**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**Submitted to:**

Molana Haq nawaz

**Submitted by:**

Mohammad Areeb Farhan

**Roll no:** BSEF18A007

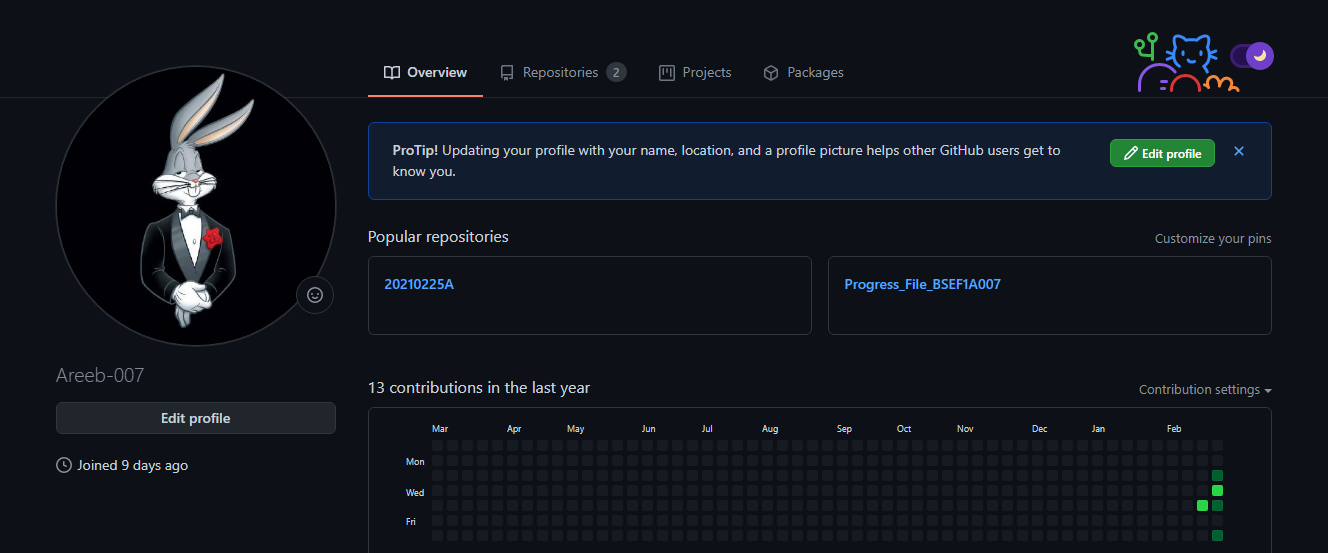
**Punjab University College of Information Technology,**

**PUCIT (Old Campus),**

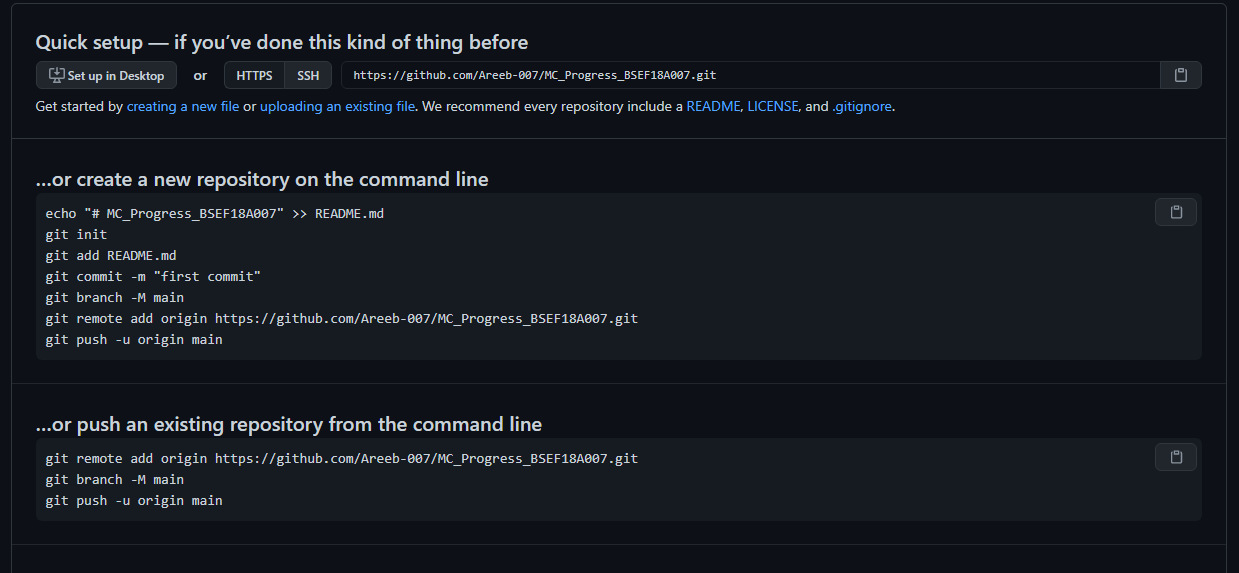
**LAHORE**

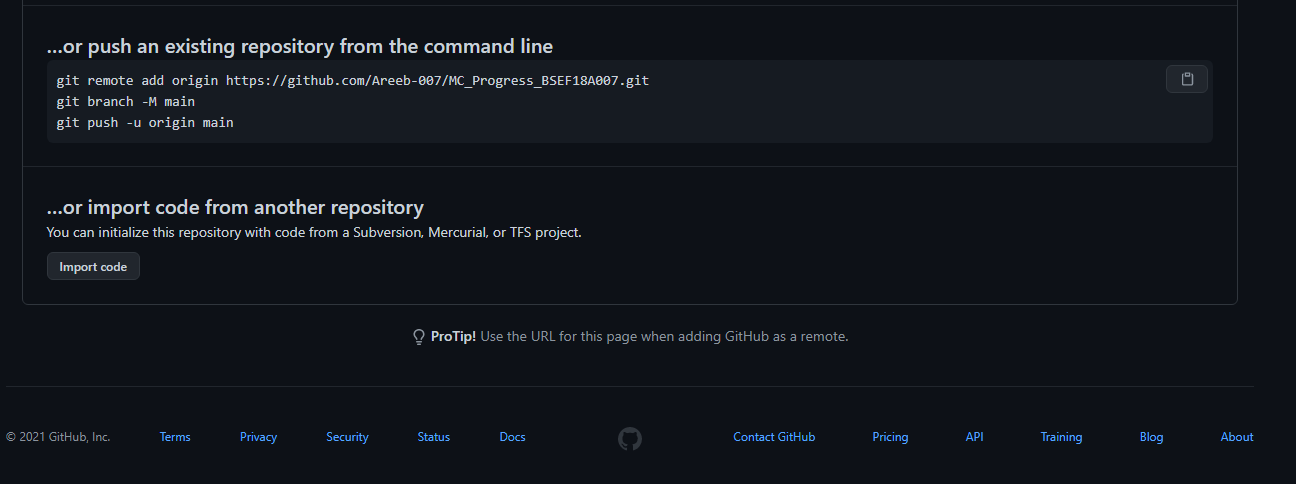
1. **Gig Account:**

I just created an account on git hub from their official website using my university email address. It is about nine days from now



1. **Git Repository**

Then I just created a new Repository by clicking on the “+” button near the profile. Then I named it as “MC\_Progress\_BSEF18A007”. 



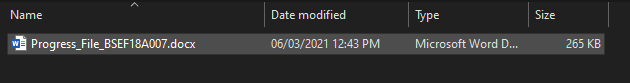
That repository provided me a link <https://github.com/Areeb-007/MC_Progress_BSEF18A007.git>.

