

WeatherWise

Real-Time Weather Forecast Dashboard

By : Aditya Kumar & Ashif Rahman

> Problem & Solution <

Problems:

- Fragmented weather information
- Cluttered interfaces with ads
- Poor data visualization
- Outdated designs

Solution:

- Clean, fast, beautiful dashboard
- Real-time data + interactive charts
- Modern glassmorphism UI
- Ad-free experience

> Key Features <

Core Features:

- Dual Location - GPS + city search
- Interactive Charts - Temperature & humidity
- Modern UI - Glassmorphism + Dracula theme
- Real-Time - Auto-refresh every 15 min
- Unit Toggle - Celsius ↔ Fahrenheit
- Smart Cache - 10-min cache, faster loads

> Technology Stack <

Frontend:

- HTML5, CSS3, JavaScript (ES6)
- Chart.js, Material Icons

Backend:

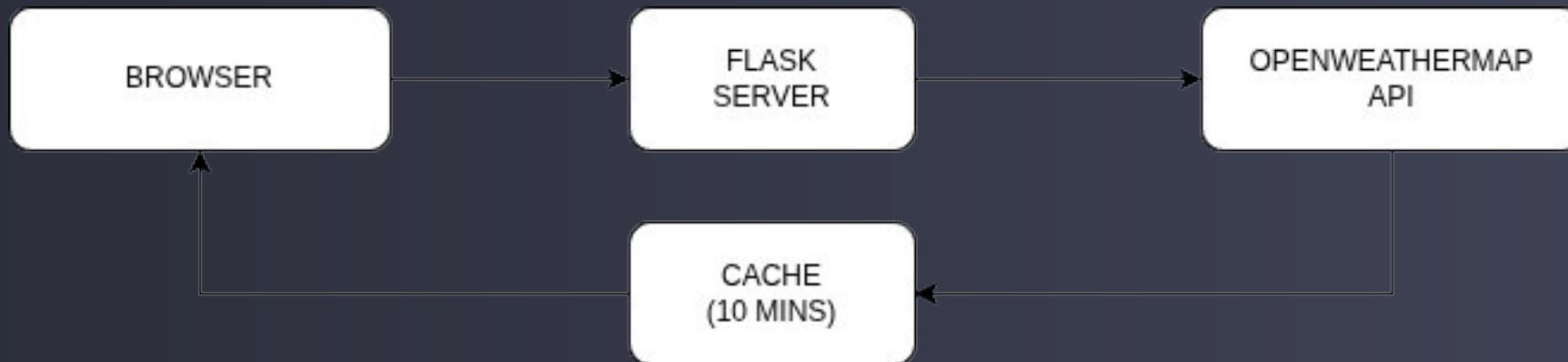
- Python 3.13, Flask 3.0
- Flask-CORS, Requests

> Technology Stack <

External:

- OpenWeatherMap API v2.5

Architecture:



> Data Flow <

Process:

1. User searches city
2. Flask checks cache (10 min)
3. If fresh → return cached data
4. If stale → call OpenWeatherMap API
5. Store in cache
6. Return to browser
7. Display with charts

Benefits:

- Fast response times
- Reduced API usage
- Better user experience

> UI & Design <

Design Principles:

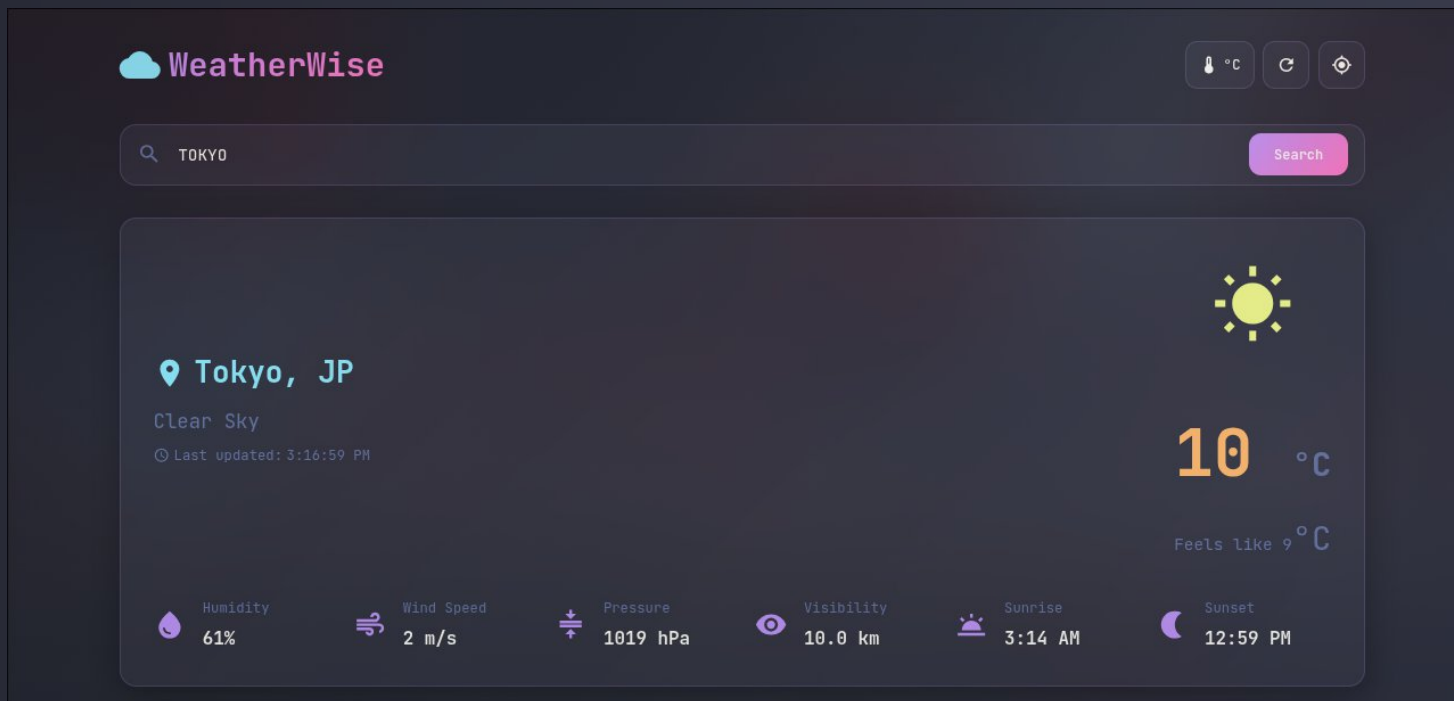
- Glassmorphism – Frosted glass effect
- Dark Theme – Dracula color palette
- Responsive – Desktop, tablet, mobile
- Animations – Smooth transitions

Color Palette:

- Purple `#bd93f9`, Pink `#ff79c6`, Cyan `#8be9fd`
- Orange `#ffb86c`, Yellow `#f1fa8c`, Dark `#282a36`

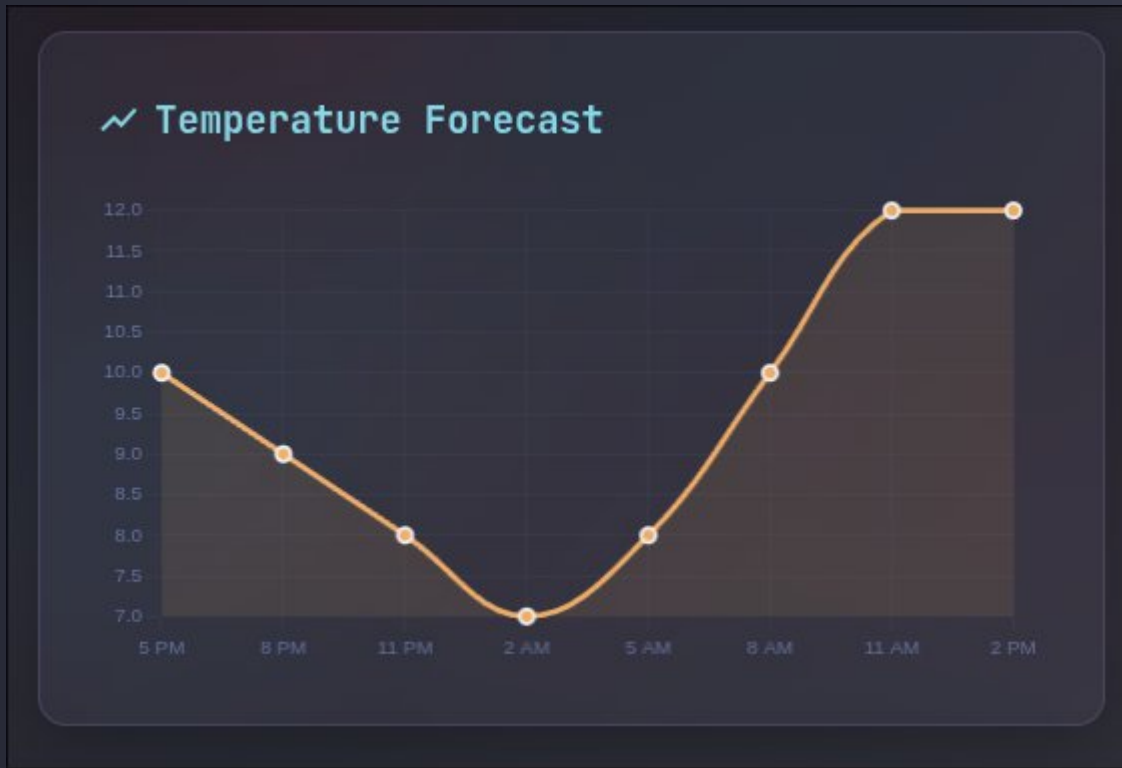
> UI & Design <

Screenshots:



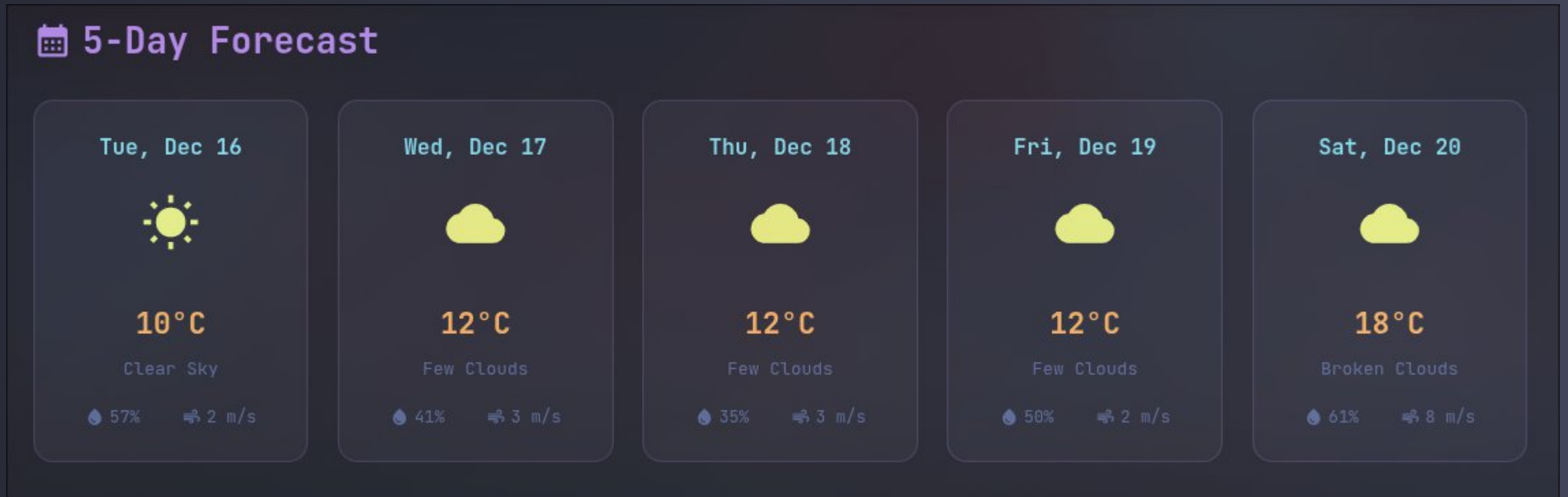
> UI & Design <

Screenshots:



> UI & Design <

Screenshots:



> Feature Deep Dive <

Location:

- Auto GPS detection
- Manual city search
- Remembers last location

Current Weather:

- Temperature + feels-like
- Humidity, wind, pressure
- Visibility, sunrise/sunset

> Feature Deep Dive <

Forecast:

- 5-day predictions
- Noon-time snapshots
- Weather icons + details

Charts:

- 24-hour temperature trend
- Humidity bar chart
- Interactive tooltips

> Technical Highlights <

Optimizations:

- Caching - 10-min cache → 83% fewer API calls
- Retry Logic - 3 attempts → 99.5% uptime
- Parallel Calls - Promise.all → 50% faster loads
- LocalStorage - Saves preferences → personalized UX

```
# Cache system  
cache = {'data': weather, 'timestamp': now}
```

```
# Retry logic  
for attempt in range(3):  
    try: return fetch_data()  
    except: continue
```

> Conclusion <

Achievements:

- Full-stack web app
- Real-time API integration
- Modern UI/UX
- Responsive design
- Open-source (GPL-3.0)

Links:

- GitHub: github.com/roxxadiiii/WeatherWise
- License: GPL-3.0

> Resources <

- GeistMono Nerd font
- MapleMono Nerd font
- Mononoki Nerd font
- Dracula colorscheme
- LLM for debugging and formatting
- Github for managing code
- OnlyOffice for making ppt



Scan for code