SKYVIEW-A REAL TIME WEATHER FORECAST

A MINI PROJECT REPORT

Submitted by

NANDITHA N 220701182

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

AUTONOMOUS, CHENNAI

NOV/DEC,2024

BONAFIDE CERTIFICATE

Certified that this project report **" SKYVIEW-A REAL TIME WEATHER FORECAST"** is the bonafide work of **" NANDITHA N (220701182)"** who carried out the project under my supervision.

SIGNATURE

Mrs. JANANEE V,

Assistant Professor,

Computer Science & Engineering

Rajalakshmi EngineeringCollege

Thandalam, Chennai -602105.

.

Submitted for the End semester practical examination to be held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I express my sincere thanks to my beloved and honourable chairman **MR.S.MEGANATHAN** and the chairperson **DR .M .THANGAM MEGANATHAN** for their timely support and encouragement.

I am greatly indebted to my respected and honourable principal **Dr. S. N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by my head of the department **Dr. P. KUMAR,** and my Academic Head **Dr. R.SABITHA,** for being ever supporting force during my project work.

I also extend my sincere and hearty thanks to my internal guide **Mrs. JANANEE V** for her valuable guidance and motivation during the completion of this project.

My sincere thanks to my family members, friends and other staff members of Computer Science and Engineering.

NANDITHA N - 220701182

ABSTRACT

Skyview is a user-centric, real-time weather forecasting platform designed to offer personalized weather updates and insights. Upon signing into their individual accounts, users are navigated to a tailored dashboard that presents the current weather forecast for their specific location. In addition to the present conditions, the dashboard extends its functionality by providing detailed forecasts for the next 24 hours and 7 days, as well as historical weather data for the previous 2 days.

Skyview also enables users to search and access weather forecasts for any location of their choice. Beyond just forecasts, the platform delivers weather-dependent recommendations that guide users on activities, dressing, and other suggestions based on the weather conditions at hand.

One of Skyview's key features is the ability to set weather alert messages, which can be delivered via email. These alerts can be customized for the user's current location or any specified location, giving flexibility to stay informed of changing weather conditions. Users have complete control over their alerts, with the option to cancel previously set alerts. Additionally, the platform provides an activity log, where users can track their alert setting and cancellation activities, ensuring full transparency and ease of use.

Skyview aims to enhance user experience by offering not only precise weather updates but also personalized tools to stay prepared and informed.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Skyview is a dynamic, real-time weather forecasting website designed to provide users with personalized weather updates and insights. By creating individual accounts, users gain access to a comprehensive dashboard that displays weather information tailored to their current location. This includes real-time updates, 24-hour forecasts, a 7-day weather outlook, and a summary of the previous 2 days' weather data. In addition, Skyview offers a search functionality, allowing users to retrieve weather forecasts for any location of their choice. The platform enhances user experience by delivering weather-dependent recommendations based on current conditions, helping users make informed decisions. Furthermore, Skyview allows users to set customizable weather alerts via email for either their current or a specific location, and offers the flexibility to cancel these alerts at any time. Users can also review their alert activity history in a dedicated log, ensuring a seamless and organized user experience.

1.2 SCOPE OF THE WORK

The scope of the "Skyview" project involves the development of a real-time weather forecasting website where users can create individual accounts and access a personalized dashboard displaying weather forecasts tailored to their current location. The system will provide detailed forecasts, including 24-hour, 7-day, and previous 2-day weather data, and allow users to search for weather information for specific locations. Additionally, the dashboard will feature weather-dependent recommendations and give users the ability to set, manage,

and cancel weather alerts via email for selected locations. Users can also track their alert activity through an integrated activity log.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of the "Skyview" project is to provide users with a personalized and comprehensive weather forecasting platform that offers real-time, location-specific weather updates. By leveraging user accounts, the system allows users to access tailored weather information, set up customized weather alerts, and receive weather-dependent recommendations, all while offering a seamless experience to manage and track their weather-related activities.

The key objectives include:

- Develop a user-friendly platform that provide real-time weather forecasts, including current conditions, 24-hour, 7-day, and past 2-day forecasts.
- Enable users to search for weather information in specific locations aside from their current location.
- Offer weather-dependent recommendations based on forecasted weather conditions.
- Implement a system for users to set, manage, and cancel email alerts for weather conditions at their current or selected locations.
- Create an activity log that tracks users' alert-setting actions and allows them to review their past alert activity.
- Ensure users can cancel or modify their weather alerts seamlessly through the platform.

CHAPTER 2
SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

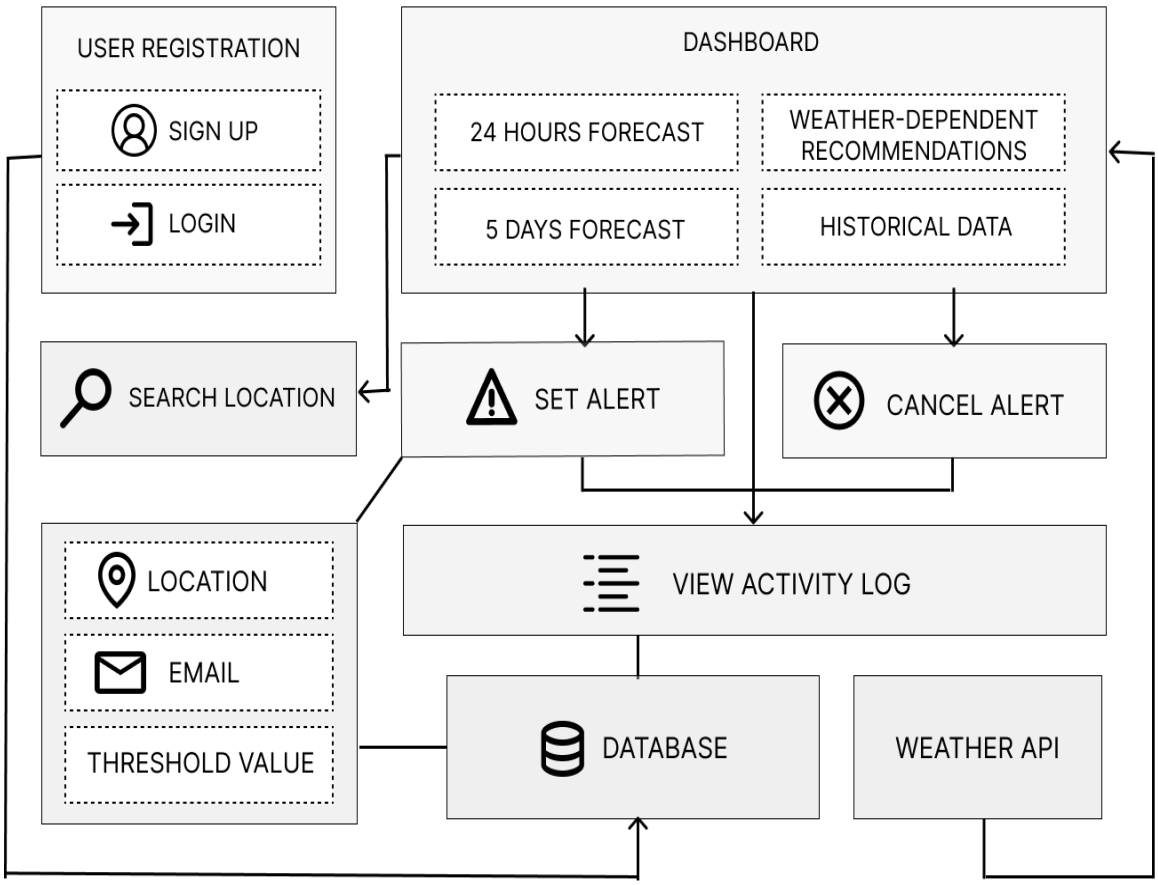
- **Processor:** Minimum 2.5 GHz dual-core processor (Intel Core i5 or equivalent)
- **RAM:** 8 GB or higher (16 GB recommended for optimal performance)
- **Storage:** 500 GB SSD or higher (preferably with cloud storage options)
- **Network:** High-speed internet connection (minimum 100 Mbps for handling multiple requests)
- **Operating System:** Linux or Windows Server 2019 for hosting the web application
- **Backup System:** Cloud or external storage backup solution for data redundancy and recovery

