SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya School of Engineering
(formerly K J Somaiya College of Engineering)

Somaiya TRUST

FOS mini project

SHADE : scheduler app

Yashraj Ola : 16010123337

Shravani Parte : 16010123319

Scheduler App: Details & Features

Key Features

- **Smart Task Management**: Add, complete, and delete tasks with a single tap
- **Time-Based Organization**: Categorize tasks by morning- noon, noon-evening, and evening-night
- **Weekly View**: Seamless navigation between weekdays
- **Instant Task Status**: Visual indicators show completion status at a glance
- **Gesture Controls**: Swipe back to home from any screen for quick navigation
- **Secure Account System**: Protected personal schedule with login/signup
- **Intuitive Timeline**: Visual representation of your day broken into logical time blocks
- **Quick Navigation**: Connected screens via navigation controller
- **Progress Tracking**: See your daily and weekly task completion rates
- **One-Touch Completion**: Mark tasks as done with a simple tap
- **Clean Task Removal**: Delete completed or unnecessary tasks instantly
- **Real-Time Updates**: See your schedule changes reflected immediately
- **User-Friendly Interface**: Clean design with minimal learning curve
- **Efficient Day Planning**: Organize your day in meaningful time blocks

User Benefits

- Organize your day with minimal effort
- Quickly see what's coming up next
- Celebrate completed tasks with visual feedback
- Seamlessly plan your entire week ahead
- Navigate your schedule with intuitive gestures
- Maintain productivity throughout different parts of your day
- Keep your daily organization simple and effective

Code :

ViewController.swift : Root Home Page

```
import UIKit

class ViewController: UIViewController {


    @IBOutlet weak var floatingImageView1: UIImageView!

    @IBOutlet weak var TITLE: UILabel!
    @IBOutlet weak var floatingImageView2: UIImageView!
    @IBOutlet weak var floatingImageView3: UIImageView!
    override func viewDidLoad() {
        super.viewDidLoad()
        TITLE.textColor = .white
        startFloatingAnimation(imageView: floatingImageView2)
        startFloatingAnimation(imageView: floatingImageView1)
        startFloatingAnimation(imageView: floatingImageView3)
        let backgroundImageView = UIImageView(frame: view.bounds)
          backgroundImageView.image = UIImage(named: "what") // Replace "background" with the
name of your image in Assets.xcassets
          backgroundImageView.contentMode = .scaleAspectFill // Or other content modes like
.scaleToFill, .scaleAspectFit
        view.insertSubview(backgroundImageView, at: 0)

    }
    @IBAction func unwind(segue:UIStoryboardSegue){}

    func startFloatingAnimation(imageView: UIImageView) {
        guard imageView != nil else {
            print("Error: imageView IBOutlet is not connected in the Storyboard!")
            return
        }

        func startFloatingAnimation(imageView: UIImageView) {
            guard imageView != nil else {
                print("Error: imageView IBOutlet is not connected in the Storyboard!")
                return
```

```
        }

        let horizontalOffset: CGFloat = 5.0
        let verticalOffset: CGFloat = 3.0
        let duration: TimeInterval = 2.0

        UIView.animate(withDuration: duration, delay: 0, options: [.repeat, .autoreverse],
animations: {
            imageView.transform = CGAffineTransform(translationX: horizontalOffset, y:
verticalOffset)
        }) { _ in
            UIView.animate(withDuration: duration, delay: 0, options: [.repeat, .autoreverse],
animations: {
                imageView.transform = .identity
            })
        }
    }
  }
}
```

abViewController : Registration page

```
//
//  newViewController.swift
//  shade
//
//  Created by KJSCE on 08/04/25.
//

import UIKit


class abViewController: UIViewController {

   override func viewDidLoad() {
      super.viewDidLoad()
      let backgroundImageView = UIImageView(frame: view.bounds)
        backgroundImageView.image = UIImage(named: "what") // Replace "background" with the
name of your image in Assets.xcassets
        backgroundImageView.contentMode = .scaleAspectFill // Or other content modes like
.scaleToFill, .scaleAspectFit
      view.insertSubview(backgroundImageView, at: 0)


  } }
```

aViewController : Login page

```
//
//  aViewController.swift
//  shade
//
//  Created by Yashraj Parag Ola on 20/04/25.
//

import UIKit

class aViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        let backgroundImageView = UIImageView(frame: view.bounds)
            backgroundImageView.image = UIImage(named: "what") // Replace "background" with the
name of your image in Assets.xcassets
            backgroundImageView.contentMode = .scaleAspectFill // Or other content modes like
.scaleToFill, .scaleAspectFit
        view.insertSubview(backgroundImageView, at: 0)
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */

}
```

welcome.swift : shows successful login

```
//
//  loginViewController.swift
//  shade
//
//  Created by Yashraj Parag Ola on 20/04/25.
//

import UIKit

class welcome: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        let backgroundImageView = UIImageView(frame: view.bounds)
          backgroundImageView.image = UIImage(named: "wlcm") // Replace "background" with the
name of your image in Assets.xcassets
          backgroundImageView.contentMode = .scaleAspectFill // Or other content modes like
.scaleToFill, .scaleAspectFit
        view.insertSubview(backgroundImageView, at: 0)
        // Do any additional setup after loading the view.
    }


    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */

}
```

