Login View controller

1. import UIKit
3. class LoginViewController: UIViewController {
5. @IBOutlet weak var profileImage: UIImageView!
7. @IBOutlet weak var passwordText: UITextField!
9. @IBOutlet weak var emailText: UITextField!
11. override func viewDidLoad() {
12. super.viewDidLoad()
13. }
15. var isImageSelected = false;
17. @IBAction func chooseImage(\_ sender: UIButton) {
18. ImagePickerManager().pickImage(self){ image in
19. self.profileImage.image = image
20. self.isImageSelected = true
21. }
22. }

25. @IBAction func submitBtn(\_ sender: UIButton) {
26. var alertMessage = "";
27. let emailvalidationMessage = getEmailValidationMessage(email: emailText.text!);
28. let passwordValidationMessage = getPasswordValidationMessage(password: passwordText.text!)
30. if (emailvalidationMessage.count > 0) {
31. alertMessage = emailvalidationMessage;
32. } else if (passwordValidationMessage.count > 0) {
33. alertMessage = passwordValidationMessage;
34. } else if (!isImageSelected) {
35. alertMessage = "Please upload profile image";
36. }

39. if(alertMessage.count > 1) {
40. let alert = UIAlertController(title: "Alert", message: alertMessage, preferredStyle: .alert)
41. alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: nil))
42. self.present(alert, animated: true, completion: nil)
43. } else {
44. let storyBoard : UIStoryboard = UIStoryboard(name: "Main", bundle:nil)
45. let TableViewController = storyBoard.instantiateViewController(withIdentifier: "tableViewController") as! TableViewController
46. TableViewController.modalPresentationStyle = .fullScreen
47. self.present(TableViewController, animated:true, completion:nil)
48. }
49. }
50. }
52. func getEmailValidationMessage(email: String) -> String {
53. var invalidEmailMessage = "";
54. if (email.count == 0) {
55. return "Please enter Email Address";
56. }
57. let emailRegEx = "[A-Z0-9a-z.\_%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,64}"
59. let emailPred = NSPredicate(format:"SELF MATCHES %@", emailRegEx)
60. if(!emailPred.evaluate(with: email)) {
61. invalidEmailMessage = "Invalid Email Address";
62. }
63. return invalidEmailMessage;
64. }
66. func getPasswordValidationMessage(password: String) -> String {
67. var invalidPasswordMessage = "";
68. if (password.count == 0) {
69. return "Please enter password";
70. }
72. let passwordValidation = NSPredicate(format: "SELF MATCHES %@", "^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9])(?=.\*?[#?!@$%^&\*-]).{8,}$");
73. let upperCaseValidation = NSPredicate(format: "SELF MATCHES %@", ".\*[A-Z]+.\*");
74. let lowerCaseValidation = NSPredicate(format: "SELF MATCHES %@", ".\*[a-z]+.\*");
75. let specialCharacterValidation = NSPredicate(format: "SELF MATCHES %@", ".\*[!&^%$#@()/]+.\*");
76. let numericValidation = NSPredicate(format: "SELF MATCHES %@", ".\*[0-9]+.\*");
78. if (!passwordValidation.evaluate(with: password)) {
79. invalidPasswordMessage = "\n Password should contain \n"
80. if (password.count < 8) {
81. invalidPasswordMessage += "\n \u{2022} Minimum 8 characters";
82. }
83. if (!upperCaseValidation.evaluate(with: password)) {
84. invalidPasswordMessage += "\n \u{2022} Atleast one upper case";
85. }
86. if (!lowerCaseValidation.evaluate(with: password)) {
87. invalidPasswordMessage += "\n \u{2022} Atleast one lower case";
88. }
89. if (!specialCharacterValidation.evaluate(with: password)) {
90. invalidPasswordMessage += "\n \u{2022} Atleast one speical symbol";
91. }
92. if (!numericValidation.evaluate(with: password)) {
93. invalidPasswordMessage += "\n \u{2022} Atleast one numeric value";
94. }
95. }
96. return inv

 alidPasswordMessage;

}

Table View controller

1. import UIKit
3. class TableViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
5. @IBOutlet weak var tableView: UITableView!
7. var tableData: [[(number: Int, occupied: Bool)]] = [
8. [(1, true), (2, true)],
9. [(101, true), (102, false), (103, true)],
10. [(201, true), (202, true)]
11. ]
13. var updatedTableData: [[(number: Int, occupied: Bool)]] = []

