

Bachelor of Computer and Information Sciences
Contemporary Issues in Software Engineering
Semester 2, 2023

ASSIGNMENT 1A: Set up Development Environment

Worksheet 1 (20% of Ass1A)

VS Code, Git, GitHub, Nest

Deliverables and Due dates:

You are required to complete the Worksheet and keep evidence as you do it by taking screenshots of your work, as well as explanations.

Each worksheet should be checked off by the TA by the end of the week's tutorial

This worksheet should be Checked off and Uploaded on CANVAS by end of Tutorial Week 1.

EXPERIMENT – BE CURIOUS – TEACH OTHERS – TAKE LOTS OF SCREENSHOTS

The worksheets will have some theory, a practical exercise, and a worksheet for answers to questions and progressively captured screenshots as evidence.

By the end of this Worksheet you should be able to....

1. Create branches and work with them appropriately in GitHub and Git
2. Keep a local repo synchronised with a GitHub repo using pull and push commands
3. Write useful commit messages
4. Use the pull request feature before merging code to Master branch in GitHub
5. Merge pull requests
6. Work with Git and GitHub from VS Code locally (make sure you know what is happening in Git!)
(Use the “GitHub Pull Requests and Issues extension”
<https://code.visualstudio.com/docs/editor/github>)
7. Understand what NEST is and how to use the Nest CLI to create a project.

Introduction

This worksheet introduces you to:

Git, a version control system (VCS) to manage code changes locally for each developer,

GitHub, a cloud-based version control system (VCS) to manage Git code repositories and share code

Visual Studio Code (VS Code), an IDE and text editor with some intelligent features and plug-ins for coding and integrating with other tools.

These tools are used to write, test, integrate and share code so developers can collaborate.

Exercise using VS Code, Git, GitHub & Nest

Do the following exercise:

1. Create a Github account if you have not already (note your token somewhere safe)
2. Create a new GitHub repository called CISE_React
3. Install Git on your local machine (<https://github.com/git-guides/install-git>)
4. Create a folder for your projects called "CISE_Repos" on your local machine
5. Install VS Code on your local machine <https://code.visualstudio.com/>
6. Install the "GitHub pull requests and Issues" extension for VS Code and authorise if necessary
7. Clone a copy of this empty repo to your local machine in the "CISE_Repo" folder using the following:

From a terminal window in VS Code, at the CLI type:

```
>git clone https://github.com/<Your GitHub username>/CISE_React
>cd CISE_React to change the working folder
>git status to check git is working.
```

Create a README.md file and push it to the main branch in GitHub

```
>echo "# CISE_React" >> README.md creates a README.md file with "CISE_React" as the title.
>git add README.md stages the README.md
>git commit -m "first commit" saves a snapshot of the changes to README.md with commit message
>git push -u origin main pushes the latest changes to GitHub remote main branch from the local main branch (You should see your README.md in GitHub now).
```

Your folder structure should now look like this:

CISE_Repo /

- CISE_React /
 - o README.md

8. Create locally a “Development” branch off the Master/Main branch and change to working in that branch

>git checkout -b Development This creates a new branch and moves you to the branch

>git branch -a to check what branches you have in your local repo type – green is the current one

9. Use VS Code to create a new file called Delete_me.html in the CISE_React folder and add the following code to the file Delete_me.html:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>Push this branch to GitHub
```

Stage and commit this file, with a useful commit message.

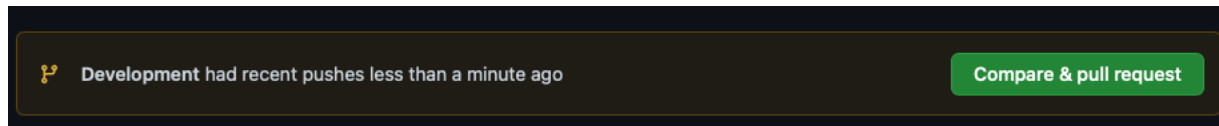
Note: Make sure to write useful commit messages (THIS IS VERY IMPORTANT – it documents the reasons for changing and what was changed). Read the following articles to understand:

<https://betterprogramming.pub/why-every-git-commit-message-must-include-its-commit-context-1171c0b2f710>

<https://dev.to/helderburato/patterns-for-writing-better-git-commit-messages-4ba0>

10. >git push -u origin Development to push your changes to GitHub

11. Check Delete_me.html has been sent to GitHub in the Development branch in GitHub. You should see a message in GitHub like the following:



12. Press the green button and continue accepting until you have merged this change with the main branch on GitHub. Usually the main branch is the production branch to be deployed to users.

It is common to create a branch structure with feature branches coming off the Development branch like this (Fig 1)

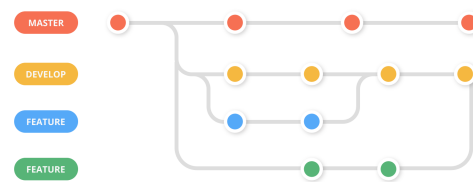


Fig 1

13. We are now going to create a simple React app using a script called “create-react-app”. In order to use the script and some other features we need to install node.js, which is a javascript run-time that allows us to run javascript outside the browser (thank you Google!)
14. Follow the instructions to install node.js on your Operating System
<https://nodejs.org/en/download/>
15. To check it has installed and the version you have, at the command prompt you can type

```
>node -v
```

16. To install all the dependencies for a React application we can install the “create-react-app”. This will create a new folder called cise-react-learn for this app.

