SIT209 Software Engineering 2: Developing IoT Applications

Pass Task 1.1: Introduce Yourself

Overview of the task

In this task you will develop a basic webpage to introduce yourself and provide a short description of which units you have already completed and what you would like to achieve in the unit.

This task will cover the following:

- Basic use of command line
- Git for basic version control
- GitHub for hosting remote repositories
- HTML to markup a webpage
- CSS to style a webpage

Submission Details

Submit a pdf document to OnTrack with captioned screenshots of the following.

1. Trello board in inital state - all cards in Todo column
2. Trello board in progress state - cards in different columns
3. Trello board in finished state - all cards in Done column
4. GitHub commit graphs
5. Styled webpage with a short introduction about yourself

Instructions

1. Create a trello board to manage your work and add four columns - Todo, Doing, On Hold and Done.

2. Add the following list as separate cards in the Todo column.

    - Take screenshot of initial Trello board
    - Setup git and GitHub for project
    - Create a webpage with a short introduction about myself
    - Take a screenshot of commit graphs in GitHub
    - Take a screenshot of Trello board in progress

3. Take a screenshot of your Trello board at this stage, showing what you need to do to complete the remainder of this task.

4. Move the `Take a screenshot of initial Trello board` card from the Todo to Done column.

    Yay! You completed your first Trello card! **Congratulations!**

5. Move the `Setup git and GitHub for project` card to the Doing column.

This will be the last instruction about moving cards on your Trello board.

You will need to manually move your cards as you progress through the project and remember to take a progress screenshot before you complete all the cards.

Using Trello in this way helps to track your progress across a project and keep you focused on completing one task before moving on to the next one.

It is also great for communication when working in groups. Any member of the team can quickly get an overview of the whole team's progress and what each member is currently working on.

6. Install Git on your system. The exact instructions depend on your Operating System. Some good instructions are here Install Git.

7. Watch the Git tutorial video Github to refresh your version control knowledge.

8. Open up the Terminal app (macOS) or Command Prompt (Windows), change the directory to a sensible location to store your code - the example uses the home directory - and create new directories for the TrackMe case study.

```
cd [Folder your going to use here] (e.g. ~)
mkdir code
cd code
mkdir SIT209
cd SIT209
mkdir TrackMe
cd TrackMe
```

9. Initialise this directory as a local git repository.

```
git init
```

This will create a new .git folder which is used for configuration, as well as tracking changes across the repository. You shouldn't ever need to change anything manually in this folder, but it might be good to take a look through the subdirectories and get an idea of how it works.

Don't worry if this seems daunting, you will not be required to change or understand this directory for this unit.

10. Check the git status of the directory

```
git status
```

This should show that there is nothing to commit, which makes sense because we haven't actually changed anything yet.

11. Open whichever text editor you would like to use for the unit and create a file `about-me.html` file. We recommend VS Code.

12. Copy the following template into the `about-me.html` file, update the `author` in the `head` section with your name, as well as the content for the `h1` element in the `body` section and save the file.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="A frontend dashboard for the
TrackMe application" />
    <meta name="author" content="ENTER YOUR NAME HERE" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>TrackMe</title>
  </head>
  <body>
    <h1>ADD YOUR NAME HERE</h1>
  </body>
</html>
```

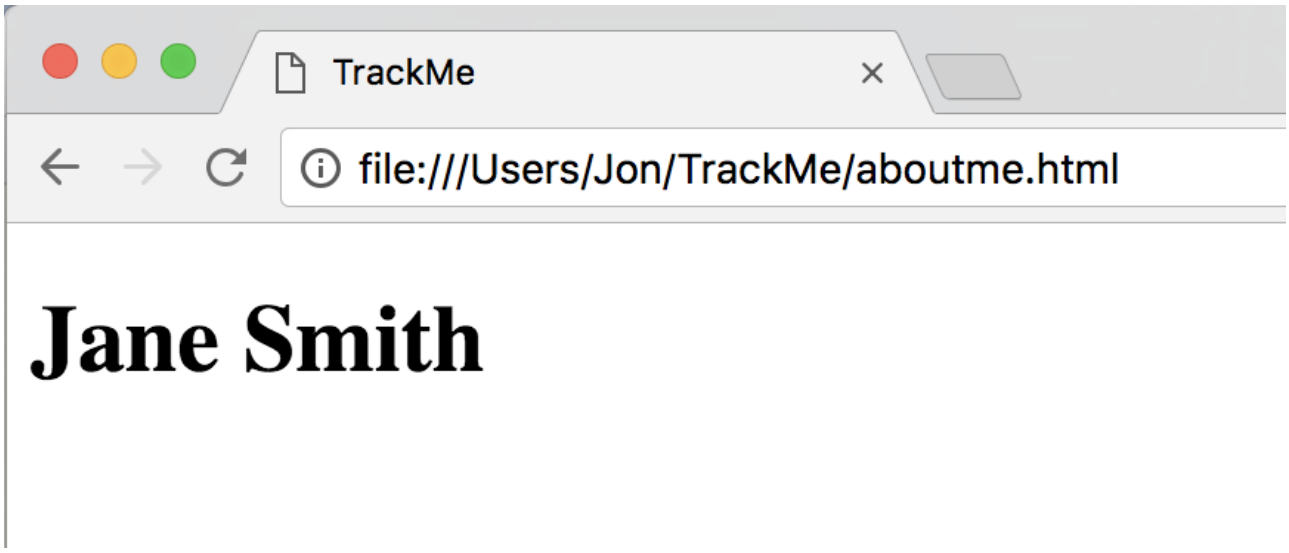
    The `head` section contains information for the browser and is not displayed on the webpage.

    The `body` section is where you create HTML elements to markup the content that you want to display on the webpage.

    At this point, you should start being careful about making sure quotes and various brackets are copied correctly. You should always check you fully understand code, and don't just copy it. Most errors in this unit are caused by wrongly copied code.

13. Save a copy of this file as `template.html` so you can use it for all your future TrackMe `html` files.

14. Open the `about-me.html` file in Google Chrome or your favourite browser - Chrome is recommended due to its development tools, which will help throughout the development of our frontend application.



The browser should display your new webpage with your name as a heading.

15. Open the Chrome developer tools. View->Developer->Developer Tools. Select the Console Tab. Have a look for any error messages. It is likely that the `initial-scale` tag hasn't copied correctly, fix it if it hasn't. If you have no error messages, then great!

16. Use git to check the status of your repo.

