

AI-POWERED WEATHER FORECASTING SYSTEM! (MAUSAM)

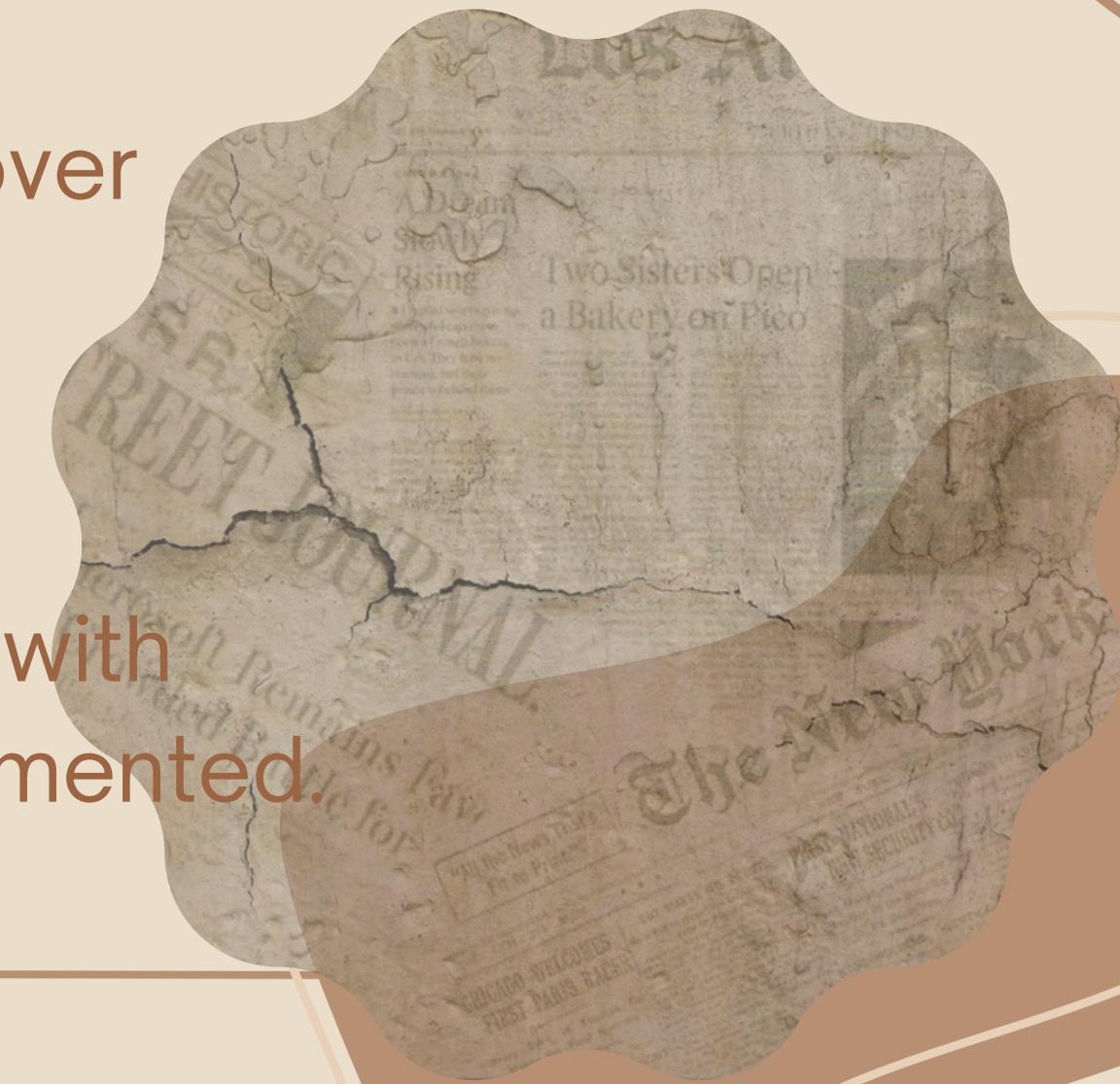
INTRODUCTION

Purpose:

- Provide real-time, AI-enhanced weather predictions.
- Improve accessibility and accuracy over traditional methods.

Key Innovation:

- Hybrid approach combining API data with predictive analytics (if AI/ML is implemented).