```
>npx create-react-app cise-react-learn
```

```
>cd cise-react-learn to change to the new project directory
```

```
>npm start
```

 will start the development web server and open a browser to show you the simple app it created with a spinning logo.

README.md is a markdown file that includes a lot of helpful tips and links that can help you while learning to use Create React App.

node_modules is a folder that includes all of the dependency-related code that Create React App has installed. You will never need to go into this folder.

package.json that manages our app dependencies and what is included in our node_modules folder for our project, plus the scripts we need to run our app.

.gitignore is a file that is used to exclude files and folders from being tracked by Git. We don't want to include large folders such as the node_modules folder

public is a folder that we can use to store our static assets, such as images, svgs, and fonts for our React app.

src is a folder that contains our source code. It is where all of our React-related code will live and is what we will primarily work in to build our app.

17.

>git add . (note the dot) to get Git to stage all files in the cise-react-learn folder.

>git commit -m "first commit" to commit all the files created

>git push origin Development to push all the files created to the Development branch of GitHub

18. Make a change to App.js file in the src folder – change “Learn React” to “Agile is about Values and Principles” and save the file. Notice the filename is yellow in VS Code – indicating it is not committed.

Note that your app should have updated in the web browser automatically.

19. Commit this change and push it to GitHub. (Remember the good quality commit message discussed in step 9!)

20. (Check that the files were updated in GitHub)

(Ctrl-c in terminal windows will stop the program running)

Collaborate on a Repository with a team

21. In Git create another branch “LogoLink” from the Development branch and move to it

22. Change the App.js file in VSCode so clicking the link takes you to aut.ac.nz instead of reactjs.org

23. Commit and push to GitHub creating the new “LogoLink” branch

24. Invite a classmate or friend who has a GitHub account (or will make one) to be a collaborator with your repository. (or make another GitHub account using a different machine).

“Settings>Manage Access” in GitHub. you will need to know your collaborator’s GitHub account name

25. **Create a pull request** for this merge of the “LogoLink” branch with the Development branch, so the changes are in this branch. Other developers working on the same repository would clone this branch to get the latest code to work on several times per day. You can create the pull request with Git command or use the extension you installed into VS Code (see Fig 3).

26. Go to GitHub on your browser and you should see one open pull request. The Reviewer assigned to the pull request would normally review this code to be merged to check for errors etc and there may be a discussion with the original author of the code to fix something or get clarification. **Get your reviewer to open the pull request and merge this change with the Develop branch**

27. Install the Prettier plug-in to VS Code and see how it works. Read about the purpose of a Linter – we will use ESLint. Try installing the ESLint plug-in for VS Code.

Work with NEST

Step 28: Install NEST CLI on your machine. Open a terminal and use the following command to install the CLI globally: <https://docs.nestjs.com/cli/overview>

```
>npm i -g @nestjs/cli
```

Step 29: Verify the installation of NEST CLI by checking its version. Type the following command:

```
>nest --version
```

(Note you may not have permissions on your system, if you are getting errors run this command for windows OS)

```
> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
```

Step 30: Create a new project using NEST CLI. In your terminal, navigate to the directory CISE_Repos and then create the project and then run the following command:

```
>nest new my-nest-project
```

Replace "my-nest-project" with the name you want for your project. NEST CLI will create a new directory with this name and set up a new NEST project inside it (Select npm as the package manager).

Your folder structure at this point should look like this:

```
CISE_Repo /
- CISE_React /
  o README.md
- my-nest-project /
  o ALL THE CREATED NEW FILES ...
```

Step 31: Navigate into your new project directory using the command:

```
>cd my-nest-project
```

Step 32: Run the NEST application. Inside the project directory, type the following command:

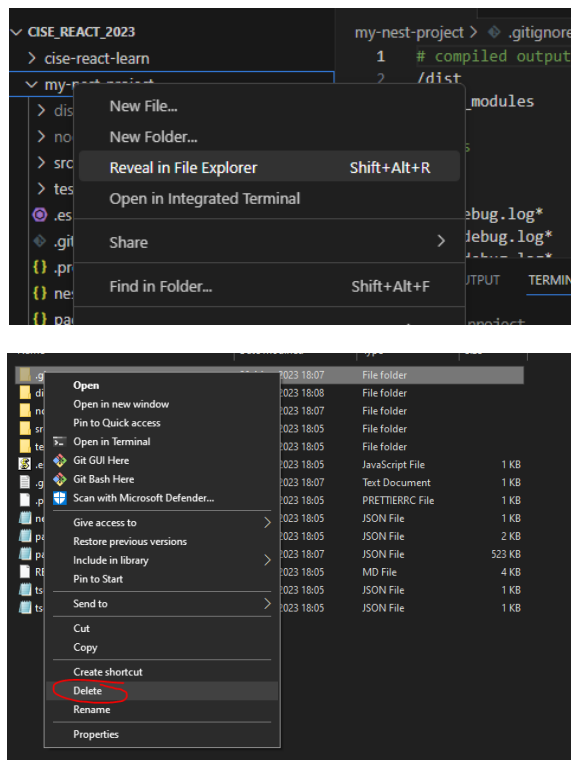
```
>npm run start
```

Your NEST application is now running at `http://localhost:3000`. You can open this URL in your browser to see your application.

Step 33: Make some changes to the application. Open the project in your VS Code, and navigate to `src/app.controller.ts`. Here, you can change the string returned by the `getHello()` function. Save your changes and check your application at `http://localhost:3000` again to see the updated message.

Step 34: Once you have made your changes, stop the NEST application by pressing `Ctrl + C` in your terminal.

When creating a new Nest project, it creates a **.git** file. We do not need this so reveal in file explorer and remove this file. (You will not see this file in the project unless you have hidden files shown enabled).



Now you can commit everything to this week repository.

Worksheet Evidence:

1. What is the purpose of Git and GitHub?

Git is a version control system for developers to code locally and track changes within every file. Github is cloud based version control system and the purpose of it is to be able to share files and codes with others as well as tracking codes and files online.

