

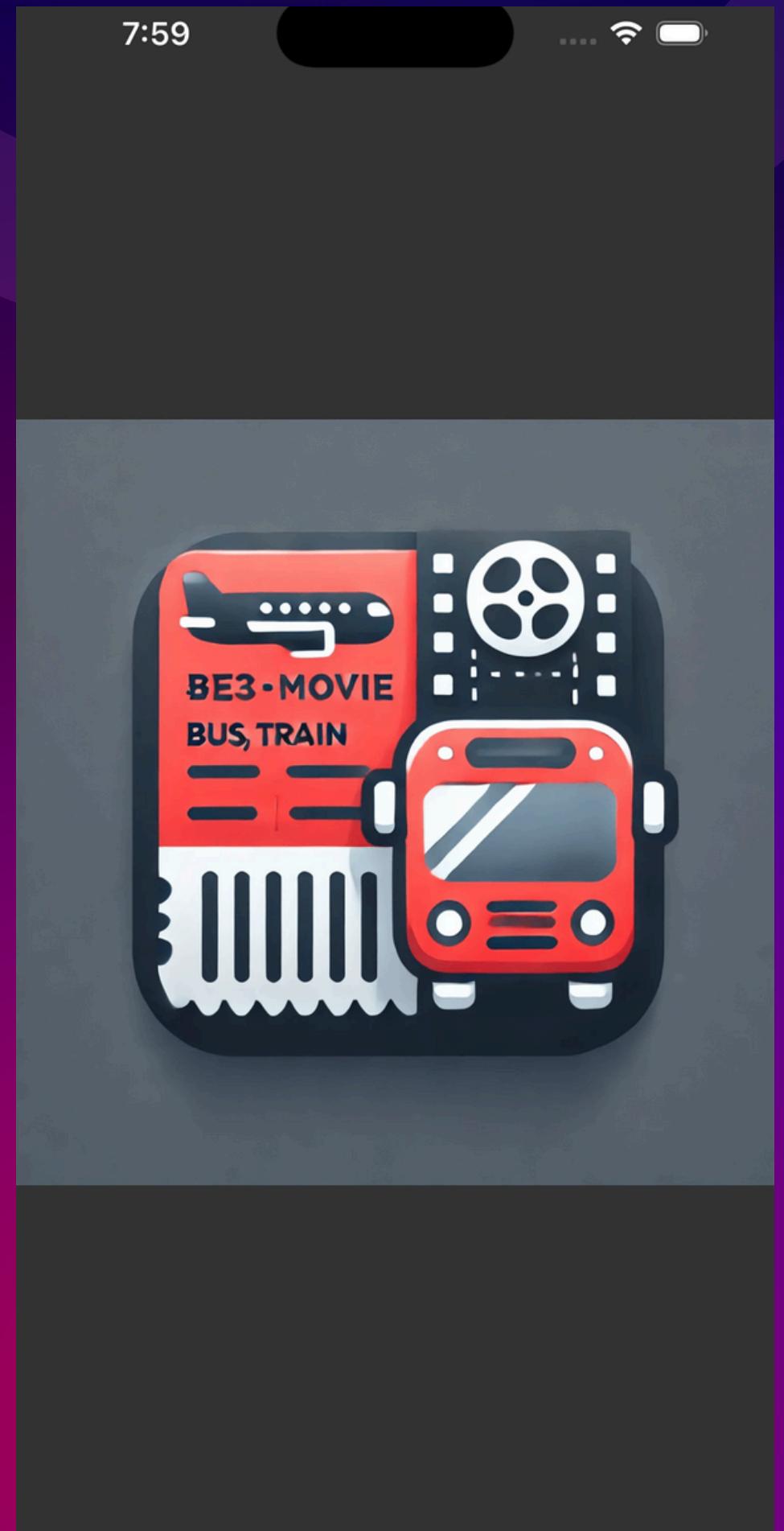


IOS TICKET BOOKING  
AND WHEATHER  
INFORMATION APP

# PRESENTATION

---

Presented by Pamaiahgari Rithik Goud





## VIEW CONTROLLER:

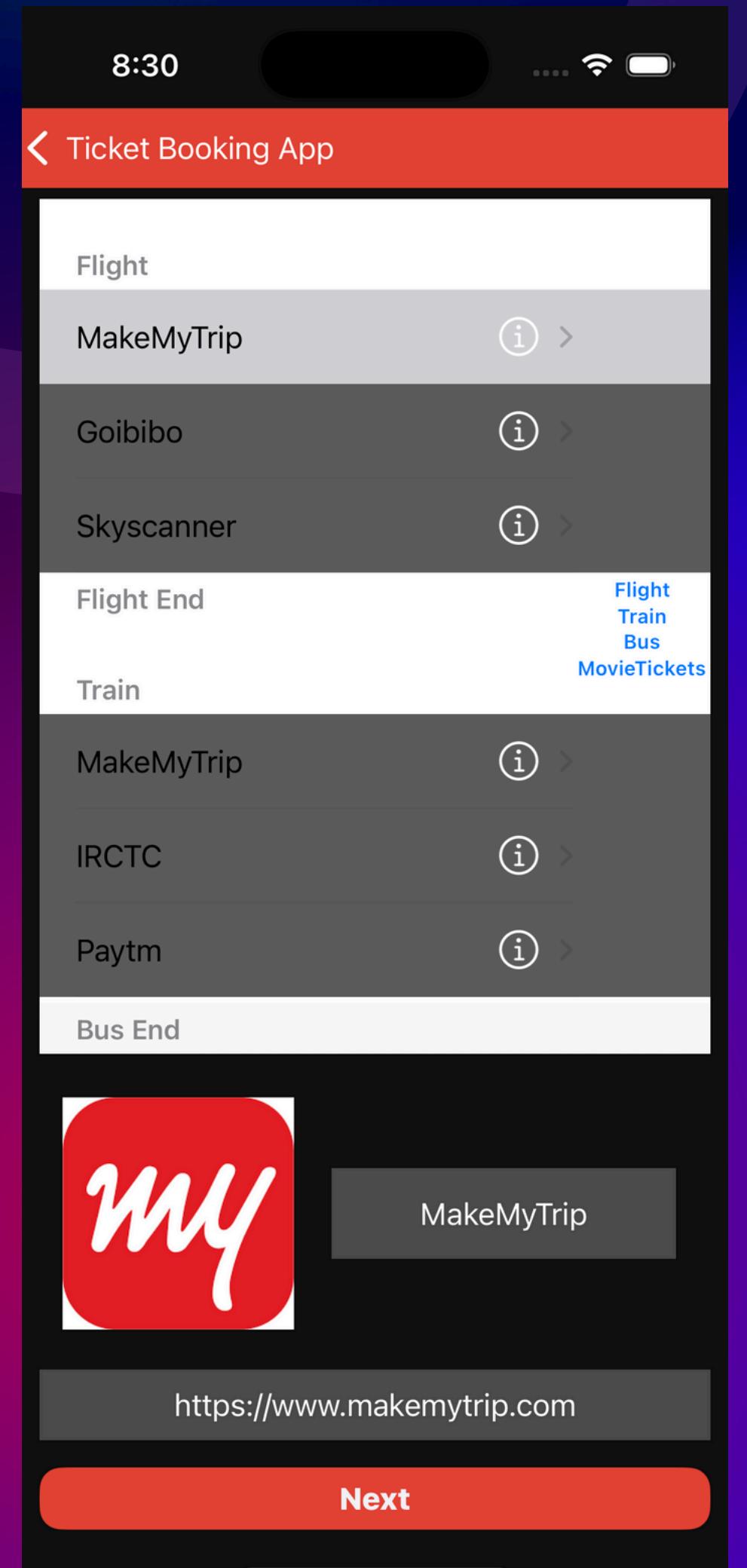
- The project consists of two modules:
  1. **Booking Module:** Allows users to book train, bus, movie, and flight tickets.
  2. **Weather Module:** Provides weather information based on the user's current location and global weather data.
- The initial view controller is set as a Navigation Controller in the storyboard to manage the navigation hierarchy. From the storyboard, two buttons are configured:
  - The first button navigates to the Booking Screen.
  - The second button navigates to the Weather Information Screen.
- In the initial view controller, the UIKit, Core Graphics, and Foundation frameworks are imported by default.
- UIKit: Used for creating and managing the user interface components.





