**Deployment of Jenkins LTS solution for achieving Continuous Build for a software.**

Sarath kumar yanamandra

2016hw86030

Wims 2016 Section B

9494897500

**Problem statement:**

Perform a Jenkins LTS Deployment and install the plugin of Git and Maven. After successful deployment change the access port for Jenkins from 8080 (default) to 9090. Also list the need and benefits of Jenkins.

How I did -

* I have downloaded Jenkins, git bash
* Installed Jenkins and created maven project
* Created a repository in git hub
* Logged in git bash and initialised local repository.
* Linked the git repo with Jenkins using git made pipeline of changes in Jenkins.

**What is Jenkins in simple words:**

**Jenkins is a software that allows continuous integration.**



**Jenkins** – an open source automation server which enables developers around the world to reliably build, test, and deploy their software.

Git is a distributed version-control system for tracking changes in source code during software development.

**Advantages of Jenkins:**

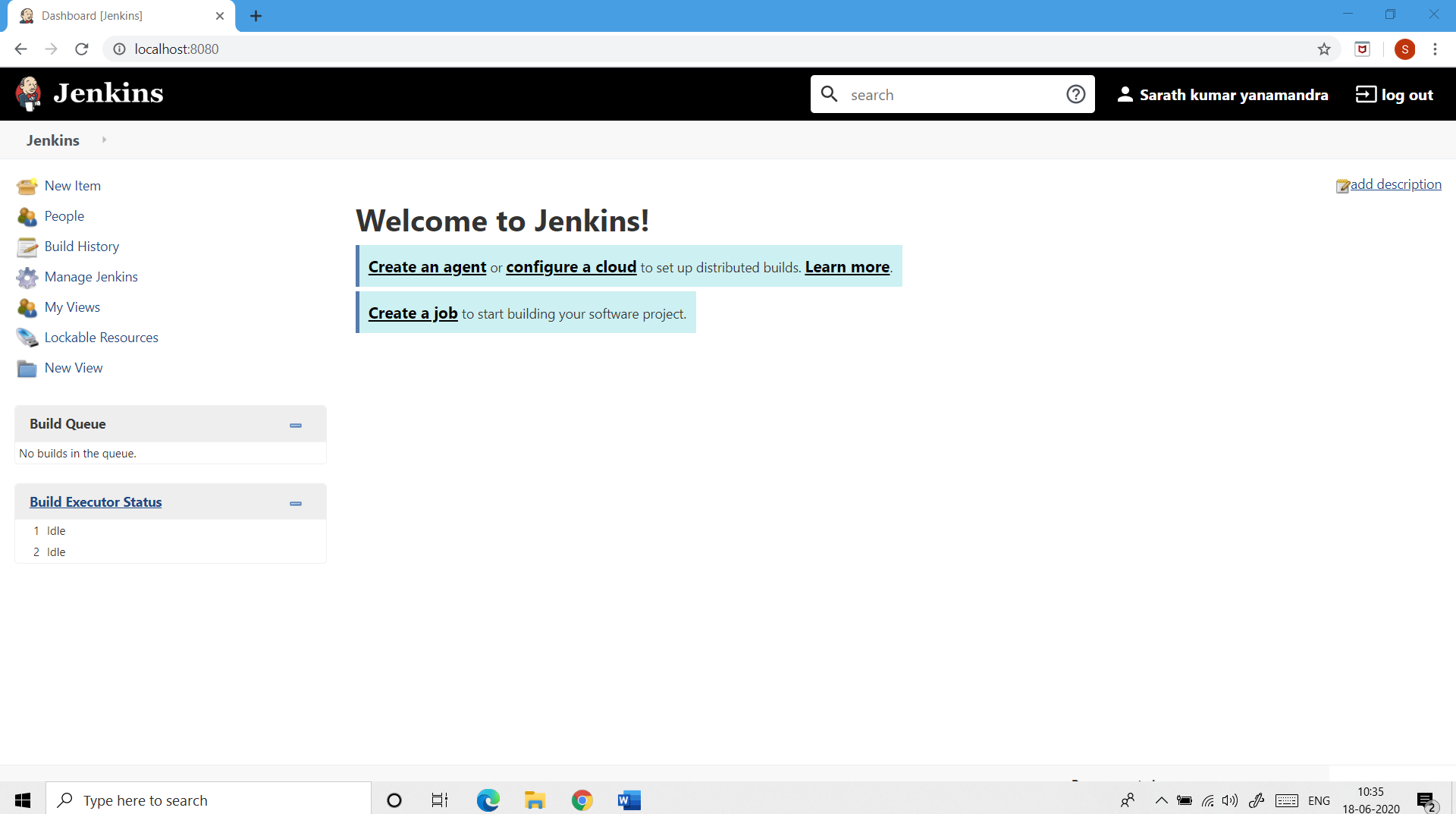
* It is open source and it is user-friendly, easy to install and does not require additional installations or components.
* It is free of cost.
* Easily Configurable. Jenkins can be easily modified and extended. It deploys code instantly, generates test reports. Jenkins can be configured according to the requirements for continuous integrations and continuous delivery.
* Platform Independent. Jenkins is available for all platforms and different operating systems, whether OS X, Windows or Linux.
* Rich Plugin ecosystem. The extensive pool of plugins makes Jenkins flexible and allows building, deploying and automating across various platforms.
* Easy support. Because it is open source and widely used, there is no shortage of support from large online communities of agile teams.
* Developers write the tests to detect the errors of their code as soon as possible. So the developers don’t waste time on large-scale error-ridden integrations.
* Issues are detected and resolved almost right away which keeps the software in a state where it can be released at any time safely.
* Most of the integration work is automated. Hence fewer integration issues. This saves both time and money over the lifespan of a project.

