

# U.eNergy: An Interactive Energy Visualization

Benjamin Sanson

School of Engineering and Computer Science  
Victoria University Wellington  
sansonbenj@myvuw.ac.nz

**Abstract**—U.eNergy is an interactive energy visualization system designed to reveal insights and show trends in energy generation. The system uses International Energy Agency (IEA) data for countries in the Organisation for Economic Co-operation and Development (OECD) for the years 2010 to 2022. There are four visualizations offered: individual country profile, global energy, production groups and producer hierarchy. Energy production relates to the UN sustainable development goals of affordable and clean energy (7), and climate action (13). The system uses bar charts, tree maps, and force-directed layout graph visualizations. The visualizations are powered by the D3 JavaScript visualization library. The remainder of the website was created with the Next.js react framework. We developed a user model with two personas to meet the needs of the system’s intended audience. We offer a reflection on the system’s ability to meet the needs of the user. A critique of the project, including the tools used, is provided.

**Index Terms**—interactive visualization, OECD, user model, information-seeking mantra, UN sustainable development goals

## I. INTRODUCTION

U.eNergy is designed to help people clear their misconceptions about global energy generation. Questions like which nation is the largest producer of nuclear energy or the largest user of coal become trivial to answer with U.eNergy. Leveraging modern web development frameworks and artificial intelligence (AI) powered programming tools we have created a powerful visualization system to support the UN sustainable development goals [1]. Specifically, this tool aids in the progression of affordable and clean energy (7) [2] by showing users which forms of energy production are in use. Goal 13, climate action [3], is also advanced by educating users about the continued use of fossil fuels and highlighting each country’s progress towards using renewable energy.



Fig. 1. The home page of U.eNergy

## II. INTERACTIVE VISUALIZATION SYSTEM DESCRIPTION

### A. Visualizations

U.eNergy encodes information visually with four visualizations. Each visualization has its own controls.

- **Country Profile** An animated bar chart showing the energy production of a single country over time. An additional line chart also shows the nation’s yearly total energy production trend. Colour mapping helps the user identify energy production methods and summary statistics are shown. The controls for the country profile visualization are a text box to select a country, an animation play/pause button and a year selection slider. Figure 2 shows the country profile view for Sweden.

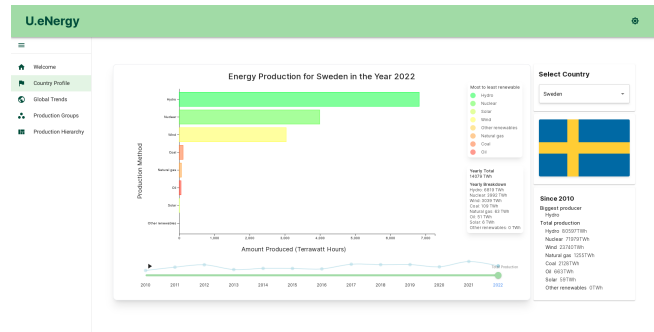


Fig. 2. The country profile view of Sweden

- **Global Energy** Another animated bar chart that relates the total energy generation of each country by sorting the nations from most energy produced to least energy produced. The visualization can also rank countries by most energy produced by sustainable generation methods to least sustainable. Global energy’s controls are a menu to select the energy or sustainability view, a year selection slider and play/pause button and a slider to select the range of countries shown, for example, the top 10 most sustainable countries. Figure 3 shows the top 10 largest producers of energy in the year 2022.
- **Production Groups** A force directed layout algorithm graph. Countries with a similar energy generation distribution are linked together in the graph, revealing previously unseen relationships between nations. The top-generation method is used to label each country in the graph with a colour. The difference in total energy produced determines the thickness of the edge linking

