Innovate Michigan!

2025 Student Led
Faculty Guided Project

ENHANCING ECONOMIC GROWTH THROUGH AIR QUALITY DATA ANALYSIS & UX DESIGN IN LOW-ECONOMIC REGIONS OF MICHIGAN

Dr. Dirk Colbry, Dr. Justin Gambrel, Deepak Reddy Sannapareddy, Abdelmoumen Medaghri Alaoui, Bhavya Jain, Adam Cooper, River Meckstroth

MICHIGAN STATE UNIVERSITY | University Outreach and Engagement

Enhancing Economic Growth through Air Quality Data Analysis and UX Design in Low-Economic Regions of Michigan

Dr. Dirk Colbry and Dr. Justin Gambrel, Instructors
Deepak Reddy Sannapareddy, Graduate Student UX Expert
Jeffrey Siarto, Director of UX program and advisor to Deepak
River Meckstroth, Adam Cooper, Abdelmoumen Medaghri Alaoui, Bhavya Jain Data Science capstone team

This Student-Led Faculty-Guided (SLFG) project focused on improving the usability and accessibility of JustAir's air quality monitoring platform for communities in Wayne County and beyond. By combining User Experience (UX) design expertise with advanced data science methods, the project aimed to make complex environmental data more understandable and actionable for residents, policymakers, and community organizations.

Background

Accessible air quality data is essential for empowering communities to protect their health and advocate for cleaner environments. When residents can easily interpret local air quality conditions, they are better equipped to make informed decisions about outdoor activities, reduce exposure to harmful pollutants, and take preventive measures for vulnerable populations such as children, seniors, and those with respiratory conditions. Beyond individual health, transparent and user-friendly data fosters trust, encourages community engagement, and supports policy initiatives aimed at reducing pollution and improving overall quality of life. By turning complex environmental metrics into clear, actionable insights, accessible air quality information becomes a powerful tool for equity and public well-being.

As a community partner, JustAir plays a critical role in advancing these goals. The Michigan-based environmental technology startup is dedicated to improving air quality awareness and public health outcomes through hyperlocal monitoring and data-driven insights. By deploying a network of sensors and providing real-time analytics, JustAir empowers communities, policymakers, and organizations to make informed decisions about environmental risks. Their commitment to accessibility and transparency aligns closely with the objectives of this project, ensuring that residents in Wayne County and beyond can easily understand and act on air quality information. Through this partnership, JustAir has demonstrated a strong focus on community engagement and innovation, making them an invaluable collaborator in promoting regional sustainability and equity.

Project Goals

The primary goals of the project were to enhance JustAir's dashboard interface to ensure clarity and accessibility for diverse audiences, develop interactive visualizations and analytical frameworks for air quality data, and provide Michigan State University (MSU) students with experiential learning opportunities while addressing real-world community needs. These objectives were pursued through a collaborative effort involving a UX graduate student and a team of undergraduate data science capstone students.

Activities and Contributions

Deepak, the UX graduate student funded through this award, played a pivotal role in redesigning JustAir's client-facing dashboard. His work included conducting a comprehensive usability and accessibility audit which resulted in identifying gaps in color contrast, keyboard navigation, and screen reader compatibility. He then proposed and implemented design improvements using Figma (a popular UX tool used to prototype designs), introducing scalable design components and a new wind analysis panel that balances technical detail with approachability for general users. In addition to his direct contributions to JustAir's platform, Deepak served as a UX consultant for all six teams in the Data Science capstone course (not just the JustAir team), offering guidance on user-centered design principles and accessibility best practices. Deepak's Final detailed case study of his work for JustAir can be found on his website (https://www.d3pk.design/project-2) and is attached as Appendix A. His final presentation slides are attached as Appendix B.

The JustAir capstone team complemented these efforts by developing a reusable visualization toolkit for air quality data. Working with pollutant and meteorological datasets from Wayne County, Michigan, the team created nine interactive visualizations, including Air Quality Index (AQI) timelines, pollutant distribution charts, and correlation heatmaps. They also implemented a short-term AQI forecasting module using machine learning techniques, demonstrating the potential for predictive analytics to support proactive public health measures. Their work emphasized reproducibility and adaptability, ensuring that the tools could be applied to future datasets and evolving community needs. A comprehensive report written by the capstone students is included in Appendix C.

Key Outcomes

The project achieved several significant outcomes. First, the dashboard redesign improved accessibility by addressing color contrast issues, ensuring keyboard navigation, and enhancing screen reader compatibility. Second, the introduction of a scalable design system established a foundation for consistent and future-proof interface development. Third, the visualization toolkit enabled JustAir to communicate air quality trends effectively to both technical and non-technical audiences. Finally, the predictive analytics module introduced forecasting capabilities that can support proactive public health measures.

Impact

This project strengthened JustAir's ability to serve Michigan communities with a focus on Wayne County by making air quality data more understandable and actionable. The collaboration provided MSU students with hands-on experience in UX design and data science, while advancing regional economic innovation through technology-driven environmental solutions. By improving the accessibility and clarity of air quality information, the project supports informed decision-making for public health and environmental policy. The tools and design principles developed through this initiative will continue to benefit JustAir and the state of Michigan as they expand their monitoring network and outreach efforts.

Acknowledgements

This project is supported in part pursuant to the receipt of financial assistance from the United States Department of Commerce-Economic Development Administration, **and the Michigan Economic Development Corporation.** The opinions expressed in the statements, findings, conclusions, and recommendations are the authors' sole responsibility and do not necessarily reflect the views of Michigan State University or any federal or state agency. We also thank JustAir, Wayne County stakeholders for their guidance and partnership.

Appendix A

Graduate Student UX Expert's final detailed case study of his work for JustAir

https://www.d3pk.design/project-2

80+

accessibility and usability issues identified, prioritized, and tied to concrete design changes

Over three months, I led a full accessibility and UX assessment of the JustAir air-quality dashboard, covering color contrast, structure, semantics, charts, navigation, and color-vision reliability. I audited the entire interface through WCAG 2.1 AA guidelines using Lighthouse, A11y, ChromeVox, keyboard-only flows, and four color-blindness simulations. This uncovered a wide range of issues across contrast, ARIA roles, heading hierarchy, form labeling, keyboard focus, and chart readability.

I then delivered a cohesive redesign strategy that improves clarity, legibility, and consistency without breaking the EPA-required AQI palette. This included a WCAG-compliant color system, simplified and unified chart styles, accessible line patterns, scaled-down axis density, pollutant-specific micro-charts, stronger semantic labeling, and a new wind-panel design that balances beginner readability with expert depth.

Context & Scope

JustAir is a Michigan-based environmental startup working with Wayne County to deploy over 100 new air-quality monitors across 43 communities.

Their goal is to Bring hyperlocal air data to residents, especially in ALICE (Asset-Limited, Income-Constrained, Employed) and Opportunity Zone neighborhoods.

My role was as a UX Consultant for both JustAir and the CMSE 495 Capstone team at Michigan State University. I helped shape how data could be seen, trusted, and understood, instead of just displayed. That meant working with both data scientists and the JustAir dev team to translate complex environmental data into something that actually makes sense to a human at 9 AM with a cup of coffee.

The collaboration had two main goals:

Redesign JustAir's core dashboard

to improve **usability, accessibility, and clarity**, making sure users with different visual abilities could navigate and interpret the data equally well.

Design a Wind Analysis Panel

a feature centered around an interactive **windrose chart** that helps users connect air quality data with wind direction and real-time conditions.

Understanding Users & the Market

I started with an exploratory competitive analysis to understand how similar platforms approach these problems. Alongside that, I created two personas using insights gathered from Reddit discussions, Google searches, and community forums, since there wasn't enough time to run full user interviews. (Reddit users, for better or worse, were very eager to share their opinions.)

I focused on understanding how other companies handle and present their environmental data, and used those observations to outline potential feature sets that align with JustAir's mission. This process also helped me provide clearer direction and more informed insights to the student team working alongside me.

Competitive Analysis

Community-Driven Transparency

Data-Driven Insights Approach

- Community validation; raw + corrected data shown
- Integrations: EPA, Weather Underground, Windy
- Deep-learning calibration for EPA alignment

Global Health Intelligence

Data-Driven Insights

- Global coverage (100+ cou
- ML-based 7-day forecasti
- Health correlation engine

Data Presentation Methods

- Interactive heat maps
- Dual display (raw vs corrected)
- QR code instant dashboard
- Community notes

Strengths / Differentiators

- Transparency
- Crowd validation
- Strong integrations
- ML calibration

Feature Ideas to Implement

Heatmap Visualization

Interactive color overlays by pollutant/time.
Interpolation modes, sensor points vs contour surfaces, quick toggles.

EPA correction display

Show math that adjusts raw sensor data.
Inline formula, factors (humidity, temp), and doc links.

Government Authority

Data-Driven Insights Approach

- EPA NowCast algorithm (stable AQI)
- Primary pollutant highlight
- QA on all data before release

Data Presentation Methods

- Central AQI dial
- Forecast circles & contour maps
- Embeddable widgets
- Trend buttons (day/week/month)

Data Presentation Me

- Health screens per pollutan
- Mobile-first UX
- Global city rankings
- Wildfire tracking

Strengths / Differenti

- Health focus
- Predictive modeling
- Strong mobile UI

Feature Ideas to Impl

Health-focused m

Convert AQI into specific
Audience-specific tips (ki

7-day AQI forecas

Predict air quality up to a
Combine met inputs and

Comparative city

Compare cities and tren
Rankings, seasonal patte

Wildfire/event int

Overlays + smoke transpo
Event dashboards and tim

Customizable Analytics

Strengths / Differentiators

- Authority
- Credibility
- Regulatory alignment

Feature Ideas to Implement

Prominent gauge with category colors.

EPA color steps; large number + category text.

Recognized AQI calculation.

Docs + validation notes to build trust.

1-tap presets for 24h/7d/30d.

Quick summaries per period (avg/max/threshold hits).

Data-Driven Insights

- Dual dashboards (simple v
- Comparative & correlation
- Data export options

Data Presentation Me

- Parameter cards for pollut
- Side-by-side comparisons
- Customizable time ranges
- Scatter plots & trends

Strengths / Differenti

- Flexibility
- Advanced analytics
- Power-user control

Feature Ideas to Impl

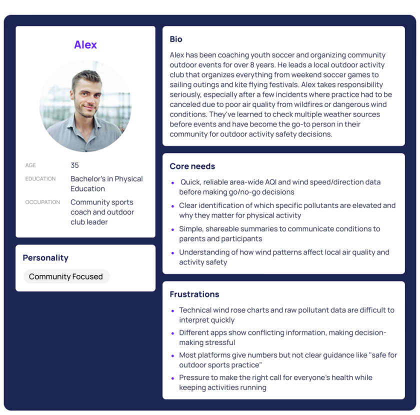
Simple for citizens; Adva

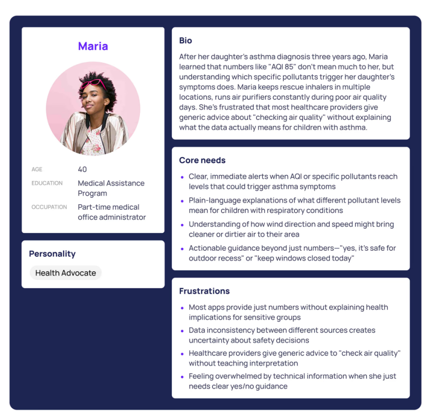
Switch complexity witho

Reference lines on chart

Tooltips with local regula

Personas





Identifying Barriers

I used Google Lighthouse and Axe (A11y engines) to automatically evaluate the webpages against WCAG 2.1 accessibility standards. These tools helped me quickly spot accessibility gaps, usability issues, and any problem areas in the existing design, which essentially became the starting point for the entire design process.

After that, I moved into manual testing. As part of this, I did keyboard-only navigation tests and screen-reader checks using ChromeVox. I also performed WCAG contrast reviews and color-accessibility checks across the user dashboard. This combination of automated and manual testing helped me get a full picture of the accessibility barriers and where improvements were needed.

For each test, I followed the exact same interaction script:

1. Navigate to .
2. Choose a → → → .
3. Once the opened, walk through every interactive element — dropdowns, graphs, date pickers, buttons, you name it.
4. Try changing graph dates, comparing monitors, downloading data, and using the calendar using only the keyboard. (The space bar is still recovering.)

Below is a quick summary of what I found.

WCAG Contrast Testing

1. **1.2:1** Aa Aa #B4FB9D / #FFFFFF
   Map badge / Tiles on light green / Compare Monitors numeric chips
2. **1.6:1** Aa Aa FFFFFF / #F7C946
   Yellow values on white
3. **1.72:1** Aa Aa #767676 / #25ABDE
   Air Quality Map btn / Recent Hours header
4. **2.16:1** Aa Aa #B4FB9D / #25ABDE
   Sign up
5. **2.35:1** Aa Aa #F5F5F5 / #25ABDE
   See Details / Links on light cards
6. **2.64:1** Aa Aa #25ABDE / #FFFFFF
   Primary links/buttons (Today, Compare, Remove Cluster)
7. **2.8:1** Aa Aa #4CAF50/ #FFFFFF
   “Geoapify” green pin/badge

View All Contrast Checks ↗

Manual & Automated Testing

Dialog Missing Name

Screen Reader Testing Lighthouse

en element" "`aria-role
ith ARIA `[role]` missing

or misread pollutant cards
ss context or labels.

e>/<section>) were given
t or need specific children.

arent/Child

ot allowed: `[role=group]`"

e the pollutant grid as

but children aren't valid

Text

ARIA Dialog High

"Elements with `role="dialog"` do not have accessible names"

Why it matters

Users can't tell what the dialog is about when it appears.

What it means

The sidebar/monitor details modal (e.g., .MuiDrawer-root) lacks a programmatic name linked with aria-labelledby.

Where

.MuiDrawer-root (sidebar / monitor-details)

Invalid ARIA Attributes

Screen Reader Testing A11Y

ARIA WCAG 4.1.2 High

IDs don't match, aria-labelledby/controls point nowhere.

Why it matters

Screen readers misannounce or lose context in tab panels/drawers.

What it means

Misapplied attributes on tabs, selects, drawers (WCAG 4.1.2).

Where

`#simple-tab-2`, `div.MuiSelect-select...`, `#location-tabpanel-1`, `#location-tabpanel-0`, `div.MuiDrawer-root.MuiDrawer-modal...`

Ghost focus (empty selectors)

Keyboard Testing Lighthouse

Media WCAG 1.1.1 Medium

Tabbing highlights elements that don't respond to Enter/Space

Form Inputs Miss

Screen Reader Testing A11Y

Forms WCAG 1.3.1 High

Inputs and selects don't have labels or

Why it matters

Users can't tell what to e
prone.

What it means

Controls are announced
box." (WCAG 1.3.1)

Where

#:r19:, input.MuiSelect-na

Prohibited / Conditior Attributes

Lighthouse

ARIA Medium

"`aria-label` cannot be used o
attribute."

Why it matters

Clutters the accessibility
announce noise.

What it means

aria-label is placed on no
without a valid role or p

Where

Decorative UI spans

Unsized / Imprope

Lighthouse

alt attribute.

; any conveyed context

alternative (WCAG

c85

(non-interactive containers with tabindex).

Why it matters

Ideally every focused element does something with Enter/Space. Non-controls never receive focus.

What it means

This creates false affordances and slows navigation.

Where

Individual Pollutants row, Header, Compare Monitors tab

Media Performance

Desktop "unsized images" sc

Why it matters

Small layout shifts (CLS) passes.

What it means

Images render without i

Where

Header logo and miscel

Open Accessibility Report ↗

Color Accessibility Testing

ed Blind)

Monitor Markers
Legend Icons
Primary CTA Buttons

nnot distinguish warning levels or selected states.
r or alert) is lost.

n the AQI category markers (e.g., Moderate / Unhealthy
tical in brightness.

ons lose contrast against the neutral background.

lends with inactive tabs — difficult to perceive which

nd)

y Action Buttons
Comparison Graph Series

selected tabs, links, and data trends) become visually

abs, accent highlights) look grayish or muted.

es and link hover effects lose visibility.

and gray (common in charts) appears identical.

Deuteranopia (Green Blind)

Where it occurs

AQI Range Indicators
Chart Legends
Success Banners

Effect

Users with green-blindness cannot differentiate early-w
"safe" vs "neutral" statuses.

Observed Issues

Green and yellow hues used in the Good and Moderat
identical.

The comparison chart lines overlap visually

Success indicators or confirmation messages (often gr
background.

Achromatopsia (Complete Color Bli

Where it occurs

Everywhere
Map Clusters

Effect

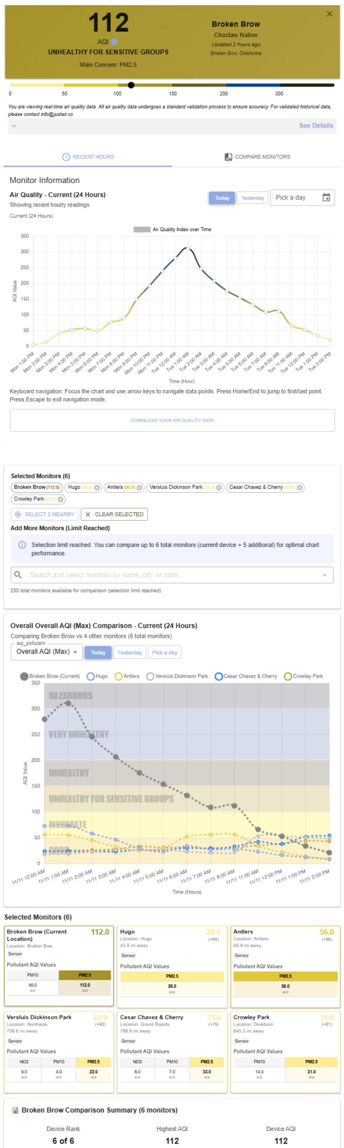
Without color or contrast cues, users cannot determine
selection at all.

Observed Issues

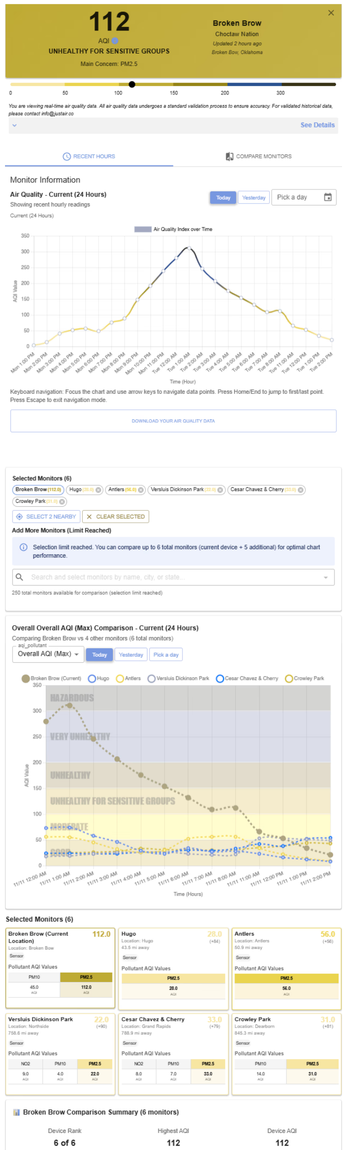
All color distinctions collapse into shades of gray.

AQI status colors (green/yellow/orange/red/purple) be
grayscale with similar luminance.

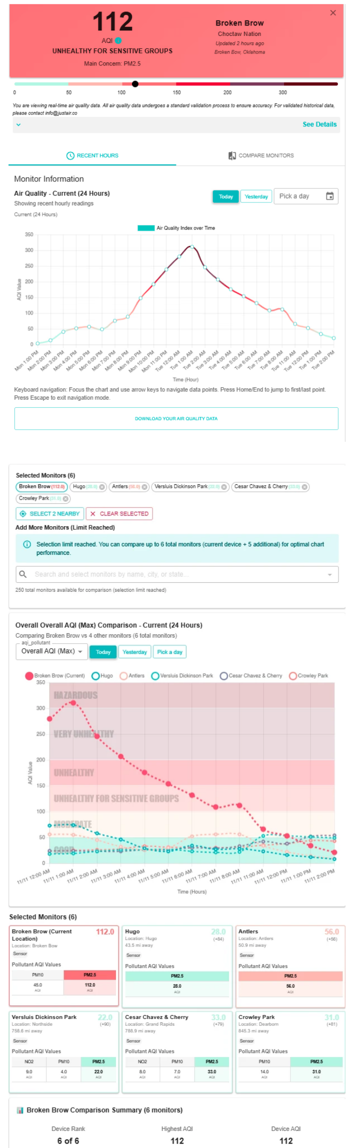
Button highlights and legends depend entirely on color
indicators.



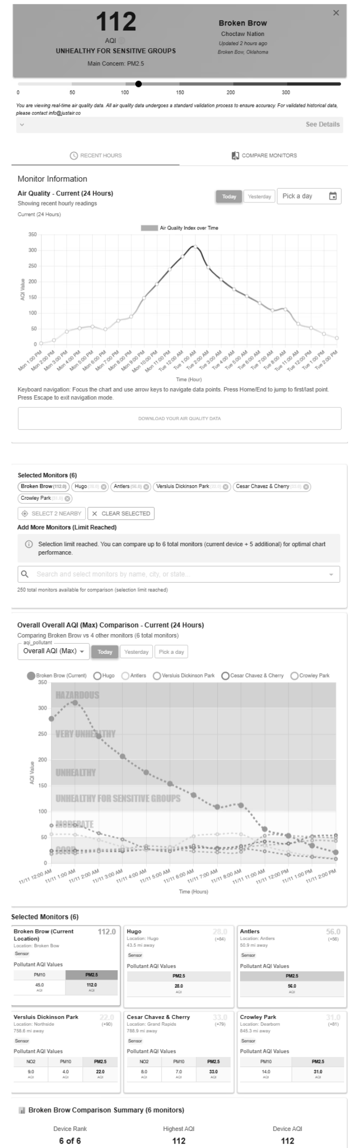
Protanopia



Deuteranopia



Tritanopia



Achromatopsia

Cividis

Viridis

Air Quality Index		
AQI Category and Color	Index Value	Description of Air Quality
Good Green	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Moderate Yellow	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Unhealthy for Sensitive Groups Orange	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Unhealthy Red	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Very Unhealthy Purple	201 to 300	Health alert: The risk of health effects is increased for everyone.
Hazardous Maroon	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

EPA verified palette

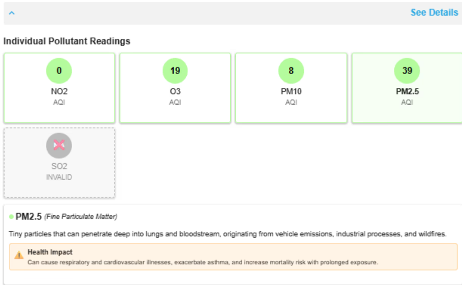
For color accessibility issues like these, and since the JustAir dashboard is very data-heavy, I would normally suggest using **Cividis** or **Viridis** color palettes. But we can't use them here because the **EPA color palette** needs to stay the same for air quality reporting. So, we need to find **accessible alternatives** that still follow the EPA guidelines.

