

WEB ICP1

Avinash Reddy T

Email: atfkh@umsystem.edu

GitHub: <https://github.com/avinashreddy3/WebDevCourse/tree/main/WebPart/ICP1>

Venubabu Linga

Email: vl6hw@umsystem.edu

GitHub: https://github.com/VenubabuLinga/WebDevCourse/tree/main/WEB_DEV/ICP1

WebStorm Tool

The smartest JavaScript IDE

WebStorm is an integrated modern JavaScript ecosystem. It includes intelligent code completion, on-the-fly error detection, powerful navigation and refactoring for JavaScript, Typescript, stylesheet languages, and all popular frameworks.

Like other Jet Brains IDEs, it makes your development experience more enjoyable, automating routine work and helping you handle complex tasks with ease.

WebStorm Features:

- Intelligent coding assistance.
 - Modern frameworks
 - Smart editor
 - Navigation & Search
- Debugging, tracing and testing.
 - Debugging
 - Testing
 - Tracing and profiling
- Seamless tool integration.
 - Build tools
 - Code quality tools
 - Project templates

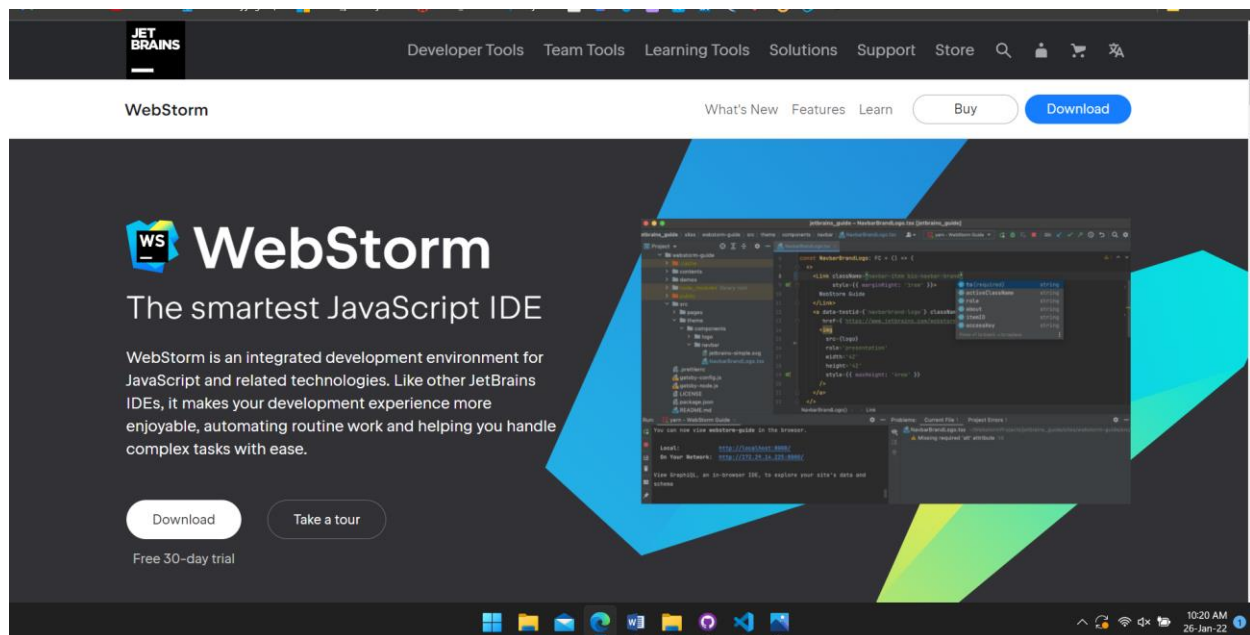
- **IDE features** - WebStorm is built on top of the open-source IntelliJ Platform. Enjoy the fine-tuned, yet highly customizable experience it provides to fit your development workflow.
 - VCS
 - Local history
 - Customization

➤ **Show that you have WebStorm installed on your machine**

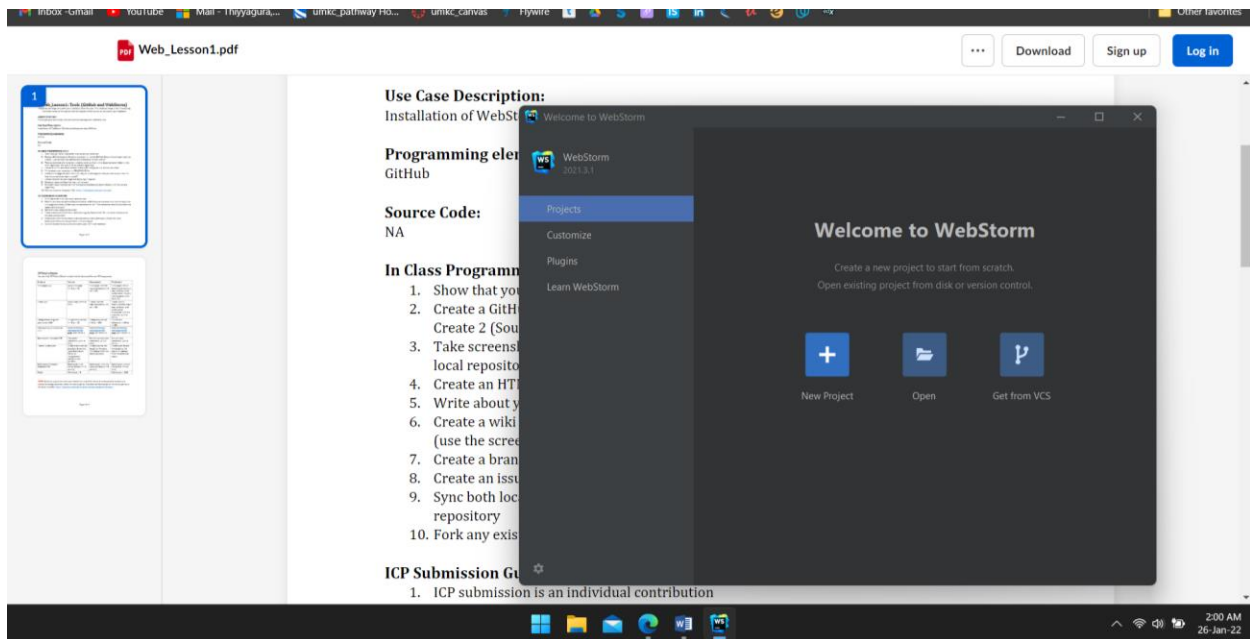
I opted webstorm tool as an editor out my interest. Since there are many features where a few are mentioned above. I use the editor in my local machine as well to develop websites and webpages with technologies like HTML, CSS, Java Script, bootstrap.

Please find the screenshots for the webstorm installation and its activation process, below.

Use this link to install webstorm, [WebStorm: The Smartest JavaScript IDE, by JetBrains](#)

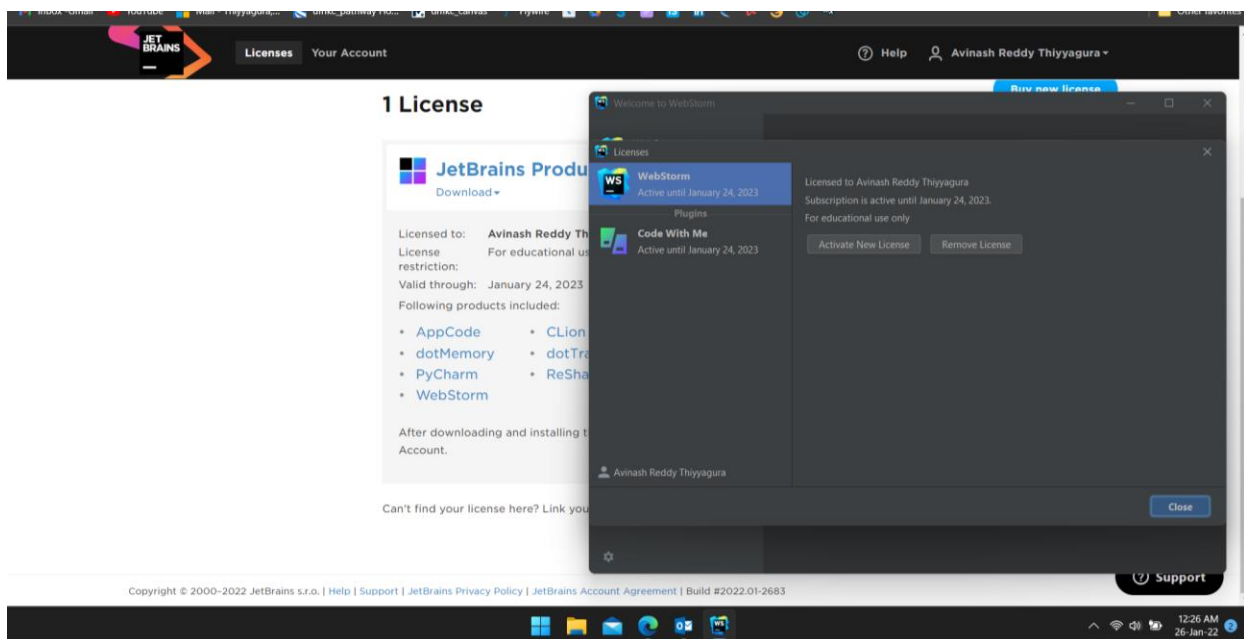


WebStorm installed on the machine

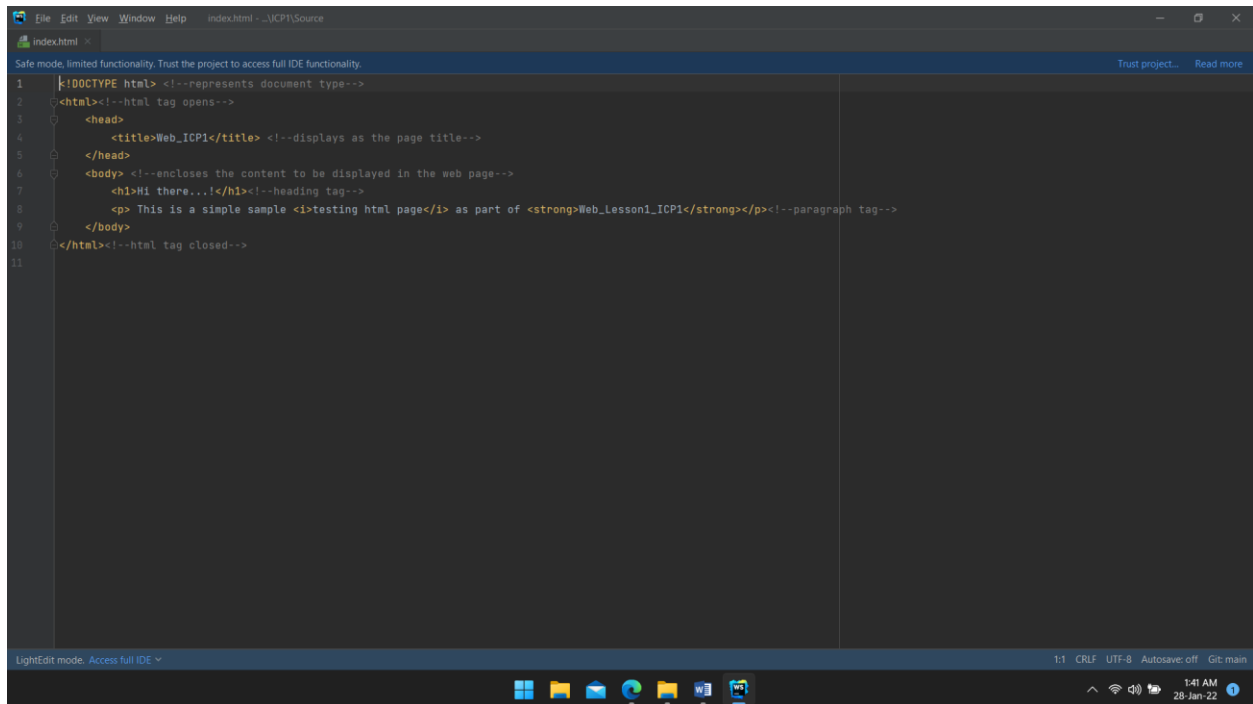


Use this link and apply with university student Id for free jet brains webstorm licensed account, [Free Educational Licenses - Community Support \(jetbrains.com\)](https://jetbrains.com/Free-Educational-Licenses-Community-Support)

Webstorm license activated.



Below screenshot shows the code development and working with webstorm on my local machine.



