PYTHON PROJECT
INT216
FINAL PROJECT REPORT

LOVELY
PROFESSIONAL
UNIVERSITY

Transforming Education Transforming India

Topic: Weather App(Forecast Fusion)

Submitted By:
Ishaant Kumar Singh
Section - K22UP
Roll no. : 47

Submitted To:
Ved Prakash Chaubey
UID: 63892

DECLARATION

I hereby declare that the project work entitled ("Weather App (Forecast Fusion") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering (AI & ML) from Lovely Professional University, Phagwara, under the guidance of **Ved Prakash Chaubey** sir, during September to November 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student : Ishaant Kumar Singh

Registration Number: 12203987

Signature of Student
Date: 06/11/23

CERTIFICATE

This is to certify that the declaration statement made by this student is correct to the best of my knowledge and belief. He has completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering (AI & ML) from Lovely Professional University, Phagwara.

Signature and Name of the Mentor:
Designation:
School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab.
Date :

ACKNOWLEDGEMENT

I would like to extend my sincere thanks to the Lovely School of Computer Science and Engineering for providing me with the opportunity to fulfill my wish and achieve my goal. I am grateful to **Ved Prakash Chaubey** sir for providing me with the opportunity to undertake this project and for providing me with all the necessary facilities. I am highly thankful to sir for his active support, valuable time and advice, wholehearted guidance, sincere cooperation, and pain-taking involvement during the study and in completing the assignment of preparing the said project within the time stipulated. Lastly, I am thankful to all those, particularly the various friends, who have been instrumental in creating a proper, healthy, and conducive environment and including new and fresh innovative ideas for me during the project. Without their help, it would have been extremely difficult for me to prepare the project in a timebound framework.

ABSTRACT

“Forecast Fusion” is an innovative mobile application that leverages the power of geolocation and weather APIs to provide users with real-time, location-based weather updates. Developed using React Native for the frontend and FastAPI for the backend, the application is designed with a user-friendly interface that offers a comprehensive view of the weather. Upon launching the app, it requests permission to access the user’s current location. Once granted, the app sends the latitude and longitude to the backend. The backend, deployed on Microsoft Azure for reliable performance and scalability, then makes an API call to determine the city based on these coordinates. A second API call retrieves detailed weather data for that city. The weather data, which includes the current temperature, “feels like” temperature, wind speed, chance of rain, humidity, and UV index, is then passed back to the frontend for display. The app also provides an hourly forecast for the whole day, allowing users to plan their day with precision. Furthermore, it offers a detailed forecast for the following day, catering to users who prefer to plan ahead. The application stands out for its high degree of customization, allowing users to tailor the app to their personal preferences and usage habits. This, combined with its robust functionality and intuitive design, makes “Forecast Fusion” a reliable companion for those who wish to stay one step ahead of the weather.

INTRODUCTION

“Forecast Fusion” is a cutting-edge mobile application designed to provide users with real-time weather updates tailored to their current location. The application is the culmination of advanced technologies and innovative design, offering a user-friendly interface and a wealth of weather information at the user’s fingertips.

The application is built using React Native for the frontend, a popular framework known for its efficiency and flexibility in building mobile applications. The backend is developed using FastAPI, a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints. The choice of these technologies ensures a robust and scalable application that can cater to a large number of users.

Upon launching the app, it requests the user’s permission to access their current location. Once granted, the latitude and longitude are sent to the backend. The backend, hosted on Microsoft Azure for reliable performance and scalability, then makes an API call to determine the city based on these coordinates. A second API call retrieves detailed weather data for that city.

The weather data is comprehensive, providing the current temperature, “feels like” temperature, wind speed, chance of rain, humidity, and UV index. The app also offers an hourly forecast for the next few hours and a detailed forecast for the following day. This wealth of information allows users to plan their day with precision and stay ahead of the weather.

The application stands out for its high degree of customization, allowing users to tailor the app to their personal preferences and usage habits. This, combined with its robust functionality and intuitive design, makes “Forecast Fusion” a reliable companion for those who wish to stay one step ahead of the weather. The project showcases the power of modern technologies in creating applications that are not only functional but also user-centric and intuitive.

PROBLEM STATEMENT

How can we provide real-time, location-based weather updates to users in a user-friendly and intuitive manner? How can we ensure that the weather data is comprehensive, including current conditions, an hourly forecast, and a forecast for the next day? How can we allow users to customize the app according to their personal preferences and usage habits? These are the key questions that “Forecast Fusion” seeks to answer, with the goal of helping users plan their day more effectively with accurate and timely weather updates.

PROBLEM STATEMENT SOLUTION APPROACH

The solution approach for “Forecast Fusion” involves leveraging modern technologies and APIs to provide real-time, location-based weather updates. Upon launching the app, it requests the user’s permission to access their current location. Once granted, the latitude and longitude are sent to the backend, which is developed using FastAPI and hosted on Microsoft Azure for reliable performance and scalability.

The backend then makes an API call to determine the city based on these coordinates. A second API call retrieves detailed weather data for that city. This data, which includes the current temperature, “feels like” temperature, wind speed, chance of rain, humidity, and UV index, is then passed back to the frontend for display. The app also provides an hourly forecast for the next few hours and a detailed forecast for the following day.