## TICKET BOOKING VIEW CONTROLLER:

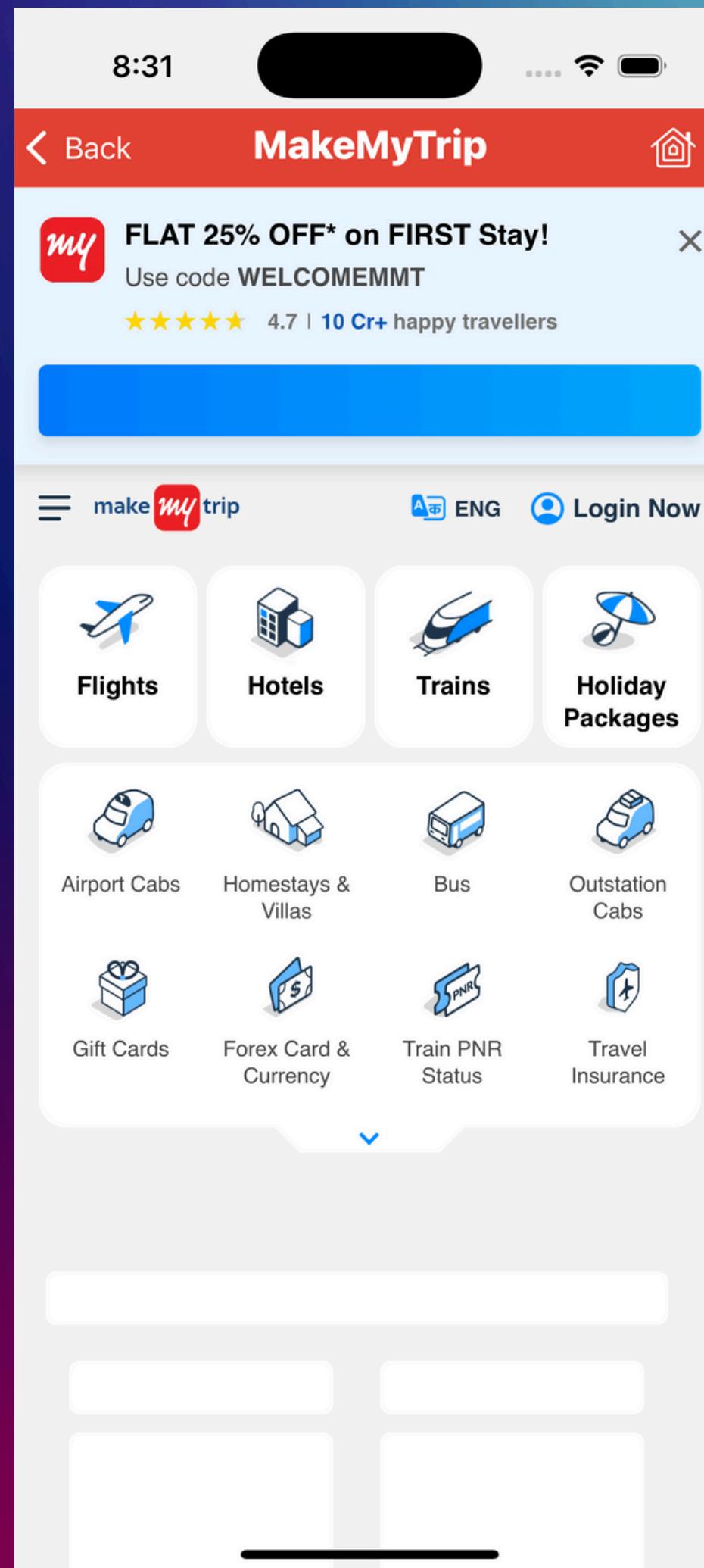
- From the View Controller, when the user clicks the "Book Ticket" button, they are navigated to the Ticket Booking View Controller.
- In the storyboard, a Table View Controller is used for the Ticket Booking View.
- The Table View Controller is divided into four sections: Train Tickets, Flight Tickets, Movie Tickets, Bus Tickets
- Each section contains three rows, with each row representing a different platform for booking tickets.
- When the user selects a row from any section:
  - The platform's image, URL name, and URL site are displayed.
  - After confirming the platform selection, the user clicks the "Next" button.
  - On clicking the "Next" button, an Action Sheet is displayed with the following options:
    1. Safari: Opens the selected platform URL in the Safari browser.
    2. WebKit: Opens the selected platform URL in an in-app WebKit-based browser.
    3. Cancel: Displays a destructive action type to dismiss the action sheet without proceeding.





