INSTALLING GIT

- For installing **GIT**, we need to go to **https://git-scm.com/** then based on our operating system, it will offer a download for us. As my current operating system is **windows**, so it's offering download for **windows**.

- When you click on it, if we are on windows, we are going to get a setup file and we can install it.

- But for **mac OS**, we get shell commands like below:

```
$ brew install git
$ sudo port install git
```

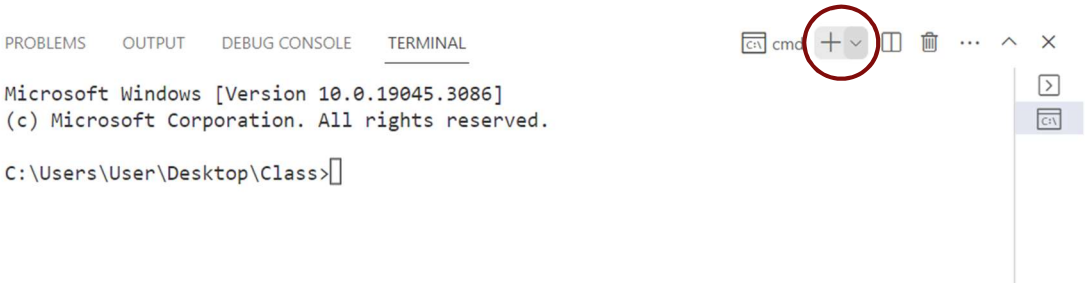
INSTALLING NODE JS

- For getting started with React JS or Node Server, we need to setup Node JS environment.
- We need to go to the **https://nodejs.org/** to install Node JS on our machine. Here we will see two different versions of Node JS, the right one is the current version and the left one is the LTS (Long Term Support) version.

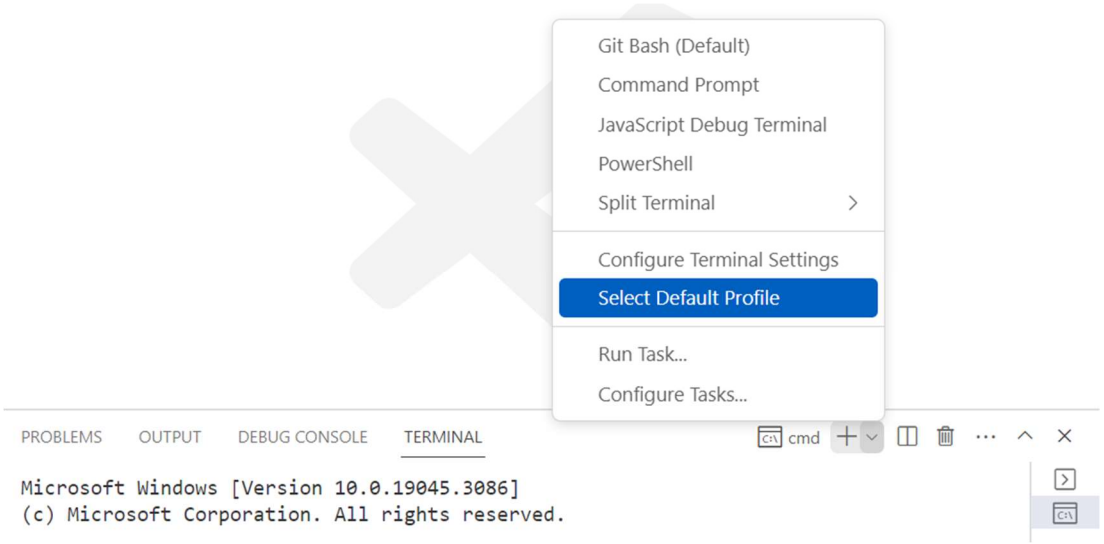
- The left one will be the stable version and the most preferable version of Node JS.
- So let's click on that and it will install Node JS based on our operating system.

INTEGRATING GIT WITH VS-CODE

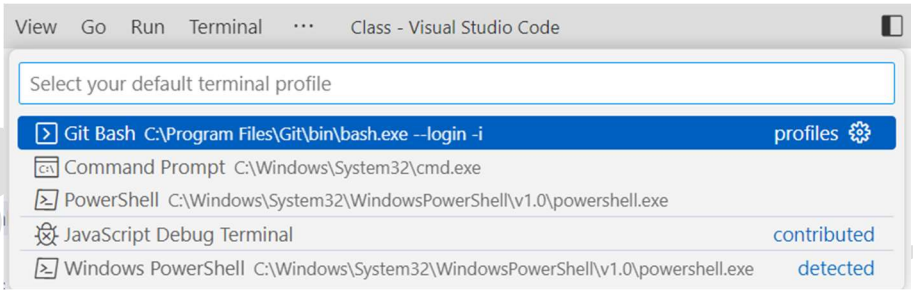
- First of all, open up the VS-Code, then go to the terminal menu and click on new terminal.
- As you can see the default terminal profile of VS-Code is set on cmd, we want to set the default profile to Git-Bash because our commands will be similar to macOS and Linux based operating systems.
- Click on the top-right down-arrow icon



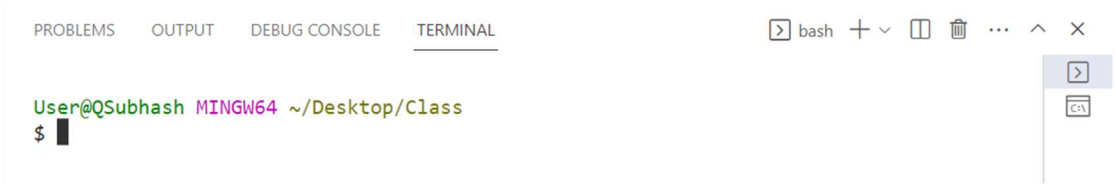
- Click on select default profile.