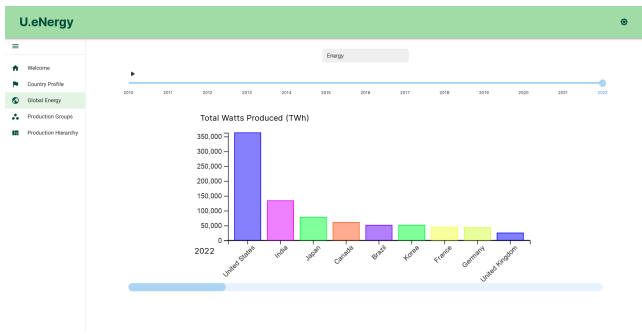


Fig. 3. The top 10 energy producers as shown by the global energy page

countries together. The controls for the production group visualization are energy generation method filtering by clicking on the method the user wishes to highlight, buttons to increase/decrease the decision threshold for whether countries are linked in the graph, a year selector, an information popup when the user clicks on a node in the graph, and zooming, panning and node dragging with the mouse/scroll wheel. The information popup has a link to quickly travel to the country profile for the selected nation. Figure 4 shows the production groups view of the year 2022, with hydro nations highlighted and the nation of Sweden selected in the details popup.

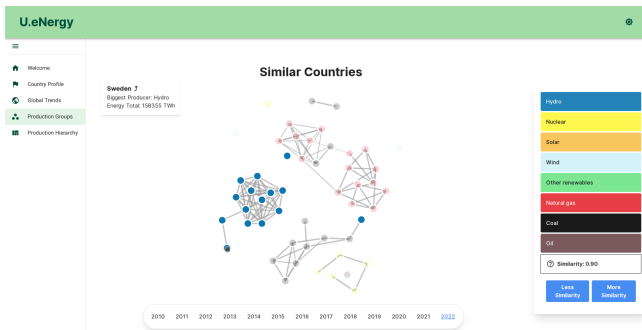


Fig. 4. The production groups view of 2022 with hydro producers highlighted. 0.9 similarity threshold selected

- Producer Hierarchy** A treemap visualization that shows which energy generation methods produce the most energy overall. Clicking on a cell in the tree map reveals which continents contribute the most to each energy generation method. Within each continent, cells show individual countries and their relative contribution of energy for the production method. A year-selector allows the user to see data from a specific year. A details popup shows the country that the mouse is hovering over and the amount of energy that country produced. The details popup has a link to the country profile page for the selected nation. The controls are: panning and zooming with the mouse and clicking on cells in the tree map. Figure 5 shows the contribution of each generation method for 2022 and Figure 6 shows the largest producer of nuclear energy in North America.



Fig. 5. Production hierarchy view of the contribution of each energy generation method to the global total

### B. Data source and processing

U.eEnergy uses OECD data from the IEA [4]. The data contains instances representing one country's energy generation by a single generation method for a single year. Data processing involves filtering undesired instances until only the instances needed to produce the visualization remain.

All visualizations begin by filtering out instances that are not from the selected year.

- Country profile** filters all the instances that do not belong to the selected country. The remaining instances are reduced by their energy generation method to single data points which can then be visualized.
- Global energy** reduces all instances belonging to a given country down to a single data point: total energy produced or % of energy from renewable methods.
- Production Groups** visualizes the data by reducing each country's instances to a 7-dimensional vector containing the energy produced by each generation method. The vector dot product is used to calculate the cosine of the angle between each country's vector. If the cosine of the angle is more than a user-defined threshold then the nations will share an edge in the resulting graph.
- Producer hierarchy** processes the data by creating a graph with a root node that contains energy generation method nodes which have continents as child nodes. The continent nodes' children are the individual countries

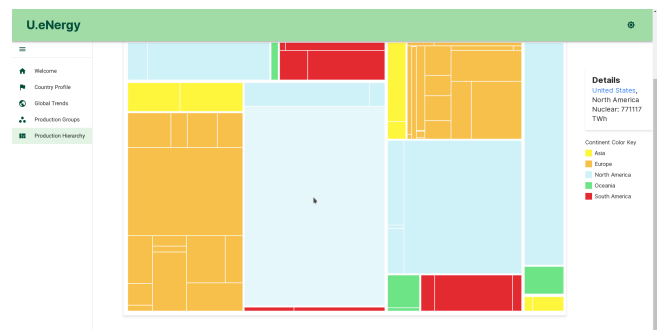


Fig. 6. The nuclear energy cell, with the largest North American producer selected

which form the leaves of the graph.

