**Set up Git:**

1. **Download and install git bash**
2. **Clone the git repository from the git bash terminal with the command:** 
   1. **Git clone** [**https://github.com/sanket0354/world-mindz.com.git**](https://github.com/sanket0354/world-mindz.com.git)
   2. **May ask you your username and password**
3. **Now, change the directory from the git bash terminal to the folder that was created after cloning: cd world-mindz.com**
4. **Run the following command from git bash terminal:**
   1. **git config --global user.name "<first name> <last name>"**
   2. **git config --global user.email <email id>**
5. **To check user name and email id are there successfully run the following command in git bash:** 
   1. **git config –list**
6. **Run the following command**
   1. **git config --global core.autocrlf true**
   2. **this commands handles the line endings of editors and keeps from messing up with the changes evertime we commit**

**Setting up braches:**

I have created a Development branch to which we will commit and push all our changes, once we see everything in the pushed changes is fine, I or anyone can merge it in to the master branch  
  
Note:   
Master is the main branch, we don’t wont to directly submit our changes to that so Development branch is created, in case we mess up, our main code still stays there ☺

* **Go to the git bash terminal and change directory to world-mindz.com wherever it is and run the following command**
  1. **git checkout -b Development**
  2. **this is just first time command to set up branch locally**
  3. **‘-b’ option creates a new branch**

**Committing and pushing changes:**

* **After you do file changes,** 
  1. **To add all the files do: “git add .”**
  2. **To commit changes, do: “git commit”**
     1. **When you run the above command a editor will open in git bash terminal, press ‘I’ to insert text, now the first line would be empty so write the message of what you are submitting example: “html and css of header “**
     2. **Now press “esc”, then press “:wq” and then press enter**
     3. **This saves the file and quits the editor**
     4. **As of now your changes are committed**
  3. **To make the changes appear in the git repo do: git push**

You **may** encounter a error while doint git push for the first time saying: “fatal: The current branch Development has no upstream branch.

To push the current branch and set the remote as upstream, use

git push --set-upstream origin Development “

if you encounter this run: git push --set-upstream origin Development

This is just one-time command, next time you want to push the changes you just need to run “git push”, as stated above

**To update the git repo in computer, pull new changes**

1. **We always want to pull from master because it is the main branch, do “git checkout master”**
2. **This pulls the changes in: “Git pull”**
3. **Now we want to merge all the new pulled changes from master to Development, so,**
   1. **“git checkout Development”: changes the local branch to Development**
   2. **“git rebase master”: this merges all the changes that we pull in to the master in to our development branch**
4. **Make sure you always work, commit and push in Development branch to avoid conflicts with master. We only do pull in master**
5. **In git bash, you can see the branch displayed something like this** 
   1. **sanks@sanky MINGW64 /D/WEBSITE/world-mindz.com (Development)**

**“git log”, if you run this command, it will show you all the recent changes**

**“git status” – a handy command:**

**This command shows the status of the repository and files**

**Suppose you made changes to some file, try running “git status”, you would be able to see files in red indicating changes are made**

**Now, suppose you add then to your commit using “git add .” , then run “git status”, you will see all the files color turning green showing you they are added to commit**

**Now once you commit the changes, then run “git status”, you will see no file instead you will see some message that everything is up to date.**

**“git diff”, if you run this command it will show you what changes you made to your files**