1. **Git Clone**

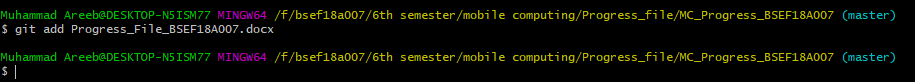
Then in order to convert my local repository into the central repository I am going to use “Git Clone ” command with addition of the previous link which was provided by the git repository.



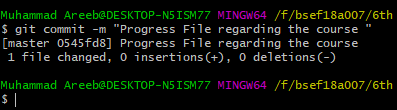
1. **Git Add**

Then I created an MS Word file named as “Progress\_File\_BSEF18A007”. 

The Purpose of this file is to upload the content we studied so far as per the task was concerned.

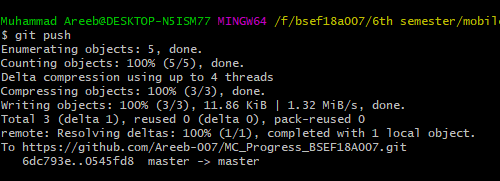
Then I ran “git add Progress\_File\_BSEF18A007.docx” command to upload this file to my local repository.  


1. **Git Commit**

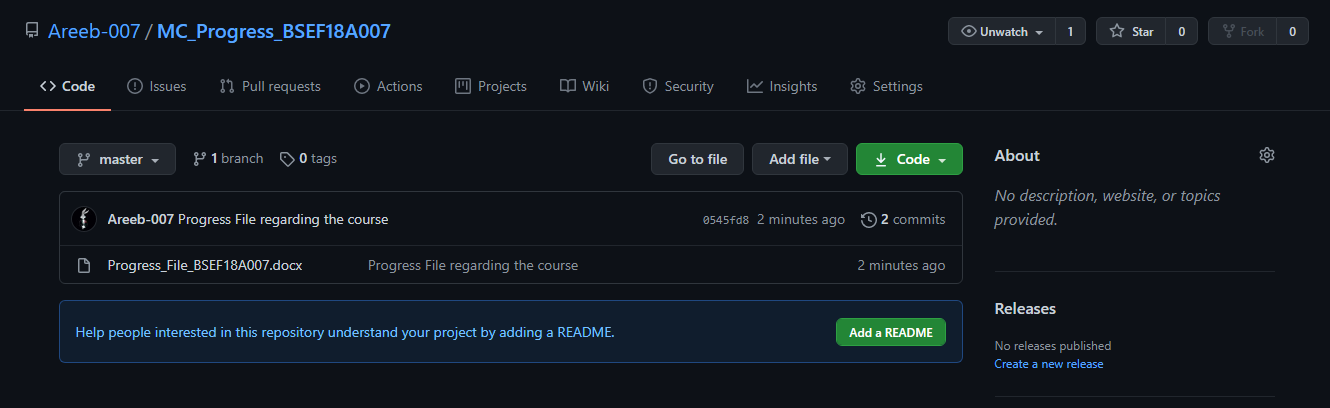
Then to upload the file with the message I used the “git commit -m “Progress File regarding the course” ”to upload that message. 

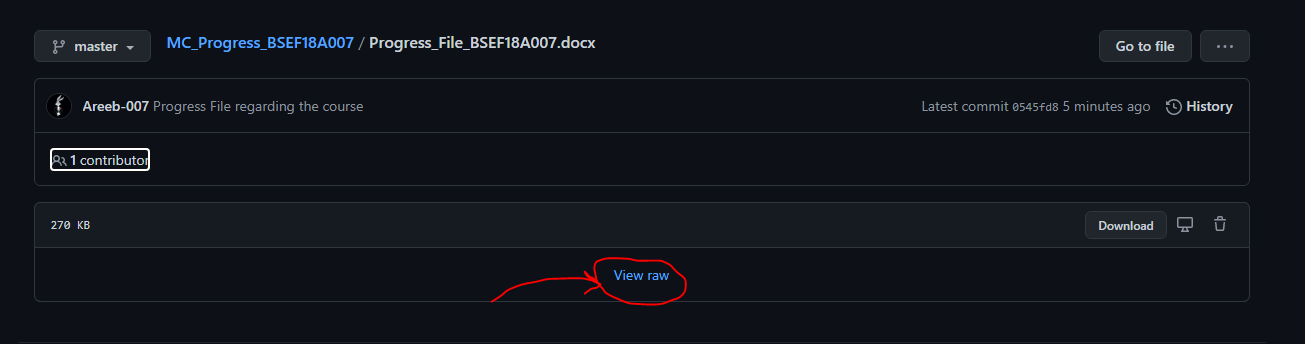
1. **Git Push**

Then to upload the file with that committed message I used “git push ” command



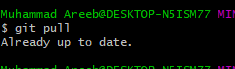
In next picture we can see that the content is uploaded on my central repository as well



And if we can click on the View Raw button we can download the button as well 

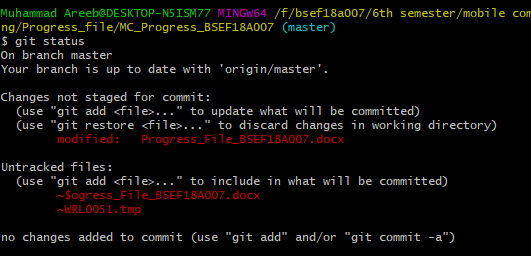
1. **Git pull**

In order to save something from your central repository to your local repository we can use “git pull” command.



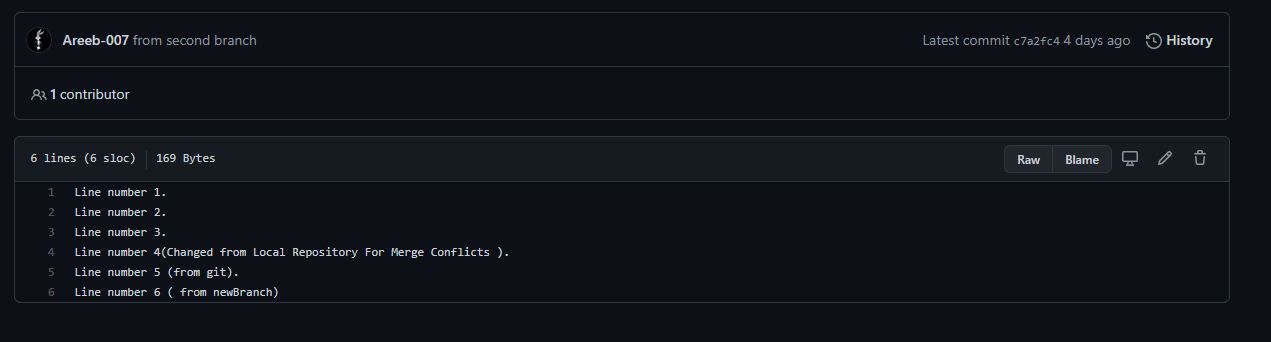
1. **Git Status**

In order to check your status of your repository one can use “git status” command



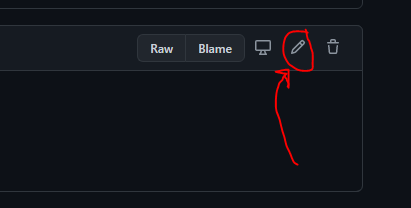
1. **View Content Online**

In order to view your file content online you can visit your git hub repository online on git hub webisite

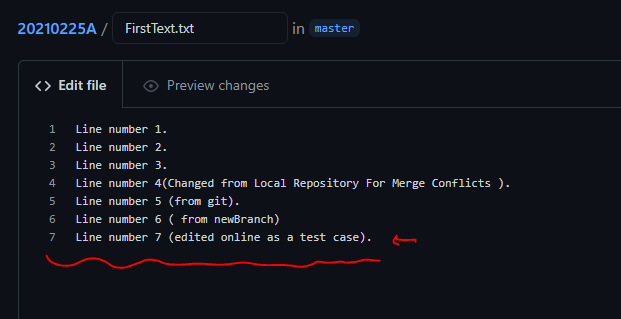


1. **Edit Online**

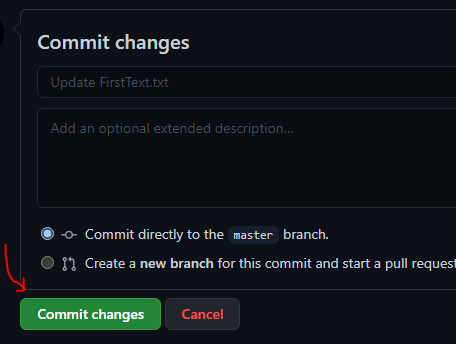
One can also edit it online from git hub website to update the content of your file.