UX Analysis

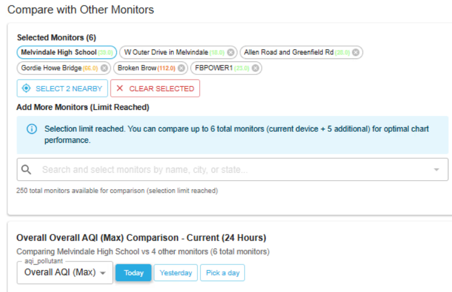
After the accessibility testing, I wanted to be certain I understood all the issues within the dash. Automated tools can't really capture contextual problems or the more nuanced difficulties users may run into.
I broke the user-facing monitor dashboard into three sections:

1. The pollutant row
2. The "compare monitors" tab
3. The graphs

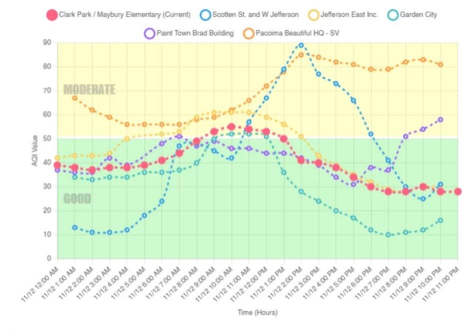
This lets me understand how each section contributes to the overall user experience, identify where friction occurs, and see how well the interface supports different user tasks.



Pollutant Row



Compare Monitors Tab



Graphs

Pollutant Row Redesign

urly Graphs

osure Interactivity

utant now sits at the top
th clear status text and
cription.

earn More" button that
es for deeper

Measurement Units Added

Units Scientific Accuracy Consistency

Each pollutant value now includes the correct measurement unit (μg/m³ or ppm), giving readings more context and scientific clarity.

Clearer Labeling

Clarity Affordance/Fee

The expandable section
Pollutants" instead of "S
obvious what users will fi

I also added a selector bo
is active or selected, impr
feedback.

Old Design Findings

Default

on Hierarchy | UX Flow

sed initially, hiding key
ual pollutant readings,
ant, and related

n the AQI category
Unhealthy ranges)
al in brightness.

dence and

ues

Inclusivity

n the AQI category
Unhealthy ranges)
al in brightness.

n the AQI category
Unhealthy ranges)
al in brightness.

Unclear Section Title

Clarity | Content Design | I/A Naming

The section currently says "See Details" without specifying what details are shown.

Red and orange tones in the AQI category markers (e.g., Moderate / Unhealthy ranges) appear nearly identical in brightness.

Hidden Hourly Pollutant Trends

Data Viz | Findability | Time-Series | IA

Hourly pollutant data exists but isn't directly visible within the pollutant tab.

Red and orange tones in the AQI category markers (e.g., Moderate / Unhealthy ranges) appear nearly identical in brightness.

Inaccurate Health Messaging

Accuracy | Content-D

Trust/Credibility

The Health Impact bar
continues to show neg
even when AQI values

Red and orange tones
markers (e.g., Modera
appear nearly identical in br

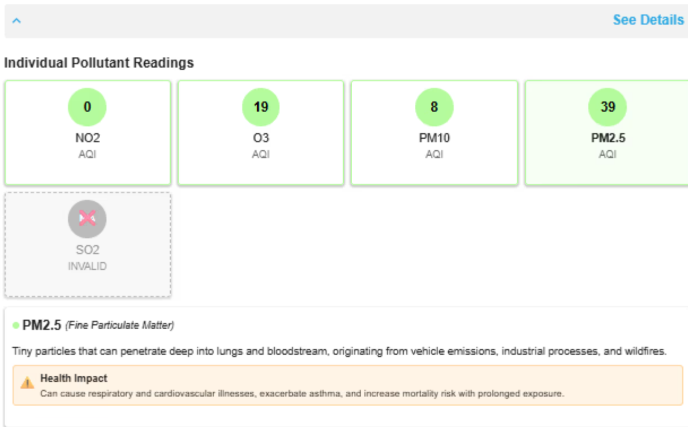
Primary AQI Dr Identified

Clarity | Labeling | Co

Information Architectur

The main pollutant driv
vaguely at the top as t
bolded in readings, bu

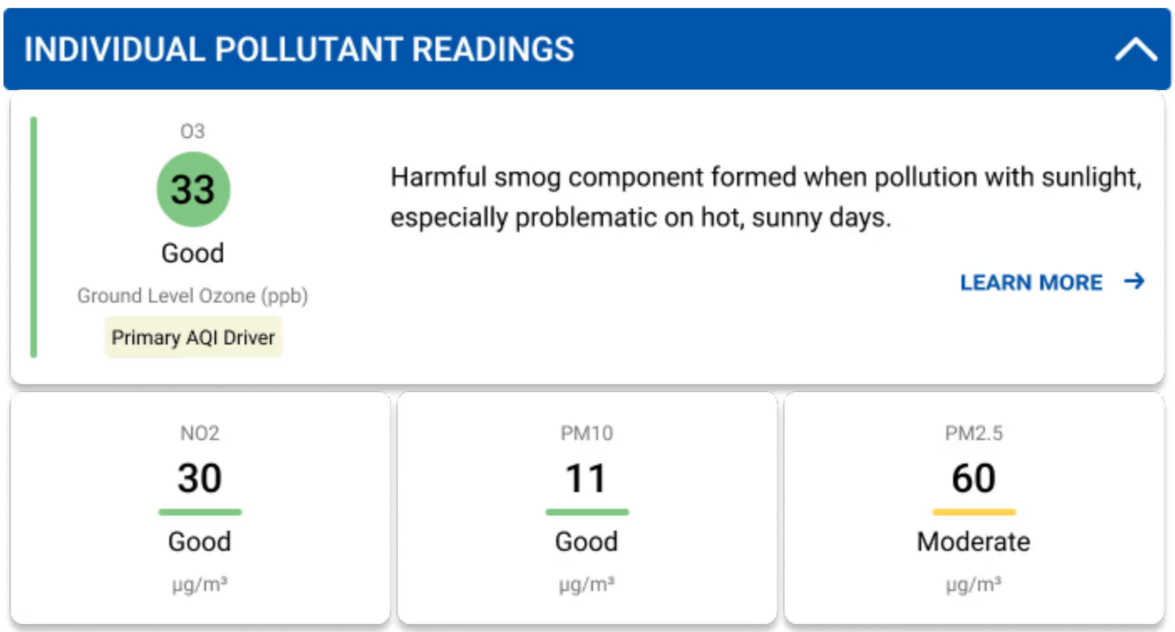
Red and orange tones
markers (e.g., Modera
appear nearly identical in br



Pollutant Row

Header

The Redesign



Redesign of Pollutant Row

Clear Primary AQI Driver Placement

Clarity | Visual Hierarchy | Status Communication | Education

The main AQI-driving pollutant now sits at the top of the row, highlighted with clear status text and more space for a short description.

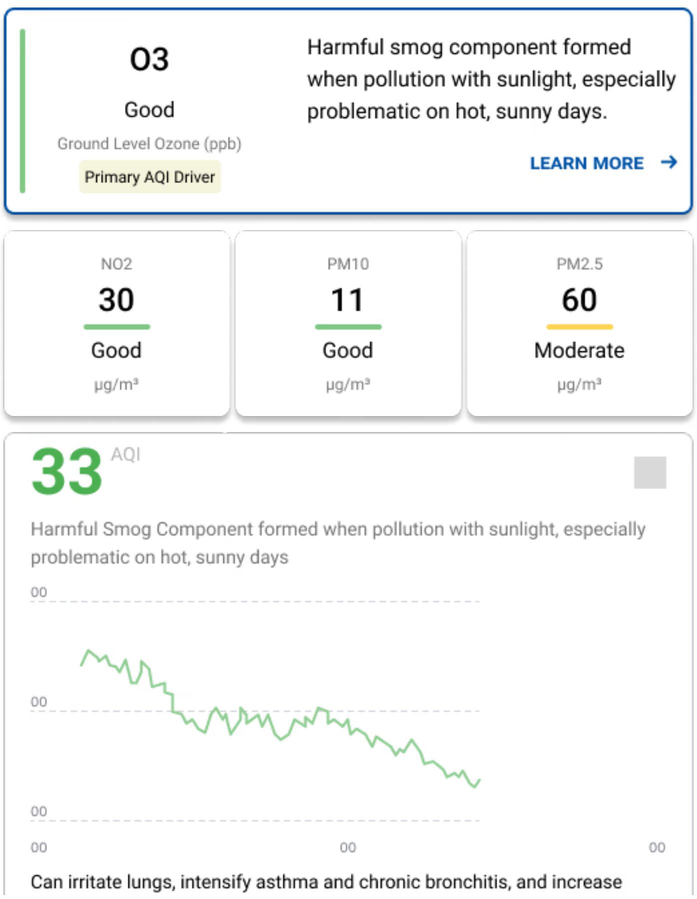
There's also room for a "Learn More" button that can link to external resources for deeper explanations.

Redundant Visual Cues for Accessibility

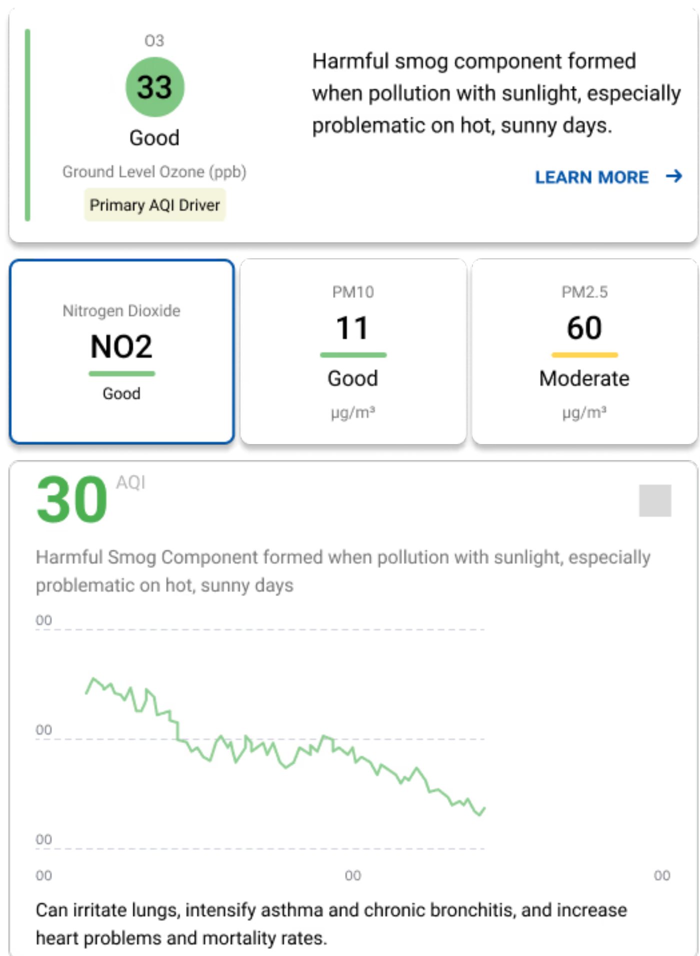
Accessibility | Redundant Cues | Status Labeling

Every pollutant card now has both color indicators and text labels (like "Good," "Moderate," "Unhealthy").

This provides redundant ways to understand status, so even users with color-vision deficiencies can easily interpret it.

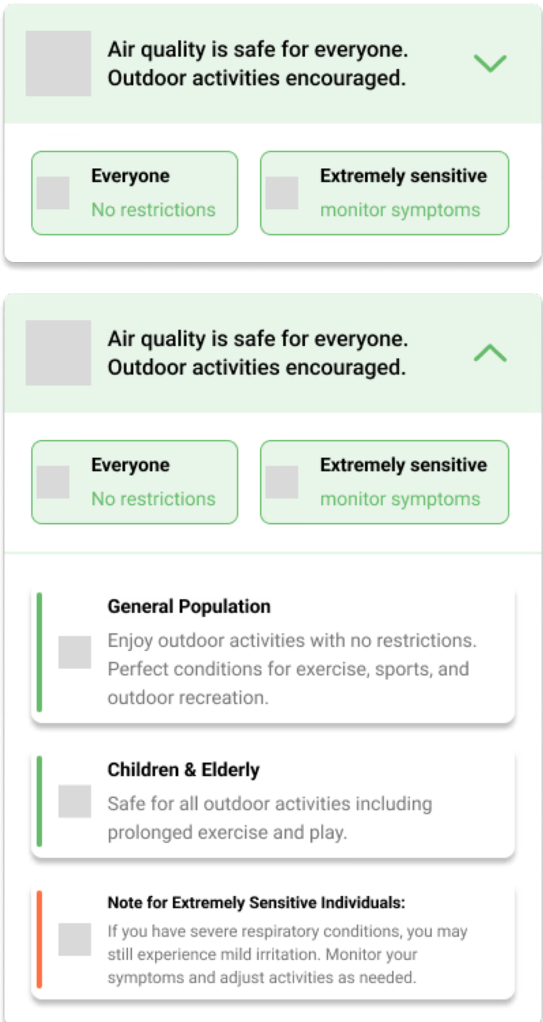


Redesign Primary AQI Driver Interaction

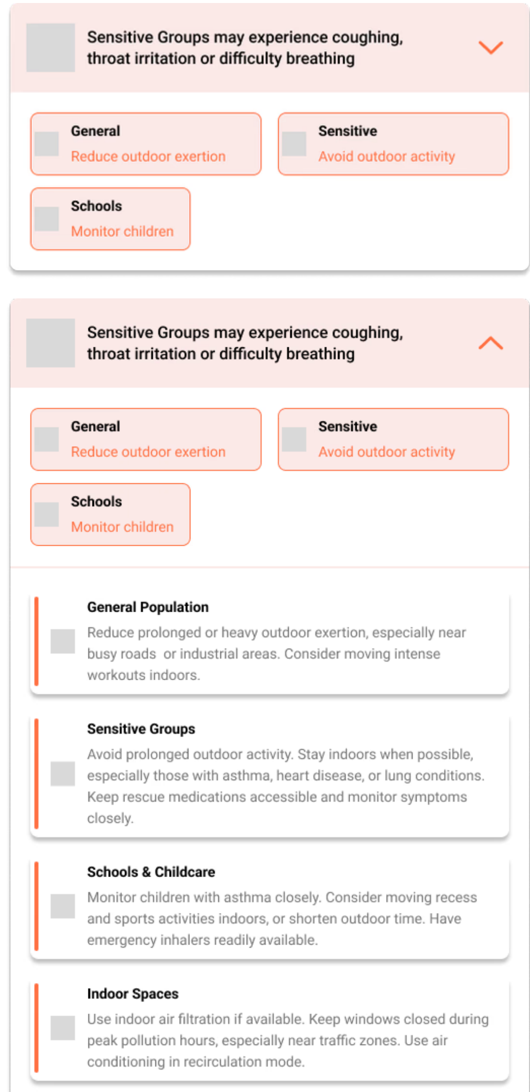


Redesign Individual Pollutants Interaction

New Addition



Redesign "Safe" Health Messaging



Redesign "Hazardous" Health Messaging

Dynamic Health Messaging

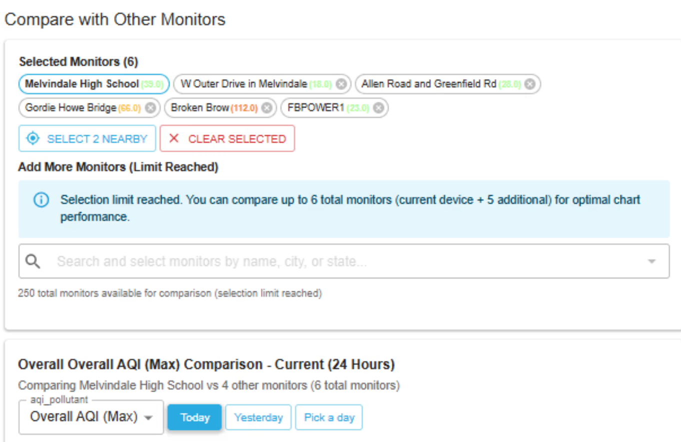
Accuracy | Content-Data Alignment | Contextual Messaging | Risk Communication

The health message below each monitor now updates based on the AQI level, so users see context-specific guidance instead of the same generic warning.

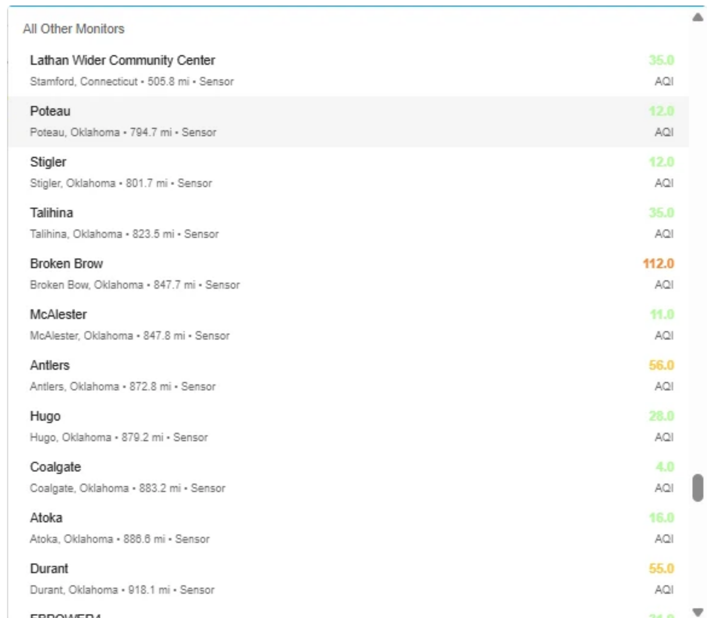
For example, safe levels show reassuring messages, while higher levels provide relevant health cautions.

Compare Monitors Redesign

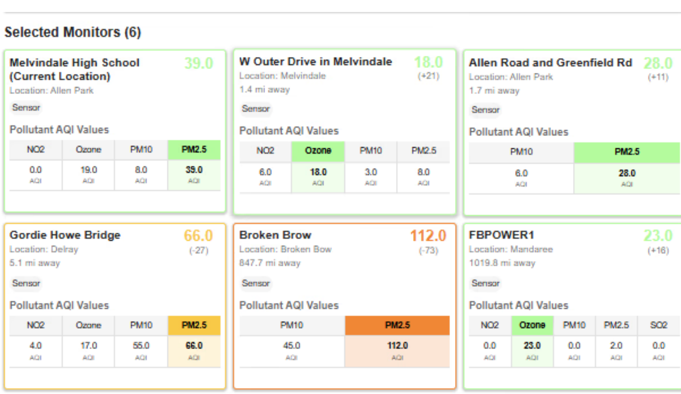
Old Design Findings



Compare Monitors Tab



Monitor Search Results



Expanded Compare Monitors

ded layout

o many controls,
ed together with minimal
hing stands out.

Dropdown label on the border

The AQI pollutant dropdown uses a label that sits on the bounding box (floating label style) in a way that clips/overlaps, reducing clarity and accessibility.

Hue-only series dif

Monitor series are distingui
fails color-blind scenario
reading.

s

et contrast guidelines,
o read against their

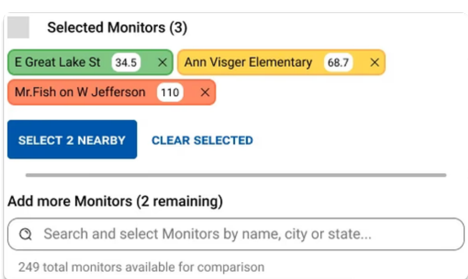
Repetitive “AQI” labeling

“AQI” appears repeatedly in line items and headers, creating label noise without adding meaning.

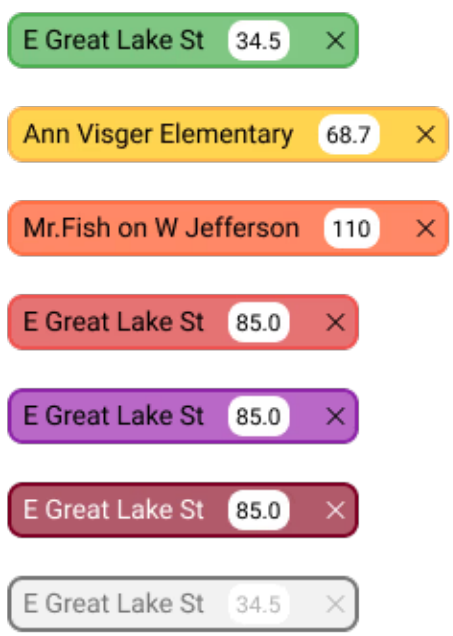
Hard-to-see AQ suggestions

The chips/suggestions r
many cases, the AQI va
at a glance.

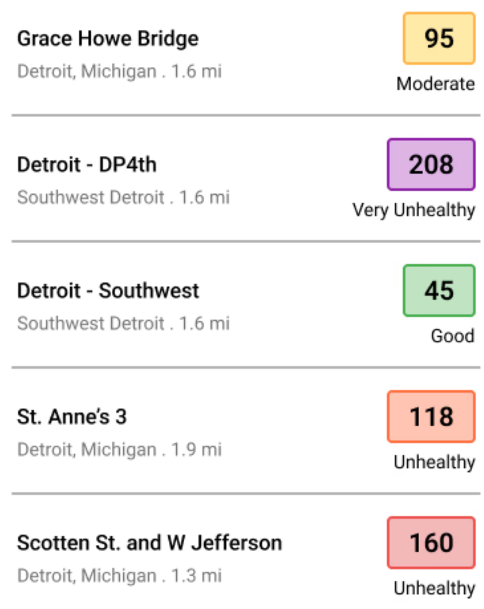
New Design



Redesign Compare Monitors Tab



Updated Color Palette



Updated Search Results

Monitor search options

Redesigned the search results so AQI numbers have high contrast and remain readable against the input background.