**Pre- requisites:**

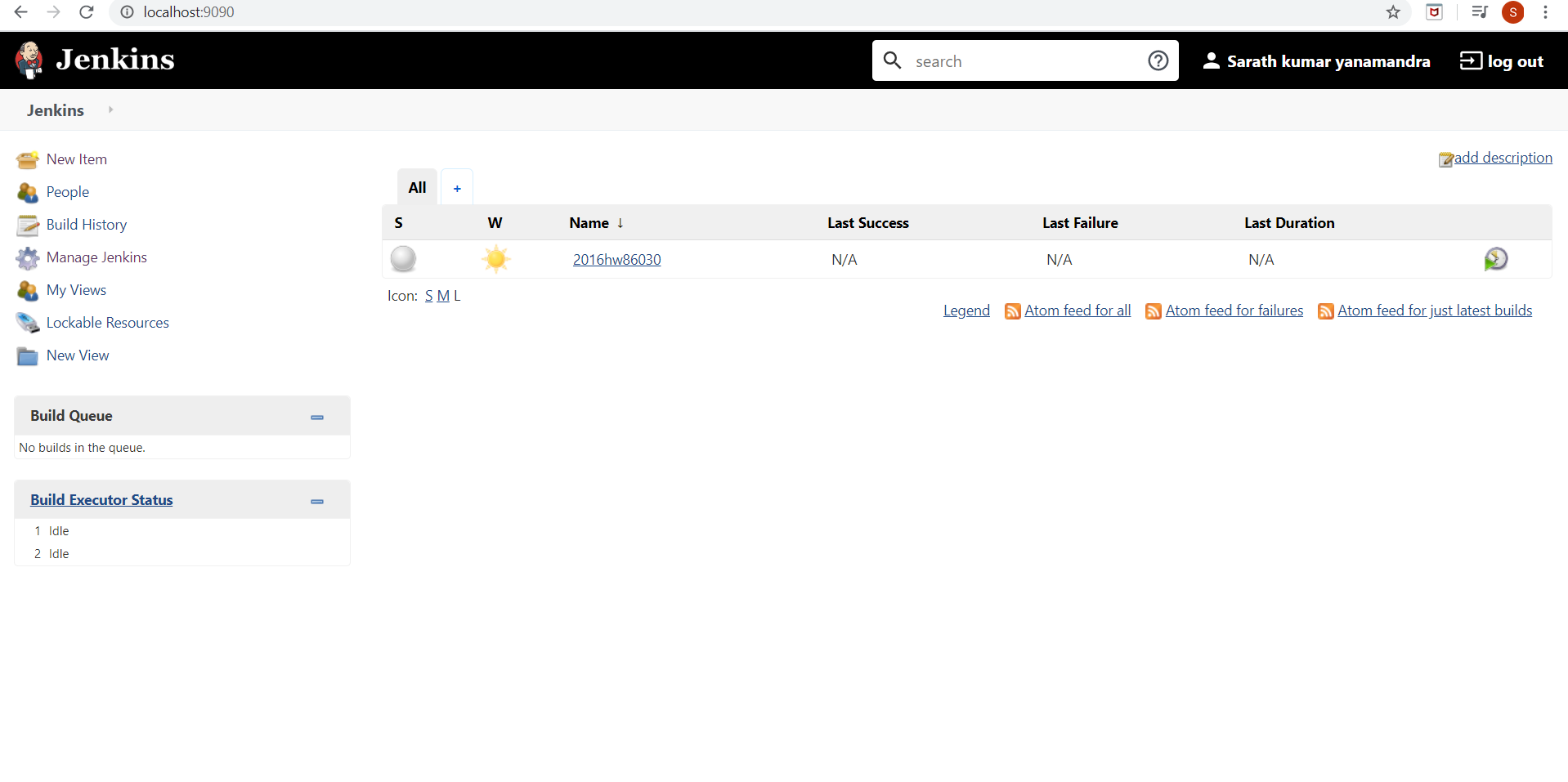
**Download the Jenkins, jdk and git bash.**

**Update the env variables for java.**

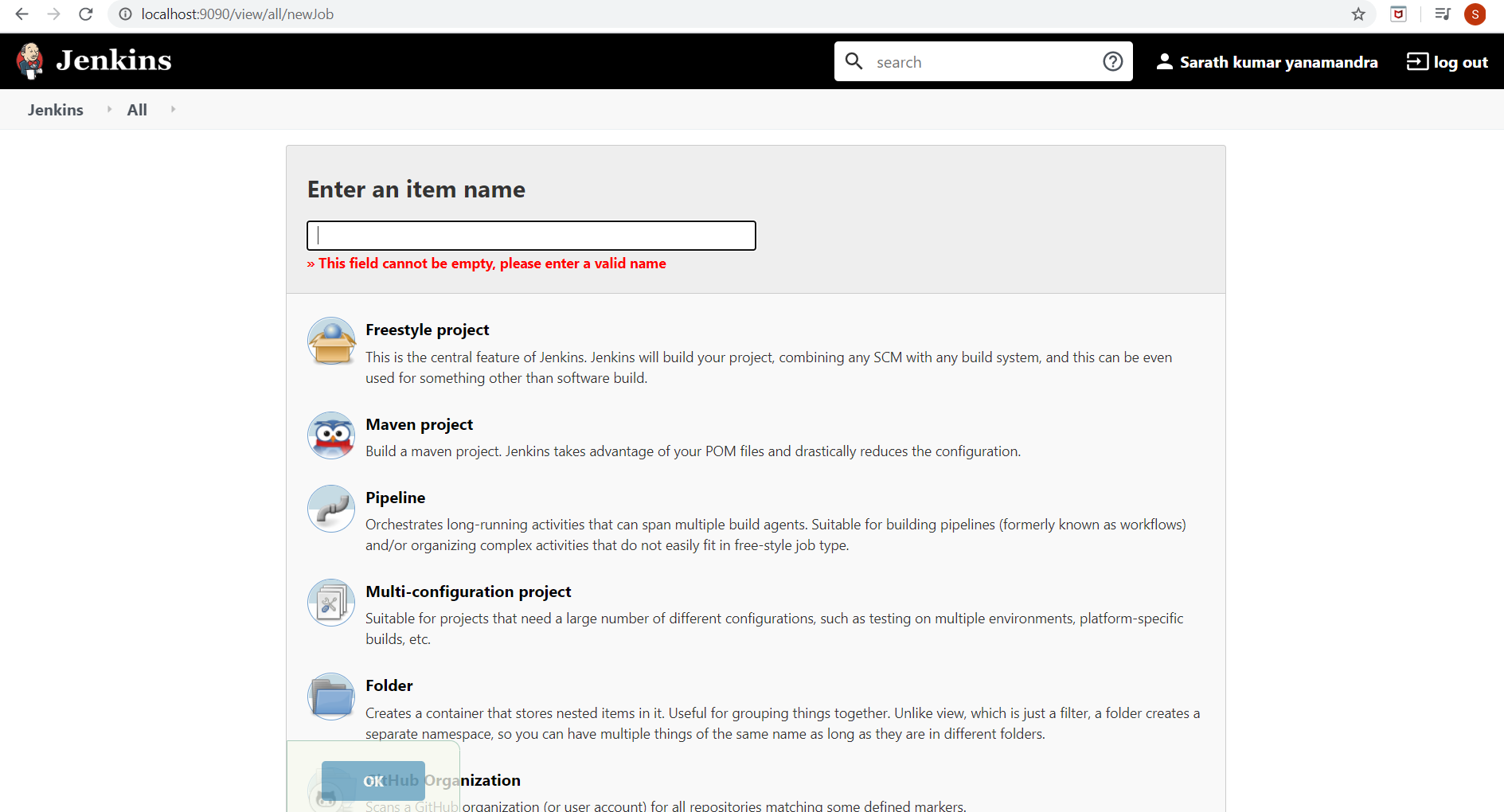
Below is the Jenkins home page, where we manage our Jenkins functionality it is currently with 8080 default port



