SIT223-SIT753
Pass Group Task:
Git Version Control (Sprint 1)

DEAKIN UNIVERSITY

Git Version Control (Sprint 1)

This is a Group Pass task.

Overview

Version Control systems allow developers to work together in an efficient manner. Using these systems, developers work on their changes to the software and can then merge in changes made by others. You need to watch all Week 4 mini-lecture videos before starting this task. You also need to use the instructions on the following pages to create a copy of a Deakin Unit Page repository and then make changes **as a part of Sprint 1**. This task should be completed in your Week 4 active class and submitted for feedback after that.

Task Instructions

In this task you will work with 2 other developers of your Scrum team to create the SIT223-SIT753 unit page using HTML and CSS. The code isn't the focus, so you will be given the changes you need to make. You want to focus on the process of using Git to work together as a team.

1. Navigate to the GitHub website.
2. Sign up for an account, or login if you already have one account.
3. Make sure that you have Git setup on your machine. Test your setup. Open a Terminal window and type the following command.
   ```
   git --version
   ```
4. Setup your name by running the following command from the Terminal:
   ```
   git config --global user.name "YOUR NAME"
   ```
5. Use the following command to setup your email:
   ```
   git config --global user.email "YOUR EMAIL ADDRESS"
   ```

These settings will be used when you are interacting with git. You want to make sure that these are correct, and the email matches the email you used when you created your account on GitHub.

You now have git setup on your machine! So now you just need to work with your team.

6. Get into your team with 2 other developers — each team should have 3 members!

Note: If needed one team member can perform multiple roles. Check with your tutor about how to do this.

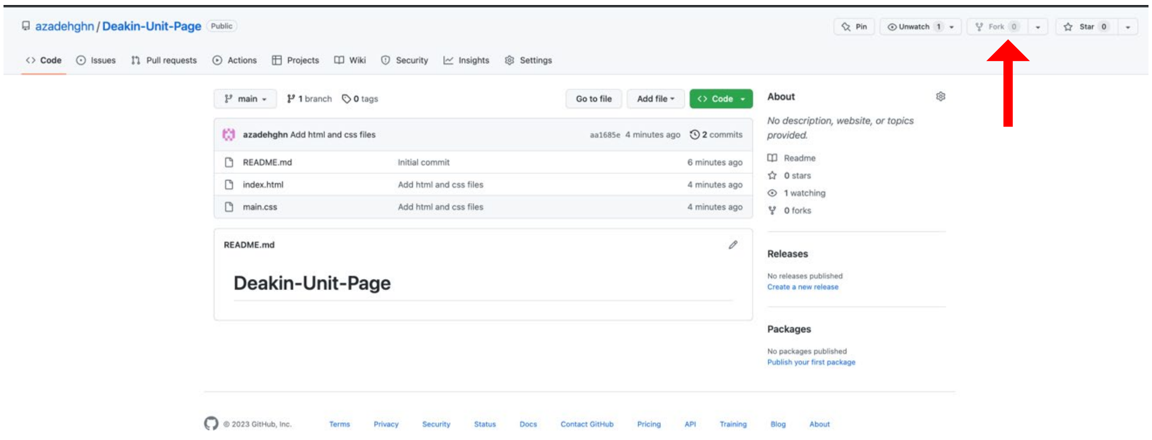
7. Assign roles to each of the three group members in the Scrum Team.
   - **Team Leader** (**TL**) — will setup the project and add fellow collaborators to the team.
   - **UI Designer** (**UI**) — will design the user interface (UI) of a website.
   - **Front-end Developer** (**FD**) —will build the front-end of a website, including the layout, styling, and interactive elements.

Now that you are in your team, you can work through the following process. Identify the steps related to your role in the project. You will need to gather screenshots and take notes at each stage.

As a team get together on the Team Leader's machine. The first task is to get a copy of the project and ensure that you are all setup as collaborators!

8. **TL:** Navigate to https://github.com/azadehghn/Deakin-Unit-Page — this is the start of the project.