Removed the repeated “AQI” text and added a concise secondary label that states the air-quality status (e.g., “Good,” “Moderate”).

Monitor tags & toolbar layout

Increased spacing and improved contrast so the tags do not feel cramped next to the “Select 2 Nearby” and “Clear Selected” buttons.

Updated the tag design to improve contrast so the aqi number is clearly visible

Expanded view for selected monitors

Removed redundant “AQI” labels and strengthened the contrast of the numbers so the key values stand out.

Added a secondary status label for each monitor so users can confirm meaning at a glance without relying only on color.

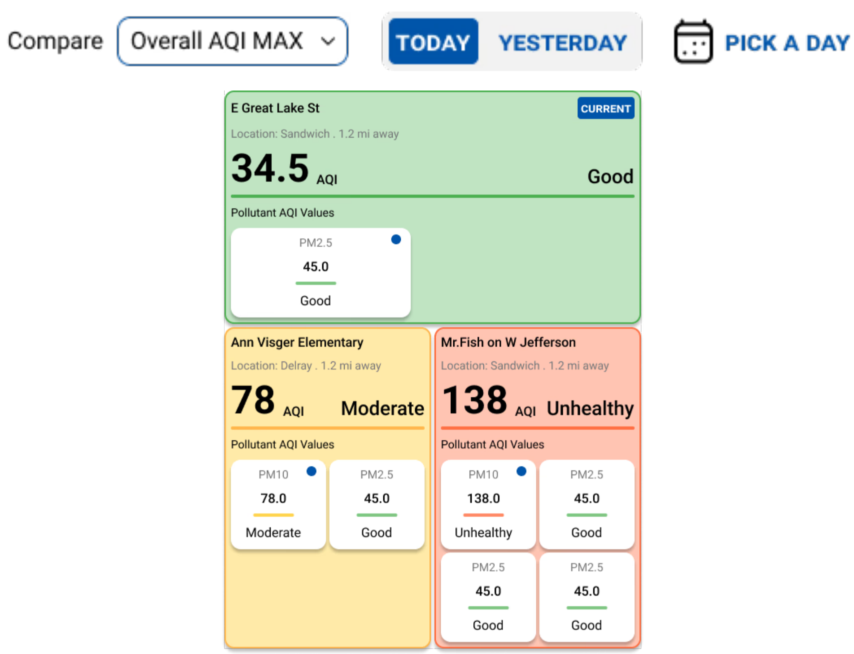
Added a “Current ” label for the primary monitor

Individual pollutants inside the expanded view

Replaced the tight table layout with roomier dialog-style cards to improve readability for values, units, and labels.

Added explicit status labels to support users with color-vision differences, so severity is clear even without color.

Introduced a circular badge to highlight the primary pollutant, which makes the main driver immediately visible. (We do this in the previous design too.)



Updated Expanded Compare Monitors

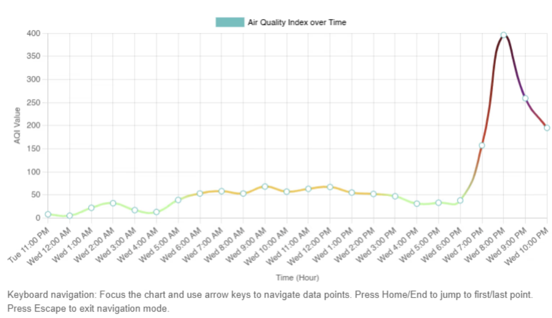
Graphs

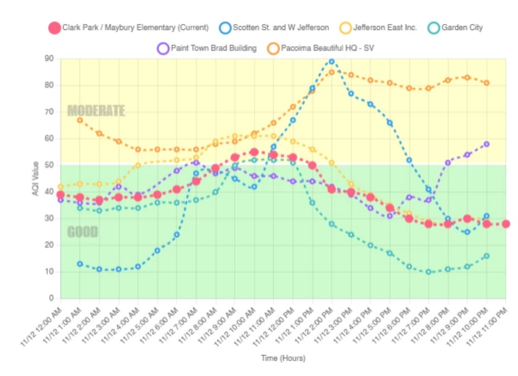
Improving Graph Accessibility and Consistency

We currently use two types of charts across the JustAir platform

- the default view and
- the compare charts.

Both share similar usability and accessibility issues:





Default View

Compare Monitors View

l Overload

rts display too much information at once . With dense
-axis labels, low-contrast label colors, and excessive
s that add visual clutter, making it harder to focus on
ıhts.

Color-Accessibility Gaps

Color-blindness testing reveals that the chart legends r
solely on color to distinguish data series. Without text, p
or shape differentiation, users with color-vision deficienci
cannot tell the lines apart.

ısistent Chart Styles

ferent chart styles are used in different sections, which
visual consistency. Unifying all graphs across the
n creates a more predictable user experience and
s accessibility.

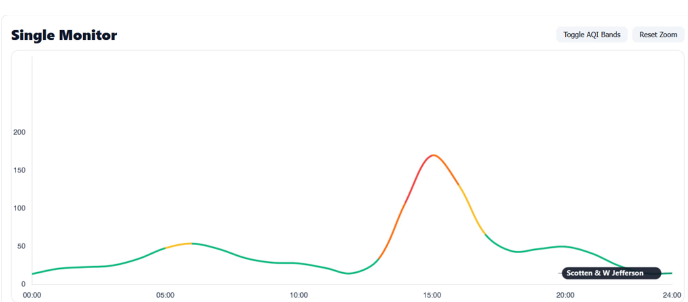
Cognitive Simplicity for Quick Viewing

For casual or at-a-glance users, displaying a plot point
every hour feels overwhelming and difficult to interpret
Reducing visual density or grouping data improves read
and comprehension.

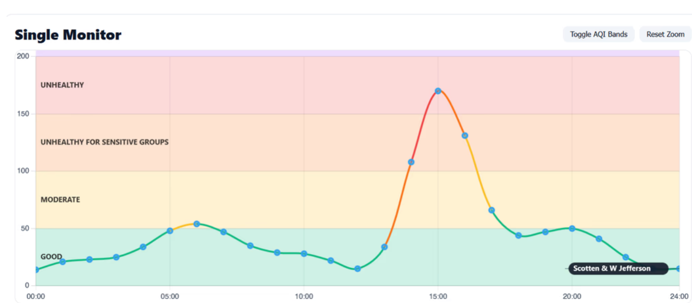
oard Navigation Support

phs already include basic keyboard navigation, which
ıg accessibility feature. My goal is to build on this
ion to make interaction smoother and more intuitive.

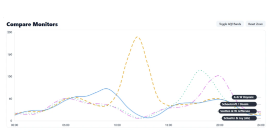
The Redesign

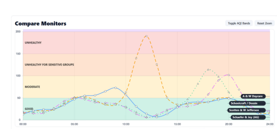


Redesign Default View



Redesign Default Hover/Click View





Redesign Compare Monitors View

Redesign Compare Monitors Hover/Click View

y Across Graphs

ssues, I focused on making all charts visually consistent.

ocks from the Compare Charts and applied them to ell. This creates a unified design language and helps QI levels across the platform.

cessibility for Color-Blind Users

aled that relying solely on color in the Compare ult for some users to distinguish between plotlines.

-based differentiation using line styles.

or's data is visually distinct even for users with red- sion deficiencies.

-based differentiation (e.g., squares, triangles, circles). But if rns are better for continuous data and easier to

Simplified and Responsive Chart Sta

The charts now include distinct interaction states:

Default State: A clean, simplified view for quick scanning.

Hover State: Highlights relevant data points and grid lines on ho clarity.

Clicked State: Allows for Keyboard navigation and Display interactive labels.

This tiered interaction reduces noise while still offering de

Reducing Visual Clutter

I streamlined the axes and grid system to make the data easi

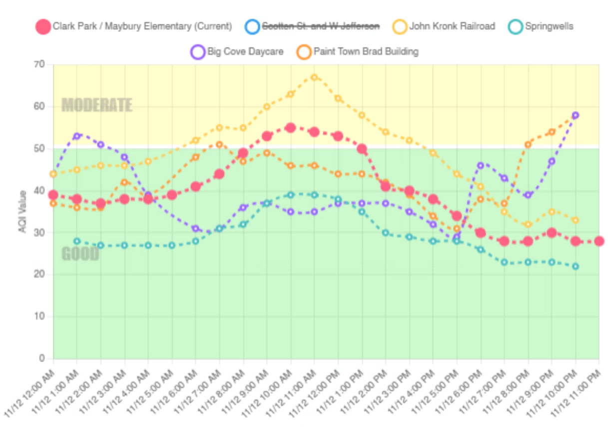
X-axis labels now appear every 5 hours instead of every

Y-axis labels appear only at standard AQI thresholds (5

Grid lines are removed from the default view and only appe

Each plotline is labeled directly on the line itself, so users forth between the chart and the legend.

These changes significantly improve readability and redu



Redesign Graph Interaction

Old Graph Interaction

Refining Legend Interactions

The Compare Monitors legend interaction is now intuitive and aligned with standard data-visualization behavior

Previously, clicking a label excluded that series and showed the rest, which felt reversed.

Now, clicking a legend label isolates that series, hiding the others.

Users can click additional labels to re-include them.

A "Reset View" button restores all series and returns the chart to its original zoom or position.

This aligns with how most analytics tools like Plotly, Chart.js, and Google Charts handle legend toggling.

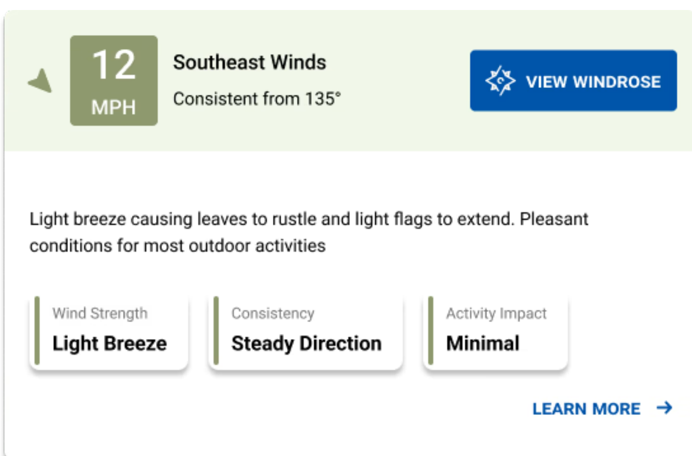
View Redesigned Graphs ↗

Wind Information Panel

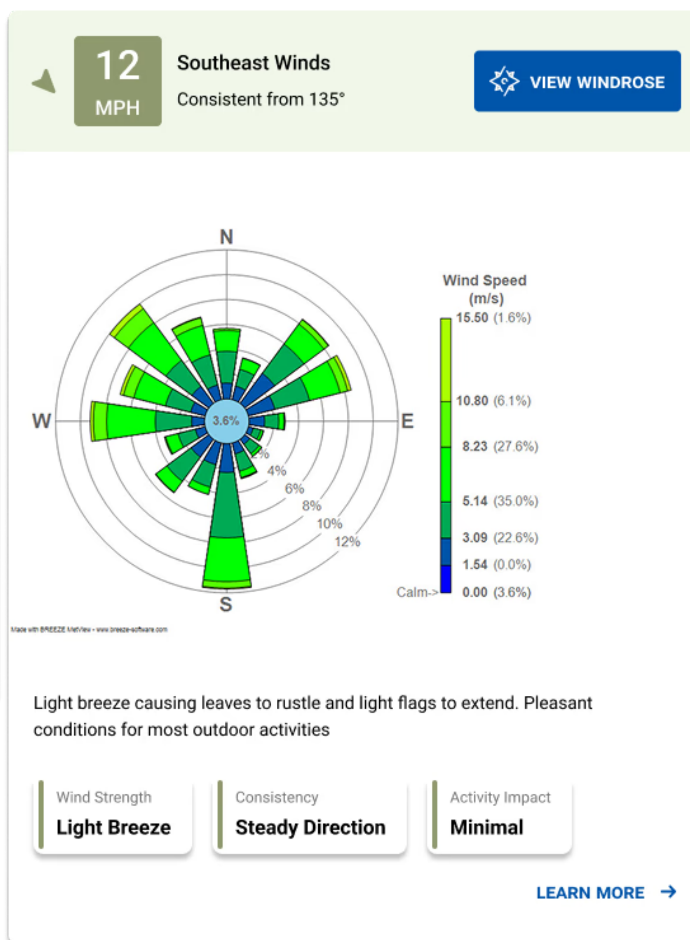
I wanted to find a balance between presenting enough wind information and keeping it easy to understand at a glance.

Wind roses were a required element, but showing a full traditional wind rose upfront can feel overwhelming or confusing for many users.

The goal was to offer clarity for people who are new to this kind of data while still providing the depth that more experienced users expect.



Wind Information Panel

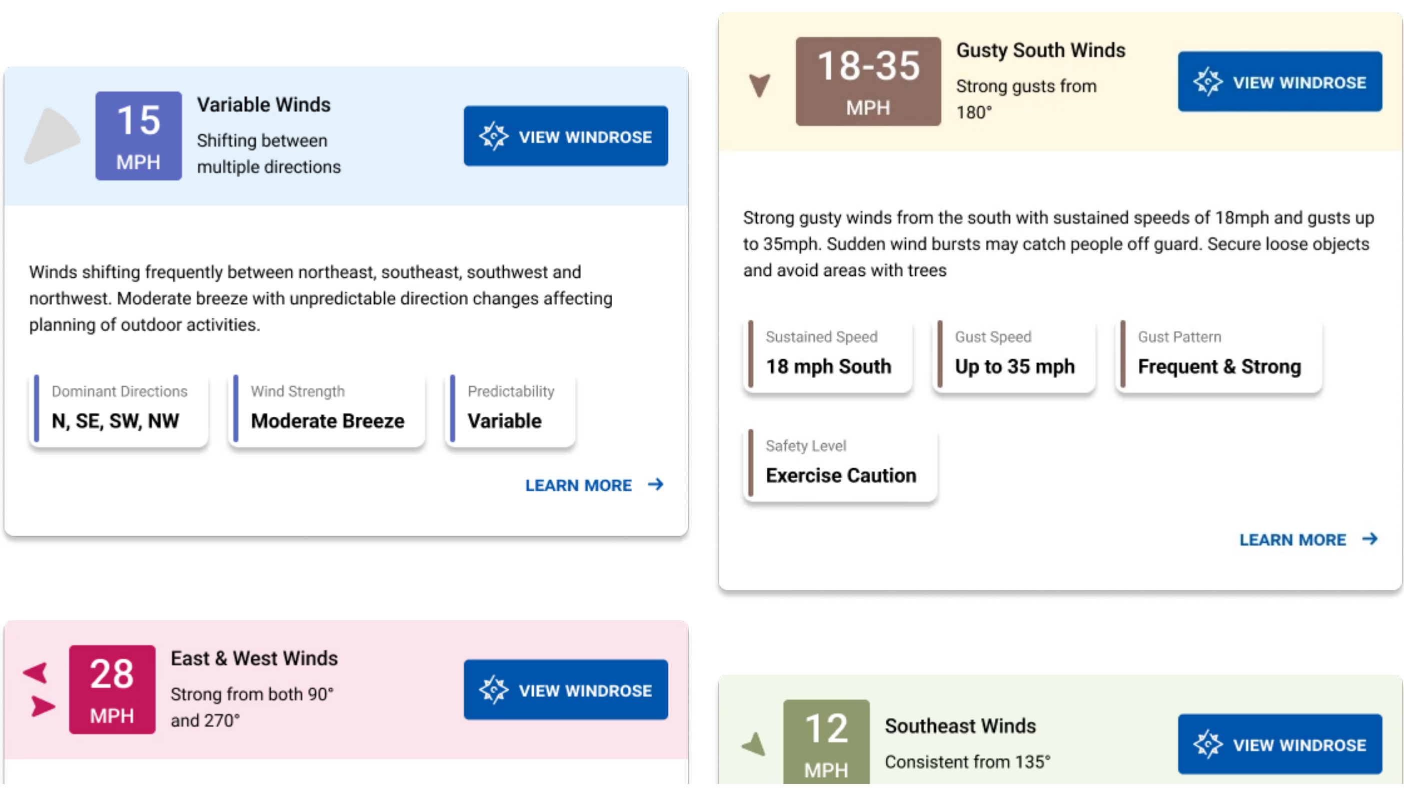


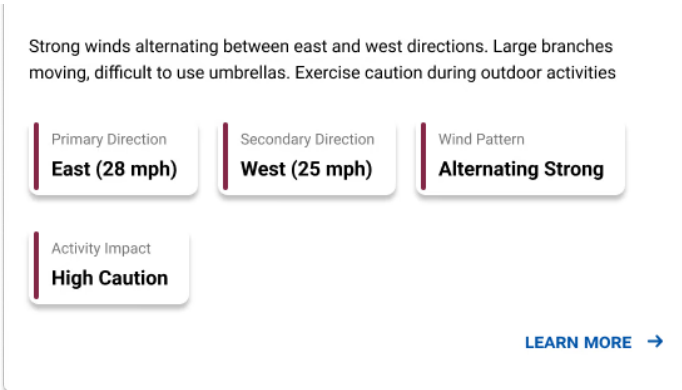
Expanded Wind Information Panel

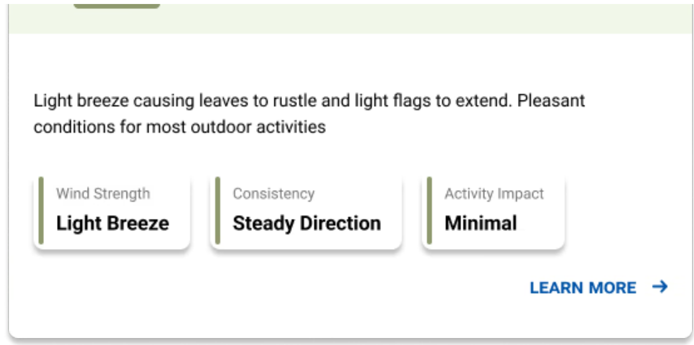
The dashboard shows wind speed and direction upfront, along with a brief contextual message to help users quickly understand current conditions. Additional details about wind behavior appear beneath that, and a button gives users the option to open the full wind rose if they want deeper insights. This structure supports both quick scanning and more advanced exploration, helping users understand real wind patterns that aren't immediately obvious.

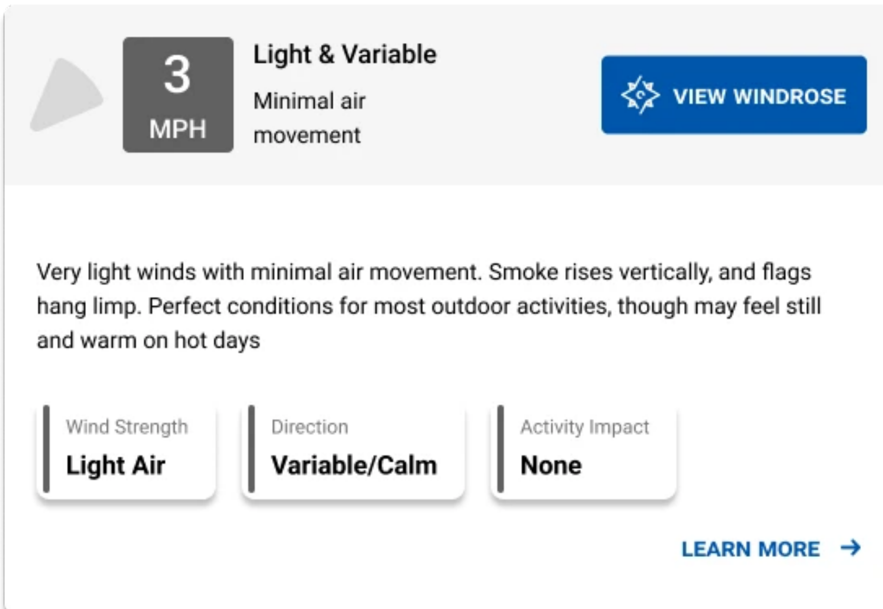
I believe this layout helps users gradually learn how to interpret wind roses by connecting them to the surrounding context. It encourages interactive learning, allowing users to explore and compare patterns to better understand real-world wind behavior. Over time, users can develop the confidence to read wind roses on their own, even without relying on the supplemental explanations.

Below are the different types of wind dialog cases I considered, including edge cases where multiple winds blow with similar strength. The goal here is to make sure the panel covers typical conditions and those less obvious, balanced cases."




