ECOVISION FINAL GROUP REPORT

Mapping London's Air Quality and Vehicle Emissions through Interactive Data Visualisation

http://10.129.111.18:3000/casa0017-web-assessment-EcoVision/

Air pollution remains one of London's most persistent environmental and public-health challenges. According to the London Atmospheric Emissions Inventory (GLA, 2023), road transport contributes nearly half of the city's nitrogen oxide emissions. Yet, while raw datasets are publicly available through portals such as DEFRA, TfL, and the London Datastore, these resources are often complex, fragmented, and inaccessible to the public.

The EcoVision Team project was conceived to bridge this gap. Our goal was to transform complex emissions data into an engaging and intuitive interactive map-based website that empowers Londoners to visualise, explore, and understand how air quality fluctuates across boroughs throughout a 24-hour period. By enabling users to interact with temporal and spatial data, we sought to support informed decision-making around mobility choices and environmental behaviour. The name 'EcoVision' symbolises our belief that data should not just inform but inspire action. We set out to tackle one of London's most persistent challenges: air pollution caused by vehicle emissions. Our interactive web platform visualises real-time air-quality data, revealing how CO_2, NO_2, and PM2.5 emissions fluctuate across London's 33 boroughs throughout the day.

The EcoVision project was structured using agile principles over a four-week sprint, where each week represented a distinct phase of the project lifecycle, from ideation and design to data integration and deployment. The team adopted an iterative approach to ensure continuous feedback, alignment, and improvement. Weekly stand-ups and GitHub commits ensured transparency and accountability, while collaborative tools like google drive and Microsoft Teams supported design, documentation, and coordination.

How data improves the environment

Objectives and Key Initiatives

- **To visualise CO_2, NO_2, and PM2.5 emissions across London's boroughs.**
- **To integrate and compare traffic and cycling data to promote sustainable transport decisions.**
- **To leverage Globe.GL for an interactive globe-based 3D view of London's air quality.**
- **To combine storytelling and data science to enhance citizen engagement.**
- **To deliver a fully functional, responsive, and data-driven website within four weeks.**

While open datasets exist from the Greater London Authority, DEFRA, and Transport for London, they are often fragmented or difficult for citizens to interpret. EcoVision emerged to bridge this gap by translating complex environmental data into a visual, interactive story.

Our project aligns with ongoing research in environmental data visualisation and urban informatics. Existing digital projects such as the 'London Air' monitoring dashboard (King's College London, 2022), The Guardian's 'Toxic London' map, and DEFRA's Air Quality Data API have demonstrated how open data can be leveraged for public awareness. Academic studies (Batty, 2018; Cox et al., 2022) highlight that visual narratives enhance comprehension and engagement when data is presented interactively. EcoVision builds upon this foundation by integrating 2D and 3D mapping through Globe.GL, an open-source WebGL framework capable of rendering immersive geospatial experiences.

Progress and Achievements

Through a **collaborative four-week design sprint**, the **EcoVision team** combined environmental research, design thinking, and technical development to create a **functional and educational web platform** that transforms complex air-quality data into **clear, actionable insight**. The project followed an **agile approach**, with **weekly team meetings and sprint reviews** that ensured steady progress, open communication, and collective problem-solving. Each session served as both a checkpoint and a creative workshop, aligning goals, refining visuals, and addressing technical challenges together.

Matilda developed initial concept sketches that guided our design direction, while **Yidan** shared inspiring references and examples of interactive websites that helped shape our visual identity. **Madina** ensured the project stayed on track, facilitating discussions and keeping our focus aligned with the main objectives and sprint milestones. Meanwhile, **Zihang** and **Qingshan** offered crucial technical insight, advising on feasibility and helping us balance ambition with realistic development goals. Our **weekly meetings and extended brainstorming sessions** fostered a strong sense of teamwork and mutual respect, making each discussion both **productive and creatively rewarding**.

The team maintained a strong **sense of cohesion and shared purpose**, with every member contributing their expertise, ranging from data analysis and UX design to backend development, with remarkable commitment to delivering results. The UX design adhered to **Nielsen's usability**

heuristics (Nielsen, 1994) and **Lynch & Horton's (2017)** principles of clear navigation, consistency, and visual hierarchy.