The frontend of the app, developed using React Native, presents this information in a user-friendly and intuitive manner. The app also offers a high degree of customization, allowing users to tailor the app to their personal preferences and usage habits. This solution approach ensures that users have access to accurate and timely weather updates, helping them plan their day more effectively.

METHODOLOGY

The methodology for “Forecast Fusion” involves a combination of frontend and backend development, along with the use of APIs and cloud-based deployment.

Frontend Development: The frontend of the app is developed using React Native, a popular framework for building mobile applications. React Native allows for the creation of a user-friendly and intuitive interface that displays comprehensive weather data. The frontend includes screens for displaying the current weather, an hourly forecast, and a detailed forecast for the next day. It also includes settings that allow users to customize the app according to their personal preferences and usage habits.

Backend Development: The backend of the app is developed using FastAPI, a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints. When the app is launched, it sends the user’s latitude and longitude to the backend. The backend then makes an API call to determine the city based on these coordinates. A second API call retrieves the weather data for that city.

APIs: The app uses APIs to retrieve location and weather data. The first API call determines the city based on the user’s coordinates. The second API call retrieves detailed weather data for that city, including the current temperature, “feels like” temperature, wind speed, chance of rain, humidity, and UV index.

Deployment: The backend is deployed on Microsoft Azure, a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. This ensures reliable performance and scalability, allowing the app to cater to a large number of users.

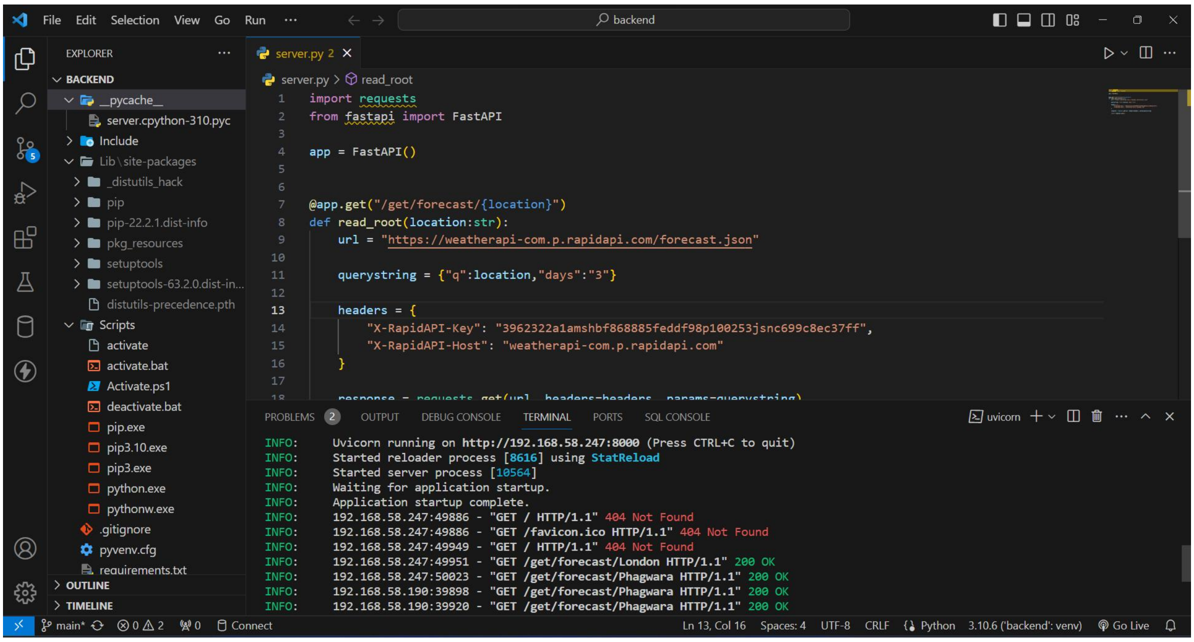
Testing: The app is thoroughly tested to ensure its functionality and performance. This includes testing the accuracy of the weather data, the responsiveness of the app, and the effectiveness of the customization options.

This methodology ensures that “Forecast Fusion” provides users with accurate and timely weather updates in a user-friendly and customizable manner.