Strong winds alternating between east and west directions. Large branches moving, difficult to use umbrellas. Exercise caution during outdoor activities
Primary Direction
East (28 mph)
Secondary Direction
West (25 mph)
Wind Pattern
Alternating Strong
Activity Impact
High Caution
LEARN MORE

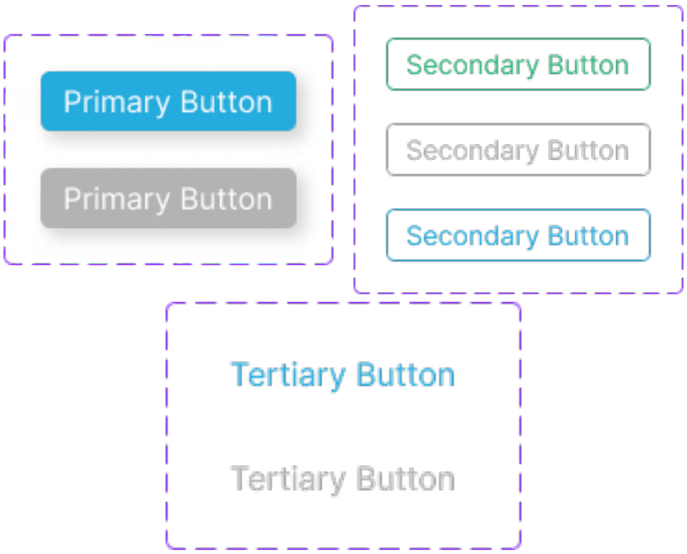

Light breeze causing leaves to rustle and light flags to extend. Pleasant conditions for most outdoor activities
Wind Strength
Light Breeze
Consistency
Steady Direction
Activity Impact
Minimal
LEARN MORE


3
MPH
Light & Variable
Minimal air movement
VIEW WINDROSE
Very light winds with minimal air movement. Smoke rises vertically, and flags hang limp. Perfect conditions for most outdoor activities, though may feel still and warm on hot days
Wind Strength
Light Air
Direction
Variable/Calm
Activity Impact
None
LEARN MORE

Global UI Improvements

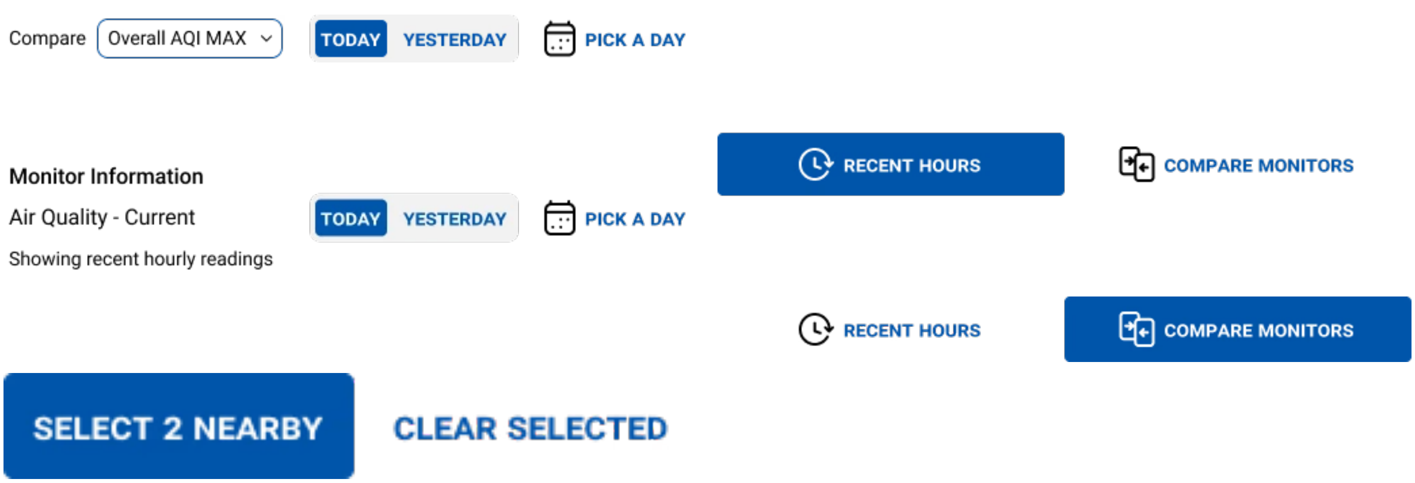
Before

After



I updated the dashboard's primary blue to to preserve the JustAir look while passing contrast on light and white surfaces. I wanted to keep the brand hue, so I searched for a nearby blue that meets ; this hit the balance between familiarity and legibility.

From there, I rebuilt the design system.



Colours

PRIMARY

0055AA 29B479

NEUTRAL

767676 B3B3B3 F2F2F2

SECONDARY

81C784 FFD54F FF8A65 E57373 BA68C8 B46BAC

4CAF50 FFB74D FF7043 EF5350 8E24AA 4A148C

TRANSPARENCY

This rule can be applied to every color seen on these palettes.
Same rule should also be applied to components like buttons, icons, etc.

100% 75% 50% 25%

Effects

SHADOWS

Shadow 1

ROUND CORNERS

4px 8px 12px 16px

Sizes & spacings

Use multiples of 4. Apply it to typography, buttons, spacings, paddings etc.

Thanks for stopping by !

Say hi

sannapar@msu.edu

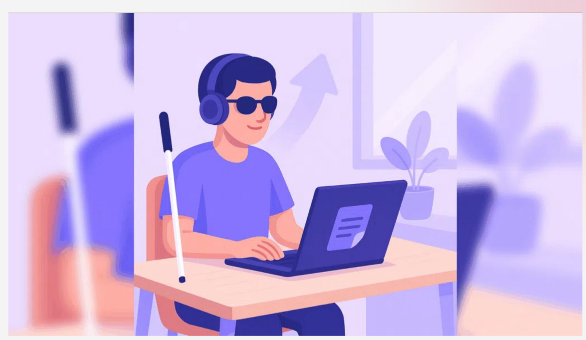
Appendix B

Graduate Student UX Expert's final presentation of his work for JustAir

2025

JustAir UX Redesign

What is Accessibility Testing?



Accessibility testing checks whether your product can be
- Perceived
- Operated
- Understood and
- Reliably used

by people with disabilities. Guided by the Web Content Accessibility Guidelines (WCAG) standard (levels A/AA/AAA).

Why Accessibility Testing?

It serves a huge audience. An estimated **1.3 billion people** (about 16% of the world) experience significant disability. Excluding them leaves users (and revenue) on the table.

W3C's WAI highlights that accessible products **extend market reach, enhance brand, drive innovation, and reduce legal risk**. Accessibility improvements often raise overall UX quality for everyone.

It improves quality & performance. Meeting WCAG typically pushes toward semantic HTML, keyboard support, captions, and better color usage. Patterns that can improve usability, SEO, and maintainability

WCAG Contrast 101

How easily text or important elements can be seen *against their background color*, based on the **Web Content Accessibility Guidelines (WCAG)**.
It ensures people with **low vision, color blindness, or poor lighting conditions** can still read and interact with content.

Contrast is measured as a **ratio** between the brightness (luminance) of foreground and background colors.

Text Type	WCAG 2.1 Level AA	WCAG 2.1 Level AAA	Example
Normal text (<18pt or <14pt bold)	4.5:1	7:1	Regular paragraph
Large text (≥18pt regular or ≥14pt bold)	3:1	4.5:1	Headings
Non-text elements (icons, charts, UI borders)	3:1	-	Buttons, charts, etc.

Some Examples

❌ Bad

Gray on White

Contrast Ratio: **1.6 : 1**

Yellow on Gray

Contrast Ratio: **1.2 : 1**

Red on Black

Contrast Ratio: **4.0 : 1**

✅ Good

Black on White

Contrast Ratio: **21 : 1**

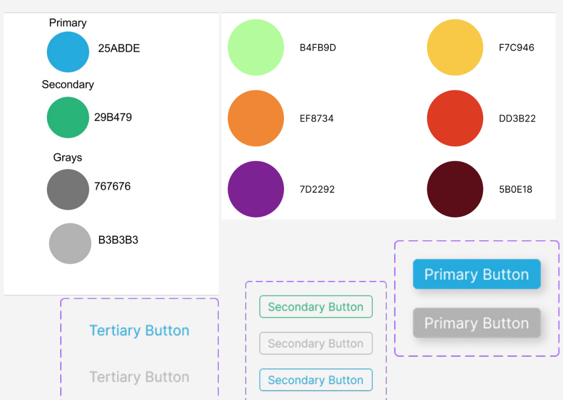
Navy Blue on Yellow

Contrast Ratio: **10.7 : 1**

Gray on off White

Contrast Ratio: **15.3 : 1**

Just Air WCAG Contrast testing



WCAG FAIL < 3.0 (normal text)

Prioritize these—users may struggle to read them in common conditions.

Contrast	Background / Text (with swatches)	Where it appears (examples)
1.2:1 Fail	#b4fb9d / #ffffff	Map badge/tiles on light green
1.2:1 Fail	#ffffff / #b4fb9d	Compare Monitors numeric chips
1.6:1 Fail	#ffffff / #f7c946	Yellow values on white
1.72:1 Fail	#767676 / #25abde	"Air Quality Map" on gray
1.72:1 Fail	#767676 / #25abde	Recent Hours header
2.19–2.42:1 Fail	#b7fca1 / #25abde 2.19:1 #f5f5f5 / #25abde 2.35:1 #f2f2f2 / #25abde 2.42:1	"Sign up / See Details / links" on light cards
2.64:1 Fail	#25abde / #ffffff #ffffff / #25abde	Primary links/buttons (Today, Compare, *Remove Cluster*)
2.8:1 Fail	#4caf50 / #ffffff	"Geoapify" green pin/badge

For the full test results please visit ->
https://deepak-s-2001.github.io/WCAG_cOLORS/wcag.html

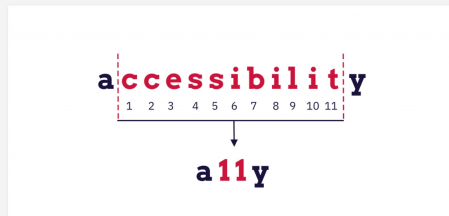
A11Y & Google Lighthouse Testing

Tools like Google Lighthouse and Axe (A11y engines) automatically test webpages for compliance with WCAG 2.1 accessibility standards.

These audits check for key usability and accessibility principles, including:

- **Alternative text for images and meaningful link names**
- **Proper form labels and input associations**
- **Logical heading hierarchy and valid ARIA attributes**
- **Keyboard navigation and visible focus indicators**
- **Clear landmark roles for screen reader navigation**

Lighthouse and A11y engines scan the DOM (Document Object Model), evaluating code structure, color contrast, ARIA roles, and semantic markup.



Lighthouse Testing Results Summary

ARIA Role Misuse

`role="group"` is not allowed for given element”
“`aria-role region` not allowed”
“Elements with ARIA `[role]` missing required children”

Meaning:
You’re using ARIA roles on semantic HTML elements (like “<article>” or “<section>”) in ways that screen readers can’t interpret correctly.
It’s like giving someone two conflicting job titles — assistive tech gets confused which “role” to follow.

Effect:
Screen readers may skip or mis-read pollutant cards or chart areas, so visually impaired users could miss the context or labels for AQI readings and charts.

ARIA Dialog Missing Name

“Elements with `role="dialog"` do not have accessible names”

Meaning:
The sidebar or monitor-details modal (“.MuiDrawer-root”) doesn’t have a clear accessible title or description element that matches “aria-labelledby”.

Effect:
Screen reader users can’t tell what the dialog contains when it opens — it just reads “dialog”.

Prohibited / Conditional ARIA Attributes

“`aria-label` cannot be used on a span with no valid role attribute.”

Meaning:
You’ve got decorative spans with ARIA labels but no interactive role or semantic purpose.

Effect:
They’re cluttering the accessibility tree — screen readers waste time announcing meaningless text.

Missing Required Parent/Child Relationships

“Element has children which are not allowed: `[role=group]`”

Meaning:
Your grid of pollutant cards uses ARIA roles that conflict (e.g., a container marked as `role=grid` with children that aren’t valid gridcells).

Effect:
Screen readers can’t navigate the pollutant grid properly — users can’t tell they’re part of the same table.

Unsized or Improperly Sized Images

Desktop “unsized images” scored ~0.5, though **mobile passed.**

Meaning:
Some `<img>` tags (like your logo) lack explicit `width` and `height`.

Effect:
This can cause **layout shifts (CLS)** — minor but noticeable content jumps while loading.

A11Y Testing Results Summary

Form Inputs Missing Labels

Multiple form controls (`#:r19:`, `input.MuiSelect-nativeInput.css-1k3x8v3`) are missing accessible labels.
These inputs have no `<label>` element or `aria-label` attribute that describes their purpose.

Effect:
Screen reader users cannot identify what information each field requires.

When navigating via assistive technologies, these unlabeled controls are announced simply as “input” or “combo box,” preventing accurate data entry.

Invalid ARIA Attributes

Several elements (`#simple-tab-2`, `div.MuiSelect-select…`, `#location-tabpanel-1`, `#location-tabpanel-0`, `div.MuiDrawer-root.MuiDrawer-modal…`) use invalid or misapplied ARIA attributes.
Attributes such as `aria-labelledby`, `aria-controls`, or roles are incorrectly assigned or reference nonexistent IDs.

Effect:
Screen readers may misinterpret, misannounce, or completely ignore these components.
Users relying on assistive technologies may hear wrong labels, lose navigation context in tab panels or drawers, and be unable to associate interactive elements with their visible counterparts.

Images Missing Alternative Text

Image element `img.MuiBox-root.css-1nbic85` has no `alt` attribute.

Effect:
Screen reader users cannot perceive or understand the purpose of the image.
The graphic is ignored entirely, meaning key context or branding information conveyed visually is lost.

Find these reports at
https://github.com/deepak-s-2001/Lighthouse-A11y-Reports

Keyboard-Only Navigation Testing

For this phase, I avoided using the mouse entirely and relied solely on keyboard controls and keystrokes to navigate through the site.
The test followed a realistic interaction flow within the app:

1. Navigate to **justair.app**.

2. Select a **state** → **region** → **cluster** → **individual monitor**.

3. Once the **monitor sidebar** opened, navigate through all interactive components — including dropdowns, graphs, date selectors, and buttons.

4. Attempt to **change dates on the graphs**, **compare monitors**, **download data**, and **interact with calendar components** using only the keyboard.

Cont..

Why do this ?

Keyboard testing ensures that all interactive elements (buttons, inputs, modals, dropdowns, and charts) are reachable, operable, and visually focused without requiring a mouse.
It helps detect issues like missing focus states, inaccessible modals, or elements that trap focus — all critical for users who rely on keyboards, switch devices, or assistive input tools.

Typical Method:

Tab / Shift + Tab: Move between focusable elements in sequence.

Enter / Space: Activate or select focused elements.

Arrow keys: Navigate within menus, sliders, or dropdowns.

Esc: Dismiss dialogs or overlays. A successful test confirms that every major action (monitor selection, data comparison, download, etc.) can be performed entirely via keyboard input.

Keyboard-Only Navigation Results

Ghost focus
(empty selectors)

What I observed
Tabbing highlights elements that don’t respond to Enter/Space (non-interactive containers with tabindex).

Why it matters
This creates false affordances and slows navigation.

What to do
- Remove tabindex from non-interactive elements.
- Only interactive controls are tabbable; everything else is skipped.

Ideally every focused element does something with Enter/Space. Non-controls never receive focus.

Screen Reader Testing

I enabled **ChromeVox**, the built-in **screen reader extension for Google Chrome**.
Screen readers like ChromeVox (on Chrome OS) or NVDA / VoiceOver (on Windows and macOS) convert on-screen elements into **synthetic speech and auditory feedback**, reading out page content, labels, ARIA roles, and contextual instructions as the user navigates.

How It Works:

When activated, ChromeVox reads aloud the currently focused element, its role (e.g., “button,” “menu,” “chart”), and any accessible text or labels attached to it.
As I navigated through the app using only the keyboard, ChromeVox announced each UI component. Confirming whether ARIA labels, roles, and relationships were correctly implemented.

Cont..

Testing Steps Performed

Navigated across all interactive elements within the sidebar.

Verified whether the **monitor name**, **AQI readings**, and **chart titles** were read correctly.

Checked that **dropdowns, calendars, and data download buttons** were recognized by the reader.

Ensured no important information was skipped or read in a confusing order.

Purpose

Screen reader (or "read aloud") testing validates that assistive technologies can interpret and vocalize the app's structure and data meaningfully.
It helps identify missing ARIA attributes, unclear labels, or improperly nested interactive elements that may block accessibility for users with visual impairments.

Screen Reader Testing Results

Dialog has no accessible name → screen reader announces just "dialog"

Tabs / tabpanels not linked correctly → labels/controls don't associate, panel context is unclear

Form control(s) missing programmatic label → announced as "input/combobox" with no purpose

Image missing alternative text → read as generic "image" (no meaning conveyed)

Unmatched / invalid ARIA references → broken name/description/controls relationships

Color accessibility Testing

What Is It?

Color-accessibility testing is a design audit that checks how a website or application appears to people with different types of **color-vision deficiencies (CVD)**.

It simulates how users with conditions such as **Protanopia** (red-blind), **Deuteranopia** (green-blind), **Tritanopia** (blue-blind), or **Achromatopsia** (no color vision) perceive the interface.

- Tools like **Chrome DevTools color filters** apply these simulations so designers can identify where color distinctions fail.

Why It's Done

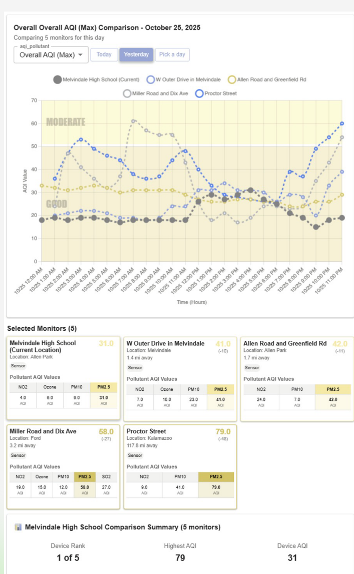
Around **8% of men and 0.5% of women** experience some form of color-vision deficiency — meaning **millions of users** can't rely on color cues alone.

Many interfaces (like dashboards, maps, and alerts) use color as the **only way to convey meaning** — e.g., green = good, red = bad.

Color-blind testing ensures that information is still clear through **contrast, text labels, icons, or patterns**, not just hue.

- It supports **WCAG 2.1** requirements for color contrast and non-color-dependent communication, preventing accessibility barriers.

Protanopia (Red Blind)



Observed issues

Red and orange tones in the **AQI category markers** (e.g., Moderate / Unhealthy ranges) appear nearly identical in brightness.

The **alert or "active" buttons** lose contrast against the neutral background.

The **selected tab highlight** blends with inactive tabs — difficult to perceive which view is active.

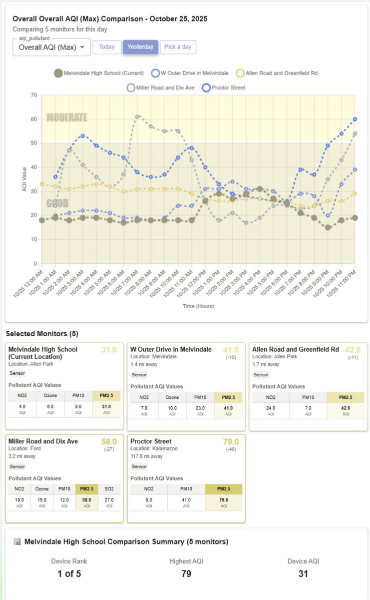
Effect

Users with red-blindness cannot distinguish warning levels or selected states — color meaning (e.g., danger or alert) is lost.

Where it occurs

AQI pollutant badges, map monitor markers, legend icons, and primary call-to-action buttons.

Deuteranopia (Green Blind)



Observed issues

Green and yellow hues used in the Good and Moderate AQI states appear nearly identical.

The **comparison chart lines** overlap visually

Success indicators or confirmation messages (often green) fade into the background.

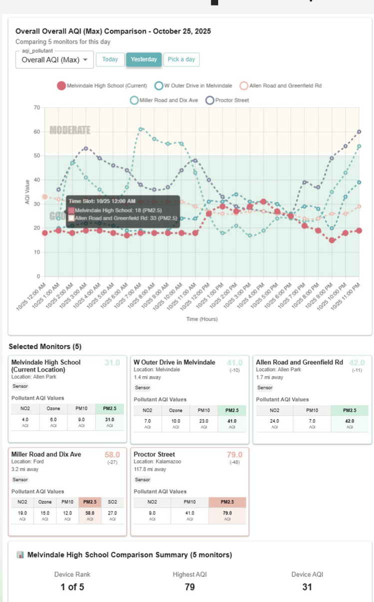
Effect

Users with green-blindness cannot differentiate early-warning pollution levels or “safe” vs “neutral” statuses.

Where it occurs

AQI range indicators, chart legends, success banners, and map clusters.

Tritanopia (Blue Blind)



Observed issues

Blue and purple elements (tabs, accent highlights) look grayish or muted.

The **secondary button states** and **link hover effects** lose visibility.

Any color pairing using blue and gray (common in charts) appears identical.

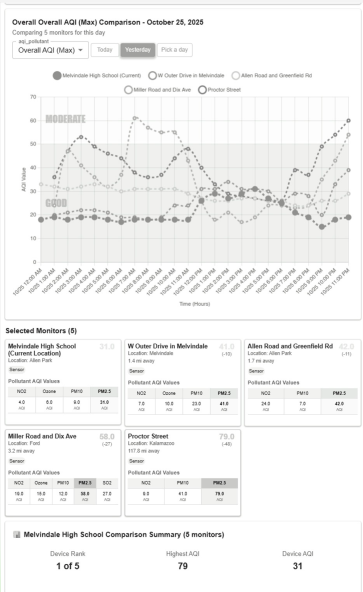
Effect

Important interactive cues (selected tabs, links, and data trends) become visually ambiguous or invisible.

Where it occurs

Navigation tabs, secondary action buttons, comparison graph series with blue/purple shades.

Achromatopsia (Complete Color Blindness)



Observed issues

All color distinctions collapse into shades of gray.

AQI status colors (green/yellow/orange/red/purple) become **indistinguishable grayscale** with similar luminance.

Button highlights and legends depend entirely on color — no text or pattern indicators.

Effect

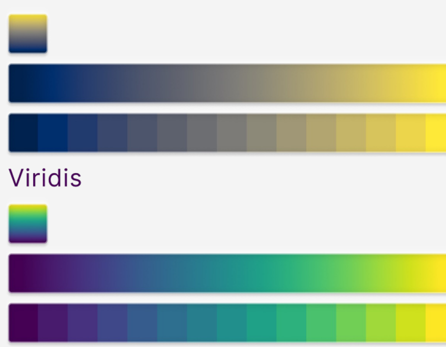
Without color or contrast cues, users cannot determine state, severity, or selection at all.

Summary

Issue Category	Description
Color Reliance Without Text / Pictorial Backup	AQI levels, chart lines, and alert states rely only on color differences. When hue perception changes, meaning disappears.
Low Luminance Contrast Between Adjacent Colors	Greens, yellows, and oranges appear almost identical; reds and browns merge visually.
Inactive vs Active Element Confusion	Selected tab or button highlight not distinct when hue is lost.
Blue / Gray Similarity in Data Lines	Blue or purple data lines in graphs appear gray and blend with background grid.
Lack of Redundancy in AQI Indicators	No text, shape, or icon describing “Good,” “Moderate,” or “Unhealthy” states — only color.
Contrast Insufficiency on Interactive Components	Borders and outlines of interactive elements are too subtle across modes.

Cividis

Viridis



AQI Basics for Ozone and Particle Pollution

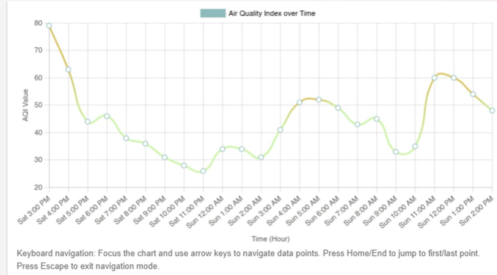
Daily AQI Color	Levels of Concern	Values of Index	Description of Air Quality
Green	Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Yellow	Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Red	Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Purple	Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

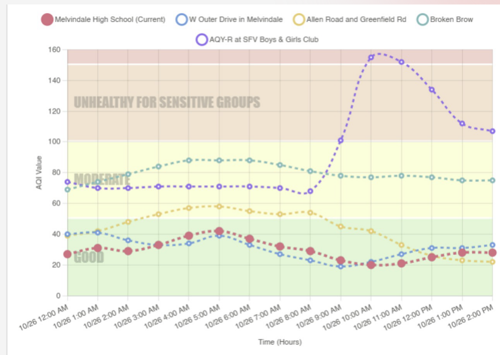
Graph Accessibility and Usability Review

We currently use **two types of charts** across the JustAir platform

- the **default view** and
- the **compare charts**.