## WEBPAGE VIEW CONTROLLER:

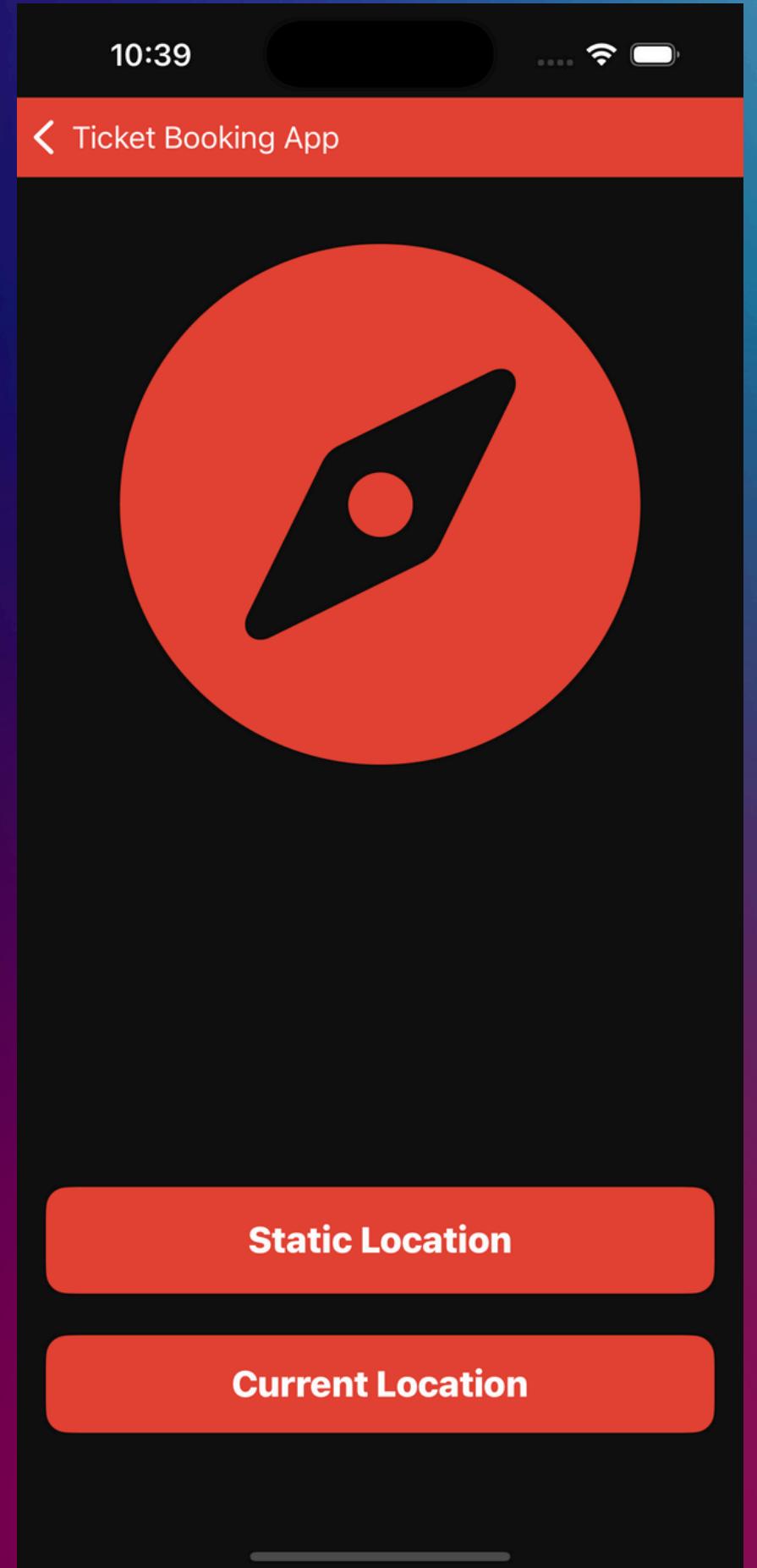
- When the user selects the WebKit option from the Ticket Booking View Controller, the app navigates to the WebPage View Controller.
- In the WebPage View Controller, the WebKit framework is imported.
- A WebView is added from the storyboard to enable in-app browsing functionality.
- If the user selects the Safari option from the Ticket Booking View Controller, the app directly opens the selected platform's URL in the default Safari browser.
- To perform this action, the SafariServices framework is imported.
- The URL is opened using the SFafariViewController class, providing a seamless redirection to Safari.