Week	Key Activities	Deliverables
Week 1	Brainstorming, defining project scope, dataset discovery, and storyboard sketches	Storyboard and initial data sources
Week 2	Data cleaning, backend setup initiation, and initial frontend wireframes	Cleaned datasets and basic UI mockups
Week 3	Development of interactivity – time slider, borough data linkages, testing API endpoints	Functional interactive map and slider
Week 4	Final integration, usability testing, presentation and documentation	Final website and project report

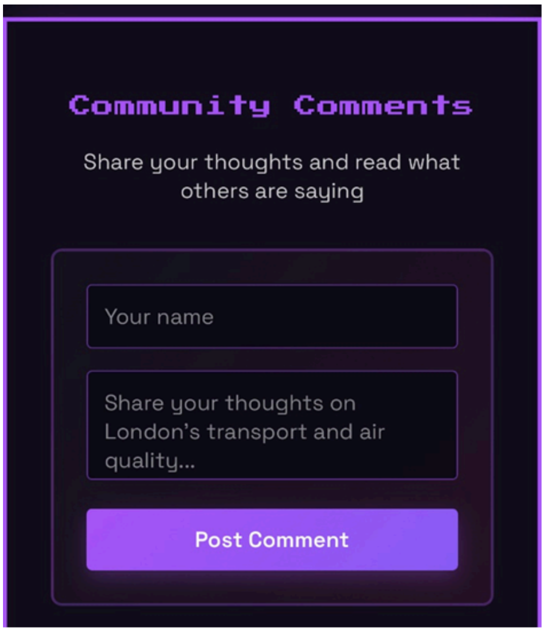
The team practically evaluated various frameworks, such as using Next.js (React, App Router) to prototype and test routing ergonomics and data fetching patterns. However, given the project's scale, these additional overheads seemed unnecessary. Ultimately, the team chose Vue 3 and Vite for faster iteration speeds, smaller package sizes, and composable logic. The project boasts a rich technology stack: Vue 3 for responsive components; Vite for improving development server speed and separating the development environment from production; Vue Router for lazy-loading routes; and styling combined utility classes and SCSS markup to ensure consistent spacing, fonts, and themes across more than 25 components. At the visualization layer, D3.js handles the scale bar and interactive charts. We reserve a full-width map "slot" so the teammate-owned map integrates cleanly while sharing our legends, filters, and layout rhythm.

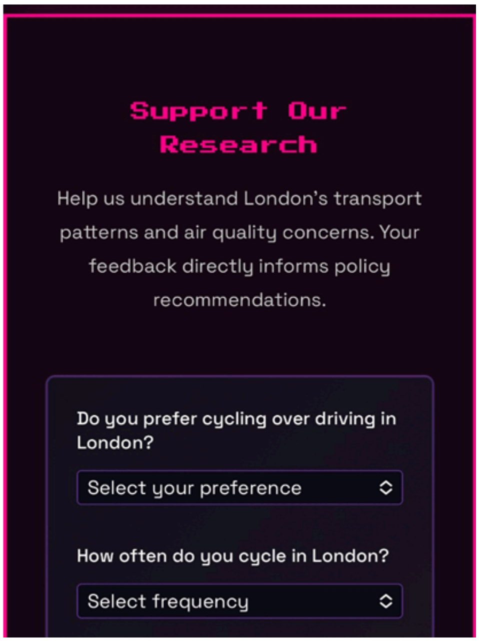
Implementation favoured performance and clarity. Route-level code splitting and asset preloading trimmed the first bundle to ~485 KB (zip). First Contentful Paint arrived near **1.2s** and Time to Interactive around **2.8s** on our test profile, with Lighthouse Performance **92/100**.

The time slider is debounced; heavy panels load on demand; animations use CSS where possible to stay smooth on mid-range laptops and phones. For charts and animated backgrounds, "requestAnimationFrame" governs timing so interactions remain at or near 60 fps. DevX also mattered: Composition API reduced duplication across components; ESLint + Prettier kept the codebase uniform; Vue DevTools shortened debugging loops. These choices let us ship small, frequent increments without destabilizing the UI.

