**Mini Project Report**

Weather App with Data Structures Submitted to

# Mohan Babu University, Tirupathi

in partial fulfillment of the requirements for the award of the degree of

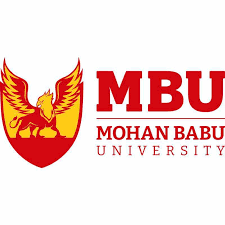
**BACHELOR OF TECHNOLOGY**

**IN**

**DATA SCIENCE**

*Submitted by*

|  |  |  |  |
| --- | --- | --- | --- |
| **C.Lokeshwari** | | | **22102A030160** |
|  | | |  |
|  |  |
| **K.Mourya Sri** | | | **22102A030198** |
|  | | |  |
|  | | |  |
| **M.Arshiya** | | | **22102A030221** |



Department of Data Science

**MOHAN BABU UNIVERSITY**

2023-2024

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **TABLE OF CONTENT** | **Page No** |
| **INTRODUCTION** | **1** |
| **PROBLEM DEFINITION** | **2** |
| **ALGORITHM** | **3** |
| PROGRAM | **4** |
| **CONCLUSION** | **5** |

# Institute Vision and Mission

**VISION**

To be one of the Nation’s premier University by achieving the highest order of excellence in Teaching and Research.

**MISSION**

To foster intellectual curiosity, pursuit and dissemination of knowledge.

To explore students’ potential through academic freedom and integrity. To promote technical mastery and nurture skilled professionals to face competition in ever increasing complex world.

**DEPARTMENT OF DATASCIENCE**

**VISION**

To become a nationally recognized quality education centre in the domain of Computer Science and Information Technology through teaching, training, learning, research and consultancy.

**MISSION**

The Department offers undergraduate program in Data Science to produce high quality information technologists and software engineers by disseminating knowledge through contemporary curriculum, competent faculty and adopting effective teaching-learning methodologies.

Igniting passion among students for research and innovation by exposing them to real time systems and problems

Developing technical and life skills in diverse community of students with modern training methods to solve problems in Software Industry.

Inculcating values to practice engineering in adherence to code of ethics in multicultural and multi discipline teams.

## PROGRAM EDUCATIONAL OBJECTIVES

After few years of graduation, the graduates of B. Tech. (DS) Program will be:

1. Enrolled or completed higher education in the core or allied areas of Computer . Science and Information Technology or management.
2. Successful entrepreneurial or technical career in the core or allied areas of . Computer Science and Information Technology.
3. Continued to learn and to adapt to the world of constantly evolving technologies in . the core or allied areas of Computer Science and Information Technology.

## PROGRAM OUTCOMES

On successful completion of the Program, the graduates of B. Tech. (DS) Program will be able to:

1. Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## PROGRAM SPECIFIC OUTCOMES

On successful completion of the program, the graduates of B.Tech. (DS) program will be able to:

**PSO1:** Design and develop database systems, apply data analytics techniques, and

use advanced databases for data storage, processing and retrieval.

|  |  |
| --- | --- |
| **PSO2:** | Apply network security techniques and tools for the development of highly secure systems. |
| **PSO3:** | Analyze, design and develop efficient algorithms and software applications to deploy in secure environment to support contemporary services using programming languages, tools and technologies. |
| **PSO4:** | Apply concepts of computer vision and artificial intelligent for the |

development of efficient intelligent systems and applications.

# INTRODUCTION

In today's fast-paced world, access to accurate and up-to-date weather information is not just a convenience but a necessity. Weather apps have become an integral part of our daily lives, helping us plan our activities, stay safe during severe weather events, and make informed decisions. This project aims to take weather forecasting and information dissemination to the next level by leveraging advanced data structures, specifically heaps, to deliver the most precise and timely weather data to users.

Traditional weather apps rely on simple data storage and retrieval methods, often presenting users with data in a linear fashion. However, by incorporating data structures like heaps, we can revolutionize how weather information is organized and presented. This project introduces a novel approach to weather forecasting and alerts, ensuring that users receive the most critical and relevant information promptly.

Heaps, a fundamental data structure in computer science, play a pivotal role in this endeavor. We utilize min-heaps to prioritize and display weather alerts, ensuring that users are immediately informed about severe weather conditions in their area. Simultaneously, max-heaps are employed to highlight the most critical weather forecasts, allowing users to plan their activities with precision.