abcdViewController : Days of the week page

```
//
//  abcdViewController.swift
//  shade
//
//  Created by Yashraj Parag Ola on 19/04/25.
//

import UIKit

class abcdViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        setupBackground()
    }

    func setupBackground() {
        let backgroundImageView = UIImageView(frame: view.bounds)
        backgroundImageView.image = UIImage(named: "sunset")
        backgroundImageView.contentMode = .scaleAspectFill
        view.insertSubview(backgroundImageView, at: 0)
    }

    @IBAction func mondayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Monday")
    }

    @IBAction func tuesdayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Tuesday")
    }

    @IBAction func wednesdayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Wednesday")
    }

    @IBAction func thursdayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Thursday")
    }

    @IBAction func fridayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Friday")
    }
```

```
    @IBAction func saturdayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Saturday")
    }

    @IBAction func sundayButtonTapped(_ sender: UIButton) {
        performSegue(withIdentifier: "goToTimeSlot", sender: "Sunday")
    }

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "goToTimeSlot",
           let timeSlotVC = segue.destination as? abcViewController,
           let selectedDay = sender as? String {
            timeSlotVC.selectedDay = selectedDay
        }
    }
}
```

abcViewController : slots of the day page

```
//
//  abcViewController.swift
//  shade
//
//  Created by Yashraj Parag Ola on 19/04/25.
//

import UIKit

class abcViewController: UIViewController {

    var selectedDay: String?

    override func viewDidLoad() {
        super.viewDidLoad()
        setupBackground()
    }

    func setupBackground() {
        let backgroundImageView = UIImageView(frame: view.bounds)
        backgroundImageView.image = UIImage(named: "what2")
        backgroundImageView.contentMode = .scaleAspectFill
        view.insertSubview(backgroundImageView, at: 0)
    }

    @IBAction func morningToNoonButtonTapped(_ sender: UIButton) {
        navigateToTodoList(withTimeSlot: "Morning to Noon")
    }

    @IBAction func afternoonToEveningButtonTapped(_ sender: UIButton) {
        navigateToTodoList(withTimeSlot: "Afternoon to Evening")
    }

    @IBAction func eveningToNightButtonTapped(_ sender: UIButton) {
        navigateToTodoList(withTimeSlot: "Evening to Night")
    }

    func navigateToTodoList(withTimeSlot timeSlot: String) {
        if let day = selectedDay {
            let dayAndTime = "\(day) - \(timeSlot)"
            performSegue(withIdentifier: "goToTodoList", sender: dayAndTime)
        }
    }
```

```
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        if segue.identifier == "goToTodoList",
           let todoListVC = segue.destination as? TodoListViewController,
           let dayAndTime = sender as? String {
            todoListVC.dayAndTime = dayAndTime
        }
    }
}
```