16. override func viewDidLoad() {
17. super.viewDidLoad()
19. tableView.delegate = self
20. tableView.dataSource = self
21. tableView.register(UITableViewCell.self, forCellReuseIdentifier: "Cell")
22. tableView.register(SectionHeader.self, forHeaderFooterViewReuseIdentifier: "sectionHeader")
24. tableData.append(contentsOf: Array(repeating: [], count: 97))
26. if (updatedTableData.count > 0) {
27. tableData = updatedTableData
28. updatedTableData = []
29. self.tableView.reloadData()
30. }
31. }

34. @IBAction func addSection(\_ sender: UIBarButtonItem) {
35. let storyBoard : UIStoryboard = UIStoryboard(name: "Main", bundle:nil)
36. let AddRoomViewController = storyBoard.instantiateViewController(withIdentifier: "addRoomViewController") as! AddRoomViewController
37. AddRoomViewController.modalPresentationStyle = .fullScreen
38. AddRoomViewController.tableData = self.tableData
39. self.present(AddRoomViewController, animated:true, completion:nil)
40. }
42. func numberOfSections(in tableView: UITableView) -> Int {
43. return tableData.count
44. }
46. func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
47. return tableData[section].count
48. }
50. func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
51. let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)
52. let roomData = tableData[indexPath.section][indexPath.row];
53. cell.textLabel?.text = "Room \(roomData.number)"
55. let switchView = UISwitch(frame: .zero)
56. switchView.setOn(roomData.occupied, animated: true)
57. switchView.tag = roomData.number
58. switchView.addTarget(self, action: #selector(self.switchChanged(\_:)), for: .valueChanged)
59. cell.accessoryView = switchView
60. return cell
61. }
63. func tableView(\_ tableView: UITableView, viewForHeaderInSection section: Int) -> UIView? {
64. if (tableData[section].count > 0) {
65. let view = tableView.dequeueReusableHeaderFooterView(withIdentifier:
66. "sectionHeader") as! SectionHeader
67. view.title.text = "Floor \(section)"
68. var count = 0;
69. \_ = tableData[section].map({
70. if($0.1) {
71. count+=1;
72. }
73. })
74. view.switcher.setOn(count == tableData[section].count, animated: true)
75. view.switcher.tag = section
76. view.switcher.addTarget(self, action: #selector(self.sectionSwitchChanged(\_:)), for: .valueChanged)
77. return view
78. }
79. return nil
80. }

83. func tableView(\_ tableView: UITableView, heightForHeaderInSection section: Int) -> CGFloat {
84. if (tableData[section].count > 0) {
85. return 50
86. }
87. return 0
88. }
90. @objc func switchChanged(\_ sender : UISwitch!){
91. let section = Int(floor(Double(sender.tag / 100)))
92. let room = sender.tag % 100
93. self.tableData[section][room - 1].occupied = sender.isOn;
94. tableView.reloadData()
95. }
97. @objc func sectionSwitchChanged(\_ sender : UISwitch!){
98. let section = sender.tag
99. let rooms = self.tableData[section]
100. self.tableData[section] = rooms.map({($0.0, sender.isOn)})
101. tableView.reloadData()
102. }
104. @IBAction func logout(\_ sender: Any) {
105. let storyBoard : UIStoryboard = UIStoryboard(name: "Main", bundle:nil)
106. let LoginViewController = storyBoard.instantiateViewController(withIdentifier: "loginViewController") as! LoginViewController
107. LoginViewController.modalPresentationStyle = .fullScreen
108. self.present(LoginViewController, animated:true, completion:nil)
109. }
111. }
113. class SectionHeader: UITableViewHeaderFooterView {
114. let title = UILabel()
115. let switcher = UISwitch(frame: .zero)
117. override init(reuseIdentifier: String?) {
118. super.init(reuseIdentifier: reuseIdentifier)
119. configureContents()
120. }
122. required init?(coder: NSCoder) {
123. fatalError("init(coder:) has not been implemented")
124. }
126. func configureContents() {
127. title.translatesAutoresizingMaskIntoConstraints = false
128. switcher.translatesAutoresizingMaskIntoConstraints = false
130. contentView.addSubview(title)
131. contentView.addSubview(switcher)
133. NSLayoutConstraint.activate([
134. title.leadingAnchor.constraint(equalTo: contentView.layoutMarginsGuide.leadingAnchor),
135. title.heightAnchor.constraint(equalToConstant: 50),
136. title.centerYAnchor.constraint(equalTo: contentView.centerYAnchor),
138. switcher.heightAnchor.constraint(equalToConstant: 30),
139. switcher.leadingAnchor.constraint(equalTo: title.trailingAnchor,
140. constant: 8),
141. switcher.trailingAnchor.constraint(equalTo:
142. contentView.layoutMarginsGuide.trailingAnchor),
143. switcher.centerYAnchor.constraint(equalTo: contentView.centerYAnchor)
144. ])
145. }
146. }

