DAY OPTIMIZER

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

INSTRUCTED BY

Most. Kaniz Fatema Isha

Lecturer

Department of Computer Science and Engineering

Khulna University of Engineering and Technology

Argha Chandra Dhar

Lecturer

Department of Computer Science and Engineering

Khulna University of Engineering and Technology

SUBMITTED BY

Rolls : 1907063, 1907068, 1907069, 1907083, 1907088

Day Optimizer - A Personal Productivity Companion

1.Introduction:

Managing everyday tasks and maximizing productivity may often be difficult endeavors in the fast-paced world of modern living. Having acknowledged the widespread need for effortless planning and productivity, we are pleased to present Day Optimizer, an innovative iOS app, that enables users to plan the most efficient and productive days possible according to the weather.

1.1 Background:

Day Optimizer stems from the idea that if we can plan our day ahead based on weather data, we can significantly boost our productivity for the day. Even without weather data, we can use the app to enlist our daily tasks. Even without considering the notes facility, we can use the app to know the condition of weather of the current location or of any other location.

1.2 Objectives:

- Maximize daily productivity.
- Plan the Day ahead based on the user's location and weather information.
- Getting weather information of any given location.
- Creating a notes feature that allows users to add or delete their task.

2.Methodology:

- Signup and Login
- To Do List or Notes
- Weather

Signup and Login:

To Do List or Notes:

Weather:

1. Data Model:

The code defines three Swift structs (Current Data, Location Data, and Weather Data) to model the structure of the JSON data received from the Weather API.

2. View Controller and UI Setup:

The WeatherViewController class is created, which is a subclass of UIViewController.

IBOutlets are used to connect UI elements from the storyboard (e.g., buttons, labels, and text fields) to the view controller.

3. View Controller Lifecycle:

The viewDidLoad method is overridden to handle setup tasks when the view controller is loaded. In this case, it calls the fetchData method to initiate the initial weather data fetch.

4. Networking (Fetch Data):

The fetchData method is responsible for fetching weather data from the WeatherAPI.

It constructs a URL based on the user-entered location, creates a data task with URLSession, and asynchronously fetches and decodes the JSON response.

The UI is updated on the main thread with the received weather data using the updateUI method.

5. User Input Handling:

The entTapped method is triggered when the "ent" (Enter) button is tapped. It is called the fetchData method to fetch weather data for the user-entered location.

6. UI Update:

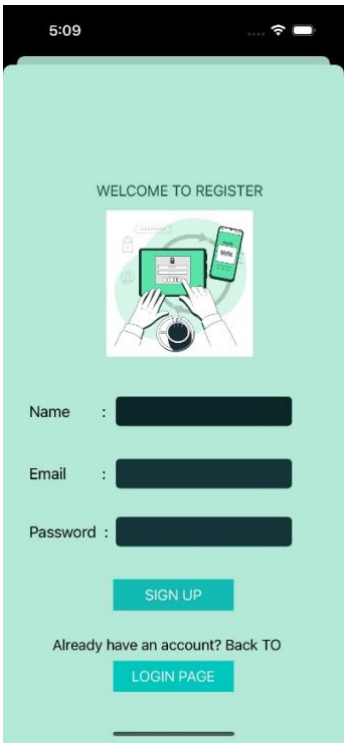
The updateUI method is responsible for updating the user interface with the received weather data. It extracts relevant information from the WeatherData model and updates the corresponding UI elements.

3.Result:

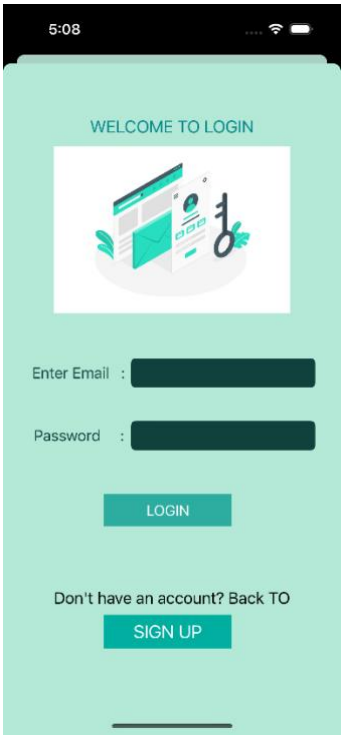
Some basic features with iOS app screenshots are given below –