2.2 SOFTWARE SPECIFICATIONS

- **Front-end:** HTML5, CSS3, Javascript
- **Back-end:** PHP
- **Database:** Mysql
- **API:** Weather API
- **Version control:** Git , Github
- **Development tools:** Visual studio code

CHAPTER 3

ARCHITECTURE DIAGRAM



CHAPTER 4
MODULAR DESCRIPTION

MODULES

- User Sign up and Login Module
- Dashboard Module
    1. 24 hours Weather Forecast
    2. 7 days Weather Forecast
    3. Historical Weather Data
    4. Weather-dependent Recommendations
    5. Weather Forecast for search locations
- Set Alert Module
- Cancel Alert Module
- Activity Log Module

1. User Sign Up and Login Module

This module allows users to create individual accounts and securely log into the Skyview platform. Users provide necessary credentials such as email and password for registration. Once signed up, they can log in to access the personalized features of the website. The system ensures secure authentication and data protection, with options for password recovery and account management.

2. Dashboard Module

Upon logging in, users are directed to a personalized dashboard that provides real-time weather data tailored to their location. This module is divided into several key components:

- **24 Hours Weather Forecast:** Displays hourly weather conditions for the next 24 hours, including temperature, precipitation, humidity, and wind speed, giving users detailed, short-term weather insights.

- **7 Days Weather Forecast:** Offers an extended weather forecast for the upcoming seven days, including daily highs, lows, and weather summaries to help users plan ahead.

- **Historical Weather Data:** Provides weather data for the previous two days, enabling users to view past weather conditions and compare trends.

- **Weather-dependent Recommendations:** Based on the current and forecasted weather conditions, users receive suggestions on activities, clothing, and precautions tailored to the forecast.

- **Weather Forecast for Search Locations:** Users can search for weather data from different locations apart from their current one. This enables them to check conditions in other cities or places of interest, ensuring comprehensive forecasting options.

3. Set Alert Module

This module enables users to set weather alerts for specific conditions. Users can define thresholds for various weather elements (e.g., temperature, rain, wind) and choose a location (current or specific) to receive email notifications when those conditions are met. This feature is highly customizable, ensuring users stay informed about weather changes important to them.

4. Cancel Alert Module

This module allows users to manage their weather alerts by cancelling previously set alerts. Users can view all active alerts and select which ones to cancel, giving them control over their notifications. The system ensures easy alert management and prevents unwanted or outdated notifications.

5. Activity Log Module

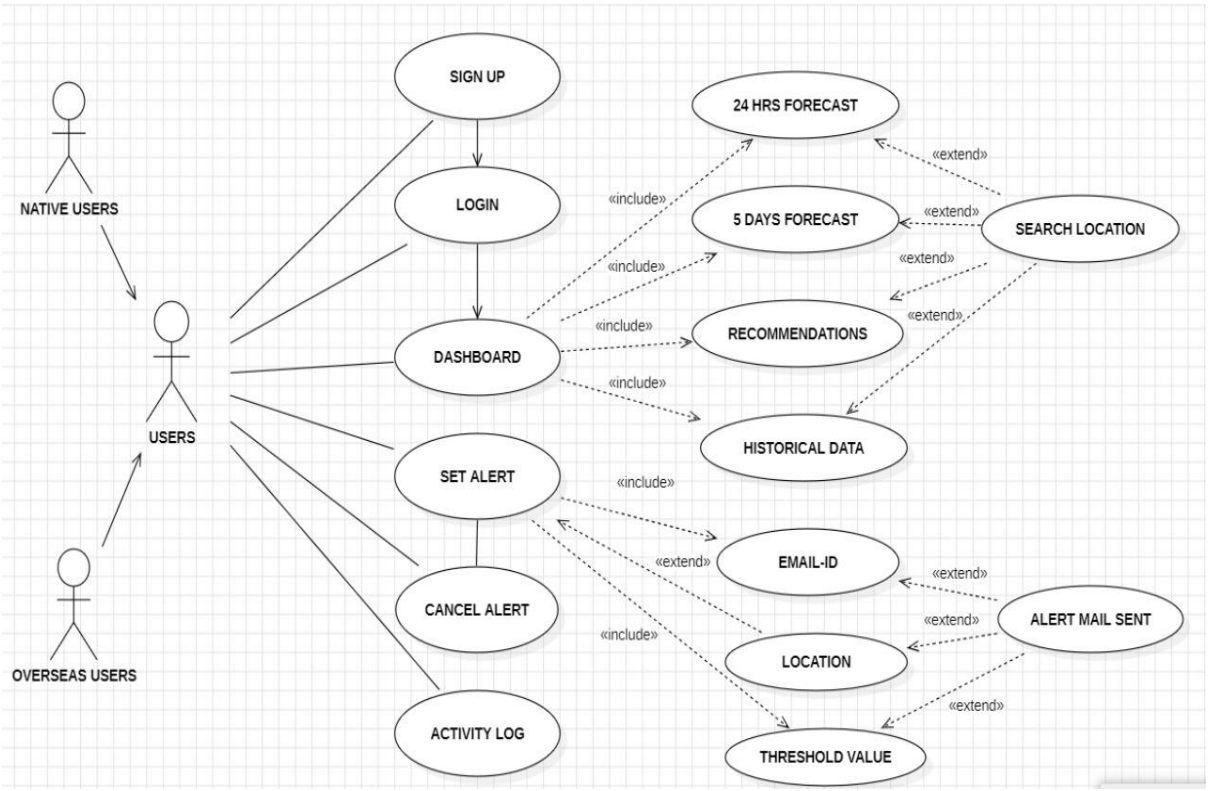
The activity log records all actions related to the user's alerts, such as when an alert was set, modified, or cancelled. This module helps users keep track of their alert-related activities and serves as a reference for managing their notifications. The activity log provides transparency and allows users to review their past actions within the platform.