```
git status
```

You should see that about-me.html has untracked changes.

17. Use git to add your changes to the staging area and check the status.

```
git add about-me.html
git status
```

It should now show that the file has changes that are ready to commit.

18. Use git to commit you changes.

```
git commit -m "NEW: Add webpage for bio"
```

Adding the -m argument and a string message allows you to document what has changed. This should be a short, concise and clear message that describes what this commit will change about the repository, rather than a past-tense message about what has changed.

`ENHANCE: Add user information to dashboard` <== Correct

`ENHANCE: Added user information to dashboard` <== Incorrect

The message should be prepended with one of the following categories in all caps to communicate what kind of change it is:

- NEW: A new feature
- ENHANCE: An improved or extended functionality for an existing feature
- FIX: Fixed a bug or issue
- QUALITY: Improved the quality of the code. No change to functionality
- LOOKS: Updated the layout or look of the application. No change to functionality
- DOC: Added or updated documentation

19. Use git to check the status of your repo.

```
git status
```

Git should show that all changes have been committed.

20. Add a short summary of your hobbies, skills and the grade you will be aiming to achieve for the unit to the `about-me.html` file.

    Add this within a HTML element as a child of the `body` section.

    Think about which HTML element would be most appropriate for this.

    Now would be a good time to have a look at the W3C HTML tutorial which is a great reference for HTML. W3C HTML Tutorial

21. Add a list of units you have completed so far.

    Think about which HTML elements could work for this and whether there is one that would be the most appropriate.

22. Use git to add and commit these changes, choosing an appropriate prepended category and commit message.

23. Navigate to the GitHub website and log in - create an account if you do not have one.

24. Create a new GitHub repository named "TrackMe".

25. Add the GitHub repository as a remote so changes can be pushed to a central location. Your remote repository URL is displayed in the Quick Setup panel on GitHub or can be found under the `Clone or download` dropdown menu on the landing page for your TrackMe repo.

```
git remote add origin https://github.com/your-username/trackme.git
```

26. Configure git's username and email - this step is only needed if you have not setup git previously

```
git config --global user.name "<your username>"
git config --global user.email <your email address>
```

27. Use git to push your commits to the GitHub repository.

```
git push origin master
```

28. Create a new `css` file called `style.css` to add styling rules for your webpage.

29. Include the new `css` file as a stylesheet in `about-me.html` using the `link` HTML element. Add the following line to the `head` section of `about.me.html`.

```
<link rel="stylesheet" href="style.css" />
```

30. Open `style.css` file in your text editor and add at least three custom styling rules.
31. Commit and push your changes to GitHub.
32. Take a screenshot of your styled webpage running in the browser.
33. Take a screenshot of your GitHub commit graphs under the Insights tab of your TrackMe repository.
34. Take a screenshot of your Done cards on Trello.
35. Compile your screenshots into one word document, add a short caption for each screenshot, export as a PDF and submit to OnTrack.