TodoListViewController : tasks in that slot page

```
//  TodoListViewController.swift
import UIKit

class TodoListViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    var dayAndTime: String?
    var todoItems: [String] = []
    @IBOutlet weak var todoTableView: UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()
        setupBackground()
        setupTableView()
        setupNavigationBar()

        if let dayAndTime = dayAndTime {
            print("Selected Day and Time in TodoListVC: \(dayAndTime)")
            loadTodoItems(for: dayAndTime)
        }
    }

    func setupBackground() {
        // Set the background image
        if let image = UIImage(named: "daysbg.jpeg") {
            view.backgroundColor = UIColor(patternImage: image)
        } else {
            view.backgroundColor = UIColor(red: 0.95, green: 0.95, blue: 0.95, alpha: 1.0)
            print("Error: Could not load background image!")
        }
    }

    func setupNavigationBar() {
        // Use a smaller, bold, and white font for the title
        let titleLabel = UILabel()
        titleLabel.text = dayAndTime
        titleLabel.font = UIFont.systemFont(ofSize: 16, weight: .bold)
        titleLabel.textColor = .white
        titleLabel.sizeToFit()
        navigationItem.titleView = titleLabel

        navigationItem.rightBarButtonItem = UIBarButtonItem(barButtonSystemItem: .add, target:
self, action: #selector(addTodoItem))
    }
```

```
    func setupTableView() {
        todoTableView.delegate = self
        todoTableView.dataSource = self
        todoTableView.register(UITableViewCell.self, forCellReuseIdentifier: "todoCell")
        todoTableView.backgroundColor = .clear
        todoTableView.separatorStyle = .none
    }

    // MARK: - Data Persistence (Using UserDefaults)
    var allTodos: [String: [String]] = [:]
    var completedItems: [IndexPath: Bool] = [:]

    func loadTodoItems(for key: String) {
        if let savedTodos = UserDefaults.standard.dictionary(forKey: "allTodosData") as? [String:
[String]] {
            allTodos = savedTodos
            todoItems = allTodos[key] ?? []
        } else {
            todoItems = []
            allTodos = [:]
        }

        if let savedCompleted = UserDefaults.standard.dictionary(forKey: "completedItemsData")
as? [String: Bool] {
            completedItems = savedCompleted.reduce(into: [:]) { (result, element) in
                let keyString = element.key
                let components = keyString.components(separatedBy: ",")
                if components.count == 2,
                   let section = Int(components[0].trimmingCharacters(in: .whitespacesAndNewlines)),
                   let row = Int(components[1].trimmingCharacters(in: .whitespacesAndNewlines)) {
                    let indexPath = IndexPath(row: row, section: section)
                    result[indexPath] = element.value
                }
            }
        } else {
            completedItems = [:]
        }
        todoTableView.reloadData()
    }

    func saveTodoItem(_ item: String, for key: String) {
        print("saveTodoItem called with item: \(item), key: \(key)")
        if var existingItems = allTodos[key] {
            existingItems.append(item)
```

```
            allTodos[key] = existingItems
        } else {
            allTodos[key] = [item]
        }
        todoItems = allTodos[key] ?? []
        UserDefaults.standard.set(allTodos, forKey: "allTodosData")

        let completedDict = completedItems.reduce(into: [String: Bool]()) { (result, element) in
            let keyString = "\(element.key.section),\(element.key.row)"
            result[keyString] = element.value
        }
        UserDefaults.standard.set(completedDict, forKey: "completedItemsData")
    }

    @objc func addTodoItem() {
        let alertController = UIAlertController(title: "Add New Task", message: nil, preferredStyle:
.alert)
        alertController.addTextField { textField in
            textField.placeholder = "Enter task description"
        }
        let addAction = UIAlertAction(title: "Add", style: .default) { [weak self] _ in
            if let textField = alertController.textFields?.first,
               let newItem = textField.text,
               !newItem.isEmpty,
               let currentKey = self?.dayAndTime {
                self?.saveTodoItem(newItem, for: currentKey)
                print("Item saved: \(newItem) for key: \(currentKey ?? "nil")")
                print("Current allTodos dictionary: \(self?.allTodos ?? [:])")
                print("Current todoItems array: \(self?.todoItems ?? [])")
                self?.todoTableView.reloadData()
            }
        }
        let cancelAction = UIAlertAction(title: "Cancel", style: .cancel)
        alertController.addAction(addAction)
        alertController.addAction(cancelAction)
        print("Presenting Alert Controller")
        present(alertController, animated: true)
    }

    // MARK: - Table View Delegate and Data Source

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return todoItems.count
    }
```

```
    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
        print("Creating cell for row: \(indexPath.row), todoItems count: \(todoItems.count)")
        let cell = tableView.dequeueReusableCell(withIdentifier: "todoCell", for: indexPath)

        let item = todoItems[indexPath.row]
        let isCompleted = completedItems[indexPath] == true

        // Apply strikethrough if completed
        if isCompleted {
            let attributedString = NSAttributedString(
                string: item,
                attributes: [
                    NSAttributedString.Key.strikethroughStyle: NSNumber(value:
NSUnderlineStyle.single.rawValue),
                    NSAttributedString.Key.foregroundColor: UIColor.gray //keep color
                ]
            )
            cell.textLabel?.attributedText = attributedString
        } else {
            cell.textLabel?.text = item
            cell.textLabel?.textColor = .white // Ensure text is white
        }

        cell.backgroundColor = .clear // Make cell background clear
        cell.layer.cornerRadius = 8
        cell.layer.masksToBounds = true
        cell.selectionStyle = .none

        // Add a delete button to the cell
        let deleteButton = UIButton(type: .system)
        deleteButton.setImage(UIImage(systemName: "trash"), for: .normal)
        deleteButton.tintColor = .red
        deleteButton.frame = CGRect(x: cell.contentView.bounds.width - 40, y:
(cell.contentView.bounds.height - 30) / 2, width: 30, height: 30)
        deleteButton.autoresizingMask = [.flexibleLeftMargin, .flexibleTopMargin,
.flexibleBottomMargin]
        deleteButton.addTarget(self, action: #selector(deleteButtonTapped(_:event:)), for:
.touchUpInside)
        cell.contentView.addSubview(deleteButton)

        // Adjust the text label's frame.  Use the cell's contentView bounds.
        cell.textLabel?.frame = CGRect(x: cell.contentView.bounds.origin.x + 16,
                                       y: cell.contentView.bounds.origin.y,
                                       width: cell.contentView.bounds.width - 80,
```