Both share similar usability and accessibility issues:





Visual Overload

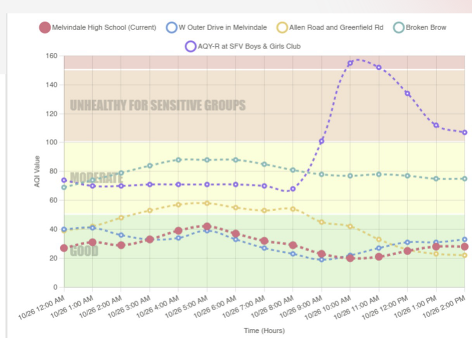
The charts display too much information at once . With dense X and Y-axis labels, low-contrast label colors, and excessive grid lines that add **visual clutter**, making it harder to focus on key insights.

Inconsistent Chart Styles

Two different chart styles are used in different sections, which breaks visual consistency. Unifying all graphs across the platform creates a more predictable user experience and improves accessibility.

Keyboard Navigation Support

The graphs already include basic keyboard navigation, which is a strong accessibility feature. My goal is to build on this foundation to make interaction smoother and more intuitive.

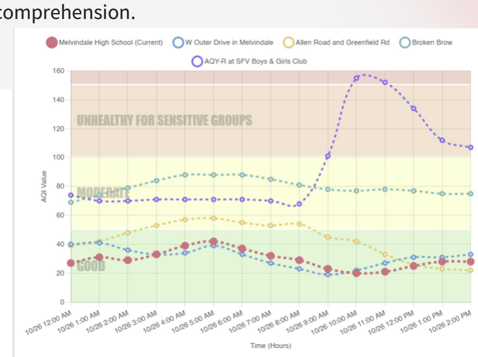


Color-Accessibility Gaps

Color-blindness testing reveals that the chart legends rely solely on color to distinguish data series. Without text, pattern, or shape differentiation, users with color-vision deficiencies cannot tell the lines apart.

Cognitive Simplicity for Quick Viewing

For casual or at-a-glance users, displaying a plot point for every hour feels overwhelming and difficult to interpret. Reducing visual density or grouping data improves readability and comprehension.

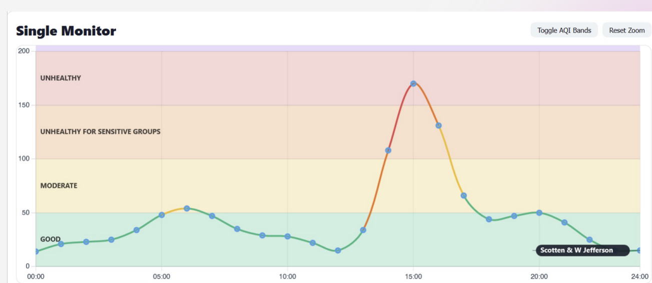


Improving Graph Accessibility and Consistency

Visual Consistency Across Graphs

To begin addressing the issues, I focused on making all charts **visually consistent**.

I adopted the **colored AQI blocks** from the Compare Charts and applied them to the **Default Chart View** as well. This creates a unified design language and helps users instantly recognize AQI levels across the platform.



Simplified and Responsive Chart States

The charts now include distinct **interaction states**:

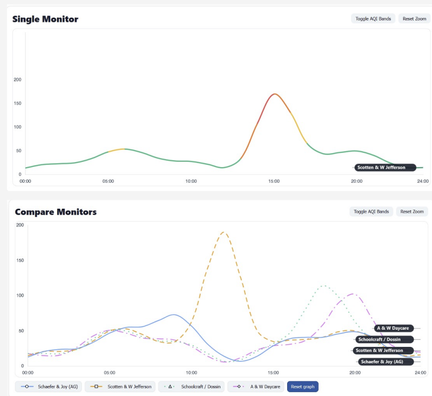
Default State: A clean, simplified view for quick scanning.

Hover State: Highlights relevant data points and grid lines on hover for contextual clarity.

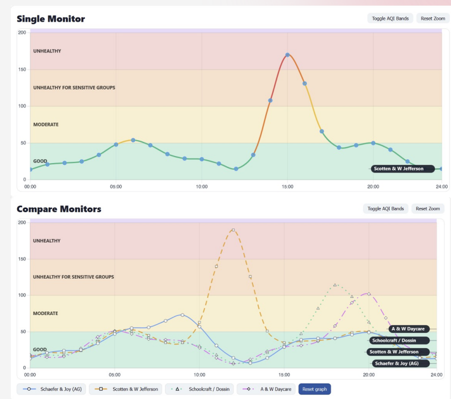
Clicked State: Allows for Keyboard navigation and Displays detailed values and interactive labels.

This tiered interaction reduces noise while still offering detail when needed.

Default View



Hover / Clicked View



Improving Accessibility for Color-Blind Users

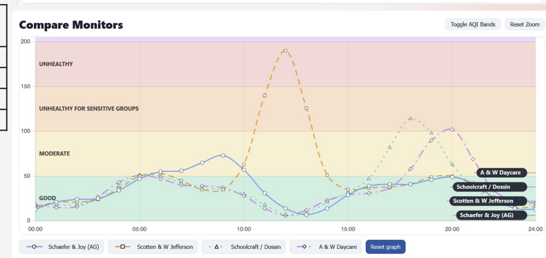
Color-blindness testing revealed that relying solely on color in the Compare Monitors chart made it difficult for some users to distinguish between plotlines.
To fix this, I added **non-color-based differentiation** using line styles.

This ensures that each monitor's data is **visually distinct** even for users with red-green or blue-yellow color vision deficiencies.

I also suggested icon-based differentiation (e.g., squares, triangles, circles). But if that's not possible, line patterns are better for continuous data and easier to maintain.

Line Styles examples

Monitor	Line Style
Monitor 1	Dotted Line
Monitor 2	Dot-Dashed Line
Monitor 3	Solid Line
Monitor 4	Dashed Line



5. Refining Legend Interactions

The Compare Monitors legend interaction is now **intuitive and aligned with standard data-visualization behavior**

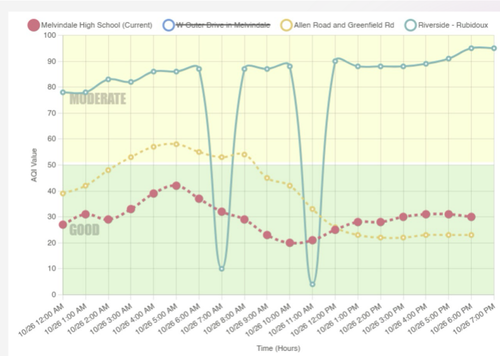
Previously, clicking a label **excluded that series** and showed the rest, which felt reversed.

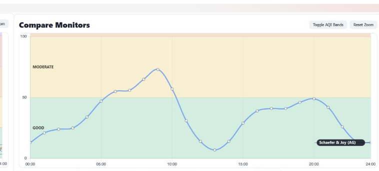
Now, clicking a legend label **isolates that series**, hiding the others.

Users can click additional labels to re-include them.

A **"Reset View"** button restores all series and returns the chart to its original zoom or position.

This aligns with how most analytics tools like Plotly, Chart.js, and Google Charts handle legend toggling.





Find these graphs and their Interaction Patterns at
https://deepak-s-2001.github.io/graphs/

Individual Pollutants Tab Review

Details Hidden by Default

The “Details” tab is collapsed initially, hiding key information such as individual pollutant readings, the main AQI-driving pollutant, and related health impacts.

Since pollutant data is more important to users than raw monitor info, this default state reduces discoverability and usability.

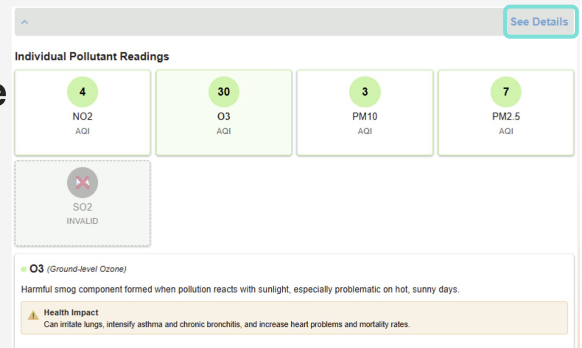
Unclear Section Title

The section currently says

“See Details”

without specifying what details are shown.

Consider adding a clear title such as “Individual Pollutant Details” or “Pollutant Breakdown”.



Inaccurate Health Impact Messaging

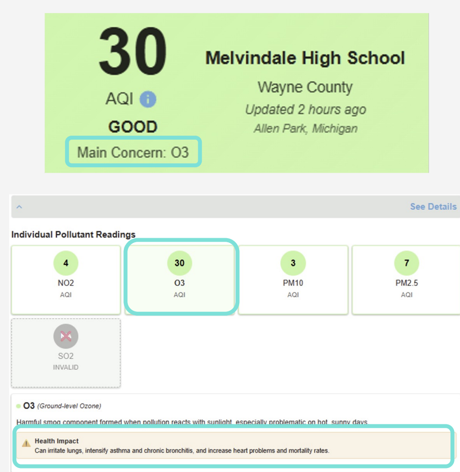
The Health Impact bar visually looks fine but continues to show negative or severe warnings even when AQI values indicate good air quality.

Text and severity level should align with actual AQI risk categories.

Primary AQI Driver Not Clearly Identified

The main pollutant driving the AQI is mentioned vaguely at the top as the “main concern” and bolded in readings, but not explicitly labeled.

Competing dashboards use a clearly marked “Primary AQI Driver” tag or highlighted card, improving clarity.



Hidden Hourly Pollutant Trends

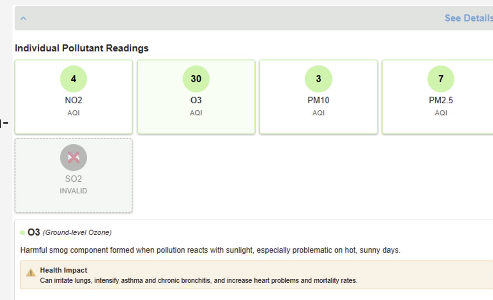
Hourly pollutant data exists but isn’t directly visible within the pollutant tab.

Relocating this data to the Pollutant Tracking section would make temporal patterns easier to explore.

Color Dependence and Accessibility Gaps

Color-blind testing reveals heavy reliance on hue to convey pollutant severity and category.

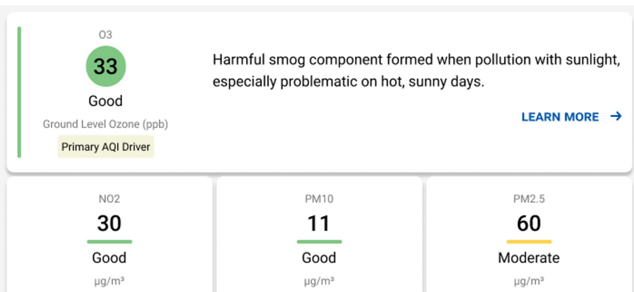
Labels and icons should provide non-color indicators (e.g., text, icons, or shapes).



Missing Measurement Units

Pollutant values are missing units such as µg/m³ or ppm, which are essential for context and scientific accuracy.

The Redesign



Clear Primary AQI Driver Placement

The main AQI-driving pollutant now sits at the **top of the row**, highlighted with clear status text and more space for a short description.

There’s also room for a **“Learn More”** button that can link to external resources for deeper explanations.

Redundant Visual Cues for Accessibility

Every pollutant card now has both **color indicators** and **text labels** (like “Good,” “Moderate,” “Unhealthy”).

This provides redundant ways to understand status — so even users with color-vision deficiencies can easily interpret it.

Expandable Hourly Graphs

Tapping a pollutant expands it into its **own mini-chart**, showing the hourly AQI trend for that specific pollutant.

This helps users quickly track how air quality changes throughout the day.

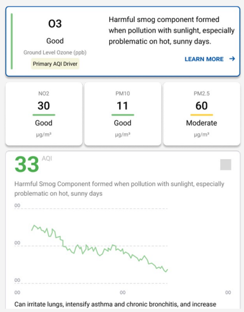
Measurement Units Added

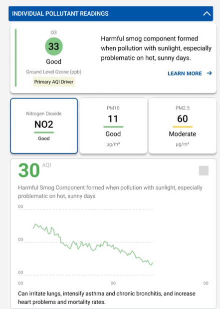
Each pollutant value now includes the correct **measurement unit** (µg/m³ or ppm), giving readings more context and scientific clarity.

Clearer Labeling and Actions

The expandable section now reads **“Individual AQI Pollutants”** instead of “See Details”, making it obvious what users will find inside.

I also added a **selector box** to show which element is active or selected, improving interaction feedback.



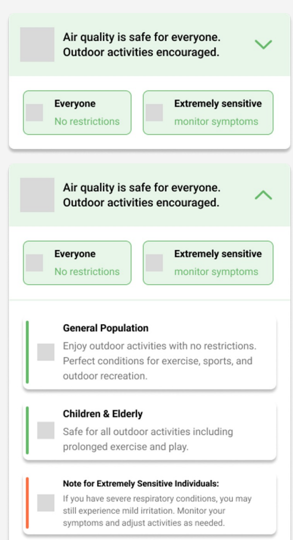


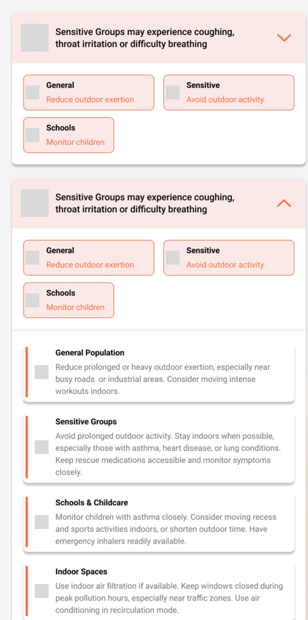
Other Ideas

Dynamic Health Messaging

The health message below each monitor now **updates based on the AQI level**, so users see context-specific guidance instead of the same generic warning.

For example, safe levels show reassuring messages, while higher levels provide relevant health cautions.





Compare Monitors Tab Review

Visually crowded layout

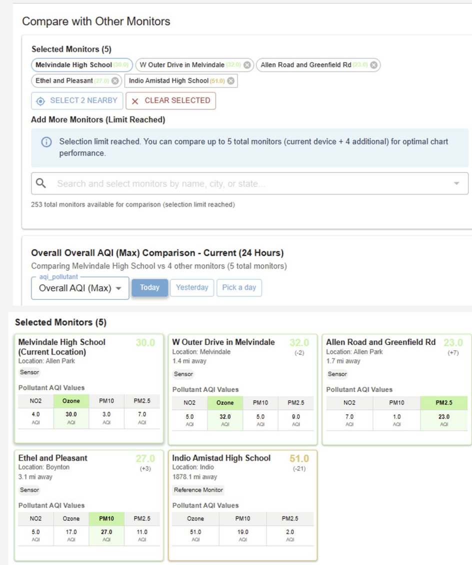
The tab feels **overloaded**, too many controls, badges, and labels clustered together with minimal spacing and contrast, so nothing stands out.

Low-contrast labels

Several label colors don't meet contrast guidelines, making **AQI numbers hard to read** against their backgrounds.

Dropdown label on the border

The AQI pollutant dropdown uses a label that sits on the **bounding box** (floating label style) in a way that **clips/overlaps**, reducing clarity and accessibility.



Compare Monitors Tab Review

Repetitive "AQI" labeling

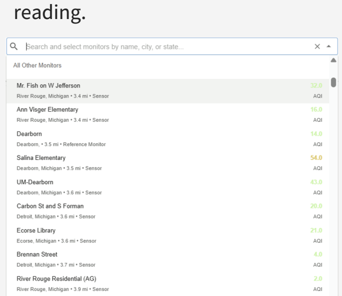
"AQI" appears repeatedly in line items and headers, creating **label noise** without adding meaning.

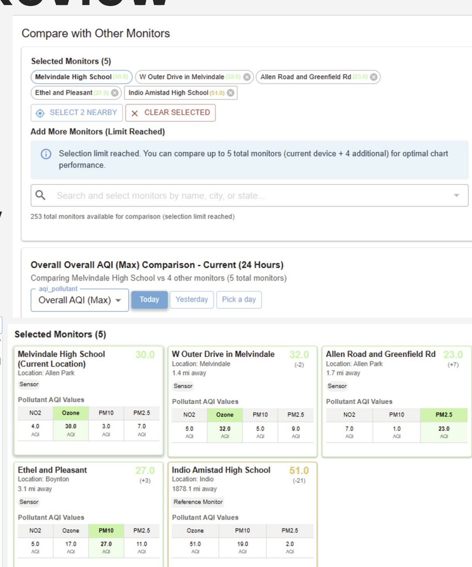
Hard-to-see AQI colors in search suggestions

The chips/suggestions rely on subtle hue shifts; in many cases, the **AQI value and state aren't legible at a glance**.

Hue-only series differentiation

Monitor series are distinguished **only by color**, which fails color-blind scenarios and hurts multi-series reading.





Compare Monitors Redesign

Monitor search options

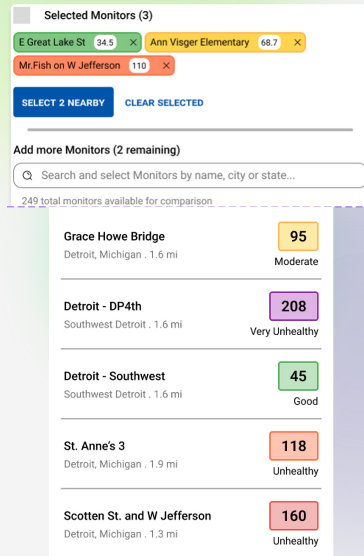
Redesigned the search results so AQI numbers have high contrast and remain readable against the input background.

Removed the repeated “AQI” text and added a concise secondary label that states the air-quality status (e.g., “Good,” “Moderate”).

Monitor tags & toolbar layout

Increased spacing and improved contrast so the tags do not feel cramped next to the “Select 2 Nearby” and “Clear Selected” buttons.

Updated the tag design to improve contrast so the aqi number is clearly visible



Expanded view for selected monitors

Removed redundant “AQI” labels and strengthened the contrast of the numbers so the key values stand out.

Added a secondary status label for each monitor so users can confirm meaning at a glance without relying only on color.

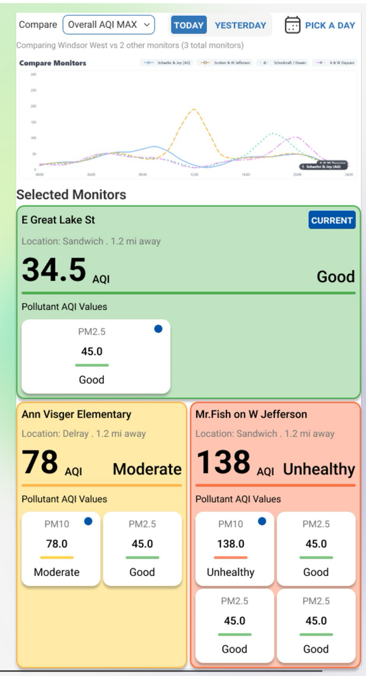
Added a “Current ” label for the primary monitor

Individual pollutants inside the expanded view

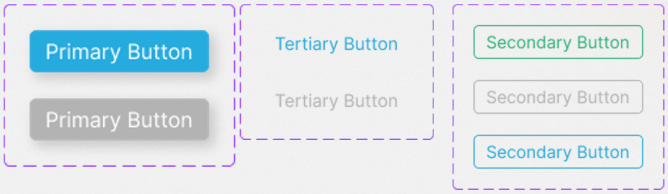
Replaced the tight table layout with roomier dialog-style cards to improve readability for values, units, and labels.

Added explicit status labels to support users with color-vision differences, so severity is clear even without color.

Introduced a circular badge to highlight the primary pollutant, which makes the main driver immediately visible. (We do this in the previous design too.)



Global UI Improvements

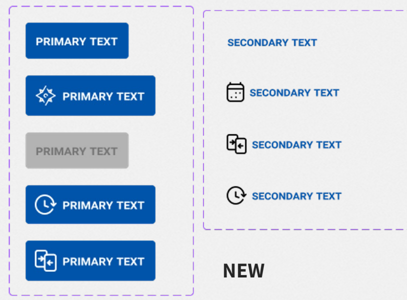


OLD

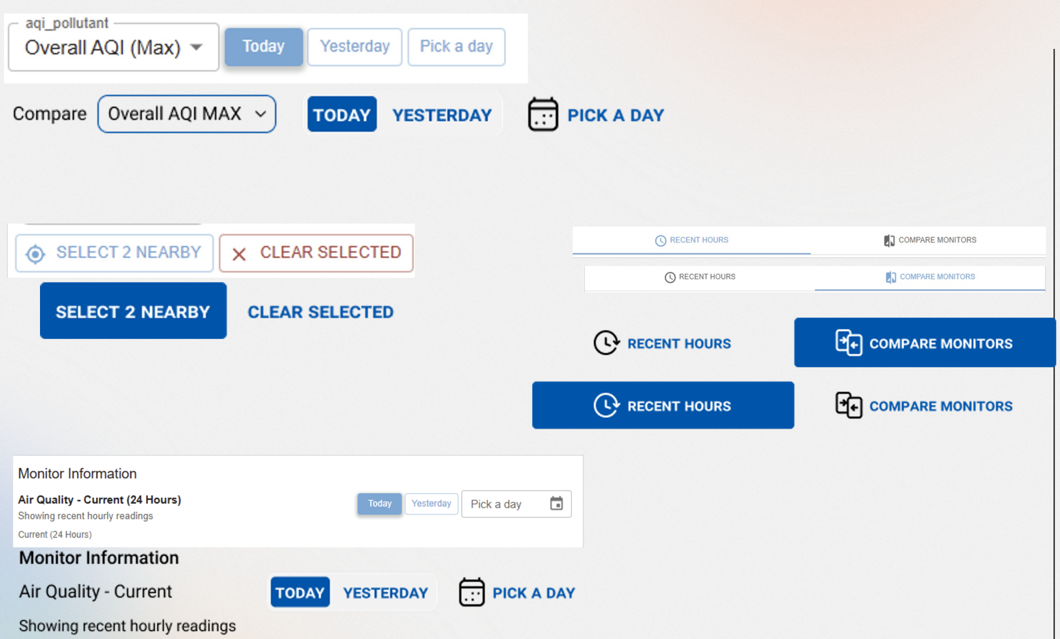
I updated the dashboard's primary blue to **# 0055AA** to **preserve the JustAir look while passing contrast** on light and white surfaces. I wanted to keep the brand hue, so I searched for a nearby blue that meets **WCAG AA**; this hit the balance between familiarity and legibility.

