# Assignment 2:

a. Create version control account on GitHub and using Git commands to create repository and push your code to GitHub.
b. Create Docker Container Environment (NVIDEIA Docker or any other).
c. Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component

Theory:
**Part a.**
1. What is Git?
Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

- Tracking code changes
- Tracking who made changes
- Coding collaboration

2. What does Git do?

- Manage projects with Repositories
- Clone a project to work on a local copy
- Control and track changes with Staging and Committing
- Branch and Merge to allow for work on different parts and versions of a project
- Pull the latest version of the project to a local copy
- Push local updates to the main project

3. Working with Git

- Initialize Git on a folder, making it a Repository
- Git now creates a hidden folder to keep track of changes in that folder
- When a file is changed, added or deleted, it is considered modified
- You select the modified files you want to Stage
- The Staged files are Committed, which prompts Git to store a permanent snapshot of the files
- Git allows you to see the full history of every commit.
- You can revert back to any previous commit.
- Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

4. Why Git?

- Over 70% of developers use Git!
- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

5. What is GitHub

- Git is not the same as GitHub.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.

6. Steps to Push and PULL version control repository to GitHub

| Step No | Command | Description |
|---|---|---|
| 1 | Git Installation | Download Git from the website: https://www.git-scm.com/ |
| 2 | Command line<br>>git –version | If Git is installed, it should show something like git version X.Y |
| 3 | `git config --global user.name "w3schools-test"`<br>`git config --global user.email "test@w3schools.com"` | Configure Git<br>Change the user name and e-mail address to your own |
| 4 | `mkdir myproject`<br>`cd myproject` | Creating Git Folder |
| 5 | `git init` | Initialize Git<br>Initialized empty Git repository in /Users/user/myproject/.git/ |
| 6 | `git status` | To check the status |
| 7 | `git add index.html` | Add file to staging environment |
| 8 | `git add --all` | add all files in the current directory to the Staging Environment: |

| | | |
|---|---|---|
| 9 | `git commit -m "First release of Hello World!"` | The committ command performs a commit, and the -m "message" adds a message. |
| 10 | `git commit -a -m "Updated index.html with a new line"` | Skips staging environment |
| 11 | `git log` | To view the history of commits for a repository, you can use the log command |
| 12 | git *command* -help | See all the available options for the specific command |
| 13 | git help --all | See all possible commands |
| 14 | `git commit -help` | See help for specific command |
| 15 | `git branch hello-world-images` | a branch is a new/separate version of the main repository. This command creates a new branch hello-world-images |
| 16 | `git checkout hello-world-images` | checkout is the command used to check out/ move to a branch |
| 17 | `git checkout master` | Used to switch between branches |
| 18 | `https://github.com/` | Create a new account on github |
| 19 | | Create a Repository on GitHub |
| 20 | `git remote add origin https://github.com/w3schools-test/hello-world.git` | Push Local Repository to GitHub |
| 21 | `git push --set-upstream origin master` | push master branch to the origin url, |
| 22 | | go back into GitHub and see that the repository has been updated: |
| 23 | `git fetch origin` | fetch gets all the change history of a tracked branch/repo |
| 24 | `git merge origin/master` | merge combines the current branch, with a specified branch. |

| | | |
|---|---|---|
| 25 | `git pull origin` | pull is a combination of fetch and merge<br>It is used to pull all changes from a remote repository into the branch you are working on. |

## Part c:

Useful link - https://www.tutorialsteacher.com/angular/install-angular

1. Angular requires a current, active LTS(long term support) or maintenance LTS version of Node.js and NPM.

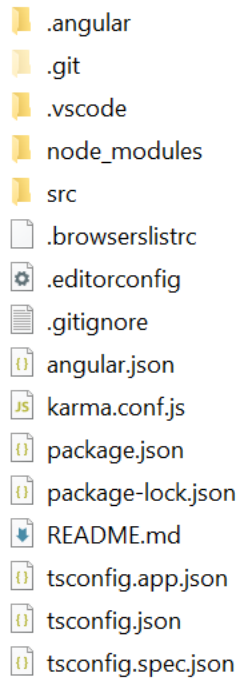
install node.js https://nodejs.org/
It will automatically install NPM - node package manager

2. Install Angular CLI

npm install -g @angular/cli@latest
To Create Angular 2 Application Angular CLI is required

3. To create new project

through CLI go to folder of the new project
Give command as -
ng new project-name
press ENTER
The project will be created as directory structure below –




Open folder src/app
Modify app.module.ts for form application

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import {FormsModule} from '@angular/forms'

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Open app.component.html
Write html code for form (representative code is mentioned here, modify for multiple inputs)

```
<h1>Simple Form</h1>
<form #simpleForm = "ngForm" (ngSubmit) = "getValues(simpleForm.value)">
  <input type ="text" ngModel name = "user" placeholder = "Enter Name">
  <br> <br>
  <input type ="text" ngModel name = "age" placeholder = "Enter age">
  <br> <br>
  <input type ="text" ngModel name = "city" placeholder = "Enter city">
  <br> <br>
  <button>Get user value</button>
</form>
```

Make changes in app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'AngProj1';
  getValues(val:any)
  {
    console.log(val);
  }
}
```

Here getValue() function which is called in form file is defined.
You can check inputted values through form in console.

build application
1. Use Angular CLI command ng serve -o to build an application.
The -o indicates to open it automatically in the default browser.
2. Use NPM command ‘npm start’ to build an application
http://localhost:4200 to see the application home page.
3.Open the terminal in VS Code from menu Terminal -> New Terminal,
and type ng serve -o command and press enter,

You can send the form contents from console to other page.
On the basis of above implementation, you can design login user, show user data.