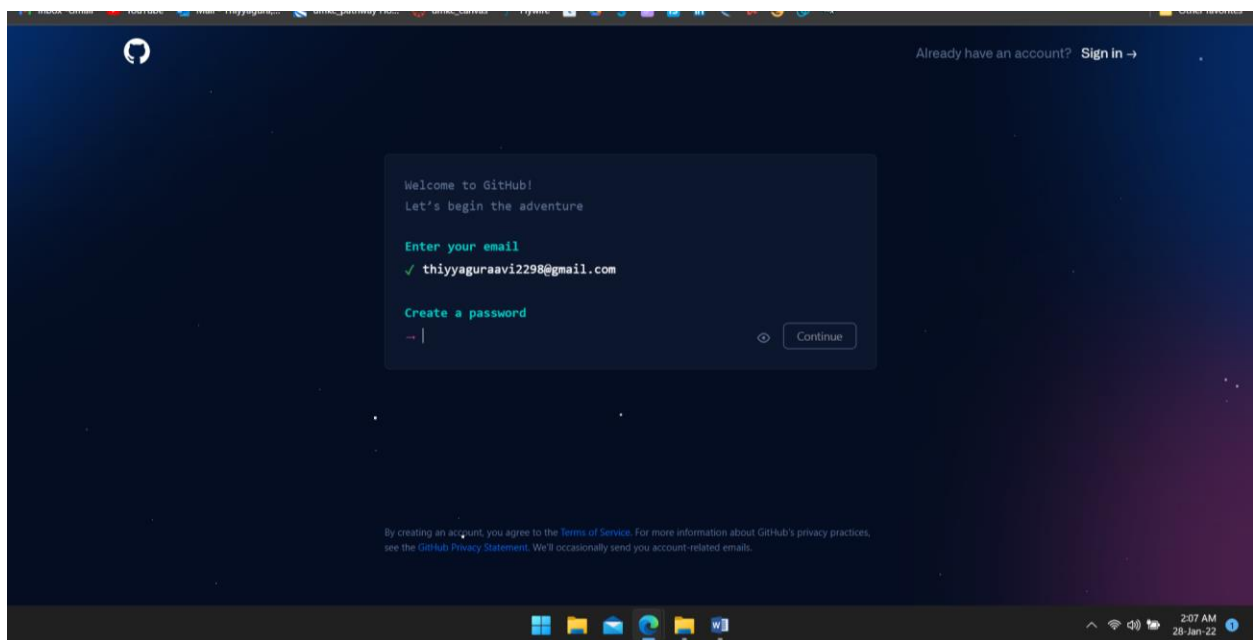
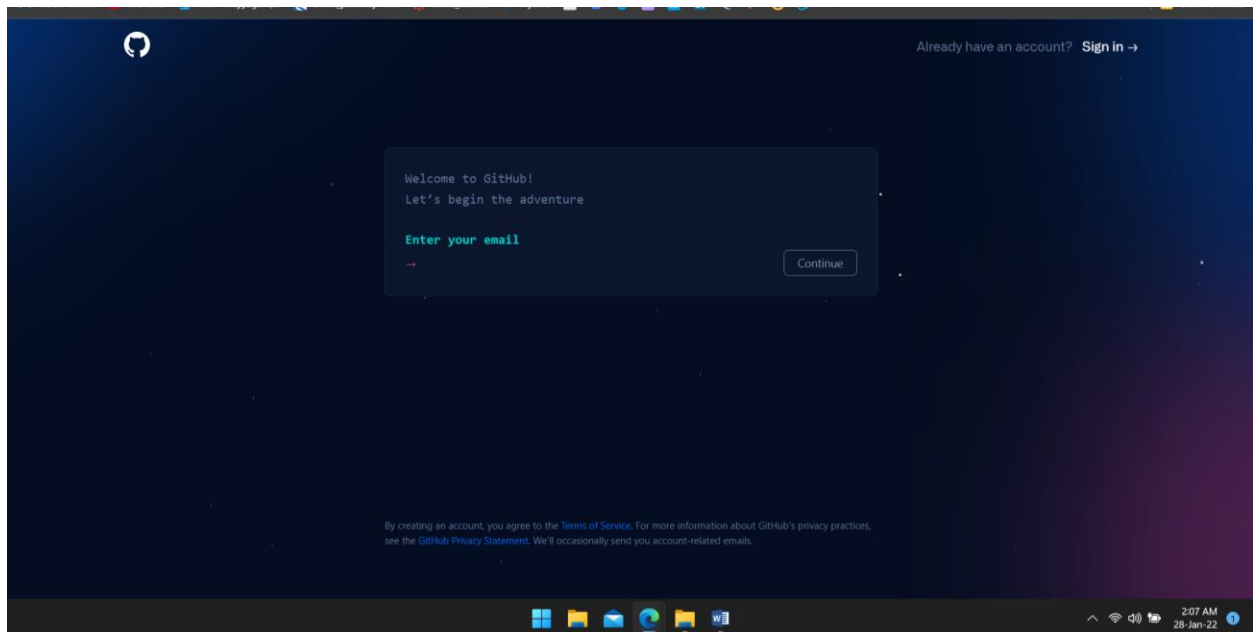
2. Create a GitHub account. Create a repository in remote GitHub. Clone it to the local machine. Create 2 (Source and Documentation) directories in local GitHub

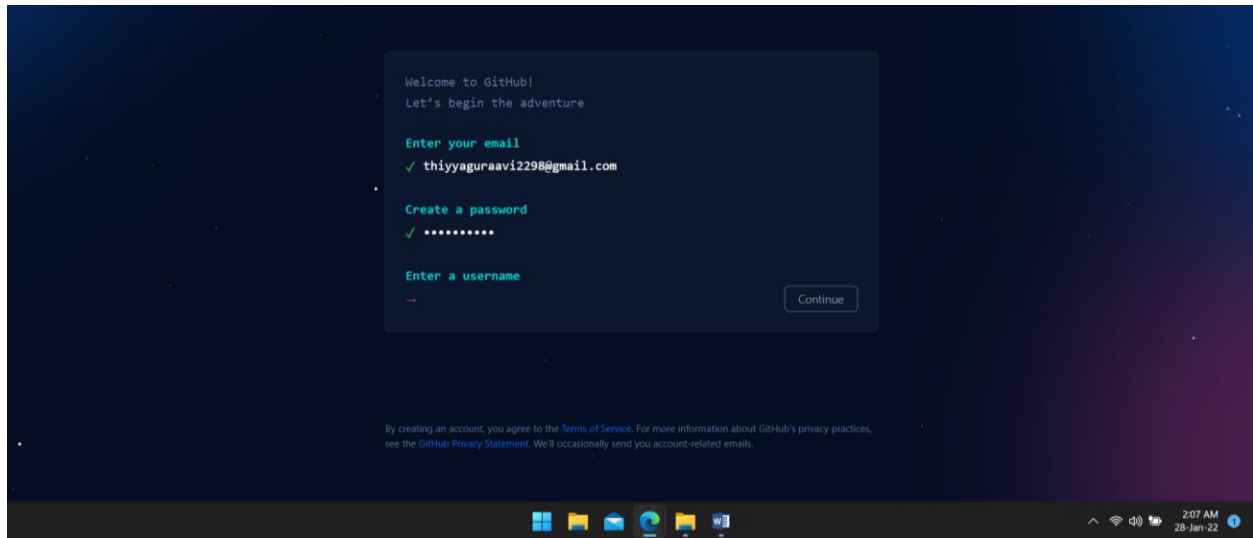
GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere to develop and customize their applications to the desire.

Main benefits of using GitHub?

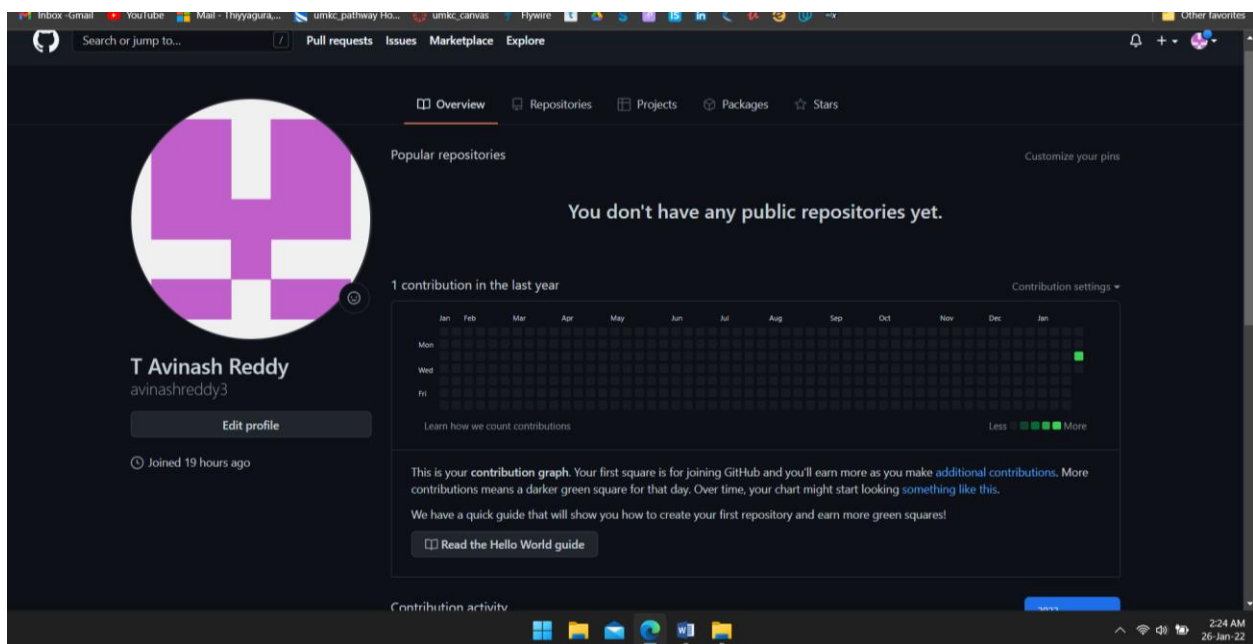
- 1) It makes it easy to contribute to your open source projects
- 2) Documentation
- 3) Showcase your work
- 4) Markdown
- 5) Easy to collaborate with Team during developing
- 6) Track changes in your code across versions
- 7) Integration options

Please find the screenshots for the GitHub account creation process, below.