9. **TL**: Fork the repository — click the **Fork** button.



azadehghn Add html and css files aa1685e 4 minutes ago 2 commits

README.md	Initial commit	6 minutes ago
index.html	Add html and css files	4 minutes ago
main.css	Add html and css files	4 minutes ago

README.md

Deakin-Unit-Page

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

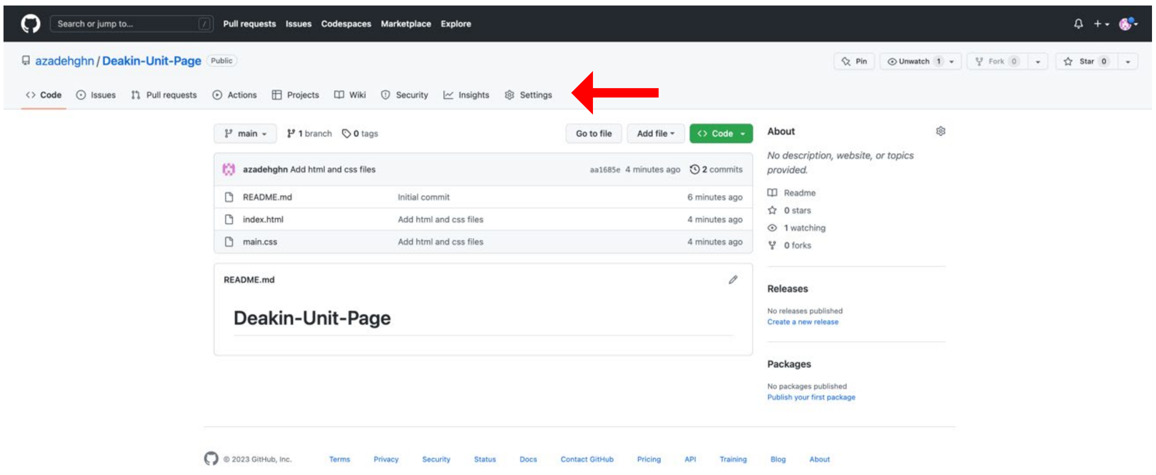
No releases published
Create a new release

No packages published
Publish your first package

Note: Repository is main term used for a project in a version control system. When you fork a repository, you get a copy of that project and its entire history!

10. **TL, UI, FD**: Notice that fork is now a part of the Team Leader's account. The next step will be to add the UI Designer and Front-end Developer accounts as collaborators.

11. **TL**: Click the **Settings** to get to the Repository's settings.



azadehghn Add html and css files aa1685e 4 minutes ago 2 commits

README.md	Initial commit	6 minutes ago
index.html	Add html and css files	4 minutes ago
main.css	Add html and css files	4 minutes ago

README.md

Deakin-Unit-Page

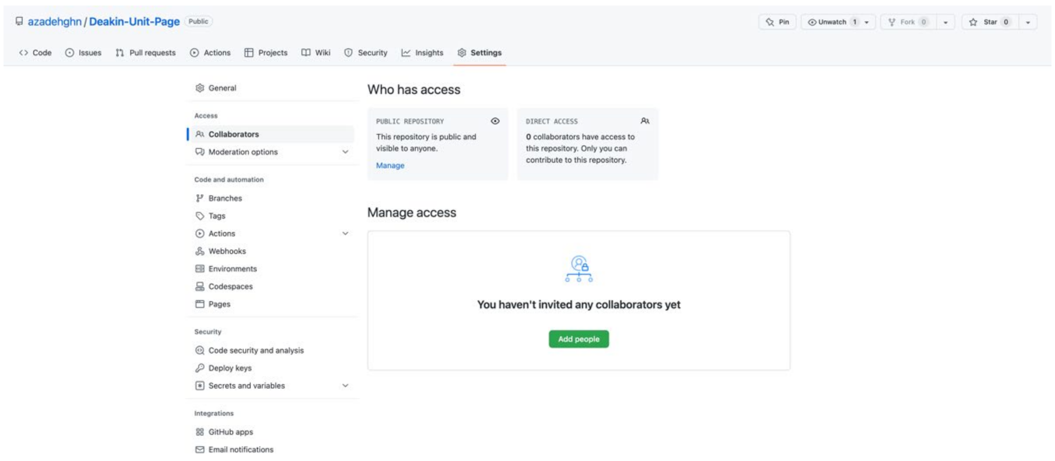
No description, website, or topics provided.

No releases published
Create a new release

No packages published
Publish your first package

© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

12. **TL**: Select the **Collaborators** tab and enter the names of your two team members. Add them both as collaborators.



Who has access

PUBLIC REPOSITORY
This repository is public and visible to anyone.
Manage

DIRECT ACCESS
0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

13. **UI, FD**: Return to your own computer.
14. **UI, FD**: Once the Team Leader has added you to the project, refresh your GitHub page in the browser and make sure you can see the Deakin-Unit-Page repo.
15. Now that you are all in the repository as collaborators, you can start to work together on this code.
16. With git you need to get your own local copy of the repository. You then work on that repository and push your changes back to the server when you have something cool working that you want to share with the other team members or when you want to make sure your work is backed up!
17. Copy the projects **HTTPS clone url** from GitHub.
18. Switch to your Terminal.