PROBLEM STATEMENT

Challenges in Traditional Forecasting:

- Limited real-time updates and accessibility.
- High computational costs for high-resolution models.

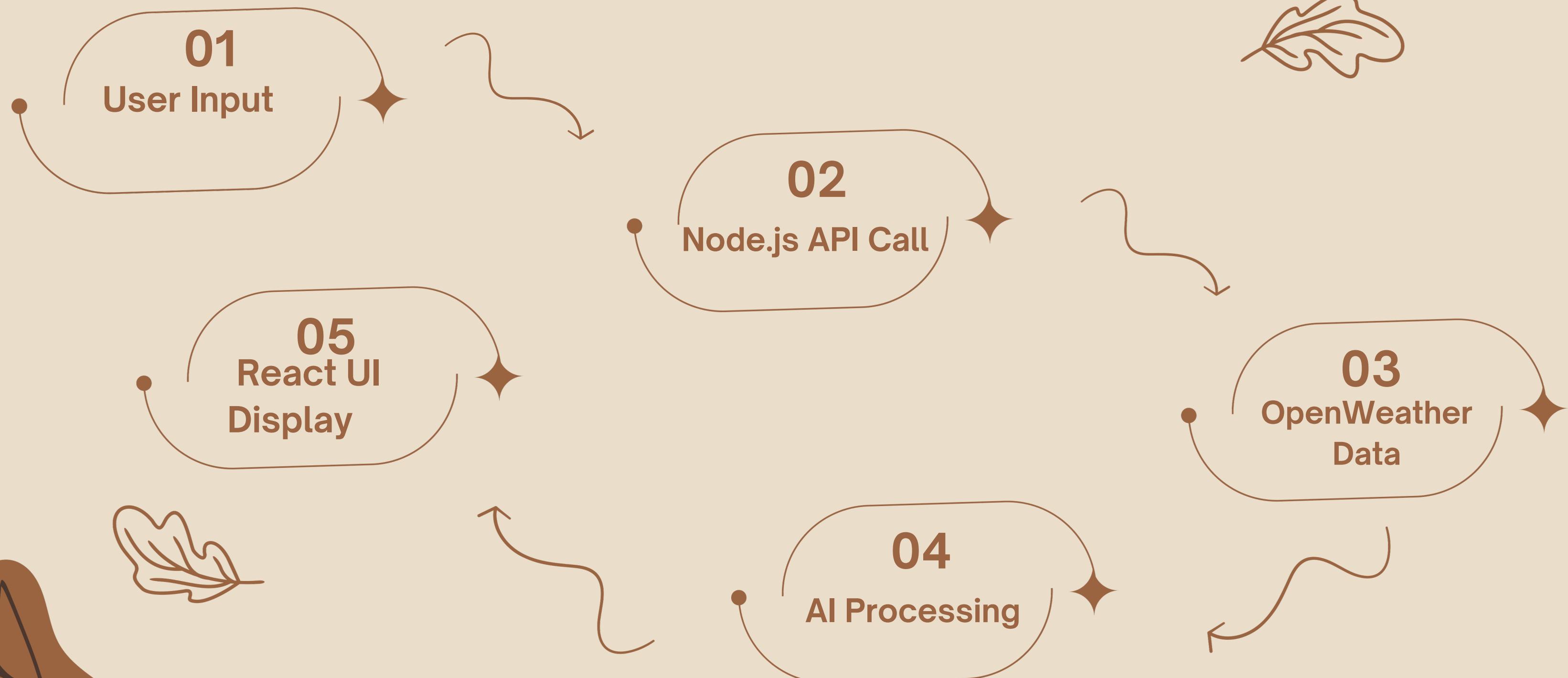
Need for AI Integration:

- Faster processing of big data (e.g., 600k+ METAR bulletins/year).
- Improved accuracy for extreme weather events.

TECHNOLOGIES USED

- **Frontend:** React.js (dynamic UI, state management with hooks).
- **Backend:** Node.js (API requests, error handling).
- **APIs:** OpenWeather (real-time data), optionally Kaggle datasets.
- **AI/ML:** Random Forest Regression.

WORKFLOW DIAGRAM



FEATURES

Core Features:

- Real-time temperature, humidity, and precipitation.
- Dynamic UI with weather-based themes (e.g., rain animations).