By clicking on the button indicated we are able to edit the file online.

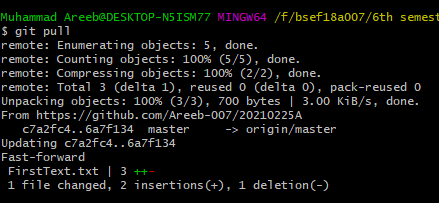


Now to save the content one just updated we click on the button below labeld as “Commit Changes”

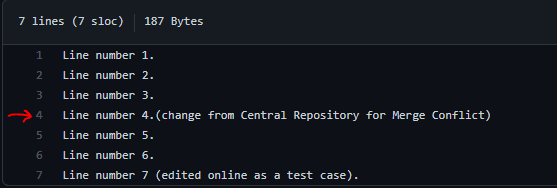


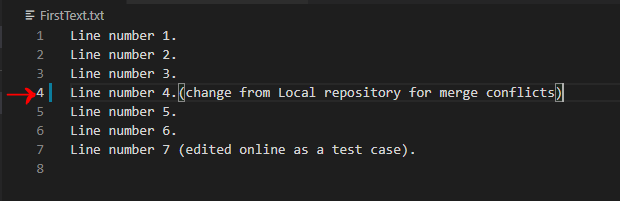
1. **Git pull**

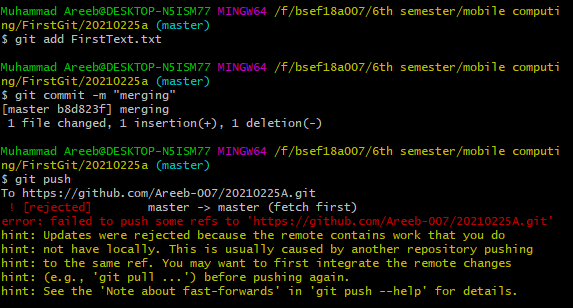
We just updated our central repository online from the website but our local repository is still as it was before update. In order to resolve that issue one can just use command “git pull”.

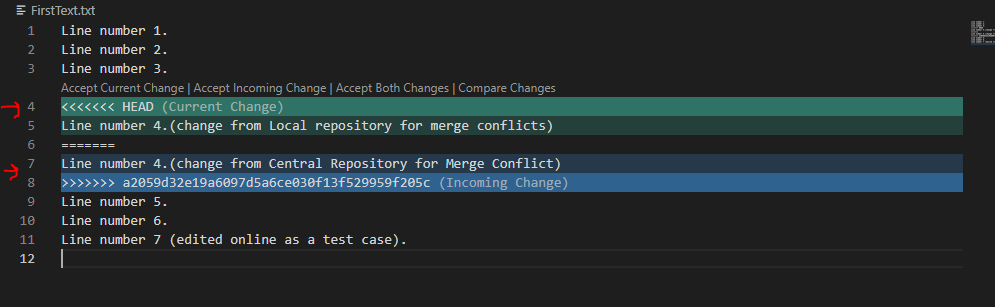


1. **Merge Conflicts**

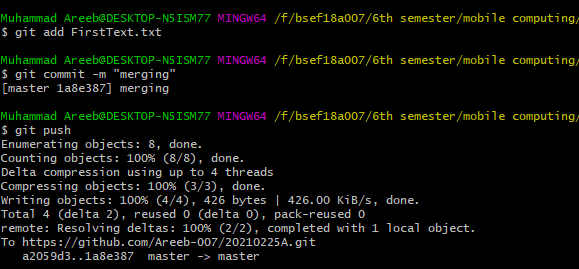
Merge Coflicts occur when we try to change same line from local and central repository at the same time Above are some changes from central repository.



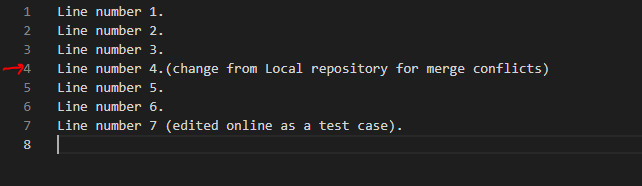
Above are some changes from local repository. Now when ever you try to add or push the file it will produce an error. 

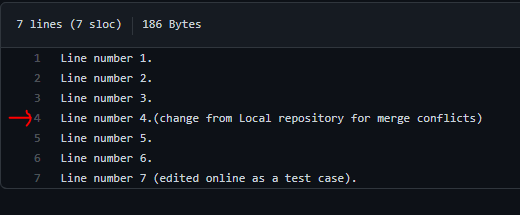
To solve this issue we have to choose one change either from local repository or central repository. Now when you run the command git pull we can see some following things. 

So we have to select one change from those. When we select one of those and then retry to add ,commit and push we get this.

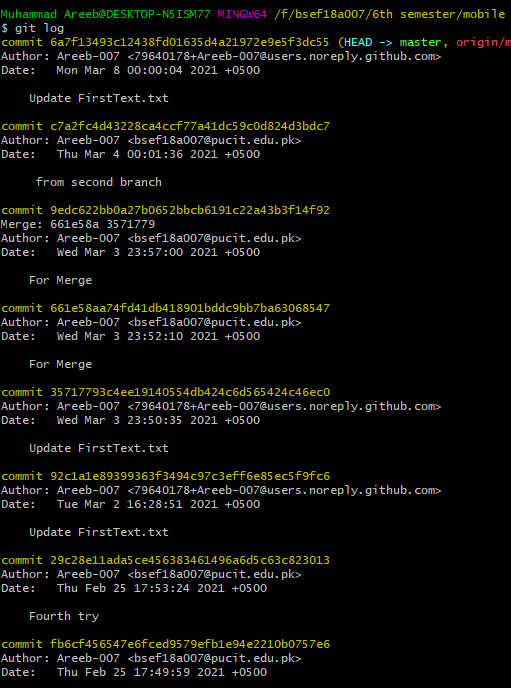


From Local Repository:



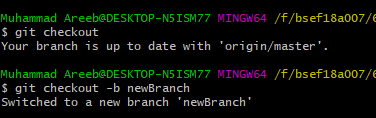
From Central Repository:  