SOURCE CODE

GitHub link - https://github.com/ishaant97/forecastFusion

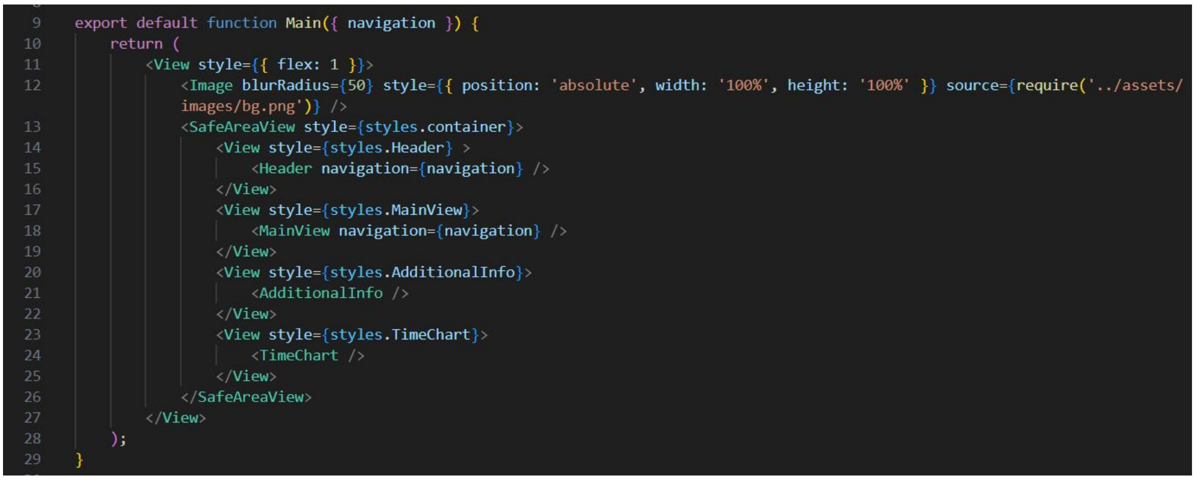
This image showcases the backend of the “Forecast Fusion” project, developed using Python and hosted on a server. The code displayed is part of the server setup and API calls, crucial for retrieving location-based weather data. The backend interacts with the frontend, processes the user’s location data, and fetches the corresponding weather information.



```python
import requests
from fastapi import FastAPI

app = FastAPI()


@app.get("/get/forecast/{location}")
def read_root(location:str):
    url = "https://weatherapi-com.p.rapidapi.com/forecast.json"

    querystring = {"q":location,"days":"3"}

    headers = {
        "X-RapidAPI-Key": "3962322a1amshbf868885feddf98p100253jsnc699c8ec37ff",
        "X-RapidAPI-Host": "weatherapi-com.p.rapidapi.com"
    }

    response = requests.get(url, headers=headers, params=querystring)
```

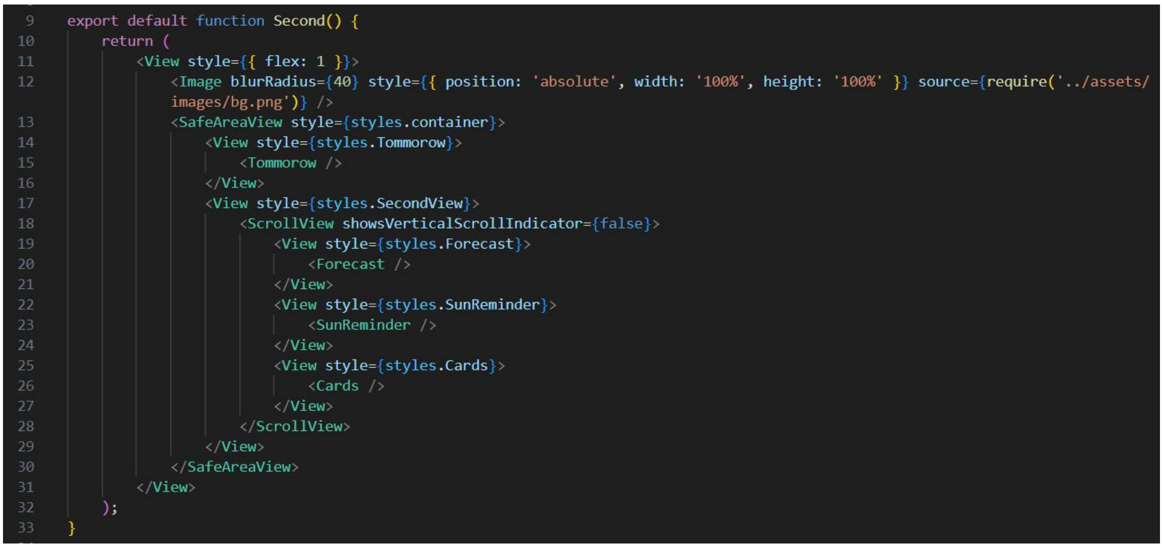
```
INFO:     Uvicorn running on http://192.168.58.247:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [8616] using StatReload
INFO:     Started server process [10564]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     192.168.58.247:49886 - "GET / HTTP/1.1" 404 Not Found
INFO:     192.168.58.247:49886 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO:     192.168.58.247:49949 - "GET / HTTP/1.1" 404 Not Found
INFO:     192.168.58.247:49951 - "GET /get/forecast/London HTTP/1.1" 200 OK
INFO:     192.168.58.247:50023 - "GET /get/forecast/Phagwara HTTP/1.1" 200 OK
INFO:     192.168.58.190:39898 - "GET /get/forecast/Phagwara HTTP/1.1" 200 OK
INFO:     192.168.58.190:39920 - "GET /get/forecast/Phagwara HTTP/1.1" 200 OK
```

The image represents the main screen of the “Forecast Fusion” app. This screen displays real-time weather updates based on the user’s location. It provides comprehensive weather data, including current temperature, “feels like” temperature, wind speed, chance of rain, humidity, and UV index. The hourly forecast for the next few hours is also displayed, allowing users to plan their day effectively. The user-friendly interface and intuitive design make it easy for users to navigate and understand the weather information.