19. Navigate to a directory where you want to store the project's code. For example, /C/Users/username/Documents/Code or something similar.

Note: When you get the project it will go into a subdirectory based on the project's name. So don't create a directory for the project itself at this stage.

20. At the terminal type the following command, pasting in the URL (example is shown).

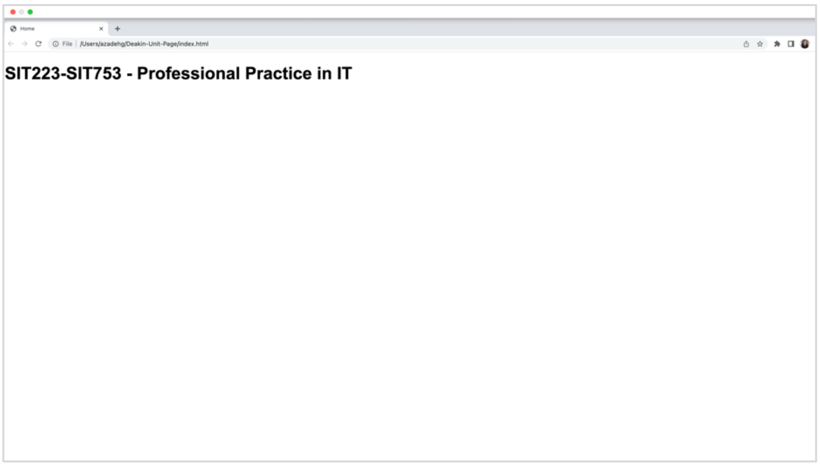
```
git clone https://github.com/azadehghn/Deakin-Unit-Page.git
```

You should now be able to see all the project's files in the Deakin-Unit-Page folder. This is your working copy of the projects.

21. Test that the project works currently. In your terminal window you should be able to compile and run the project with the following steps.

```
cd Deakin-Unit-Page
open index.html
```

22. At this stage you should be able to see the SIT223-SIT753 webpage as shown below.



SIT223-SIT753 - Professional Practice in IT

There are the following issues with the program:

- Things for the **UI Designer** to fix:
  - It would be good to have a header, and style it with a green background color.
  - Deakin logo should be shown on the header.
- Things for the **Front-end Developer** to fix:
  - A table with unit details including unit chair, enrolment mode and year should be added.
  - The table should be styled.
- Things for the **Team Leader** to fix:

- A paragraph to detail unit content needs to be added.
- It would be good to have a link to enrolment and fee page at the end.

You can all work on this now and get git to merge all your changes.

UI Designer Tasks

Get your team together at your computer to observe the process.

23. In the repository you should always be working on a **branch** other than then main — which keeps a "good" state of the project. Let’s add a header first.

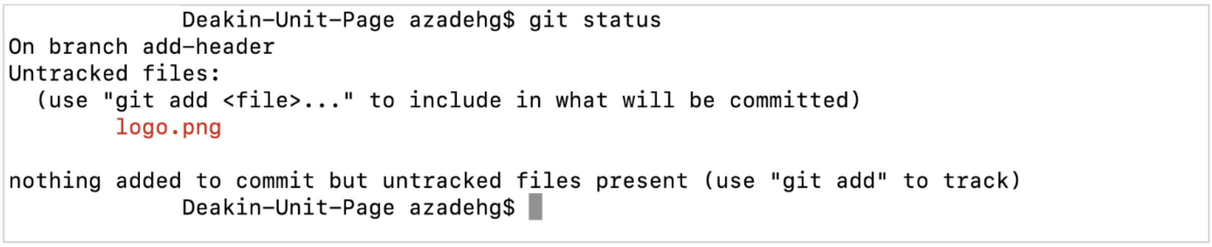
    Start a new branch using the following command:

    ```
    git checkout -b add-header
    ```

24. Get the Deakin logo from here for this task and **save** it in the folder of the project. **The file should be called "logo.svg".**

25. Check the status of the repository, and you should be able to see the new file. The image should be listed as "Untracked" as it is not in the repository.

    ```
    git status
    ```



```
             Deakin-Unit-Page azadehg$ git status
On branch add-header
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logo.png

nothing added to commit but untracked files present (use "git add" to track)
             Deakin-Unit-Page azadehg$ 
```

26. Add the file to the repository using the following command. This will stage this change, ready for you to commit it when you are ready.