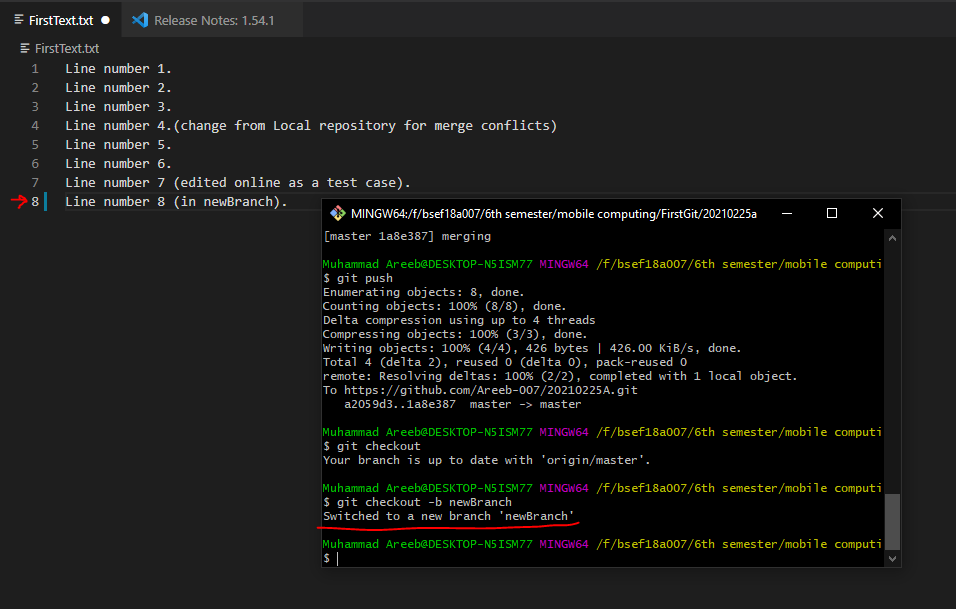
1. **Git Log**

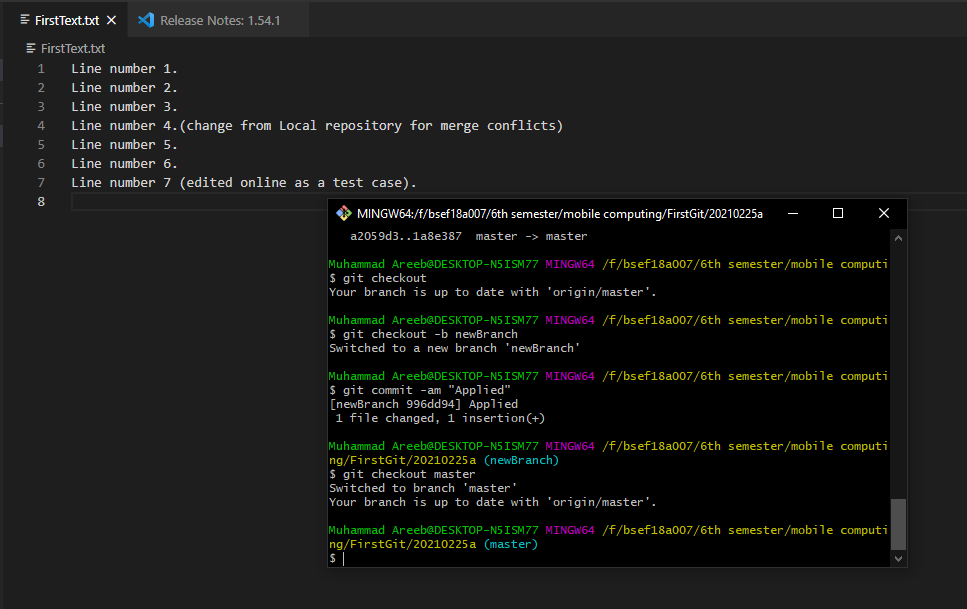
In order to see the commit details we can use “git log” command. 

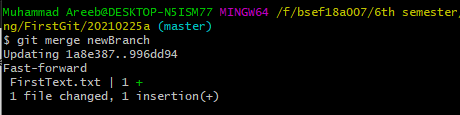
1. **Branching**

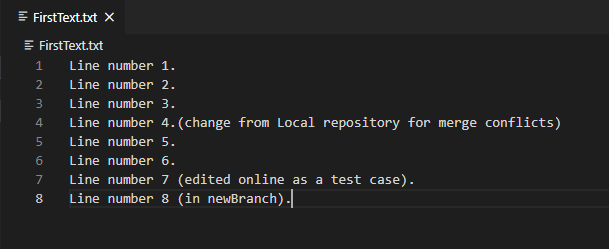
Branching is a very helpful and important feature. It is used to perform working in a parallel mode.

* **To Create a new Branch**
* Now if we edit the file it will be edited in this branch but whenever we shift to the other branch there will be nothing changed, which will be seen in the following pictures:



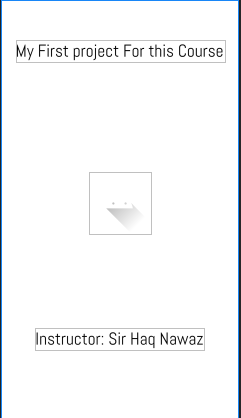


Now in order to merge those branches we can use “git merge branchname” command

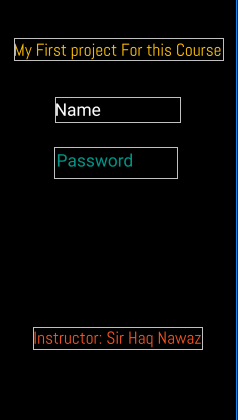
Content in the file are as follows now  


**Android Project**

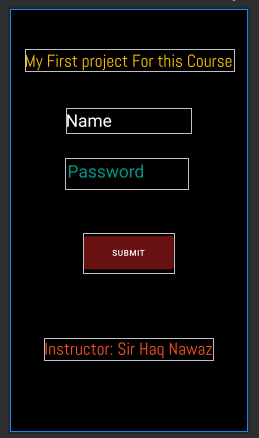
1. **First Attempt**



1. **Second Attempt**



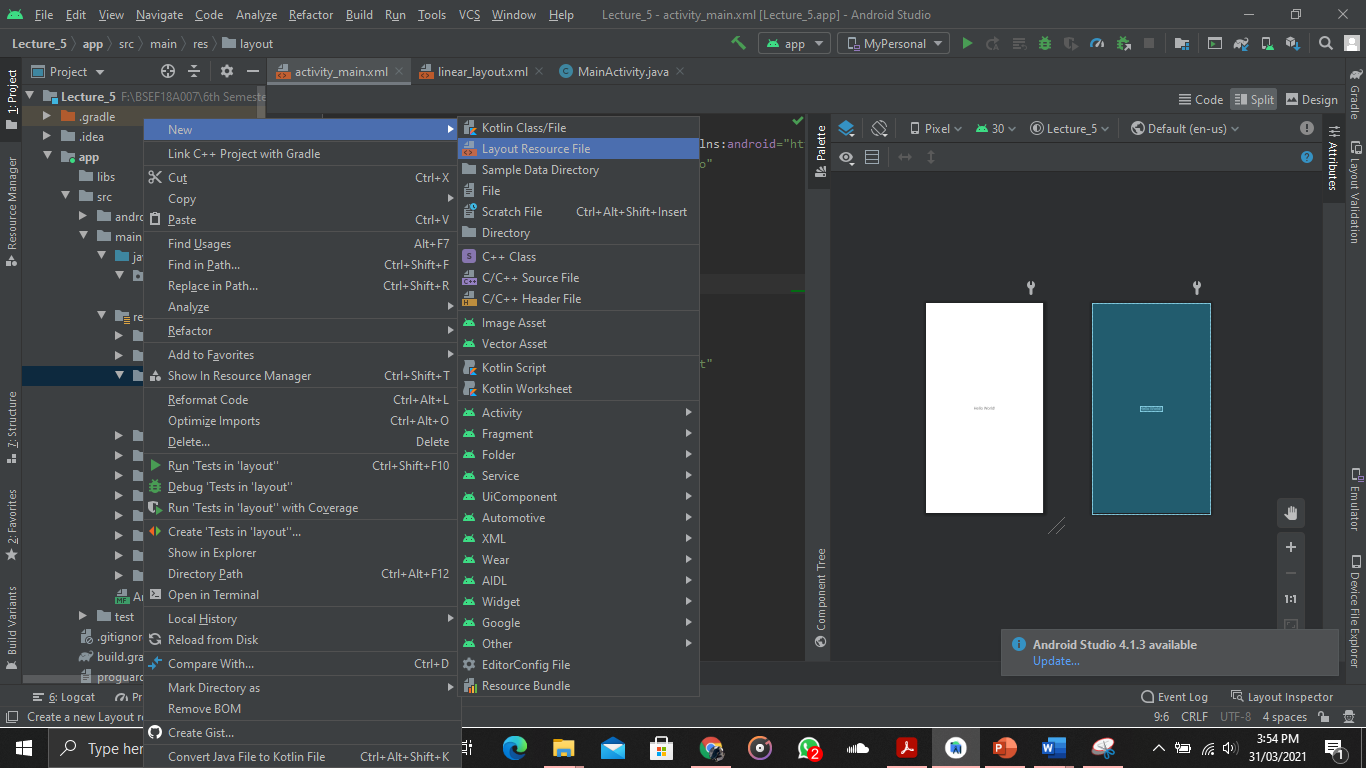
1. **Third Attempt**

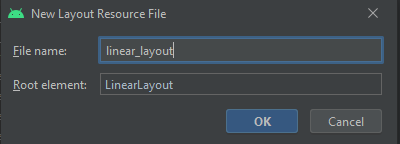


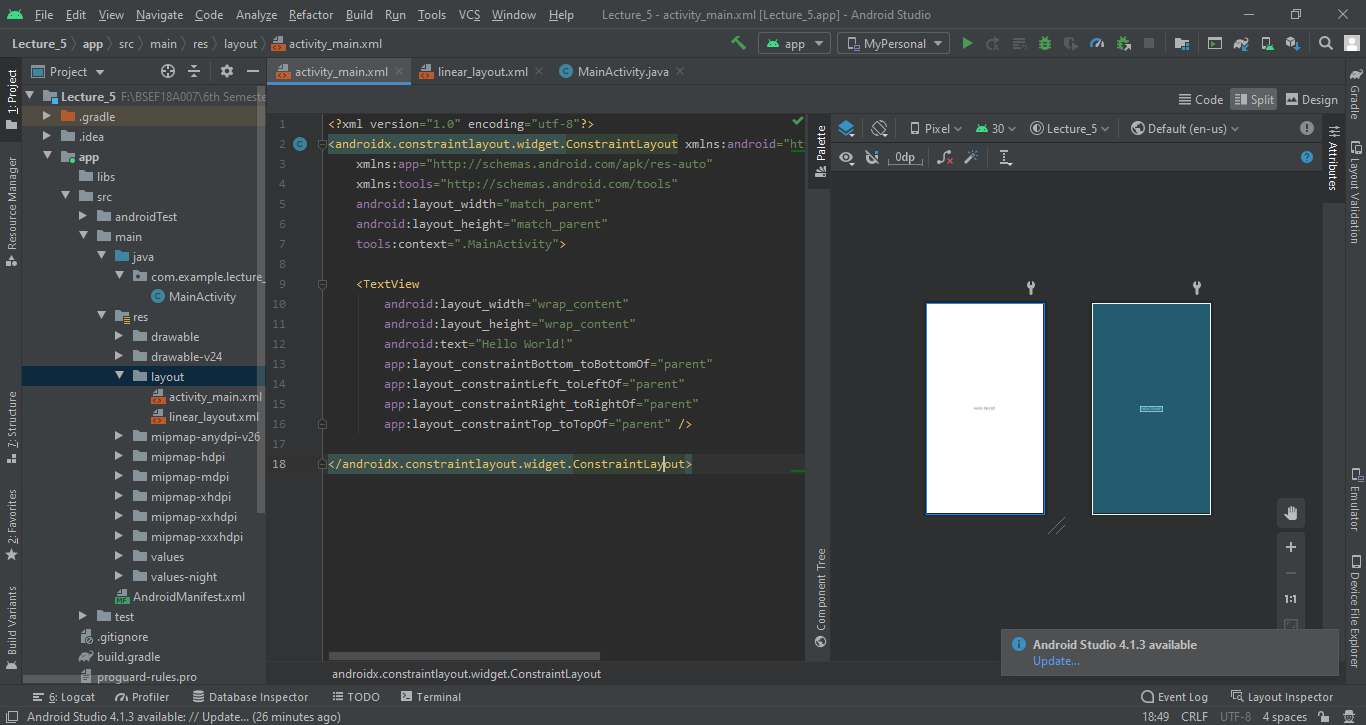
**Views and Layouts**

There are multiple views and view groups for example the button view, text view, list view etc. All these views are grouped in the Layout. There are multiple layouts available in the android Like Linear layout, relative layout, grid layout etc. But every layout must need to be the part of a root layout. All other things are the child of the root layout. So here is an example of how to do it.

At First I added another layout as “linear\_layout” (naming conventions must be followed)