Innovative Add-ons:

- Speech recognition for accessibility (inspired by research).
- Multi-day forecasts (future scope).

IMPLEMENTATION

Backend (Node.js):

- Axios for API calls with error handling.
- Example: GET requests to OpenWeather's /weather endpoint.

Frontend (React.js):

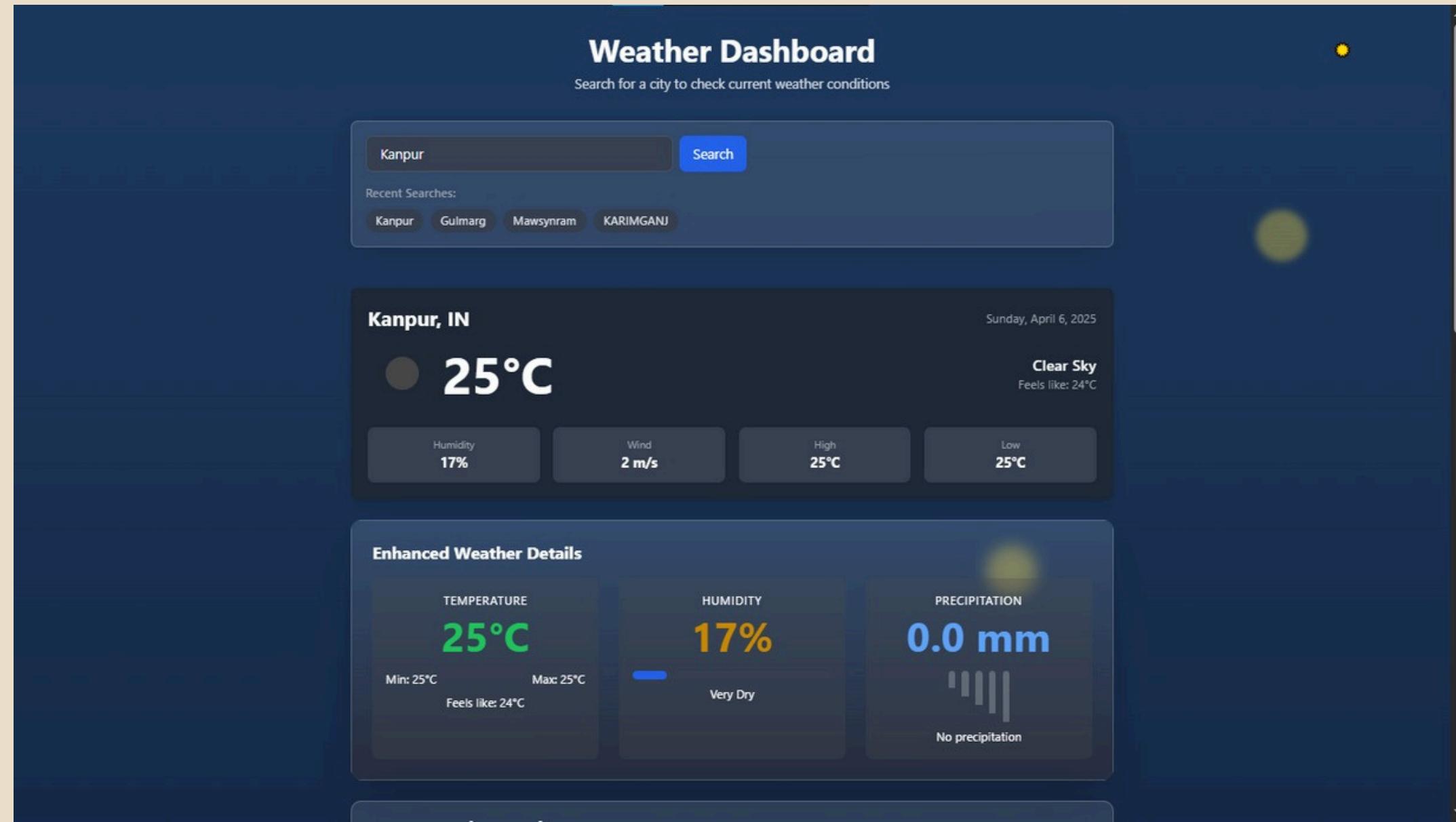
- useEffect for API data fetching.
- Conditional rendering for weather icons.