    ```
    git add logo.svg
    ```

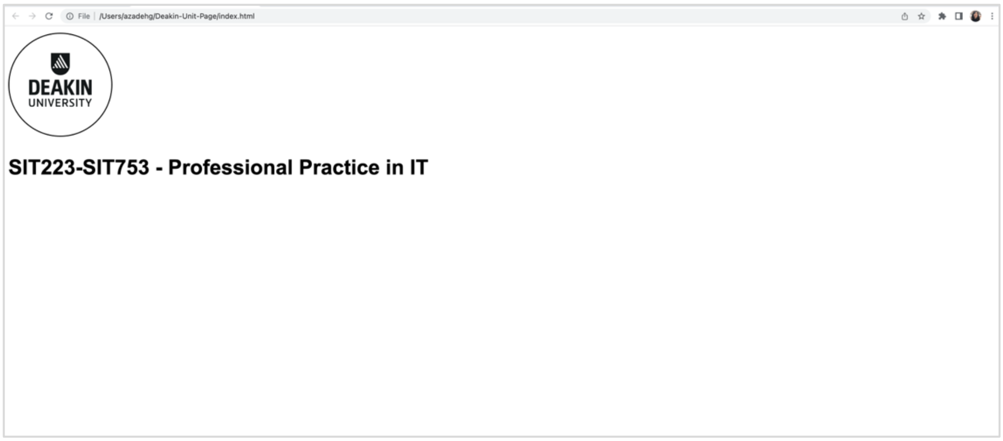
27. Before we commit, let us use this logo in the webpage. Open **index.html** file with VS code or any other text editor and add the following **<div >** in purple as shown below to draw the header and add logo to it.

```
<!DOCTYPE html>
<html>
<head>
<title> Home </title>
<link rel="stylesheet" type="text/css" media="screen" href="main.css"/>
</head>

<div >
   <img src="logo.svg" alt="Deakin" width="200" height="210" >
</div>
```

```
<body>
<h1>SIT223-SIT753 - Professional Practice in IT</h1>
</body>
</html>
```

28. **Save** the index file and **open** the file to see the changes.

```
open index.html
```



File | /Users/azadehg/Deakin-Unit-Page/index.html

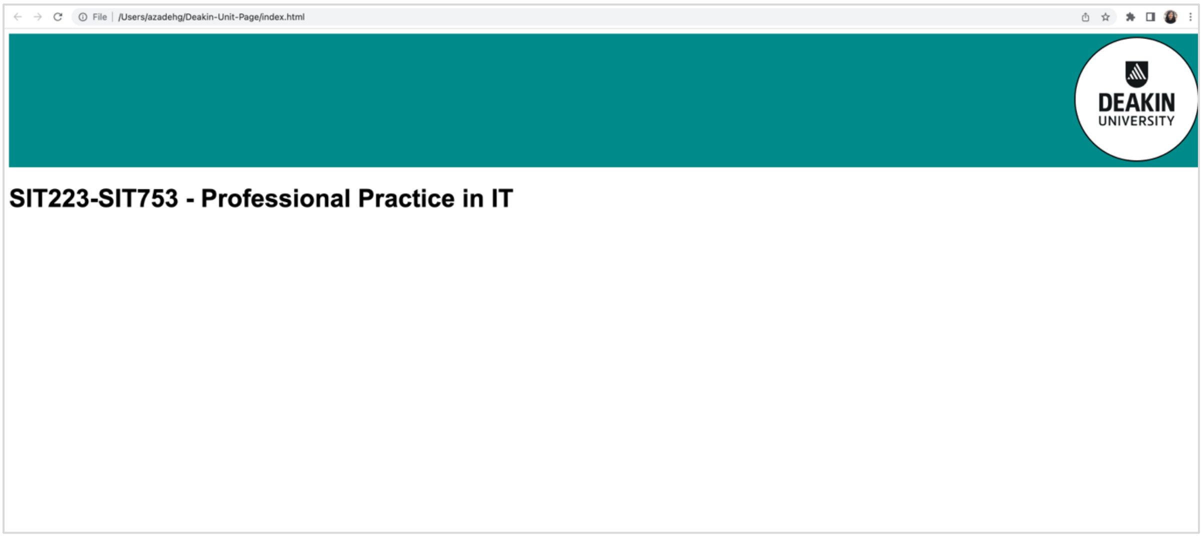
SIT223-SIT753 - Professional Practice in IT

29. Open the main.css file and add the **background color** and **text align** as shown in purple.

```
h1 {
  font-size: 40px;
  font-family: Arial
}
h2 {
  font-size: 25px;
  font-family: Arial
}
h3 {
  font-family: Arial
}
p {
  font-size: 14px;
  font-family: Arial
}

div {
  background-color: darkcyan;
  text-align: right;
}
```

30. **Save** the main.css file and **open** the index.html to see the changes as shown below.

```
open index.html
```



SIT223-SIT753 - Professional Practice in IT

31. Check the status of the project again to see the new changes. Note that the new files are ready to commit, but that the changes to webpage are *not staged*.