Storyboarding and User Journey (UX) - Yidan Gao

Yidan led the user experience and interface work, translating the team's storyboard into a smooth and readable user experience, transitioning from a welcoming homepage to the "Explore London" interaction, and ultimately to the solution and team pages ,and she built the site code end to end. She used Miro and Canva to sketch early storyboards of the journey: a calm home page with an “Explore the Interactive Map” button, then a map where users slide through time, click boroughs, and see daily pollution patterns. She turned these plans into working pages with Vue 3, Vite, and Tailwind CSS. She set up Vue Router for clean paths between Home, Map, and Solutions. She built the time slider, click actions, tooltips, and clear legends. She added a Globe.GL background to give scale and context. Colors and contrast follow WCAG 2.1 so the site is easy to read. The layout is simple and fast, with small, reusable components. Finally, Yidan documented the Technical Stack & Tools for maintainability and scale: build/runtime (Vue 3, Vite, Node, npm), state/routing (Pinia, Vue Router), viz/3D (D3.js; reserved 3D context), styling (CSS/SCSS), data and real-time (Axios; MQTT option for sensor feeds), quality (ESLint, Prettier, Vue DevTools, Lighthouse), CI/CD (GitHub Actions → GitHub Pages). The component architecture and store design let us add pages and metrics quickly, while SCSS variables propagate theme changes instantly. Looking ahead, Yidan recommend swapping the file-based feeds for a thin API, keeping the same UI contracts; the front end is already wired for that evolution. Her goal was not only to show data, but to make people feel it: warm gradients, plain language, and smooth motion help visitors learn and act.





Timely user feedback

Data Collection and Handling - Matilda Nelson

Matilda identified and curated open datasets relevant to the project. From these sources, she extracted key datasets focusing on air traffic pollution emissions, traffic counts per borough per hour, and transport mode data distinguishing motor vehicles from bicycles. She organised and documented each dataset and played a central part in conceptual sketching and group ideation, often producing diagrams and storyboards during discussions to ensure the team remained aligned on how the data should shape the website's structure and storytelling. She advocated for a data-led design process, making sure that every visual and narrative decision was rooted in the project's environmental focus rather than aesthetic preference alone. She reimagined the solutions hub, connecting visualised emissions and traffic data to actionable insights for individuals and communities. This approach transformed the embedded solutions from a static reference section into a meaningful, evidence-based extension of the EcoVision story.

Dataset	Source	Description
Road Transport Emissions	London Datastore / DEFRA	Borough-level CO_2 and NO_2 emissions
Traffic Volume by Hour	TfL Open Data API	Average hourly traffic counts per borough
DEFRA's Air Quality Archive	UK Department for Environment, Food & Rural Affairs	Hourly and annual measurements of key pollutants such as **NO_2, $PM_{2.5}$, PM_{10}, O_3, CO, and SO_2**, as well as air quality indices for specific locations.
Bicycle Use per Borough	TFL Cycling Data	Daily cycling rates for comparison

Backend Development and Data Collection & Processing - Qinshang Luo

Qingshan managed the backend data pipeline and database deployment. He established a MySQL database on the server using OpenSSH to transfer the structured CSV files. Qingshan was responsible for developing the backend architecture that powered EcoVision's data-driven functionality. His main goal was to create a seamless connection between the datasets collected by Matilda and the interactive front-end designed by Zihang. He built two main APIs:

-The first API delivered borough-level data on CO_2, NO_2, and PM2.5 emissions, along with traffic flow per hour, enabling the map and time-slider to display real-time updates.

-The second API provided information on the busiest streets, comparing car and bicycle travel times while showing emissions from vehicles to promote sustainable travel choices.

For data sourcing, Qingshan retrieved detailed pollution records from the UK National Atmospheric Emissions Inventory (NAEI), focusing on emissions linked to road transport. Using a custom Python web crawler, he extracted and formatted the data into CSV files. He then incorporated Google Maps travel data and emission rate calculations to estimate pollutants produced per route and per time. The backend environment used Node.js and Express.js, supported by. env, package. Json, and server.js files for configuration, dependencies, and routing.