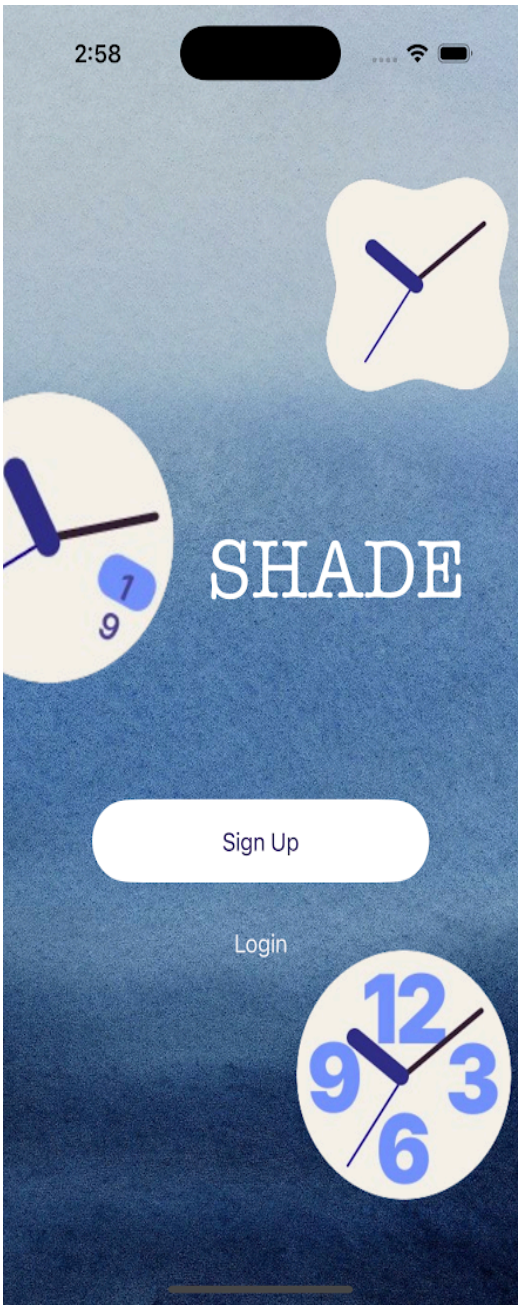
```
                                height: cell.contentView.bounds.height)

        return cell
    }

    @objc func deleteButtonTapped(_ sender: UIButton, event: UIEvent) {
        if let touch = event.touches(for: sender)?.first,
           let cell = touch.view?.superview?.superview as? UITableViewCell,
           let indexPath = todoTableView.indexPath(for: cell),
           let currentKey = dayAndTime {

            todoItems.remove(at: indexPath.row)
            allTodos[currentKey]?.remove(at: indexPath.row)
            if allTodos[currentKey]?.isEmpty == true {
                allTodos.removeValue(forKey: currentKey)
            }
            completedItems.removeValue(forKey: indexPath)

            UserDefaults.standard.set(allTodos, forKey: "allTodosData")
            let completedDict = completedItems.reduce(into: [String: Bool]()) { (result, element) in
                let keyString = "\(element.key.section),\(element.key.row)"
                result[keyString] = element.value
            }
            UserDefaults.standard.set(completedDict, forKey: "completedItemsData")

            todoTableView.deleteRows(at: [indexPath], with: .fade)
        }
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
        completedItems[indexPath] = !(completedItems[indexPath] == true)

         let completedDict = completedItems.reduce(into: [String: Bool]()) { (result, element) in
            let keyString = "\(element.key.section),\(element.key.row)"
            result[keyString] = element.value
        }
        UserDefaults.standard.set(completedDict, forKey: "completedItemsData")

        tableView.reloadRows(at: [indexPath], with: .automatic)
    }

    func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
        return 50
    }
}
```

Storyboard :


Title
Navigation Controller
SHADE
Sign Up
Login
SIGN UP
Enter your name here
Enter your email here
Enter your password here
Sign Up
Today is a....
MONDAY
TUESDAY
WEDESDAY
THURSDAY
FRIDAY
SATURDAY
SUNDAY
NAMASTE
MORNING - NOON
AFTER NOON
- EVENING
EVENING TO NIGHT
Table View
LOGIN
Enter your name here
or
Enter your email here
Enter your password here
Continue
To continue
swipe right

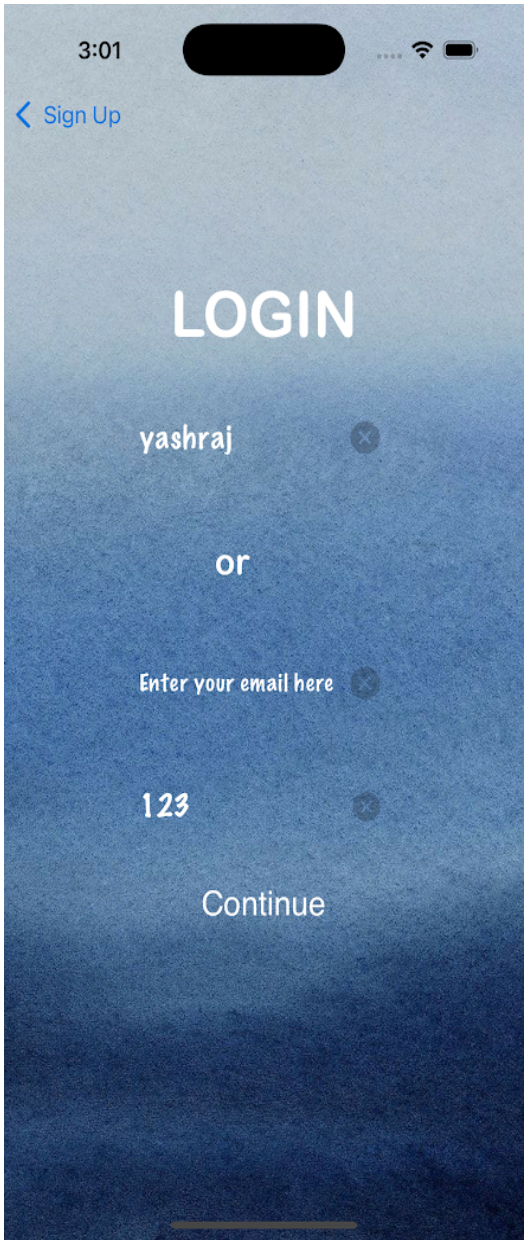
Simulator :