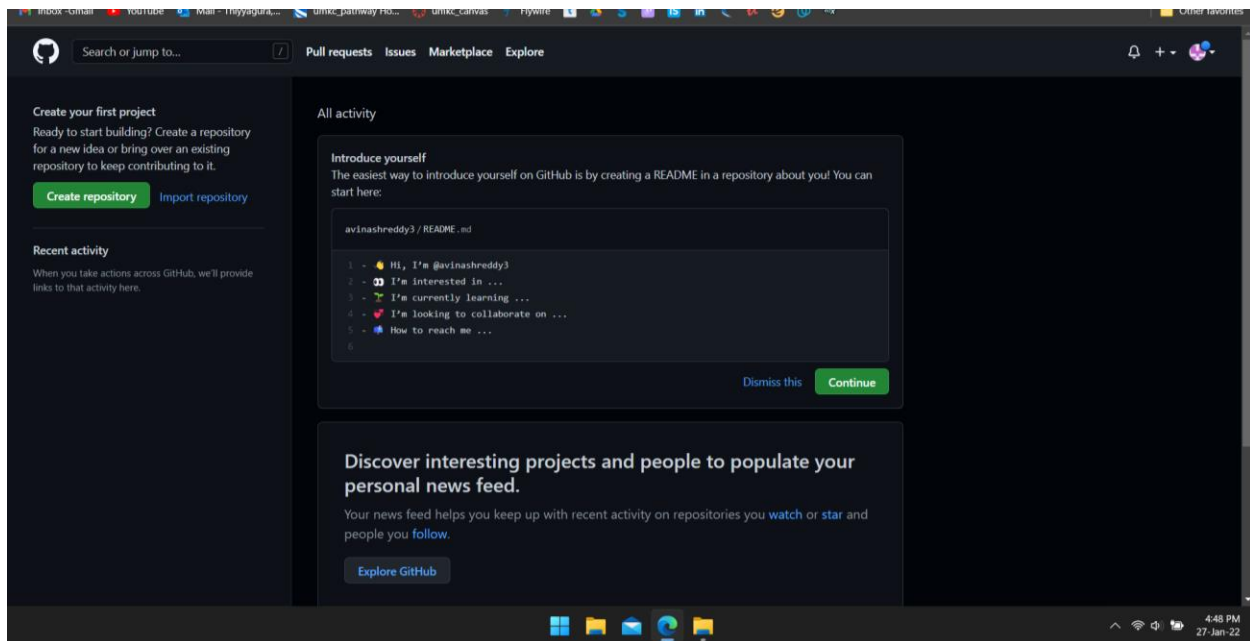




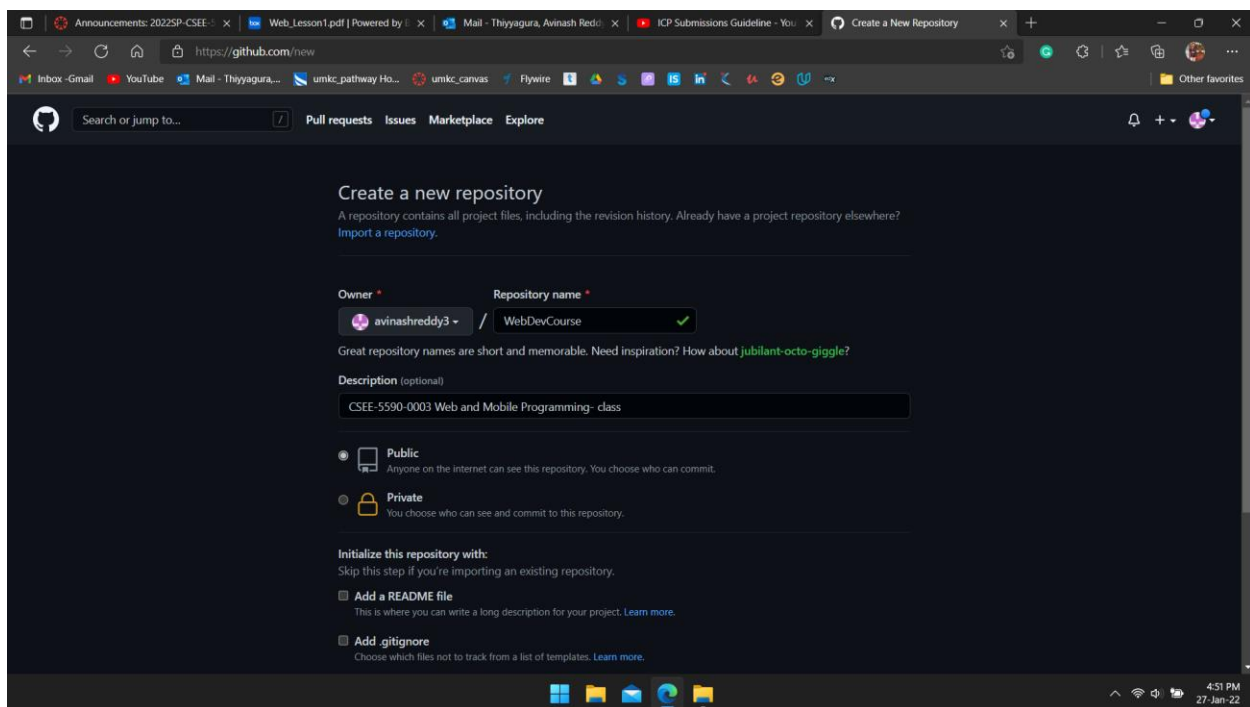
Created GitHub account.



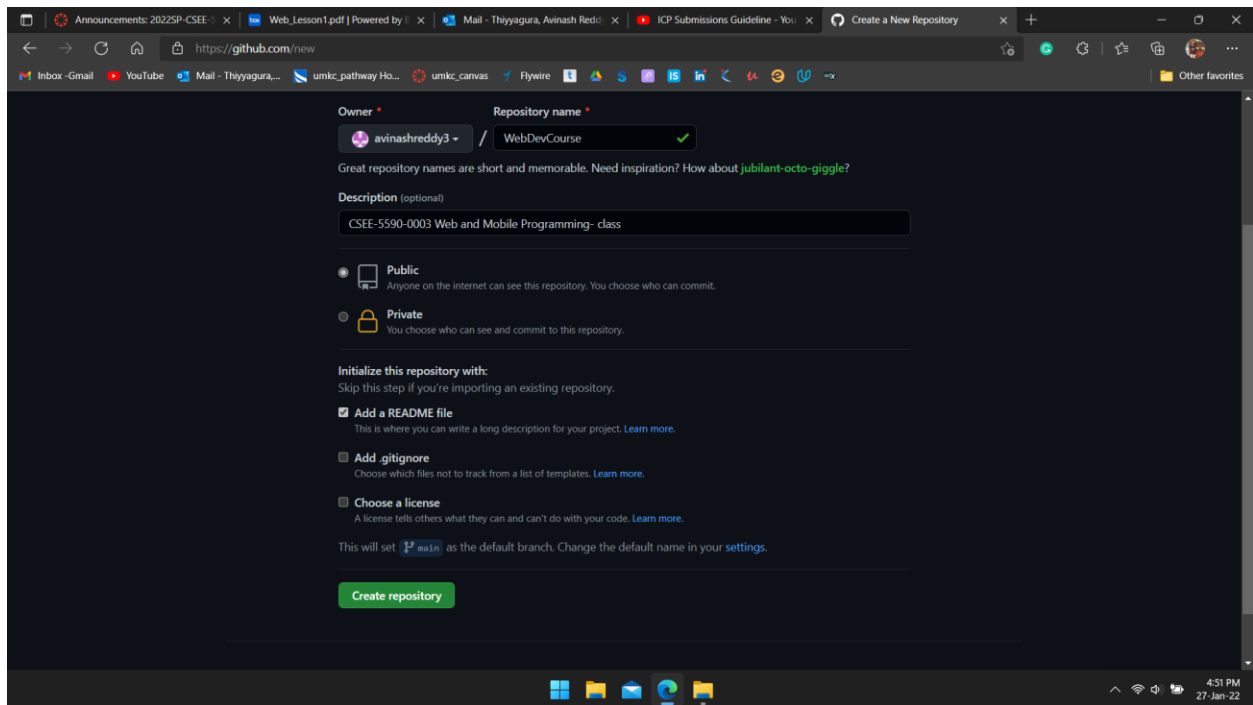
Creating a repository on the remote GitHub.



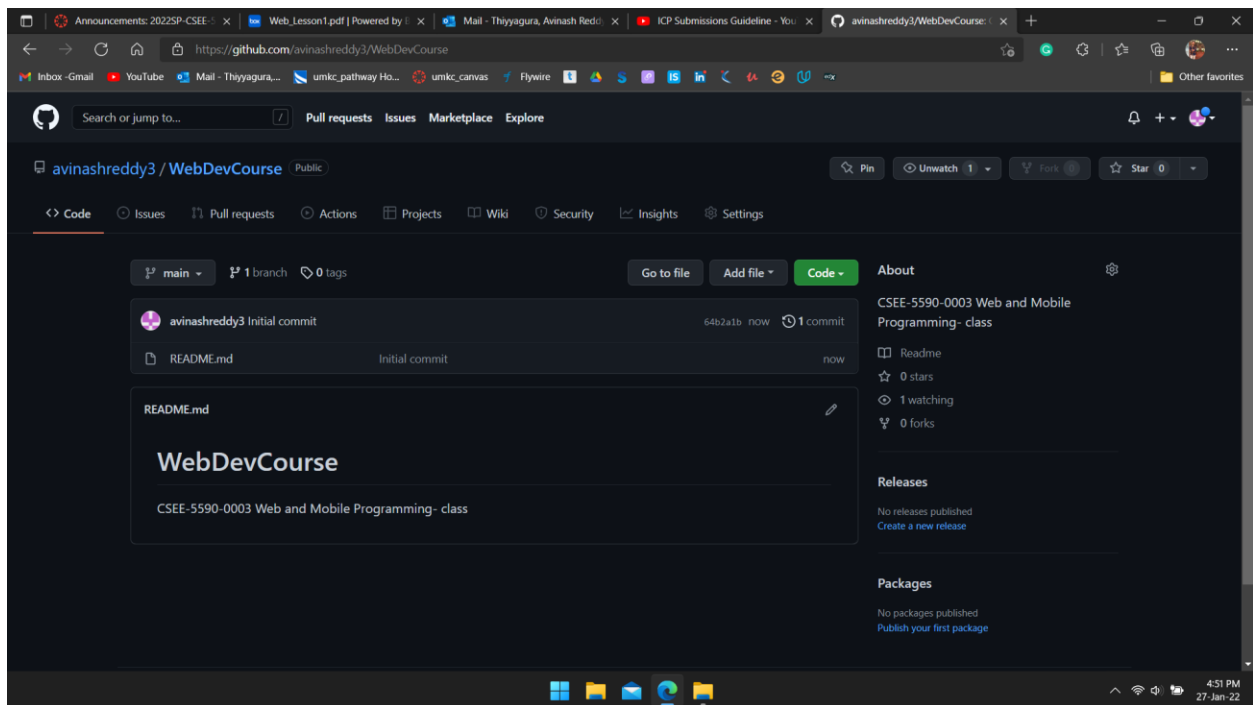
Give the repo name and its description if any and make the repo as public.



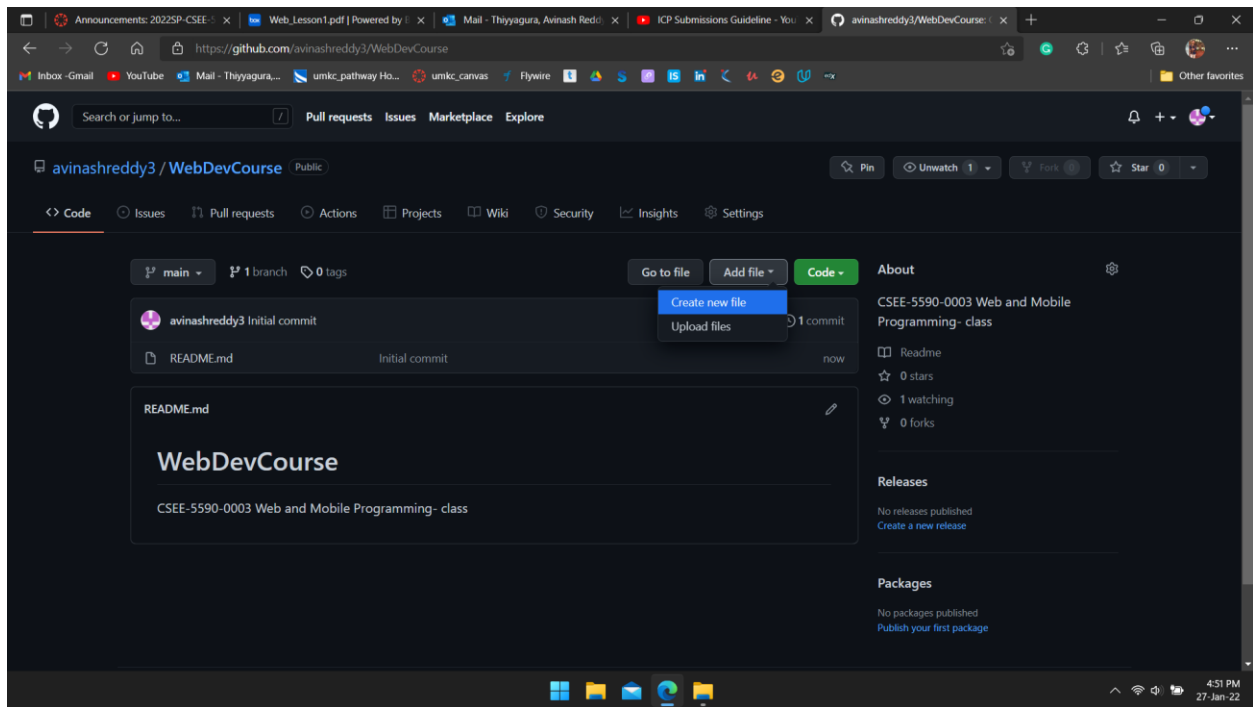
Next step in this process is to add a README file before creating repository.