After testing endpoints on a local network, he verified that both APIs were delivering accurate and responsive results to the front end. His backend system became the foundation that enabled real-time data visualization across the site.

```sql
CREATE TABLE borough_hourly (
  `borough` VARCHAR(100) NOT NULL,
  `co2_kg_per_h` INT UNSIGNED NULL,
  `co_g_per_h`   INT UNSIGNED NULL,
  `pm25_g_per_h` INT UNSIGNED NULL,
  `vehicles_per_hour` INT UNSIGNED NULL,
  PRIMARY KEY (`borough`)
);
```

```sql
INSERT INTO borough_hourly
(`borough`,`co2_kg_per_h`,`co_g_per_h`,`pm25_g_per_h`,`vehicles_per_hour`)
SELECT
  TRIM(`A`)                                          AS borough,
  NULLIF(REPLACE(TRIM(`B`), ',', ''), '') + 0        AS co2_kg_per_h,
  NULLIF(REPLACE(TRIM(`C`), ',', ''), '') + 0        AS co_g_per_h,
  NULLIF(REPLACE(TRIM(`D`), ',', ''), ''), + 0       AS pm25_g_per_h,
  NULLIF(REPLACE(TRIM(`I`), ',', ''), '') + 0        AS vehicles_per_hour
FROM table1_raw;
```

Back-end data processing

```javascript
const pool = mysql.createPool({
  host: process.env.DB_HOST,
  port: process.env.DB_PORT,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME
});

app.get("/api/health", async (req, res) => {
  try {
    const conn = await pool.getConnection();
    await conn.ping();
    conn.release();
    res.json({ ok: true });
  } catch (e) {
    res.json({ ok: false, error: e.message });
  }
});
```

Backend configuration

Frontend Development and Interactivity - Zihang He

The interactive London map, created by He Zihang, is a core component of the EcoVision website. We used this website to visualize real-time air quality data for London's 33 boroughs. The map is built on Vue 3 and the Composition API, and uses Vite for rapid development and efficient production builds. In the project's single-page architecture, the map view (Map.vue) is registered through Vue Router and integrates several modular components: TimeSlider, MapLegend, BoroughPopup, and RoutePopup. These components are used for different functions, such as controlling time, legend display, and dynamic data dashboards.

First, a suitable base map needs to be selected, allowing switching between 2D and 3D perspectives. Therefore, Zihang chose to use Mapbox GL JS and Mapbox Standard styles for rendering. The base map needs to clearly delineate London's regions, so regional data needs to be collected. The boundary data comes from the Greater London Authority's statistical GIS boundary files, which were simplified using Mapshaper and converted to GeoJSON format. The dataset (london_boroughs.geojson) is loaded into Mapbox as a custom data source and styled using multiple layers: a fill layer (BOROUGH_FILL_ID) for coloring based on PM2.5 concentration; a boundary glow layer (BOROUGH_BORDER_GLOW_ID) and an outer glow layer (LONDON_OUTER_GLOW_ID) for visual differentiation; and a symbol layer (BOROUGH_LABEL_ID) to display borough names with glowing text rendering for improved readability. To ensure the camera automatically selects London areas during loading, the overall bounding boxes for all boroughs are precisely calculated using Mapbox.

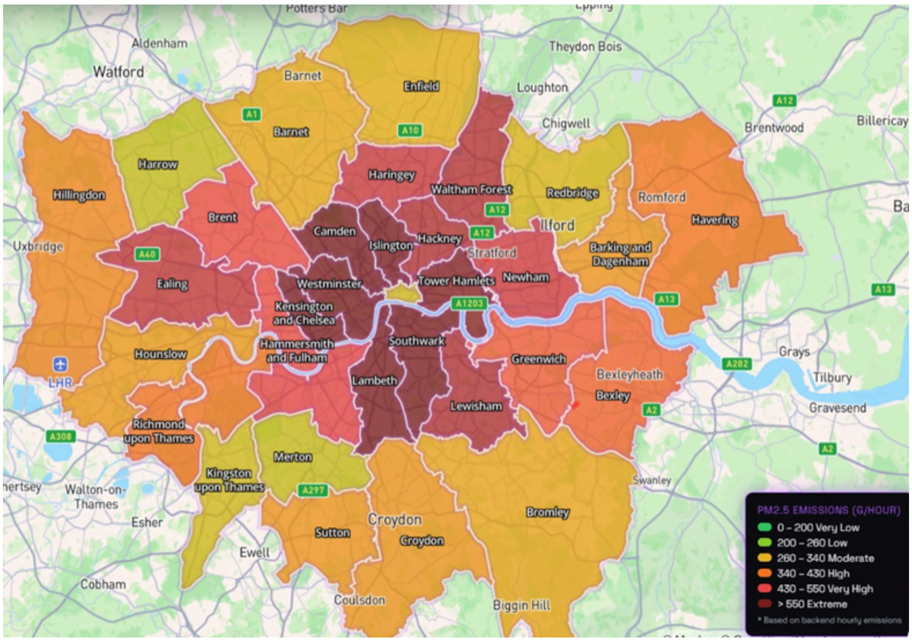
```
src
  assets
  components
    BoroughPopup.vue
    MapLegend.vue
    RoutePopup.vue
    TimeSlider.vue
```

```
import MapLegend from "../components/MapLegend.vue";
import BoroughPopup from "../components/BoroughPopup.vue";
import TimeSlider from "../components/TimeSlider.vue";
import RoutePopup from "../components/RoutePopup.vue";
```