### III. SYSTEM COMPONENT INTERACTION DISCUSSION

The global energy visualization, producer hierarchy visualization, and production group visualization all interact with the individual country profile visualization. The country profile visualization serves as a sort of detail on demand for an individual country. The other visualizations reveal relationships between separate countries. The user can quickly reach the country profile view for any country they select in each of the other visualizations. Observe in Figure 4, the information popup contains a link back to Sweden's country profile page. The selected year is also preserved when transitioning to the country profile page from any of the other visualizations.

The visualizations complement each other as producer hierarchy shows aggregate totals for energy production, global trends page shows relative rankings of countries and the production groups view reveals which countries are similar.

An example journey through U.eNergy could be starting on the production groups page and seeing that Bulgaria and the Czech Republic are similar. Following the link to each country's profile view reveals that both nations use coal, nuclear, natural gas and hydro as their top four production methods. Visiting the production hierarchy shows that despite both countries being majority coal, combined, they use less coal than Germany but more than Canada. Taken together the visualizations allow the user to develop a comprehensive understanding of energy generation in the OECD.

### IV. REFLECTION ON INFORMATION SEEKING MANTRA IMPLEMENTATION

Ben Shneiderman's information-seeking mantra is a taxonomy of tasks that users complete when interacting with information visualizations [5]. It is useful to reflect on whether U.eNergy gives users the ability to complete the information-seeking mantra to determine its usefulness as a visualization tool. U.eNergy implements the information-seeking mantra.

- **Overview.** An overview of the whole system is provided by the global energy, production groups and production hierarchy visualizations. Each visualization gives an overview of how countries relate to each other by their respective metrics (GWh produced, % sustainability, generation similarity, and proportion of production) by starting in a zoomed-out state. For example, in each of the aforementioned visualizations respectively, the overview shows the user: The United States of America (USA) produces far more energy than any other country, there is a cluster of highly similar countries producing energy by hydro, and coal, oil and natural gas combined produce about the same amount of energy as wind, hydro, solar, and nuclear.
- **Zoom.** Zooming is implemented in the production groups and production hierarchy visualizations. The user can zoom with their mouse scroll wheel. Zooming allows the user to see the intricate relationships between elements and details that are difficult to see at a distance. For

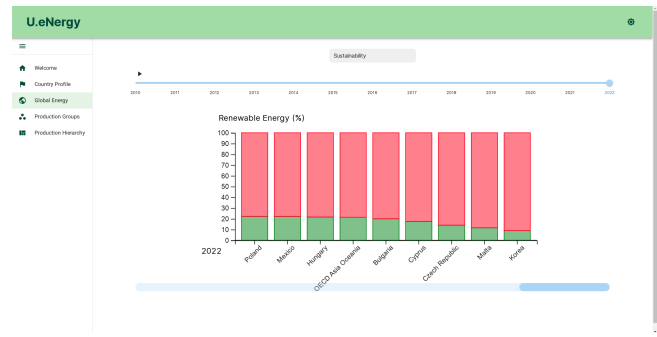


Fig. 7. Global energy view showing the 9 least sustainable countries

example, in the production hierarchy, zooming in allows the user to see that Cyprus was the smallest producer of energy by wind generation in Europe for 2022.