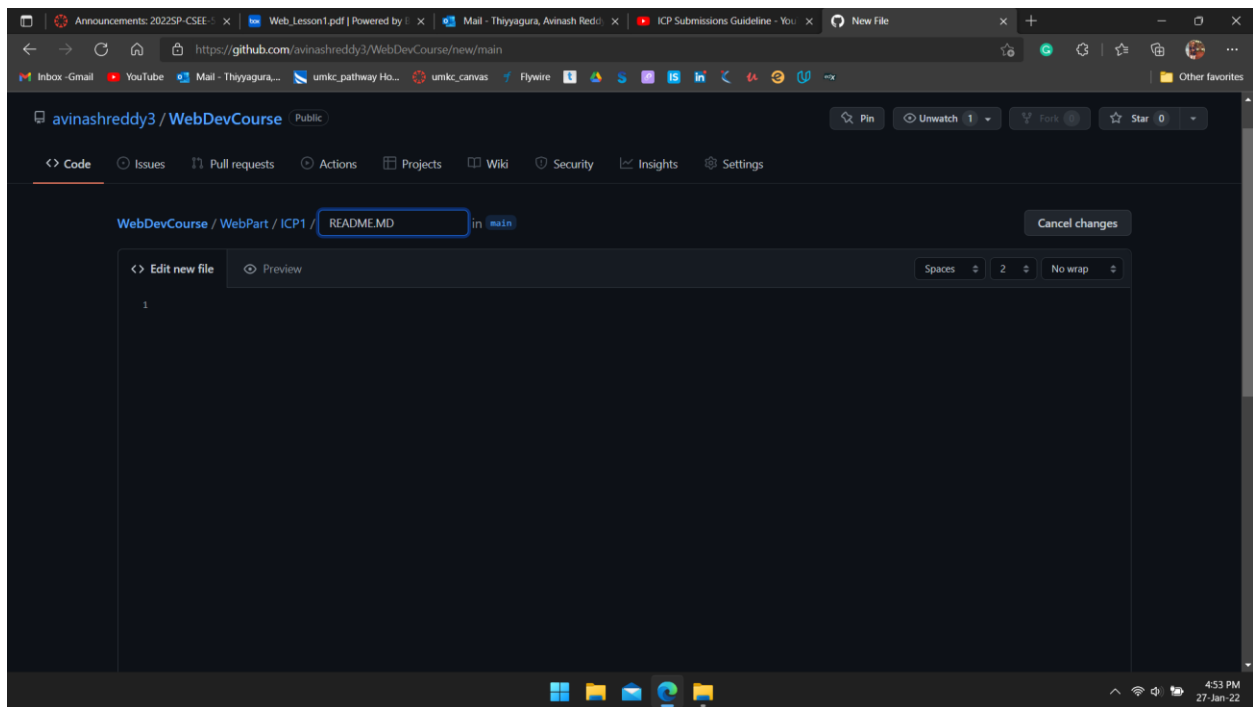
Repo created on remote GitHub platform.

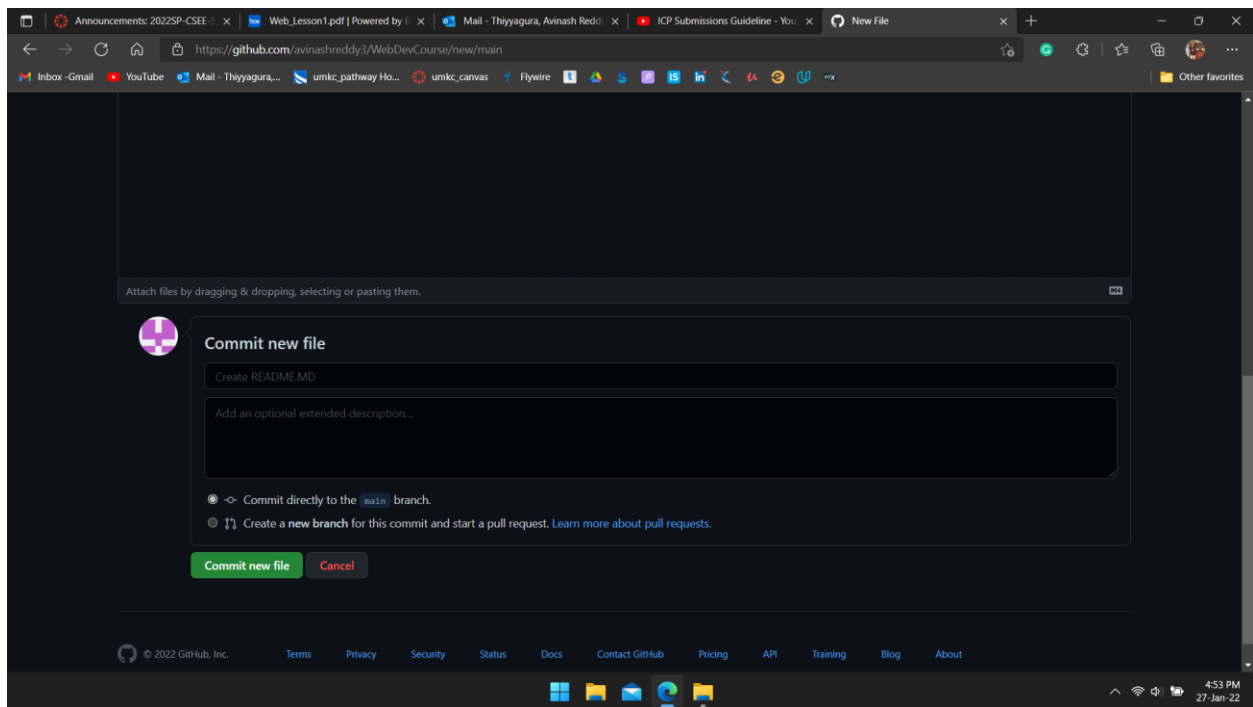


I have also created the folder for web part ICP with ICP1.

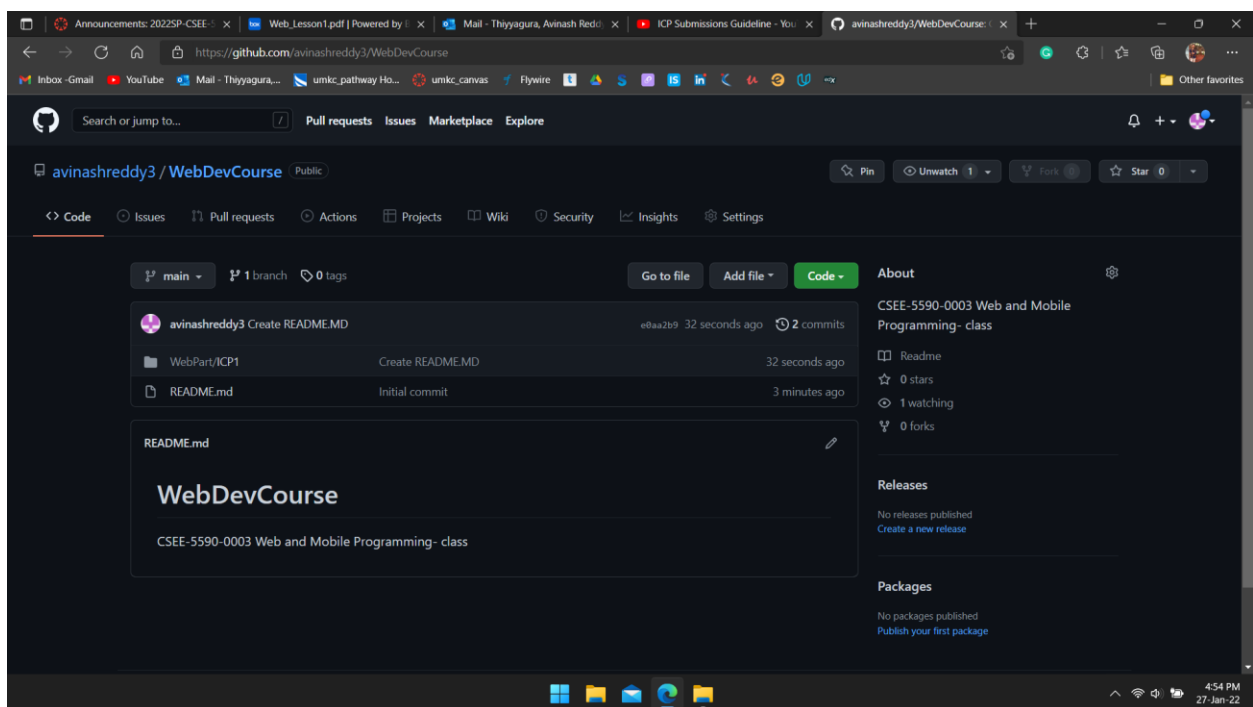


Create a README file for the ICP1 and commit the new file changes for file creation.





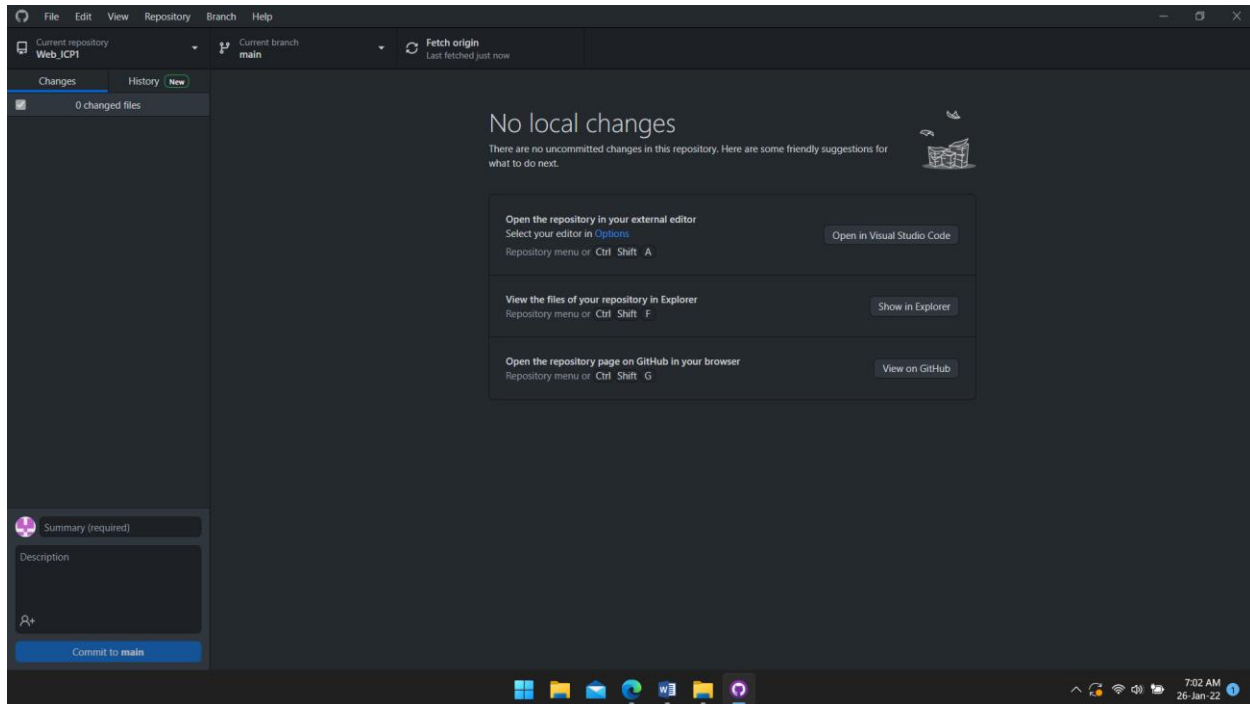
WebPart with ICP1 is created and below is the snap for the reference.



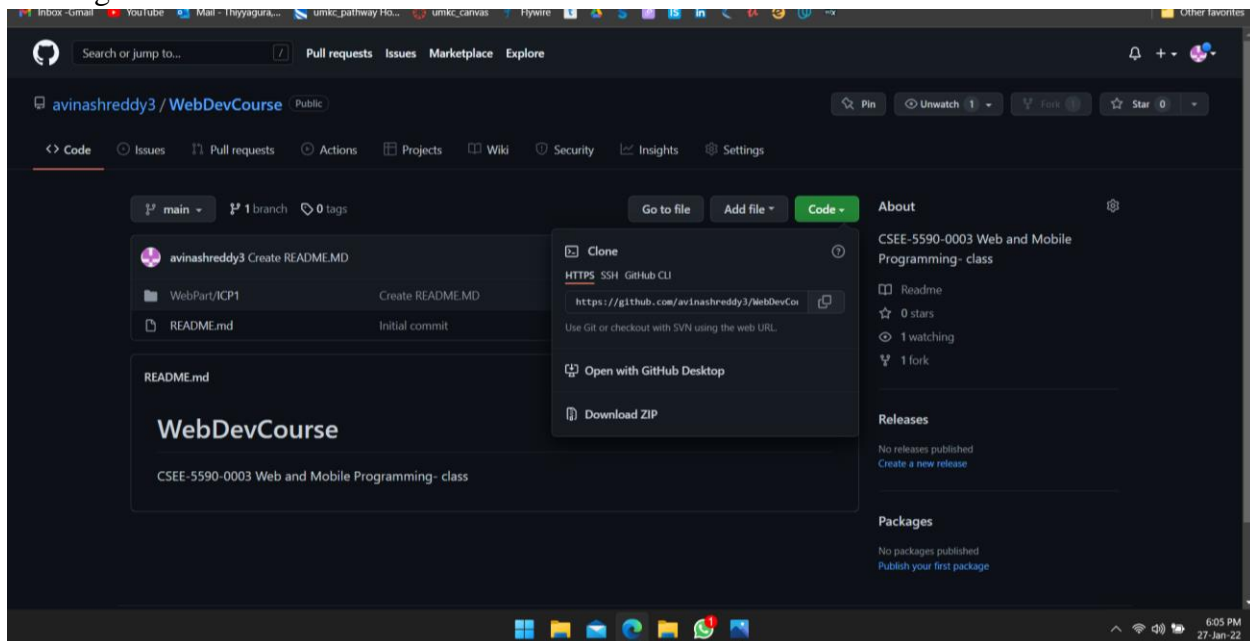
Clone the remote repository created above into the local machine using GitHub desktop application and two folders namely