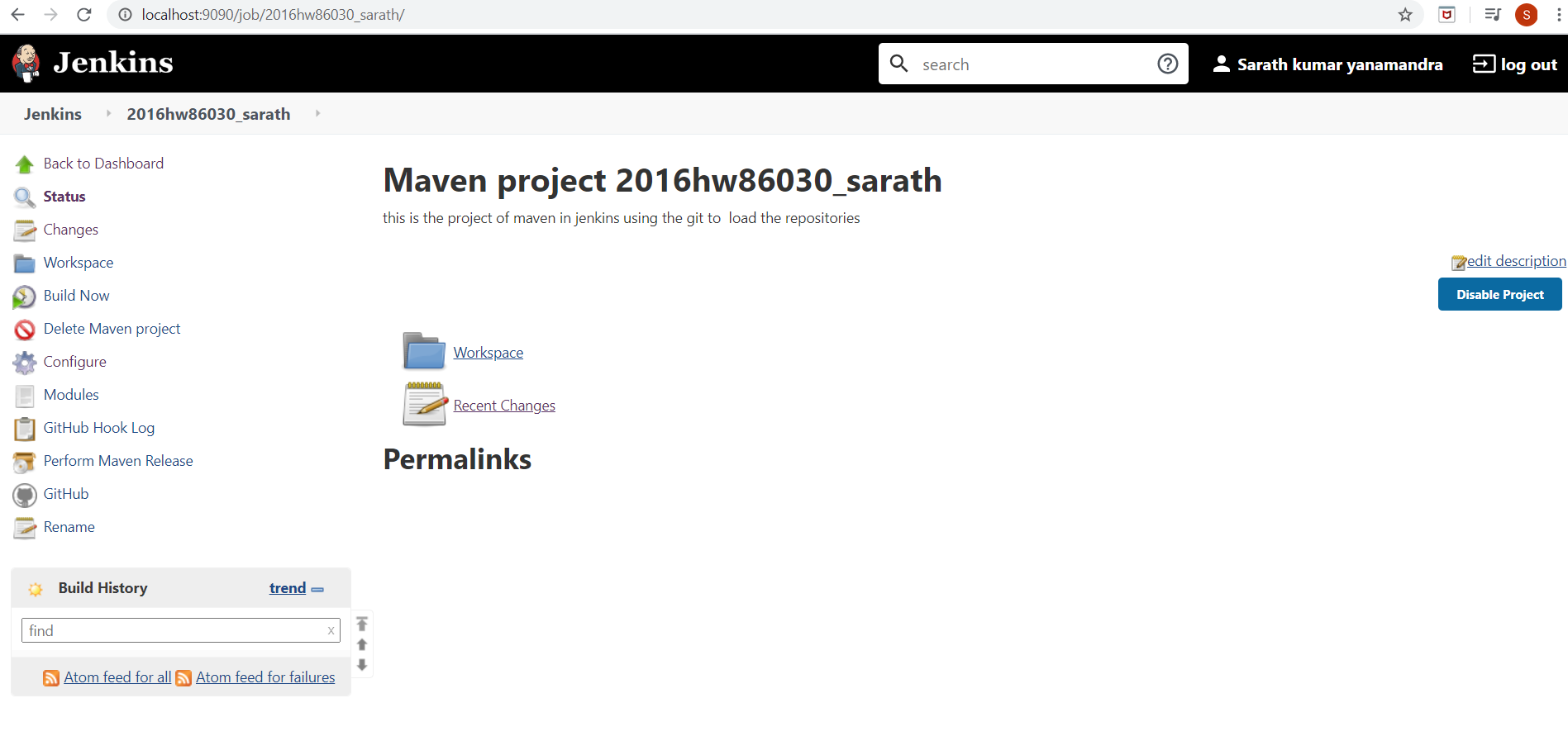
**I have modified the port to 9090 and below is the one empty project which I created.**



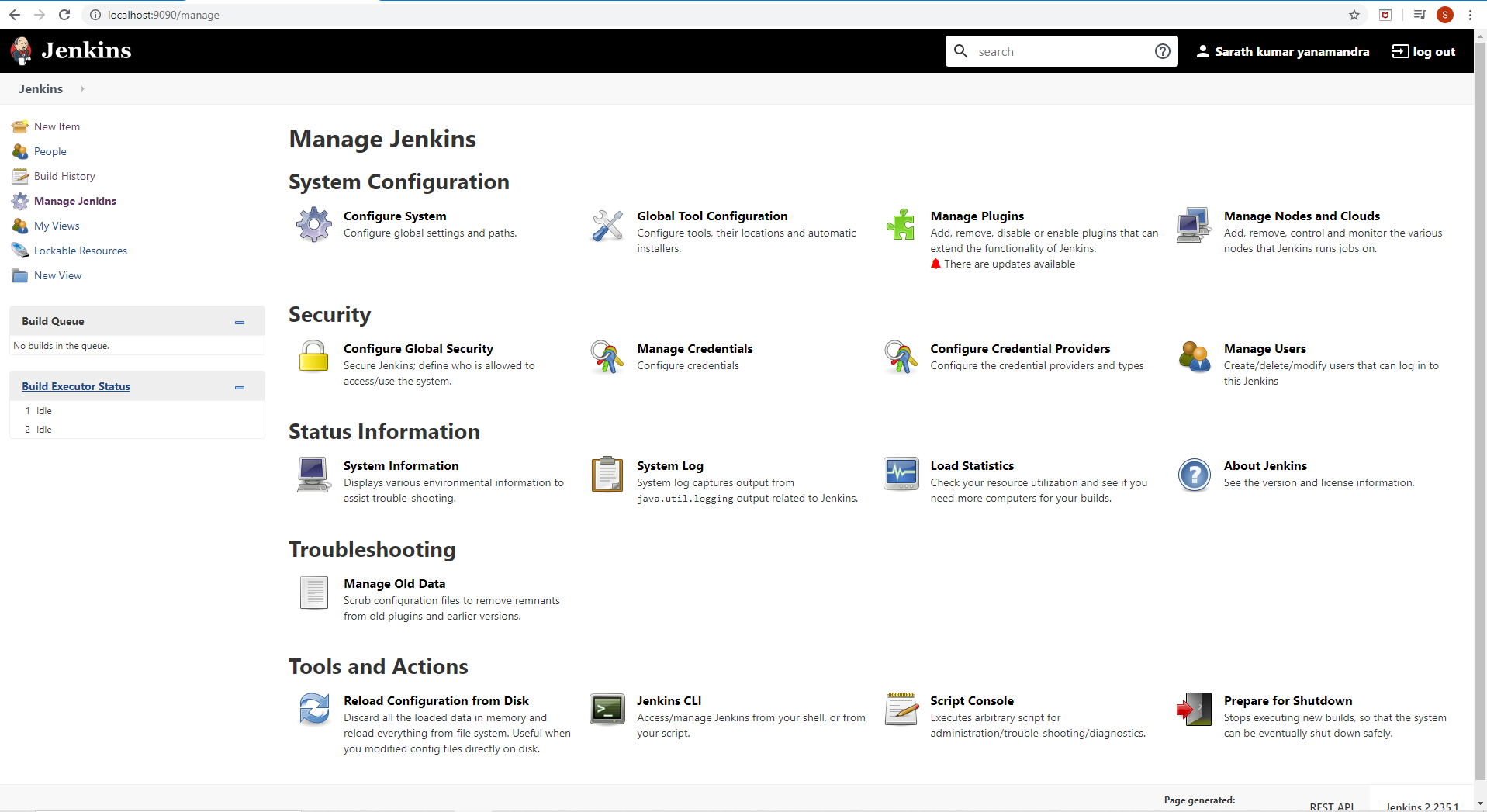
**We need to start a new project with maven project option.**