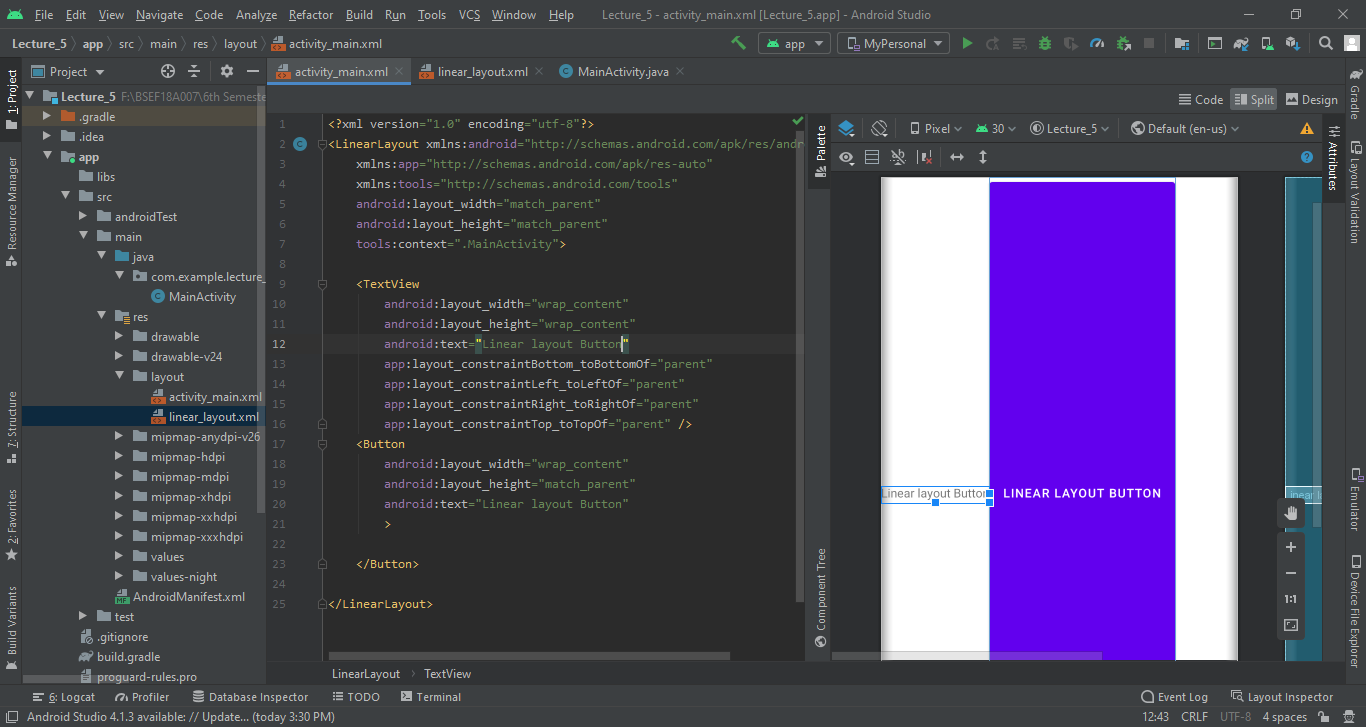


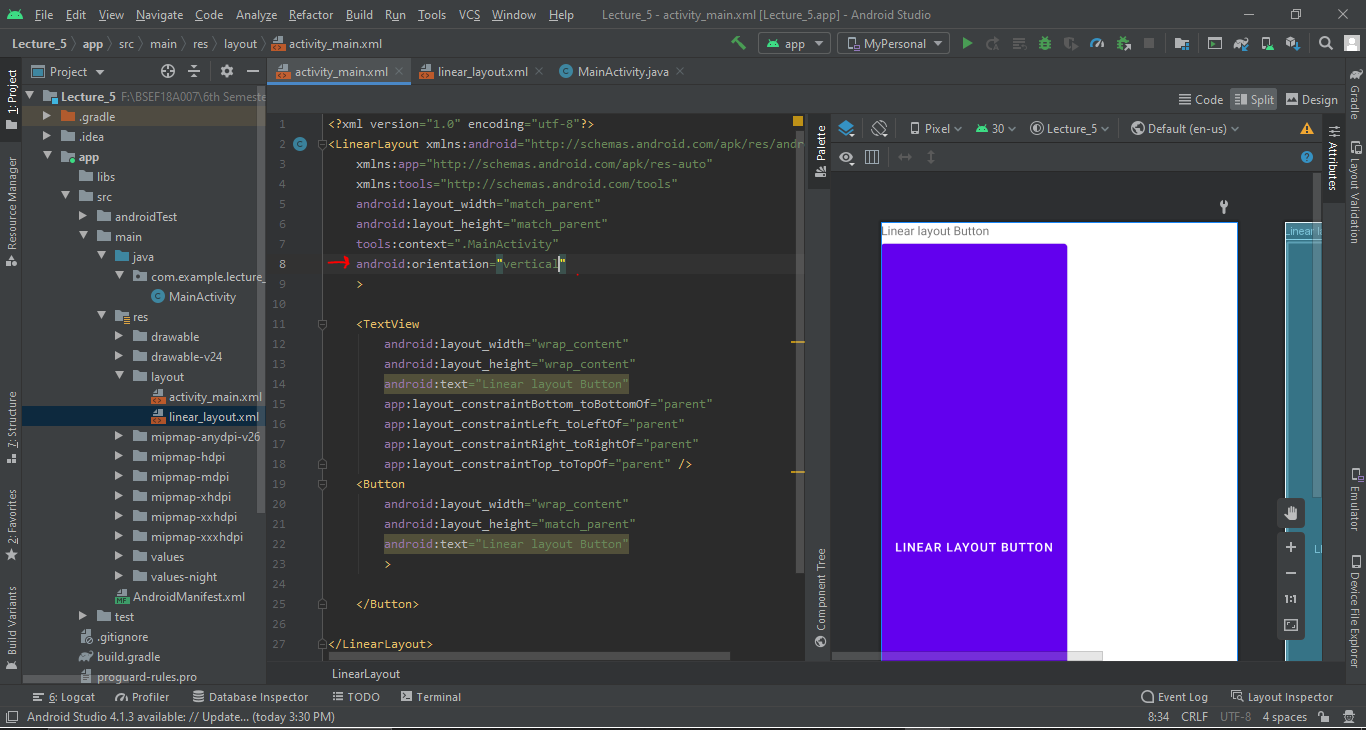
That is how we are going to add a new layout in our project. 

And then we can see that the layout is there 

As you can see I am using split mode it is very helpful in setting costraints layouts and items with in it.

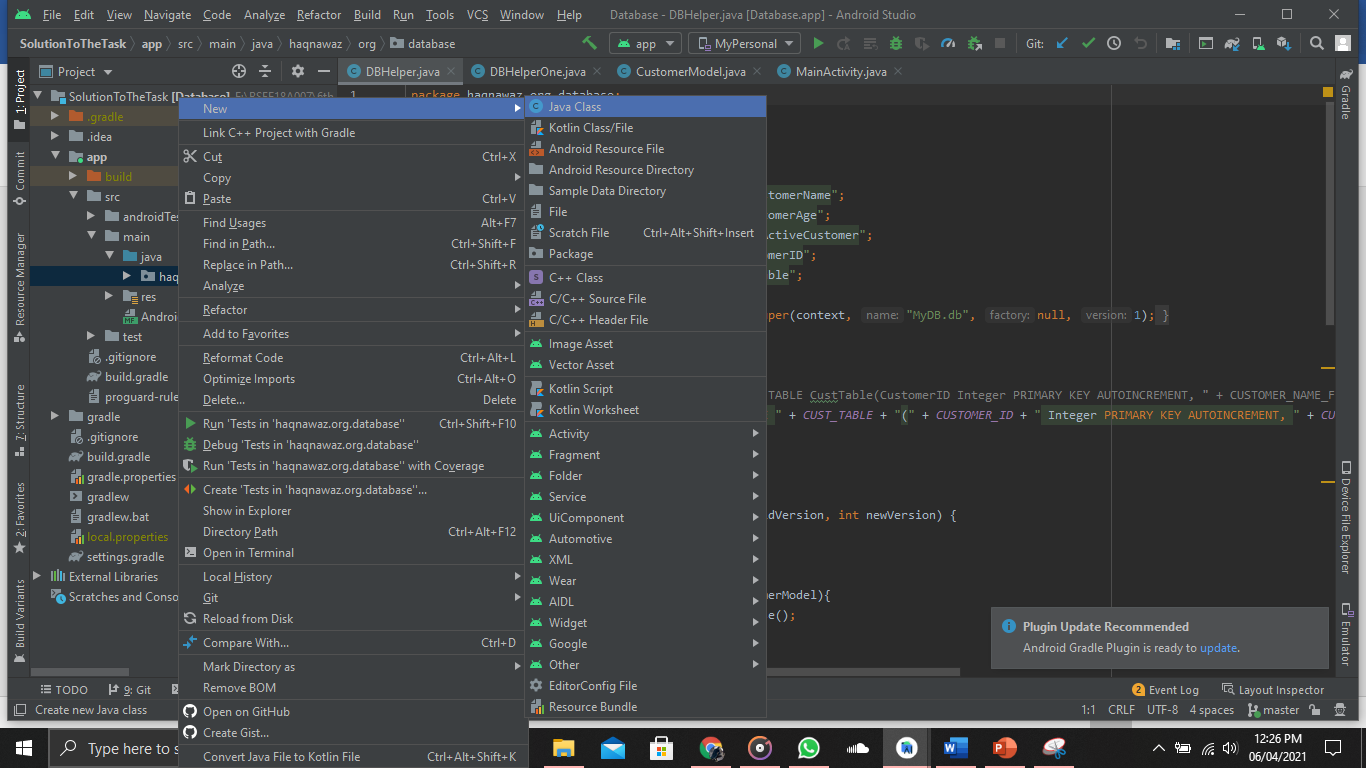
So if I add Text view and a button It will look like this



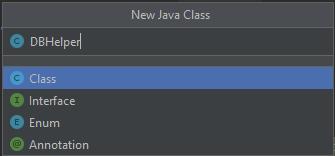
Now if I add the property of orientation vertically it will look like this. 