APPLICATION DEMO

INTERACTIVE SHOWCASE OF CORE FEATURES

SEARCH INTERFACE:

- REACT-BASED CITY SEARCH WITH AUTOCOMPLETE (INSPIRED BY OPENWEATHER CITIES API INTEGRATION).
- REAL-TIME VALIDATION FOR INVALID INPUTS

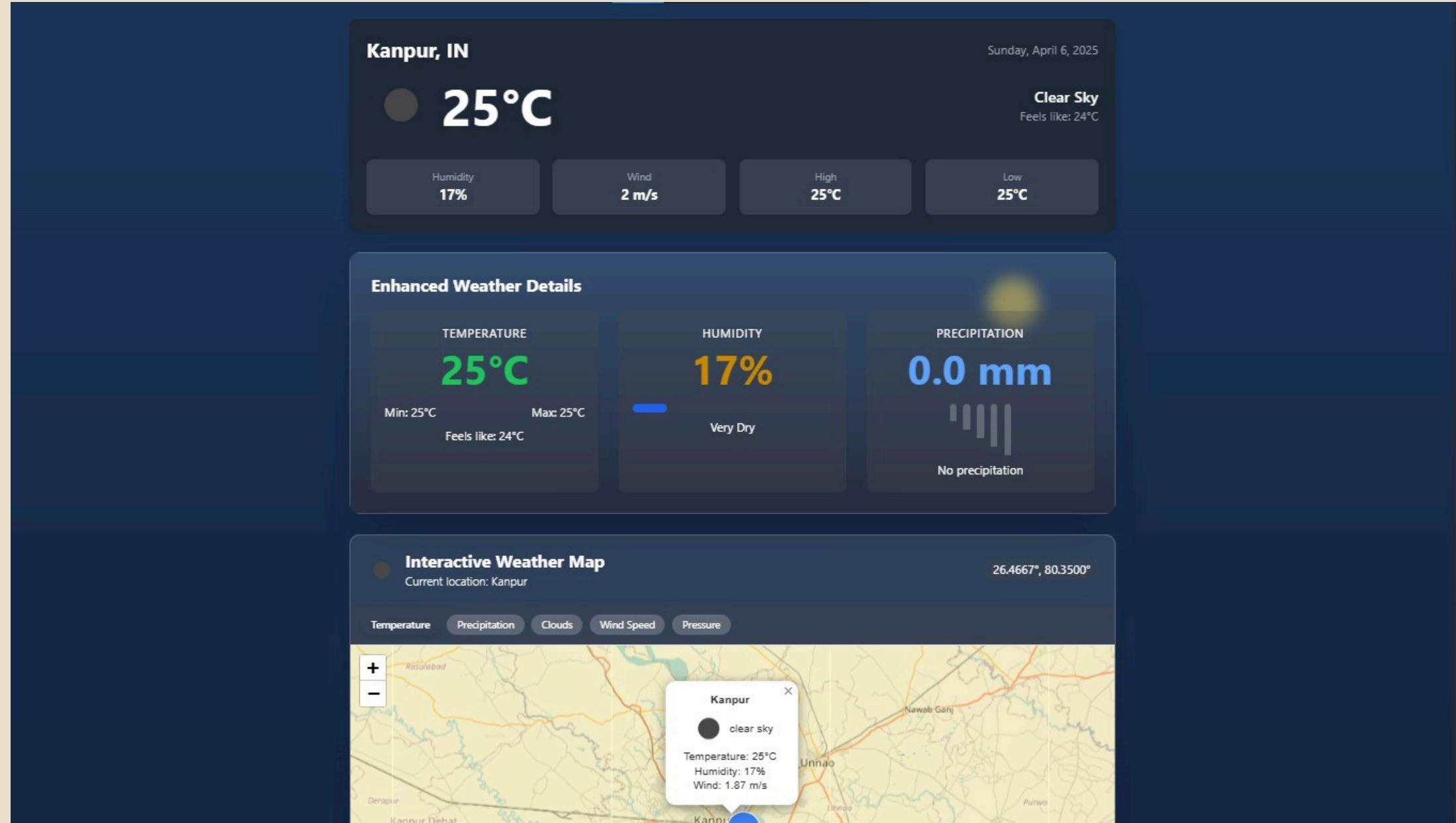


APPLICATION DEMO

INTERACTIVE SHOWCASE OF CORE FEATURES

WEATHER DASHBOARD:

- DYNAMIC CARDS DISPLAYING TEMPERATURE, HUMIDITY, AND CONDITION ICONS.
- RESPONSIVE LAYOUT OPTIMIZED FOR MOBILE/DESKTOP.