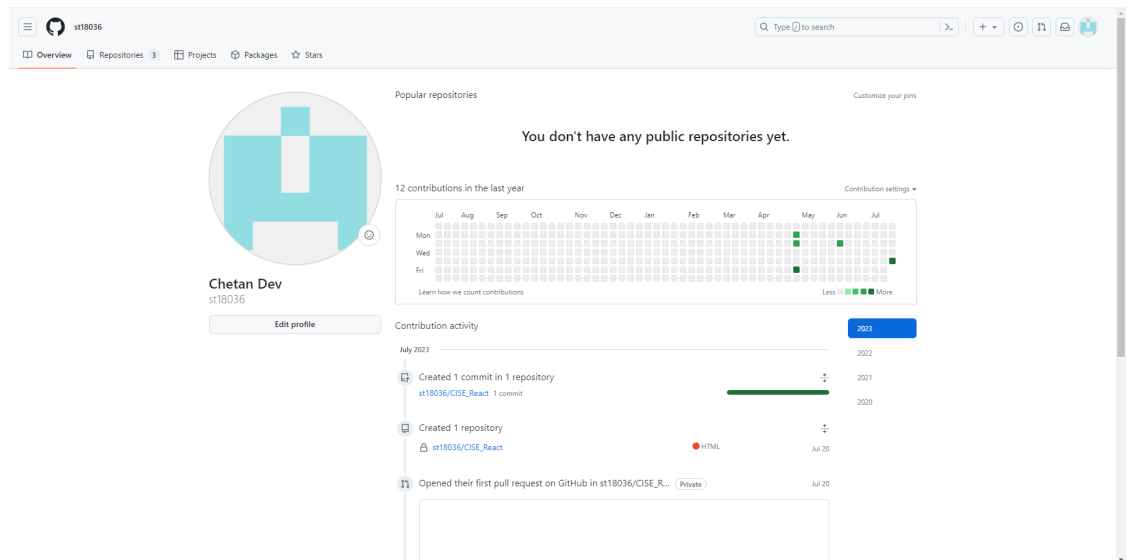
2. Explain the difference between a local repository and a remote repository in the context of Git and GitHub.

A local repository is different to a remote repository because a local repository can only be accessed locally which means it can only be accessed if you or on the exact computer where the repository has been created. The remote repository can be accessed through github digitally therefore anyone that is a part of the repository can connect and access the files from any computer as it is cloud based server.

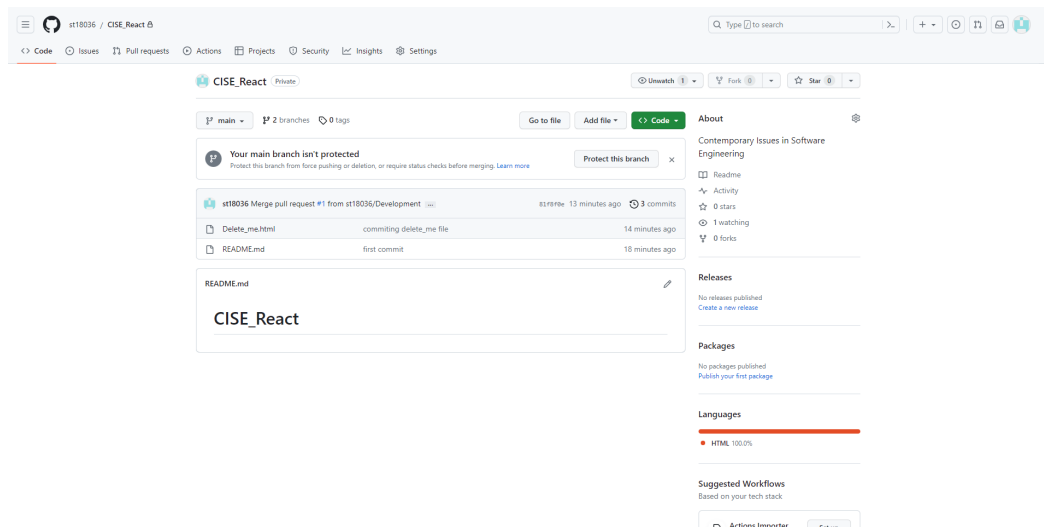
3. Provide screenshots to evidence the following tasks:

a. Creating a GitHub account

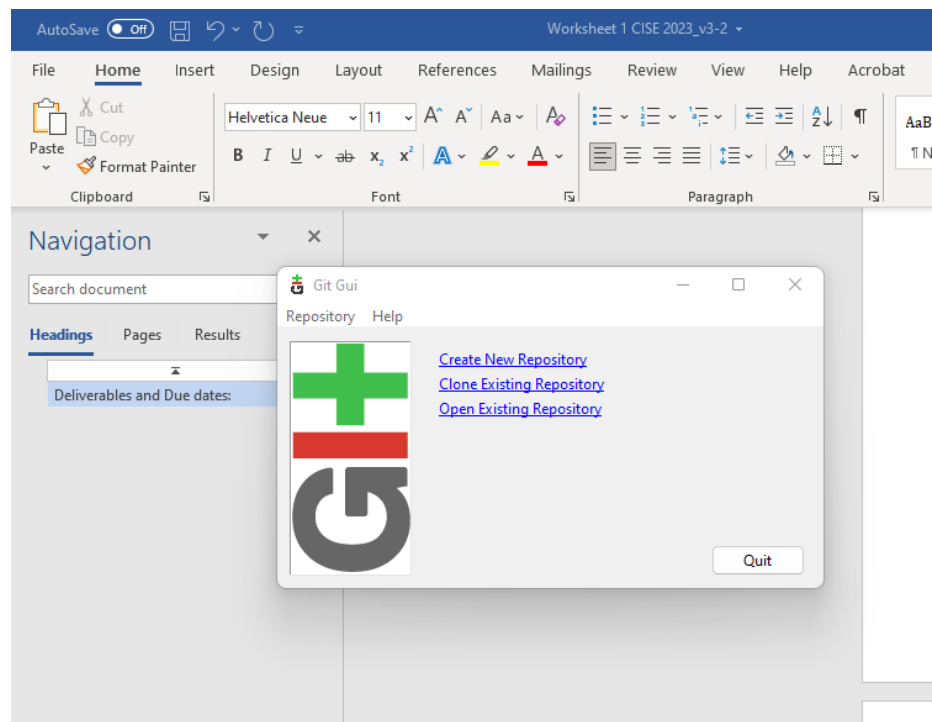
I already had a GitHub Account



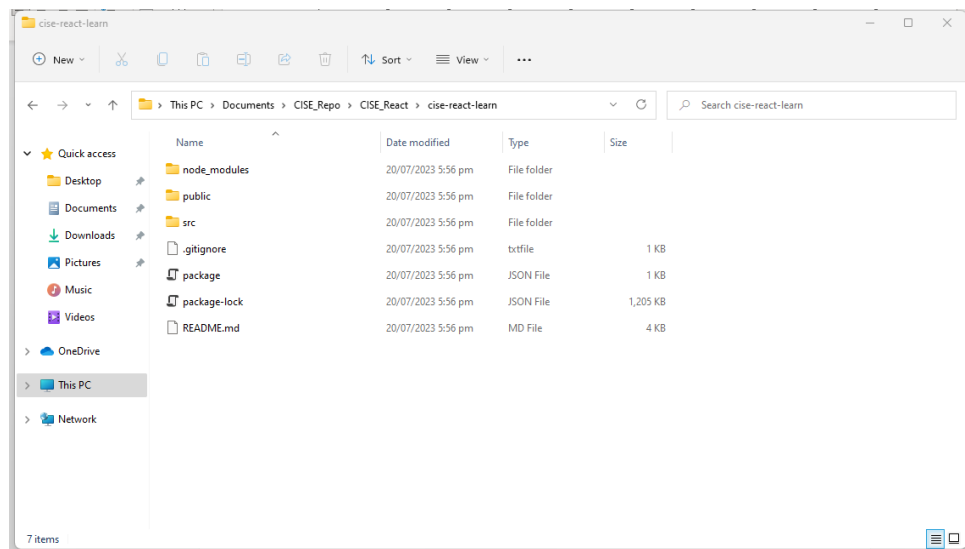
b. Creating a GitHub repository named CISE_React



c. Installing Git on your local machine

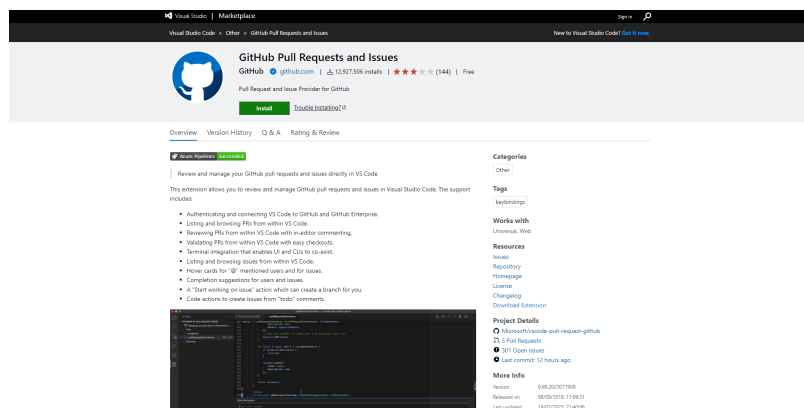


d. Cloning the CISE_React repository to your local machine



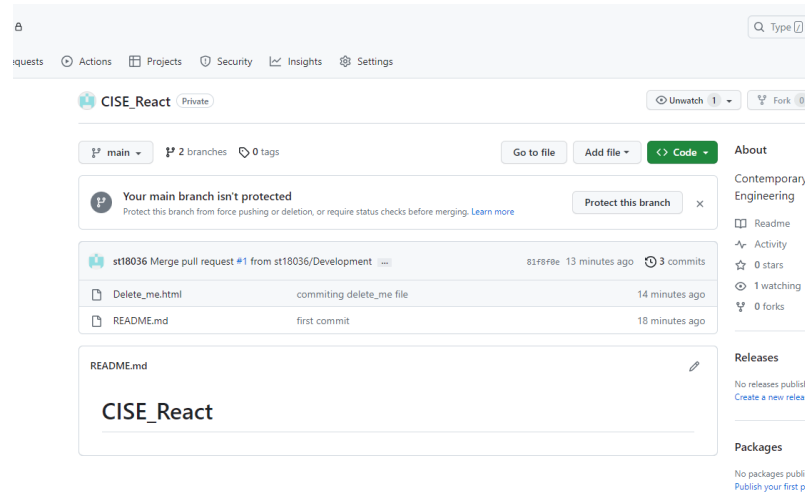
f. Installing and authorizing the “GitHub Pull Requests and Issues” extension for VS Code.

I have previously installed it



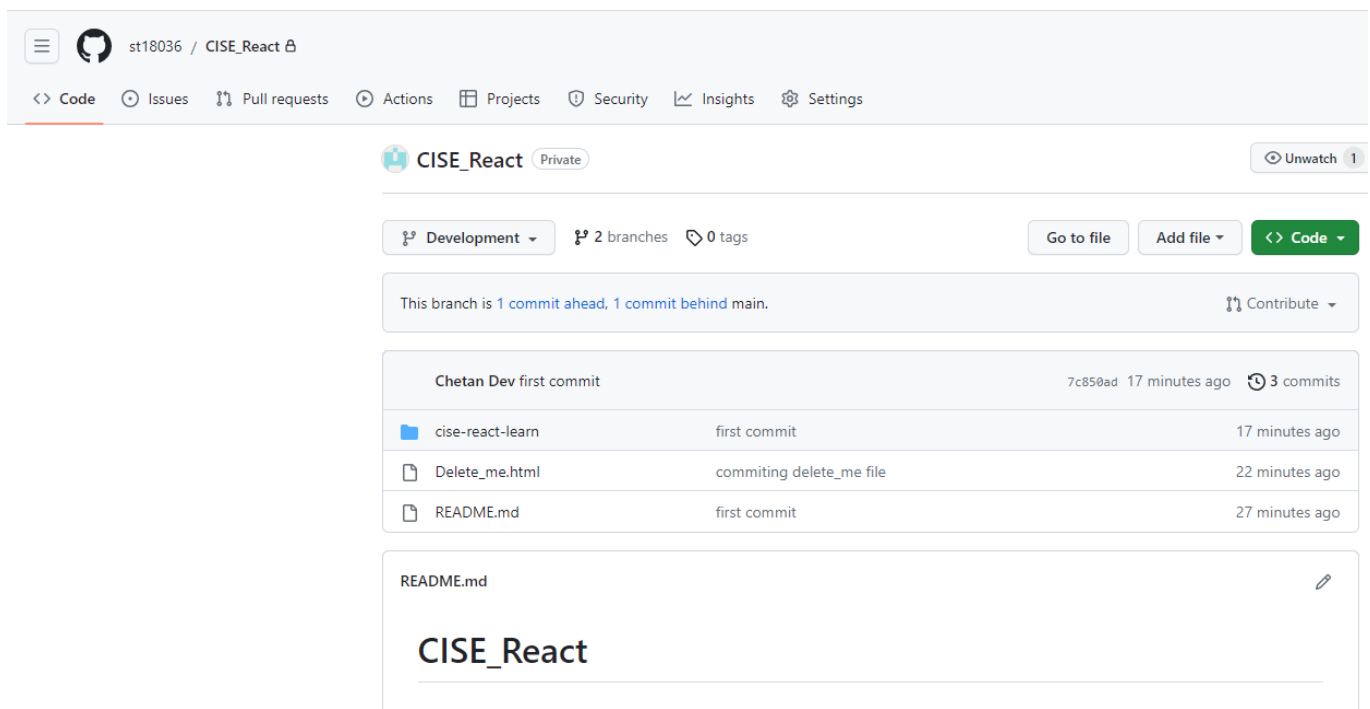
4. What is the role of README.md file in a GitHub repository? Provide a screenshot of the README.md file you have pushed to your main branch.