- 1) Documentation (screenshots of a repository creation is stored)
- 2) Source (created a simple HTML document "index.html")

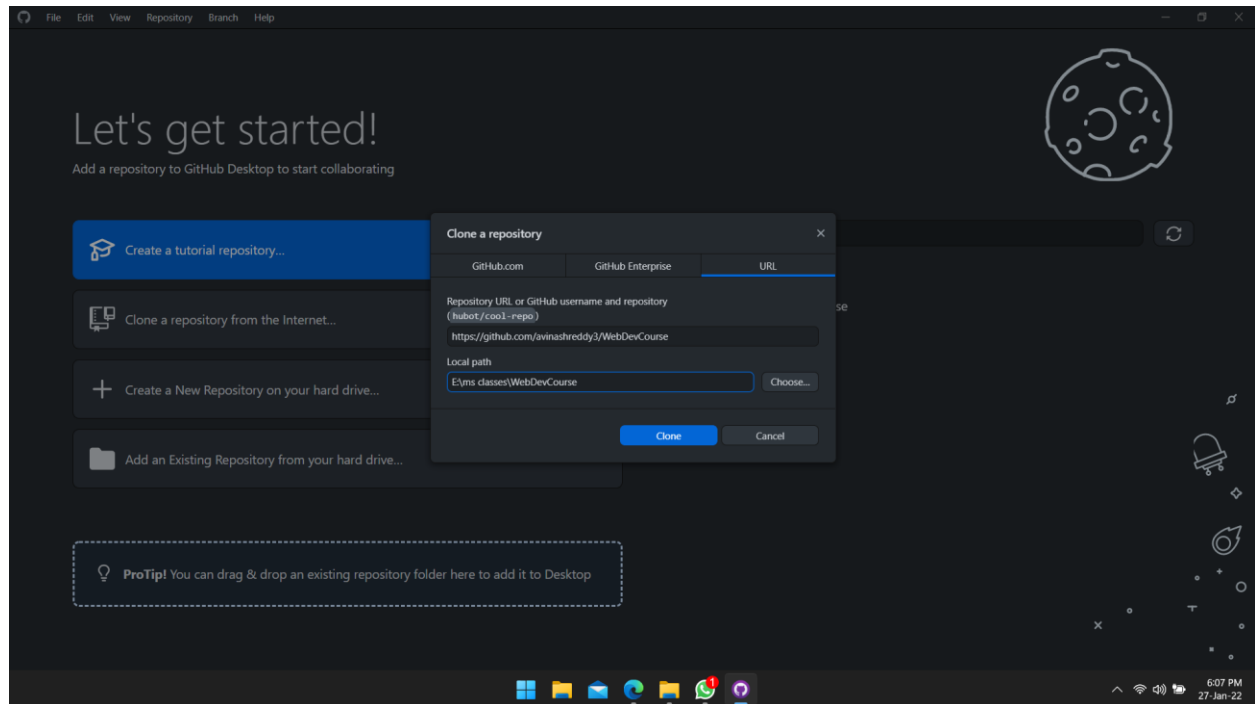
Screenshot below is the snap of GitHub desktop application on my local machine.



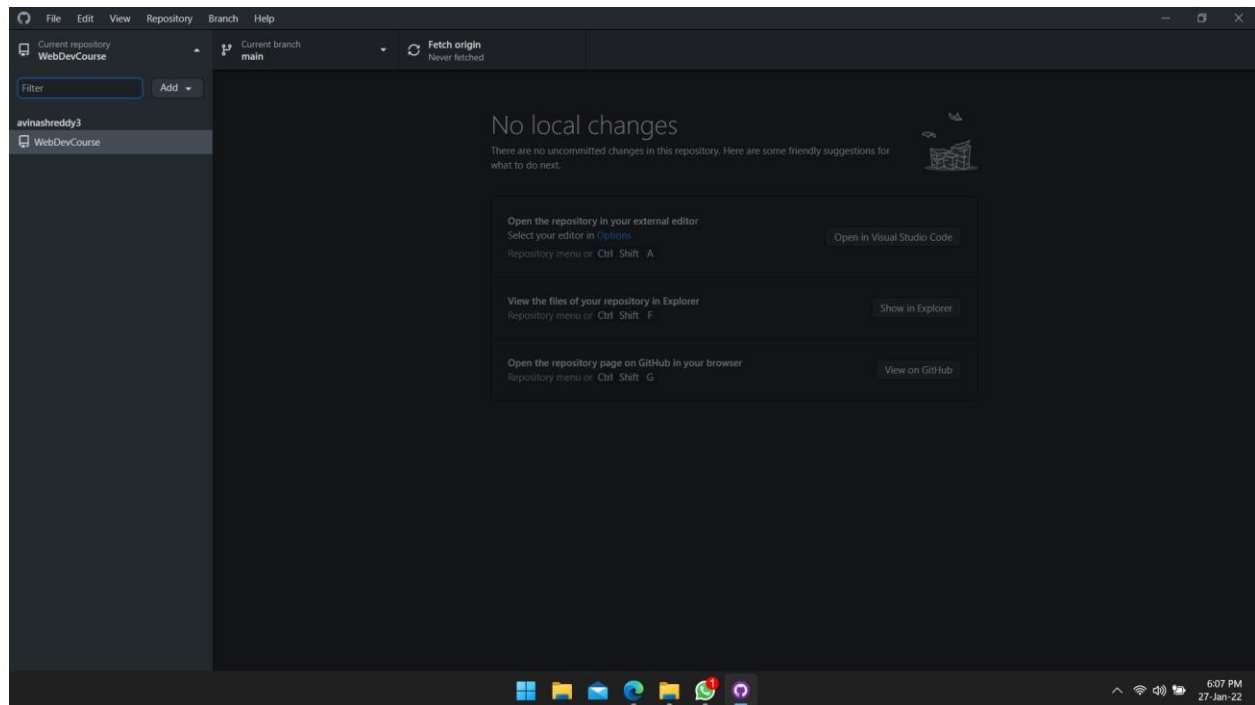
Cloning to local machine from the remote GitHub.



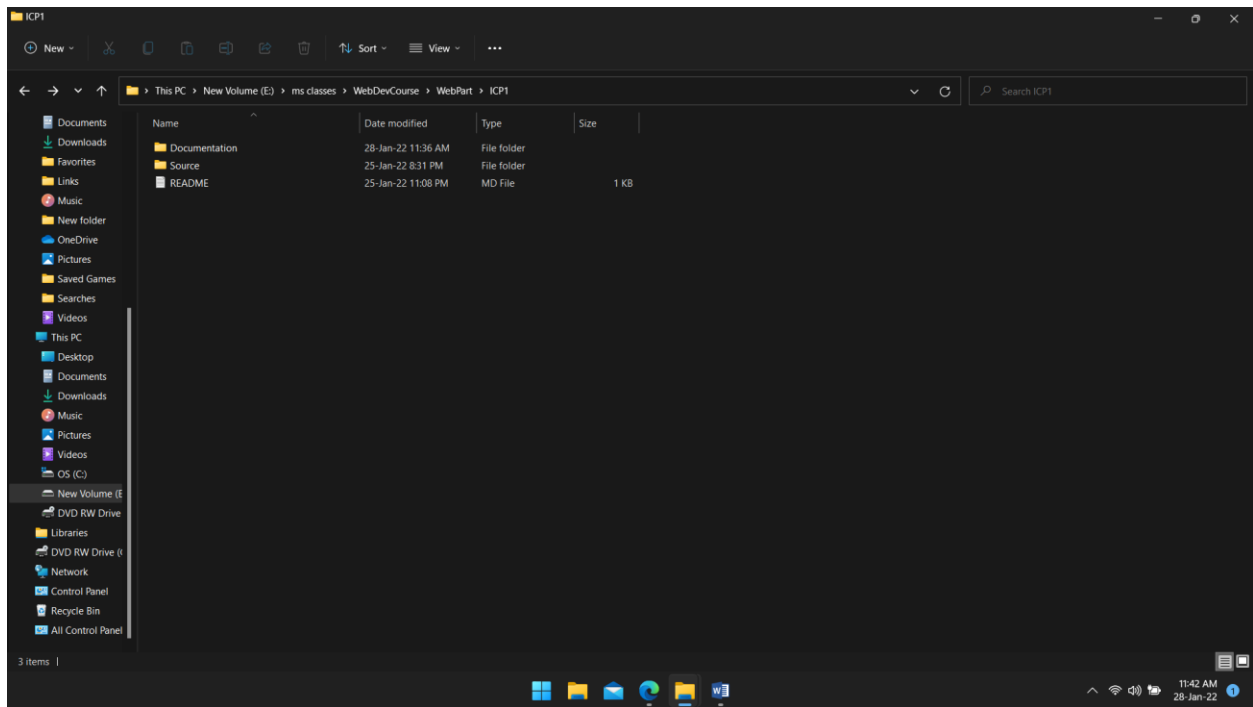
Give the path on local machine at which the cloned repository from the remote GitHub, had to be saved/ stored.



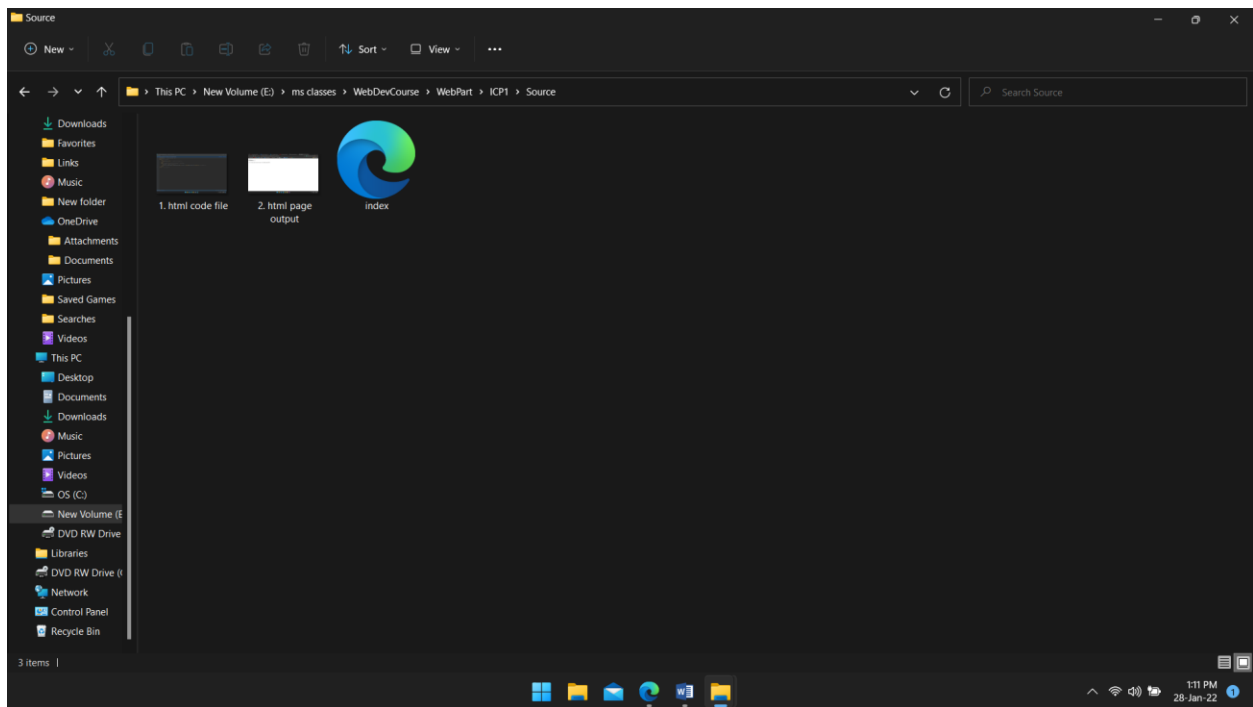
Successfully cloned as the local GitHub desktop repo.