- **Filter.** Filtering is implemented in all visualizations. In the country profile, the user filters out a single country that interests them. Global Energy has a range slider that allows the user to filter out a range of values from the whole data set. Moving the slider to the left shows the top 9 countries and moving the slider to the right shows the bottom 9 countries. The size of the range can also be changed from the entire set down to just one element. The Production Groups visualization has buttons to filter out uninteresting energy generation methods. For example, clicking coal will highlight only the nations that have coal as their top method. Production hierarchy can filter out uninteresting values by zooming and panning the view.
- **Details on Demand.** The country profile serves as the ultimate detail on demand, showing all the details of a single country. To provide details on demand in the other visualizations users can click on countries that interest them and are given the option to visit the selected country's profile page along with contextual detail relevant to the visualization. For example, figures 4 and 6 show the details popup in the production groups and production hierarchy visualizations.
- **Relate.** Aside from the country profile page, all visualizations allow the user to see how different countries relate to each other by their respective metrics. As seen in Figure 4, Production Groups relates countries with similar energy production methods together. Production hierarchy relates countries geographically and by total energy production. Figure 6 shows how each continent contributes to nuclear power generation, North America produces about the same amount of nuclear energy as Europe. Global Energy shows the ordinal rankings of countries for total energy produced and % of energy produced by sustainable methods. Figure 7 shows the 9 worst countries for sustainable energy generation.
- **History.** The need for an action history and undo is limited in U.eNergy. All actions can quickly be undone. For example, zooming in is cancelled by zooming out, and panning left is cancelled by panning right. However,

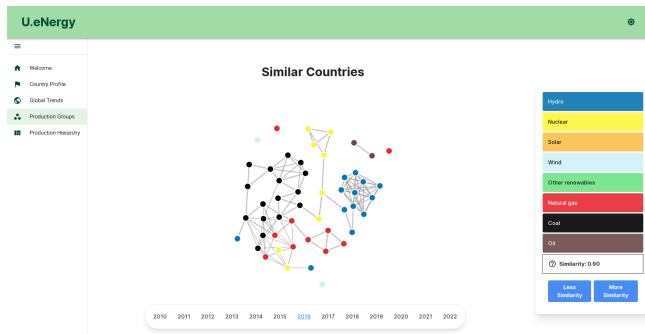


Fig. 8. Similar countries grouped together in 2016 with a 0.9 similarity threshold

action history is preserved for the year selection. When the user transitions from a visualization to the country profile page, the year the user was selected is remembered so they can immediately continue their exploration where they left off on the last visualization. For example, in Figure 4, the year 2022 is selected. Clicking the Sweden link would open the country profile seen in Figure 2, and the selected year is preserved.

- **Extract.** The extraction of query parameters is ever present in U.eNergy as the selected year that data is pulled from. Additionally, in the production groups visualization, the similarity threshold that determines the groupings of similar countries is displayed to the user so it can be preserved and revisited later. Figure 4 shows the groupings of countries in 2022 with a 0.9 similarity. Figure 8 shows the groupings in 2016 with the same similarity threshold. The Country Profile and Production Groups visualizations also allow the displayed results to be exported to a file so they can be distributed.

## V. CRITIQUE OF SYSTEM'S ABILITY TO MEET USER NEEDS

We developed 2 personas, 3 key-path scenarios, and 5 use cases for U.eNergy. This user model captures the requirements and characteristics of U.eNergy's user base. The user model can be viewed in the user model appendix.

### A. Evaluating U.eNergy

We performed cognitive walkthroughs [6] with each persona to confirm that U.eNergy is aligned with the needs of its user base.

- **Aroha Goals.** In the global energy page, Aroha selects the sustainability view and moves the range slider to the end. She sees Korea, Japan and Poland are among the bottom 10 sustainable countries. In the production groups visualization, she finds that many Icelandic countries share similar energy generation distributions.
- **Aroha Frustrations.** Aroha can discover how much energy in New Zealand (NZ) is produced by coal and hydro by visiting NZ's profile page. She can learn NZ's relative generation by locating NZ in the global energy visualization, it is the 15th smallest producer of energy

in the OECD. Aroha finds which continents use nuclear power and that there are 21 nuclear power producers in the production hierarchy visualization.