The role of the readme file in a git repository is so you can write descriptions of your project/repository for anyone that is going to view it to give an understanding of what the repository is about

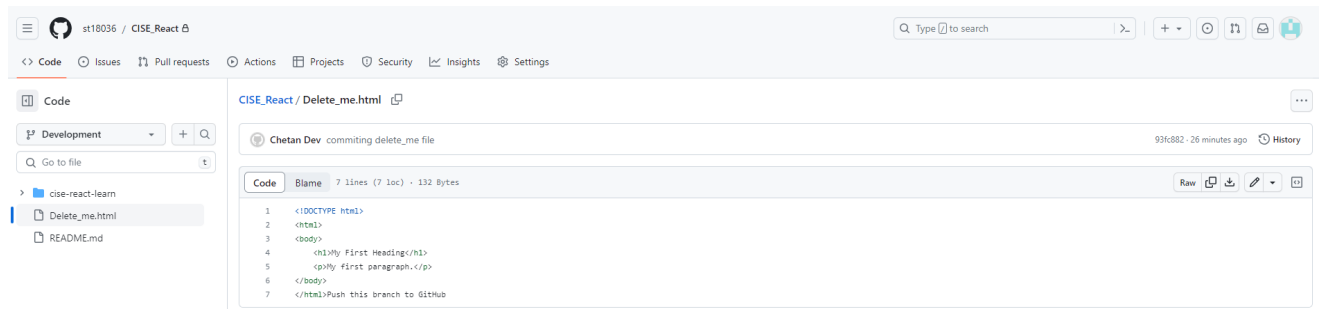


5. Explain the purpose of creating branches in GitHub. Show a screenshot of your “Development” branch.

There are multiple purposes when creating a branch in GitHub, it could be to test new code in a different area so it doesn't affect the original branch code. It could be to create a nice understanding repository for anyone that is viewing if your project is large than it could be a good idea to branch files off rather than have everything in one branch. It could also be to develop features.



6. Provide a screenshot showing the successful addition of the Delete_me.html file to your Development branch on GitHub.



7. Explain the steps you took to merge your changes to the main branch on GitHub. Provide screenshots to support your answer.

There was a notification on github stating “Development had recent pushes less than a minute ago”.

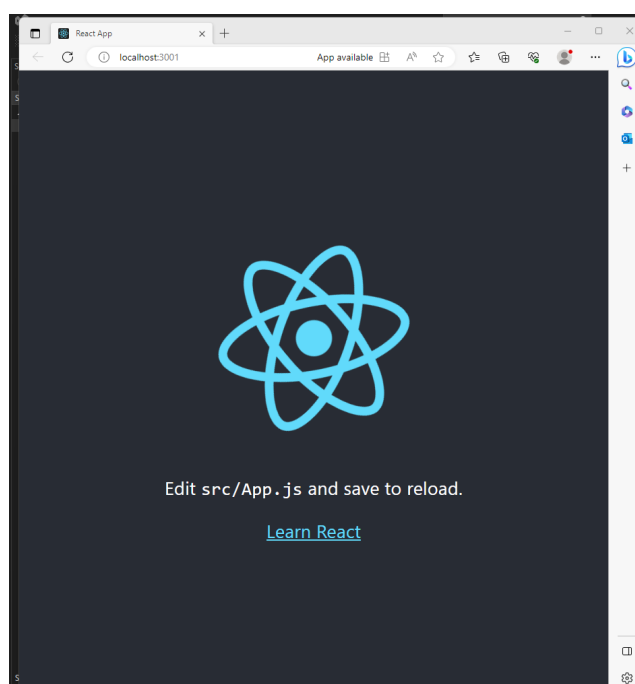
Then I clicked a green button “Compare and Pull request” which took me to a page to confirm and accept the merge with the main branch.

8. Explain what Node.js is and why it's important for running JavaScript outside of a browser.

