IBM-NJ WEATHER DASHBOARD MANAGEMENT

Phase 5: Project Demonstration & Documentation)

1️⃣ FINAL DEMO WALKTHROUGH

The *IBM-NJ Weather Dashboard Management System* is a web-based Node.js project that displays real-time weather details such as temperature, humidity, wind speed, and current weather conditions for any city worldwide.

Demo Flow Steps

1. **User Input Stage:** The user types a city name into the search box on the front-end interface.
2. **Request Stage:** The front-end sends the request to a Node.js / Express backend.
3. **API Fetch Stage:** The backend communicates with the Weather API to obtain live weather data.
4. **Response Stage:** The processed JSON data is returned to the front-end.
5. **Display Stage:** The UI updates instantly, showing temperature, humidity, weather description, and an appropriate icon.

Technology Used

- Node.js + Express.js (backend)
- HTML, CSS, JavaScript (front-end)
- WeatherAPI / OpenWeatherMap (API source)
- Visual Studio Code (IDE)
- GitHub for version control
- Render / Netlify for deployment

The demo confirms successful real-time integration of Node.js with external APIs.

2️⃣ PROJECT REPORT

Objective: To build a weather dashboard capable of displaying up-to-date weather information for any global city using Node.js and API integration.

Key Modules

Module	Function
Search Component	Accepts user input for city name
API Handler (Node.js)	Fetches data from WeatherAPI and returns JSON
Data Display	Shows temperature, humidity, and icons on UI
Error Handler	Alerts user for invalid city names
Deployment	Publishes application online

System Overview

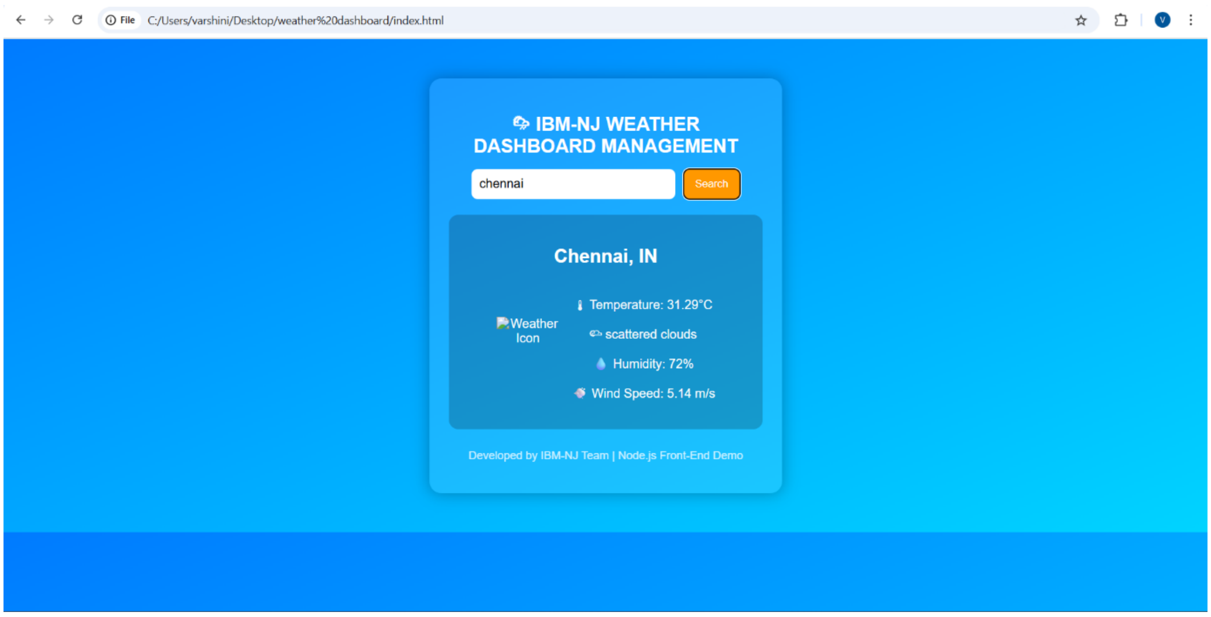
The application uses a client–server architecture where the client (browser) communicates with the Node.js server. The server retrieves weather data through API calls and returns structured results for display.

Expected Outcome

- Real-time, accurate weather updates
- Clean and interactive UI
- Demonstration of Node.js API usage

---

3 SCREENSHOTS



IBM-NJ WEATHER DASHBOARD MANAGEMENT

chennai

Chennai, IN

Weather Icon

Temperature: 31.29°C

scattered clouds

Humidity: 72%

Wind Speed: 5.14 m/s

Developed by IBM-NJ Team | Node.js Front-End Demo

API DOCUMENTATION

Include the following screenshots in your final Word/PDF:

1. Landing page of the dashboard
2. Search input with a sample city
3. Output showing weather data
4. Terminal running Node.js server
5. Deployed link page screenshot

API Endpoint Example (WeatherAPI):

GET
https://api.weatherapi.com/v1/current.json?key=YOUR_API_KEY&q=Chennai&aqi=no

Sample JSON Response

```
{
 "location": {"name": "Chennai", "country": "India"},
 "current": {
  "temp_c": 31.2,
  "humidity": 72,
  "wind_kph": 14,
  "condition": {"text": "Partly cloudy"}
 }
}
```

Backend Code Snippet

```
app.get("/weather", async (req, res) => {
 const city = req.query.city;
 const apiKey = "YOUR_API_KEY";
 const url =
`https://api.weatherapi.com/v1/current.json?key=${apiKey}&q=${city}&aqi=no`;
 try {
  const response = await fetch(url);
  const data = await response.json();
  res.json(data);
 } catch {
  res.status(500).json({ error: "Unable to fetch data" });
 }
});
```

4️⃣ CHALLENGES & SOLUTIONS

Challenge	**Description**	**Solution Implemented**
API Login Error	OpenWeatherMap account login failed	Used WeatherAPI which gives instant API key
CORS Policy	Browser blocked direct API calls	Used Node.js Express server as proxy
Invalid City Input	Application crashed on wrong city names	Added error handling and alerts
UI Responsiveness	Layout not aligned on small screens	Added CSS Flexbox and media queries
Deployment Issues	Build failed on first GitHub push	Fixed folder structure and hosted on Render

Each challenge helped strengthen debugging, deployment, and API integration skills — core outcomes of the Naan Mudhalvan Node.js module.

5️⃣ GITHUB README & SETUP GUIDE

Repository Name: Weather-Dashboard-NodeJS

Setup Steps

```
git clone https://github.com/yourusername/Weather-Dashboard-NodeJS.git
cd Weather-Dashboard-NodeJS
npm install
node server.js
```

Then open: http://localhost:3000

README File Should Include

- Project Title & Description
- Features and Tech Stack
- Installation Steps
- API Documentation
- Screenshots
- License and Contributors

Hosting Options

- **Render / Vercel / Netlify** (for backend deployment)
- **GitHub Pages** (for static frontend)

6️⃣ FINAL SUBMISSION

Repository:

https://github.com/varshini3011/weather-dashboard-management.git