These modules together form a comprehensive weather forecasting system that is user-friendly, customizable, and designed to provide timely weather insights.

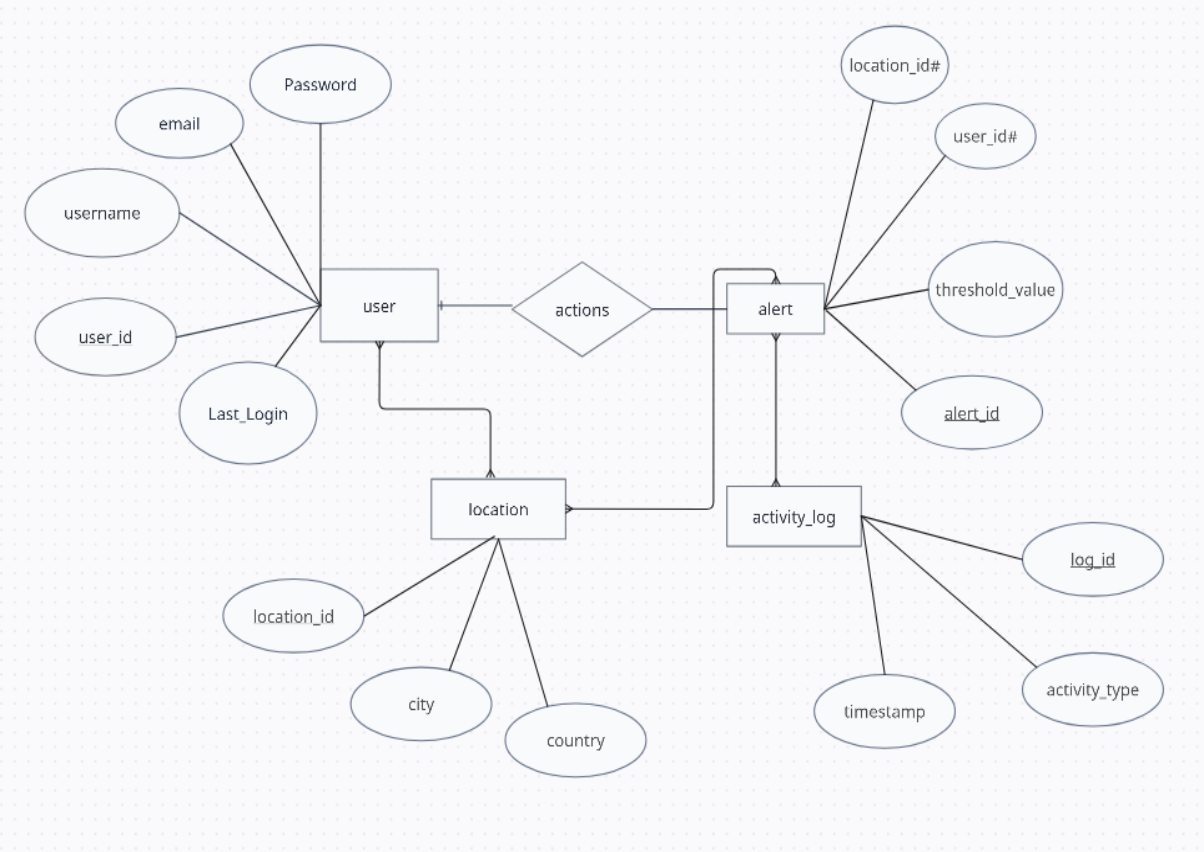
CHAPTER 5

SYSTEM DESIGN

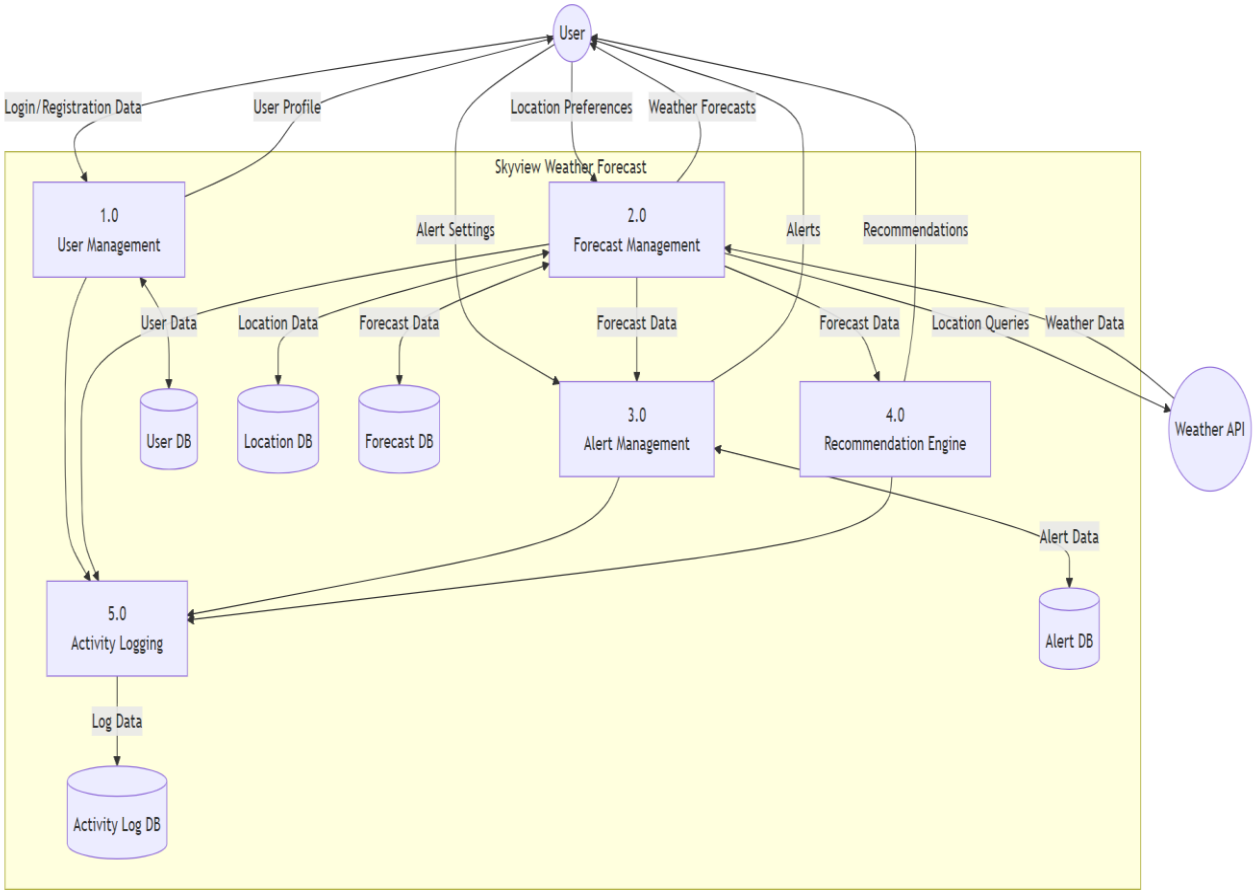
5.1 USE CASE DIAGRAM



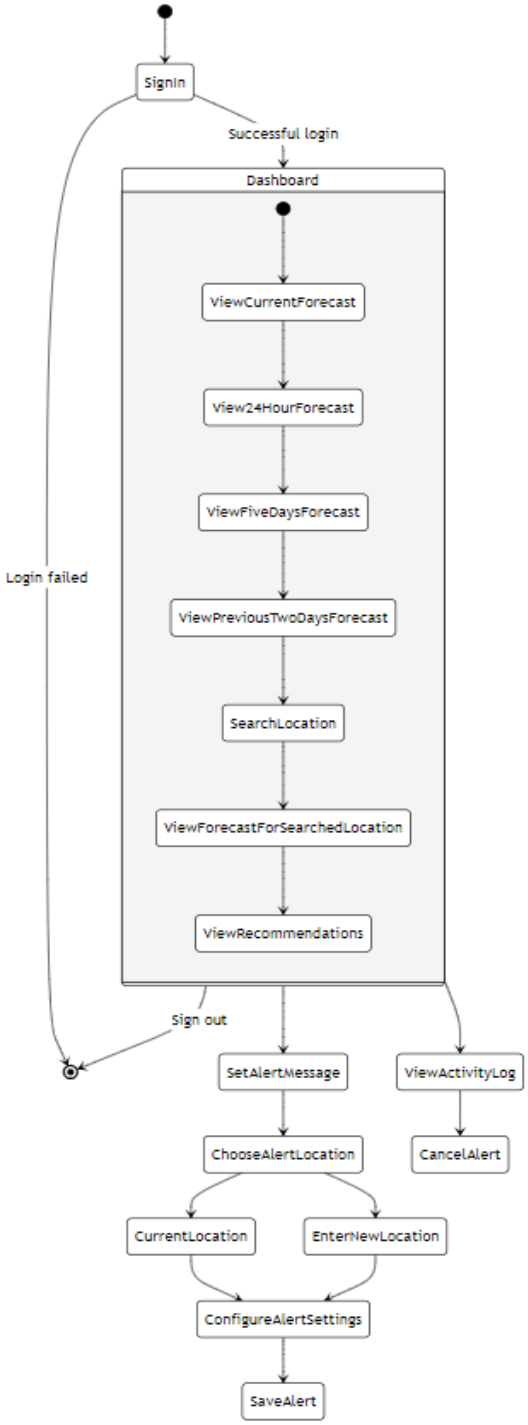
5.2 ER DIAGRAM



5.3 DATA FLOW DIAGRAM



5.4 ACTIVITY DIAGRAM



CHAPTER 6
SAMPLE CODING

FRONTPAGE.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Skyview</title>
<link rel="stylesheet" href="frontpagestyle.css">
</head>
<!-- Loading Screen -->
<div id="loading-screen">
<img src="loading.gif" alt="Loading" id="loading-gif">
</div>
<!-- Main Content (Initially Hidden) -->
<div id="main-content" style="display: none;">
<h1 id="animated-title"></h1>
<h3>A Real-time Weather Forecast</h3>
</div>
<nav class="navbar">
```

```
<div class="navbar-title">
<span class="skyview-icon">Skyview</span>
</div>
<div class="navbar-links">
<a href="#" class="nav-link" onclick="setActiveLink(event)">About Us</a>
<a href="signup.php" class="nav-link" onclick="setActiveLink(event)">
 Sign Up</a>
<a href="login.php" class="nav-link"
onclick="setActiveLink(event)">Login</a>
 </div>
 </nav>
 <div id="button-container">
 <button class="custom-btn"  type=
 "button"   onclick="window.location.href='login.php'">Login</button>
  <button class="custom-btn"  type=
  "button" onclick="window.location.href='signup.php'">Sign Up</button>
  </div>
  </div>
  <script src="frontpagescript.js"></script>
  </body>
  </html>
```

SIGNUP.PHP

```
<?php
include 'db.php';
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_BCRYPT);
$stmt = $conn->prepare("INSERT INTO users (username, password) VALUES (?, ?)");
 $stmt->bind_param("ss", $username, $password);
 if ($stmt->execute()) {
  echo "Sign up successful! You can now <a href='login.php'>log in</a>.";
   }
  else {
   echo "Error: " . $conn->error;
  }
  $stmt->close(); // Close the statement
 }
 ?>