Now I am going to add some widgets in our pervious layout (Constraint Layout) and see what happens if we manage them.  
 -----------------------------------------------------------------------------------------------------------------------------------------

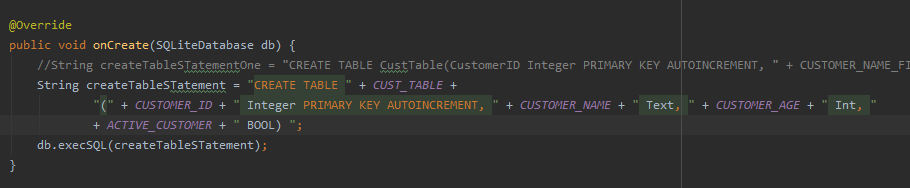
**Database Manipulation**

In order to save some thing on the database we have to create a Database on our android device so that we can use the functionality of the databse. The database we are going to use is the SQLite Database which is provided by the Android studio. So in order to create the database we must define a database handler class. 

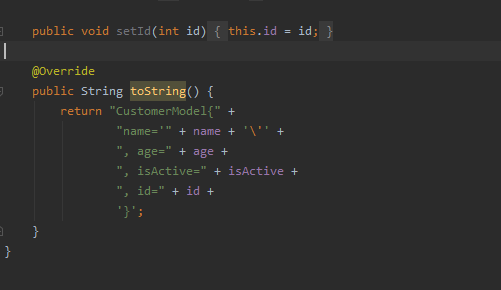
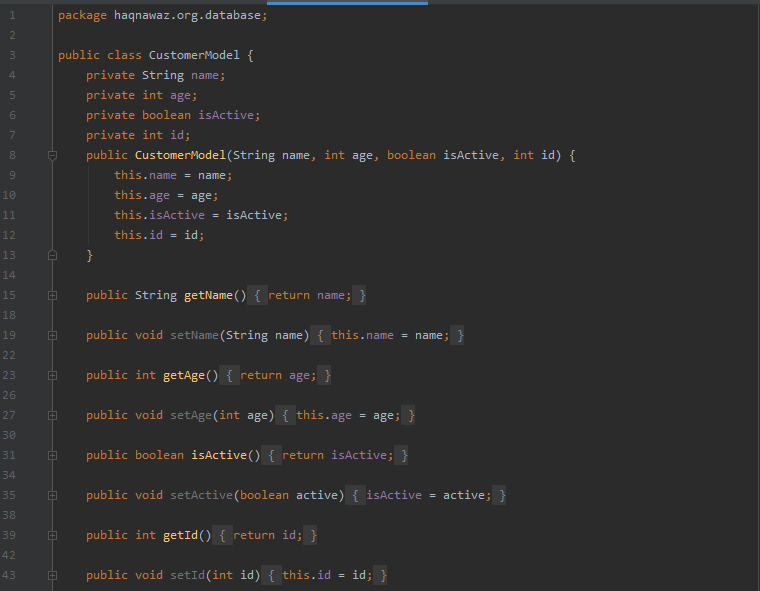
When we click on that it will ask us for the Class Name We have to name it as “DBhelper” which is playing the role as the DB handler Class.



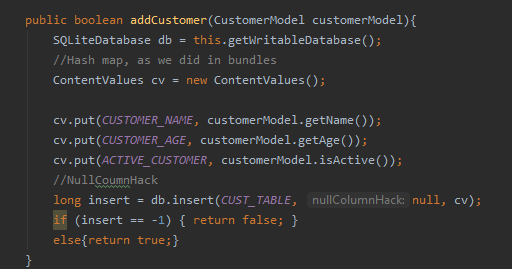
This will create a java file where we will implement our Database handling code. We will create a class with the name of DBHelper and will iherit with the SqliteOpenhelper Class.This will help us to use the sqlite functions like oncreate and onupdate which are used for some other purposes.   
In the On create method we will write the Query to create a table we want to use.



Then we will create an onther java class just like the same way we did it preciously to define a customer model which we will add in the database table customer. Also we have declared some static variable for name of table and columns of the table (which are the attributes of the class we are going to implement).

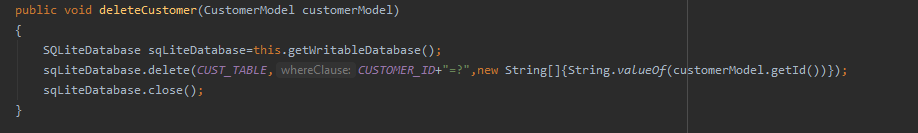
Java Class with desired attribute and their getter and setter as well

Now the Add Customer method in the DB Helper so that the it can add the customers In the Database



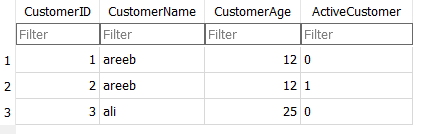
Now the get all customer method so that we can view it on the App

Now a delete method so that it can delete an item from the database .



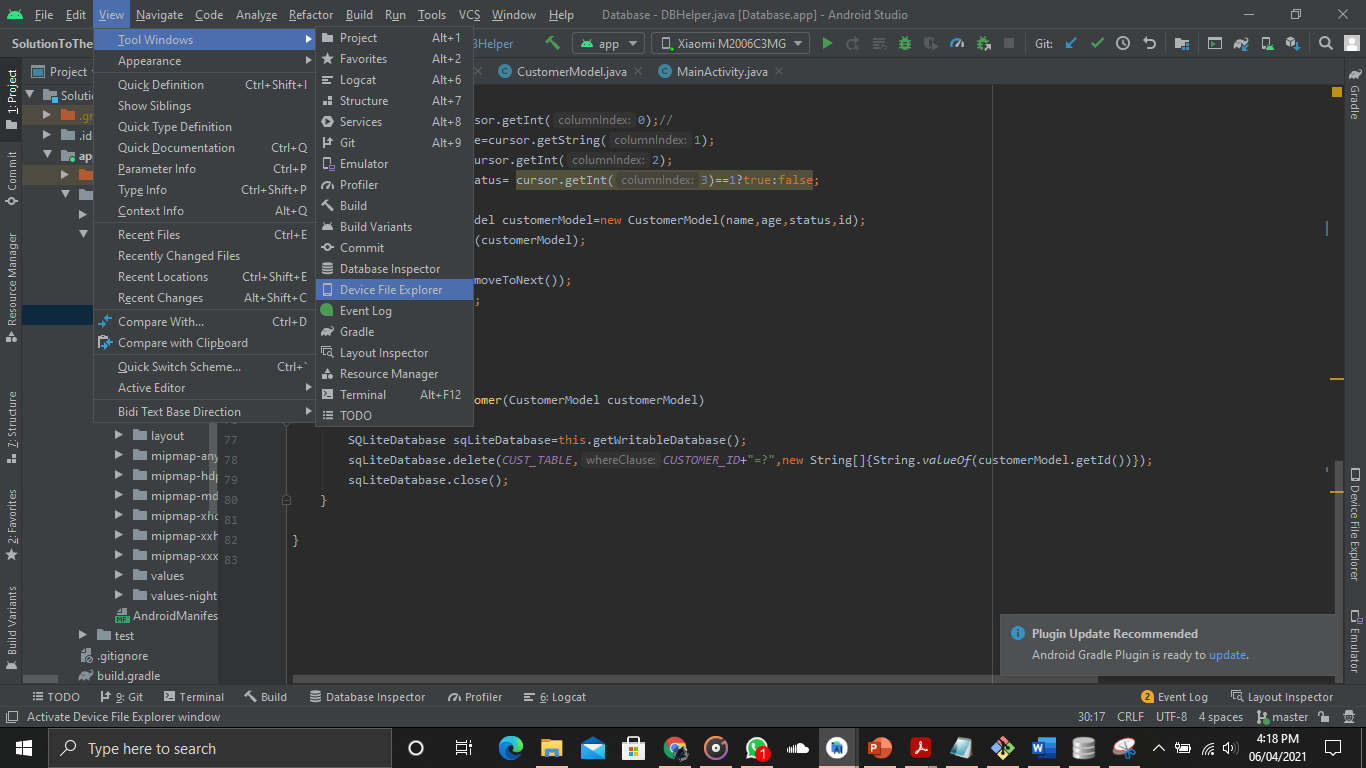
Though the above method can also be implemented using the query method as well.