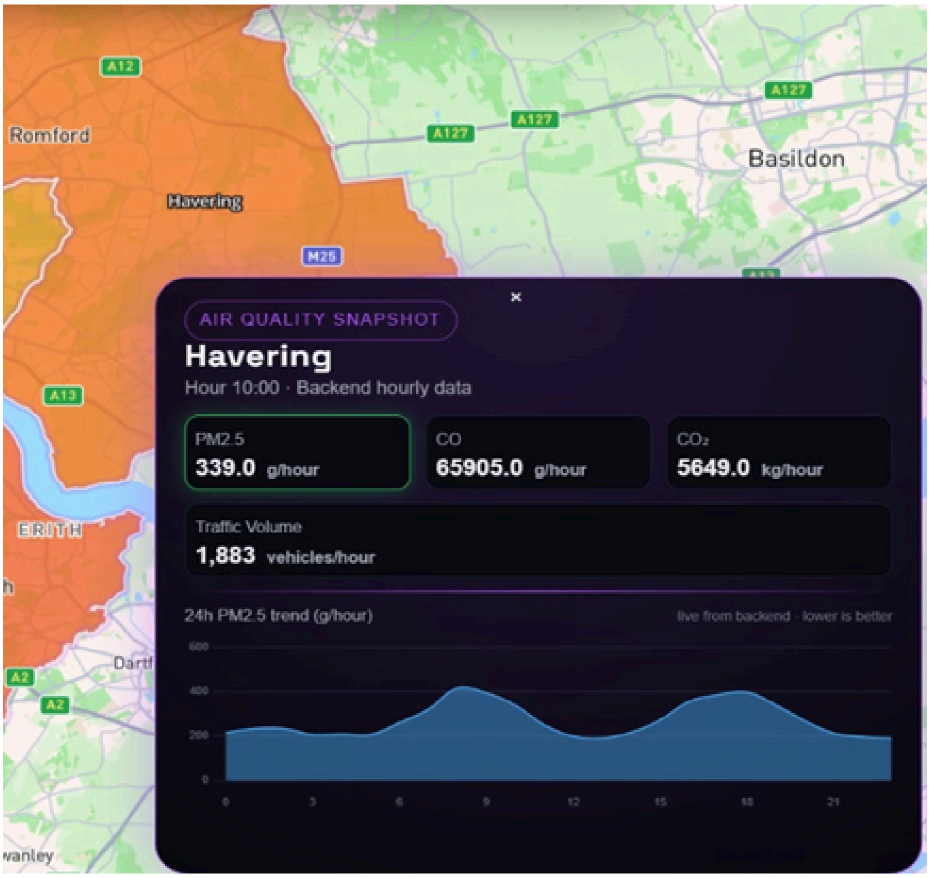
Components

Our team was committed to implementing a complete front-end and back-end project, so obtaining real-time data via a real API and rendering it was extremely important. The data layer communicates with the Node.js + Express backend via a RESTful interface.

Zihang used regular expressions in map.vue to normalise the name of each administrative district, ensuring consistency with the database ID (e.g., borough_hour), thus correctly requesting hourly records from /api/hourly. The backend data includes PM2.5, CO, CO_2, and traffic flow. These metrics are merged into a GeoJSON feature set via a custom function (buildStyledGeojson) and dynamically rendered on the map. Linear interpolation expressions convert the PM2.5 range into a green-to-red gradient (combined with the scale automatically changing color according to the selected time), and the legend component (MapLegend) also uses the same scale for user reference. All updates are reactively executed through Vue's state system, without requiring page reloads.



