GIS Application

NCGSA Internship – week 4

8/8/22 | IST NAV App development | Mam Aneeqa

Kiran Sardar | 200901064

Syeda Farwa Rizvi | 200901098

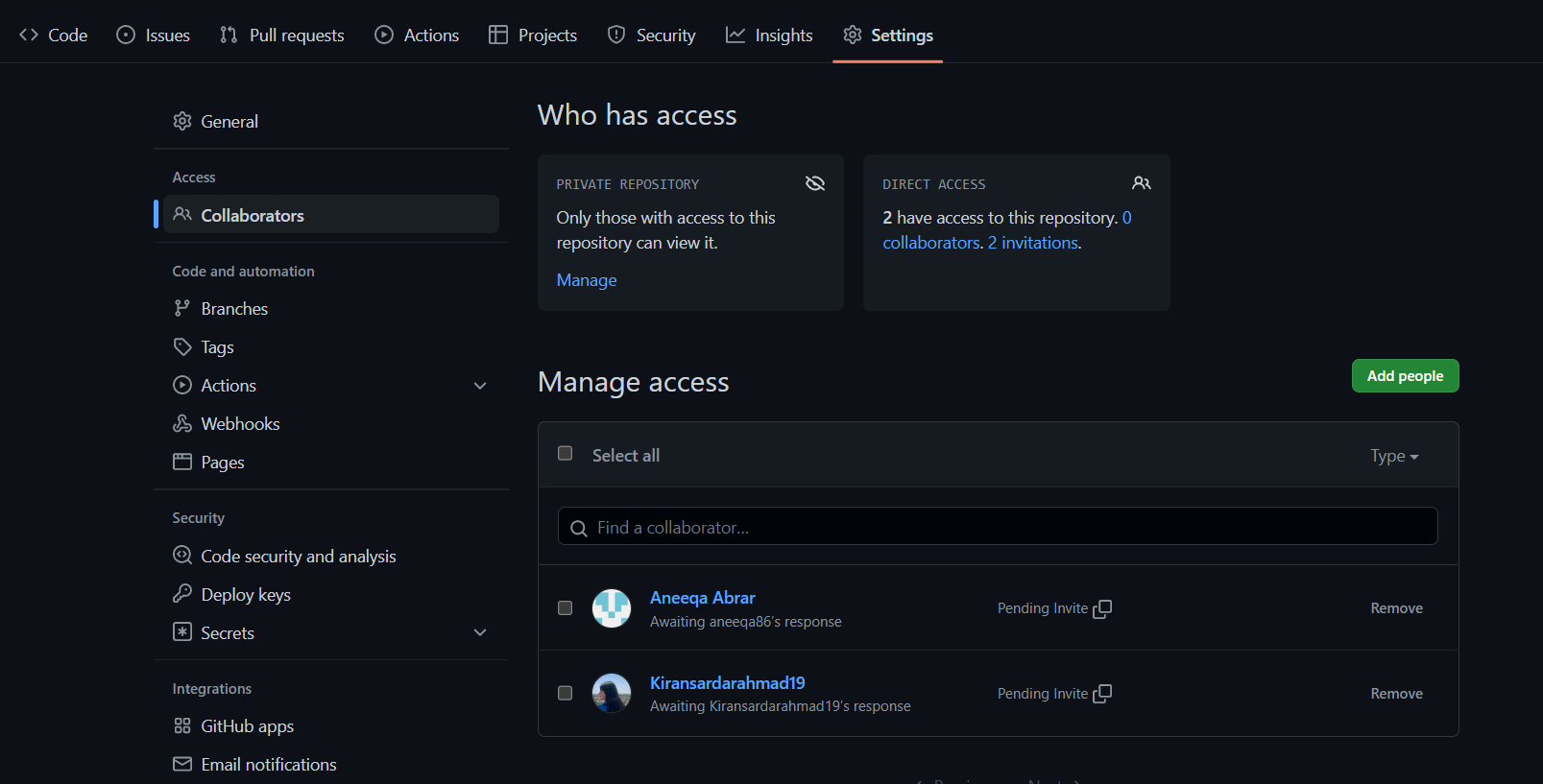
# Literature Review

In the 4th week, we Set up our Git Hub repository for developing and updating code simultaneously.

