Creating a website to visually demonstrate how data is handled using different algorithms and data structures can be a fascinating and educational project. Here's a step-by-step guide to help you get started:

1. **Define the Scope and Content:** Decide which algorithms and data structures you want to showcase on your website. Keep it focused and manageable, as each algorithm and data structure might require significant effort to implement visually.
2. **Choose a Tech Stack:** Consider the technologies you'll use for building the website. Common options include HTML, CSS, JavaScript for the frontend, and a backend language like Python, Java, or Node.js to handle server-side processing.
3. **Design the Website:** Create a design for your website, focusing on user interface and experience. Ensure the design is intuitive and visually appealing, as it will enhance user engagement.
4. **Implement the Data Structures and Algorithms:** Start building the visual representations and functionalities for each data structure and algorithm. You'll need to simulate how they work and handle data.
5. **Frontend Development:** Implement the frontend of the website using HTML, CSS, and JavaScript. Use libraries like D3.js or Chart.js to create interactive visualizations.
6. **Backend Development (Optional):** If your website requires server-side processing, set up a backend using your chosen language and framework. The backend may also be necessary if you want to include user accounts, saving progress, etc.
7. **Interactive Features:** Make the visualizations interactive. Users should be able to input their data and see how the algorithms and data structures handle it.
8. **Documentation:** Include detailed explanations for each algorithm and data structure. Provide examples and use cases to help users understand their applications.
9. **Testing:** Test your website thoroughly to identify and fix any bugs or issues. Ensure it works smoothly on different devices and browsers.
10. **Deployment:** Choose a web hosting service and deploy your website so that it's accessible to users worldwide.
11. **Gather Feedback and Improve:** Encourage users to provide feedback on the website's usability and usefulness. Based on their input, make necessary improvements and updates.
12. **Expand the Content (Optional):** Once your initial version is up and running, you can consider expanding the website to include more algorithms and data structures or add advanced features.

Remember that visualizing data structures and algorithms can be complex, so it's crucial to take your time and build each component with care. Good documentation and clear visualizations will be key to making your website valuable for learners and developers. Happy coding!

1. Data structure: Hash table
2. Algorithms: Search algorithms and Sorting

Front end: HTML, CSS, JS

For interactive: D3.js/ chart.js

Make it user input