<!DOCTYPE html>
```

```
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Skyview - Sign Up</title>
  <link rel="stylesheet" href="signupstyle.css">
  <link href='https://unpkg.com/boxicons@
    2.1.4/css/boxicons.min.css'   rel='stylesheet'>
  </head>
  <body id="background">
  <div class="wrapper">
  <form action="" method="POST">
  <h1>Sign Up</h1>
  <div class="input-box">
  <input type="text" name="username" placeholder="Username" required>
  <i class='bx bxs-user'></i>
   </div>
  <div class="input-box">
   <input type="password" name="password" placeholder=
  "Password"    required>
    <i class='bx bxs-lock-alt' ></i>
      </div>
```

```
    <button type="submit" class="btn">Sign Up</button>
    <div class="register-link">
     <p>Already have an account? <a href="login.php">Login</a></p>
    </div>
    </form>
    </div>
    <script>
   const gifs = [
    'snow.gif',
    'rain.gif',
    'sunny.gif',
    'autumn.gif'
   ];
   let currentGifIndex = 0;
   function changeBackground() {
   document.getElementById('background').style
   .backgroundImage= `url(${gifs[currentGifIndex]})`;
    currentGifIndex = (currentGifIndex + 1) % gifs.length
    setInterval(changeBackground, 2000);
   changeBackground();
    </script>
</body>
```

```
</html>
```

LOGIN.PHP

```
<?php
include 'db.php';
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
 $username = $_POST['username'];
 $password = $_POST['password'];
 $sql = "SELECT * FROM users WHERE username='$username'";
 $result = $conn->query($sql);
 if ($result->num_rows > 0) {
 $row = $result->fetch_assoc();
 if (password_verify($password, $row['password'])) {
 $_SESSION['user_id'] = $row['id'];
 $_SESSION['username'] = $username;
 header('Location: dashboard.php');
 exit();
 } else {
  echo "Invalid username or password.";
  }
} else {
```

```
echo "Invalid username or password.";
}
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Skyview-Login</title>
<link rel="stylesheet" href="loginstyle.css">
 <link href='https://unpkg.com/boxicons@2.1.4/css/
  boxicons.min.css' rel='stylesheet'>
</head>
<body id="background">
<div class="wrapper">
<form action="login.php" method="POST">
<h1>Login</h1>
<div class="input-box">
<input type="text" name="username" placeholder="Username" required>
<i class='bx bxs-user'></i>
  </div>
```

```
<div class="input-box">
<input type="password" name="password" placeholder="Password" required>
<i class='bx bxs-lock-alt' ></i>
</div>
<div class="remember-forgot">
<label><input type="checkbox">Remember Me</label>
<a href="#">Forgot Password</a>
</div>
<button type="submit" class="btn">Login</button>
<div class="register-link">
<p>Dont have an account?  <a href="signup.php">Sign Up</a></p>
</div>
</form>
</div>
</body>
</html>
```

DASHBOARD.PHP

```
<?php
session_start();
if (!isset($_SESSION['username'])) {
header('Location: login.php');
```

```
 exit();
}
$user_id = $_SESSION['user_id'];
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Weather Dashboard</title>
<link rel="stylesheet" href="dashboardstyle.css">
 </head>
 <nav class="navbar">
<form id="search-form" class="search-bar">
<input type="text" id="search-input" placeholder=
 "Search location..."   required/>
<button type="submit">Search</button>
 </form>
 <h2 id="location">Loading location...</h2>
 <div class="weather-display">
 <img id="weather-icon" src="" alt="Weather Icon" />
  <h3 id="weather-status">Loading weather...</h3>
```

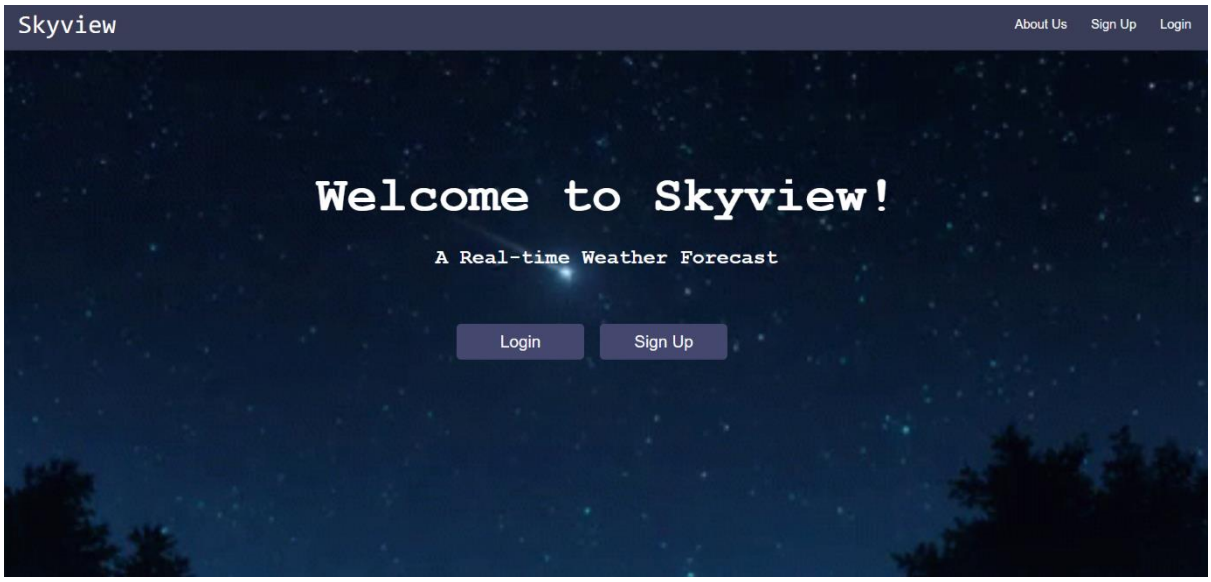
```
</div>
<h1 id="temperature">--°C</h1>
 <div id="datetime"></div>
<ul>
<li><a href="#section1">24 hrs forecast</a></li
<li><a href="#section2">5 days forecast</a></li>
<li><a href="#section3">Recommendations</li>
</ul>
</nav>
<nav class="navbar1">
< div class="navbar-title">
 <span class="hello">Hello
 <?php echo    $_SESSION['username']?>! 
Welcome to  Skyview:)</span>
</div>
<div class="navbar-links"
<a href="dashboard.php" class=
"nav-link"   onclick="setActiveLink(event)">Dashboard</a>
 <a href="alert.php" class=
 "nav-link" onclick="setActiveLink(event)">Set Alert</a>
  <a href="logout.php" class=
   "nav-link" onclick="setActiveLink(event)">Logout</a>
```