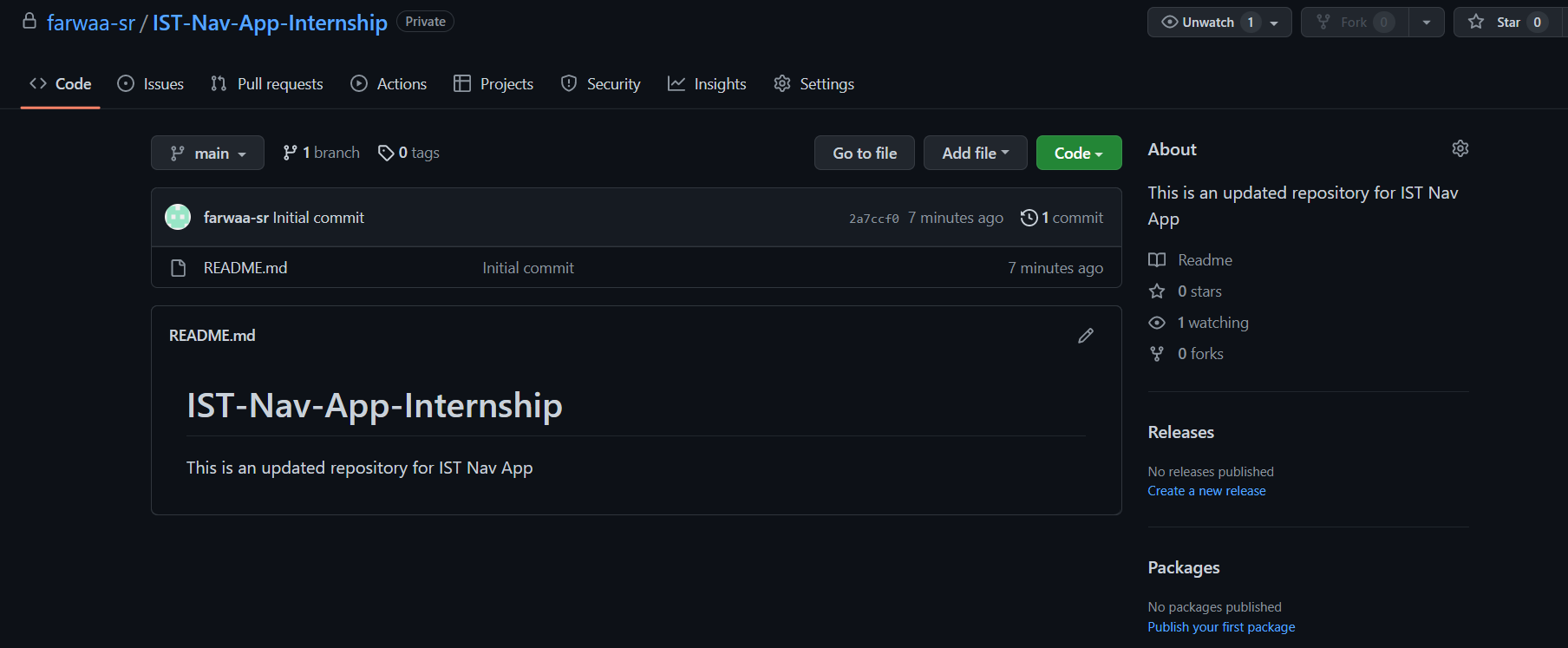
Leaflet was self-taught to learn how to implement the map onto IST Nav. The libraries required, how to edit and add the map on an individual module.