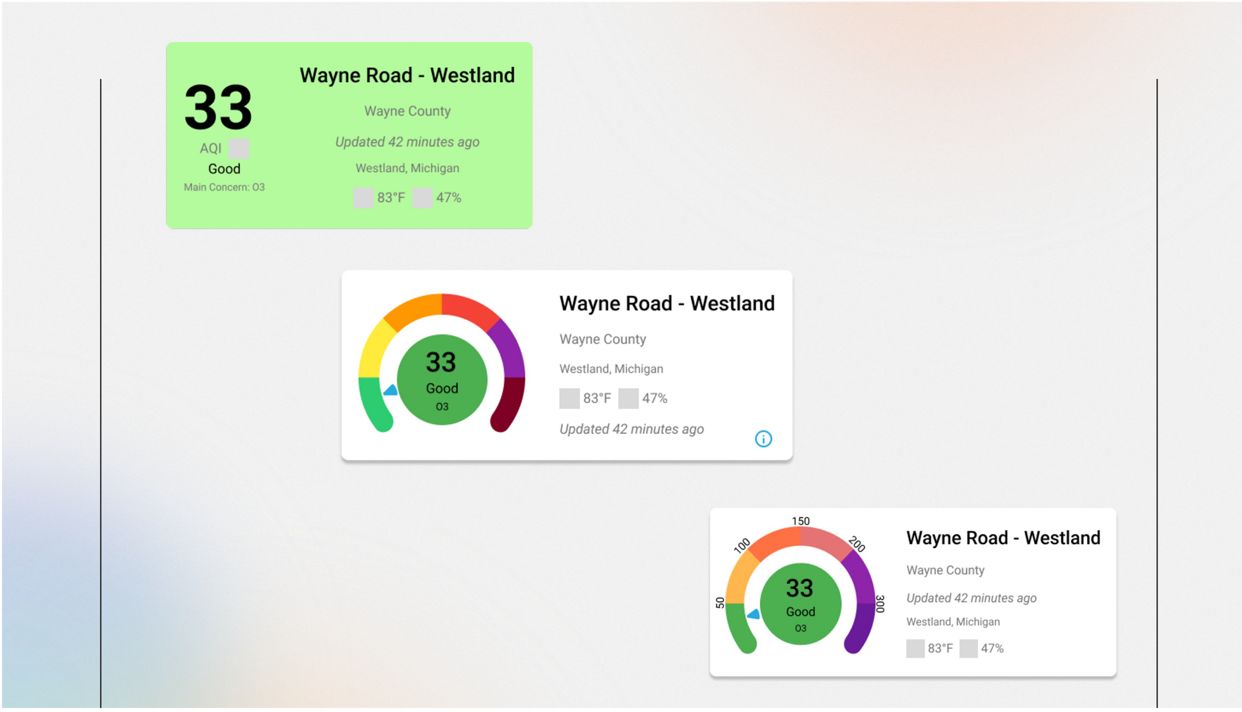
From there, I **rebuilt the button system**. Rather than patching individual buttons, I treated the color as a **design token**, which lets changes cascade across the UI without one-off fixes.

I **rolled the palette through high-touch components** (buttons, inputs, chips, links, focus rings) so text, icons, borders, and outlines are **readable at a glance**. I validated in real contexts—white cards, light-gray panels, and over imagery—to make sure contrast held up, including in edge cases like disabled buttons.



NEW





Wind Information Panel

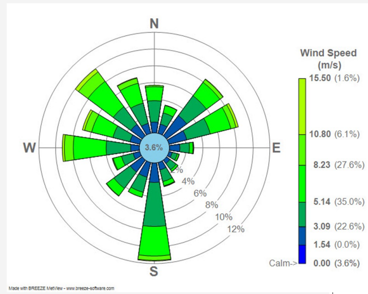
Balancing detail and simplicity

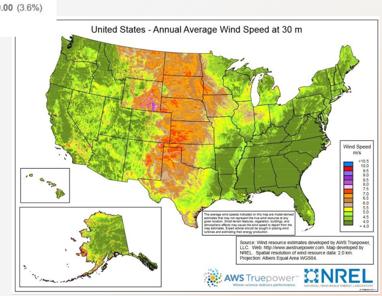
I tried to **find the sweet spot** between showing enough wind data and keeping it easy to glance at.

Windroses were a requirement, but a standard upfront windrose can be **intimidating or confusing** for many users.

The goal was to **offer both clarity for beginners** and **detail for experts**.

Designed a layout that **introduces users** to windroses without overwhelming them.





Core layout features

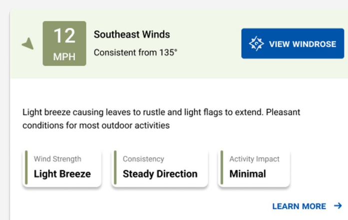
Wind **speed and direction** are displayed upfront for immediate clarity.

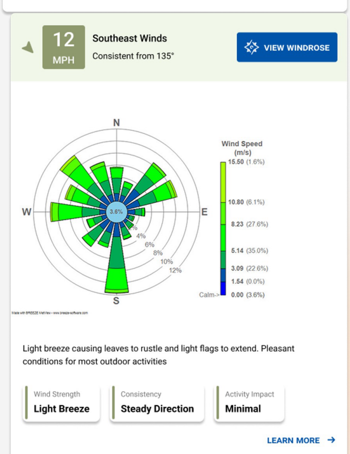
A **contextual message** accompanies the speed and direction, giving users a quick sense of current conditions.

Below that, there's **extra context about wind behavior**

A **button reveals the full windrose**, catering to **expert users** or those who want deeper insights.

This allows users to **see real wind behavior scenarios**, including more complex situations that may not be obvious at first glance.



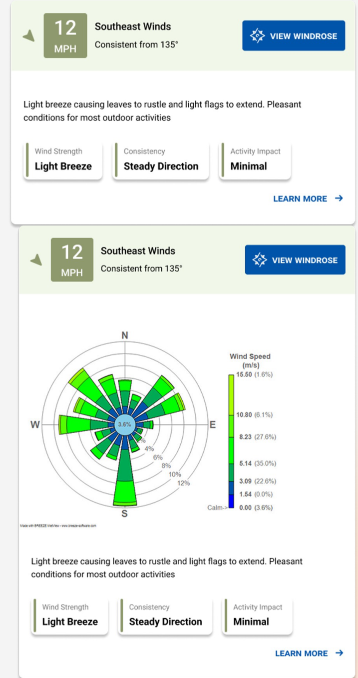


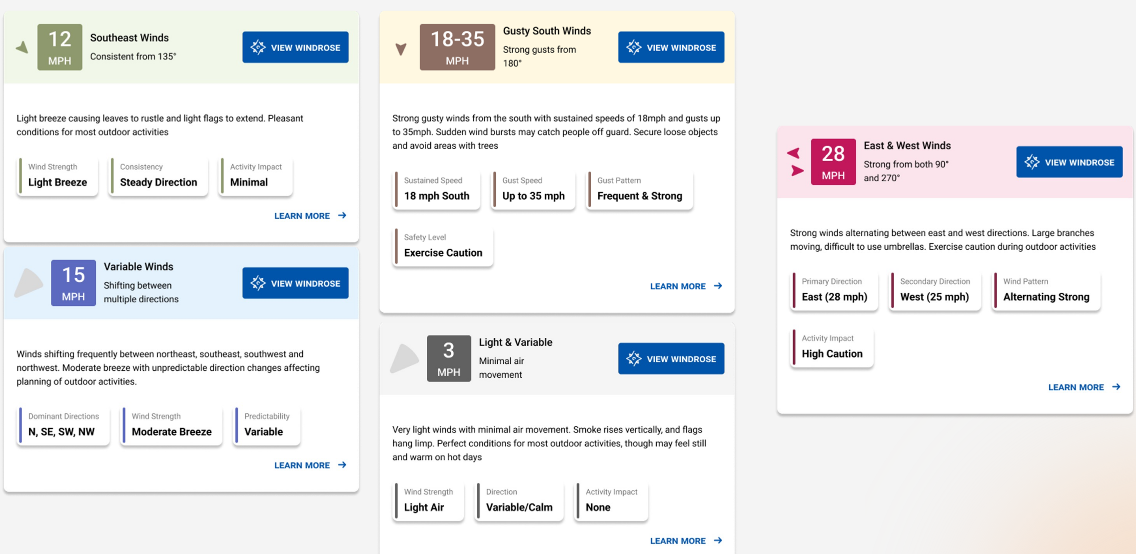
User learning and onboarding

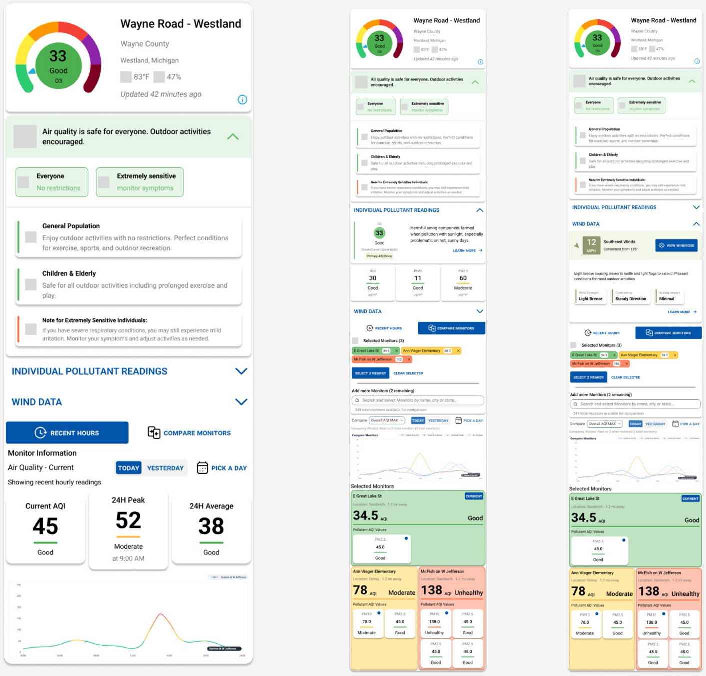
The layout helps users **gradually learn to interpret windroses** by connecting them with surrounding context.

Encourages **interactive learning**, as users can explore and compare patterns to understand real-world wind behavior.

Over time, users can **read windroses more confidently**, even without relying on the supplemental information.







Some other Recommendations

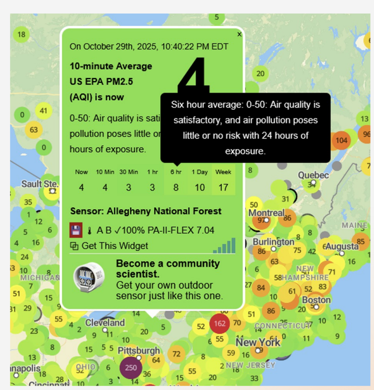
Can we turn the monitor overlay into a sidebar so the user can control the map and the monitor details at the same time ?

Can we change the cluster monitor icon. This doesn't pass the WCAG colors.

Can we add a hover state on the map? If I hover over a monitor icon for ~2 seconds, show a tooltip with that monitor's current AQI and their relevant info. This lets users browse more monitors before clicking.

Entire Figma File
https://www.figma.com/design/jzdkY7xZhutFhDYvGxadp6/Untitled?node-id=0-1&t=5ppf91L3OkvlqLa3-1

To view the redesigned dashboard and comment on it
https://www.figma.com/proto/jzdkY7xZhutFhDYvGxadp6/Untitled?page-id=0%3A1&node-id=37-658&viewport=-1278%2C2604%2C0.13&t=rW6knm7WrDEB4F7a-1&scaling=min-zoom&content-scaling=fixed&starting-point-node-id=37%3A658



IMG Taken from PurpleAir website

Thanks !

Appendix C

Data Science Capstone Student’s final report

An Air Quality Visualization Toolkit for Community Outreach

River Meckstroth, Adam Cooper, Abdelmoumen Medaghri Alaoui, Bhavya Jain

Abstract

This project is focused on the designing and creation of tools to transform large-scale air quality data into visualizable results that can be utilized by a wide audience. Using air quality data collected from sensors throughout Detroit, Michigan, our team developed a reusable visualization toolkit designed to make environmental data easier to explore and understand. The data consisted of many different pollutants such as PM1, PM2.5, and PM10, but the primary focus was finding a way to uniquely convey the air quality index levels. After integrating data from multiple locations and sources, we built interactive charts, maps, and analytical templates that reveal trends and particulars across time, locations, and pollutant types.

The purpose of the toolkit is to provide JustAir with potential means to visualize daily and long-term air-quality patterns to be presented to whatever audience they see fit. Our emphasis on reproducibility and robustness allows the tools to effectively support both public engagement and government reporting. In conclusion, our goal is to demonstrate that large-scale environmental monitoring data can be distilled into intuitive, adaptable visualizations that support informed decisions and policy efforts aimed at improving public health outcomes.

Background and Motivation

We live in a world where the amount of data produced is growing rapidly by the day. While much of this data sits stagnant, efforts to store and extract information for analysis are increasing across many industries. The insights gained from utilizing this data can improve levels of efficiency, support decision making, and promote positive societal change. This semester, the Computational Math, Science, and Engineering (CMSE) department at Michigan State University has partnered with JustAir, a company who is attempting to provide accessible insights and foster change with air quality data analysis. Our capstone team's goal was to help JustAir develop quality analysis and visualizations that can be used to assist with their mission to make air quality data more accessible.

Our capstone team consists of four members and was assisted by our professors, Dr. Dirk Colbry and Dr. Justin Gambrell. Further, we also worked closely with members of the JustAir team to ensure our project stayed within the scope of expectations. Their guidance was critical to our progress and development throughout the course of this project.

JustAir has a large network of air quality sensors set up across the country that collect information on many different pollutants such as PM1, PM 2.5, PM 10, SO2, and NO2. These sensors also record the temperature and humidity of the surrounding area. A subset of data collected from these sensors was processed and cleaned by JustAir to be used for this project. The data contained the location, active state, and name of each sensor as well as the time of data collection. An additional dataset containing the Air Quality Index (AQI), a metric calculated to indicate overall daily air quality, was also provided to assist our visualization building. All of the data that we used was sourced from Wayne County, Michigan.

Currently, JustAir manages a public dashboard that conveys information to the public using a geographical map and a variety of visualizations. Our goal throughout this semester was to assist with the development of these visualizations and develop tools to aid in report writing for community outreach. Using a variety of programming tools, primarily Python for its data handling and visualization tools, we built a visualization toolkit that stressed reproducibility and robustness to aid JustAir in its efforts to provide air quality information to a variety of audiences at varying levels of technicality. Our toolkit consists of a variety of types of information, ranging from easily digestible graphs to more complex, data rich figures.

Methodology

Our project followed a modular, reproducible workflow built around Python-based data processing and visualization tools with a primary goal of designing a flexible analytical framework and visualization toolkit that could be applied to any large-scale air-quality datasets that JustAir deemed necessary for analysis. Below is a discussion of our methods for our data pipeline, project structure, visualization methods, and software tools used.

The first portion of this project involved handling the ingestion and preprocessing of the provided data. As discussed above, large datasets containing air quality information from Detroit, MI were provided to us by JustAir. In order to handle this data, we wrote a data_loader.py python script to handle data preparation appropriately. Our set up instructions recommend placing all data files within the root of the repository in a folder entitled 'Air_Quality_Data'. From there, the loader identifies relevant CSV files within the folder and reads them into memory using the pandas 'read_csv' method. Pandas is the premier tool for handling data within python. As a result, we made use of its advanced handling and analysis features throughout the course of this project. The loader is hard coded to identify some of the specific files we were provided with, specifically, the 'TIZ.all.data.csv' and 'TIZ.AQIs.csv' data files. All other files in the folder are scanned for location-based information and sorted based on sensor name or city name depending on what information was available. Similarly labeled dataframes are concatenated and stored in a python dictionary, so they are easily available for visualization development. The data loader returns this dictionary upon call and is keyed by city or sensor name depending on information provided within the csv file. Given the small subset of data we worked with, there will be necessary expansion to the data loader if future work is needed, but it is equipped to handle the data we were given and any similarly labeled data.

Once data ingestion was handled, our team shifted focus to how best to build, deliver, and present any visualizations we designed. Ultimately, we built a modular structure that splits visualization delivery into two parts: implementation and presentation. All of the code behind the visualizations in the toolkit was contained in a python script. Then, a clean, easy-to-read Jupyter notebook with usage instructions was utilized to present the visualization and provide a description of its features and use cases. The scripts load all necessary python libraries or modules, while the notebooks simply import the functions built in our code to reduce clutter and increase the readability of the notebook. All Python scripts, one per visual, are found within the visuals folder of our project repository. The Jupyter notebooks are in the root directory of the repository.

Furthermore, to ensure their functionality and reproducibility, all visuals were designed to be filterable based on date, sensor, and pollutant when applicable. Within each Jupyter notebook, filtering options are provided for users so that they can choose whatever timeframe, location or metric they deem necessary. This filtering is done by changing values within notebook cells, in which options and descriptions are laid out.

The visualizations that were designed for this project utilized a variety of python-based software tools, Plotly, seaborn, and matplotlib libraries were the most prevalent. Plotly is primarily useful for building interactive visuals, as it enables hovering tooltips that contain useful information. Seaborn and matplotlib were used for more in-depth, detailed visuals. They typically produce static visualizations but allow for deeper levels of customization. All of the significant software we used is cited in the References section of this report.

Results

We are proud to present the JustAir Visualization Toolkit. The toolkit contains 9 unique, informative figures, and an additional predictive forecasting model, all designed to help JustAir convey air quality information to its partners and audience as effectively as possible. Each visualization was built to convey analytical information while remaining accessible to any non-technical audience as well. The following section walks through each piece of the toolkit and provides a description of its function, use case, and indented audience. More in-depth information can be found within the corresponding Jupyter notebook for all of the visualizations below.

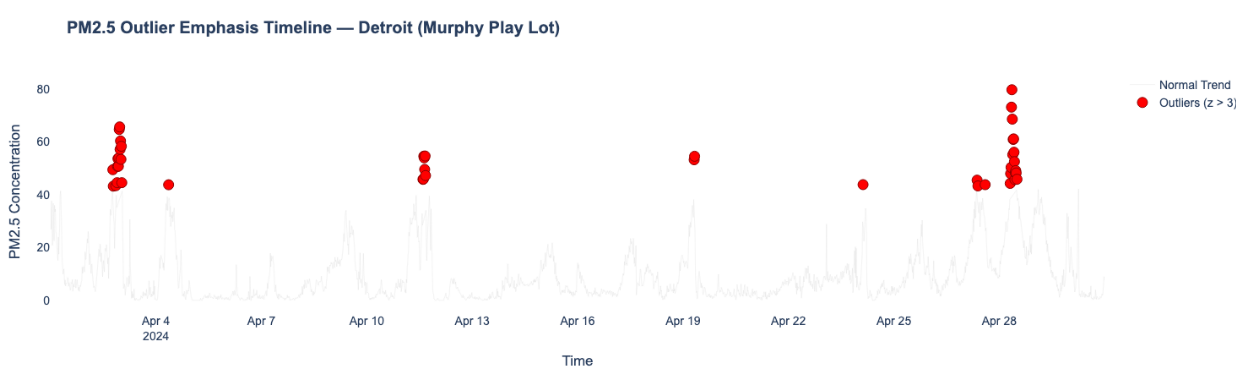


Figure 1: *Pollutant Outlier Timeline plotted across a month-long time series for a single sensor using standardized z –scores to identify outlier values for analysis*

The Outlier Emphasis Timeline visualization was designed with more technical audiences in mind. Outliers were detected using standardized z-scores and were emphasized along a time series plot that displayed pollutant information throughout the course of a month. **Figure 1** displays the outliers as red points and all other data as a light grey tracing.

This figure is interactive, and provides additional information, such as timestamp, pollutant, and percentage above mean to further assist users. The figure is filterable by year, month, pollutant, and sensor. The various customizable options will hopefully allow the appropriate audience to make useful insights.

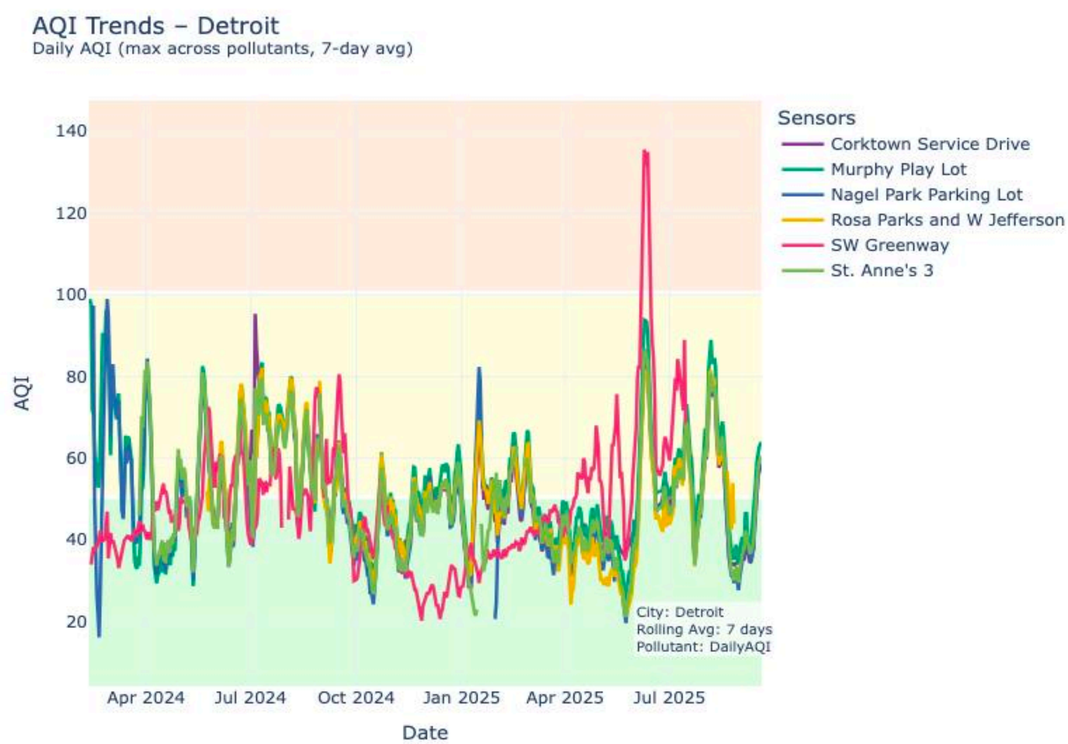


Figure 2: *Daily Air Quality Index (AQI) plotted across multiple Detroit sensors using a 7-day rolling average. Peaks in mid-July correspond to wildfire smoke events and post-holiday emissions*.