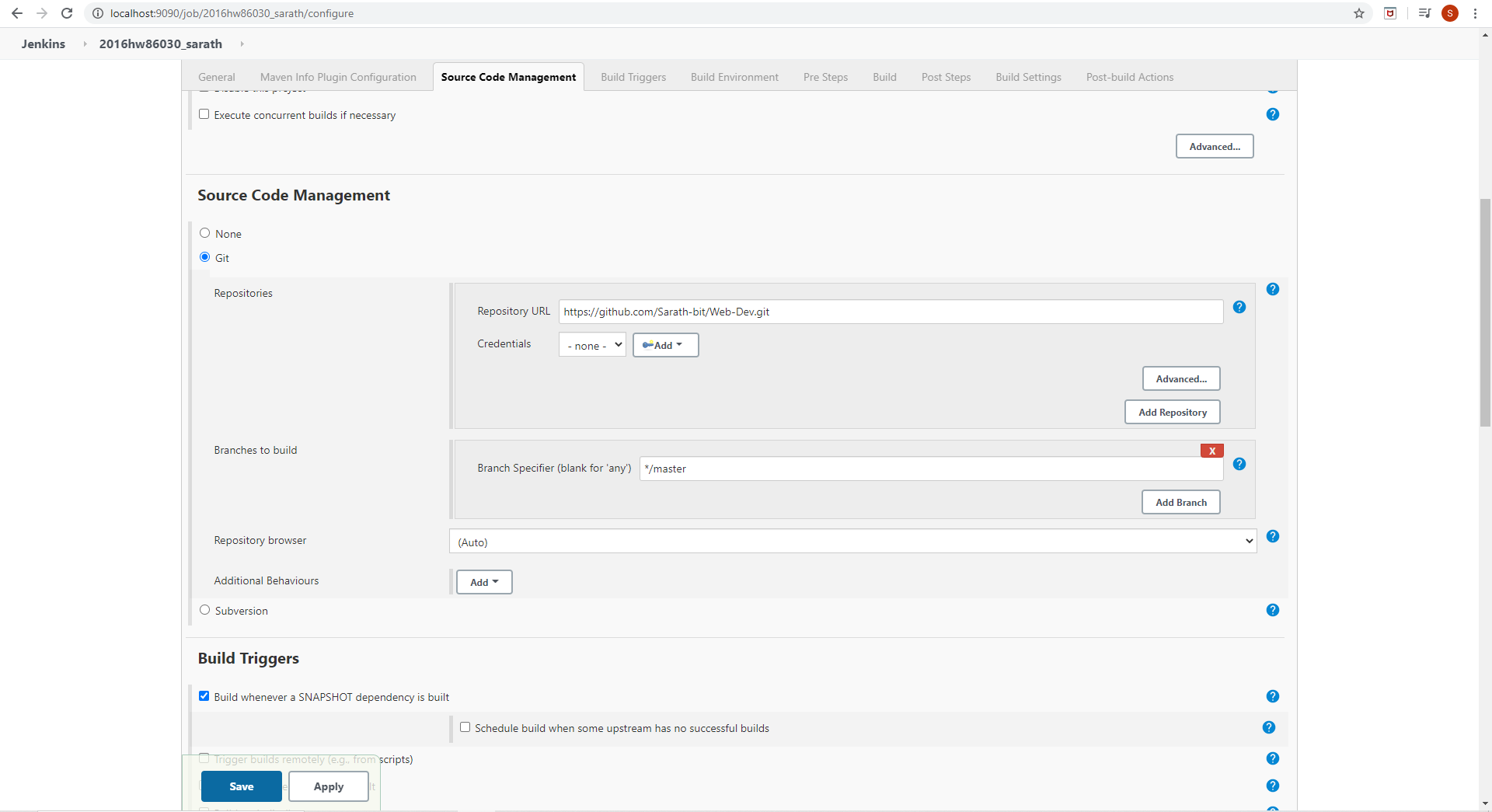
**Once created below is how project lists in you jenkins**



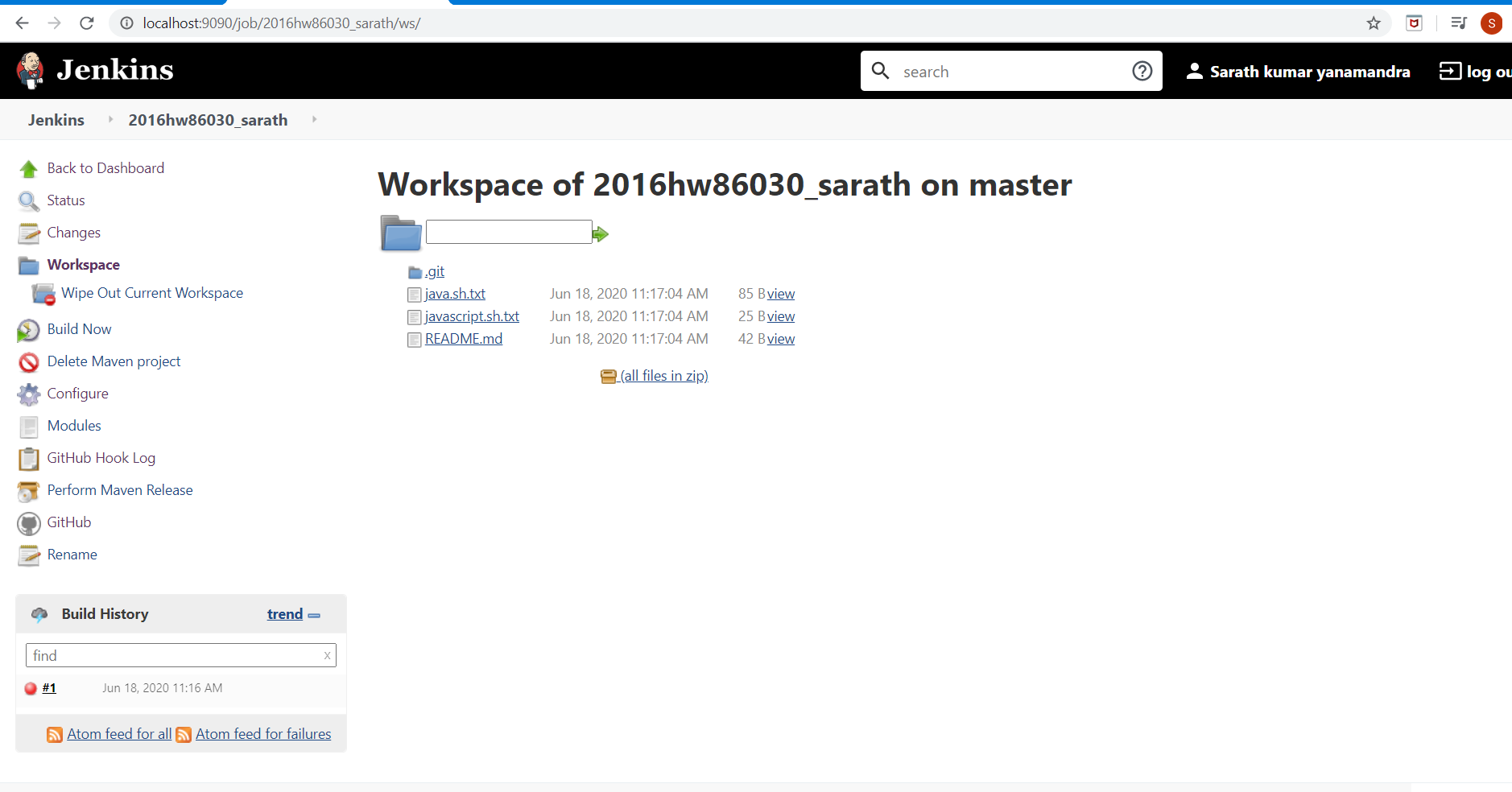
**Below is the page to manage the plugins and configurations of created project.**