- Select Git-Bash

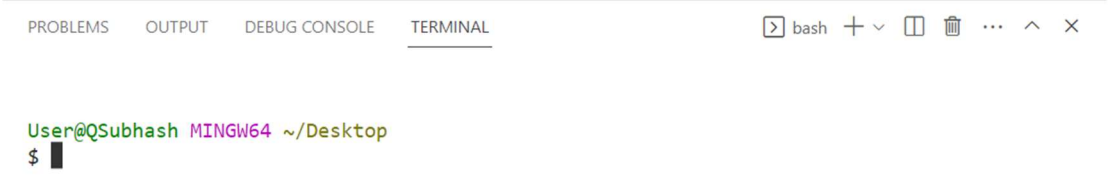


- And we can see, our profile is set to bash.



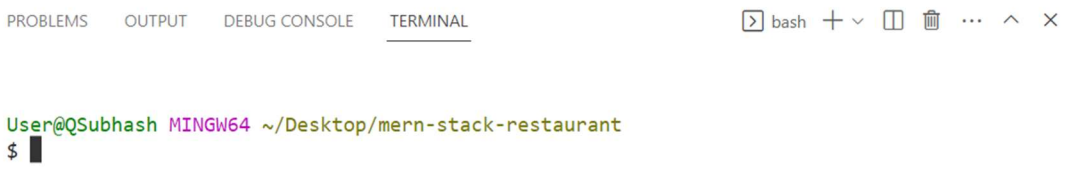
CREATING A NEW REACT JS APPLICATION

- Now we want to create our new project. Go to the desktop directory by running the command **`$ cd desktop`** from root directory.



- Create a new folder (in our case it can be named as mern-stack-restaurant) by running the command `$ mkdir mern-stack-restaurant` and go inside it by the command `$ cd mern-stack-restaurant`

- Upto now, our command line looks like below:



- Let us now create a new react application with the name of frontend inside the above created folderby writing `$ npx create-react-app frontend`

- After the successful creation of the application, we will get the final greeting, wishing us Happy Hacking.!

- Now for serving our application on server, we have to navigate to the project by writing `$ cd frontend` and then `$ npm run start`

- The above command automatically serves our new react application by default on the server port http://localhost:3000

PUSHING THE PROJECT TO GITHUB

- First of all, we need to remove Git from the frontend folder. Because when we create a new React project, there is a Git folder inside it.

- We have to remove it, because it tracks the changes only from the frontend folder. But in the future we are going to have a backend folder here. So the changes should be tracked from the whole project.

- So to remove the Git folder from the frontend, open the new terminal and navigate into frontend folder and write `$ rm -rf .git` and it removes the Git folder and everything inside it.

- Close the terminal and open the Source Control located at the side bar menu of the VS-Code and click on Initialize Repository.

- You can see that it tracked all the changes from the root folder.

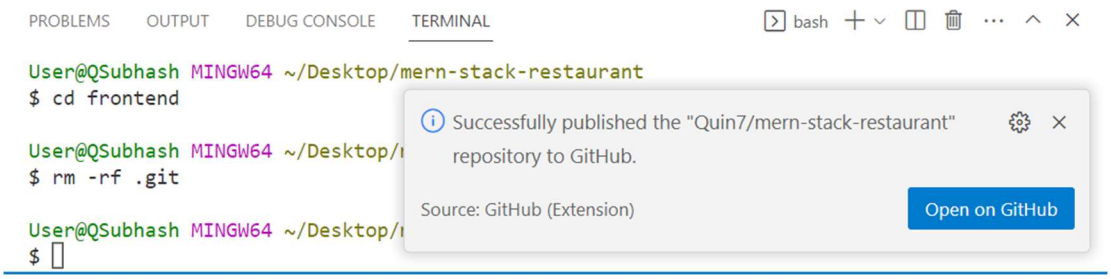
- Let's write the Commit Message and in our case it can be Create React Application.

- Click on Commit button to commit our changes. It says that GitHub extension wants to sign-in with GitHub. Click allow and we can see it will open up a new page inside our browser to get the authorization from us.

- Click on continue and then click on Open Visual Studio Code.

- Now you can set the name of our repository if we want to, or it will by default takes the name of our project folder an then select whether we want to publish your project to public or private repository.

- The above steps will automatically create a new repository on the name of our project in our GitHub profile. So we don't need to do it from our GitHub website.



- Click on Open on GitHub and you be able able to see our repository inside GitHub website. And from this moment we can push all the changes to Github with separate comments.