## WEATHER VIEW CONTROLLER:

- From the View Controller, when the user clicks on the "Weather Information" button, the app navigates to the Weather View Controller.
- In the Weather View Controller, two buttons are available:
- Current Location Weather Info: Displays weather information for the user's current location.
- Global Weather Info: Allows the user to search for and view weather information for any location worldwide.





## LOCATION VIEW CONTROLLER:

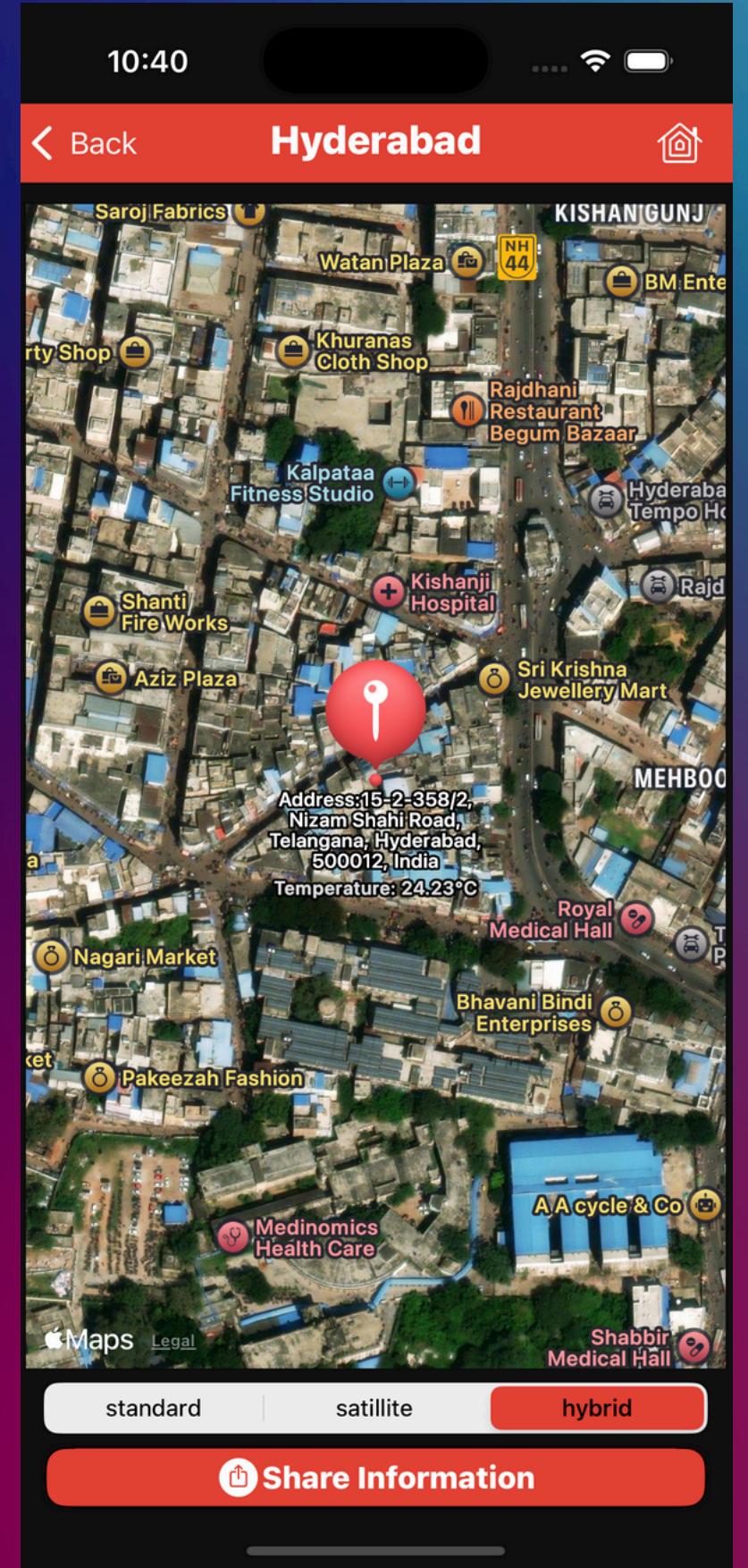
- When the user selects the Global Weather Info button, the app navigates to the Location View Controller.
- In the Location View Controller, the Core Location framework is imported to perform reverse geocoding, which fetches the address for a specific latitude and longitude.
- Using web services and JSON parsing, the app retrieves global weather information, including temperature, humidity, wind speed, latitude, longitude, and climate details.
- In the Location View Controller, text fields, buttons, and labels are used for user input and display:
- The user enters the city name and country in the text fields.
- Upon submitting the information, the app fetches the corresponding weather details and displays them in the labels.
- Additionally, a "Show Map" button is available to display the location on a map