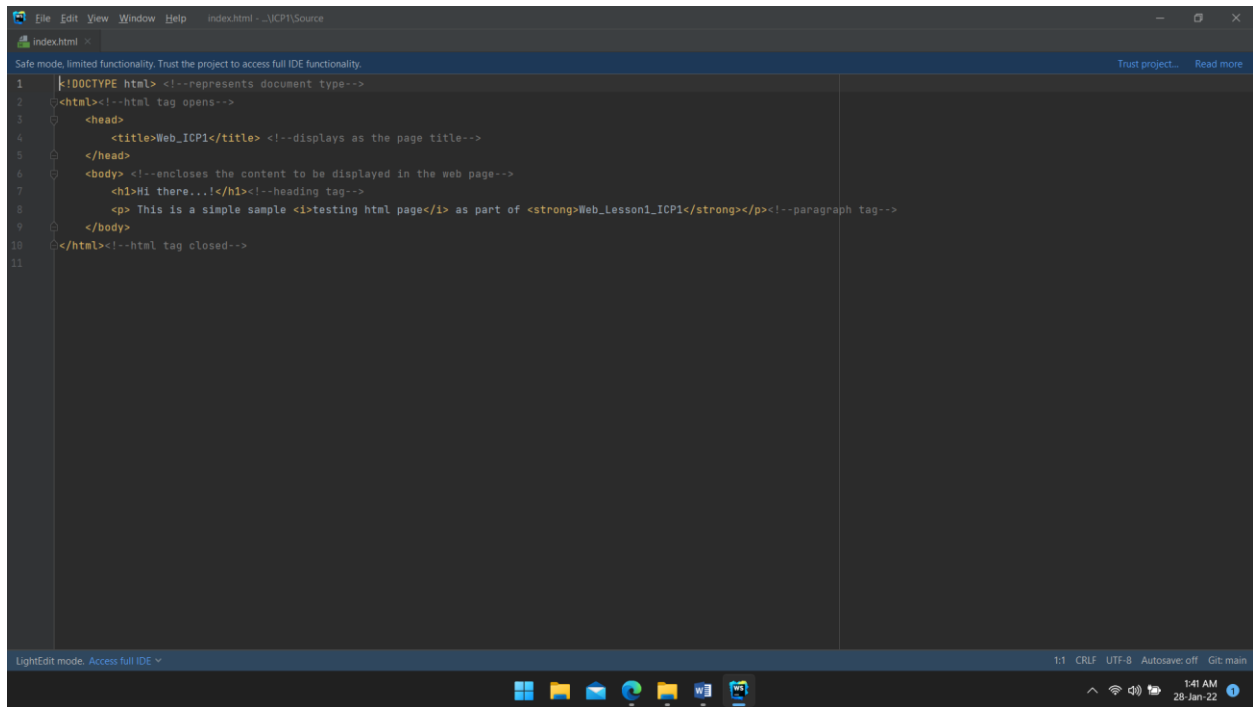
Created two files/ folders in local machine directory.



3. Create an HTML document named "index.html" and place it in the source folder

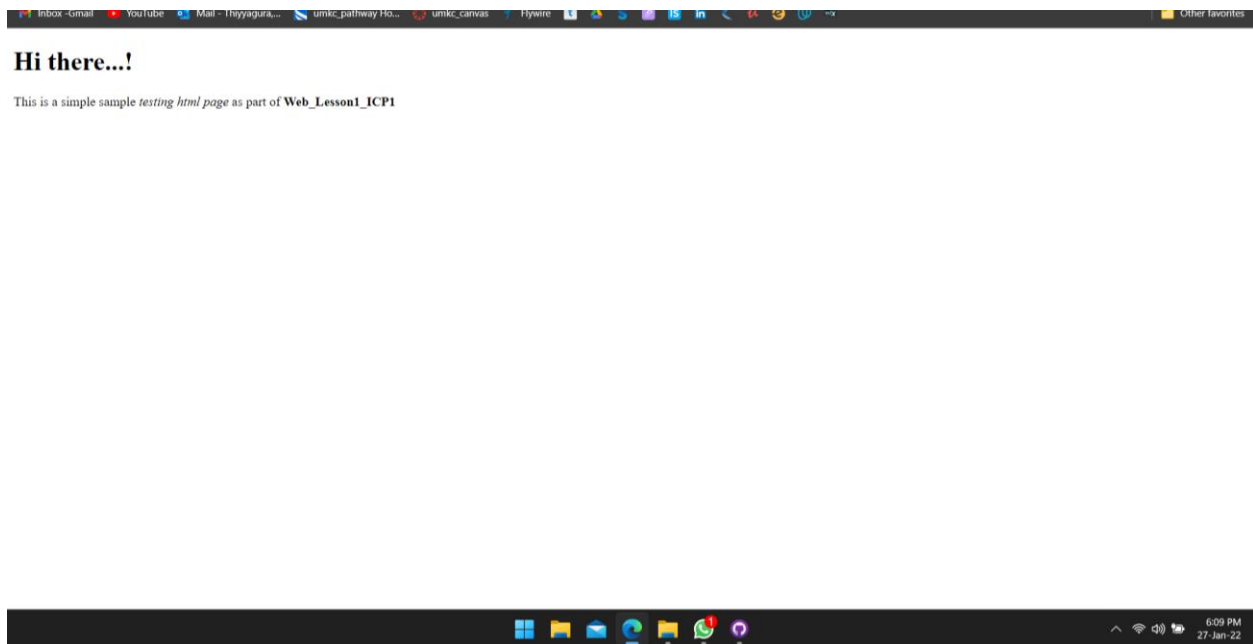


This is the development of index.html page in the local GitHub and I do make the commit with some comments on the html codes for the reference purpose.



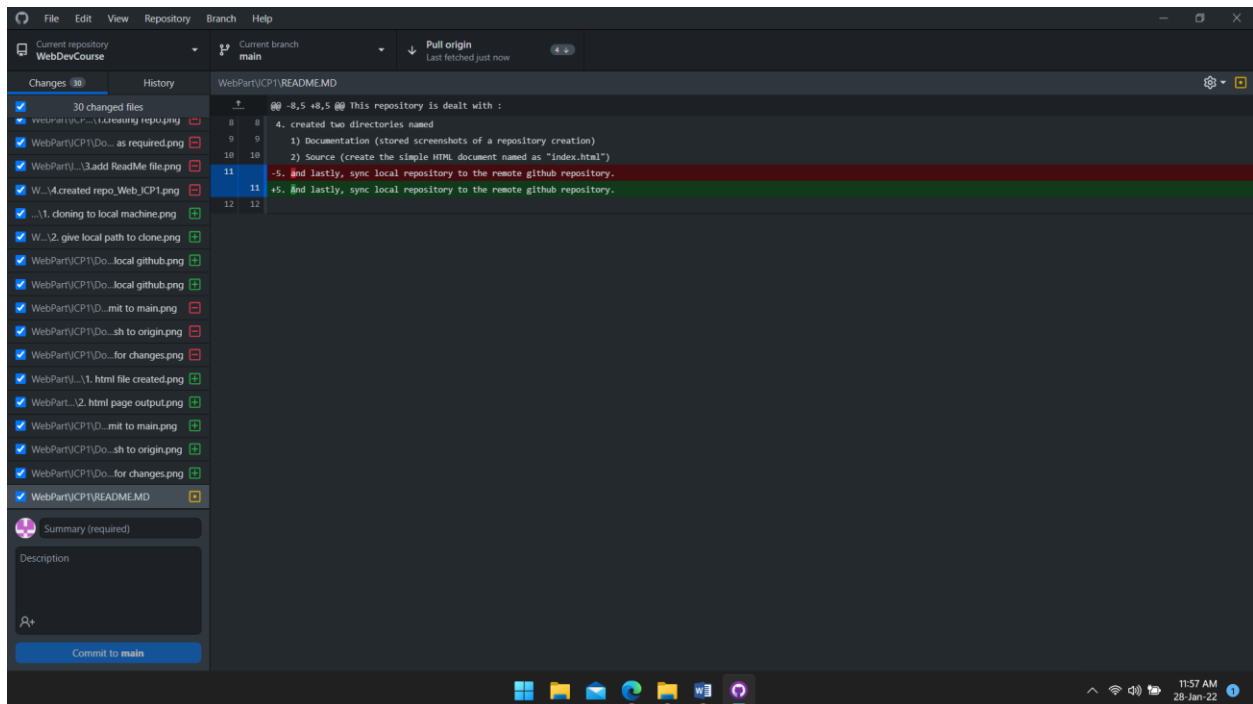
```
1  <!DOCTYPE html> <!--represents document type-->
2  <html><!--html tag opens-->
3  <head>
4  <title>Web_ICP1</title> <!--displays as the page title-->
5  </head>
6  <body> <!--encloses the content to be displayed in the web page-->
7  <h1>Hi there...!</h1><!--heading tag-->
8  <p> This is a simple sample <i>testing html page</i> as part of <strong>Web_Lesson1_ICP1</strong></p><!--paragraph tag-->
9  </body>
10 </html><!--html tag closed-->
11
```

This is the snap below of the web page output for the html document created.

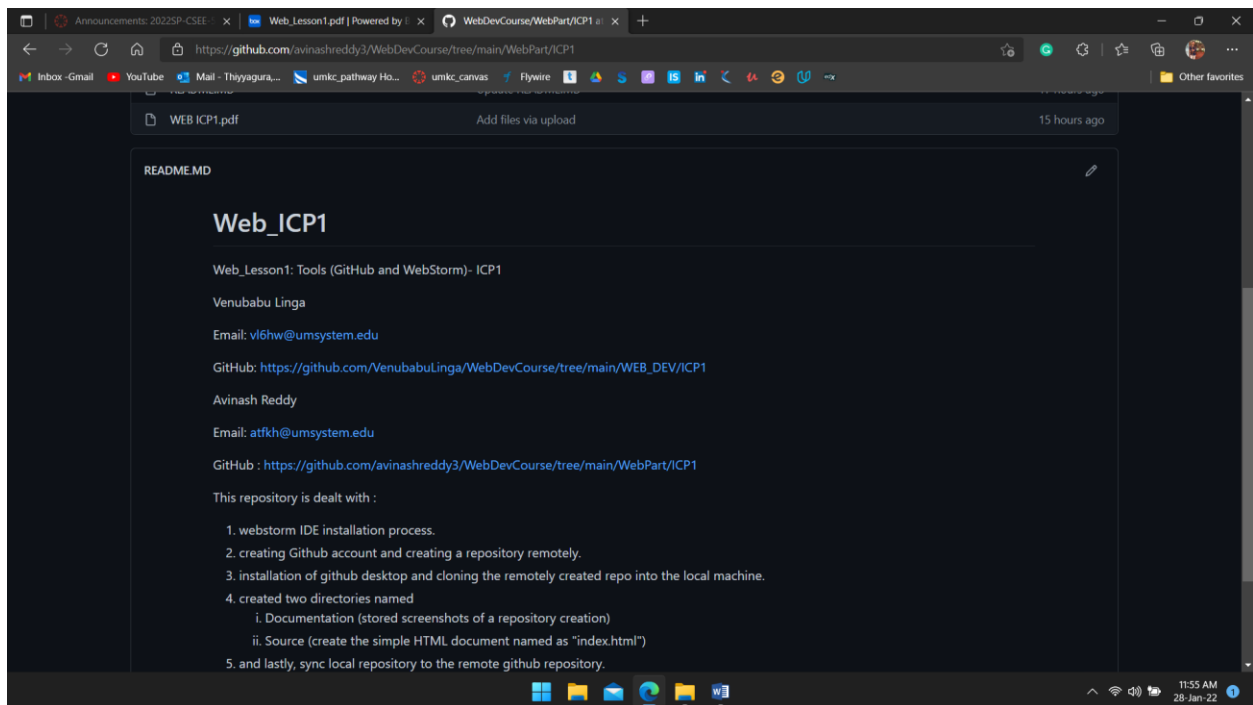


4. Write about your repository in README.MD file

Below is a snap of readme file on local GitHub desktop application.