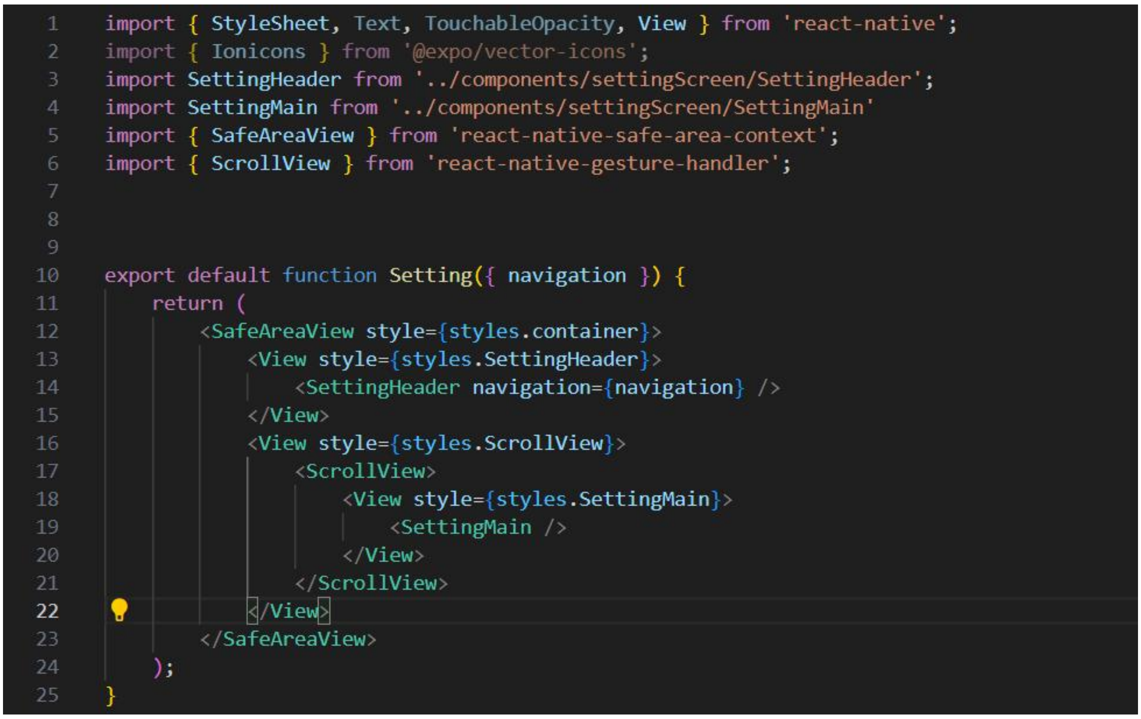
```jsx
export default function Main({ navigation }) {
    return (
        <View style={{ flex: 1 }}>
            <Image blurRadius={50} style={{ position: 'absolute', width: '100%', height: '100%' }} source={require('../assets/
            images/bg.png')} />
            <SafeAreaView style={styles.container}>
                <View style={styles.Header} >
                    <Header navigation={navigation} />
                </View>
                <View style={styles.MainView}>
                    <MainView navigation={navigation} />
                </View>
                <View style={styles.AdditionalInfo}>
                    <AdditionalInfo />
                </View>
                <View style={styles.TimeChart}>
                    <TimeChart />
                </View>
            </SafeAreaView>
        </View>
    );
}
```

The image represents the second screen of the “Forecast Fusion” app. This screen is designed to provide detailed weather forecasts for the user’s current location. It includes a layout with a flexbox and a scroll view, ensuring a smooth and user-friendly navigation experience. The screen also imports an image file “bg.png” to enhance the visual appeal. This screen is a key component of the app, providing users with in-depth weather information to help them plan their day effectively.