1.Home Page :



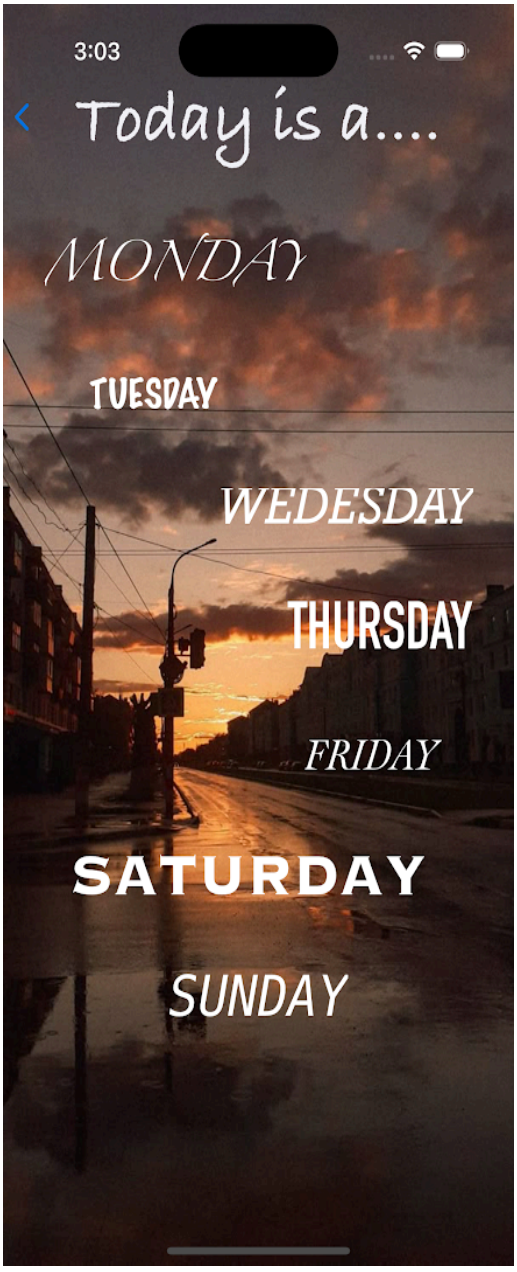
2. Clicking on sign up option :

3. On clicking on Sign Up button :



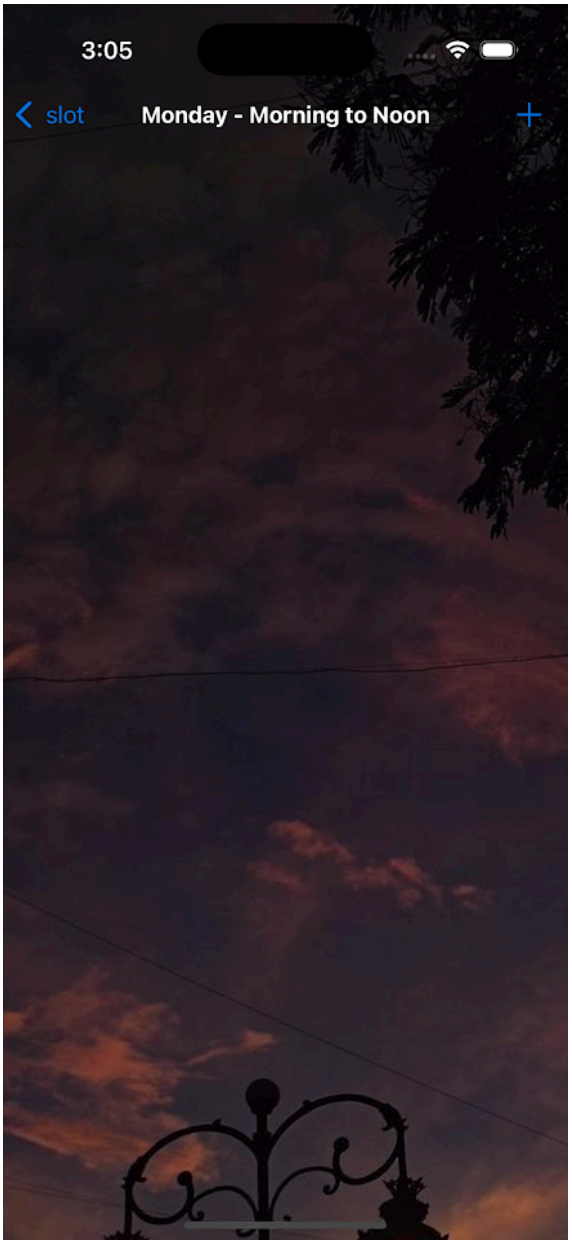
4.On clicking on continue button :