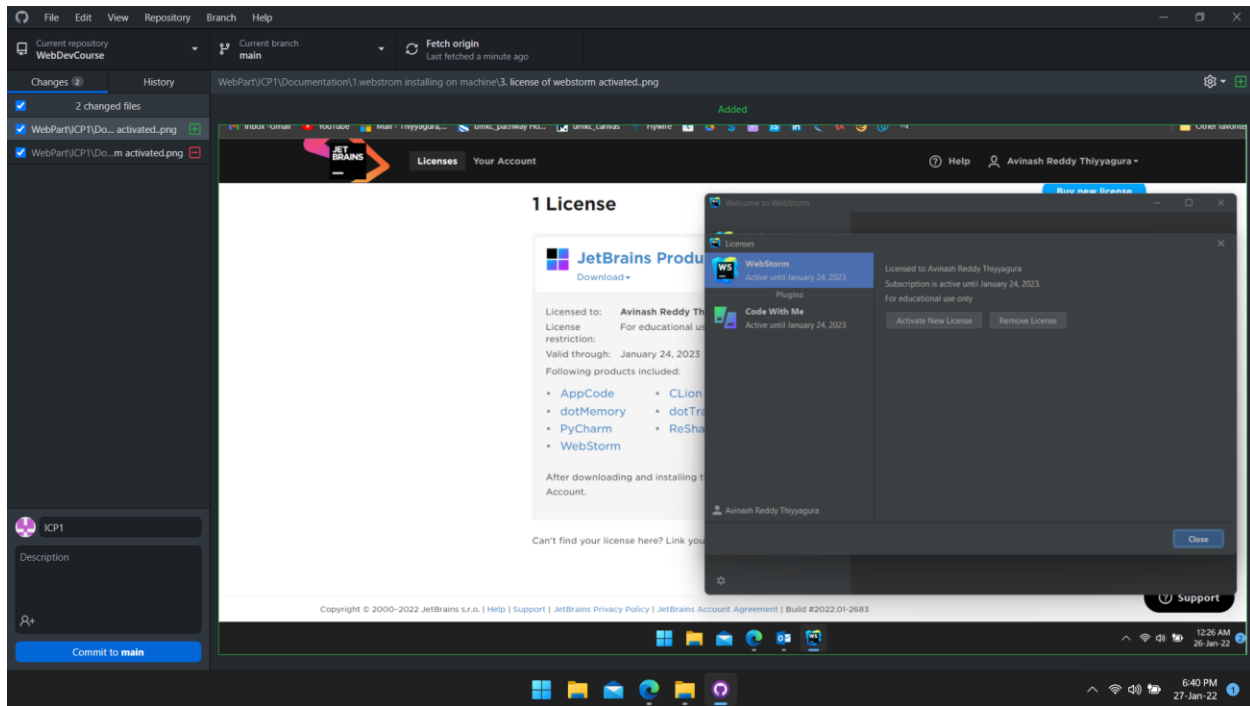


Below is a snap of readme file on remote GitHub repository.

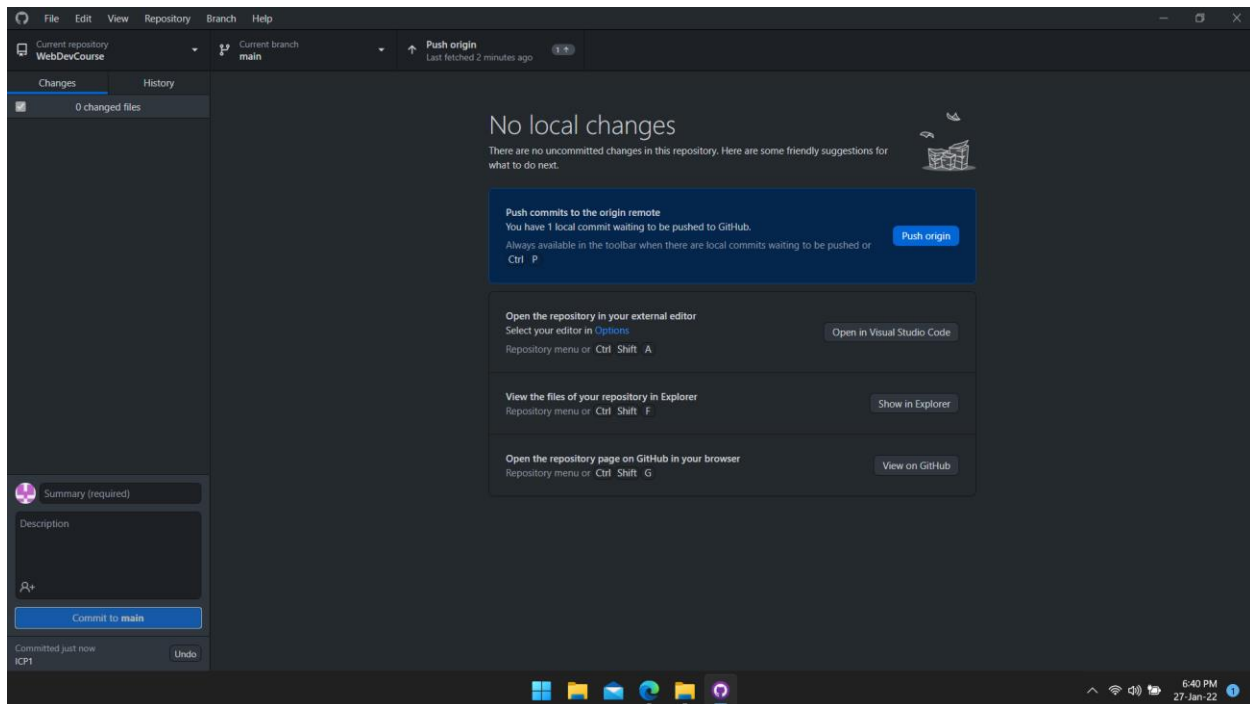


Syncing to local repo and remote GitHub repo.

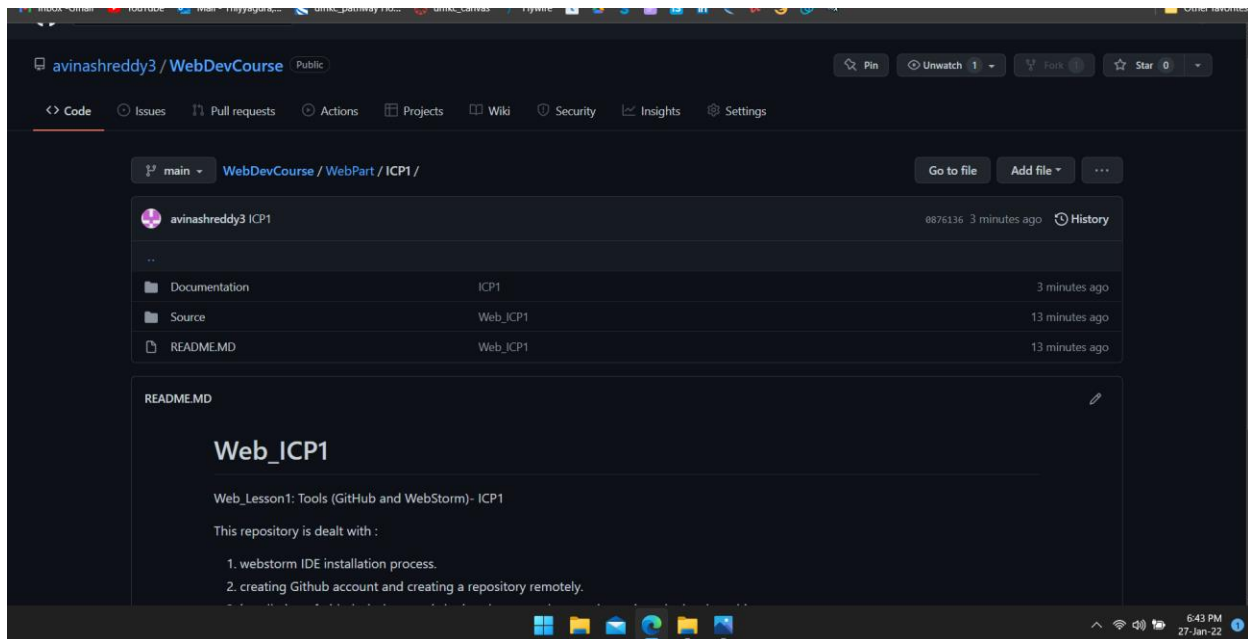
After the necessary changes and updates to the files, codes, documents commit the all them to main repo. This is shown in the below snap.



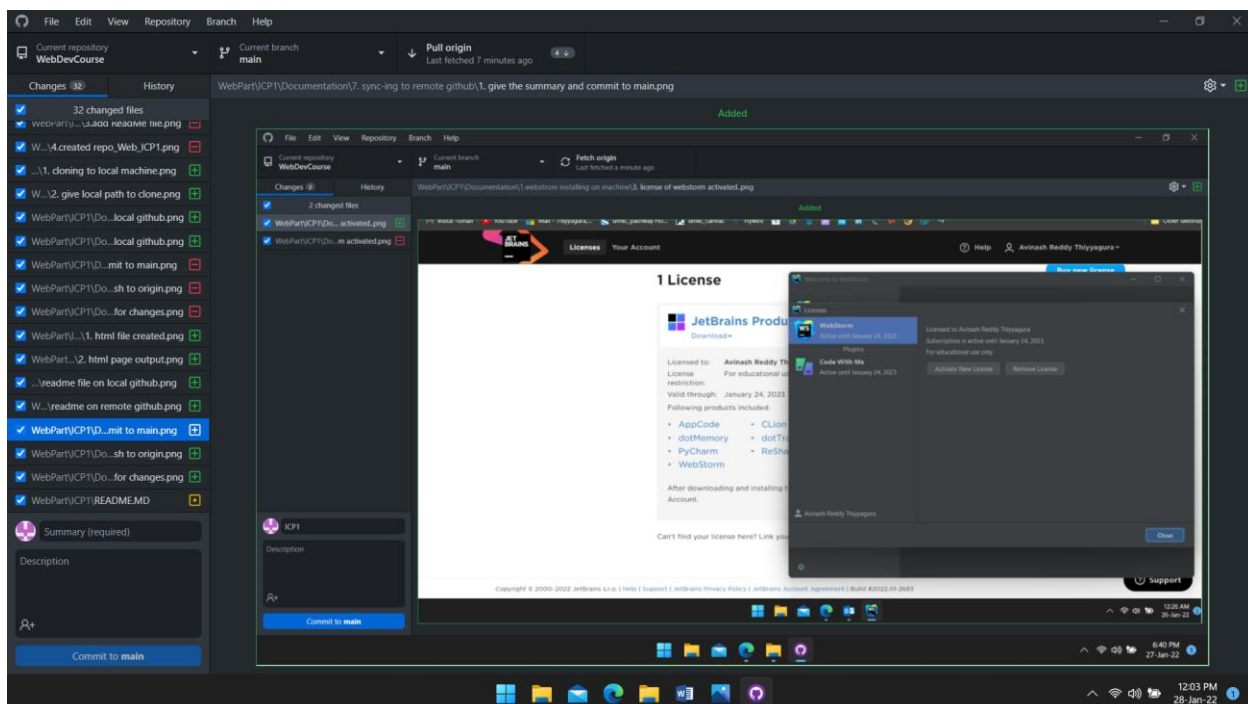
After committing to the main repo, push it to the origin for the changes and updates to take place on the remote GitHub.



I have also pushed the updated git repository to the web GitHub The updated index .html was successfully pushed to the web GitHub account