## Git hub setup

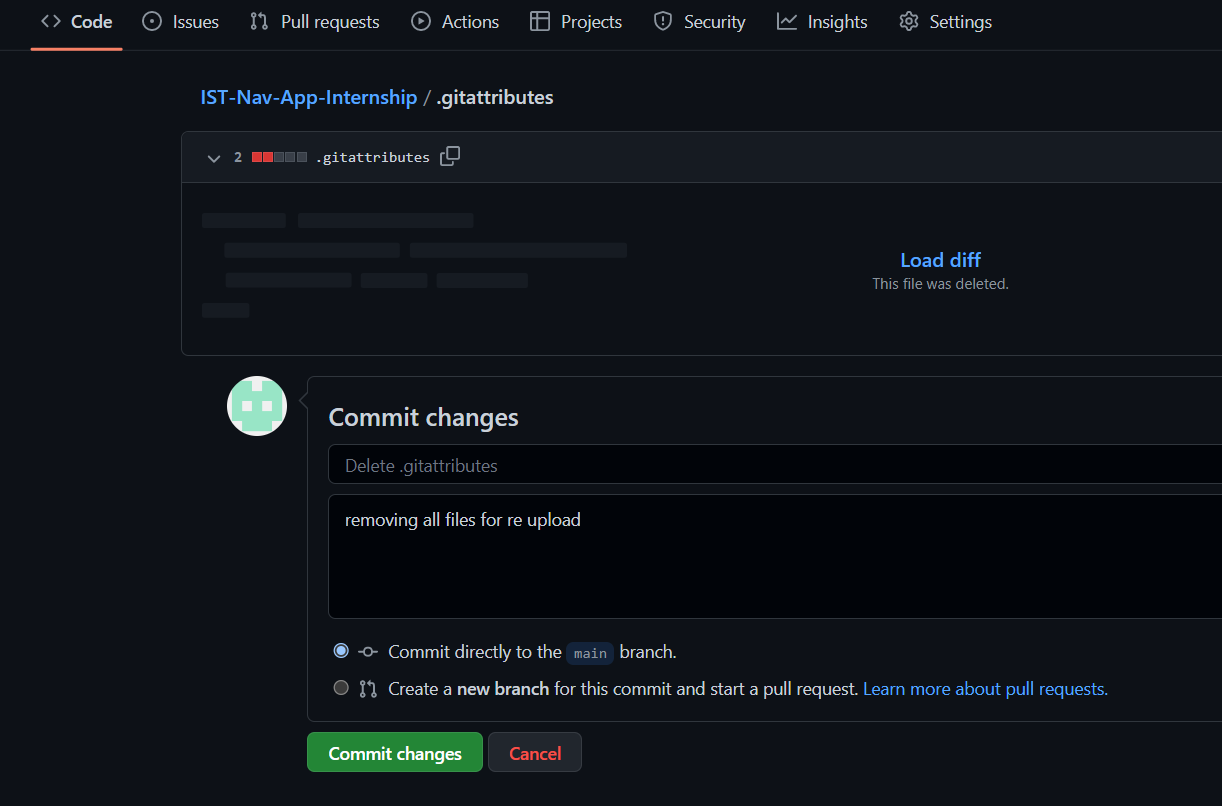
### After Creating a new Repository, Add respective members as collaborators so they have access to code as well.



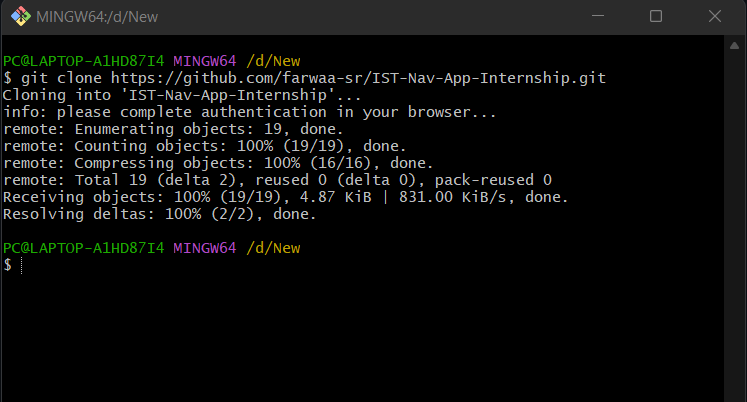
### An initially empty repository:



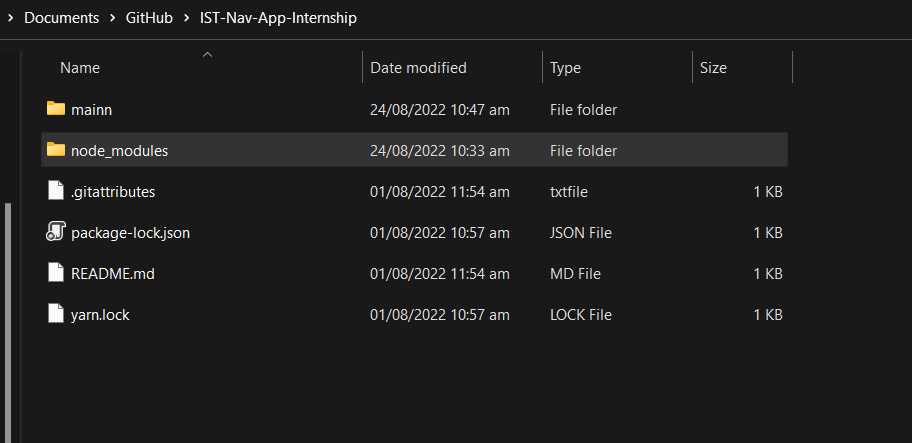
### The Code so far uploaded was unable to be set up as the files were too large therefore worked on: <https://git-lfs.github.com/> LFS Git hub and git desktop



### Git Bash Procedure initialized



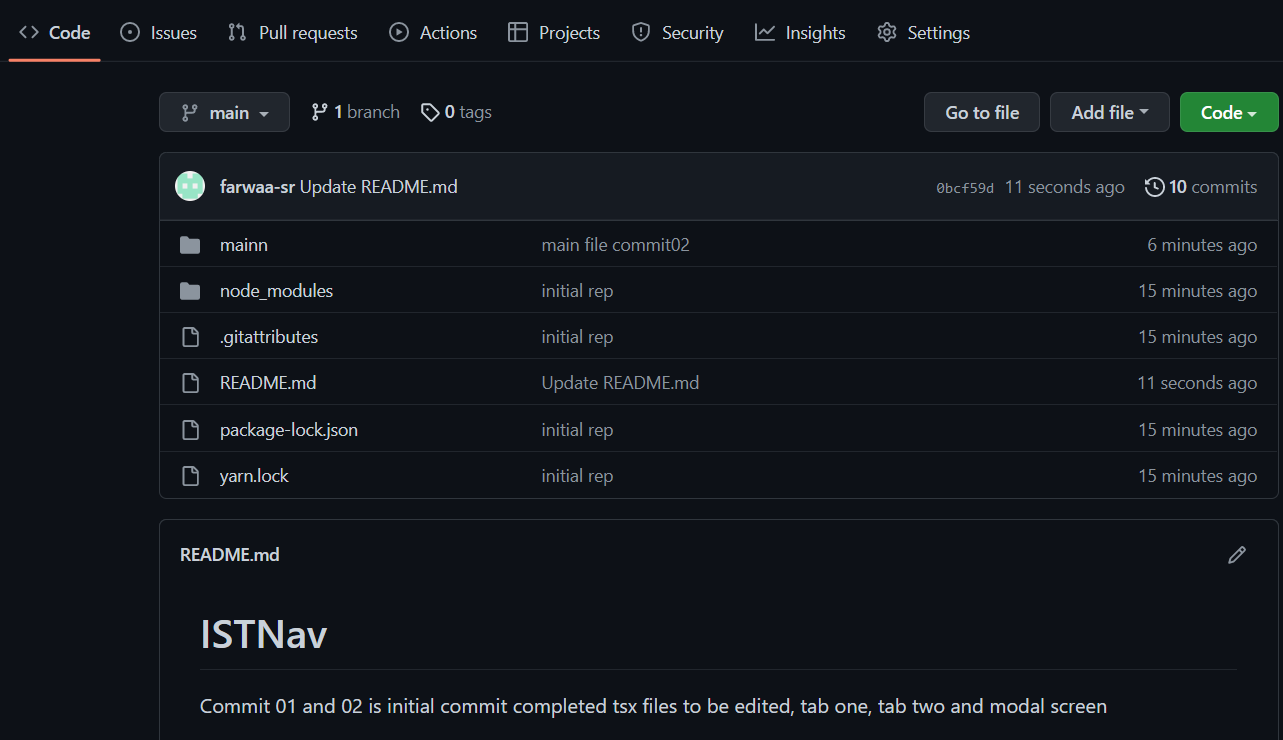
### Initial commit complete:



Graphical user interface, text, application

Description automatically generated

) Commit 01 and 02



) Final commited git

Graphical user interface, application

Description automatically generated

## React leaflet

<https://react-leaflet.js.org/>

### Core concepts

React Leaflet makes it possible for React and Leaflet to work together. It doesn't replace Leaflet, but it uses Leaflet to make React components out of Leaflet layers. Because of this, it can act differently than other React components, such as:

### DOM (Doc Object Model) rendering

Leaflet layers are drawn to the DOM, not by React. This is done by Leaflet on its own. When providing the MapContainer element and the components of UI layers components, React only renders a <div> element.

### Component properties

When an element is rendered for the first time, the characteristics passed to it are used to generate the relevant Leaflet instance. By default, these properties should not be changed.

During the initial render, all of these properties should work as they do in Leaflet. However, if they change, they won't be reflected in the UI unless it says so in the documentation.

Changes to mutable properties are compared by reference (unless otherwise stated) and are made by calling the appropriate method on the Leaflet component example.

### Leaflet elements references

Unless otherwise stated, all React Leaflet-exported components can use refs to show the creation Leaflet element incident or DOM element (for panes).

This lets applications use Leaflet's essential APIs when they are needed, but it may cause problems when props are set, so it should be used with care.

### React context

React Leaflet is using React's background API to give children components that need it access to instances of some Leaflet elements.

The MapContainer component makes a React context for each instance of a Leaflet map. Other React Leaflet components and hooks can only be used as children of a MapContainer.

### Basic Process

* The MapContainer shows a div> element that holds the map. If the temporary replacement prop is established, it will be shown inside the div> that holds it.
* The MapContainer creates a Leaflet Map instance for the created div> using the properties of the component and makes the React context that the map instance will be in.
* The MapContainer draws the components inside it.
* Using the properties and context of the parent component, each child component creates a new Leaflet example for the element and keeps adding it to the map.
* When a child element is rendered again, the map is updated with any changes to its backed mutable props.
* When a part of the render tree is taken out, its layer is taken out of the map as needed.

### Limitations

When Leaflet is loaded, it means direct calls to the DOM. Because of this, React Leaflet cannot be used with server-side rendering.

The components that are shown are not DOM elements, but rather abstractions for Leaflet layers. Some of them have characteristics that can be changed directly by dialing the setters that Leaflet makes available, while others must be completely replaced by giving their key property a unique value so that React's algorithm can handle them correctly.

### Installation

#### By full Library:

import {  
 MapContainer,  
 TileLayer,  
 useMap,  
} from 'https://cdn.esm.sh/react-leaflet'

#### By module:

import { MapContainer } from 'https://cdn.esm.sh/react-leaflet/MapContainer'  
import { TileLayer } from 'https://cdn.esm.sh/react-leaflet/TileLayer'  
import { useMap } from 'https://cdn.esm.sh/react-leaflet/hooks'

references:

<https://react-leaflet.js.org/docs/start-installation/>