5. On swiping right :

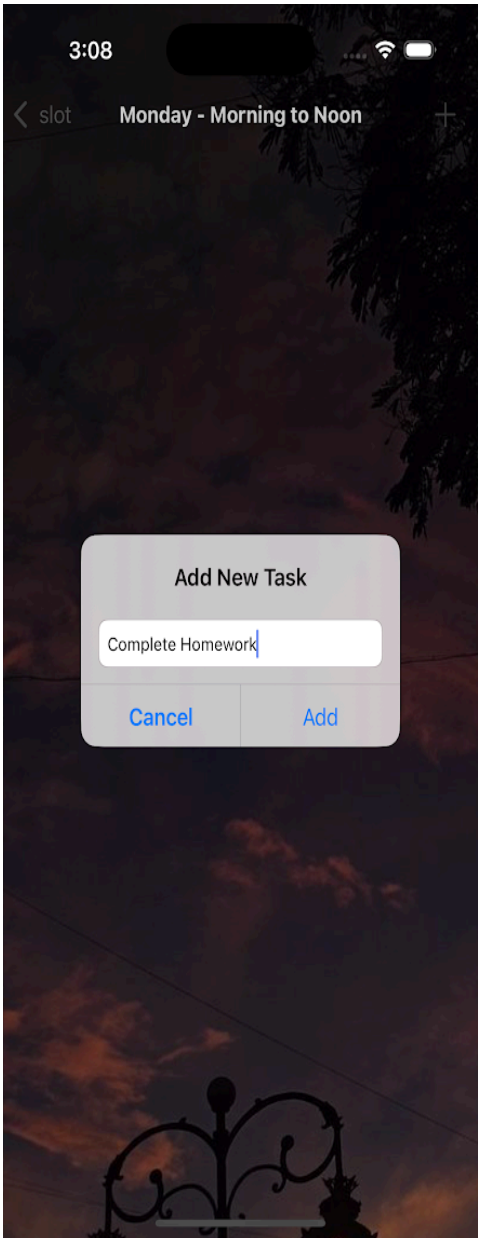


6. On clicking “Monday” button :

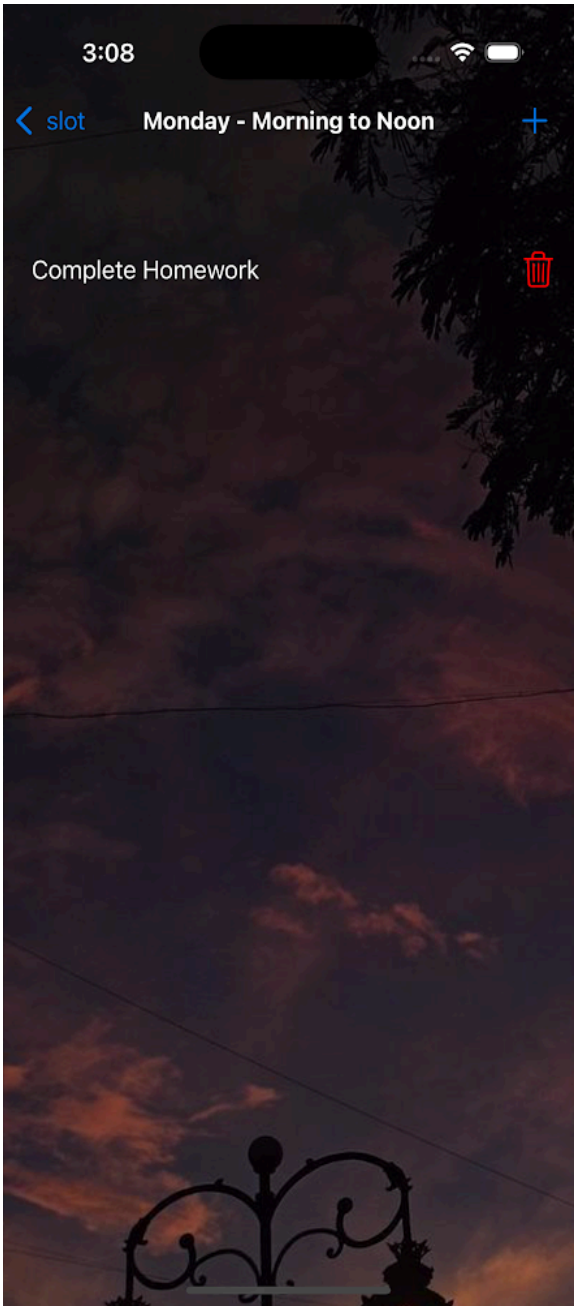
7.On clicking on button “Morning-Noon” slot :