Node.js is a open source server that is able to run on many operating systems. In a nutshell Node.js is a back end Java Script runtime environment. Node.js is important because it allows developers to run Node.js on server-side for both front and back-end development making it more efficient.

9. What is the "create-react-app" script used for? Provide screenshots of your successful creation of the cise-react-learn React application.

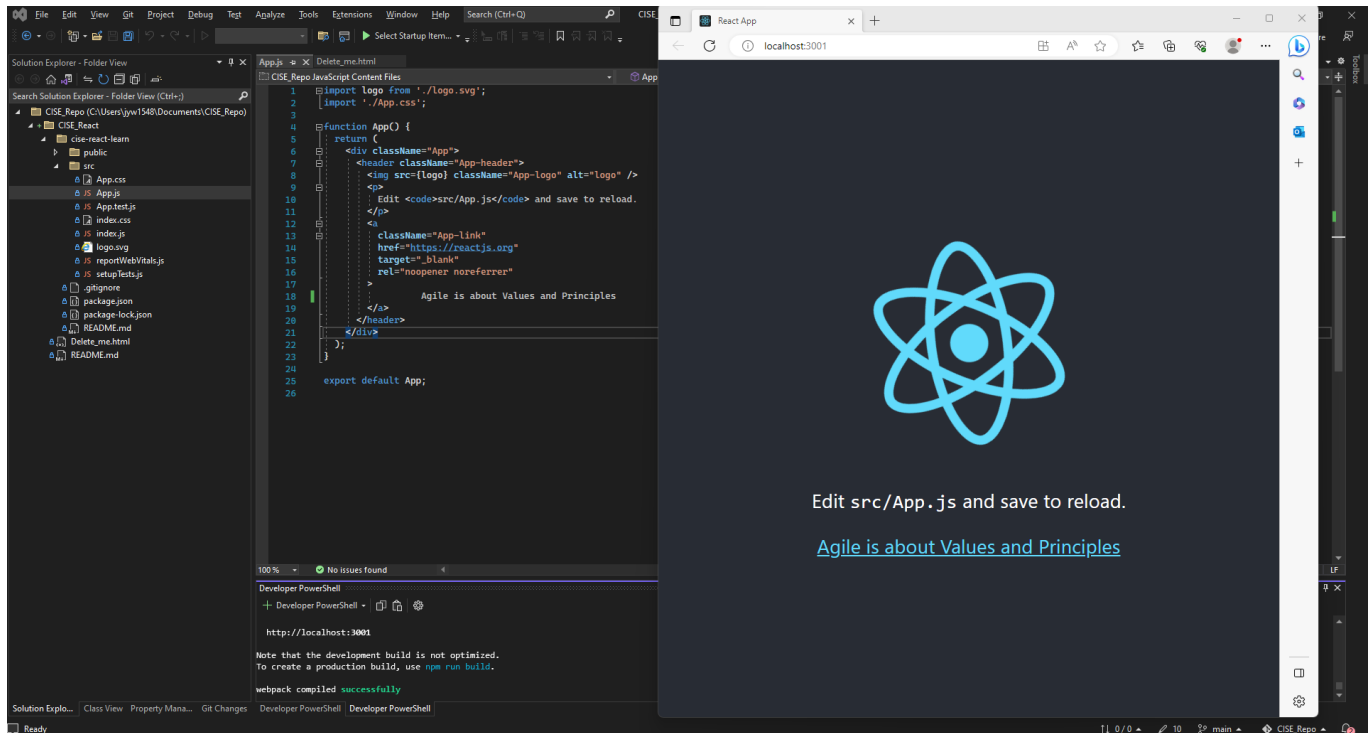
The script is used for beginners to quickly create and run the project onto the browser. The script is a tool to single-page React applications.



10. What are the roles of the package.json and .gitignore files in a React application?

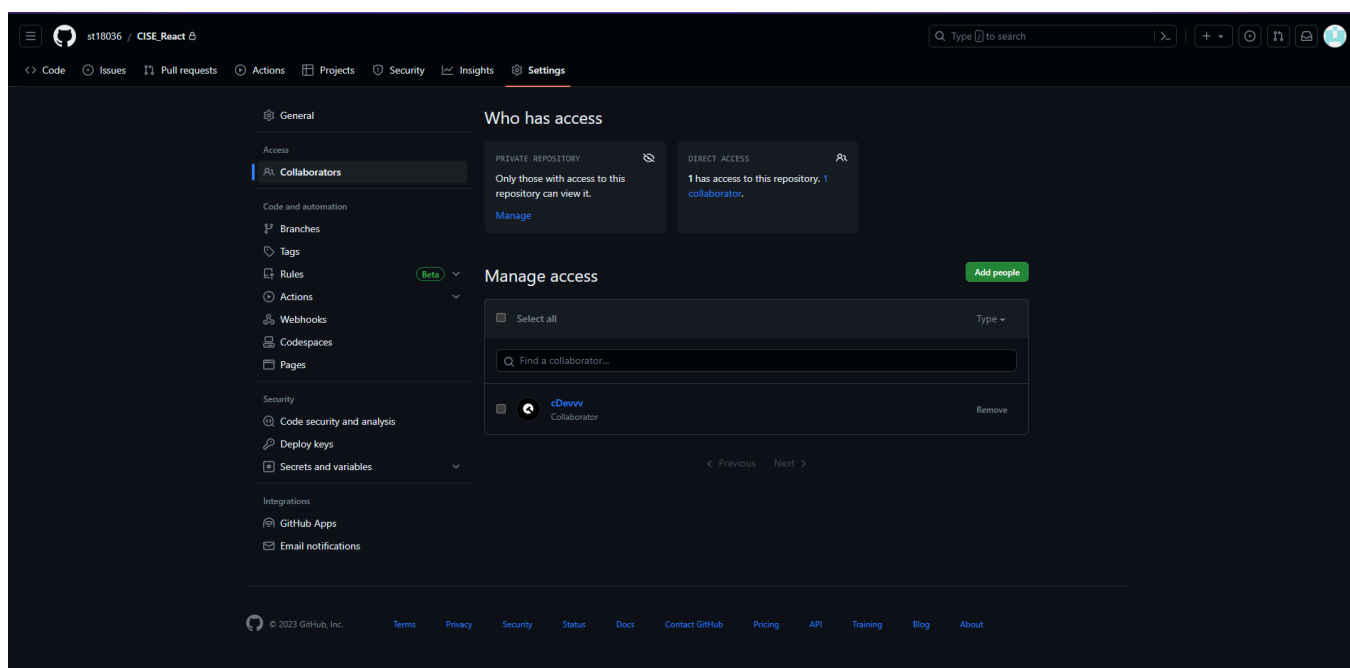
The package.json files contain metadata about the project, it enables npm to start the project and run scripts. .gitignore files are files that are to be ignored by git when you make a commit.

11. Show screenshots of changes made to the App.js file in the src folder and confirm these changes have been updated in the browser automatically.



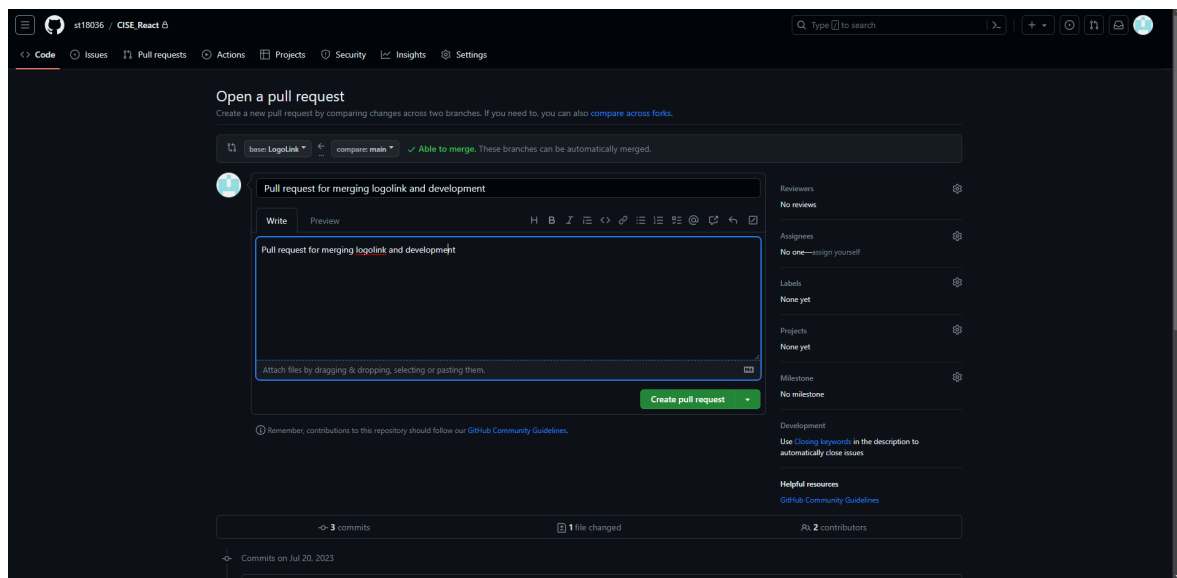
12. How can you invite others to collaborate on a GitHub repository? Provide a screenshot showing you've invited a collaborator to your repository.