- **Mozillo Goals.** Mozillo achieves his first goal by visiting the production hierarchy page. To learn which country produces the most energy he visits the global trends page and confirms his suspicion that it is the USA. Mozillo finds the most sustainable countries changing to the sustainability view and finds that Norway is consistently in the top 2 most sustainable countries from 2010 to 2022.
- **Mozillo Frustrations.** To find the smallest energy producers Mozillo uses the range slider on the global energy page and sees Luxembourg, Malta, and Cyprus are the smallest energy producers in the OECD.

The cognitive walkthroughs show that U.eNergy is highly aligned with the needs of its user base.

## VI. CRITIQUE OF SOFTWARE DEVELOPMENT TOOLS

We made use of a variety of tools during the development of U.eNergy. An exploration of each tool's strengths, weaknesses, constraints, and opportunities is provided.

- **Typescript language.** Typescript (TS) is a programming language built on top of JavaScript (JS) by Microsoft [12]. Typescript adds a type system but otherwise acts similarly to JS. We found the addition of a type system did not help development. We frequently reverted to inspecting objects logged to the console to figure out the data types and using TS's 'any' type and 'ignore' annotation to get around strict and confusing D3 type requirements.
- **D3 Visualization library.** D3 is a JS library that abstracts the assembly and rendering of visualizations [7]. We found that D3 was able to provide the visualizations we needed. It was clear that the library is designed to work with JS rather than TS because we encountered many type errors and frustrations while using D3 in TS. D3 added a constraint to the tree map as we were unable to traverse the graph one layer at a time because the click event handler didn't specify which cell the user clicked on.
- **Next.js react framework.** Next is a react framework that abstracts the updating of the web page in response to user inputs [8]. Using Next was a quality-of-life improvement as the framework automatically updates the page when developer code modifies the DOM keeping the client view synced with the server. It allowed the creation of reusable components such as the navigation menu. Next was useful for increasing the velocity of development. It did not constrain the system.
- **Git.** Git is a source control management tool [9]. The usage of Git allowed the developers to independently work on their visualizations without affecting the other developers' code with Git's branching feature. Git keeps a record of additions allowing unintended changes to be reverted, which saved development time.



- **Visual Studio Code (VSC).** VSC is an integrated development environment (IDE) [10]. VSC met the needs of the development team. The developers were unable to get the VSC-integrated debugger working, constraining them to use console log debugging. The Git integration feature made using Git easier than Git via the command line interface.
- **Tailwind CSS framework.** Tailwind (TW) is a framework that abstracts website styling [11]. The development team did not have to write any manual CSS which greatly increased the speed of development and reduced cognitive load. The provided styling afforded the opportunity to give the system a professional appearance.
- **Copilot.** Github’s Copilot is an IDE extension that allows an AI to generate context-aware code suggestions [13]. Copilot was beneficial once the tool gathered enough context about the developer’s intentions. The extension effectively wrote the application code while the developer served in a pair programmer role, increasing the development speed, and affording the opportunity to add additional features. When the extension did not have enough context to deduce the developer’s intention it would become a nuisance, suggesting detrimental code that did not advance the objectives of the visualization.
- **ChatGPT.** OpenAI’s ChatGPT is an advanced chatbot [14]. ChatGPT was useful for gathering contextual information, like searching the web for tutorials. The tool was able to occasionally debug developer code, an impressive feat. The tool provided the opportunity to add additional functionality such as the export functionality. The weakness of ChatGPT is the detrimental effect it has on development when it does not understand the code the developer is asking it to adjust. Because the tool sounds authoritative, developers often went with the tool’s suggestions, even when they were incorrect, wasting important development time. This happened when implementing the tree map and was resolved by writing the tree map ‘AI free’.
- **Figma.** Figma is a design tool [15]. Figma was an enormous benefit for the development of u.eNergy. It supports real-time collaboration through web browsers and the creation of user models and wireframes. This allowed the team to quickly form a shared vision for the tool and proceed with development with clearly defined requirements. Figure 9 shows the Figma plan for the production groups visualization, observe the finished visualization’s similarity in Figure 4.
- **Gitlab.** Gitlab is a project management tool [16]. Gitlab’s issue system allowed the delegation of responsibilities to developers, ensuring they knew what they needed to create. The tool tracked the progression of each developer, allowing the team to assist struggling developers, and keeping the project on schedule. The tool also provides a record of team discussion so the rationale behind decisions is preserved. We did not identify any weaknesses with the tool and it did not add constraints to the project.