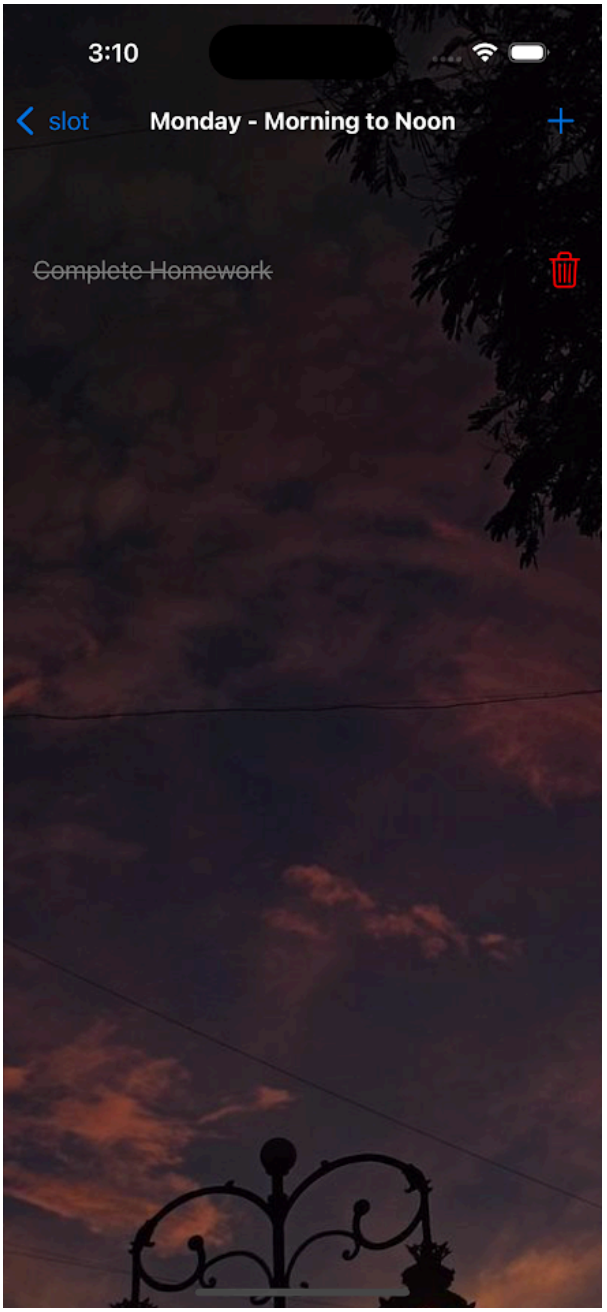
8.On clicking on add task “+” :



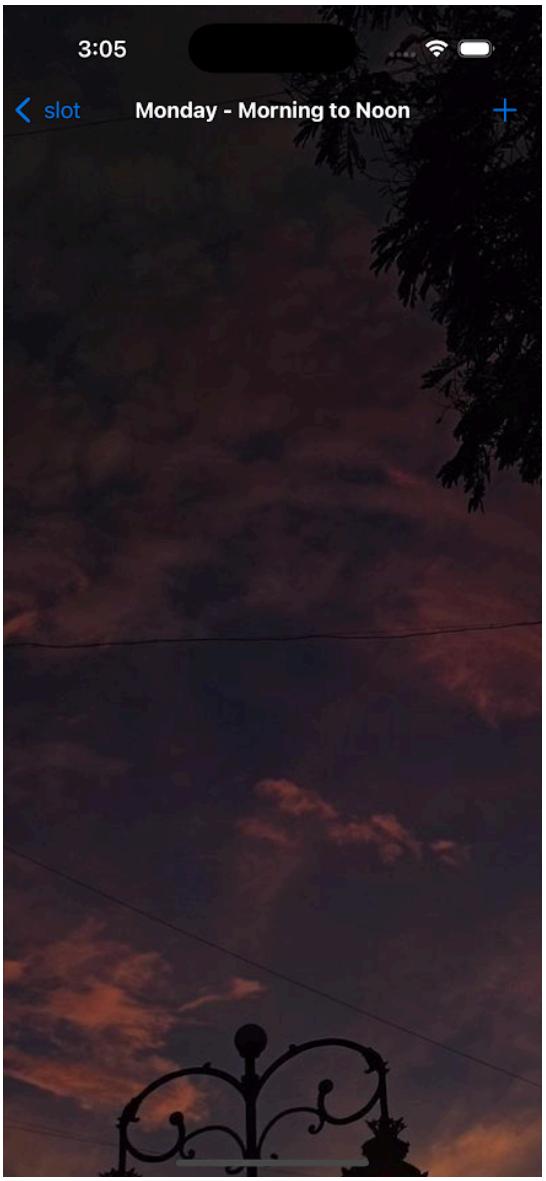
9.On clicking on “Add” button :



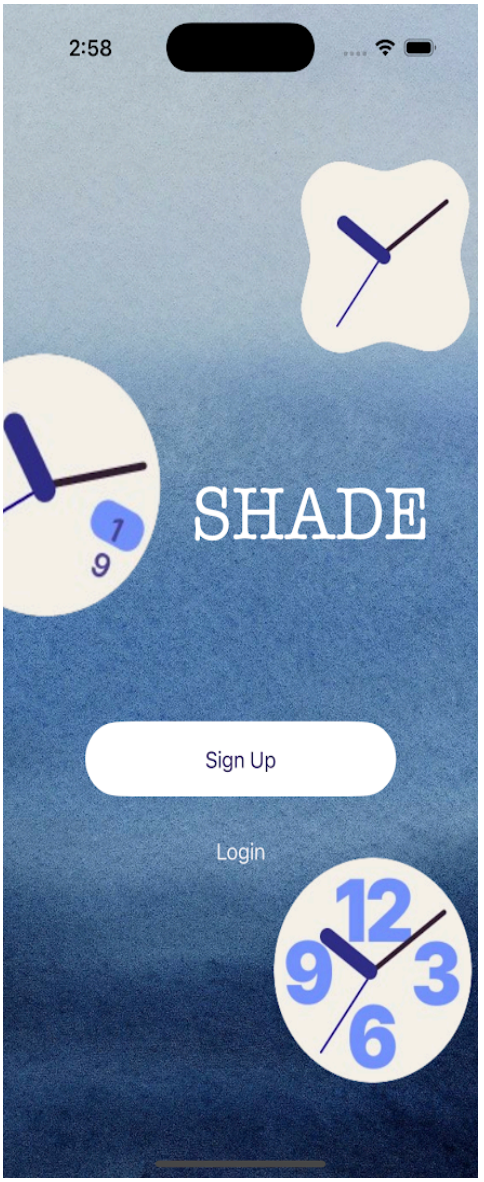
10.To mark task as completed- simply click on the task :



