Robel Abebe  
Project Title: Running the Personal Portfolio Website Locally Documentation  
Dec 18/2023

**Project Summery**

The documentation provides clear and concise instructions on how to run the Personal Portfolio Website project locally. The project is a simple, yet responsive personal portfolio website consisting of three pages: Home, About, and Contact. The frontend is built using HTML, CSS, and JavaScript, with a navigation bar implemented for easy page navigation. Media queries are used to ensure the website is mobile-friendly. The backend is based on Node.js technology, incorporating an API endpoint to handle form submissions from the Contact page. Form submissions are stored in a MongoDB database, with server-side validation implemented for form inputs and the website's code is managed using Git version control. Finally, the website is hosted on the free hosting service Scalingo.com.

Table of Contents:

1. Introduction

* Purpose of Documentation
* Target Audience

1. Prerequisites

* Required Software and Tools

1. Installation Steps  
   3.1 Downloading the Project Files  
   3.2 Frontend Setup  
   3.3 Backend Setup  
   3.4 Database Setup  
   3.5 Local Testing
2. Conclusion

* Troubleshooting
* Support

1. **Introduction**:  
   Welcome to the comprehensive documentation for running the Personal Portfolio Website project locally. This documentation provides a step-by-step detailed instructions on setting up and running a simple, yet responsive personal portfolio website with three main pages: Home, About, and Contact.

The frontend of the website is built using HTML to structure the content, CSS to style the elements, and JavaScript to add interactivity. The website design is carefully crafted to be visually appealing and user-friendly. The navigation bar allows users to seamlessly navigate between the different pages, ensuring a smooth browsing experience.

To cater to the growing number of users accessing websites through mobile devices, media queries are implemented. These media queries enable the website to adapt and provide an optimal viewing experience on various screen sizes, regardless of whether the user is accessing it on a desktop, tablet, or smartphone.

For the backend, the project utilizes the powerful Node.js technology. Node.js allows you to run JavaScript code on the server-side, providing a robust and efficient server environment. A simple backend server is set up to handle HTTP requests and serve the required files to the frontend. This integration ensures smooth communication between the frontend and backend, facilitating dynamic and responsive website behavior.

One of the essential features of the Contact page is the form submission functionality. By implementing an API endpoint, the project enables users to submit their contact information via the form. The backend server handles the form submission, applies basic server-side validation to ensure data integrity and security, and then stores the contact information in a lightweight database called MongoDB. This database solution allows for efficient data storage and retrieval, ensuring the seamless management of submitted form entries.

To make the website accessible to a wider audience, the deployment process includes hosting the website on a free hosting service called Scalingo.com. Scalingo.com simplifies the deployment process, allowing you to quickly and easily make your personal portfolio website accessible online.

By following the detailed instructions in this documentation, you will be able to set up and run the Personal Portfolio Website project locally. Try showcasing your skills and accomplishments with a responsive, functional, and visually appealing personal portfolio website!

* **Purpose of Documentation**:  
  The purpose of this documentation is to provide a clear and comprehensive guide on how to run the Personal Portfolio Website project locally. It aims to assist individuals involved in the project by offering step-by-step instructions and troubleshooting tips to ensure a smooth setup and testing process.

The documentation serves the following purposes:

1.1. Instructional Guide: It acts as an instructional guide, providing detailed steps and explanations for each stage of the setup process. It helps users understand the necessary prerequisites, installation steps, and configuration requirements.

1.2. Standardization: By documenting the recommended procedures and guidelines, this documentation helps standardize the setup process and ensures consistency and minimizes potential confusion or errors during the setup.

1.3. Troubleshooting Resource: The documentation includes a troubleshooting section, which addresses common issues that may arise during the setup or testing phases. It offers tips and guidance to help users identify and resolve these issues effectively.

1.4. Support and Communication: The documentation provides contact information for personalized support, allowing users to reach out for specific assistance tailored to their needs. This direct support further enhances the user experience and facilitates effective communication between users and the project's support team.

1.5. Knowledge Transfer: The documentation serves as a means of knowledge transfer, enabling users to understand the project's structure, dependencies, and setup process. It helps users develop a deeper understanding of the project and its underlying technologies.

By fulfilling these purposes, this documentation aims to empower individuals involved in the Personal Portfolio Website project, enabling them to set up and test the website with confidence and ease.