This visualization presents a high-level overview of daily Air Quality Index (AQI) trends across multiple air quality sensors. **Figure 2** displays daily AQI values smoothed with a 7-day rolling average. This allows viewers to quickly identify both long-term trends and short-term pollution spikes. For instance, the figure highlights notable AQI peaks in July 2025, a period associated with poor air quality due to the July 4th holiday.

The visual is dynamic, and is able to be filtered by pollutant, sensor, and rolling average window. Meaning a user can select which sensors they'd like, what pollutant and the rolling window (used for smoothing). This time-series figure was engineered with public engagement and policy communication in mind. With clear labeling, color-coding, and automatic smoothing, the hope is that residents and city officials can easily interpret when and where Detroit experienced poor air quality conditions.

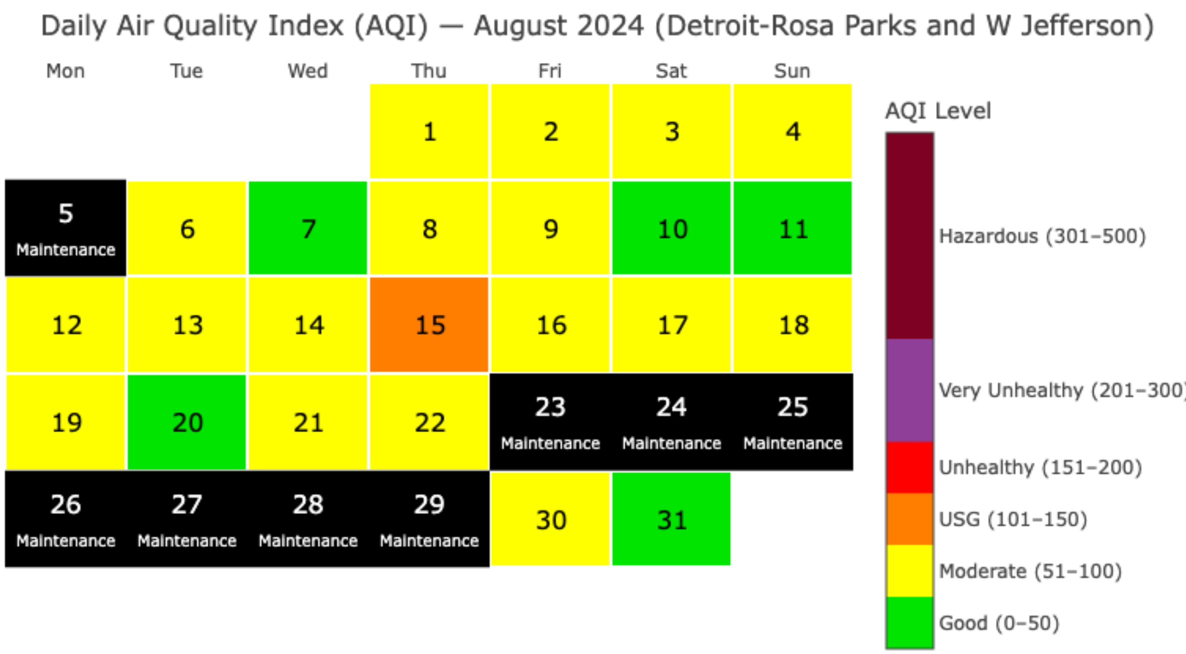


Figure 3: *Monthly AQI calendar visualization for a selected sensor, with color-coding for EPA AQI categories and black cells indicating maintenance days or missing data.*

The calendar visualization condenses a full month of AQI observations into an intuitive grid format where each cell represents a single day (**Figure 3**). Colors correspond to EPA AQI categories (e.g., *Good*, *Moderate*, *Unhealthy*), allowing for immediate interpretation. Days with missing or invalid data—commonly caused by sensor maintenance—are shaded in black.

This visualization is intentionally simple and communication-focused, making it suitable for community outreach, public dashboards, and summary reports. Its readability and compact structure allow users of all technical backgrounds to interpret daily air quality patterns quickly. Users can filter the calendar based on the sensor, year, and month they choose.

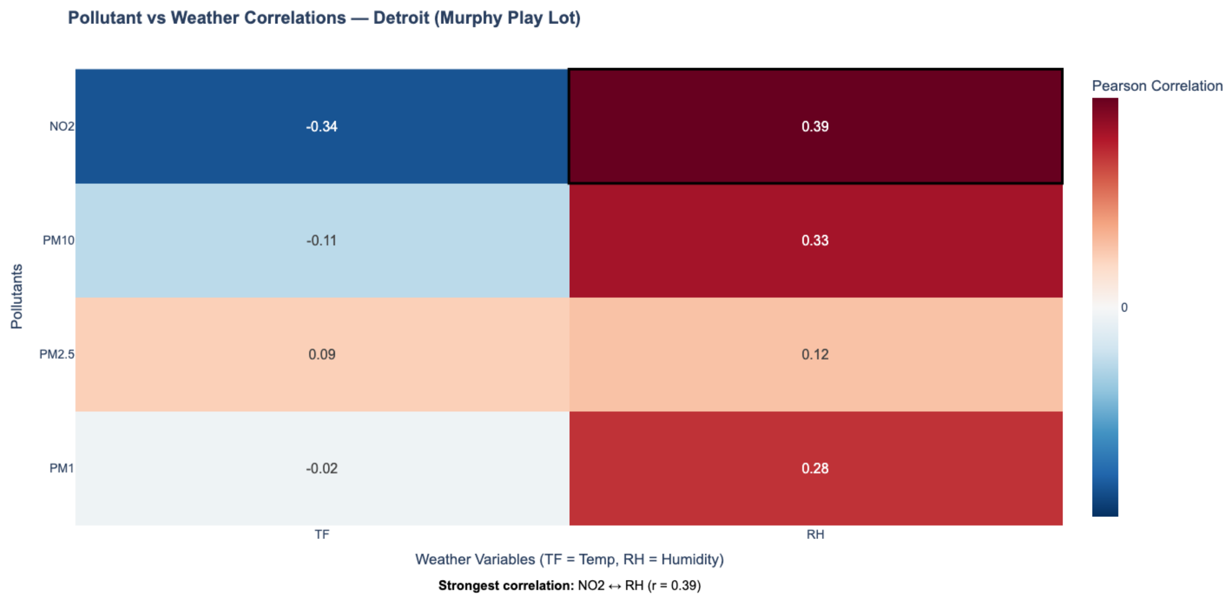


Figure 4: *Pearson correlation heatmap illustrating relationships between particulate pollutants and meteorological variables at a Detroit monitoring site.*

Designed for more technical audiences, we developed a visualization that quantifies the strength of relationships between pollutants and weather variables. **Figure 4 displays Pearson correlation coefficients for major pollutants (NO_2, PM10, PM2.5, PM1) and meteorological indicators (temperature, relative humidity) at a selected monitoring site.**

This figure helps environmental analysts identify meaningful patterns such as humidity-related particulate concentration changes or temperature-linked ozone fluctuations and is filterable by year, month, and sensor name. Because correlation matrices require more scientific background to interpret, this figure is meant primarily for researchers, environmental agencies, and data analysts.

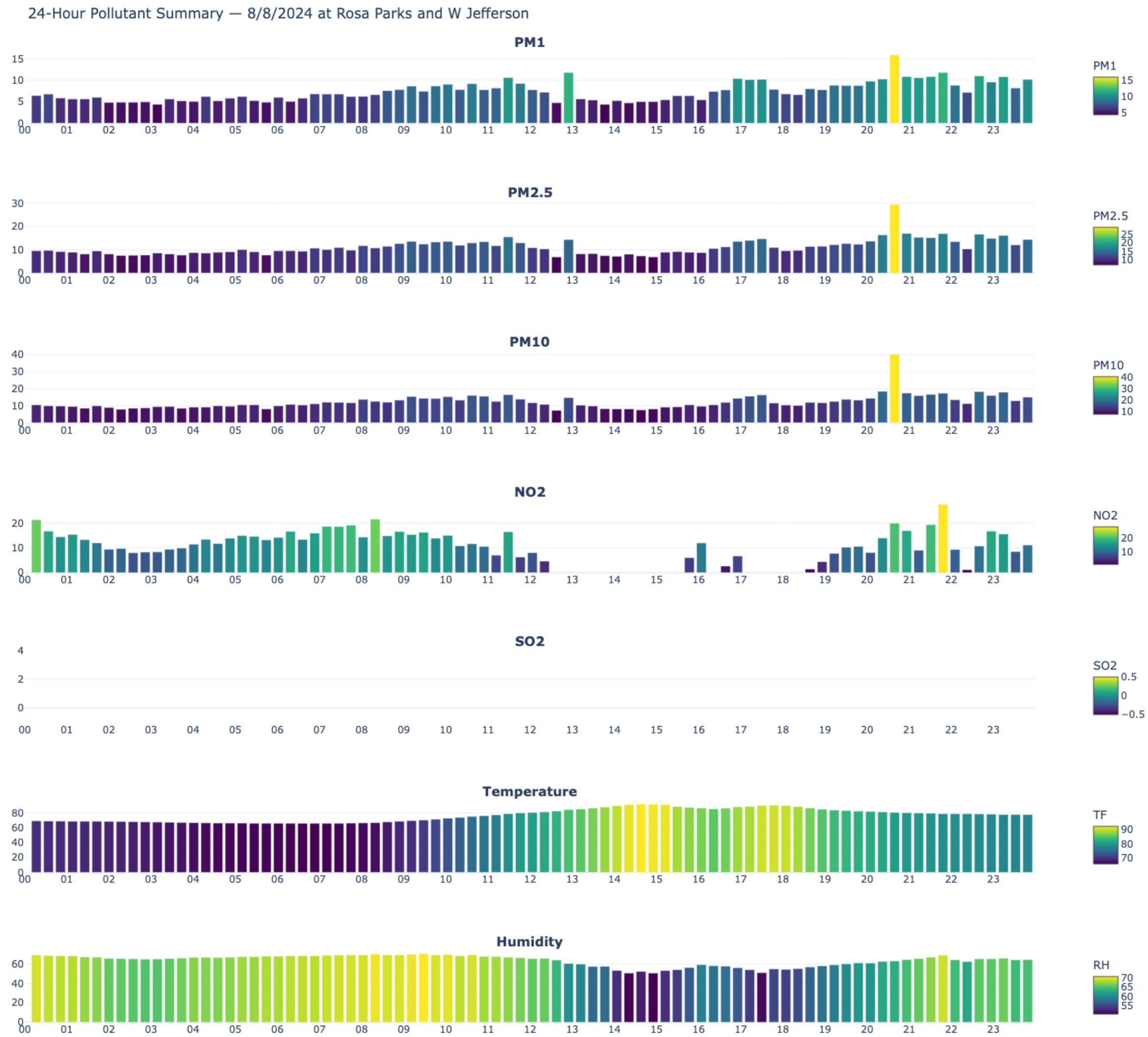


Figure 5: *Series of Bar Charts distribution of pollutants and other metrics over a 24-hour period*

The daily pollutant distribution summary visualization is designed to provide a look at the distribution of many air-quality metrics throughout a single day. As shown in **Figure 5**, this figure displays 24-hour bar charts for major pollutants (PM_1, $PM_{2.5}$, PM_{10}, NO_2, SO_2) alongside temperature and humidity profiles for the same period. The user can specify which metrics will be included, the sensor, as well as the specific year, month, and day they would like visualized.

This visualization is particularly useful for highlighting daily pollution patterns, such as morning and evening peaks associated with traffic; midday decreases driven by atmospheric mixing, or sudden pollutant spikes caused by localized events. It can also have some utility for internal team use, by allowing a quick look at when sensors are inactive

By presenting many pollutant channels on a synchronized timeline, this figure could be used by analysts, city officials, and community members. It serves as a valuable tool for diagnosing short-term environmental behaviors, communicating daily air-quality patterns, and supporting time-of-day–specific policy or outreach recommendations.

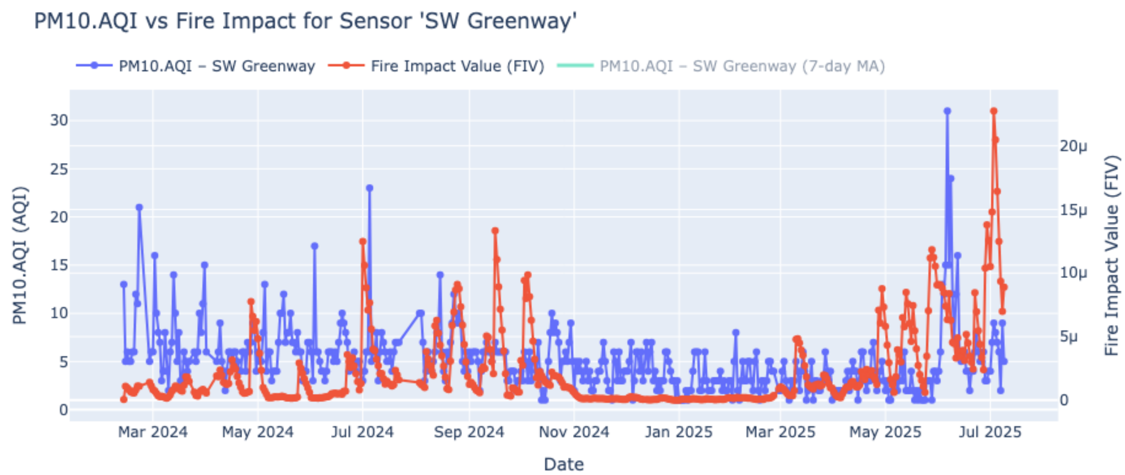


Figure 6: *Time series of PM10.AQI plotted along with a 'Fire Impact Value' calculated from MODIS satellite data*

The Fire Impact Value (FIV) and Air Quality Index (AQI) plot is designed to show how regional wildfire activity may relate to changes in local air quality. AQI values come from ground-based monitoring stations and capture pollutant levels such as PM2.5, PM10, or an overall Daily AQI at a specific sensor location. Plotted over time alongside FIV, they allow visual assessment of whether periods of poor air quality coincide with increased wildfire activity in the surrounding region.

The wildfire information used to construct FIV is sourced from NASA's Fire Information for Resource Management System (FIRMS), which processes data from the Moderate Resolution Imaging Spectroradiometer (MODIS) instruments on the Terra and Aqua satellites (see Appendix F for more information about the dataset). MODIS detects active fires as thermal anomalies and provides parameters such as brightness temperature, fire radiative power, location, and detection confidence. For each day and for a defined radius around the area of interest, FIV is computed by combining contributions from all detected fires. Each fire's contribution is weighted by its intensity (brightness temperature and/or fire radiative power), detection confidence, distance to the location (with closer fires weighted more strongly, typically using an inverse-distance-squared relationship), and recency within a multi-day window. Summing these weighted contributions produces a single FIV value per day that reflects the relative potential impact of wildfire smoke on that location.

The FIV–AQI plot can be used to investigate whether wildfire activity is a plausible driver of observed air-quality events. When peaks in FIV align with increases in AQI, it suggests that wildfire smoke may have contributed to degraded air quality. When AQI changes occur without corresponding FIV changes, local emissions or meteorological factors are more likely explanations. This combined view is useful for source attribution, contextualizing high-pollution days, comparing pollutant sensitivity (for example, PM2.5

versus PM10), and supporting communication of wildfire smoke impacts to public health and environmental stakeholders.

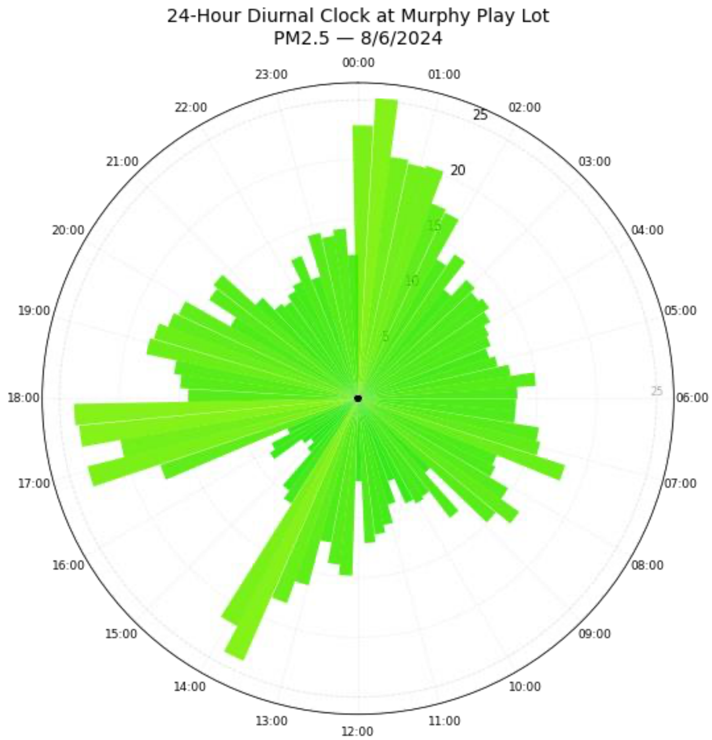


Figure 7: *24 Hour Clock Time series of pollutant distribution for a single sensor*

This diurnal plot was designed as an alternative to the traditional, linear time-series plot. Its unique formatting is designed for a general audience, as **Figure 7** attempts to convey simple information about the distribution of a single pollutant throughout a single day. The clock format spans 24 hours, unlike a traditional 12-hour clock, but provides a unique view of air quality trends over a very short span. The visual is designed to allow the user to pick a specific date (year, month, and day), as well as the sensor and metric they would like to view

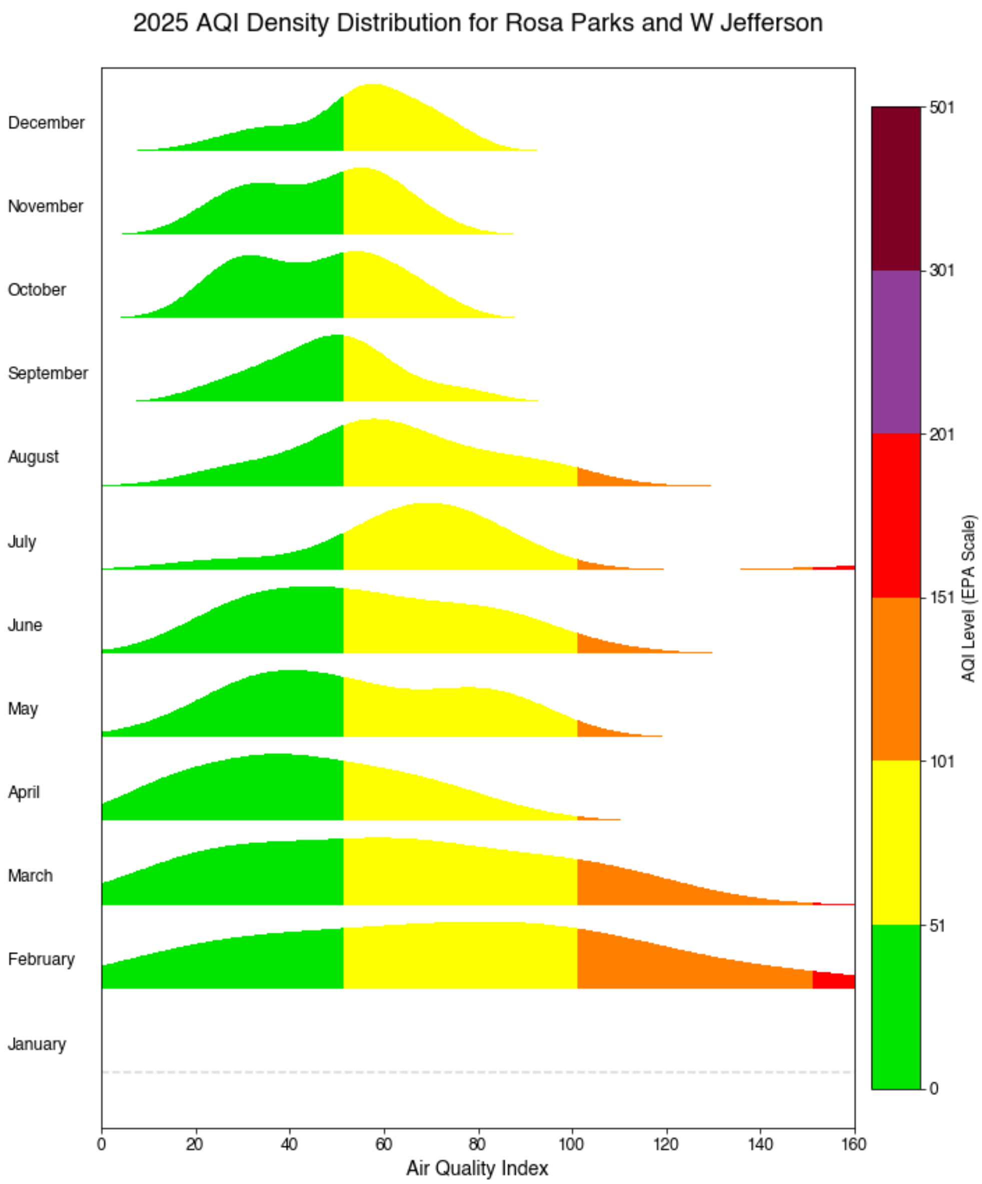
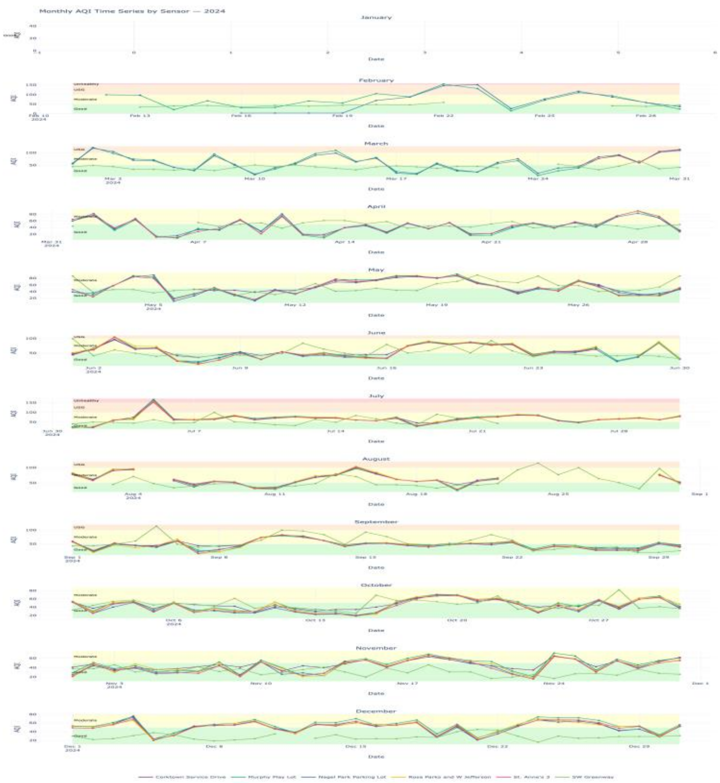


Figure 8: *AQI Density Distribution for a single sensor using KDE distribution over monthly subplots*

Designed for statistical purposes, the AQI Density Distribution visualization presented in **Figure 8** utilizes a kernel density estimation (KDE) to display the distribution of AQI measurements on a monthly basis. Unlike most of our visualizations, time is not plotted on the X axis. A histogram of AQI measurements is plotted each month to convey which AQI values appear the most frequently. The figure is segmented by color according to EPA standards for AQI safety level.