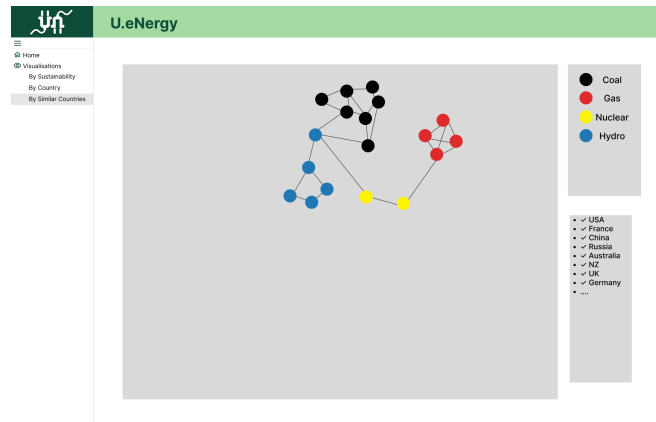


Fig. 9. Production group wire frame generated in Figma

## VII. REFLECTION ON PROJECT

When reflecting on this project and what we got right and wrong, several key themes emerge. The productivity boost of AI tools, using a web development framework and its conventions, following strong human-created guides for external libraries, and leveraging design tools.

**AI Tools.** If we were to do this project again AI tools like ChatGPT and Github Copilot would be a part of it. The productivity increase from leveraging these tools cannot be understated. A healthy understanding of the risks of these tools is required before using them otherwise developers can easily be led astray trying to use code that has no hope of working. AI is good for simple tasks like code completion. When the developer already knows what the code should do the tool simply writes it for them after inferring their intention from the context, saving time. It is more ambitious to expect the tool to write your code for you when you have no idea how you would tackle the problem in the first place.

**Frameworks.** The Next.js and Tailwind frameworks were good time savers and allowed the website to have a professional appearance. If we were to repeat the project, these frameworks would continue to be used. However, it is important to learn their conventions and expectations beforehand. Often, the framework is already able to solve problems that the developers faced, but a lack of knowledge prevented them from leveraging the framework. Functionality like shared page state would have allowed the data to be loaded a single time instead of each page loading the data individually. Widgets and tooltips can be made with the framework provided the developer knows how to use it. Component sharing functionality meant that there was an opportunity for common functionality like year selection to be reused instead of implementing functionality from scratch on each page.

**Human tutorials.** Relying on ChatGPT to try and create the treemap was a disastrous aspect of this project. There were several high-quality human-made tutorials on how to make a tree map with D3 that were ignored in favour of ChatGPT code that did not work. An entire day of productivity was lost trying to fix the AI code before the team decided to cut its

losses and start again with a proper guide. This demonstrates the importance of not relying on AI tools as an oracle. Instead, AI tools should be used to complete monotonous and uninteresting tasks. A human should be making high-level implementation decisions.

**Design Tools.** Figma should be used if we were to do this project again. Design ideas often sound appealing in theory but are found insufficient in implementation. Figma allowed the design to be validated before implementation, a time when it is cheap to make changes. The result was a website mock-up that gave the team a shared vision before implementation. The mock-up could be referred back to during implementation making that stage of the project relatively smooth.

### VIII. CONCLUSION

U.eNergy is an interactive information visualization system that uses IEA data of OECD countries to clear users' misconceptions about global energy generation. It uses bar charts, force-directed layout graphs, tree maps, and line graphs to present data in an understandable and appealing way. Interactivity, animation and the relation between visualizations allow the tool to align with the needs of the user base. The system implements each aspect of the information-seeking mantra, resulting in a usable system. Cognitive walkthroughs were used to test if the system's functionality was aligned with the personas that were developed to represent the needs of the user base. AI tools were found to increase productivity but not developer understanding. Development frameworks allowed the website to be professional in appearance. Project management tools kept the project in scope and on schedule. If this project were to be repeated, it is suggested that the same tools and approach be used, but the conventions of the frameworks are followed and AI tools be treated with more scepticism.