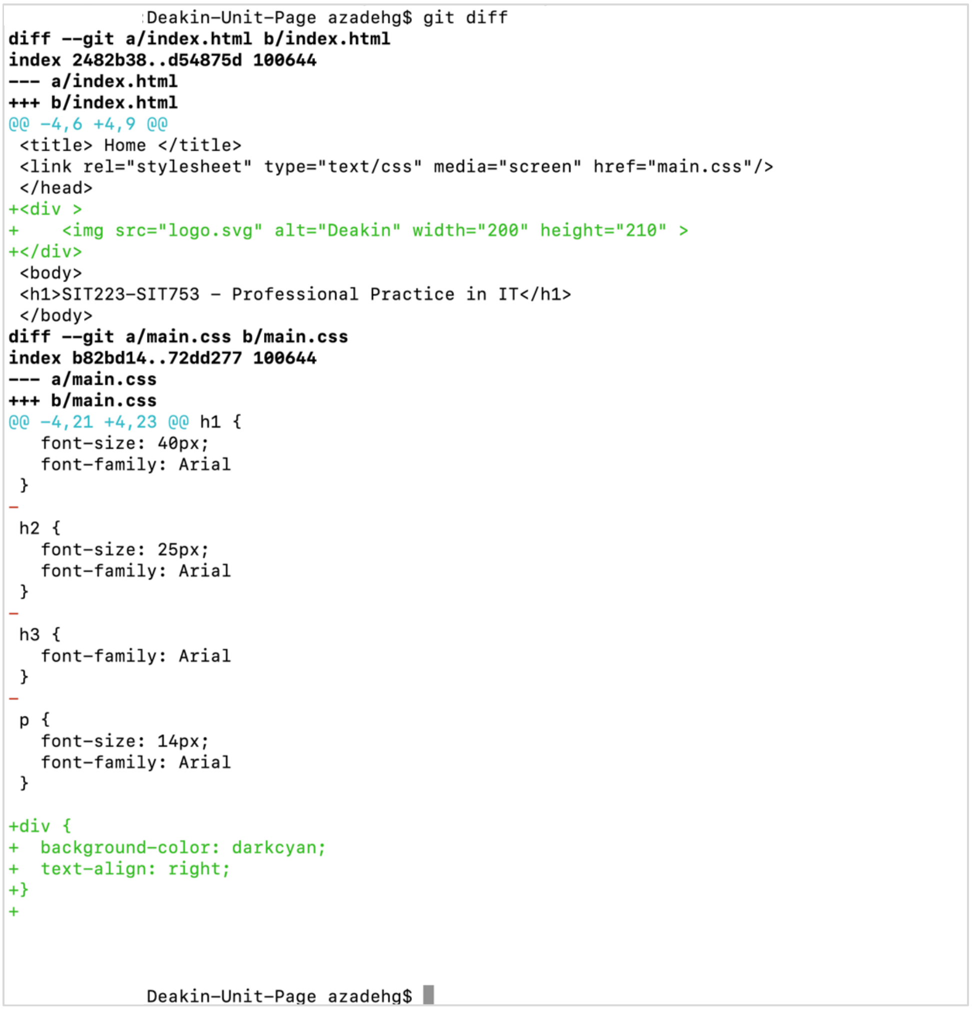
```
             Deakin-Unit-Page azadehg$ git status
On branch add-header
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
        modified:   main.css

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        logo.png
        logo.svg

no changes added to commit (use "git add" and/or "git commit -a")
```

32. To view the changes that will be made use git's diff. You can get the changes of all files in the project or for a single file.

```
git diff
```

OR

```
git diff index.html
```



```
                    Deakin-Unit-Page azadehg$ git diff
diff --git a/index.html b/index.html
index 2482b38..d54875d 100644
--- a/index.html
+++ b/index.html
@@ -4,6 +4,9 @@
 <title> Home </title>
 <link rel="stylesheet" type="text/css" media="screen" href="main.css"/>
 </head>
+<div >
+    <img src="logo.svg" alt="Deakin" width="200" height="210" >
+</div>
 <body>
 <h1>SIT223-SIT753 - Professional Practice in IT</h1>
 </body>
diff --git a/main.css b/main.css
index b82bd14..72dd277 100644
--- a/main.css
+++ b/main.css
@@ -4,21 +4,23 @@ h1 {
   font-size: 40px;
   font-family: Arial
 }
-
 h2 {
   font-size: 25px;
   font-family: Arial
 }
-
 h3 {
   font-family: Arial
 }
-
 p {
   font-size: 14px;
   font-family: Arial
 }

+div {
+  background-color: darkcyan;
+  text-align: right;
+}
+


                    Deakin-Unit-Page azadehg$
```

33. **Add** the updated file to the repository using the following command. This will stage this change, ready for you to commit it when you are ready.

```
git add .
```

34. Now you need to **commit** the changes to save them into the repository. To do this you will need to provide a message, indicating what the commit does to the repository.

```
git commit -m "add new header and logo"
```

Note: The message is what the commit does, so it will **add** the new background etc. Do not state what you did, so not "added new header" for example.

