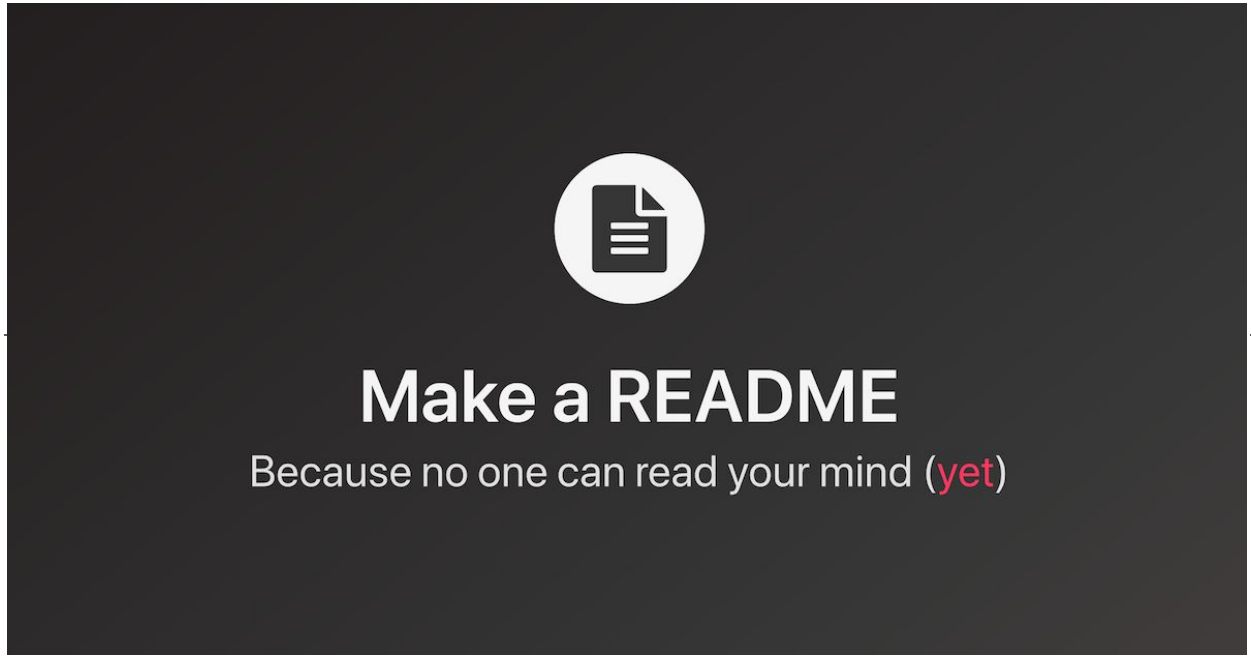


README

README.md

WHY make it?

- A README is often the first item a visitor will see when visiting your repository. README files typically include information on: What the project does. Why the project is useful.
- Credit your sources
- Explain the design and development process
- Demonstrate your testing
- Show your thought process

What is it?

- In simple words, we can describe a README file as a guide that gives users a detailed description of a project you have worked on.
- Displayed when your github repo is viewed
- A crucial part of the development process

How is it made?

- Markdown language
- .md file extension
- Created and lives in your project folder

What is required in it?

- Project overview
- UX with user stories
- Manual testing
- Deployment
- Crediting sources
- Knows bugs
- Future features

code institute

README Sections

README.md

Love Running

In this section, you will include one or two paragraphs providing an overview of your project. Essentially, this part is your sales pitch. At this stage, you should have a name for your project so use it! Don't introduce the project as a Portfolio project for the diploma. In this section, describe what the project hopes to accomplish, who it is intended to target and how it will be useful to the target audience.

For example; Love Running is a site that hopes to help keep people motivated to meet up for runs on a regular basis in Dublin, Ireland. The site will be targeted toward runners who are looking for a way to socialise and keep themselves fit. Love Running will be useful for runners to see exactly when and where they should be to join the running club.

Features

In this section, you should go over the different parts of your project, and describe each in a sentence or so. You will need to explain what value each of the features provides for the user, focusing on who this website is for, what it is that they want to achieve and how your project is the best way to help them achieve these things.

Important readme sections

- Project overview
- List of features
- UX/UI
- Testing
- Deployment
- Citation of **ALL** sources(code, images, text)
- Future features
- Known Bugs

README.md

Deployment

GitHub Pages

I used the following steps to deploy the project.

1. Log in to Github and navigate to the repository.
2. On Github repository page, click on settings tab.
3. Scroll down to Pages tab on the menu on the left hand side.
4. Under Source, click the Branch drop down menu and select main.
5. Click Save.
6. The page will update and show the text "Your site is published at https://sblakedev.github.io/Project-2/"

Forking the Repository

To fork the repository in order to make a copy so as to view or make changes to the site without affecting the original repository, the following steps are to be used.