This project combines the power of modern technology, data science, and intuitive user interface design to create a weather app that goes beyond the basics. By implementing advanced data structures, we aim to provide a superior user experience, helping individuals and communities make well-informed decisions in the face of changing weather conditions. From tracking daily weather forecasts to receiving immediate notifications about critical weather events, this app is designed to be a reliable and indispensable tool in your everyday life.

As you delve deeper into this project, you'll gain insights into the world of data structures, API integration, user interface design, and the intricacies of managing and presenting weather data. By the end of this endeavor, you'll have not only a functional weather app but also a valuable learning experience in software development and data science. Welcome to the Weather App with Data Structures Project, where innovation and weather information converge to enhance the way we interact with the world around us.

# 

# PROBLEM DEFINITION

The fundamental problem we aim to address is the inefficiency and limited prioritization of weather data in existing weather applications. Our project seeks to revolutionize the way weather information is managed and presented, recognizing that not all weather data is of equal importance.

**ALGORITHMS**

* Create a class named WeatherAlert to represent weather alerts.
  + WeatherAlert has attributes: severity (an integer) and message (a string).
  + Implement the Comparable interface to enable sorting based on severity.
* Create a class named WeatherAlertApp.
  + WeatherAlertApp has a PriorityQueue of WeatherAlerts to manage weather alerts.
* Initialize the PriorityQueue in the constructor of WeatherAlertApp.
* Implement methods for WeatherAlertApp:
  + addAlert(severity, message):
  + Create a new WeatherAlert with the provided severity and message.
  + Add the new alert to the PriorityQueue.
  + getTopAlert():
  + Check if the PriorityQueue is not empty.
  + If not empty, retrieve and return the message of the top (most severe) alert.
  + If the PriorityQueue is empty, return a message indicating no alerts.
* In the main method of WeatherAlertApp:
  + Create an instance of WeatherAlertApp.
  + Add sample weather alerts using the addAlert method.
  + Retrieve and print the top alert using the getTopAlert method.
* Compile and run the WeatherAlertApp class.

**PROGRAM**

import java.util.PriorityQueue;

class WeatherAlert implements Comparable<WeatherAlert> {

private int severity;

private String message;

public WeatherAlert(int severity, String message) {

this.severity = severity;

this.message = message;

}

@Override

public int compareTo(WeatherAlert other) {

// Compare alerts based on severity

return Integer.compare(other.severity, this.severity);

}

public String getMessage() {

return message;

}

}

class WeatherAlertApp {

private PriorityQueue<WeatherAlert> alertQueue;

public WeatherAlertApp() {

alertQueue = new PriorityQueue<>();

}

public void addAlert(int severity, String message) {

WeatherAlert alert = new WeatherAlert(severity, message);

alertQueue.add(alert);

}

public String getTopAlert() {

if (!alertQueue.isEmpty()) {

return alertQueue.peek().getMessage();

} else {

return "No alerts";

}

}

public static void main(String[] args) {

WeatherAlertApp app = new WeatherAlertApp();

// Adding sample alerts

app.addAlert(3, "High winds expected");

app.addAlert(5, "Heavy rain and flooding");

app.addAlert(2, "Partly cloudy with a chance of showers");

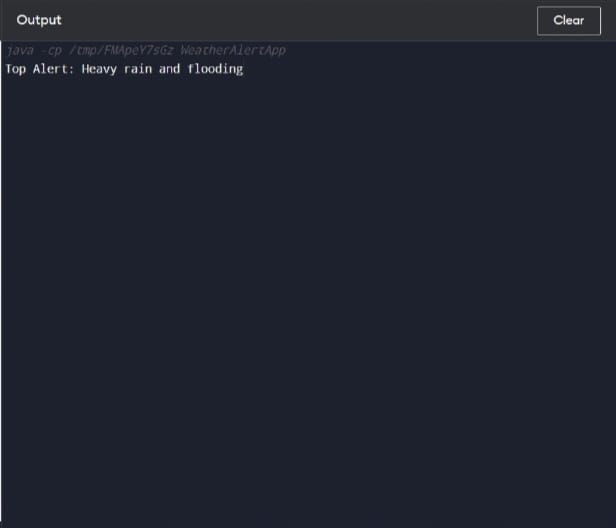
// Retrieving and printing the top alert

System.out.println("Top Alert: " + app.getTopAlert());

}

}

**OUTPUT**



**CONCLUSION**

In this project, we designed and implemented a Java-based Weather Alert Application that uses a min-heap to efficiently manage and prioritize weather alerts. The application demonstrated how to create and use a priority queue to handle alerts of varying severities, ensuring that the most critical alerts are processed first.