```
</div>
</nav>
<section id="section1">
<div class="hourly-weather">
 <h2 style="color: white;">24 Hours</h2><br>
 <div class="hourly-forecast"></div
 </div>
</section>
<section id="section2">
<div class="daily-weather">
 <h2 style="color: white;">5 days</h2><br>
 <div class="five-day-forecast"></div
 </div>
 </section>
 <div id="recommendation-container" class=
 "recommendation-container"></div>
 <script src="dashboardscript.js"></script>
 </body>
 </html>
```

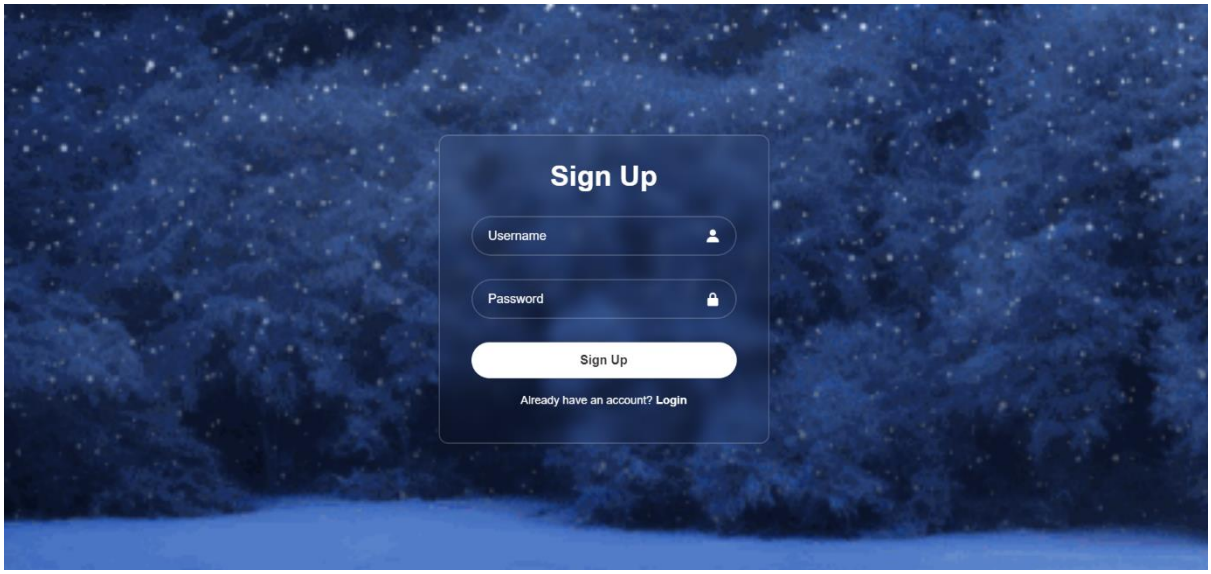
CHAPTER 7

SCREENSHOTS

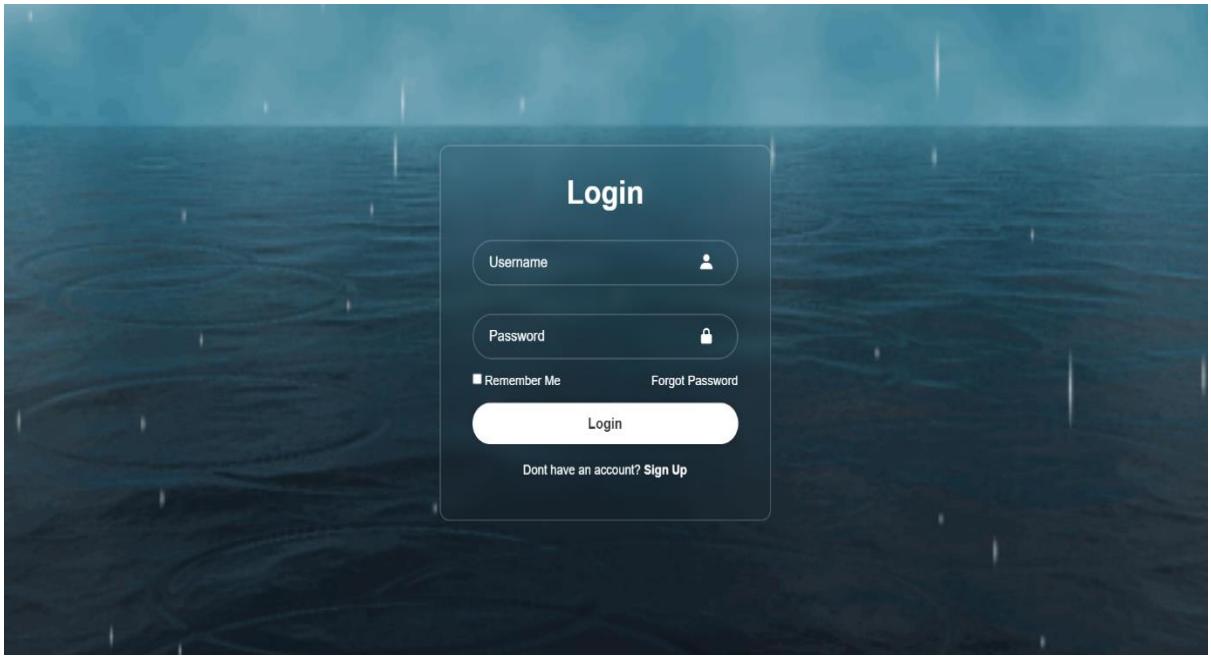
1. FRONT PAGE



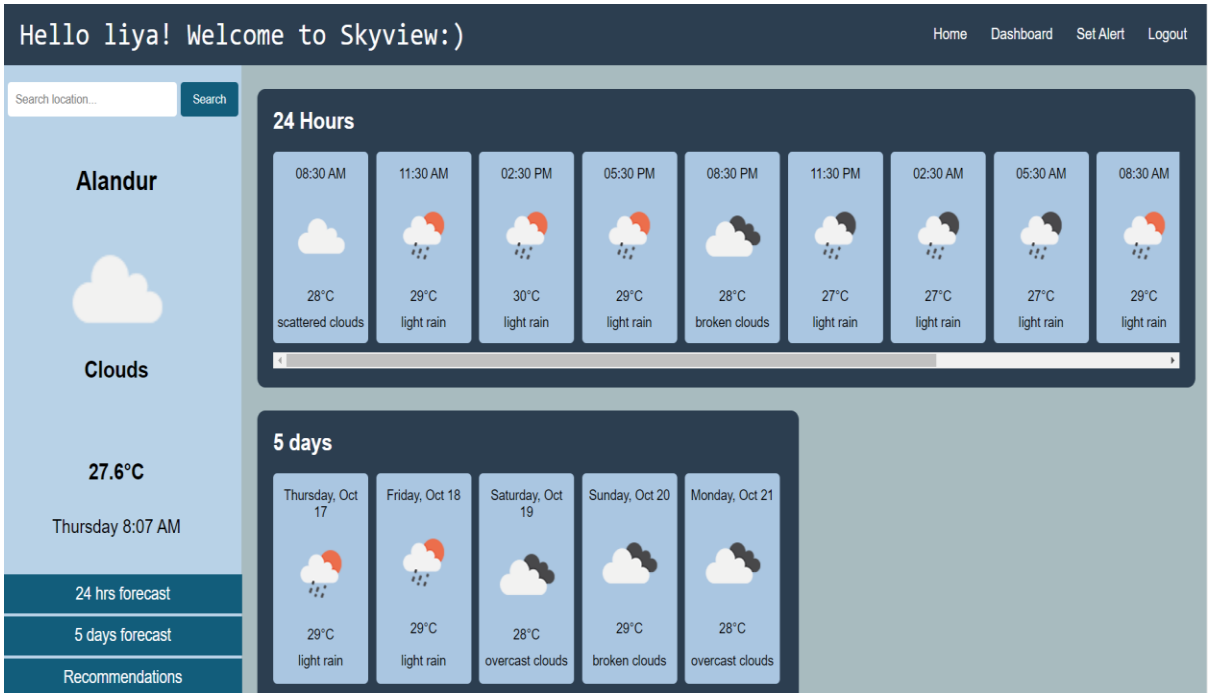
2. SIGN UP PAGE



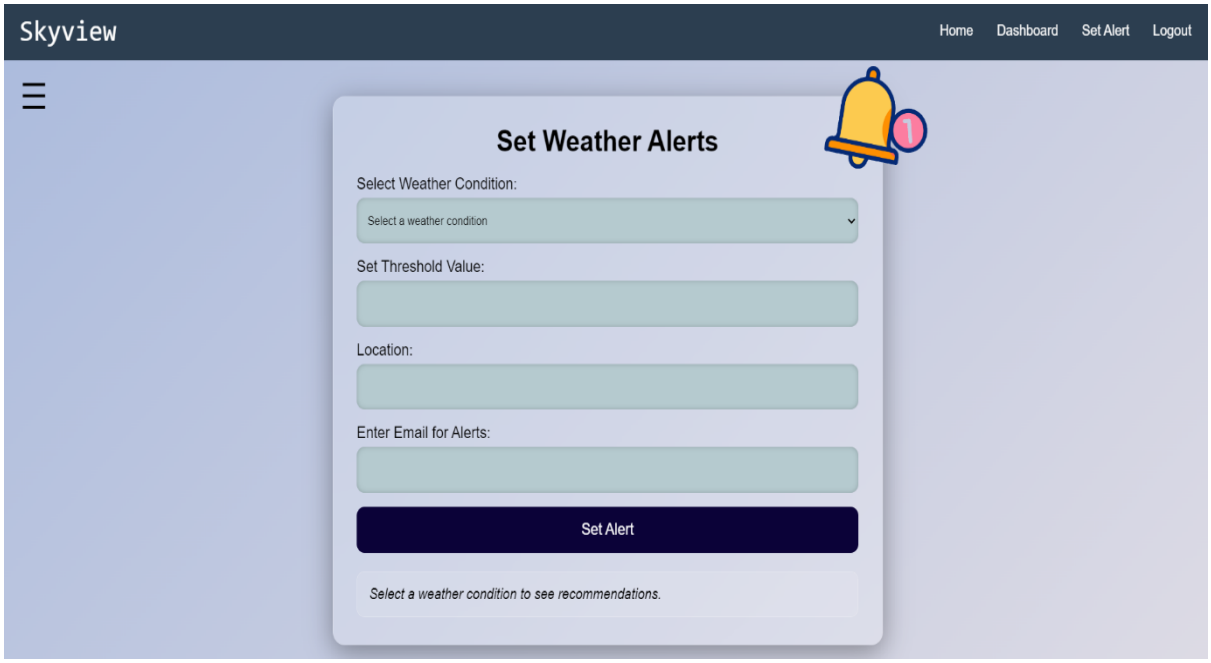
3. LOGIN PAGE



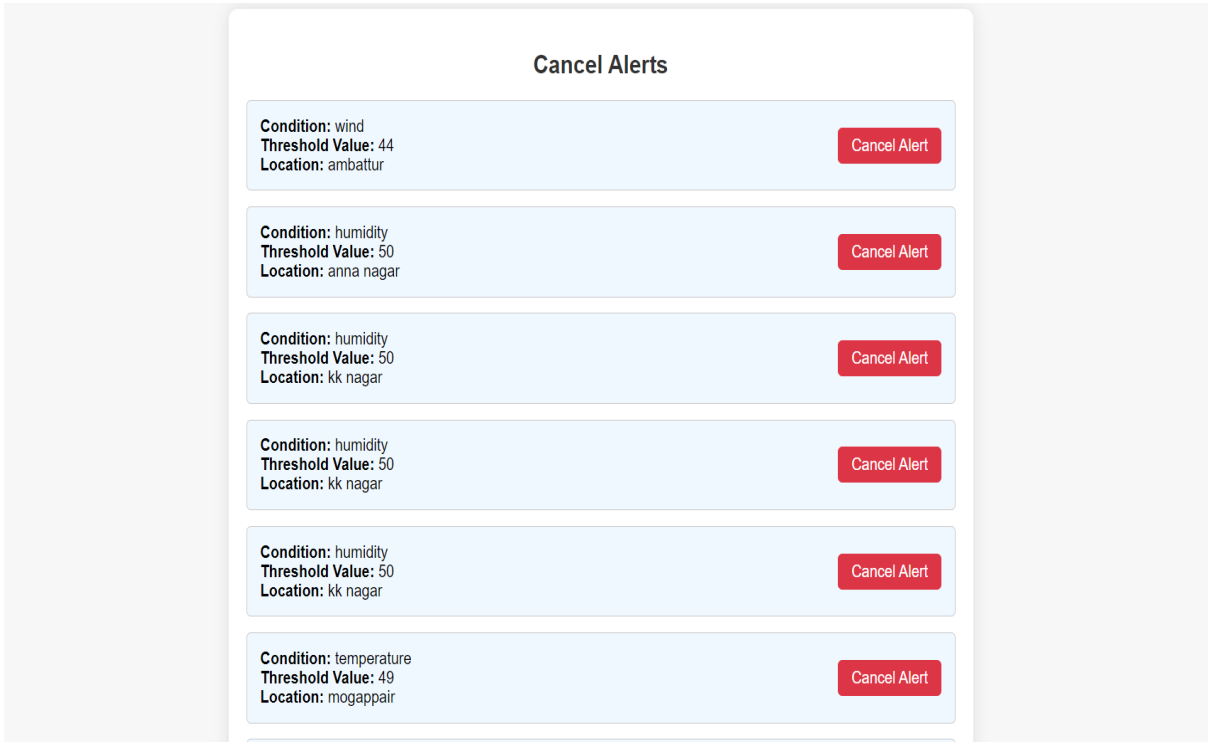
4. DASHBOARD



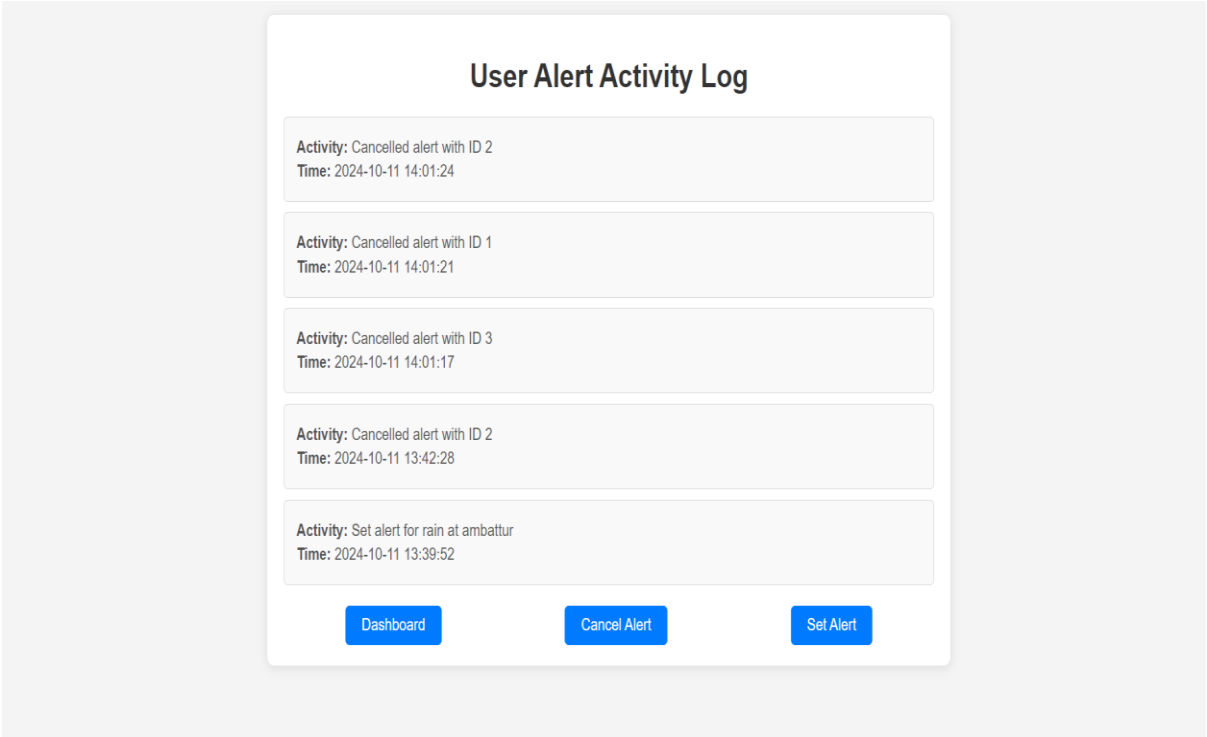
5. SET ALERTS PAGE



6. CANCEL ALERTS PAGE



7. USER ACTIVITY LOG PAGE



CHAPTER 8

CONCLUSION

Skyview is a tailored, user-friendly weather platform that offers real-time, location-specific forecasting and alerts. Upon signing in, users access a personalized dashboard