This figure could be helpful in identifying potential trends in which AQI is the worst or the most variable. Hopefully, this unique visual can provide some interesting insight to its appropriate audience. The visual is filterable by year and by sensor to give users the ability to track AQI distribution by time and location.



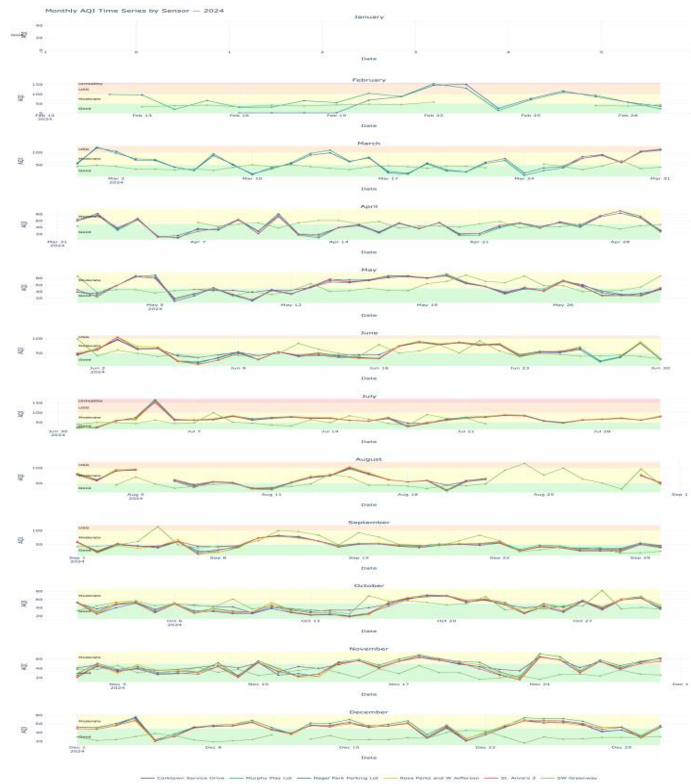


Figure 9: *AQI Time Series formatted in monthly subplots for many sensors throughout an entire year*

Finally, the Monthly AQI Time Series Visual is a very traditional air quality visualization. Split into subplots by month, **Figure 9** displays the AQI values of different sensors throughout the course of each month. The y axis is color coded and labeled according to the EPA color scale for AQI safety. The figure is interactive, hovering over a timestamp will reveal the date, and the AQI value of each sensor on said date.

This visual is a staple of air quality data and was designed to be readable and helpful for all audiences. It has potential for usage in report writing and displaying historical air quality data over time. Users can select which sensors they'd like to include to provide additional levels of analysis if they see fit.

Additional work

In addition to the visualization toolkit and analysis components described above, our team developed a short-term AQI forecasting module using machine learning. We evaluated two approaches (Prophet and XGBoost) and found that XGBoost captured short-term AQI patterns much more effectively. A sample 30-day forecast, and full model comparison are included in **Appendix E**. While this module is exploratory, it provides a foundation for future predictive analytics for JustAir.

Future Work

We are very proud of what we were able to accomplish throughout this project, but we recognize the potential for expansion of our work in the future. Below are some ideas for future initiatives related to the work we have done.

- Integrate additional external datasets (e.g., wind speed/direction, traffic intensity, regional smoke products, and other meteorological variables) to better explain when and why AQI patterns change.
- Develop a broader “health impact” toolkit that, alongside AQI, also reports UV index, heat index, and related exposure metrics, so users see a combined picture of environmental risks in their area.
- Refine and extend the Fire Impact Value (FIV) by using more satellite products and improved distance/lag modeling to quantify how strongly wildfire activity contributes to local AQI changes. Also explore more satellite datasets given on the NASA website. Fire and wind together should be a strong predictor (Appendix F).
- Build more rigorous statistical and machine-learning models linking FIV and pollutant-specific AQI (e.g., $PM_{2.5}$, PM_{10}) to move from visual association toward quantitative source attribution.
- Integrate an LLM API that can read the underlying data/plots and automatically generate plain-language summaries, highlight unusual events, and provide tailored health-related interpretations for residents, partners, and city officials.

Acknowledgements and Conclusion

First, we would like to thank our professors, Dr. Colbry and Dr. Gambrell, for all their help and guidance throughout this semester. Their insight and willingness to aid us in whatever way they could was greatly appreciated. Their impact on this project cannot be understated, and our team is very grateful to have had the opportunity to work with them and learn from them this semester.

Finally, we would like to thank JustAir for the opportunity to utilize the skills and techniques we have learned throughout our time as undergraduate students in the CMSE department. The opportunity for us to work with a community partner and gain insightful work experience in a professional environment cannot be understated as soon to be graduates. As we progressed through this semester's long partnership, we were increasingly excited to grow our understanding of the practical applications of the data science tools we have been learning about. So, thank you again JustAir for this incredible opportunity!

In summary, we hope that our semester-long collaboration with JustAir has expanded our technical and professional skills. We hope that the deliverables provided can assist with JustAir's mission to provide useful air quality information to those who need it. Our goal was to use the provided data to develop reliable and reproducible tools that can be used to assist JustAir as they see fit. We hope to see that the potential impact of this project will continue to grow as more and more recognize how much of a difference data science can make.

References

Figure 1. Authors (2025). Pollutant Outlier Emphasis *Timeline for a single sensor across one month.* Generated using project visualization scripts (Plotly, Python).

Figure 2. Authors (2025). *Daily Air Quality Index (AQI) across multiple Detroit sensors with 7-day rolling average.* Generated using project visualization scripts (Plotly, Python).

Figure 3. Authors (2025). *Monthly AQI calendar visualization for selected Detroit sensor.* Generated using project calendar visualization module (Plotly, Python).

Figure 4. Authors (2025). *Pearson correlation heatmap showing pollutant and meteorological variable correlations.* Generated using correlation visualization script (Seaborn/Matplotlib, Python).

Figure 5. Authors (2025). *Distribution of Pollutants over a 24 hour period.* Generated using daily pollutant distribution visualization script (Plotly, Python).

Figure 6. Authors (2025). *AQI vs Fire Impact Value calculated using NASA Firms*. Generated using project visualization script (Python, Seaborn, Pandas, Plotly).

Figure 7. Authors (2025). *24 Hour Clock Time series of pollutant distribution for a single sensor.* Generated using correlation visualization script (Matplotlib, Python).

Figure 8. Authors (2025). *Density Distribution of AQI over a year for a single sensor with monthly subplots.* Generated using project visualization scripts (Matplotlib, Python).

Figure 9. Authors (2025). *AQI Time Series over a year separated into monthly subplots for a single sensor.* Generated using project visualization scripts (Plotly, Python).

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software, 6*(60), 3021. https://doi.org/10.21105/joss.03021

Plotly Technologies Inc. (2015). *Plotly: Collaborative data science* [Computer software]. https://plotly.com

NASA Fire Information for Resource Management System (FIRMS). NASA Earth Science Data and Information System (ESDIS). Available at: https://firms.modaps.eosdis.nasa.gov/

Appendices

Appendix A: Deliverables

All project deliverables, including visualization scripts, data processing tools, generated figures, notebooks, and documentation, are stored in our GitLab repository:

GitLab Repository: https://gitlab.msu.edu/meckstr3/justair_data_analysis/-/tree/e180a4a74cd6c0b0963de2187588fd0bb6110f34/

Final Product Video Presentation:
https://mediaspace.msu.edu/media/2025_11_30%20CMSE495%20JustAir%20Final%20Presentation%20Video/1_90gir9rs

Appendix B: Data Bibliography/Dictionary

All of the data used in this project was provided by JustAir in several CSV files.

- 8 CSV files centered around 4 sensors in the Grand Rapids Area across all of 2024.
  - Rows timestamped by 17 minutes and 32 seconds (one row per 17min 24sec)
  - Standard Pollutant Variables - (PM 2.5, PM 10, NO2)
  - Meteorological Variables - (Humidity, Temp)
  - AQI Pollutant Variables - (PM2.5 AQI, PM10 AQI, NO2 AQI, Max AQI)

- One large CSV file containing individual pollutant information on 5 sensors in the Detroit area during 2024-2025.
  - Rows timestamped by 17 minutes and 24 seconds (one row per 17min 24sec)
  - Name, City, and Zip columns labeled recording (sensor) location
  - Standard Pollutant Variables - (PM1, PM 2.5, PM 10, NO2, SO2, O3, CO, NO)
  - Meteorological Variables - (TF, RH)

- One summarized CSV file containing cleaned AQI data for 5 sensors in the Detroit area during 2024-2025.
  - Rows timestamped and separated by day (one row per day)
  - Name, City, and Zip columns labeled recording (sensor) location
  - Daily Pollutant Variables - (PM 2.5, PM 10, NO2, SO2, O3, CO, NO)
  - Valid Hours Pollutant Variables - (PM 2.5, PM 10, NO2, SO2, O3, CO, NO)
  - Daily AQI column – calculated AQI for given day
  - Pollutant column – contributing pollutant for DailyAQI
  - Category column – EPA category for AQI safety

Appendix C: Installation / Reproducibility instructions

Prerequisites

To run the JustAir Visualization Toolkit, users must install the following:

Required Software

Git- Used to clone the project repository. Available at: https://git-scm.com/downloads

Miniconda or Anaconda - Used to manage the Python environment. Available at: https://docs.conda.io/en/latest/miniconda.html

System Requirements

- Windows 10/11, macOS 10.15+, or Linux
- Minimum 4 GB RAM (8 GB recommended)
- At least 2 GB free disk space
- Internet connectivity (for downloading packages)

Quick Setup Instructions

These steps describe the essential path to installation using Conda (recommended for all platforms).

1. **Clone the Repository**
   a. Open a terminal, PowerShell, or command prompt.
   b. Run:
      i. git clone <REPOSITORY_URL>
      ii. Navigate into the project directory.
2. **Create the Conda Environment**
   a. Create a new environment:
      i. conda create -n justair_env python=3.11 pip -y
   b. Activate the environment:
      i. conda activate justair_env
3. **Install Project Dependencies**
   a. Install required Python packages from requirements.txt using:
      i. pip install -r requirements.txt
4. **Set Up the Data Directory**
   a. Create a folder named **Air_Quality_Data** in the **parent directory** of the project.
   b. Place all provided CSV data files into that folder, including:
      i. TIZ.all.data.csv
      ii. TIZ.AQIs.csv
      iii. Any additional sensor files

5. **Run the Toolkit**
    a. Open any notebook in the notebooks/ directory.
    b. Run the cells to generate visualizations and reproduce project results.

Data Directory Requirements

The visualization toolkit expects a specific directory layout:

- The folder **Air_Quality_Data** must be located within the root of the project directory.
- The folder should contain all air quality CSV files provided by JustAir.
- Required columns include:
    - Time
    - Name
    - Pollutants (e.g., PM2.5, PM10, NO2)
    - AQI fields (PM2.5.AQI, Daily_AQI, etc.)

Troubleshooting Summary

- If packages fail to import, verify the environment with:
    - conda list
- Ensure that the data directory is in the correct location.
- Re-check CSV file formatting and column names if errors occur.
- Consult the full **INSTALL.md** document for detailed guidance.

Appendix D: AQI Forecasting

Overview

As part of our extended work for this project, we developed an exploratory machine-learning forecasting module to predict short-term Daily AQI values. The goal of this module is to demonstrate how predictive modeling can complement JustAir's visualization tools by estimating near-future air quality based on historical patterns.

To evaluate forecasting performance, we tested two modeling approaches commonly used in time-series analysis:

• **Prophet** – a general-purpose forecasting model developed by Meta

• **XGBoost** – a gradient boosting machine-learning model well-suited for structured tabular data

Both models were trained on the same input data, which included past AQI values, rolling volatility, calendar features (day of week, month, etc.), and basic weather variables such as temperature and humidity.

Model comparison

The figure below compares each model's predictions with the actual observed AQI values on the test set.

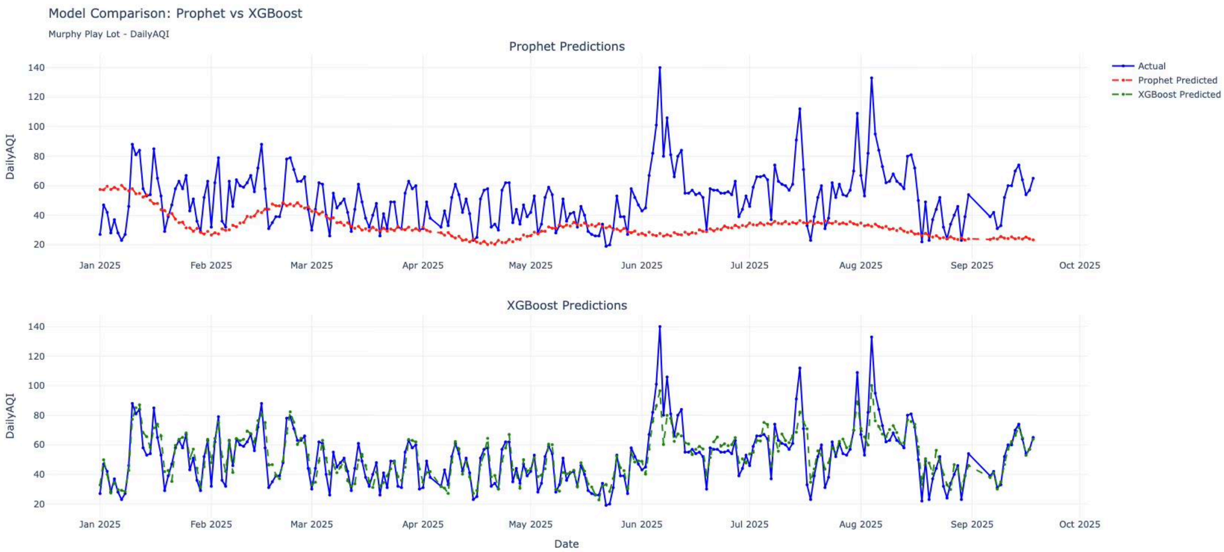


Figure D1: Prophet vs XGBoost Prediction Performance

Prophet (top) struggled to capture AQI variability and produced overly smoothed predictions. XGBoost (bottom) closely followed real AQI fluctuations and demonstrated strong short-term predictive accuracy.

Prediction Accuracy

To quantify performance, we compared each model's predicted AQI values against the actual test values.

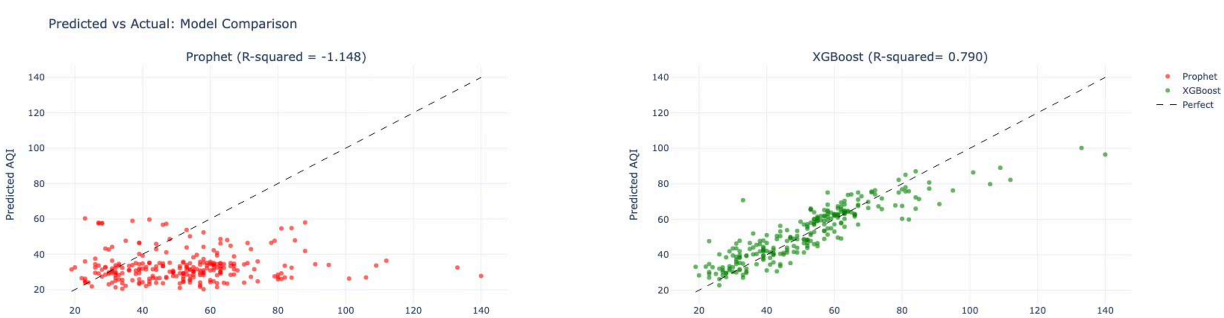


Figure D2: Predicted vs Actual Scatter Plots

Prophet achieved a negative R^2, indicating poor fit and systematic under-prediction. XGBoost achieved an R^2 of approximately 0.79, demonstrating strong alignment with actual AQI values.

30-Day Forecast Example

After validating model performance, we generated a sample 30-day forecast using the XGBoost model.

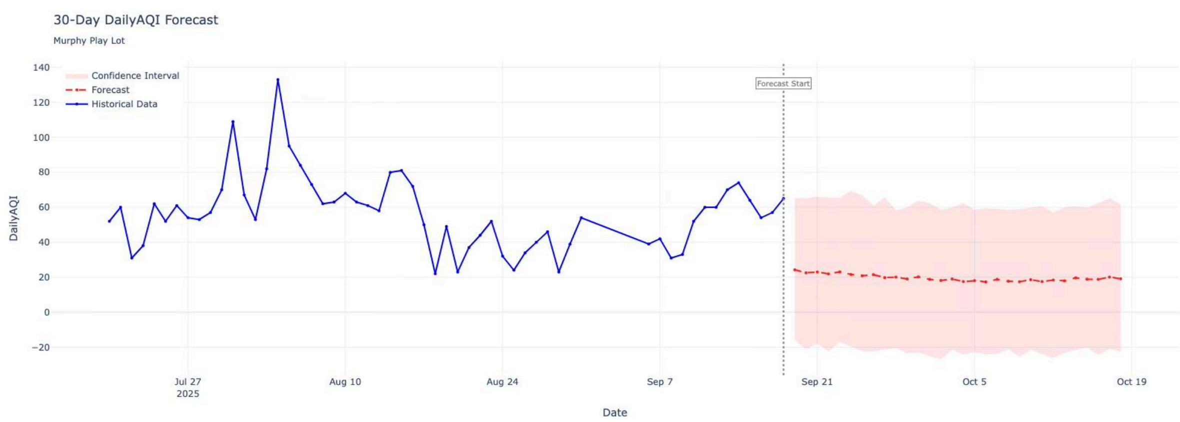


Figure D3: 30-Day Forecast Example

The model predicts a stable AQI trend over the next month, with values generally in the "Good" range. The shaded area represents uncertainty, which increases naturally the farther into the future the forecast extends.

Notes and limitations

- This module is intended as a proof of concept, not a production forecasting tool.
- Forecasts rely on past AQI values and historical weather, not weather forecasts.
- Long-range predictions naturally become smoother due to recursive forecasting effects.
- Sudden pollution events (e.g., wildfires, industrial releases) cannot be predicted without additional data sources.

Conclusion

This forecasting module illustrates how predictive analytics can enhance JustAir's platform by providing short-term air-quality outlooks. While preliminary, the results demonstrate that machine-learning methods such as XGBoost can effectively capture AQI patterns and form a strong baseline for future forecasting work.

Appendix E: NASA FIRMS

https://firms.modaps.eosdis.nasa.gov/

NASA FIRMS integrates wildfire detections from multiple Earth-observing satellites, each offering different spatial resolutions, revisit frequencies, and detection capabilities. While MODIS (1 km resolution) remains one of the most widely used long-term datasets, several higher-resolution sensors are also incorporated into FIRMS and can be used to enhance fire–air-quality analyses.

The Visible Infrared Imaging Radiometer Suite (VIIRS) instruments aboard multiple platforms provide substantially finer spatial resolution than MODIS. The VIIRS 375 m active fire product is available from three satellites: Suomi National Polar-orbiting Partnership (S-NPP), NOAA-20, and NOAA-21. These sensors detect smaller and lower-intensity fires more effectively than MODIS due to their improved sensitivity and spatial detail, making them particularly valuable for regional smoke-impact studies. Because VIIRS satellites maintain polar orbits with multiple daily overpasses, they also offer improved temporal coverage relative to a single-sensor system.

In addition to MODIS and VIIRS, FIRMS also supports wildfire detections from the Landsat program. Landsat multispectral instruments provide 30 m resolution, enabling extremely fine-scale characterization of burn scars and active thermal anomalies. Although Landsat's temporal revisit interval is longer (typically 8–16 days depending on satellite and latitude), its high spatial resolution makes it useful for validation, land-cover impact assessments, and detailed mapping of fire progression.

Together, these satellite systems—MODIS at 1 km, VIIRS at 375 m across S-NPP, NOAA-20, and NOAA-21, and Landsat at 30 m—form a comprehensive multi-resolution fire-monitoring network. FIRMS harmonizes these products into a unified interface and API, allowing rapid access to global fire detections across multiple resolutions and sensor technologies. This flexibility makes FIRMS well suited for constructing fire influence indicators such as the Fire Impact Value (FIV), integrating observations from diverse satellite platforms to support environmental, atmospheric, and public-health analyses.

Regional Economic Innovation (REI) seeks to identify and develop new economic development tools, models, policies, and practices to support innovative economic development, high-growth enterprises, and job creation in distressed regions across the state. REI is establishing a new economic development ecosystem to cope with the ever-changing global and regional dynamics. Through this ecosystem, REI engages innovative and creative minds which results in new economic development practices.

REI was established in 2011 with support from the U.S. Department of Commerce, Economic Development Administration, and in collaboration with the following Michigan State University offices:

Office of the Provost
Office of the Vice President for Research and Innovation
University Outreach and Engagement
MSU Extension
College of Communication Arts and Sciences

MICHIGAN STATE UNIVERSITY
REGIONAL ECONOMIC INNOVATION

2025 SLFG Project

MICHIGAN STATE UNIVERSITY | University Outreach and Engagement