35. At this stage your changes are not on the server. You can now **push** your changes so that they are backed up. The first time you push a new branch you must tell git you want it to do up to the server (to the *origin*) and give it a name.

```
git push -u origin add-header
```

Now let’s switch to the **Front-end Developer** in your team and add the table.

Front-End Developer Tasks

Get your team together at your computer to observe the process.

36. Start by creating a new branch, name it add-table

    ```
    git checkout -b add-table
    ```

37. Open the index.html file with VS Code and **copy** the following header <h3> and table <table> as shown in purple into the index file and **save** the file.

```
<!DOCTYPE html>
<html>
<head>
<title> Home </title>
<link rel="stylesheet" type="text/css" media="screen" href="main.css"/>
</head>
<body>
<h1>SIT223-SIT753 - Professional Practice in IT</h1>
<h2> Unit details</h2>
<table style="width:100%">
  <tr>
    <td>Year: </td>
    <td>2023 unit information</td>
  </tr>
  <tr>
    <td>Enrolment modes:</td>
    <td>Trimester 1: Burwood (Melbourne), Geelong, Cloud (online)</td>
  </tr>
  <tr>
    <td>Credit point(s):</td>
    <td>1</td>
  </tr>
  <tr>
    <td> Unit Chair:</td>
    <td> Trimester 1: Azadeh Ghari Neiat</td>
  </tr>
</table>
</body>
</html>
```

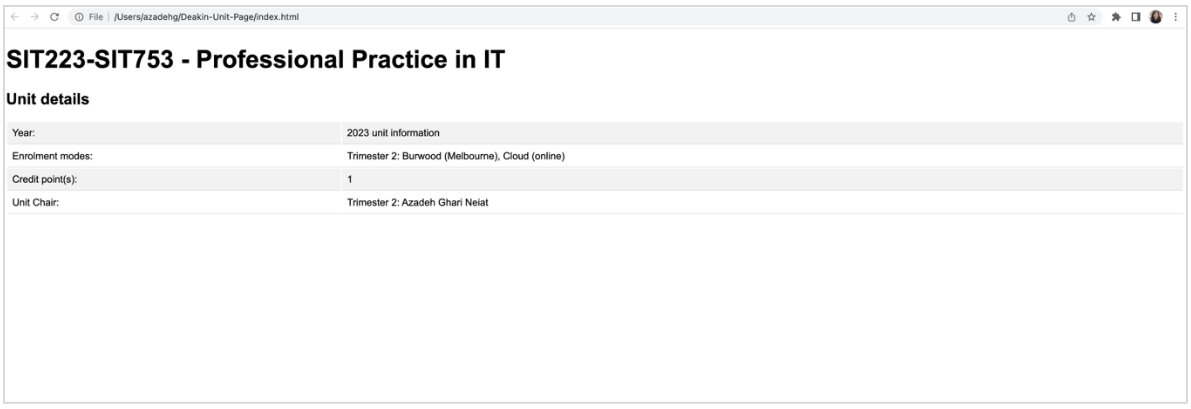
38. Open the main.css file and **copy** the following style for the table as shown in purple into the main.css file.

```
h1 {
  font-size: 40px;
  font-family: Arial
}
h2 {
  font-size: 25px;
  font-family: Arial
}
h3 {
  font-family: Arial
}
p {
  font-size: 14px;
  font-family: Arial
}
th, td {
  padding: 8px;
  text-align: left;
  border-bottom: 1px solid #ddd;
  font-family: Arial
}

tr:nth-child(odd) {background-color: #f2f2f2;}
```

39. **Save** the main.css file and **switch** to terminal and open the index.html to see the changes as shown below.

```
open index.html
```

Note: Why don't you see the new header and logo? That is in a different branch! You can work on your features without worrying about what others are doing… for a time.