1. Log in to Github and navigate to the repository.
2. On the right hand side, above the setting tab, click the 'Fork' button.
3. There will now be a copy of the original respository in your GitHub account.

Testing

In this section, you need to convince the assessor that you have conducted enough testing to legitimately believe that the site works well. Essentially, in this part you will want to go over all of your project's features and ensure that they all work as intended, with the project providing an easy and straightforward way for the users to achieve their goals.

In addition, you should mention in this section how your project looks and works on different browsers and screen sizes.

You should also mention in this section any interesting bugs or problems you discovered during your testing, even if you haven't addressed them yet.

If this section grows too long, you may want to split it off into a separate file and link to it from here.

Validator Testing

- HTML
  - No errors were returned when passing through the official W3C validator
- CSS
  - No errors were found when passing through the official (Jigsaw) validator

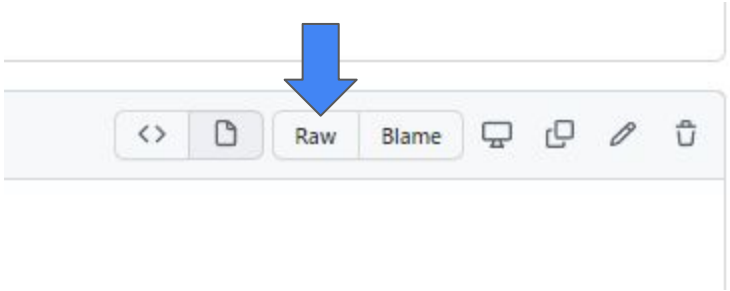
Unfixed Bugs

You will need to mention unfixed bugs and why they were not fixed. This section should include shortcomings of the frameworks or technologies used. Although time can be a big variable to consider, paucity of time and difficulty understanding implementation is not a valid reason to leave bugs unfixed.

code institute

README CI TEMPLATE

- Use CI README Template
    - Click on **Raw** to reveal the code
- Paste code to your Workspace and fill in with project specific information

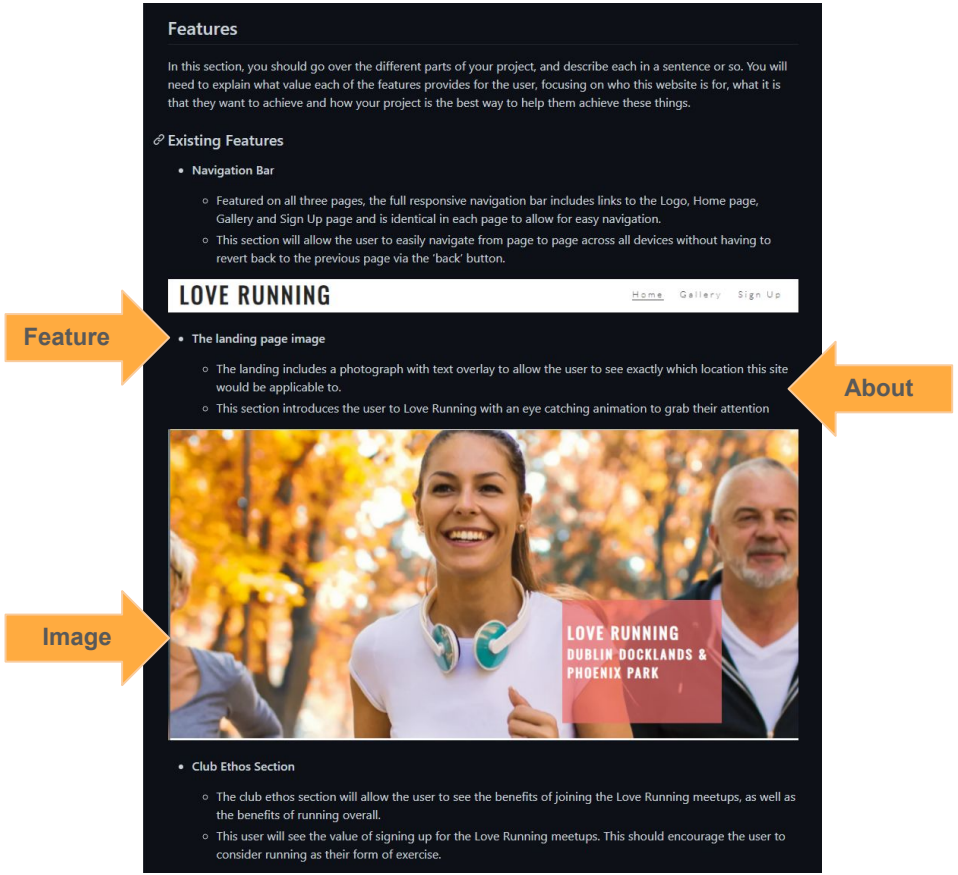


code institute

Structure of features section

- Feature name
- A description about the feature and how it works
- Screenshot image of the feature

List of Features



code institute

UX/UI Section

This section is for showing your

- site goals
- design thought process
- planning and wireframe design

The sections you should have here

- Site Goals
- Design choices
- User stories
- Wireframes
- Anything else you want to add that relates to UX/UI

UX

The site is presented in a minimalist but effective way, providing all the information the user needs. In general, users can find out more about Vanna's Beauty Salon, his team, services and how to contact them, there is also a gallery wher users can find inspiration in nail design or view works before and after appropiate treatments.

