



Unleash your potential



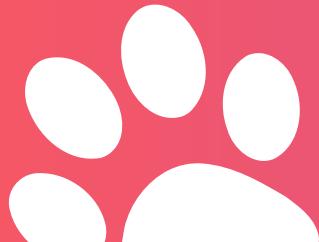
DEVELOPER GUIDE





1. INTRODUCTION

This guide is made for developers who want to understand ,set up and maintain the pet care website. It explain the tool used, setup steps, the folder structure and how the code works. The main purpose is to make it easy for new developers to run the project on their system and update it when needed.

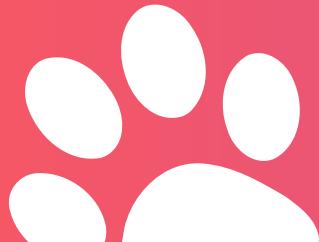




2. TECHNOLOGIES USED

This project is created using simple and modern web technologies:

- HTML → For structure of web pages
- CSS → For styling and design
- JavaScript (JS) → For adding functionality
- JSON file → For storing and fetching data



3. DEVELOPMENT ENVIRONMENT SETUP

To run my Website, you first need to install Vs Code .

Then, Unzip my Project file .

After that, open the Project folder in Vs code .

Next, Open the Index.html file.

Finally, Run it using live server.

Your Website Will Open Successfully.

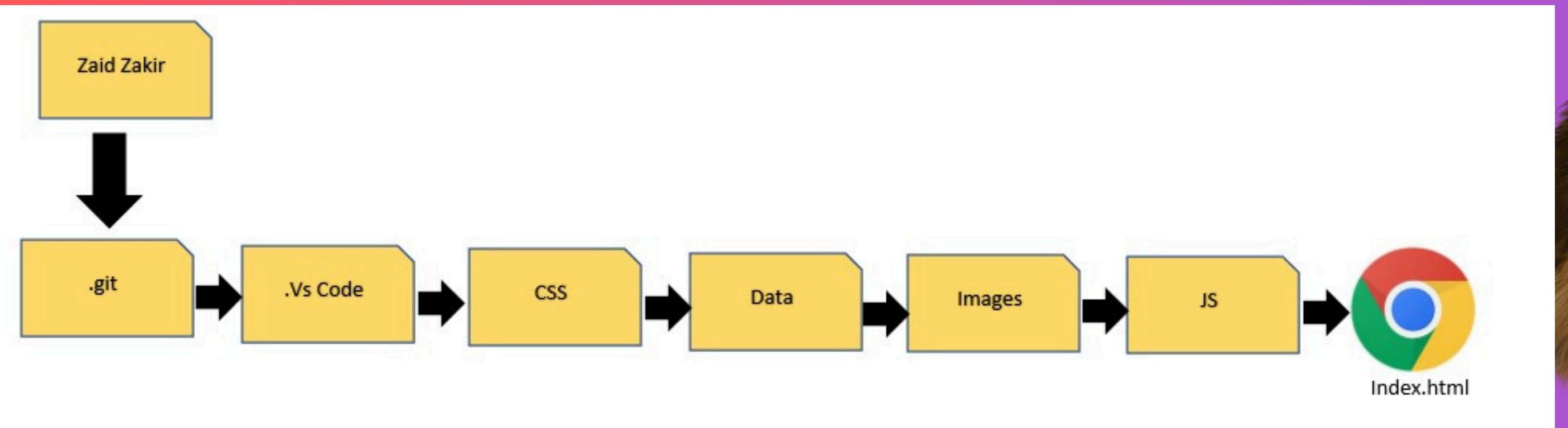
Or, If you Open this Link, my website will open

https://cherry7355608.github.io/petcare/index.html?utm_source=chatgpt.com



4. FOLDER STRUCTURE

The project has a simple folder and file structure.





5. INSTALLATION INSTRUCTIONS

To run the project locally, follow these steps:

1. Download or clone the project code from GitHub.
2. Open the project in VS Code.
3. Run the command `npm install` to install dependencies.
4. Run the command `npm start` to start the development server.
5. Open the link <http://localhost:3000> in the browser to view the website.





6. CONFIGURATION DETAILS

By default, the project runs on port 3000. If you want to change the port, it can be done in the React configuration file. If JSON files are used for data, the file path must be correct in the code. Developers should check these paths when moving the project to another system.



7. DATABASE / DATA HANDLING

Currently, the website uses a JSON file to store and fetch data like ambulance details or pet info. In the future, a real database (like MySQL or MongoDB) can be added. The schema may include fields like ID, Name, Type, Price, and Availability. Data is fetched in React using simple fetch functions.



8. CODING STANDARDS & NAMING CONVENTIONS

TO KEEP THE CODE CLEAN AND EASY:

Currently, the website uses a JSON file to store and fetch data like ambulance details or pet info. In the future, a real database (like MySQL or MongoDB) can be added. The schema may include fields like ID, Name, Type, Price, and Availability. Data is fetched in React using simple fetch functions.





9. TESTING INSTRUCTIONS

- Search → Type an area name and check if correct results are shown.
- Filter → Apply filters (like type or price) and check if the list updates correctly.
- Feedback Form → Fill the form with name, email, and message, then submit. Check if it is stored or shown in the system.
- Navigation → Move from one page to another using the menu and confirm all links are working.
- Responsive Design → Open the website on mobile and computer to see if layout works on both.



10. DEPLOYMENT GUIDE

- Build the project using npm run build.
- Upload the build folder to the hosting service.
- If using GitHub Pages, push the code to GitHub and enable GitHub Pages in repository settings.
- If using Netlify or Vercel, drag the project folder or connect GitHub repo, and it will deploy automatically.
- After deployment, a live link will be available to share.

11. TROUBLESHOOTING GUIDE

Some common problems and solutions are:

- File not found → Check file path and spelling, React is case-sensitive.
- Port already in use → Change port number in configuration or stop the other service.
- Blank page → Make sure data file (JSON) is in correct location and code is fetching it properly.

12. VERSION CONTROL

- Always pull the latest code before starting new work.
- Use branches for new features (e.g., feature/search-bar).
- After testing, merge the branch into the main branch.
- Write clear commit messages (e.g., “Added search filter option”).
- This keeps the project organized and avoids conflicts.



13. SECURITY NOTES

To keep code and data safe:

- Do not use hardcoded passwords in code.
- Validate all user inputs to avoid fake or harmful data.
- Keep dependencies updated to avoid security bugs.
- Hide private files like .env and database passwords from public repos.



14. FUTURE ENHANCEMENTS

In the future, developers can add more advanced features such as:

- GPS tracking of ambulances for live location.
- Online payment system for booking.
- User accounts to save history and reminders.
- Push notifications for updates.
- Multi-language support for wider audience.



15. CONTRIBUTORS

This project was created with teamwork. Each member had a

-  Frontend Developer →
- Backend Developer →
- Tester →
- Documentation Writer →





**THANK
YOU**