## MAP VIEW CONTROLLER:

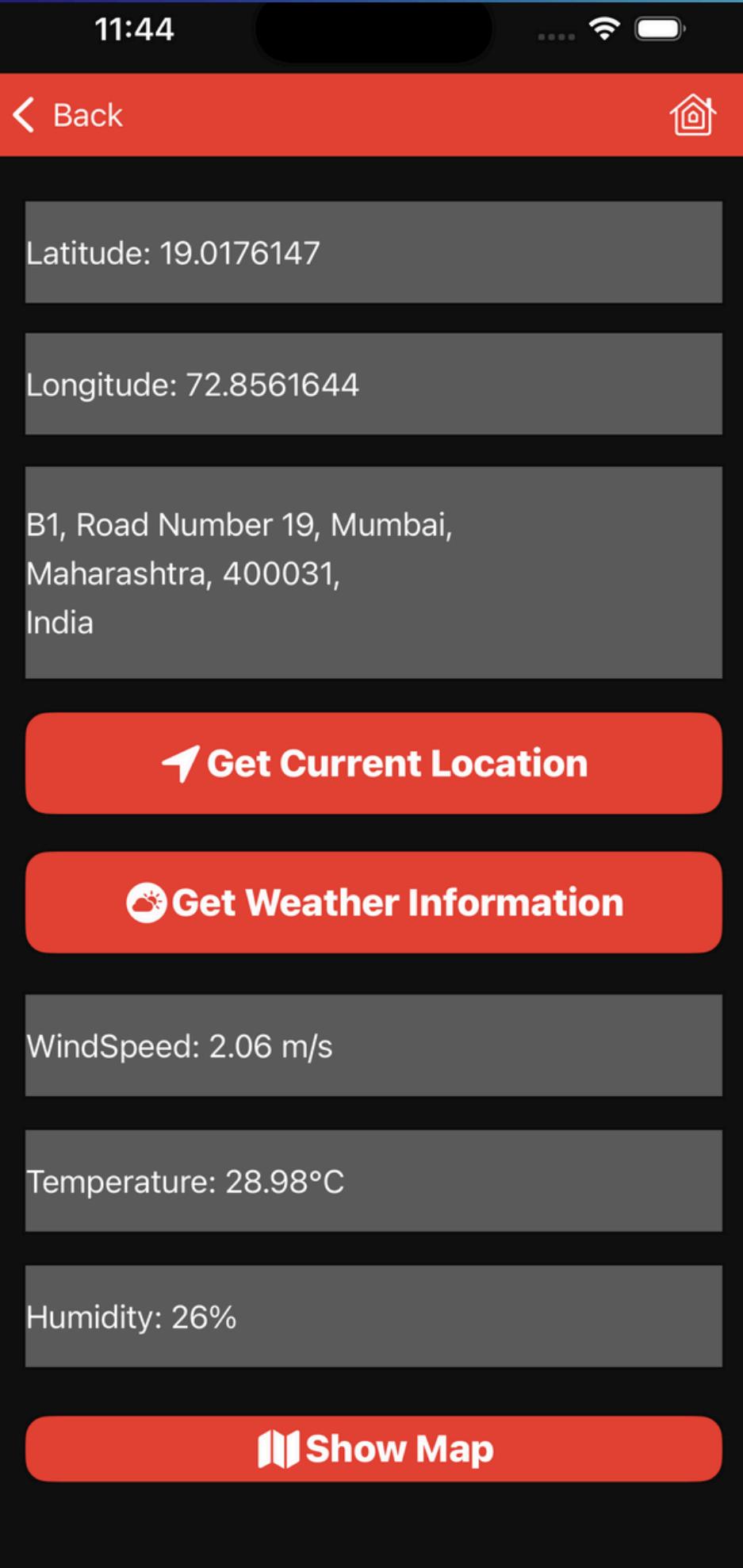
- When the user clicks on the "Show Map" button, the app navigates to the Map View Controller.
- In the Map View Controller, the MapKit framework is imported to provide map functionality.
- From the storyboard, a MapView is added, along with a Segmented Control.
- The Segmented Control is used to switch between three map types:  
**Standard , Satellite, Hybrid**
- When the user clicks on an annotation on the map, it displays the address and temperature of the specific location, providing detailed weather and location information.