displaying the current forecast, 24-hour and 7-day predictions, as well as data from the previous two days, empowering them to make informed choices for both daily and travel plans.

Skyview's advanced features include weather-dependent recommendations, providing users with guidance on activities and precautions suited to forecasted conditions. Additionally, the platform's customizable alert system allows users to set email notifications for specific weather changes in their current or selected locations, ensuring timely updates. Users retain full control with options to cancel or review alerts via an activity log, supporting ongoing engagement and flexible alert management.

Through its seamless combination of real-time data, proactive alerting, and actionable recommendations, Skyview enhances weather awareness and simplifies planning. The platform is a valuable tool for users who prioritize preparedness, providing all essential weather information in a convenient, easy-to-navigate format.

CHAPTER 9

REFERENCES

1. **OpenWeatherMap API Documentation**

OpenWeatherMap provides weather data including real-time weather, forecasts, and historical data, making it a crucial resource for weather-related projects.

https://openweathermap.org/api

2. **Weatherstack API Documentation**

Weatherstack offers real-time and historical weather data, which can be useful for integrating accurate weather forecasts into Skyview.

https://weatherstack.com/documentation

3. **Google Maps Geolocation API**

This API can help obtain users' locations based on IP addresses or GPS data, essential for providing location-specific weather forecasts.

https://developers.google.com/maps/documentation/geolocation

4. **React.js Official Documentation**

React is an essential front-end library for building the user interface of Skyview. This documentation provides guidelines and best practices for developing with React.

https://reactjs.org/docs/getting-started.html

5. **Node.js with Express.js Documentation**

The combination of Node.js and Express.js is commonly used for building the backend of web applications like Skyview.

https://expressjs.com/