Add room controller

1. import UIKit
3. class AddRoomViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource, UITextFieldDelegate {

6. @IBOutlet weak var picker: UIPickerView!
8. var pickerData: [Int] = [Int]()
9. var tableData: [[(number: Int, occupied: Bool)]] = []
10. var seletedFloor: Int = 0;
12. override func viewDidLoad() {
13. super.viewDidLoad()
15. self.picker.delegate = self
16. self.picker.dataSource = self
18. pickerData = Array(0...99)
19. }
21. override func didReceiveMemoryWarning() {
22. super.didReceiveMemoryWarning()
23. }
25. func numberOfComponents(in pickerView: UIPickerView) -> Int {
26. return 1
27. }
29. func pickerView(\_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
30. return pickerData.count
31. }
33. func pickerView(\_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {
34. return "Floor \(pickerData[row])"
35. }
37. func pickerView(\_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {
38. self.seletedFloor = self.pickerData[row];
39. }

42. @IBAction func addFloor(\_ sender: UIBarButtonItem) {
43. let storyBoard : UIStoryboard = UIStoryboard(name: "Main", bundle:nil)
44. let TableViewController = storyBoard.instantiateViewController(withIdentifier: "tableViewController") as! TableViewController
45. TableViewController.modalPresentationStyle = .fullScreen
47. if (self.seletedFloor > -1) {
48. let existingRoomsOnFloor = tableData[seletedFloor]
49. tableData[seletedFloor].append(((seletedFloor \* 100) + (existingRoomsOnFloor.count + 1), true))
50. TableViewController.updatedTableData = tableData;
51. self.seletedFloor = 0
52. }
54. self.present(TableViewController, animated:true, completion:nil)
55. }
56. @IBAction func cancel(\_ sender: UIBarButtonItem) {
57. let storyBoard : UIStoryboard = UIStoryboard(name: "Main", bundle:nil)
58. let TableViewController = storyBoard.instantiateViewController(withIdentifier: "tableViewController") as! TableViewController
59. TableViewController.modalPresentationStyle = .fullScreen
60. self.present(TableViewController, animated:true, completion:nil)
61. }
62. }

Image Picker Manager

1. import UIKit
3. class ImagePickerManager: NSObject, UIImagePickerControllerDelegate, UINavigationControllerDelegate {
5. var picker = UIImagePickerController();
6. var alert = UIAlertController(title: "Choose Image", message: nil, preferredStyle: .actionSheet)
7. var viewController: UIViewController?
8. var pickImageCallback : ((UIImage) -> ())?;
10. override init(){
11. super.init()
12. }
14. func pickImage(\_ viewController: UIViewController, \_ callback: @escaping ((UIImage) -> ())) {
15. pickImageCallback = callback;
16. self.viewController = viewController;
18. let cameraAction = UIAlertAction(title: "Camera", style: .default){
19. UIAlertAction in
20. self.openCamera()
21. }
22. let galleryAction = UIAlertAction(title: "Gallery", style: .default){
23. UIAlertAction in
24. self.openGallery()
25. }
26. let cancelAction = UIAlertAction(title: "Cancel", style: .cancel){
27. UIAlertAction in
28. }
30. picker.delegate = self
31. alert.addAction(cameraAction)
32. alert.addAction(galleryAction)
33. alert.addAction(cancelAction)
34. alert.popoverPresentationController?.sourceView = self.viewController!.view
35. viewController.present(alert, animated: true, completion: nil)
36. }
37. func openCamera(){
38. alert.dismiss(animated: true, completion: nil)
39. if(UIImagePickerController .isSourceTypeAvailable(.camera)){
40. picker.sourceType = .camera
41. self.viewController!.present(picker, animated: true, completion: nil)
42. } else {
43. let alert = UIAlertController(title: "Warning", message: "You don't have camera", preferredStyle: .alert)
44. alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: nil))
45. self.viewController!.present(alert, animated: true, completion: nil)
46. }
47. }
48. func openGallery(){
49. alert.dismiss(animated: true, completion: nil)
50. picker.sourceType = .photoLibrary
51. self.viewController!.present(picker, animated: true, completion: nil)
52. }

55. func imagePickerControllerDidCancel(\_ picker: UIImagePickerController) {
56. picker.dismiss(animated: true, completion: nil)
57. }
59. func imagePickerController(\_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any]) {
60. picker.dismiss(animated: true, completion: nil)
61. guard let image = info[.originalImage] as? UIImage else {
62. fatalError("Expected a dictionary containing an image, but was provided the following: \(info)")
63. }
64. pickImageCallback?(image)
65. }


69. @objc func imagePickerController(\_ picker: UIImagePickerController, pickedImage: UIImage?) {
70. }
72. }