Three layers

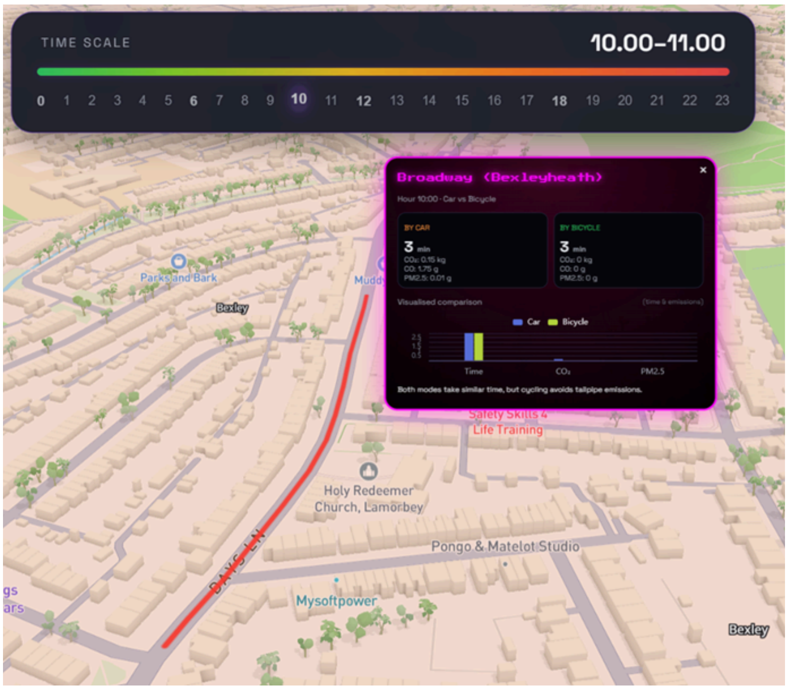


data pop-up windows

The most crucial synchronization aspect of this project is controlled by a time slider. The `TimeSlider` component enables time switching, binding its value to the `activeHour` variable. When the user changes the slider, the map automatically re-fetches data for the selected hour and updates the color intensity for each district. Clicking on any district triggers a Mapbox popup, which mounts a

`BoroughPopup` Vue component and displays real-time metrics and chart placeholders within the map container.

Time



pop-up windows

Project Management and delivery - Madina Diallo

Madina guided the EcoVision project from concept to delivery, ensuring that all workstreams, research, data, design, and technical development, remained fully aligned with the project scope, timelines, and quality standards.

She established a clear project structure through **weekly milestones, sprint planning, and regular review meetings**, fostering collaboration and accountability across the team. Madina facilitated effective communication, coordinated GitHub version control, and ensured coherence between backend logic and frontend design so that every dataset, visualisation, and line of code contributed to a unified, functional, and visually engaging platform.

Her responsibilities included **task scheduling, risk and issue management, and progress tracking**, as well as maintaining documentation to support transparency and consistency throughout the development process. Madina also verified data accuracy and visual consistency, managed all project documentation, and oversaw the final presentation and report editing, ensuring alignment with the **assessment criteria** and the overall project vision.

Through her structured leadership and team-focused approach, Madina helped maintain strong momentum, clear communication, and a shared sense of purpose that drove EcoVision to successful completion.

Evaluation and Reflection

The final EcoVision website successfully delivers a coherent and visually engaging representation of London's air-quality dynamics. User testing revealed that participants intuitively understood the map's colour coding and time-slider interaction. Feedback praised the clarity of the layout and educational value of the cycling vs car solutions .

Challenges were inevitable. Integrating datasets from multiple sources required extensive data wrangling. Backend deployment faced configuration errors, and synchronising front-end animations occasionally caused lag. Despite these challenges, the project met its objectives of creating an accessible, data-driven storytelling platform. The process strengthened our collaborative, analytical, and technical skills, reflecting the interdisciplinary nature of connected environments.