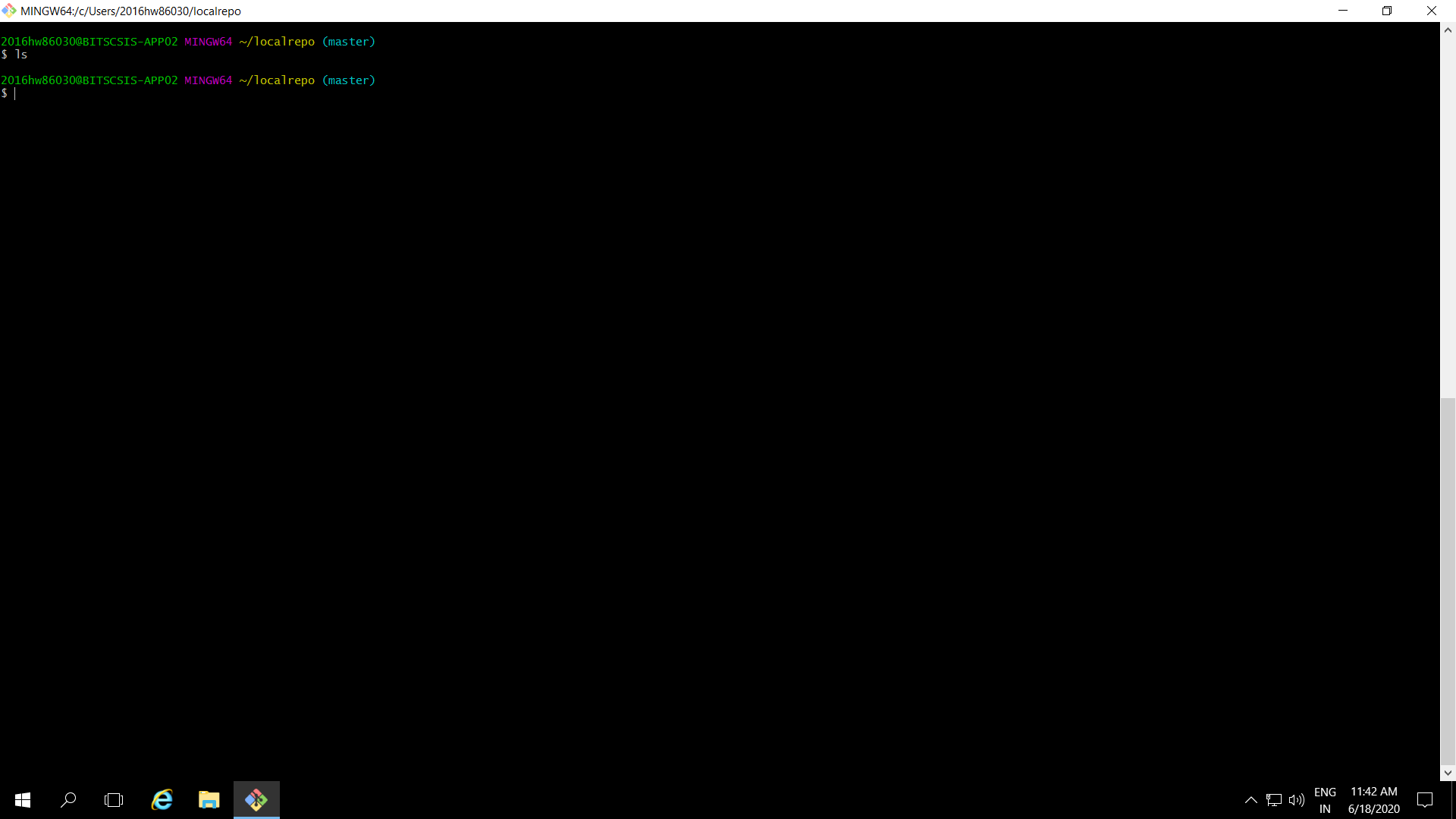
**Below is where I am defining my jenkins to integrated with GIT**



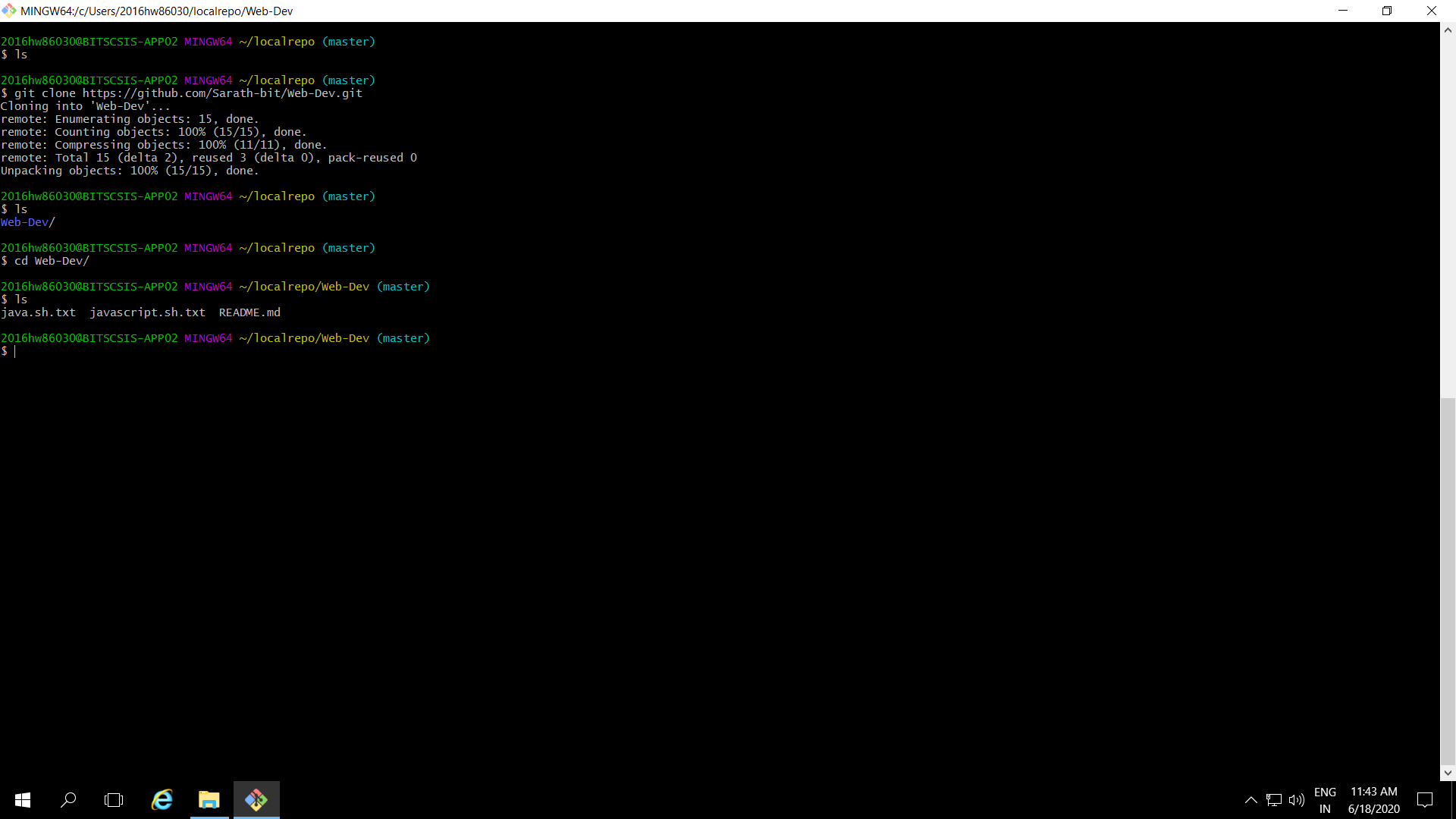
**Once configured I can able to see the data which is in my git repository**