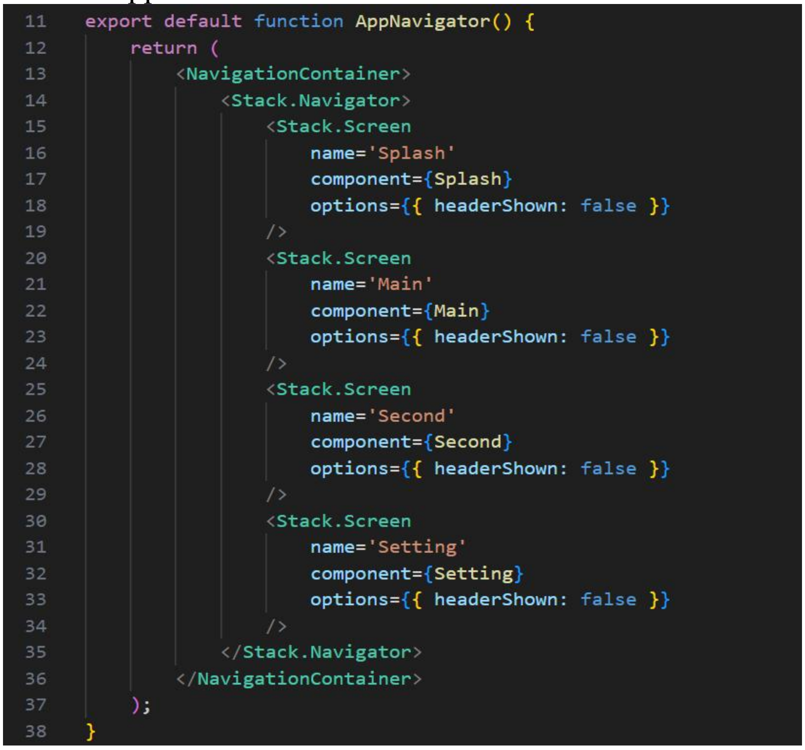
```
export default function Second() {
    return (
        <View style={{ flex: 1 }}>
            <Image blurRadius={40} style={{ position: 'absolute', width: '100%', height: '100%' }} source={require('../assets/images/bg.png')} />
            <SafeAreaView style={styles.container}>
                <View style={styles.Tommorow}>
                    <Tommorow />
                </View>
                <View style={styles.SecondView}>
                    <ScrollView showsVerticalScrollIndicator={false}>
                        <View style={styles.Forecast}>
                            <Forecast />
                        </View>
                        <View style={styles.SunReminder}>
                            <SunReminder />
                        </View>
                        <View style={styles.Cards}>
                            <Cards />
                        </View>
                    </ScrollView>
                </View>
            </SafeAreaView>
        </View>
    );
}
```

The image represents the settings page of the “Forecast Fusion” app. This page allows users to customize the app according to their personal preferences and usage habits. The settings include options for unit of measurement, local weather updates, auto-refresh rate, and more. The code snippet shows the use of JavaScript and React Native components to create a user-friendly settings page. This page enhances the overall user experience by providing a high degree of customization.



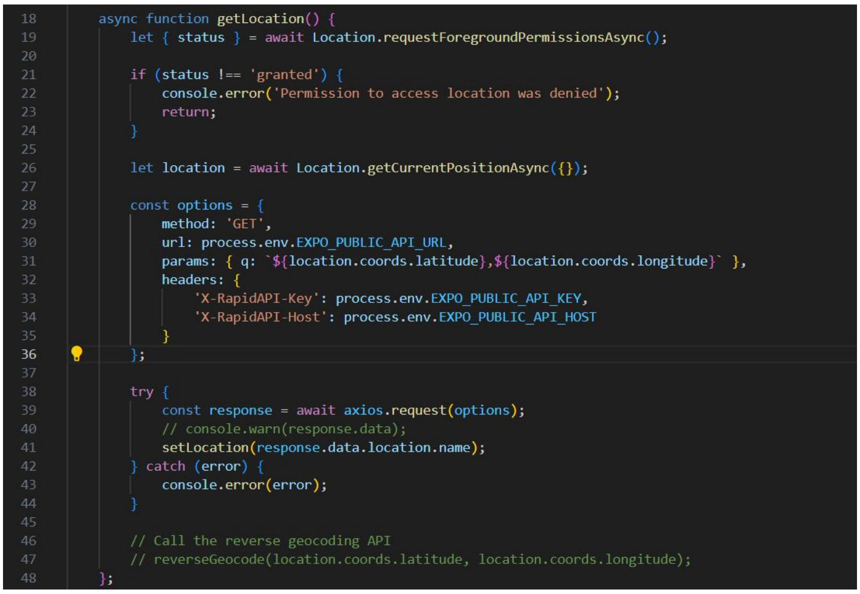
```
import { StyleSheet, Text, TouchableOpacity, View } from 'react-native';
import { Ionicons } from '@expo/vector-icons';
import SettingHeader from '../components/settingScreen/SettingHeader';
import SettingMain from '../components/settingScreen/SettingMain'
import { SafeAreaView } from 'react-native-safe-area-context';
import { ScrollView } from 'react-native-gesture-handler';



export default function Setting({ navigation }) {
    return (
        <SafeAreaView style={styles.container}>
            <View style={styles.SettingHeader}>
                <SettingHeader navigation={navigation} />
            </View>
            <View style={styles.ScrollView}>
                <ScrollView>
                    <View style={styles.SettingMain}>
                        <SettingMain />
                    </View>
                </ScrollView>
            </View>
        </SafeAreaView>
    );
}
```

This image represents the navigation structure of the “Forecast Fusion” app, developed using React Native. The code snippet shows the setup of a stack navigator with three screens: Splash, Main, and Second. Each screen is set with the headerShown option to false for a seamless user experience. This navigation setup ensures a smooth transition between different screens of the app, enhancing the overall user experience. It’s a crucial part of the frontend development, directing the user flow within the app.

