLabels() {
        // Resets all errors and successes, called when view is opened
        _username.text = ""
        _password.text = ""
        wrongPasswordLabel.isHidden = true
        _password.layer.borderWidth = 0
        connectionErrorLabel.isHidden = true
    }
}
```