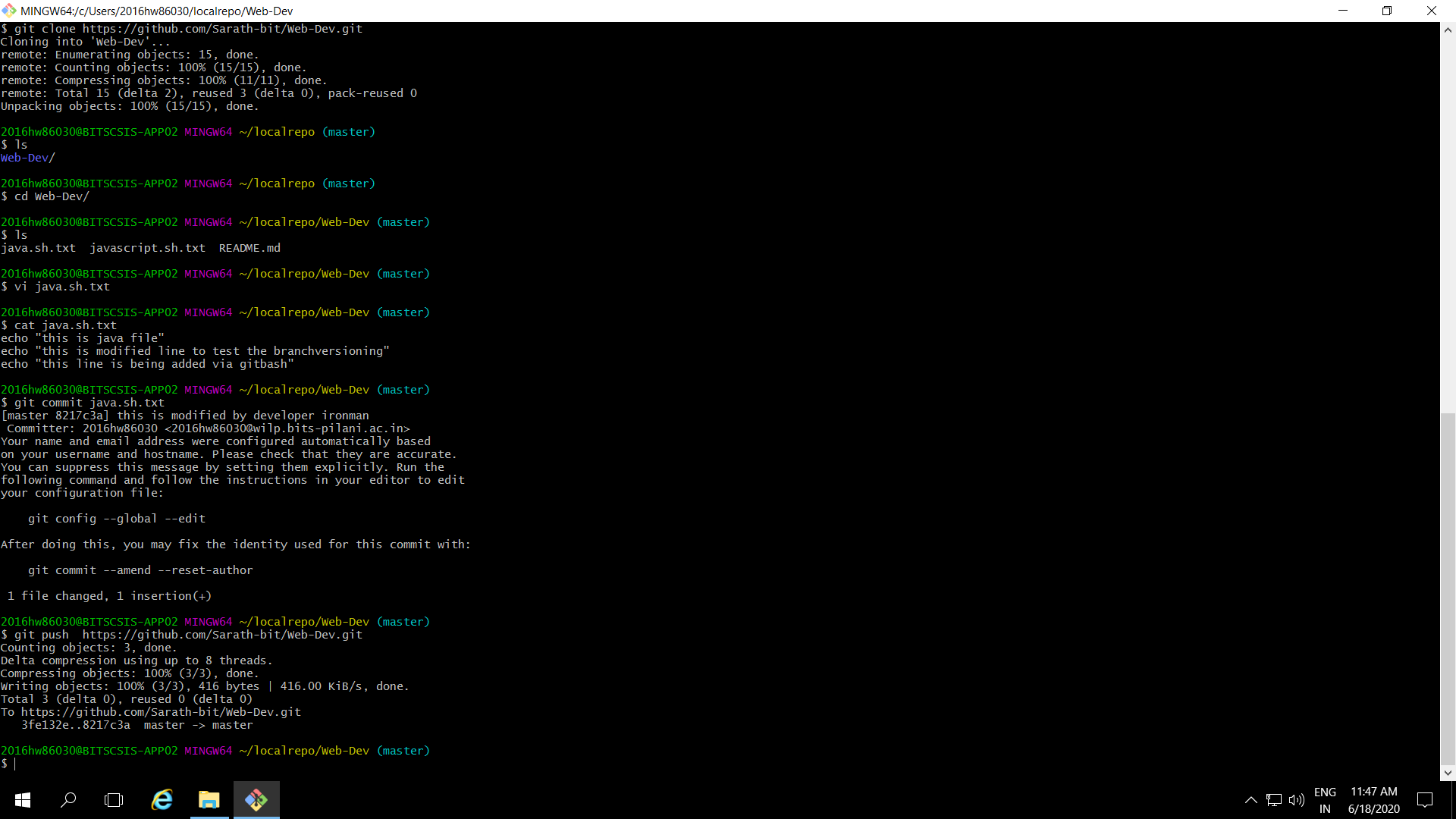
**Below is where I am accessing my git bash**