You can invite others by going into settings > manage access to invite others to collaborate.



13. Explain the purpose of a pull request in GitHub. Provide a screenshot of the open pull request for merging the "LogoLink" branch with the Development branch.

The purpose of the pull request is to tell others about changes to a branch in a repository where you can discuss and review the potential changes with collaborators.

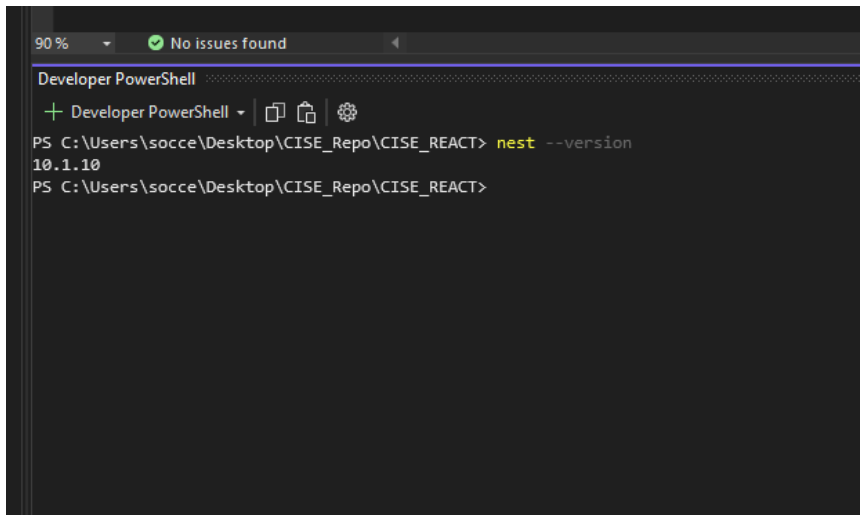


14. What are the benefits of using a Linter like ESLint and code formatting tool like Prettier in your project?

The benefit to using ESLint is that it does automated scans of JavaScript files for common syntax and style errors. Prettier is a tool that analyses your code files to detect any style-related problems and automatically applies formatting changes.

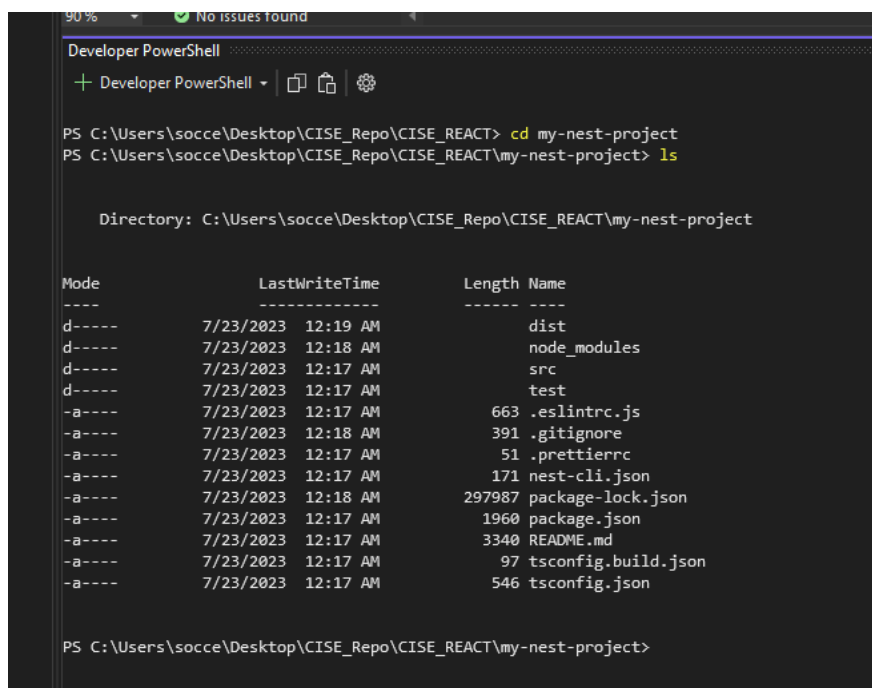
15. Explain what NEST is and what the NEST CLI is used for. Provide a screenshot of the installed NEST CLI on your machine.

NEST is a framework for building server-side applications with Node.js using TypeScript. It helps developers create powerful and reliable applications. The NEST CLI is a tool that makes it easier to create and manage NEST projects. It automates repetitive tasks and helps in generating code components like modules, controllers, and services.