Below we can see the customer Table in the Database using DB Browser Sql Lite

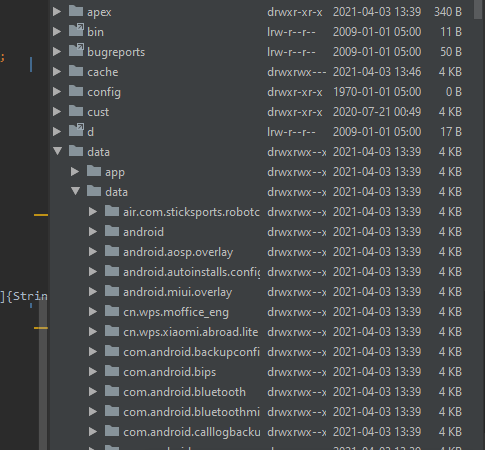


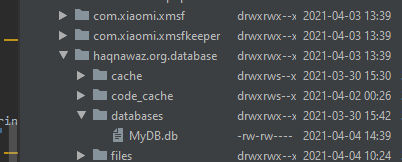
In order to use this we have to generate the database file first which we can do in the followig method:

Go the View->Tools Window->Device Explorer and then click

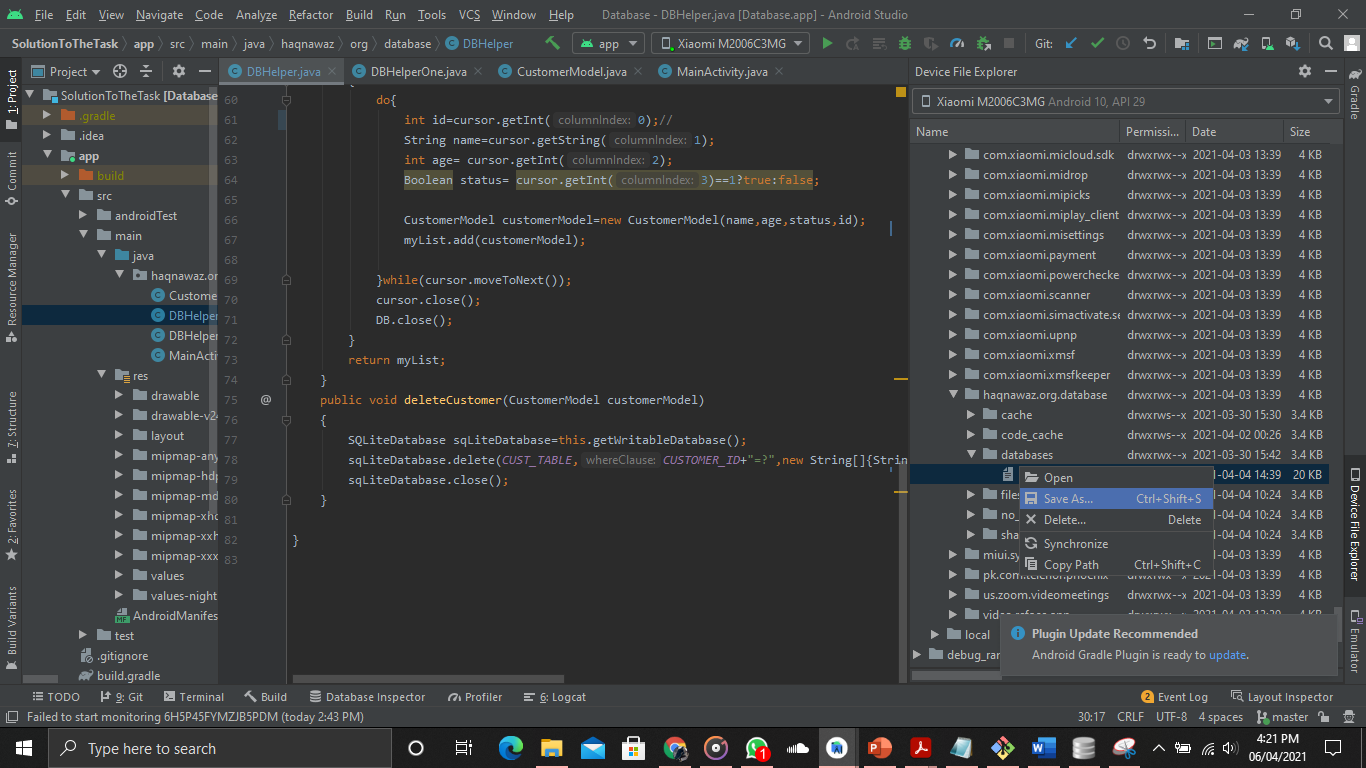


Then data-> data-> your package name -> databases -> your database

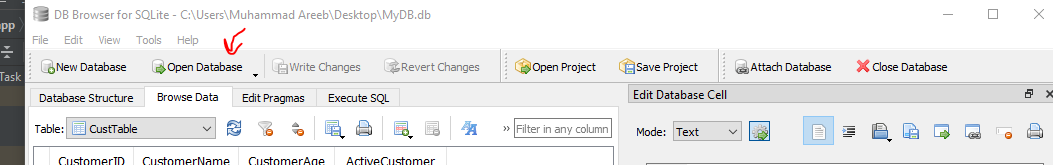




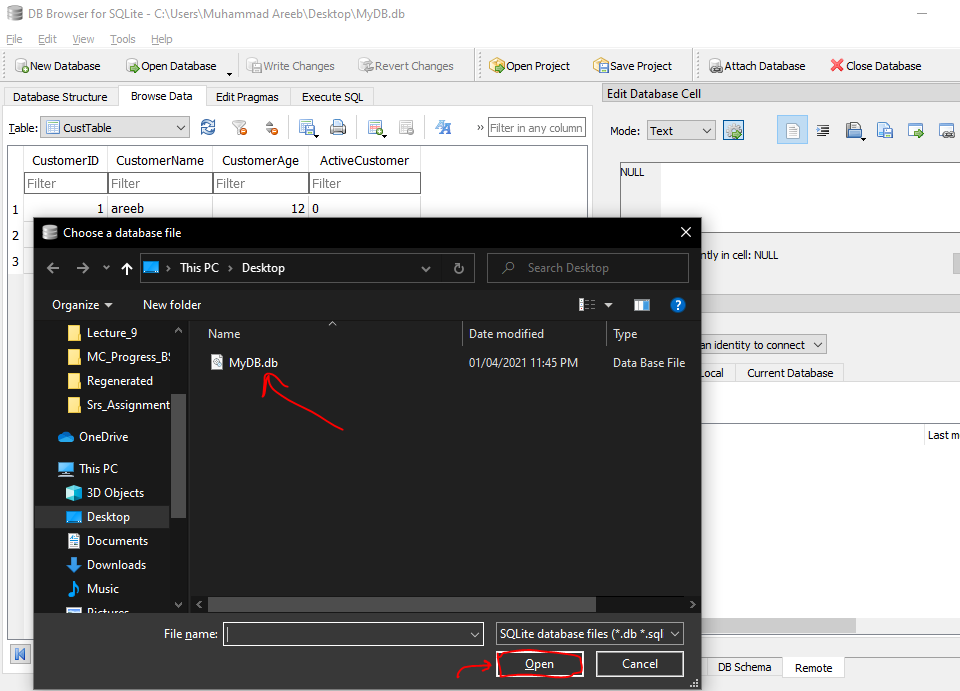
The right click on the database and save as name.db on any desired location you want.



Now open you DB Browser and click open database , browse your db file and then click on it



It will view it on the db browser application



Now here comes Main activity code for the database lecture