/Users/azadehg/Deakin-Unit-Page/index.html

SIT223-SIT753 - Professional Practice in IT

Unit details

Year:	2023 unit information
Enrolment modes:	Trimester 2: Burwood (Melbourne), Cloud (online)
Credit point(s):	1
Unit Chair:	Trimester 2: Azadeh Ghari Neiat

40. Now check your status to see what has changed.

```
git status
```

41. You can also check the changes in the files using git diff.

```
git diff
```

Note: Press the spacebar to move through the pages in diff. Lines in red starting with a - have been removed, those in green with a + have been added. Pressing **q** gets you out of the diff.

42. Things look good, so you can now stage the files using git add, and commit them to the repository. Because you don't have any new files, and you want to add all the changes to the commit, you can do this with the -a option on git commit.

```
git add .
git commit -am "add table"
git status
```

Note: Watch out, git commit -am "..." will not add any new files. So only use this if you have just changed existing files.

43. Push your changes to the server, so that you won't lose them if something happens to your machine.

```
git push -u origin add-table
```

Now let's switch back to the **Team Leader**, and add unit contents and fee link.

Team Leader Tasks

Get your team together at your computer to observe the process.

44. Start by creating a branch, add-unit-content

```
git checkout -b add-unit-content
```

45. Open the **index.html** file and add the following header and paragraph in purple to it and then **save** it.

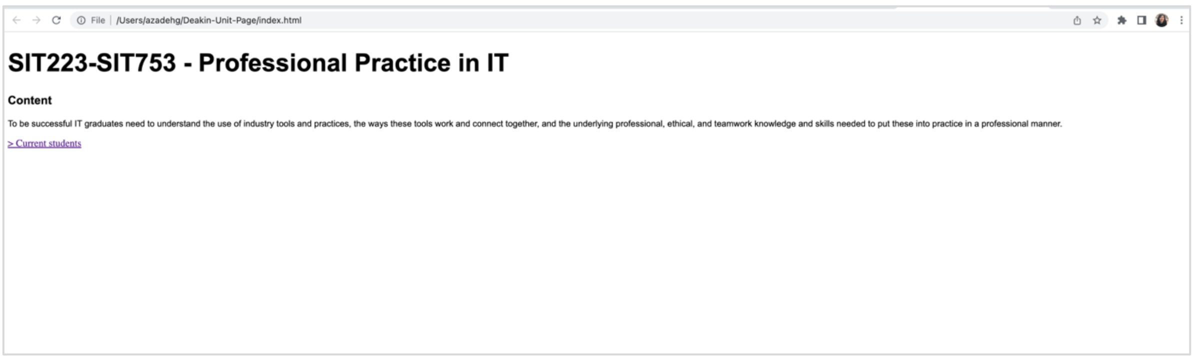
```
<body>
<h1>SIT223-SIT753 - Professional Practice in IT</h1>
<h3> Content</h3>
<p>
  To be successful IT graduates need to understand the use of industry tools and practices,
  the ways these tools work and connect together, and the underlying professional, ethical,
  and teamwork knowledge and skills needed to put these into practice in a professional manner.
</p>
</body>
</html>
```

46. After the paragraph, copy the following **link <a>** in purple into the index.html file.

```
<body>
<h1>SIT223-SIT753 - Professional Practice in IT</h1>
<h3> Content</h3>
<p>
  To be successful IT graduates need to understand the use of industry tools and practices,
  the ways these tools work and connect together, and the underlying professional, ethical,
  and teamwork knowledge and skills needed to put these into practice in a professional manner.
</p>
<a href="https://www.deakin.edu.au/students/enrolment-and-fees/fees"> > Current students</a>
</body>
</html>
```

47. **Save** the index.html file and **switch** to terminal and open the index.html to see the changes as shown below.

```
open index.html
```



SIT223-SIT753 - Professional Practice in IT

Content

To be successful IT graduates need to understand the use of industry tools and practices, the ways these tools work and connect together, and the underlying professional, ethical, and teamwork knowledge and skills needed to put these into practice in a professional manner.

> Current students

48. Check the status of the repository, then commit your changes and push to the server.

```
git status
git diff
git add .
git commit -am "add unit content and fee link"
git push -u origin add-unit-content
```

Now let’s switch back to the **UI Designer**, who can start to merge their changes into the main branch.

UI Designer Tasks

Get your team together at your computer to observe the process.

You are currently on the add-header branch, but we have made all of the necessary changes. What we can do now is merge all of our changes back into the main branch.

49. Checkout the main branch. This will switch everything back to what it was like in main when you started.

```
git checkout main
```

Note: Checkout -b will create a new branch, without -b and you switch to an existing branch. You could use git switch main instead. You could rename your **main** branch to **master** branch using `git branch -m master`.