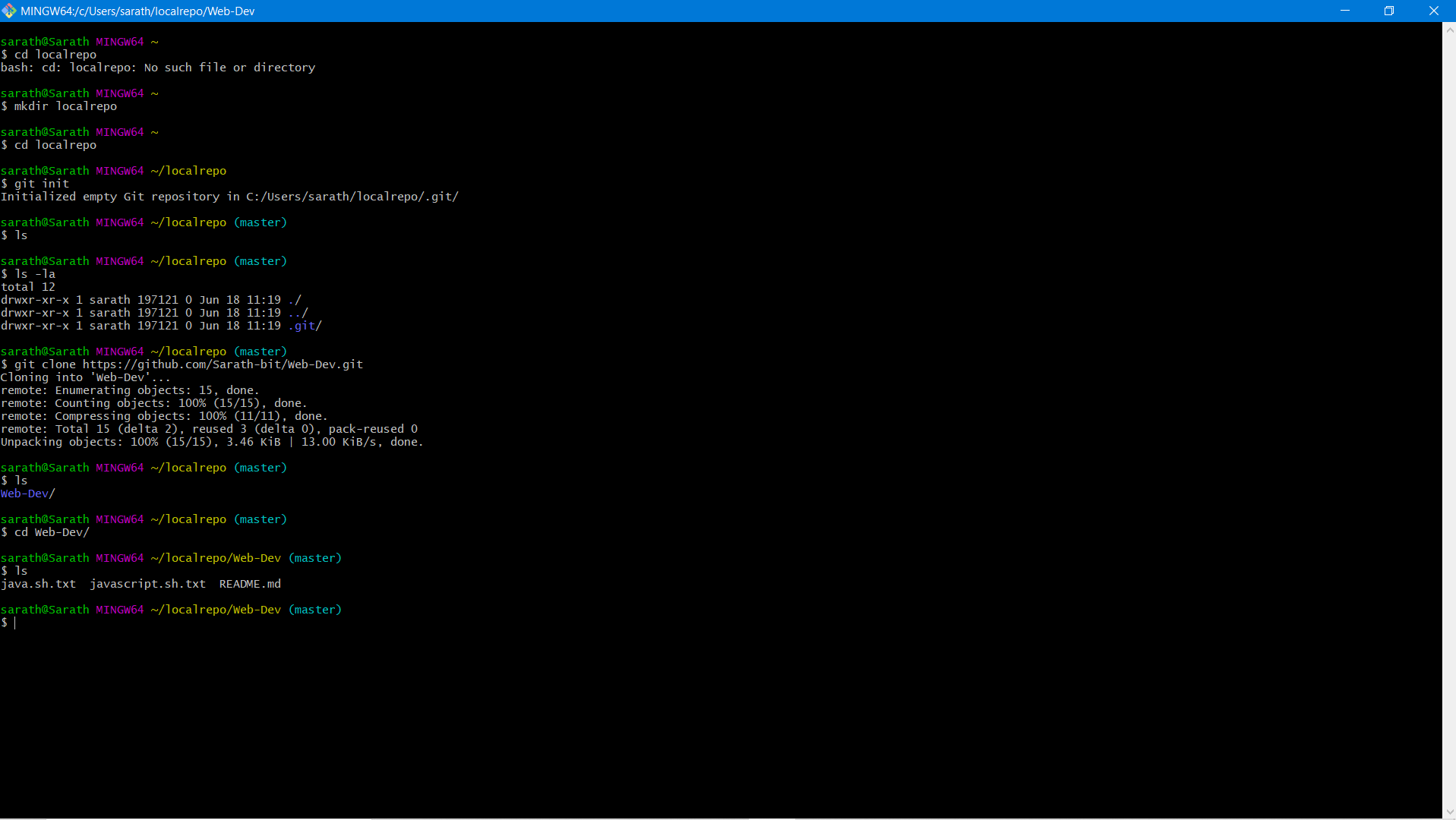
**I have initialised git**



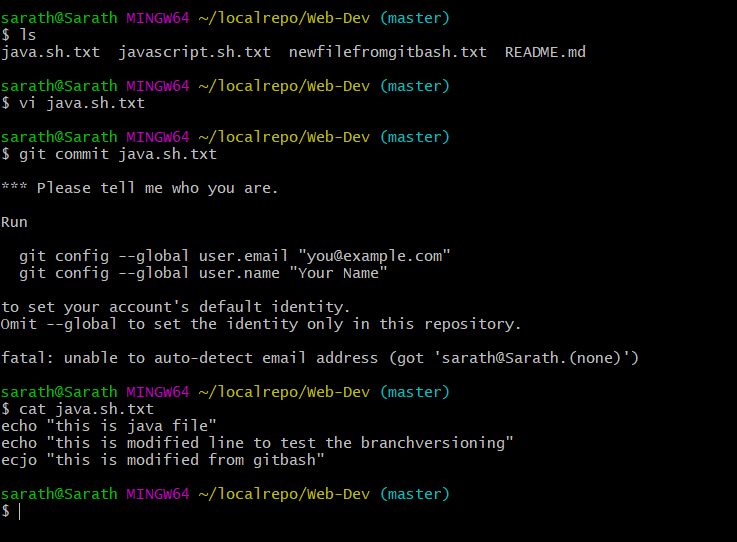
**and cloned with my git url**



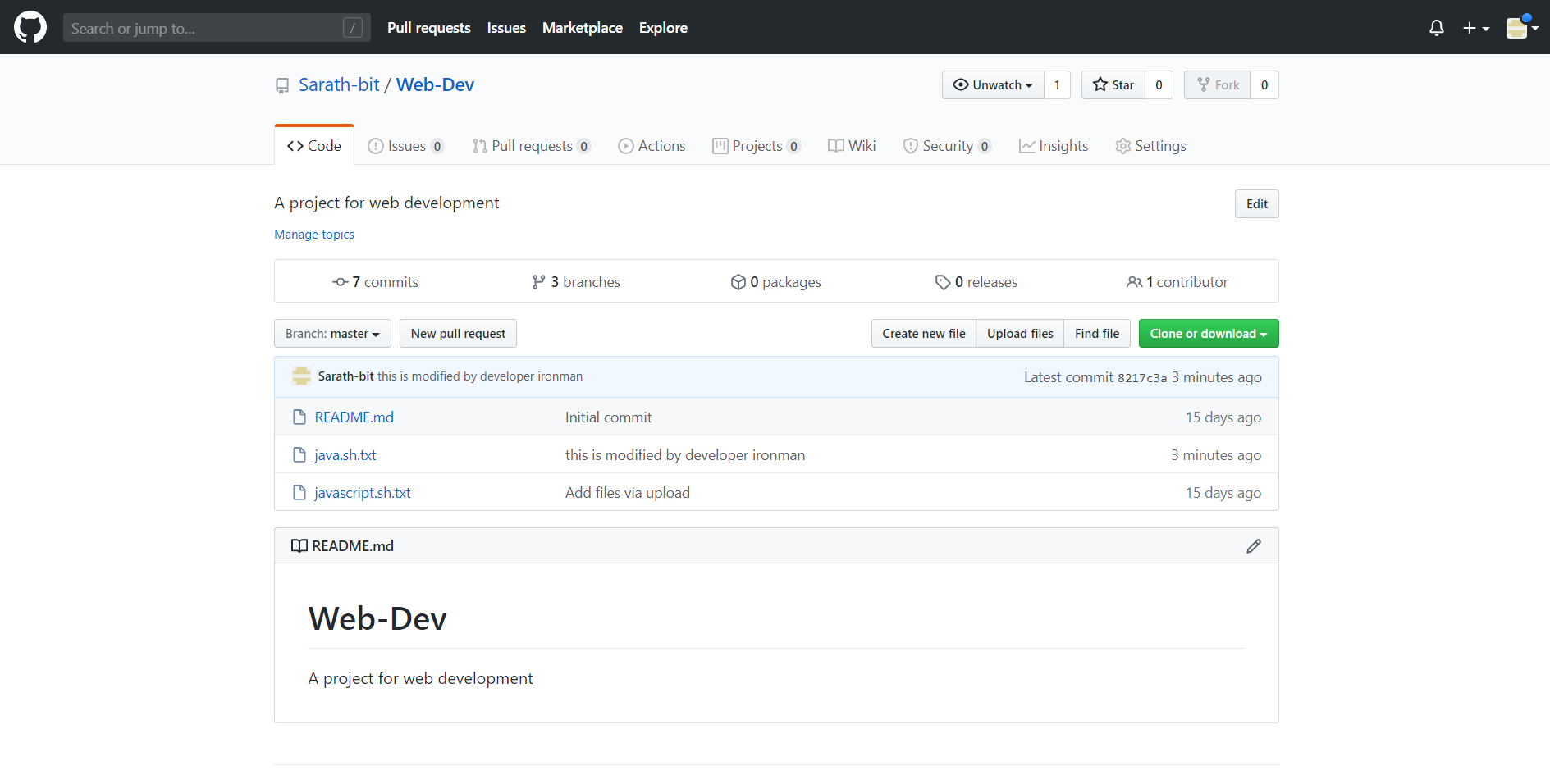
**I have made changes in to scripts which are in my git repository via git bash**



