

**PYTHON PROJECT**  
**INT216**  
**FINAL PROJECT REPORT**



**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, whole-hearted 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.

<b>S. NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>1.</b>	<b>TITLE</b>	<b>1</b>
<b>2.</b>	<b>CERTIFICATE</b>	<b>2</b>
<b>3.</b>	<b>DECLARATION</b>	<b>3</b>
<b>4.</b>	<b>ACKNOWLEDGEMENT</b>	<b>4</b>
<b>5.</b>	<b>ABSTRACT</b>	<b>5</b>
<b>6.</b>	<b>INTRODUCTION</b>	<b>7</b>
<b>7.</b>	<b>PROBLEM STATEMENT</b>	<b>8</b>
<b>8.</b>	<b>PROBLEM STATEMENT</b> <b>SOLUTION APPROACH</b>	<b>8</b>
<b>9.</b>	<b>METHODOLOGY</b>	<b>9</b>
<b>10.</b>	<b>RESULT AND ANALYSIS</b>	<b>10</b>
<b>11.</b>	<b>CONCLUSION</b>	<b>11</b>
<b>12.</b>	<b>REFERENCES</b>	<b>12</b>

# 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.

## **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.