APPLICATION DEMO

INTERACTIVE SHOWCASE OF CORE FEATURES

ERROR HANDLING:

- USER-FRIENDLY INVALID CITY DETECTION
- FALBACK UI FOR API FAILURES

The screenshot shows a dark-themed web application titled "Weather Dashboard". At the top, there is a search bar with the placeholder "Search for a city to check current weather conditions". Below the search bar, a user has typed "afsnf" into the input field. A blue "Search" button is positioned to the right of the input field. Underneath the search bar, there is a "Recent Searches:" section containing five small buttons: "New York", "unnao", "Yakutsk", "London", and "Rio de Janeiro". In the center of the page, a large modal window is displayed with a dark background. The modal contains the text "Couldn't find weather data for this location." in white. Below this text, it says "Request failed with status code 404" and "We couldn't find weather data for 'afsnf'". At the bottom of the modal, there is a "Try one of these popular cities:" section with five buttons: "New York", "London", "Tokyo", "Sydney", and "Paris". The footer of the application includes sections for "Mausam Weather", "Stay Updated", and "Connect With Us". The "Mausam Weather" section describes the app as "Your go-to application for accurate weather forecasts, personalized alerts, and detailed climate information." It includes a "View Team" link. The "Stay Updated" section has a "Subscribe" button and a form for entering an email address. The "Connect With Us" section shows five social media icons and the text "New South Wales, AU" and "contact@mausam-weather.com". The footer also includes copyright information ("© 2025 Mausam Weather App. All rights reserved.") and a note about the API used ("Powered by OpenWeather API").

The screenshot shows a dark-themed web application titled "Weather Dashboard". At the top, there is a search bar with the placeholder "Search for a city to check current weather conditions". Below the search bar, a user has typed "afsnf" into the input field. A blue "Search" button is positioned to the right of the input field. Underneath the search bar, there is a "Recent Searches:" section containing five small buttons: "New York", "unnao", "Yakutsk", "London", and "Rio de Janeiro". In the center of the page, a large modal window is displayed with a dark background. The modal contains the text "Couldn't find weather data for this location." in white. Below this text, it says "Request failed with status code 404" and "We couldn't find weather data for 'afsnf'". At the bottom of the modal, there is a "Try one of these popular cities:" section with five buttons: "New York", "London", "Tokyo", "Sydney", and "Paris". To the right of this modal, there is a "Mausam Weather" section with the text "Your go-to application for accurate weather forecasts, personalized alerts, and detailed climate information." It includes a "View Team" link. The "Stay Updated" section has a "Subscribe" button and a form for entering an email address. The "Connect With Us" section shows five social media icons and the text "New South Wales, AU" and "contact@mausam-weather.com". The footer also includes copyright information ("© 2025 Mausam Weather App. All rights reserved.") and a note about the API used ("Powered by OpenWeather API").

CHALLENGES

1. Data Limitations:

- **Scarcity of Annotated Data:** Extreme events are rare, leading to insufficient training data for AI models.
- **Data Biases:** Models inherit biases from training data (e.g., underrepresentation of Himalayan regions in India).

2. Operational Challenges:

- **Computational Costs:** High GPU/energy demands for training large models.
- **Real-Time Integration:** Difficulty adapting models to noisy, real-world data streams.

FUTURE ENHANCEMENTS

Short-Term:

- Geolocation-based auto-detection.
- Severe weather alerts.

Long-Term:

- GPU-accelerated forecasting (like GRAF model).
- High-resolution spatial modeling.