50. Pull any changes that others have made. You should always do this before you merge so that you have the latest copy of main.

```
git pull
```

51. Now you can merge your branch into main. This will take all of your changes and apply them to the main branch.

```
git merge add-header
```

52. When you check the status, notice that we are now ahead of the server's main branch (origin/master). Push your changes so that main is updated for everyone.

```
git status
git push
```

Note: No need for -u on this push as the branch exists on the server already!

Let's switch over to the Front-end Developer again to have them add the table.

Front-end Developer Tasks

53. Now to merge them into main… but first let's **pull** any other changes.

```
git checkout main
git pull
```

54. Now merge your branch in… notice that git does all the hard work for you!

```
git merge add-table
```

55. Switch to the terminal and open the index.html file to see all changes.
56. Push the changes back to the server!

```
git push
```

Notice now that you have your work, and the header! Git takes care of merging as much of the work as it can. In some cases, you will need to intervene and resolve conflicts, but most of the time it should combine (merge) multiple peoples work without issue.

Now let's switch back to the Team Leader to finish off the last changes merge them in.

Team Leader Tasks

Get your team together at your computer to observe the process.

57. Switch to the main branch, and **pull** the remove changes.

```
git checkout main
git pull
```

58. Merge in your changes.

```
git merge add-unit-content
```

59. Switch to the terminal and open the index.html file. You should be able to see all the team's changes!

60. Push the changes back to the server!

    ```
    git push
    ```

Now it is time for each of you to make a small change on your own, and merge these back in to complete the project.

- Things for the **UI Designer** to change:
  - It would be good to change the header color to blue in main.css.

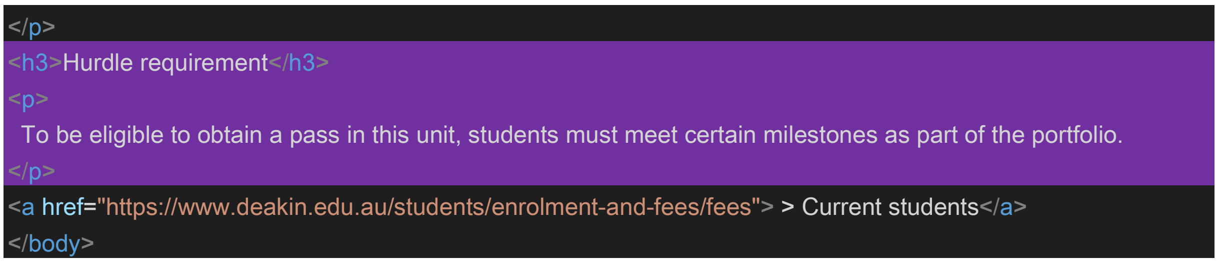
```
div {
  background-color: blue;
  text-align: right;
}
```

- Things for the **Front-end Developer** to change:
  - It would be good to add scheduled learning activities to the table as follows.

```
<table style="width:100%">
  <tr>
    <td>Year: </td>
    <td>2023 unit information</td>
  </tr>
  <tr>
    <td>Enrolment modes:</td>
    <td>Trimester 2: Burwood (Melbourne), Cloud (online)</td>
  </tr>
  <tr>
    <td>Credit point(s):</td>
    <td>1</td>
  </tr>
  <tr>
    <td> Unit Chair:</td>
    <td> Trimester 2: Azadeh Ghari Neiat</td>
  </tr>
  <tr>
    <td> Scheduled Learning Activities:</td>
    <td> 1 x 3 hour active class per week, weekly drop in support.</td>
  </tr>
</table>
```

- Things for the **Team Leader** to change:
  - It would be good to add the following hurdle requirement.

```
<h3> Content</h3>
<p>
  To be successful IT graduates need to understand the use of industry tools and practices,
    the ways these tools work and connect together, and the underlying professional, ethical,
    and teamwork knowledge and skills needed to put these into practice in a professional manner.
```



```
</p>
<h3>Hurdle requirement</h3>
<p>
  To be eligible to obtain a pass in this unit, students must meet certain milestones as part of the portfolio.
</p>
<a href="https://www.deakin.edu.au/students/enrolment-and-fees/fees"> > Current students</a>
</body>
```

Task Submission Instructions:

You must submit the following files to OnTrack:

- Images of the git contributions and branches. On GitHub select the Insights tab and get screenshots of the Contributors page and the Network page (two screenshots are required). These screenshots should reflect the people who contributed to the project repository, and a timeline of commits across different branches respectively.

Important Note: If the network graph is unavailable or does not display all branches as expected, please use the provided Git command to generate the log file and upload it via OnTrack.

```
git log --pretty=format:"%h - %an - %ad - %s" --date=short >
group_no_contribution_log.txt
```

- Images of the Sprint 1 Trello board with updated status.