Users Stories

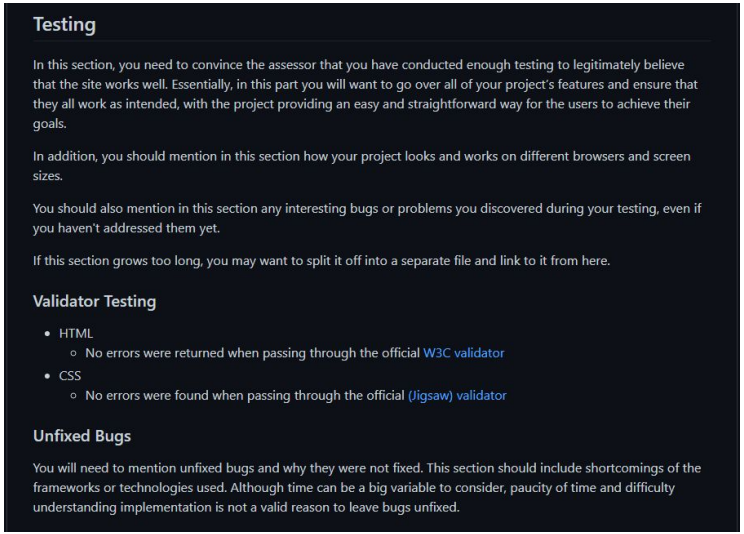
- As a new user of this website, I want to easily find out the purpose of the site
- As a new user of this website, I want to quickly and easy find the services you provide
- As a new user of this website, I want to quickly and easyily find the price list of your services
- As a new user of this website, I want to see the experience of other client
- As a new user of this website, I want to see pictures of your work
- As a new user of this website, I want to easily found a way to contact you
- As a new user to this website, I want to meet the team
- As a user, I want to follow the salon on social media so I can keep up to date with the latest information

code institute

Writing the Testing Section

The testing section should be used to

- show the assessor you have tested your site well
- show that your site is responsive on multiple devices
- test user stories
- show that your code passes validation
- list any unfixed bugs in your code



Testing

In this section, you need to convince the assessor that you have conducted enough testing to legitimately believe that the site works well. Essentially, in this part you will want to go over all of your project's features and ensure that they all work as intended, with the project providing an easy and straightforward way for the users to achieve their goals.

In addition, you should mention in this section how your project looks and works on different browsers and screen sizes.

You should also mention in this section any interesting bugs or problems you discovered during your testing, even if you haven't addressed them yet.

If this section grows too long, you may want to split it off into a separate file and link to it from here.

Validator Testing

- HTML
  - No errors were returned when passing through the official W3C validator
- CSS
  - No errors were found when passing through the official (Jigsaw) validator

Unfixed Bugs

You will need to mention unfixed bugs and why they were not fixed. This section should include shortcomings of the frameworks or technologies used. Although time can be a big variable to consider, paucity of time and difficulty understanding implementation is not a valid reason to leave bugs unfixed.

code institute

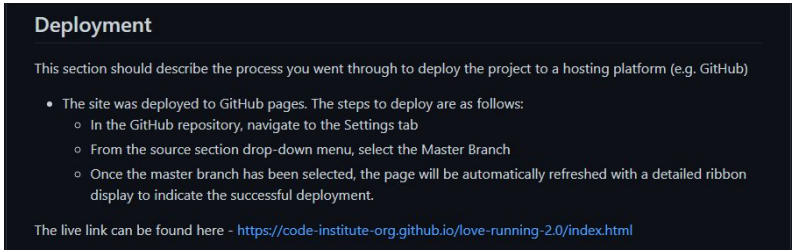
Deployment

Why have the deployment section

- Show the assessor you know how to deploy your project
- Show future collaborators and clients how to use your software
- Have a record for your future self as a reminder of how to deploy on the platform

What to have in the deployment section

- Detailed list of steps taken to deploy
- Deploying to Github Pages
- Cloning/forking the Github repo



Deployment

This section should describe the process you went through to deploy the project to a hosting platform (e.g. GitHub)

- The site was deployed to GitHub pages. The steps to deploy are as follows:
  - In the GitHub repository, navigate to the Settings tab
  - From the source section drop-down menu, select the Master Branch
  - Once the master branch has been selected, the page will be automatically refreshed with a detailed ribbon display to indicate the successful deployment.