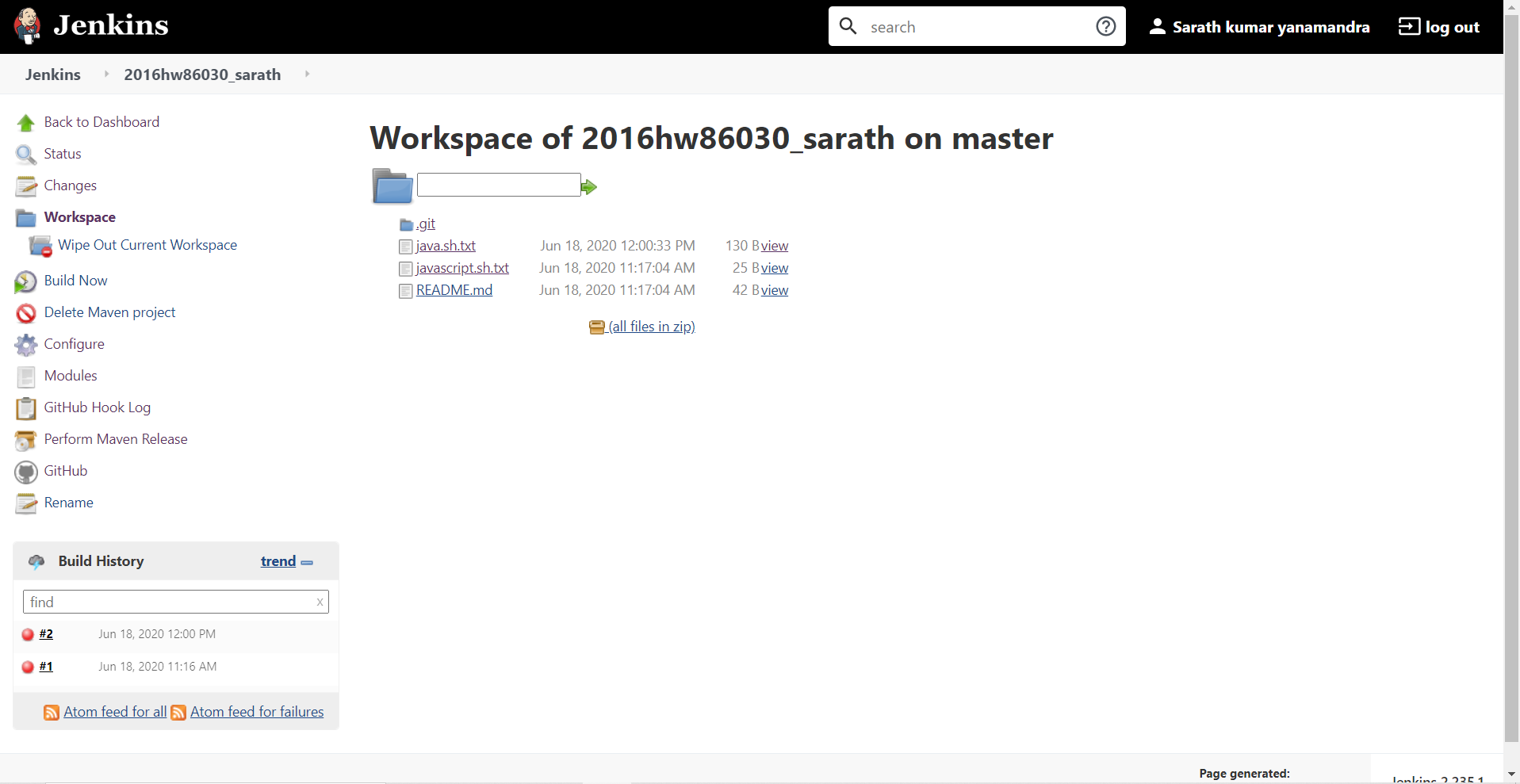
**I have saved the changes and pushed to git**



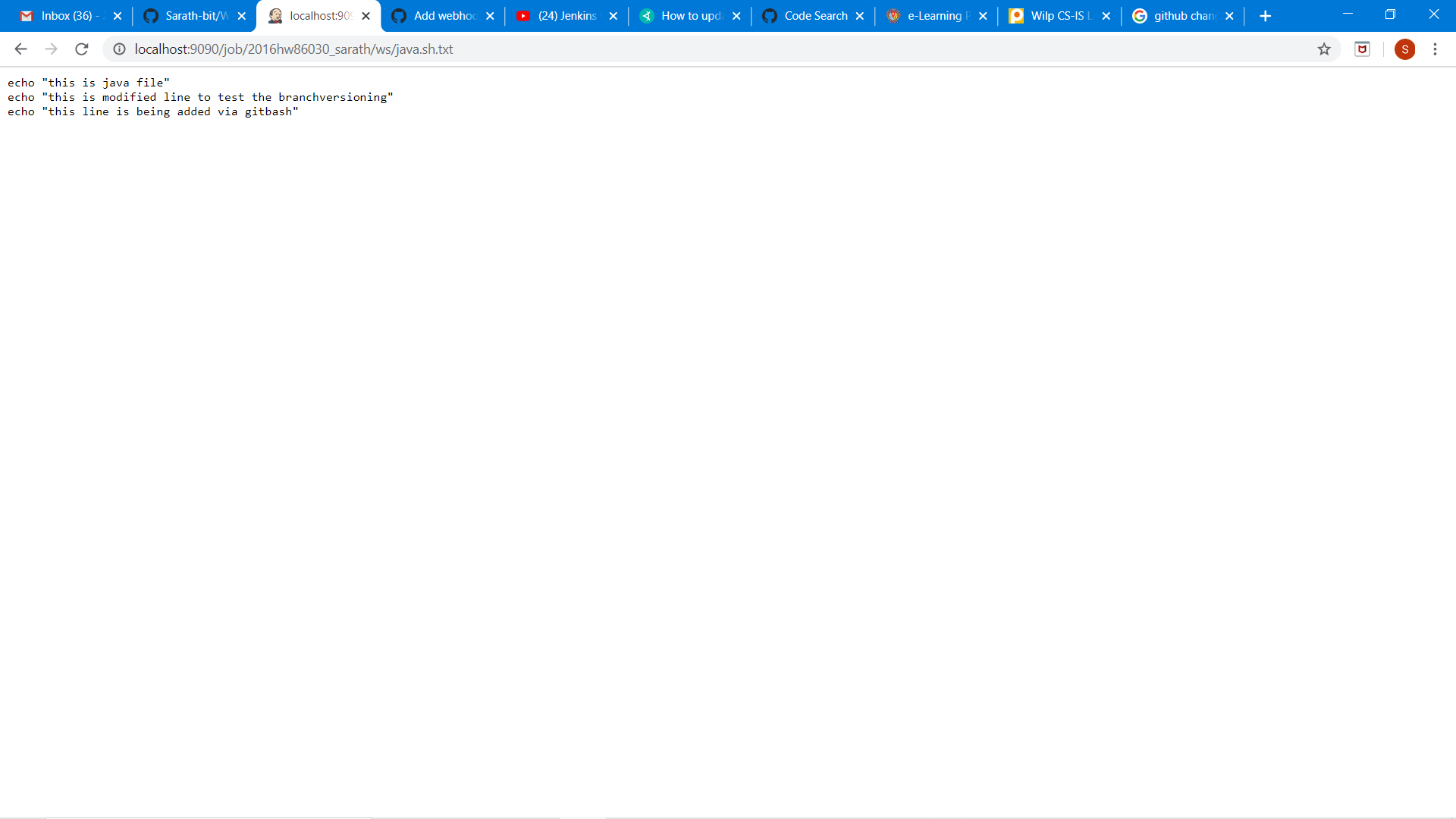
**I can now see the changes made by me from git bash are visible here.**



**Since I have linked/mapped my git via Jenkins I am able to see those change reflecting here**



**Below the line which I have modified earlier.**



**We can view all the change pipeline over here at jenkins**