## CURRENT VIEW CONTROLLER:

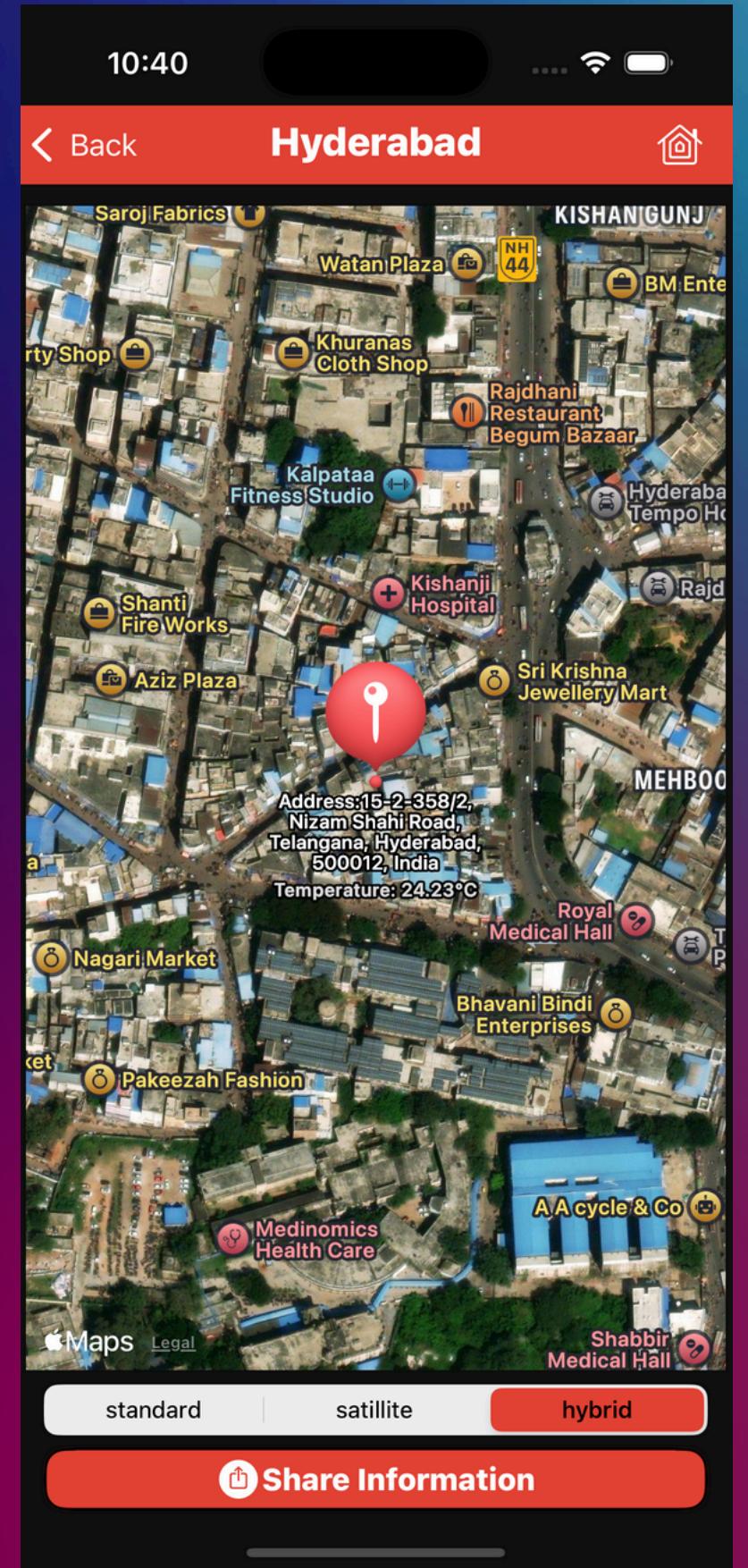
- When the user clicks on the Current Location button from the Weather View Controller, the app navigates to the Curr View Controller.
- In the Curr View Controller, the Core Location framework is imported to access the user's current location.
- Using web services and JSON parsing, the app fetches the current weather information based on the user's location.
- When the user clicks the Current Location button, the app fetches the user's latitude and longitude.
- Using these coordinates, the app performs reverse geocoding to retrieve the address of the user's current location.
- When the user clicks the Weather Info button, the app fetches the following weather details for the current location:
- Temperature, Humidity, Wind speed





## MAP VIEW CONTROLLER:

- When the user clicks on the "Show Map" button, the app navigates to the Map View Controller.
- In the Map View Controller, the MapKit framework is imported to provide map functionality.
- From the storyboard, a MapView is added, along with a Segmented Control.
- The Segmented Control is used to switch between three map types:
- Standard, Satellite, Hybrid
- When the user clicks on an annotation on the map, it displays the address and temperature of the specific location, providing detailed weather and location information.





## CONCLUSION:

---

Working on this app has given me immense confidence to create many more apps in the future. I used to only use apps, never expecting that I would one day build my own. Now, whenever I see an app, I can identify the UI objects they use and understand the functionalities implemented for those specific objects.

I am truly happy and proud to have learned so much during this journey into iOS development. This experience has been incredibly rewarding and has laid a solid foundation for me to grow as an iOS developer.