* **Target Audience:**  
  The target audience for this documentation includes individuals who are interested in running the Personal Portfolio Website project locally which includes researchers, developers, innovators, content creators and others.

1. **Prerequisites:**  
   Before proceeding with running the project locally, ensure that you have the following prerequisites installed on your machine:

**Node.js**: Download and install from the official website ([https://nodejs.org](https://nodejs.org/)). Verify the installation by running node -v in your terminal or command prompt.

**MongoDB**: Install MongoDB from the official website (<https://www.mongodb.com/try/download/community>). Ensure the MongoDB server is running.

* **Required Software and Tools:**  
  To effectively run and utilize the system, you will need the following software and tools:

3.1. Web Browser:  
You will need a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge to access the system's interface and interact with its features.

3.2. Internet Connection:  
A stable internet connection is crucial as the system operates online and requires connectivity to access its powerful language processing capabilities.

3.3. Code Editor (Optional):  
If you plan to integrate the system into your own applications or develop custom solutions, a code editor is recommended for writing and editing code. Popular code editors include Visual Studio Code, Sublime Text, Atom, or Notepad++.

1. **Installation Steps**
   1. **Downloading the Project Files:**

Download the project files from the designated source. You can either download a zip file or clone the project repository using Git. Extract the downloaded zip file to a location of your choice.

* 1. **Frontend Setup:**
* Open the project folder and navigate to the "frontend" directory. This directory contains all the files that make up the frontend of the website.
* Open the "index.html" file in a code editor of your choice.
* Customize the content and design of the Home, About, and Contact pages according to your specific requirements. You can modify the HTML structure, apply CSS styles, and add JavaScript functionality based on your preferences and personal information.
  1. **Backend Setup:**
* Open a terminal or command prompt and navigate to the project folder, specifically the "backend" directory.
* Install the required dependencies by running the command npm install. This command will read the project's package.json file and download and install the necessary packages.
* Once the installation is complete, you are ready to set up the backend server.
  1. **Database Setup:**
* Open a terminal or command prompt and navigate to the project folder, specifically the "backend" directory.
* Ensure the MongoDB server is running.
* Run the following command to set up the database: mongoimport --db portfolio --collection submissions --file submissions.json. This assumes you have a JSON file (submissions.json) with the necessary data for initial setup.
  1. **Local Testing:**
* Start the backend server by running the command node server.js in the terminal or command prompt. This will start the server and make the website accessible at http://localhost:3000/.
* Open a web browser and enter the above URL in the address bar. The website should now load, and you can navigate through the pages using the provided navigation bar.
* To test the form submission functionality, visit the Contact page, fill out the required details, and submit the form. The server will handle the form submission, perform basic server-side validation, and store the form data in the MongoDB database.

You can verify the successful submission by checking the database or by inspecting the server logs displayed in the terminal or command prompt.

1. **Conclusion:**Now you have successfully set up and tested the Personal Portfolio Website project locally using MongoDB. You can now further customize the website by adding more sections, refining the design, or incorporating additional features as per your preferences.

* **Troubleshooting:**  
  If you encounter any issues during the setup or testing process, refer to the following troubleshooting tips:
* Ensure Prerequisites: Double-check that you have correctly installed Node.js and verified its version using the command node -v.
* Verify Dependencies: Make sure all the project dependencies are installed correctly by running the command npm install in the backend directory.
* Check Database Connection: If the database setup is not working properly, verify that you have executed the node db-setup.js command in the backend directory to create the necessary database and table structure. Ensure that the database file location and connection details in the configuration are accurate.
* Server Startup: If the backend server fails to start, check for any error messages displayed in the terminal or command prompt. Look for potential errors in the server.js file or any other relevant backend files.
* Form Submission: If the form submissions are not being stored correctly, ensure that the form data is properly sent to the backend API endpoint and that the server-side validation is functioning correctly.
* **Support:**  
  If you require further assistance or encounter persistent issues, please feel free to reach out for personalized support. You can contact:

Robel Abebe

Address: Addis Abeba, Ehiopia

Phone: (+251)911 946041

Email: [robel.sean@gmail.com](mailto:robel.sean@gmail.com)

Please provide a detailed explanation of the issue you are facing along with any relevant error messages or logs. You can expect a response within short period of time. We are committed to helping you resolve any difficulties you may be experiencing with the Personal Portfolio Website project.

Please note that this concludes the documentation. If you have any further questions or concerns, feel free to reach out using the provided contact details.

Thank you!