# Documentation + Guidelines:

This document provides detailed instructions and guidelines for maintaining and managing content on the Reds Hair Studio Website, which is built purely with HTML, CSS, and JavaScript without a Content Management System

Overview:

The website files are organised in a directory structure that includes folders for HTML, CSS, JavaScript, Images and other assets.

Directory Structure:

Root Folder (PLACEHOLDER) – Contains the HTML files + Asset Folder

‘/Docs’ – Contains any documentation or additional files

‘/Assets’ – Contains all subdirectories (CSS Files, Images, JavaScript Files, Jsonfiles)

‘/Assets/Images’ – Contains all the images used in the website (PNG, JPEG)

‘/Assets/CSS Files’ – Contains all the CSS Files

‘/Assets/JavaScript Files’ – Contains all the JavaScript Files

## Setting Up the Website Locally:

Prerequisites:

A text editor (Sublime Text, Notepad ++, Visual Studio Code)

A web browser for testing (Google Chrome, Firefox, etc.)

An FTP/SFTP Client (e.g., FileZilla) if you will be uploading to a web server

To view locally do the following:

Open the main index.html (the main html file) – should open directly to your default web browser

Use the text editor to open and edit the HTML, CSS, and JavaScript files

## Uploading to a Web server:

Using FTP/SFTP:

Use an FTP/SFTP client to connect to your web server.

Enter the server address, username, and password provided by your hosting provider.

Upload the entire website directory to the server’s public HTML folder (often named /public\_html or /www).

Testing the Upload:

After uploading, navigate to your domain in a web browser to ensure the website is displaying correctly.

Check all pages and links to ensure they are functioning properly.

Updating Content:

Editing HTML Files:

Locate the relevant HTML file (e.g., index.html, services.html, location.html) in your text editor.

Modify the content within the appropriate HTML tags.

Save changes and re-upload the file to the server if hosted online.

Updating Text Content:

Navigate to the specific section of the HTML file where the text is located.

Edit the text between the HTML tags.

Ensure proper formatting and HTML structure.

Updating Images:

Replace existing images in the /images directory.

Update the <img> tags in the HTML files to reference new images if filenames change.

Ensure new images are optimized for web use to maintain fast loading times.

6. Adding New Pages

Creating a New HTML File:

Create a new HTML file in your text editor.

Use the existing HTML files as a template for consistent layout and styling.

Save the new file with an appropriate name (e.g., newpage.html).

Linking to the New Page:

Update the navigation bar in header.html or the relevant section of each page to include a link to the new page.

Ensure the link uses the correct relative path.

Updating Styles:

Editing CSS Files:

Locate the CSS file (e.g., styles.css) in the /CSS Files directory.

Open the file in a text editor.

Modify the CSS rules to change styles such as colours, fonts, and layout.

Save changes and re-upload the file to the server if hosted online.

Adding New Styles:

Add new CSS rules to the styles.css file as needed.

Ensure new styles do not conflict with existing ones.

Managing JavaScript:

Editing JavaScript Files:

Locate the JavaScript file (e.g., scripts.js) in the /JavaScriptFiles directory.

Open the file in a text editor.

Modify or add JavaScript functions as needed.

Save changes and re-upload the file to the server if hosted online.

Testing Changes:

Test all changes in multiple browsers to ensure compatibility.

Use browser developer tools to debug and fix any issues.

9. Maintaining the Contact Form

Editing the Form:

Locate the HTML file containing the contact form.

Modify the form fields as needed.

Form Submission Handling:

Ensure the form’s action attribute points to the correct server-side script (e.g., contact\_form\_handler.php).

Update the server-side script to handle new form fields if changes are made.

10. Keeping the Website Updated

Content Updates:

Regularly review and update website content to keep it current.

Ensure all text, images, and links are up-to-date.

Security and Backups:

Implement regular backups of the website files.

Store backups in a secure location.

Regularly update and review server security settings.

Code Review and Optimization:

Periodically review HTML, CSS, and JavaScript code for optimization.

Minify CSS and JavaScript files to improve load times.

Use tools like W3C Validator to check HTML code for errors.