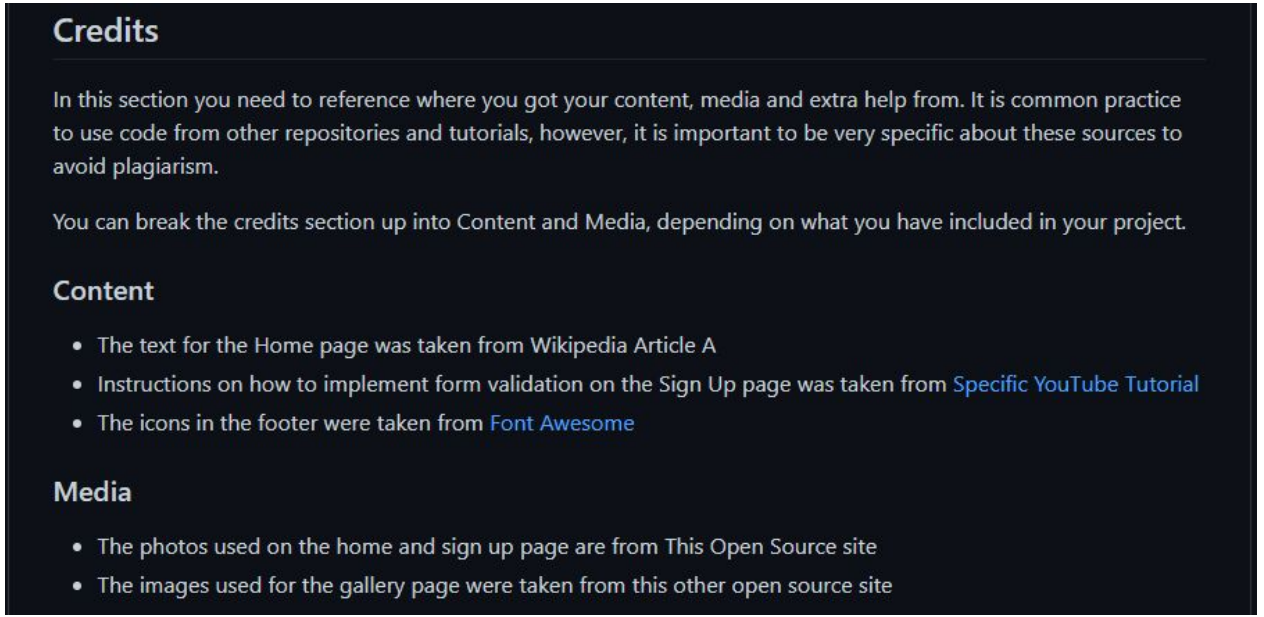
The live link can be found here - https://code-institute-org.github.io/love-running-2.0/index.html

code institute

Citation of Sources

Crediting Sources

- Always credit ALL sources
- Images
- Code
- Text
- Link to relevant site
- Authors name/pseudonym
- Link to actual content



Credits

In this section you need to reference where you got your content, media and extra help from. It is common practice to use code from other repositories and tutorials, however, it is important to be very specific about these sources to avoid plagiarism.

You can break the credits section up into Content and Media, depending on what you have included in your project.

Content

- The text for the Home page was taken from Wikipedia Article A
- Instructions on how to implement form validation on the Sign Up page was taken from Specific YouTube Tutorial
- The icons in the footer were taken from Font Awesome

Media

- The photos used on the home and sign up page are from This Open Source site
- The images used for the gallery page were taken from this other open source site

code institute

Future Features

Features you did not add yet

- You ran out of time
- You weren't sure if it was viable
- Weren't comfortable implementing

Features Left to Implement

- User profile picture which displays in the navbar when logged in and on user reviews.
- Model and page for animal care products.
- Option to select home delivery or collect in store.
- Custom error screens such as 500, 404 etc. Django defaults are currently being used.
- Remove unused fields in models, Ive had many issues with migration so I didn't want to mess with that so close to submission as it works currently and don't have time to fix if an issue crops up.

code institute

Summary

As you can see the README is a crucial part of your projects and writing a well-documented README is a skill. By the time you graduate you will be comfortable with markdown and writing good software documentation.

The README should be started with your project and never as an afterthought. This way you can add any bugs that crop up and ideas you may have a long the way.

The README is your direct line of communication to the Assessor within this course and to a potential client/collaborator in your career as a developer. This is a powerful tool for communicating your ideas and process and should be treated with the utmost importance.

Project examples

Thrive	Repository	Deployed
Batala Bangor Website	Repository	Deployed
Bodelschwinger Hof	Repository	Deployed

code institute