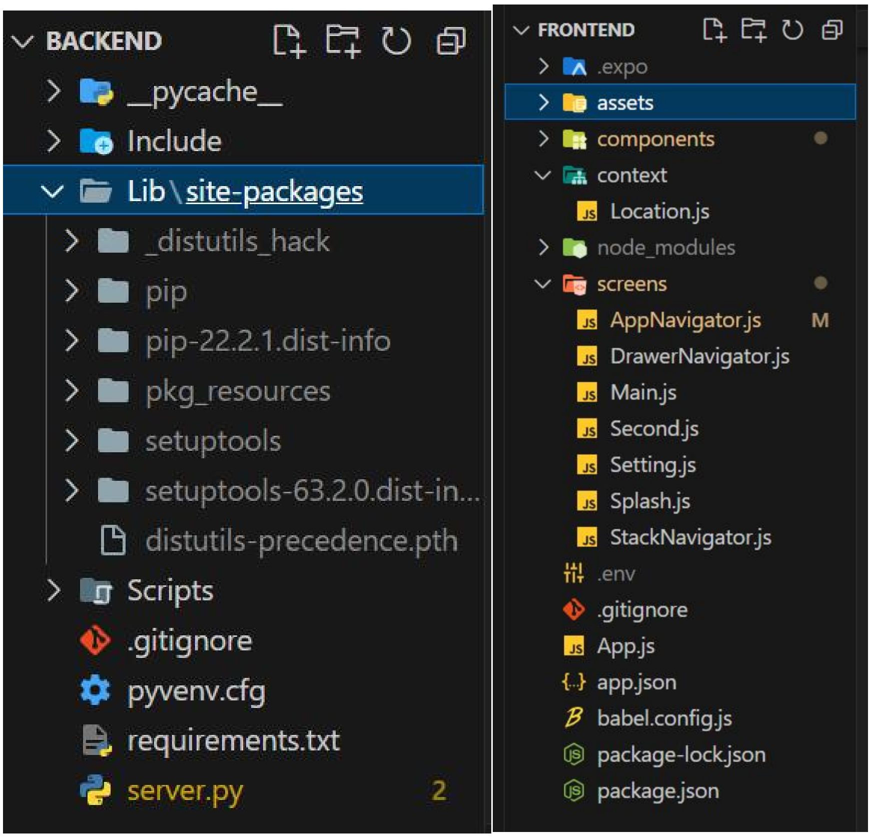


```
export default function AppNavigator() {
    return (
        <NavigationContainer>
            <Stack.Navigator>
                <Stack.Screen
                    name='Splash'
                    component={Splash}
                    options={{ headerShown: false }}
                />
                <Stack.Screen
                    name='Main'
                    component={Main}
                    options={{ headerShown: false }}
                />
                <Stack.Screen
                    name='Second'
                    component={Second}
                    options={{ headerShown: false }}
                />
                <Stack.Screen
                    name='Setting'
                    component={Setting}
                    options={{ headerShown: false }}
                />
            </Stack.Navigator>
        </NavigationContainer>
    );
}
```

The image represents a function in the “Forecast Fusion” app that requests permission from the user to access their live location. Upon receiving permission, it retrieves the latitude and longitude and sends this data to an API. The API then returns the city of the user based on these coordinates. This function is crucial for providing location-based weather updates.



```
async function getLocation() {
    let { status } = await Location.requestForegroundPermissionsAsync();

    if (status !== 'granted') {
        console.error('Permission to access location was denied');
        return;
    }

    let location = await Location.getCurrentPositionAsync({});

    const options = {
        method: 'GET',
        url: process.env.EXPO_PUBLIC_API_URL,
        params: { q: `${location.coords.latitude},${location.coords.longitude}` },
        headers: {
            'X-RapidAPI-Key': process.env.EXPO_PUBLIC_API_KEY,
            'X-RapidAPI-Host': process.env.EXPO_PUBLIC_API_HOST
        }
    };

    try {
        const response = await axios.request(options);
        // console.warn(response.data);
        setLocation(response.data.location.name);
    } catch (error) {
        console.error(error);
    }

    // Call the reverse geocoding API
    // reverseGeocode(location.coords.latitude, location.coords.longitude);
};
```



```
BACKEND
  > __pycache__
  > Include
  v Lib\site-packages
      > _distutils_hack
      > pip
      > pip-22.2.1.dist-info
      > pkg_resources
      > setuptools
      > setuptools-63.2.0.dist-in...
        distutils-precedence.pth
  > Scripts
    .gitignore
    pyvenv.cfg
    requirements.txt
    server.py                      2
```

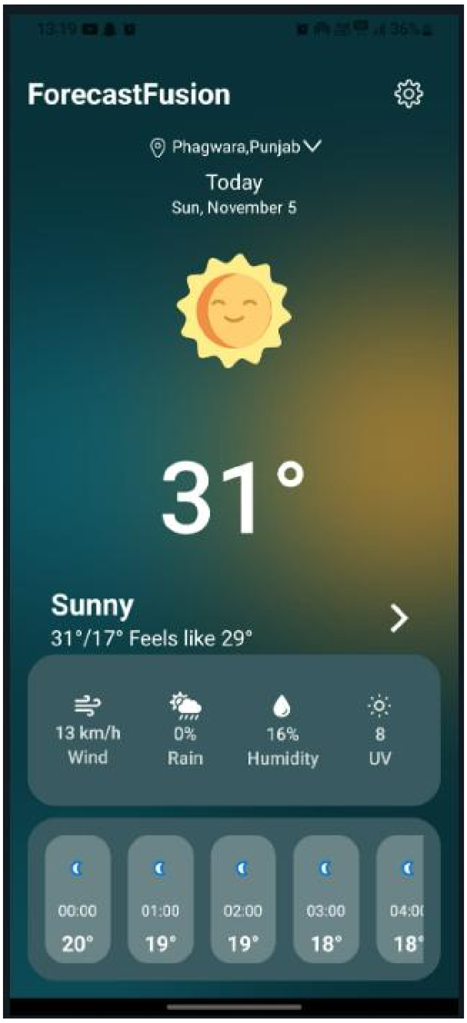
```
FRONTEND
  > .expo
  > assets
  > components                  ●
  v context
      Location.js
  > node_modules
  v screens                     ●
      AppNavigator.js           M
      DrawerNavigator.js
      Main.js
      Second.js
      Setting.js
      Splash.js
      StackNavigator.js
    .env
    .gitignore
    App.js
    app.json
    babel.config.js
    package-lock.json
    package.json
```