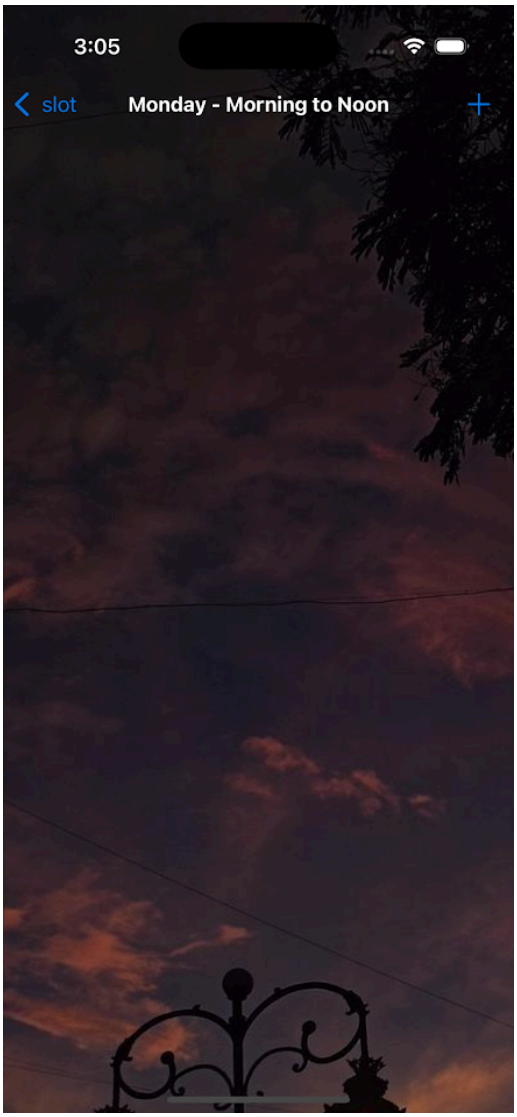
11.To delete the task -
click on the ‘bin’ button

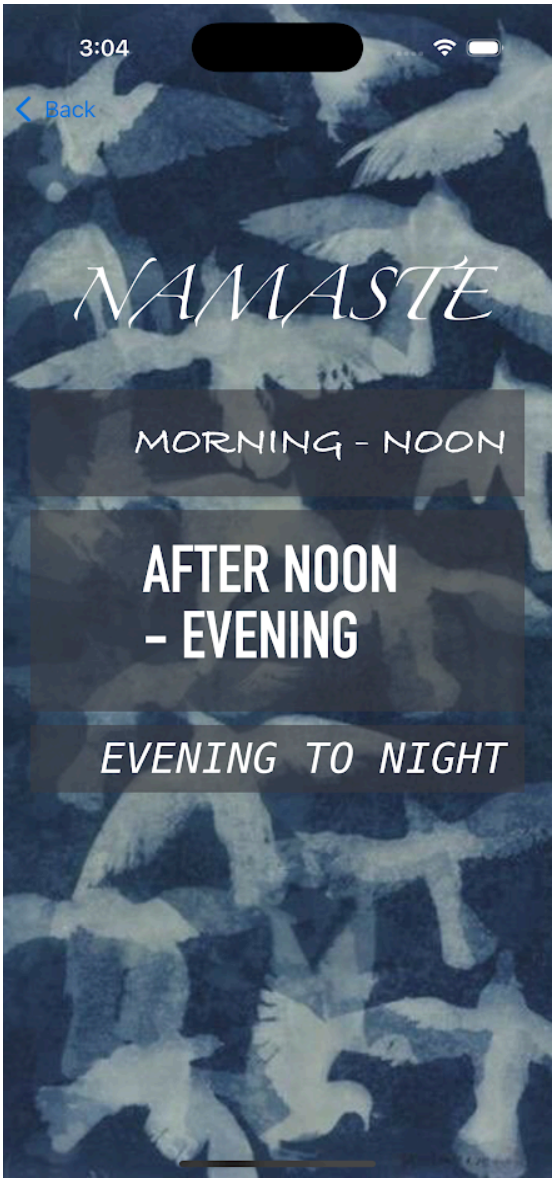


12.To go back to home
screen- swipe left



You can navigate between the pages using back button : (for eg)





Autolayout done of signup page as follows :