CONCLUSION

Technical Achievement:

- Built a full-stack weather app with React, Node.js, and OpenWeather API
- Demonstrated seamless API integration and real-time data handling

Collaborative Success:

- Delivered a functional product through effective team coordination
- Mastered Git workflows and agile development practices

Future-Ready Foundation:

- Scalable architecture for AI/ML integration (when implemented)
- Responsive design adaptable to new features like geolocation

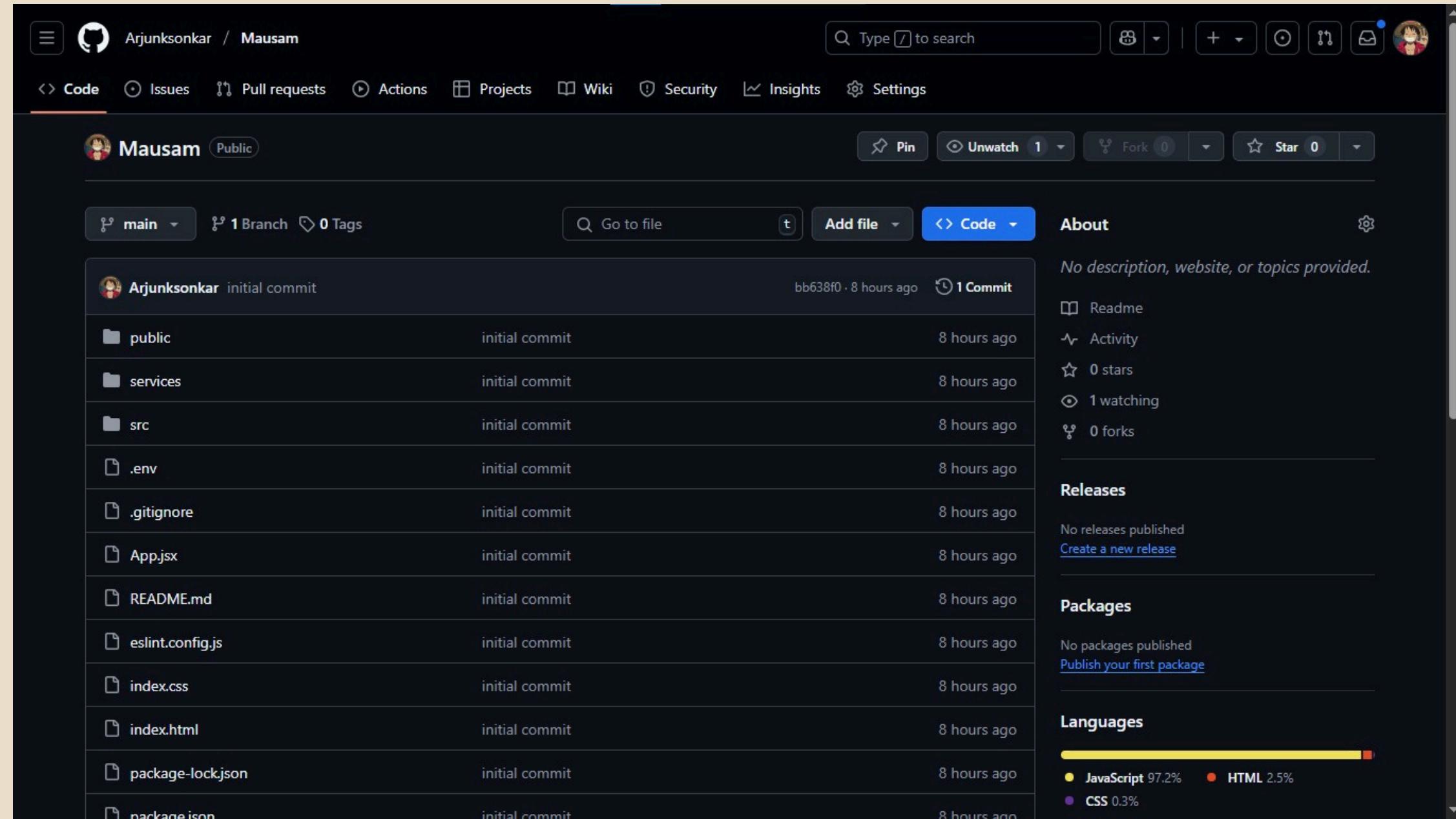
GITHUB

LINK:

[HTTPS://GITHUB.COM/KUSH1309/MAUSAM](https://github.com/kush1309/mausam)

[HTTPS://GITHUB.COM/ARJUNKSONKAR/MAUSAM](https://github.com/arjunksonkar/mausam)

[HTTPS://GITHUB.COM/X-ZAKARIYA-X/MAUSAM](https://github.com/x-zakariya-x/mausam)



THANK YOU