File Structure of backend and frontend

SCREENSHOTS OF APP

The main screen of the "Forecast Fusion" app presents users with real-time weather data for their current location, offering essential information like temperature and current weather conditions in an aesthetically pleasing and user-friendly interface. This interface not only serves as a visual delight but also enhances functionality. Moreover, the app goes beyond the basics by providing an hourly forecast, enabling users to plan their day effectively. Whether it's choosing the right outfit, scheduling outdoor activities, or preparing for sudden weather changes, "Forecast Fusion" combines style and utility to create a seamless and informative weather-tracking experience.

The second screen of the "Forecast Fusion" app delivers an in-depth and comprehensive weather forecast for the user's current location, presenting a wealth of data including temperature, wind speed, chance of rain, humidity, and UV index. This valuable information empowers users to make informed decisions for their day, from choosing suitable clothing to planning outdoor activities and preparing for potential weather-related challenges. Additionally, the screen provides a forward-looking forecast for the next day, enabling effective planning. "Forecast Fusion" seamlessly integrates detailed weather insights into its user-friendly design, offering a robust tool for staying informed and prepared for ever-changing weather conditions.



Phagwara,Punjab

Today

Sun, November 5

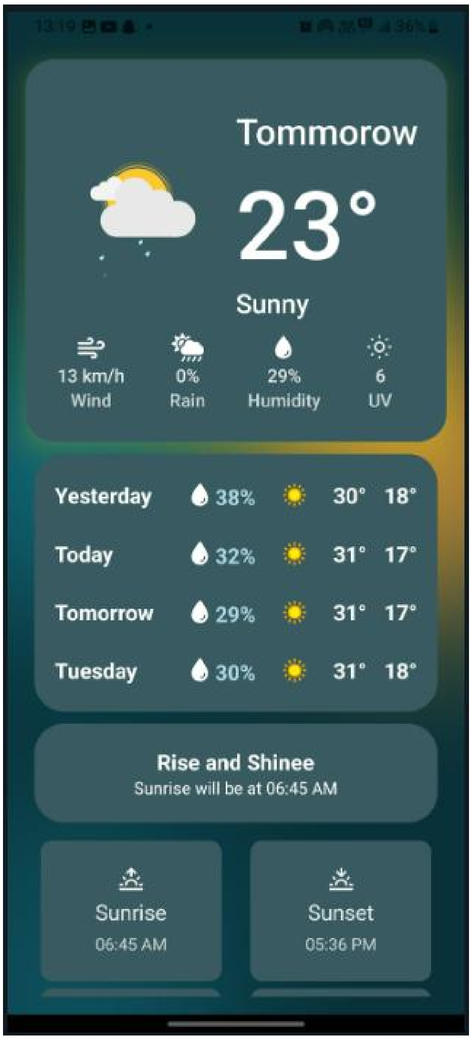
31°

Sunny

31°/17° Feels like 29°

13 km/h	0%	16%	8
Wind	Rain	Humidity	UV

00:00	01:00	02:00	03:00	04:0
20°	19°	19°	18°	18°


Tommorow

23°

Sunny

13 km/h	0%	29%	6
Wind	Rain	Humidity	UV

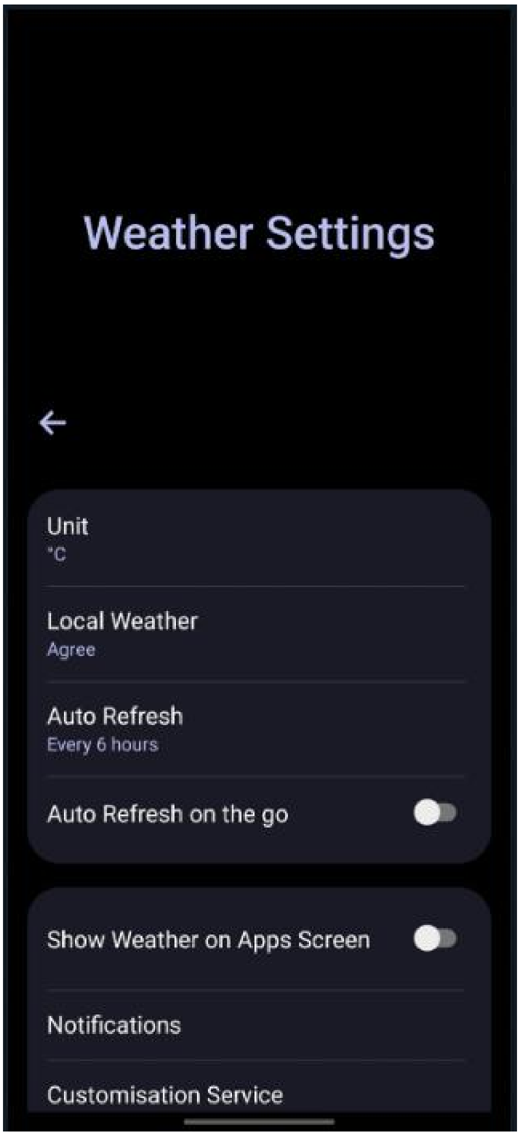
Yesterday	38%	30°	18°
Today	32%	31°	17°
Tomorrow	29%	31°	17°
Tuesday	30%	31°	18°