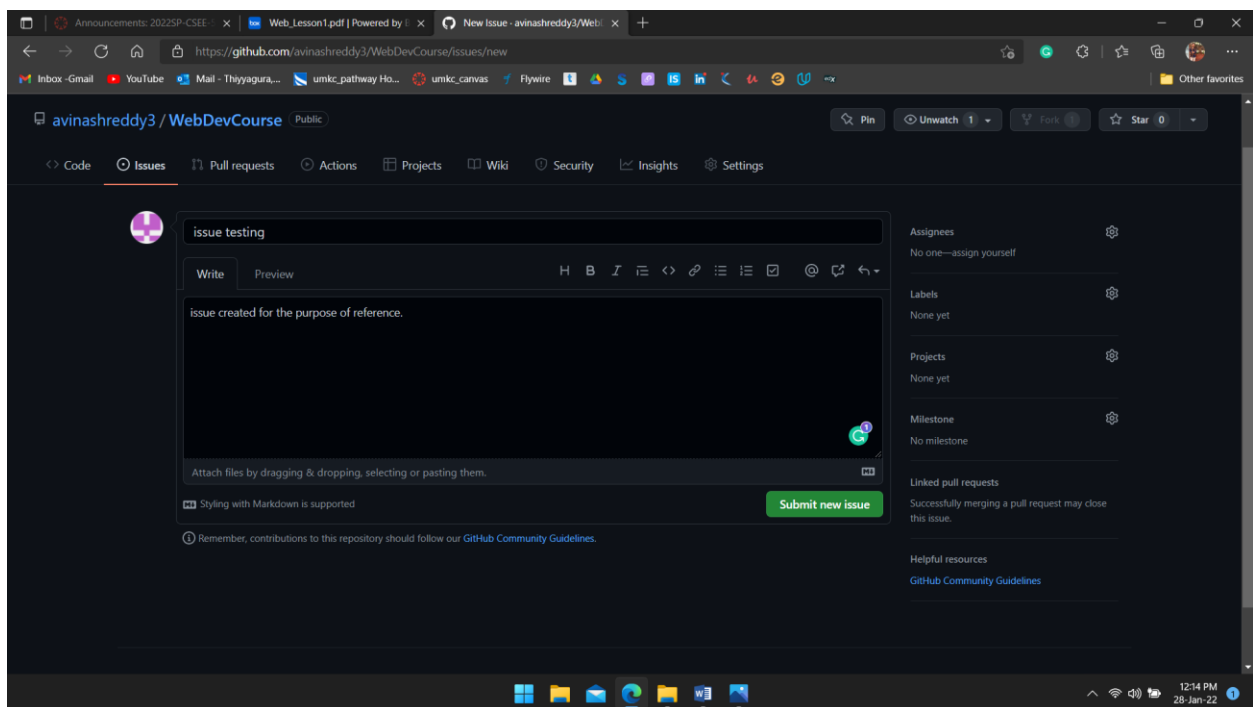
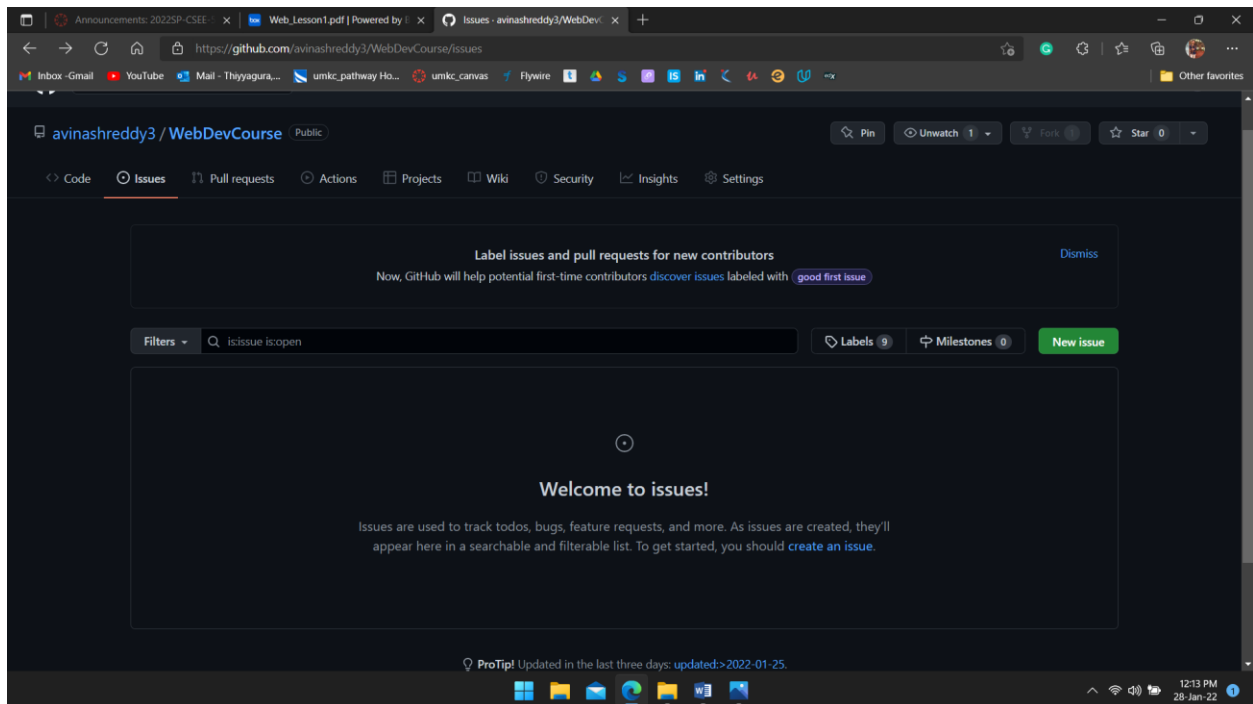


Created the (Source and Documentation) directories in local GitHub and the screenshots of a repository creation were placed in the documentation folder in the local repository, and sync it to the remote repository.

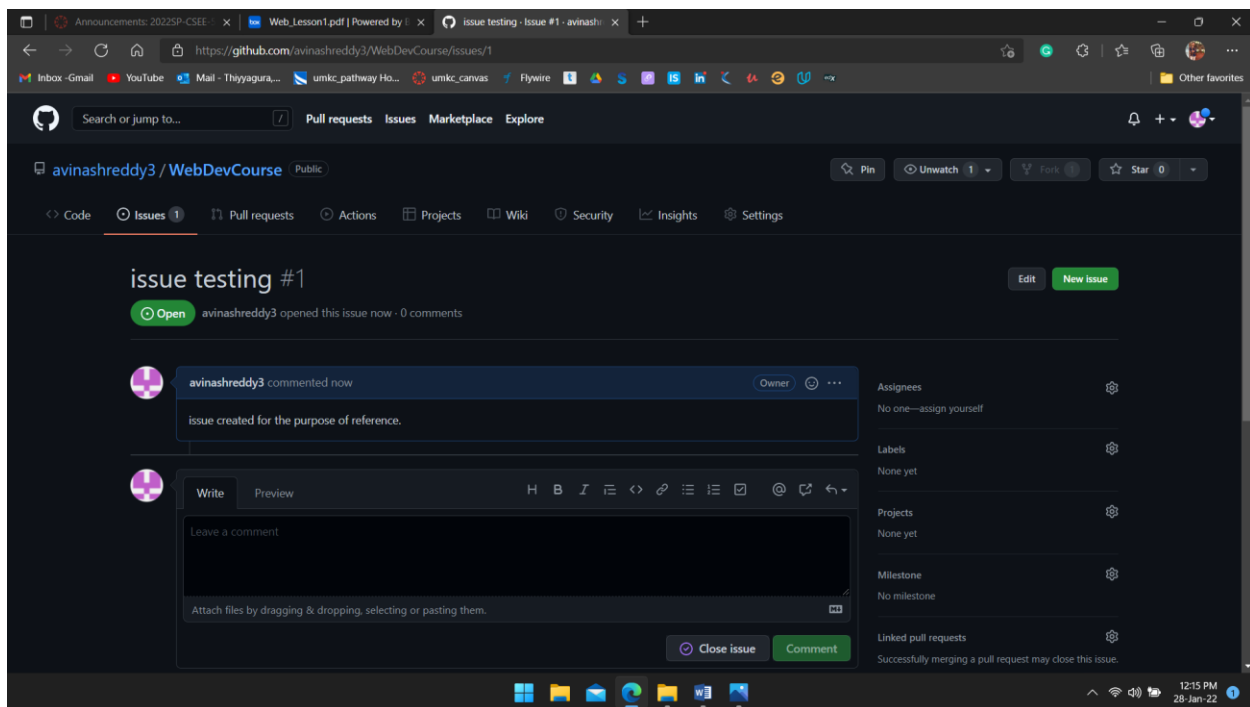


5. Create an issue and described your pull request

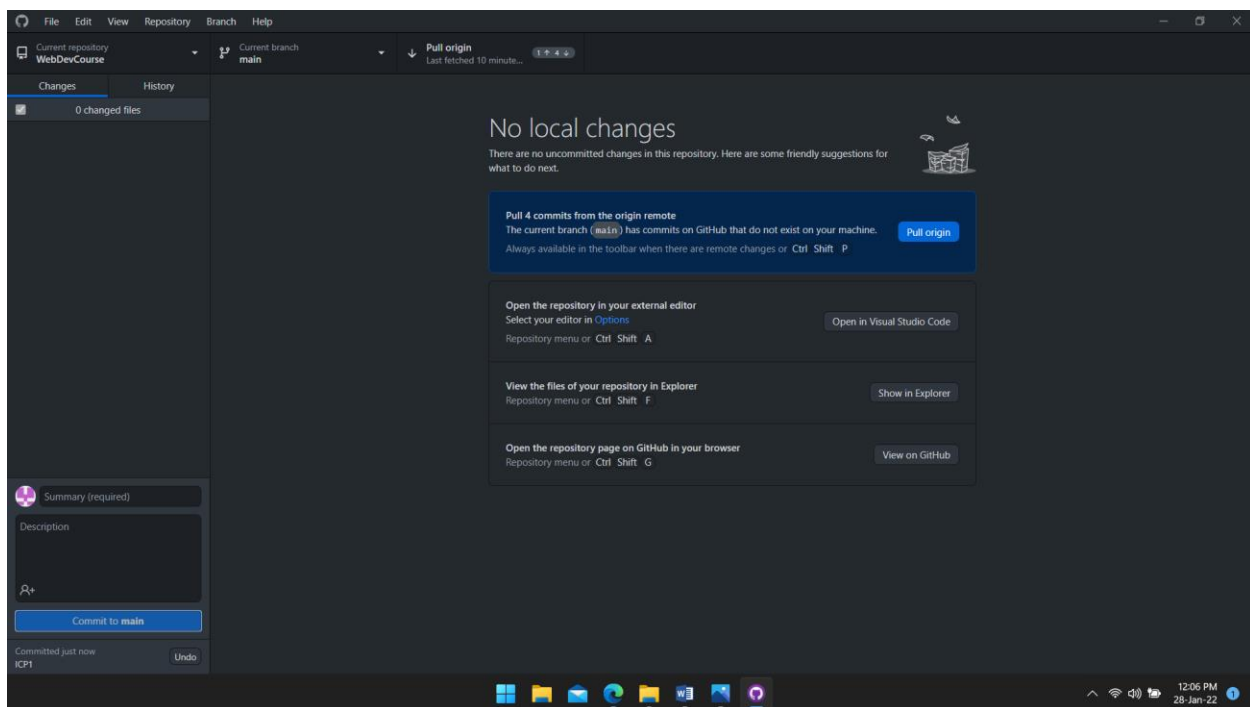
Created an issue in my repository and made a pull request. From the local GitHub.



Sample Issue created for reference and made a pull request.



Pull request on local machine for reference. Use pull origin option to pull the issue and changes into the local GitHub machine.



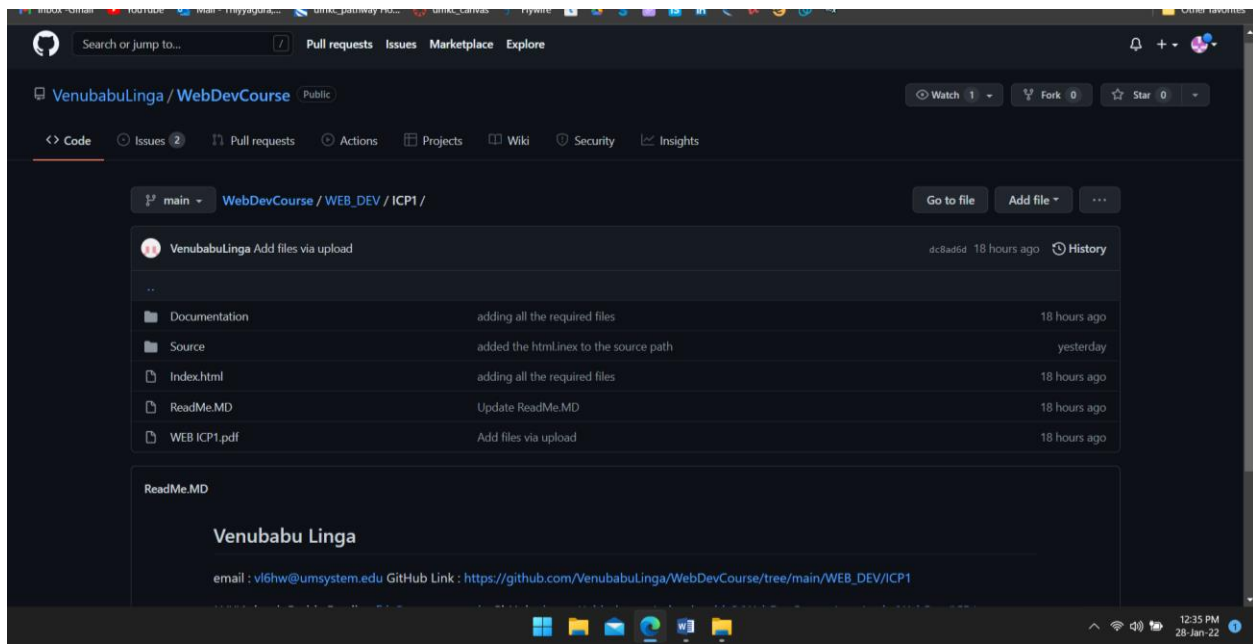
6. Fork any existing repository

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

We can use a pull request to suggest changes from user-owned fork to the original repository, also known as the upstream repository.

We can bring changes from the upstream repository to your local fork by synchronizing your fork with the upstream repository.

Here is snap where I forked my partner repository VenubabuLinga repo as below.



7. Conclusion

In this ICP1, I have learnt working on the IDE Tool called WebStorm and GitHub. And also the cloning of the code from local to web GitHub. I do have an insight about pull & push requests, sharing of issues, codes and discussions and collaborating with the team members.