App Name with Logo

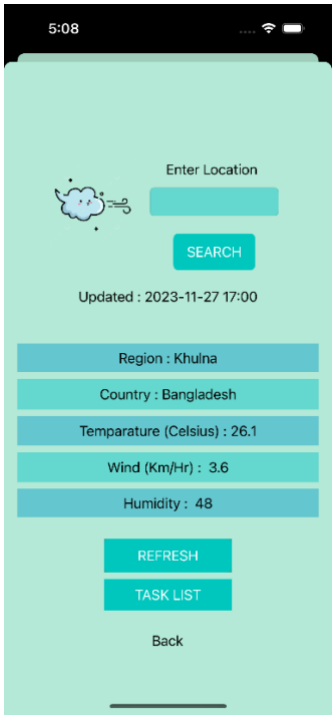
Splash Screen



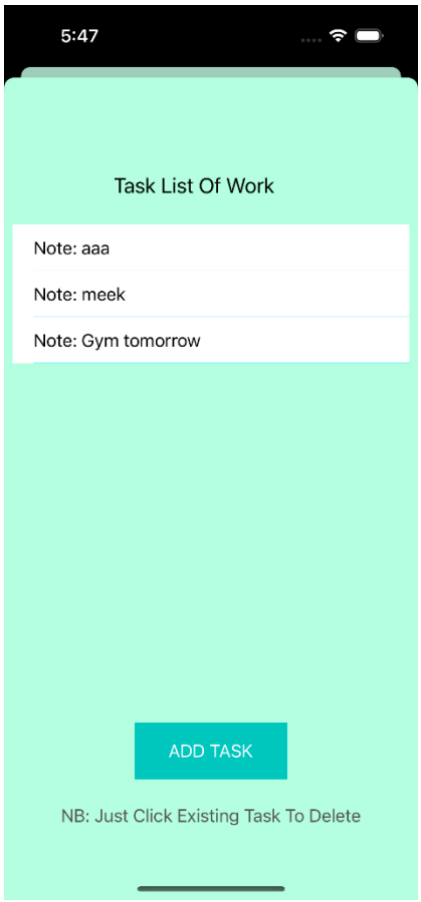
Registration



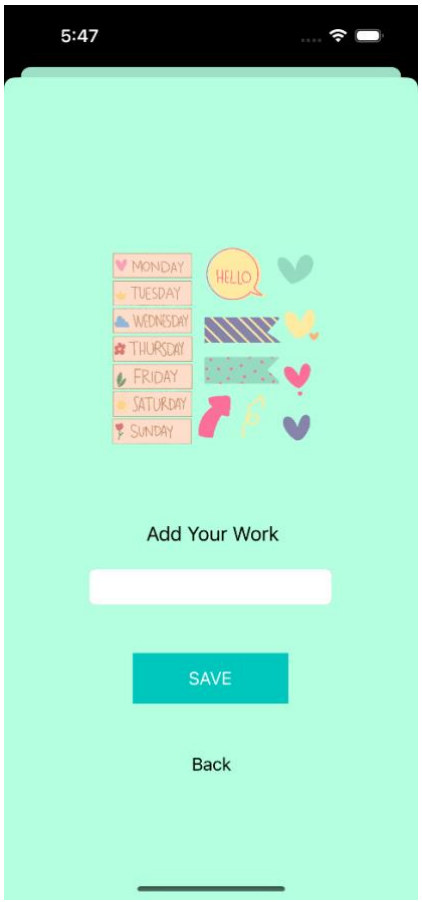
Login



Weather Show



Notes Record



Notes Add

4.Discussion:

Day Optimizer, an app that combines weather information with a to-do list providing users with a seamless experience for planning their day. Some key benefits of this app are –

- Efficient Planning
- Seamless Integration
- Personalization
- Improved Productivity
- Time Management

In case of considerations, it includes Data Privacy (ensuring compliance with data privacy regulations and providing transparent information about how user data, especially location data, will be used) , Accurate Weather Information (rely on reliable weather APIs to provide accurate and up-to-date weather information), User Interface Design (designing an intuitive and user-friendly interface that makes it easy for users to navigate between the weather section and the to-do list), Cross-Platform Compatibility (considering developing the app for multiple platforms (iOS, Android) to reach a wider audience), Testing and Feedback (conducting thorough testing to ensure the app's functionality and gather user feedback for continuous improvement) and above all, Regular Updates (staying updated with the latest weather APIs and technologies to provide users with accurate and relevant information). By combining weather information with a note's functionality, the app aims to enhance users' daily planning, productivity, and overall experience. Continuous improvement through user feedback and updates ensures that the app remains relevant and valuable to its users.

5.Conclusion:

Day Optimizer is a comprehensive iOS app that caters to the needs of users seeking weather information, location tracking, and task management. By leveraging APIs and incorporating a user-friendly interface, the app successfully delivers on its intended purpose. The ability to save tasks and share them with different users further enhances its functionality and appeal.

5.1 Future Work:

To further enhance the capabilities of Day Optimizer, the following areas of improvement can be considered:

- Personalized Weather Recommendations: Implement a feature that analyzes user habits and preferences to provide personalized weather recommendations, such as suggesting activities based on the forecast.
- Real-time Location Tracking with Alerts: Enhance the location tracking feature by enabling real-time updates and sending alerts for designated locations, ensuring user safety and convenience.
- Task Management with Priority Levels: Introduce a priority system for tasks, allowing users to categorize and prioritize their tasks effectively for better organization.
- Collaborative Task Management: Integrate collaborative task management features, enabling users to assign tasks to specific individuals and monitor progress collectively.
- Cross-platform Compatibility: Expand the app's compatibility to other operating systems, such as Android, to reach a wider audience and increase its usability.

6.References:

- **https://www.programiz.com/swift-programming**
- **https://developer.apple.com/xcode/**
- **https://www.avanderlee.com/swift/json-parsing-decoding/**
- **https://www.youtube.com/watch?v=JV9Oqyle3iE**