```
90 %  No issues found
Developer PowerShell
+ Developer PowerShell
PS C:\Users\socce\Desktop\CISE_Repo\CISE_REACT> nest --version
10.1.10
PS C:\Users\socce\Desktop\CISE_Repo\CISE_REACT>
```

16. Provide screenshots to show the successful creation of a new project using NEST CLI and running of the NEST application.



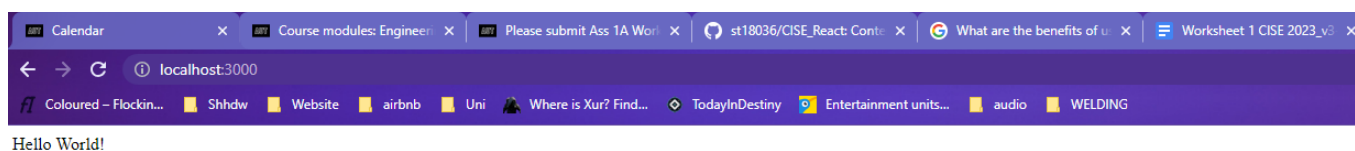
```
90 %  No issues found
Developer PowerShell
+ Developer PowerShell
PS C:\Users\socce\Desktop\CISE_Repo\CISE_REACT> cd my-nest-project
PS C:\Users\socce\Desktop\CISE_Repo\CISE_REACT\my-nest-project> ls

Directory: C:\Users\socce\Desktop\CISE_Repo\CISE_REACT\my-nest-project

Mode                LastWriteTime         Length Name
----                -
d-----          7/23/2023  12:19 AM             dist
d-----          7/23/2023  12:18 AM          node_modules
d-----          7/23/2023  12:17 AM             src
d-----          7/23/2023  12:17 AM             test
-a-----          7/23/2023  12:17 AM           663 .eslintrc.js
-a-----          7/23/2023  12:18 AM           391 .gitignore
-a-----          7/23/2023  12:17 AM            51 .prettierrc
-a-----          7/23/2023  12:17 AM           171 nest-cli.json
-a-----          7/23/2023  12:18 AM          297987 package-lock.json
-a-----          7/23/2023  12:17 AM           1960 package.json
-a-----          7/23/2023  12:17 AM           3340 README.md
-a-----          7/23/2023  12:17 AM            97 tsconfig.build.json
-a-----          7/23/2023  12:17 AM            546 tsconfig.json

PS C:\Users\socce\Desktop\CISE_Repo\CISE_REACT\my-nest-project>
```

17.



18. Show a screenshot of changes you made to the getHello() function in the src/app.controller.ts file of the NEST application and the updated message in the browser.