Future Improvements

If given more time, the EcoVision team would extend the platform by incorporating real-time API feeds from DEFRA, enabling users to monitor current air quality. A mobile-first redesign would enhance accessibility for commuters. Additional features could include comparative dashboards allowing borough-by-borough analysis and a user feedback portal where citizens can share local air-quality observations. On the backend, migrating to a cloud-based architecture such as MongoDB Atlas or Firebase would improve scalability. Integrating predictive analytics using Python's Scikit-learn could forecast pollution patterns, offering proactive insights for city planners and communities.

From a technical perspective, our current project relies on preprocessed static datasets provided by the backend to the frontend. While this approach ensures performance and robustness, it also limits the platform's temporal depth; we cannot use real-time data. In future iterations, if time permits, we hope to integrate real-time environmental data streams from open APIs or IoT sensor networks, enabling users to observe real-time changes in air quality and traffic emissions. This will require deploying a more dynamic backend capable of managing continuous data ingestion, validation, and caching, potentially leveraging technologies like WebSocket or MQTT for real-time communication.

The frontend architecture can iterate towards a more scalable and responsive design. Improvements in data visualisation could also include multi-layered overlays, combining air quality data with other metrics such as temperature, humidity, noise pollution, or traffic density. These layers will allow users to explore the interrelationships between variables and understand broader environmental dynamics.

Beyond data integration, the map can also support user interaction and engagement. Adding features such as community-driven data uploads or personalised dashboards, which can transform the platform from a static viewer into an active citizen science tool. Integrating machine learning models to predict pollution levels or simulate emission reduction schemes will further enhance its analytical value.

Conclusion

EcoVision is more than a website, it is a statement about how technology and storytelling can converge to promote environmental awareness. Through interdisciplinary teamwork, the project evolved from an abstract idea into a tangible digital product that visualises London's atmospheric data with clarity and empathy. The experience reinforced that data, when combined with thoughtful design, becomes a catalyst for change. The EcoVision journey demonstrated how collaborative learning, technical innovation, and creative problem-solving can together produce a meaningful contribution to the Connected Environments discipline.

The EcoVision project demonstrates how open data, web technologies, and design thinking can converge to address pressing urban sustainability challenges. By translating emissions datasets into an interactive visual narrative, the team has built not only a website but also a tool for awareness and change. Our collective effort displayed the synergy between research, creativity, and technical execution. The iterative design process, from sketching to coding, reflected our shared belief that **understanding data should be as intuitive as breathing the air we seek to improve**.

EcoVision stands as a prototype for future public-facing environmental dashboards, one that could evolve into a real-time urban monitoring platform inspiring citizens, policymakers, and researchers alike. The EcoVision project demonstrates how open data, web technologies, and design thinking can converge to address pressing urban sustainability challenges.

By translating emissions datasets into an interactive visual narrative, the team has built not only a website but also a tool for awareness and change.

References

- Batty, M. (2018) Inventing Future Cities. Cambridge, MA: MIT Press.
- Cox, M., Longley, P. and Shao, Z. (2022) 'Interactive geovisualisation for environmental awareness', Environment and Planning B: Urban Analytics and City Science. [Online]. Available at: https://journals.sagepub.com/home/epb

- Department for Environment, Food & Rural Affairs (DEFRA) (2023) Air Quality Data API. [Online]. Available at: https://uk-air.defra.gov.uk/data
- Greater London Authority (GLA) (2023) London Atmospheric Emissions Inventory. [Online]. Available at: https://data.london.gov.uk
- Heer, J., Bostock, M. and Ogievetsky, V. (2010) 'A tour through the visualization zoo', Communications of the ACM, 53(6), pp. 59–67.
- Munzner, T. (2014) Visualization Analysis and Design. Boca Raton, FL: CRC Press.
- Nielsen, J. (1994) Usability Engineering. San Francisco, CA: Morgan Kaufmann
- Transport for London (TfL) (2023) Open Data Portal. [Online]. Available at: https://tfl.gov.uk/info-for/open-data

Team Members and Report Authors

Members:

- **Madina Diallo**
- **Matilda Nelson**
- **Yidan Gao,**
- **Zihang He**
- **Qingshan Luo**

Project Website: http://10.129.111.18:3000/casa0017-web-assessment-EcoVision/
GitHub Repository: https://github.com/xms12138/casa0017-web-assessment-EcoVision