### REFERENCES

- [1] United Nations, "The 17 sustainable development goals," United Nations, 2015. [Online]. Available: <https://sdgs.un.org/goals> [accessed Jun. 19, 2023].
- [2] United Nations, "Goal 7 — Department of Economic and Social Affairs," [sdgs.un.org](https://sdgs.un.org/goals/goal7), 2022. [Online]. Available: <https://sdgs.un.org/goals/goal7> [accessed Jun. 19, 2023].
- [3] United Nations, "Goal 13 — Department of Economic and Social Affairs," United Nations, 2022. [Online]. Available: <https://sdgs.un.org/goals/goal13> [accessed Jun. 19, 2023].
- [4] CCBAN23 "Monthly Electricity Production in GWh [2010-2022]," [www.kaggle.com](https://www.kaggle.com/datasets/ccanb23/iea-monthly-electricity-statistics). [Online]. Available: [www.kaggle.com/datasets/ccanb23/iea-monthly-electricity-statistics](https://www.kaggle.com/datasets/ccanb23/iea-monthly-electricity-statistics) [accessed Jun. 19, 2023].
- [5] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," Proceedings 1996 IEEE Symposium on Visual Languages, Boulder, CO, USA, 1996, pp. 336-343, doi: 10.1109/VL.1996.545307.
- [6] J. Nielsen and R. L. Mack, "Usability Inspection Methods", 1st ed. New York: John Wiley & Sons, 1994, pp. 105-140.
- [7] M. Bostock, "D3.js - Data-Driven Documents," [D3js.org](https://d3js.org/), 2013. [Online]. Available: <https://d3js.org/> [accessed Jun. 19, 2023].
- [8] Vercel, "Next.js by Vercel - The React Framework," [nextjs.org](https://nextjs.org/), 2016. [Online]. Available: <https://nextjs.org/> [accessed Jun. 19, 2023].
- [9] Git, "Git," [Git-scm.com](https://git-scm.com/), 2019. [Online]. Available: <https://git-scm.com/> [accessed Jun. 19, 2023].

- [10] Microsoft, "Visual Studio Code," [Visualstudio.com](https://code.visualstudio.com/), Apr. 14, 2016. [Online]. Available: <https://code.visualstudio.com/> (accessed Jun. 19, 2023).
- [11] Tailwind Labs, "Tail Wind CSS," [GitHub](https://github.com/tailwindlabs/tailwindcss), Oct. 21, 2022. [Online]. Available: <https://github.com/tailwindlabs/tailwindcss> [accessed Jun. 19, 2023].
- [12] Microsoft, "TypeScript - JavaScript that scales.," [Typescriptlang.org](https://www.typescriptlang.org/), 2015. [Online]. Available: <https://www.typescriptlang.org/> [accessed Jun. 19, 2023].
- [13] GitHub, "GitHub Copilot · Your AI pair programmer," [GitHub.com](https://github.com/features/copilot), 2023. [Online]. Available: <https://github.com/features/copilot> [accessed Jun. 19, 2023].
- [14] OpenAI, "Introducing ChatGPT," [OpenAI](https://openai.com/blog/chatgpt), Nov. 30, 2022. [Online]. Available: <https://openai.com/blog/chatgpt> [accessed Jun. 19, 2023].
- [15] Figma, "Figma: the collaborative interface design tool.," [Figma.com](https://www.figma.com/), 2022. [Online]. Available: <https://www.figma.com/> [accessed Jun. 19, 2023].
- [16] GitLab, "The first single application for the entire DevOps lifecycle - GitLab," [GitLab.com](https://about.gitlab.com/), 2016. [Online]. Available: <https://about.gitlab.com/> [accessed Jun. 19, 2023].