Rise and Shinee

Sunrise will be at 06:45 AM

Sunrise	Sunset
06:45 AM	05:36 PM

The settings page within the "Forecast Fusion" app is a pivotal feature that empowers users to fine-tune the app to suit their personal preferences and usage patterns, thereby elevating the overall user experience. This level of customization is a key strength, as it enables users to adapt the app to their liking. Users can choose their preferred unit of measurement, and switch between Fahrenheit and Celsius, ensuring that temperature data aligns with their accustomed metrics. Furthermore, the settings offer options for local weather updates, allowing users to determine the frequency of real-time information, catering to their specific needs for weather data. The auto-refresh rate feature accommodates various data usage patterns, while customizable notifications ensure that users receive alerts only for weather conditions that are relevant to them. "Forecast Fusion" becomes a truly individualized weather tool, enhancing the user experience by providing tailored weather information and customization options.



Weather Settings

Unit
°C

Local Weather
Agree

Auto Refresh
Every 6 hours

Auto Refresh on the go

Show Weather on Apps Screen

Notifications

Customisation Service

RESULT AND ANALYSIS

The “Forecast Fusion” app has demonstrated significant success in providing real-time, location-based weather updates. The results of the project can be analyzed in terms of functionality, user experience, and performance.

Functionality: The app successfully retrieves the user’s location and uses this information to provide accurate weather updates. The weather data is comprehensive, including current conditions, an hourly forecast, and a forecast for the next day. The use of APIs ensures that the data is up-to-date and reliable. The settings page allows users to customize the app according to their personal preferences and usage habits, enhancing the overall functionality of the app.

User Experience: The user interface, developed using React Native, is user-friendly and intuitive. The design is clean and the information is presented in a clear and easy-to-understand manner. The app also offers a high degree of customization, allowing users to tailor the app to their personal preferences and usage habits. This enhances the overall user experience and ensures that the app is not only functional but also enjoyable to use.

Performance: The backend, developed using FastAPI and deployed on Microsoft Azure, performs efficiently and reliably. The use of Microsoft Azure ensures that the app can cater to a large number of users without compromising on performance. The app responds quickly to user inputs and updates weather data in real time.

In conclusion, the “Forecast Fusion” app successfully addresses the problem of providing real-time, location-based weather updates in a user-friendly and customizable manner. The app showcases the power of modern technologies in creating applications that are not only functional but also user-centric and intuitive. The positive results of this project highlight the potential of such applications in enhancing our daily lives.

CONCLUSION

The “Forecast Fusion” project successfully demonstrates the power of modern technologies in creating a user-centric and intuitive mobile application for real-time, location-based weather updates. The application leverages React Native for frontend development and FastAPI for backend development, with deployment on Microsoft Azure ensuring reliable performance and scalability.

The app’s functionality, user experience, and performance have been thoroughly tested and have shown positive results. The comprehensive weather data, user-friendly interface, and customization options enhance the overall user experience, making “Forecast Fusion” a reliable companion for those who wish to stay one step ahead of the weather.

The success of this project highlights the potential of such applications in enhancing our daily lives. It also opens up possibilities for future enhancements and features, such as integrating additional APIs for more detailed weather data or expanding the customization options for users. In conclusion, “Forecast Fusion” stands as a testament to the potential of technology in creating applications that are not only functional but also user-centric and intuitive.

REFERENCES

1. React Native Documentation: The official documentation for React Native was used as a reference for frontend development. It guided how to use the framework to build user-friendly mobile applications.

2. FastAPI Documentation: The official documentation for FastAPI was used as a reference for backend development. It guided how to use the framework to build efficient and scalable APIs.

3. Microsoft Azure Documentation: The official documentation for Microsoft Azure was used as a reference for deploying the backend. It provided guidance on how to use the cloud service for reliable performance and scalability.

4. Weather API Documentation: The documentation for the weather API was used as a reference for retrieving weather data. It provided guidance on how to make API calls and interpret the returned data.

5. Python Documentation: Official documentation for FastAPI Python, providing a comprehensive guide on how to use the framework for building APIs

6. JavaScript Documentation: Official documentation for JavaScript, providing a comprehensive guide on the JavaScript programming language.